เอกสารประกอบการสอน

วิชาการออกแบบคลังข้อมูล

DATA WAREHOUSE DESIGN

อาจารย์โกเมศ อัมพวัน

คณะวิทยาการสารสนเทศ

มหาวิทยาลัยบูรพา

สารบัญ

คำนำ

เอกสารประกอบการสอนนี้จัดทำขึ้นเพื่อใช้ในการเรียนการสอนวิชาการออกแบบคลังข้อมูล ซึ่งเป็นวิชาที่ประกอบไปด้วยภาคทฤษฎีและภาคปฏิบัติ โดยเนื้อหาในวิชาจะเน้นที่การสร้างระบบสนับสนุนการตัดสินใจรูปแบบหนึ่งที่เรียกว่า **“คลังข้อมูล”** โดยระบบนี้จะเป็นระบบที่ใช้สร้างหรือจัดเตรียมข้อมูลเชิงกลยุทธ์เพื่อนำไปประกอบการตัดสินใจต่อไป เนื้อหาในเอกสารประกอบการสอนนี้ได้ถูกจัดทำขึ้นใหม่ เนื่องจากปัจจุบันนิสิตยังขาดเอกสารที่ใช้ในการอ่านเพื่อทำความเข้าใจเนื้อหาต่างๆ ของรายวิชา ตำราส่วนใหญ่จะเป็นภาษาอังกฤษที่มีเนื้อหาที่ค่อนข้างจะหลากหลายและมีราคาค่อนข้างสูง

ด้วยเหตุดังกล่าว ผู้เขียนจึงได้เรียบเรียงเนื้อหาจากตำราต่างๆ ให้สอดคล้องกับเนื้อหาของรายวิชาเพื่อให้นิสิตได้ใช้อ่านประกอบการเรียนและสามารถทำความเข้าใจในเนื้อหาต่างๆ ได้มากยิ่งขึ้น

เอกสารประกอบการสอนนี้จะประกอบไปด้วยเนื้อหาที่ครอบคลุมเกี่ยวกับความต้องการและความจำเป็นในการสร้างคลังข้อมูล นิยามและส่วนประกอบของคลังข้อมูล การเก็บรวมรวมความต้องการของผู้ใช้งานคลังข้อมูล การออกแบบการสร้างและวิธีการสร้างคลังข้อมูล และปัจจัยต่างๆ ที่เกี่ยวข้องกับคลังข้อมูลตามลำดับ ซึ่งจากเนื้อหาข้างต้น ผู้เขียนหวังว่าเอกสารประกอบการสอนเล่มนี้จะเป็นประโยชน์ในการเรียนการสอนวิชาการออกแบบคลังข้อมูลของนิสิตคณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพาและผู้สนใจทั่วไป รวมทั้งเป็นพื้นฐานในการเรียนวิชาอื่นๆ เช่น การทำเหมืองข้อมูล โครงงานทางด้านวิทยาการคอมพิวเตอร์ และอื่นๆ รวมถึงสามารถนำไปประยุกต์ใช้ในการทำงานสืบไป

โกเมศ อัมพวัน
คณะวิทยาการสารสนเทศ
มหาวิทยาลัยบูรพา

CHAPTER 1

ความต้องการในการสร้างคลังข้อมูล

SECTION 1

แผนการสอนประจำบท

วัตถุประสงค์ของบทเรียน

- เข้าใจเกี่ยวกับความต้องการข้อมูลเชิงกลยุทธ์
- รู้จักปัญหาวิกฤตข้อมูล (Information crisis) ที่ทุกองค์กรอาจจะประสบพบเจอ
- สามารถแยกความแตกต่างระหว่างระบบการดำเนินงานและระบบสารสนเทศได้
- เข้าใจถึงเหตุผลเกี่ยวกับสร้างคลังข้อมูล
- เข้าใจเกี่ยวกับการทำธุรกิจอย่างชาญฉลาด (Business intelligence) สำหรับองค์กรต่างๆ

เนื้อหาของบทเรียน

เนื้อหาในบทนี้จะประกอบด้วย สาเหตุของความต้องการในการสร้างคลังข้อมูล ปัจจัยที่ต้องคำนึงถึงก่อนการตัดสินใจสร้างคลังข้อมูล คุณลักษณะและข้อแตกต่างระหว่างระบบการดำเนินงานและระบบสนับสนุนการตัดสินใจ นิยามและภาพรวมของคลังข้อมูล และการทำธุรกิจอย่างชาญฉลาด

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารประกอบการสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

กิจกรรมการเรียน-การสอน

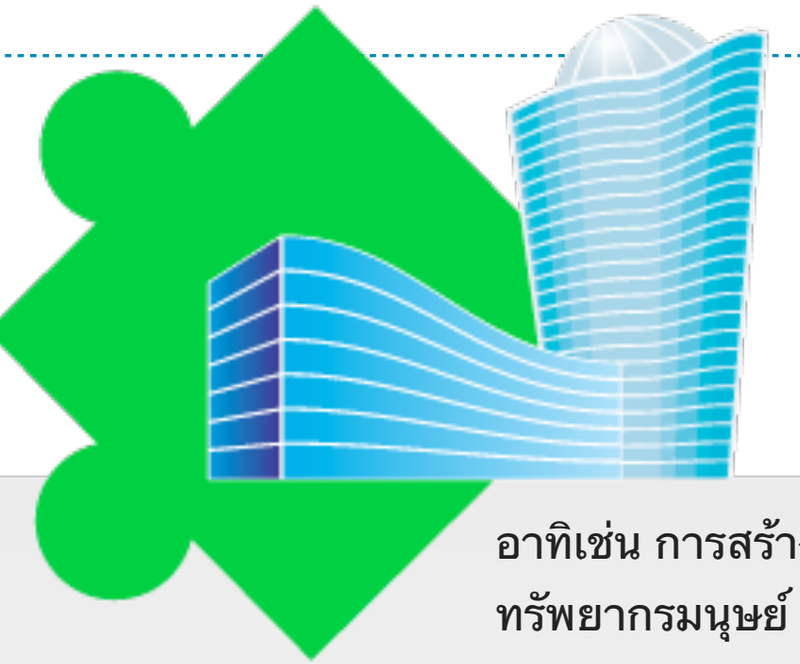
- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารประกอบการสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

การวัดและประเมินผล

- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

SECTION 2

บทนำ



ในยุคปัจจุบันเป็นยุคที่การดำเนินธุรกิจการค้ามีการแข่งขันกันค่อนข้างสูงจึงเป็นเหตุให้หลายธุรกิจให้ความสนใจกับการประยุกต์ใช้ข้อมูลข่าวสาร และการประยุกต์ใช้เทคโนโลยีคอมพิวเตอร์หรือระบบพื้นฐานต่างๆ เพื่อช่วยในการดำเนินธุรกิจ

อาทิเช่น การสร้างระบบเพื่อสนับสนุนการสั่งสินค้า ระบบบัญชี การจัดการคลังสินค้า การจ่ายเงิน การจัดการทรัพยากรมนุษย์ การตรวจสอบข้อมูล และอื่นๆ

ระบบเหล่านี้จะเป็นระบบที่ใช้ในการดำเนินงานในแต่ละวันที่จะทำให้สามารถลดความยุ่งยากในการดำเนินธุรกิจ มีความสะดวกสามารถดำเนินการได้อย่างรวดเร็ว และสามารถลดค่าใช้จ่ายได้ จากข้อดีที่มีอยู่ค่อนข้างมากของการประยุกต์ใช้เทคโนโลยีในการดำเนินธุรกิจ จึงเป็นเหตุให้บริษัทจำนวนมากได้เริ่มสร้างระบบพื้นฐานในการดำเนินธุรกิจตั้งแต่ปี 1960 ซึ่งนับตั้งแต่ยุคนั้นระบบพื้นฐานมีการเจริญเติบโตเป็นอย่างมากจนมีแอพพลิเคชันพื้นฐานถูกพัฒนาขึ้นเป็นจำนวนหลายร้อยแอพพลิเคชัน ต่อมาในปี 1990 การดำเนินธุรกิจมีการเจริญเติบโตและมีความซับซ้อนมากยิ่งขึ้นโดยหลายๆธุรกิจได้มีการกระจายการดำเนินธุรกิจไปยังต่างประเทศและภูมิภาคต่างๆ ทั่วโลก การทำธุรกิจในลักษณะนี้จะทำให้เกิดการแข่งขันทางการตลาดที่สูงขึ้นทั้งจากภายในประเทศและภายนอกประเทศ

เมื่อองค์กรหนึ่งๆต้องเผชิญกับสภาวะการแข่งขันทางการค้าที่ค่อนข้างสูง จึงเป็นเหตุให้ผู้บริหารขององค์กรนั้นอาจมีความต้องการ**ข้อมูลข่าวสาร**เพื่อช่วยประกอบการตัดสินใจในการดำเนินการต่างๆ เพื่อช่วงชิงส่วนแบ่งทางการตลาดจากคู่แข่งทางการค้าทั้งภายในและภายนอก จากความต้องการดังกล่าว จึงให้เกิดแนวคิดที่จะประยุกต์ใช้ระบบคอมพิวเตอร์เพื่อช่วยในการสร้างหรือจัดเตรียมข้อมูลข่าวสารที่จำเป็นต่อการดำเนินธุรกิจนั้นๆ โดยข้อมูลข่าวสารที่ผู้บริหารต้องการมักจะเป็นข้อมูลที่สามารถช่วยในการตัดสินใจเชิงกลยุทธ์และมีความแตกต่างจากข้อมูลข่าวสารทั่วๆ ไป อาทิเช่น ผู้ที่มีอำนาจตัดสินใจอาจต้องการที่จะทราบถึงข้อมูลพื้นที่ทางภูมิศาสตร์ที่มีการผลิตสินค้าชนิดต่างๆ ของบริษัท ข้อมูลนี้อาจช่วยให้ผู้บริหารสามารถทำการตัดสินใจเกี่ยวกับการขยายฐานกำลังการผลิตและการดำเนินการอื่นๆได้

จากตัวอย่างเราจะสามารถเรียกข้อมูลเหล่านี้ว่า ***"ข้อมูลเชิงกลยุทธ์ (Strategic information)"*** ที่จะต้องมีทั้ง**ความถูกต้องแม่นยำ**และ**มีรูปแบบที่เหมาะต่อความต้องการ**ของผู้บริหาร จากความต้องการข้อมูลเชิงกลยุทธ์ที่จะสามารถช่วยให้ผู้บริหารทำการติดสินใจต่าง ๆ จึงได้เกิดแนวคิดที่จะทำ ***"การสร้างคลังข้อมูล (Data warehousing)"*** ที่ซึ่งเป็นกรอบหรือแบบจำลองสำหรับการจัดเตรียม/ค้นหาข้อมูลเชิงกลยุทธ์ โดยคลังข้อมูลมีการเริ่มสร้างขึ้นตั้งแต่ปี 1990 เพื่อช่วยเหลือผู้บริหารให้ได้รับข้อมูลที่ใช้ประกอบการตัดสินใจเชิงกลยุทธ์ได้

รูปที่ 1-1 ตัวอย่างองค์กรที่มีการสร้างคลังข้อมูล

จากรูปที่ 1-1 จะเป็นการแสดงตัวอย่างธุรกิจที่มีการสร้างคลังข้อมูลเพื่อช่วยในการดำเนินธุรกิจ อาทิเช่น ธุรกิจค้าปลีกที่ต้องการใช้คลังข้อมูลเพื่อให้ข้อมูลเชิงกลยุทธ์สำหรับรักษาฐานลูกค้า และวางแผนเกี่ยวกับการตลาด ธุรกิจการเงินจะใช้ข้อมูลเชิงกลยุทธ์เพื่อใช้ในการจัดการกับความเสี่ยงต่างๆ ที่อาจเกิดขึ้นกับบริษัท และเพื่อตรวจสอบ/ค้นหาธุรกรรมทางการเงินที่มีความผิดปกติ ธุรกิจสายการบินจะใช้ข้อมูลเชิงกลยุทธ์เพื่อสร้างผลกำไรจากเส้นทางการบินต่างๆ และการจัดการผลตอบแทนต่างๆ ธุรกิจที่มีการผลิตสินค้าจะใช้ข้อมูลเชิงกลยุทธ์เพื่อช่วยลดต้นทุนและช่วยในเรื่องของการจัดการเกี่ยวกับโลจิสติก ธุรกิจที่เกี่ยวข้องกับสาธารณูประโภคจะใช้ข้อมูลเชิงกลยุทธ์ในการจัดการทรัพย์สินและทรัพยากรต่างๆ และท้ายสุดรัฐบาลจะใช้ข้อมูลเชิงกลยุทธ์เพื่อทำการวางแผนเกี่ยวกับพนักงาน/กำลังคนและการควบคุมค่าใช้จ่าย เป็นต้น

SECTION 3

ความต้องการข้อมูลเชิงกลยุทธ์

ในการสร้างระบบสำหรับสร้างหรือจัดเตรียมข้อมูลเชิงกลยุทธ์ เราจะต้องทำการศึกษาปัจจัยต่างๆ เป็นจำนวนมาก อาทิเช่น

- โอกาสที่จะได้รับข้อมูลเชิงกลยุทธ์จากระบบที่สร้างขึ้นมีมากน้อยเพียงใด?
- โอกาสที่การสร้างระบบคลังข้อมูลจะประสบความสำเร็จมีมากน้อยเพียงใด?
- ความเสี่ยงที่บริษัทจะต้องพบเจอเมื่อไม่มีข้อมูลเชิงกลยุทธ์สำหรับช่วยในการตัดสินใจเป็นอย่างไร?
- พิจารณาว่าข้อมูลใดที่จะสามารถเป็นข้อมูลเชิงกลยุทธ์ได้บ้าง? ผู้ที่ต้องการข้อมูลเชิงกลยุทธ์เป็นใคร?
- ผู้ใช้ระบบแต่ละคนต้องการข้อมูลอะไรบ้าง?

ในองค์กรใหญ่ๆทั่วๆไป ผู้บริหาร ผู้จัดการหรือผู้ที่มีอำนาจในการตัดสินใจมักจะต้องการข้อมูลเพื่อนำไปประกอบตัดสินใจที่จะวางกลยุทธ์ทางธุรกิจ ตั้งเป้าหมาย กำหนดวัตถุประสงค์ และทำการเฝ้าดูผลลัพธ์เพื่อเพิ่มโอกาสในการแข่งขันทางการค้ากับบริษัทอื่นๆ โดยเป้าหมายจะที่สามารถกำหนดได้จะมีตัวอย่างดังต่อไปนี้

การรักษาฐานลูกค้าปัจจุบัน

การได้รับส่วนแบ่งทางการตลาด
เพิ่มขึ้น 10% ภายใน 3 ปีข้างหน้า

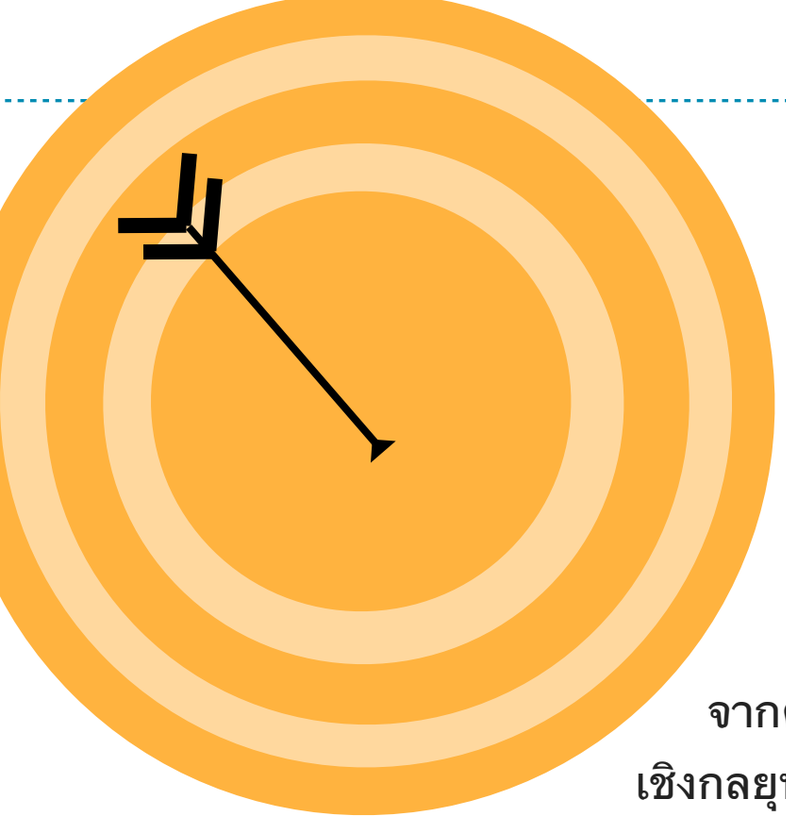
การเพิ่มฐานลูกค้า 15% จากเดิมภายใน 5 ปีข้างหน้า

การเพิ่มคุณภาพของการบริการ
จัดส่งสินค้าให้กับลูกค้า

การนำผลิตภัณฑ์ใหม่ 3 รายการ
ออกสู่ท้องตลาดใน 2 ปีข้างหน้า

การเพิ่มคุณภาพของสินค้าให้ไปติดอยู่ใน
1 ใน 5 ของสินค้าที่ดีที่สุดในสินค้ากลุ่มเดียวกัน

การเพิ่มยอดขาย 15%
ในเขตตะวันออกเฉียงเหนือ



จากตัวอย่างเป้าหมายข้างต้น ผู้บริหารและผู้ที่มีอำนาจในการตัดสินใจอาจต้องการข้อมูลเชิงกลยุทธ์ด้วยเหตุผลหลายประการด้วยกัน อาทิเช่น

(1) เพื่อที่จะทราบถึงการดำเนินธุรกิจในเชิงลึก
(2) เพื่อที่จะสามารถแสดงความคิดเห็นและตรวจสอบตัวชี้วัดประสิทธิภาพที่สำคัญและทราบถึงผลกระทบของตัวชี้วัดเหล่านั้นในการดำเนินธุรกิจ
(3) เพื่อทำการติดตามความเปลี่ยนแปลงในธุรกิจว่าจะมีความเปลี่ยนแปลงอย่างไรเมื่อเวลาล่วงเลยผ่านไป
(4) เพื่อทำการเปรียบเทียบประสิทธิภาพการทำงานของบริษัทกับบริษัทคู่แข่งและมาตราฐานของอุตสาหกรรม/การทำธุรกิจของบริษัท และอื่นๆ

นอกจากการดำเนินกิจกรรมต่างๆ ผู้บริหารระดับสูงอาจให้ความสนใจเกี่ยวกับข้อมูลที่เกี่ยวข้องกับปัจจัยอื่นๆ อีกมากมาย ตัวอย่างเช่น

1. ความต้องการและความชอบของลูกค้า
2. เทคโนโลยีใหม่ ๆ
3. ยอดขายและส่วนแบ่งทางการตลาด
4. คุณภาพของสินค้าและบริการ

ข้อมูลเหล่านี้จะเป็นข้อมูลเชิงกลยุทธ์ที่มีความสำคัญหรือมีอิทธิพลต่อการกำหนดกลยุทธ์แบบแผนและการดำเนินการตามกลยุทธ์ที่วางไว้ และจะสามารถช่วยให้การดำเนินธุรกิจนั้นเป็นไปอย่างมีประสิทธิภาพมากขึ้น

นอกจากนั้นข้อมูลเชิงกลยุทธ์ยังอาจจะส่งผลถึงความอยู่รอดขององค์กรได้อีกด้วย โดยข้อมูลเชิงกลยุทธ์นั้นจะมีคุณสมบัติหลายประการด้วยกัน ดังแสดงในรูปที่ 1-2

INTEGRATED	Must have a single, enterprise-wide view.
DATA INTEGRITY	Information must be accurate and must conform to business rules.
ACCESSIBLE	Easily accessible with intuitive access paths, and responsive for analysis.
CREDIBLE	Every business factor must have one and only one value.
TIMELY	Information must be available within the stipulated time frame.

รูปที่ 1-2 คุณลักษณะของข้อมูลเชิงกลยุทธ์

โอกาสและความเสี่ยงขององค์กรที่มีและไม่มีข้อมูลเชิงกลยุทธ์

ถ้าบริษัทหนึ่งๆ มีการประยุกต์ใช้เทคโนโลยี คอมพิวเตอร์ และระบบพื้นฐานต่างๆ จะทำให้บริษัทนั้นสามารถดำเนินธุรกิจได้สะดวกรวดเร็ว และราบรื่น ซึ่งในการประยุกต์ใช้ระบบพื้นฐานเราอาจประยุกต์ใช้แอพลิเคชันต่างๆ เป็นจำนวนมาก และเมื่อเราทำการพิจารณาถึงรายละเอียดของระบบพื้นฐานและแง่มุมต่างๆ เราจะพบข้อเท็จจริง 2 ข้อด้วยกัน คือ

1 องค์กร/บริษัทต่างๆ มีการจดบันทึกข้อมูลค่อนข้างมาก

2 ข้อมูลที่ถูกเก็บไว้ไม่ได้มีการสรุปผลเพื่อสร้างเป็นข้อมูลเชิงกลยุทธ์เลย

จากข้อเท็จจริงทั้งสองจะเป็นสิ่งที่สะท้อนถึงปัญหาเกี่ยวกับการใช้ข้อมูลในองค์กรที่ไม่มีการประยุกต์หรือจัดทำ/สรุปผลข้อมูลเพื่อช่วยในการตัดสินใจเชิงกลยุทธ์ แต่อย่างไรก็ดี ณ ปัจจุบัน ปัญหาเกี่ยวกับการมีข้อมูลปริมาณมากแต่ไม่ได้ใช้ประโยชน์ได้อย่างเต็มที่ หรือที่เรียกว่า "***information crisis***" ได้จางหายไปจากองค์กรต่างๆ เนื่องจากบริษัทต่างๆ ได้ริเริ่มที่จะทำการประยุกต์ใช้เทคโนโลยีในการจัดเตรียมข้อมูลเชิงกลยุทธ์ ซึ่งข้อมูลเชิงกลยุทธ์ที่จะนำไปให้ผู้บริหารใช้ประกอบการตัดสินใจนั้น จะเป็นข้อมูลที่อยู่ในรูปแบบที่ง่ายต่อการวิเคราะห์ถึงแถวโน้มของข้อมูลที่เปลี่ยนแปลงตามกาลเวลา และเป็นข้อมูลสื่อถึงมุมมองหรือปัจจัยทางธุรกิจที่มีความหลากหลายและแตกต่างกันได้ ตัวอย่างเช่น ผู้บริหารสามารถทราบถึงข้อมูลยอดขายสินค้าของแต่ละรายการสินค้า ยอดขายของพนักงานขายแต่ละคน ยอดขายในแต่ละพื้นที่ ยอดขายในแต่ละกลุ่มลูกค้า และอื่นๆ ซึ่งถ้าผู้บริหารทราบเกี่ยวกับข้อมูลเหล่านี้จะทำให้ผู้บริหารทราบถึงข้อมูลในหลายๆ มุมมอง ซึ่งจะช่วยให้สามารถทำการตัดสินใจเลือกทิศทางในการดำเนินธุรกิจที่เหมาะสมได้

ดังนั้น ในการจัดเตรียมข้อมูลเชิงกลยุทธ์ เราจะต้องทำการศึกษาถึง**โอกาสที่บริษัทจะได้รับเมื่อมีการประยุกต์ใช้ข้อมูลเชิงกลยุทธ์** และ**ความเสี่ยงในการดำเนินธุรกิจของบริษัทเมื่อไม่มีข้อมูลเชิงกลยุทธ์** โดยทั้งสองปัจจัยนี้จะเป็นเหตุผลสำคัญในการตัดสินใจที่จะสร้างระบบหรือใช้เทคโนโลยีมาช่วยในการจัดเตรียมข้อมูลเชิงกลยุทธ์ เพื่อให้เข้าใจถึงโอกาสและความเสี่ยงที่จะเกิดขึ้นกับบริษัทต่างๆ ลองพิจารณาตัวอย่างดังนี้

โอกาสที่จะเกิดขึ้นกับบริษัทจากการใช้ข้อมูลเชิงกลยุทธ์

บริษัทที่เกี่ยวกับการให้บริการโทรศัพท์ทางไกลได้ให้อำนาจกับพนักงานขายสินค้าในการตัดสินใจต่างๆ ทางธุรกิจ และมีการประยุกต์ใช้ระบบที่มีการเชื่อมต่อกับเว็บเพื่อรวบรวมข้อมูลทั้งจากภายในและภายนอกเพื่อสร้าง/จัดทำข้อมูลเชิงกลยุทธ์ที่จะทำให้บริษัทสามารถเฝ้าดูหรือเฝ้าติดตามการดำเนินธุรกิจที่มีการแข่งขันกันสูงได้

ธนาคารที่ใหญ่ที่สุดในอเมริกาที่มีทรัพย์สินประมาณ 250,000,000,000 ดอลลาร์ จะใช้ข้อมูลเชิงกลยุทธ์ที่จะทำให้ผู้บริหารสามารถตัดสินใจดำเนินการต่างๆเพื่อรักษาฐานลูกค้าของพวกเขาได้อย่างรวดเร็ว

ในองค์กรที่ดูแลสุขภาพของประชาชนจะใช้ข้อมูลเชิงกลยุทธ์ในการปรับปรุงโปรแกรมการดูแลสุขภาพ ซึ่งจะทำให้ผู้คน 22% ลดการเข้าห้องฉุกเฉิน และ 29% ของเด็กไม่ต้องนอนโรงพยาบาลเพื่อรักษาโรคหืดหอบ

บริษัทค้าปลีกที่ใหญ่ติด 1 ใน 5 ของอเมริกานำข้อมูลเชิงกลยุทธ์ไปรวมกับเครื่องมือที่ใช้ในการวิเคราะห์ข้อมูลบนเว็บ (Web-enabled analysis tools) ซึ่งจากการรวมของเทคโนโลยีทั้งสองจะทำให้ร้านค้าต่างๆขององค์กรได้รับข้อมูลเชิงลึกของฐานลูกค้า มีกระบวนการจัดการคลังสินค้าที่ดีขึ้น จัดวางผลิตภัณฑ์ที่เหมาะสมให้อยู่ในตำแหน่งที่เหมาะสมและใน

ความเสี่ยงที่จะเกิดขึ้นกับบริษัทจากการไม่มีข้อมูลเชิงกลยุทธ์

บริษัทผู้ผลิตชิ้นส่วนรถยนต์และรถบรรทุกที่ไม่ได้มีการใช้ระบบสำหรับสร้าง/จัดเตรียมข้อมูลเชิงกลยุทธ์ ซึ่งจะทำให้บริษัทนั้นประสบพบเจอกับปัญหาต่างๆมากมาย เช่น ความไม่สอดคล้องกันของข้อมูลที่จัดเก็บไว้ในแต่ละโรงงาน การเสียเวลาในการเก็บข้อมูลด้วยมือ (manual) และการสิ้นเปลืองเวลาหลายสัปดาห์ในการจัดทำรายงานที่จะสามารถสนับสนุนการตัดสินใจ เป็นต้น

บริษัทขายยาที่มีห้างร้านมากกว่า 800 สาขาในพื้นที่ต่างๆ ใช้ข้อมูลเชิงกลยุทธ์ในการทำความเข้าใจถึงสิ่งที่ลูกค้าซื้อเป็นประจำ ซึ่งจะช่วยให้มีการพัฒนากระบวนการจัดเก็บยาในคลังยาให้ดีขึ้นช่วยเพิ่มประสิทธิภาพของการจัดทำโปรโมชั่นและการวางกลยุทธ์ทางการตลาด และช่วยเพิ่มผลกำไรให้กับบริษัทได้อีกด้วย

บริษัทสาธารณูปโภคที่ให้บริการไฟฟ้ากับลูกค้าประมาณ 25 ล้านครัวเรือน ไม่ได้ใช้ข้อมูลกลยุทธ์ในการยกเลิกกฎระเบียบบางประการ ที่จะทำให้มีผู้เสียประโยชน์เป็นจำนวนมากและผู้ได้ประโยชน์เพียงเล็กน้อยเท่านั้น

บริษัทเครื่องใช้ไฟฟ้าจะใช้ข้อมูลเชิงกลยุทธ์เพื่อช่วยให้การจัดการคลังสินค้าดีขึ้น ซึ่งจะทำให้สามารถประหยัดเงินได้หลายล้านดอลล่าร์ต่อปี

SECTION 4

การเปรียบเทียบระหว่างระบบการดำเนินงานและระบบสนับสนุนการตัดสินใจ

อย่างที่เราทราบกันดีว่าบริษัทและองค์กรต่างๆ ในยุคปัจจุบันจะมีการสร้างระบบการดำเนินงาน (Operational system) ต่างๆ มากมายเพื่อสนับสนุนการดำเนินธุรกิจในแต่ละวัน โดยระบบที่มักจะถูกสร้างขึ้นจะประกอบไปด้วย ระบบการสั่งซื้อสินค้า ระบบควบคุมคลังสินค้า ระบบการเคลมสินค้า และอื่นๆ โดยระบบที่ถูกสร้างขึ้นเหล่านี้จะไม่ได้มีเจตนาและไม่มีความสามารถในการสร้างหรือจัดเตรียมข้อมูลเชิงกลยุทธ์ แต่ถ้าเราต้องการที่จะทำการสร้างข้อมูลเชิงกลยุทธ์จากระบบการดำเนินงานเหล่านี้ เราจะต้องทำการเรียกดูข้อมูลหรือใช้ข้อมูลจากหลายๆ ระบบแล้วนำข้อมูลเหล่านั้นมารวมกันเพื่อสร้างเป็นระบบสนับสนุนการตัดสินใจหรือระบบสารสนเทศ แต่ก่อนที่เราจะทำการสร้างระบบสนับสนุนการตัดสินใจหรือระบบสารสนเทศ เราควรทราบถึงรายละเอียดต่างๆ เกี่ยวกับระบบทั้งสอง รวมถึงข้อแตกต่างระหว่างระบบทั้งสองด้วย

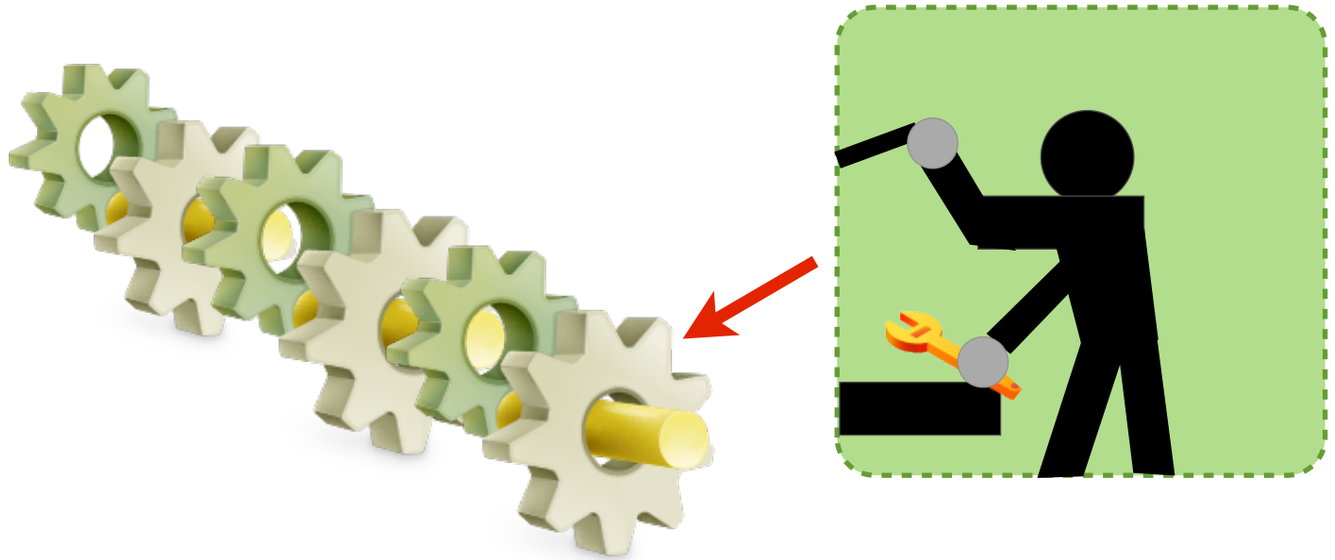
วัตถุประสงค์ของการสร้างระบบการดำเนินงาน

ระบบการดำเนินงานหรือที่เรียกว่า “***OnLine Transactional Processing (OLTP)***” เป็นระบบที่ใช้สนับสนุนการทำธุรกิจในแต่ละวัน ซึ่งสามารถกล่าวได้อีกอย่างหนึ่งว่าเป็นระบบที่ทำให้การดำเนินธุรกิจมีการเคลื่อนไหวดำเนินไปได้ซึ่งจะประกอบไปด้วยระบบพื้นฐานต่างๆ ดังแสดงตัวอย่างในรูปที่ 1-3

Get the data in

Making the wheels of business turn

- Take an order
- Process a claim
- Make a shipment
- Generate an invoice
- Receive cash
- Reserve an airline seat



รูปที่ 1-3 ตัวอย่างฟังก์ชันการทำงานของระบบการดำเนินงาน

ฟังก์ชันการทำงานหลักของระบบการดำเนินงานส่วนใหญ่จะยุ่งเกี่ยวกับการจัดเก็บหรือเรียกดูข้อมูลจากฐานข้อมูล ตัวอย่างเช่น การเก็บข้อมูลแต่ละรายการที่ดำเนินการทำกับธุรกิจ อาทิเช่น การสั่งสินค้า 1 ครั้ง การออกใบแจ้งหนี้ 1 รายการ และการเก็บข้อมูลลูกค้า 1 ราย เป็นต้น

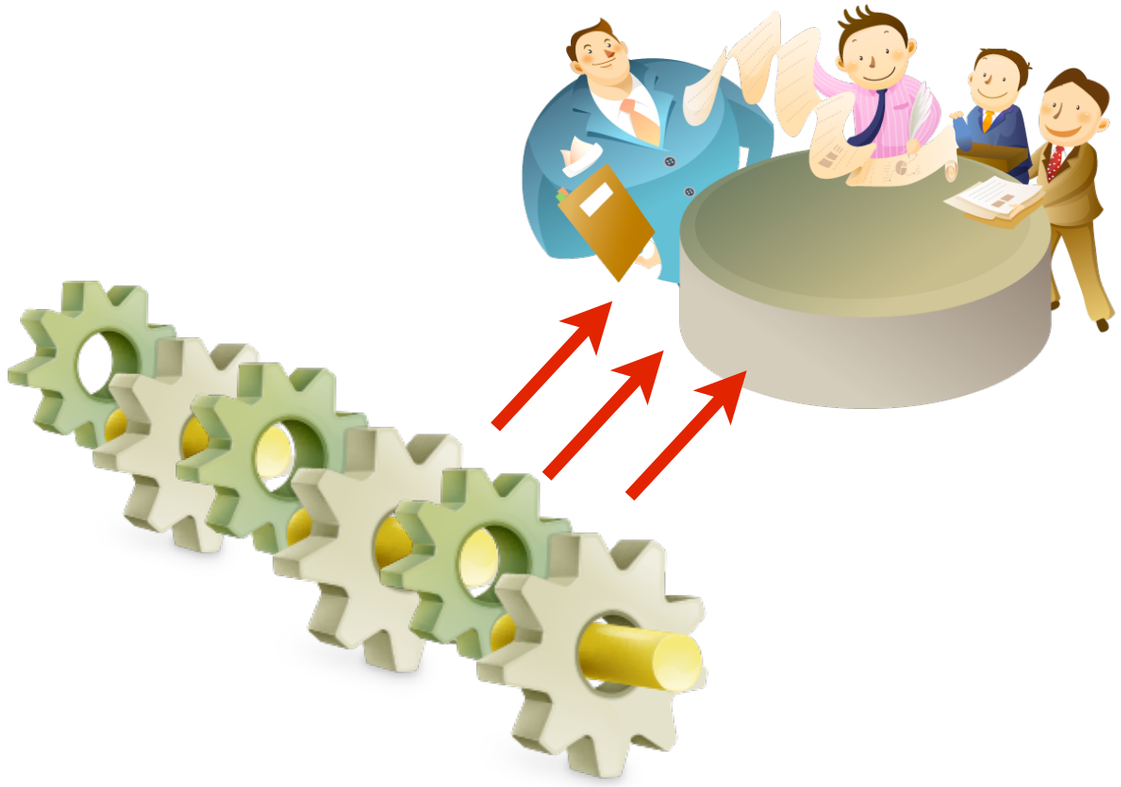
วัตถุประสงค์ของการสร้างระบบสนับสนุนการตัดสินใจ/ระบบสารสนเทศ

ในส่วนของระบบสนับสนุนการตัดสินใจจะเป็นระบบที่ไม่ได้ใช้ในการดำเนินธุรกิจในแต่ละวัน แต่จะเป็นระบบที่ไว้ใช้สำหรับเฝ้าดูหรือตรวจสอบการดำเนินธุรกิจ และเป็นระบบที่ใช้สำหรับสร้างหรือจัดเตรียมข้อมูลเพื่อสนับสนุนการตัดสินใจเชิงกลยุทธ์ที่จะช่วยให้ธุรกิจต่างๆ มีการพัฒนาการดำเนินธุรกิจที่มีประสิทธิภาพดียิ่งขึ้น เพื่อให้เข้าใจเกี่ยวกับฟังก์ชันการทำงานของระบบสนับสนุนการตัดสินใจมากขึ้น ลองพิจารณาตัวอย่างของการดำเนินการต่างๆ ดังแสดงในรูปที่ 1-4 ที่โดยส่วนใหญ่จะเป็นการให้ข้อมูลกับผู้ใช้ในแง่มุมต่างๆ

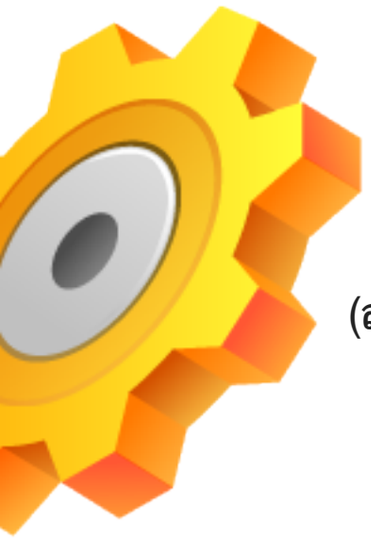
Get the data in

Watching the wheels of business turn

- Show me the top-selling products
- Show me the problem regions
- Tell me why (drill down)
- Let me see other data (drill across)
- Show the highest margins
- Alert me when a district sells below target



รูปที่ 1-4 ตัวอย่างฟังก์ชันการทำงานของระบบสนับสนุนการตัดสินใจ



ระบบสนับสนุนการตัดสินใจโดยส่วนใหญ่จะถูกสร้างขึ้นเพื่อทำการสร้างหรือค้นหาข้อมูลเชิงกลยุทธ์ “**ออกจาก**” ฐานข้อมูลที่สร้างขึ้น ซึ่งระบบนี้จะแตกต่างจากระบบการดำเนินงานที่ถูกออกแบบเพื่อทำการเพิ่มข้อมูล **“เข้าสู่”** ฐานข้อมูล โดยข้อมูลที่ได้จากระบบสนับสนุนการตัดสินใจและระบบการดำเนินงานจะมีความแตกต่างกันด้วย (ลองพิจารณารูป 1-4 อีกครั้งจะทำให้เข้าใจเกี่ยวกับข้อมูลที่จะได้รับจากระบบสนับสนุนการตัดสินใจมากขึ้น)

ความแตกต่างระหว่างขอบเขตและวัตถุประสงค์ของระบบการดำเนินงานและระบบสนับสนุนการตัดสินใจ

ในการจัดเตรียมข้อมูลเชิงกลยุทธ์ เราอาจจำเป็นที่จะต้องสร้างระบบสารสนเทศที่มีความแตกต่างจากระบบการดำเนินงานดั้งเดิมทั้งในแง่ลักษณะและฟังก์ชันการทำงานต่างๆ ซึ่งสามารถแจกแจงได้ดังนี้

- ☑ ระบบสารสนเทศควรที่จะสามารถตอบสนอง/บริการในหลายๆ วัตถุประสงค์
- ☑ ระบบสารสนเทศควรที่จะมีขอบเขตที่แตกต่างจากระบบการดำเนินงาน
- ☑ ระบบสารสนเทศควรที่จะมีเนื้อหา/ข้อมูลในระบบที่แตกต่างจากระบบการดำเนินงาน
- ☑ ระบบสารสนเทศควรที่จะมีรูปแบบการใช้งานและการเข้าถึงข้อมูลที่แตกต่างจากเดิม

จากลักษณะและความแตกต่างอย่างคร่าวๆ ระหว่างระบบการดำเนินงานและระบบสารสนเทศทั้ง 4 ข้อ เราควรที่จะทราบถึงความแตกต่างที่แท้จริงของทั้งสองระบบที่แสดงอยู่ในรูปที่ 1-5 ที่จะแสดงความแตกต่างในแง่มุมต่างๆ ทั้งเนื้อหาของข้อมูล โครงสร้างข้อมูล การเข้าถึงข้อมูล ความถี่ในการเข้าถึงข้อมูล การใช้งานระบบ เวลาที่ใช้ในการเข้าถึงข้อมูล และลักษณะผู้ใช้งาน เป็นต้น

How are they different?

	OPERATIONAL	INFORMATIONAL
Data Content	Current values	Archived, derived, summarized
Data Structure	Optimized for transactions	Optimized for complex queries
Access Frequency	High	Medium to low
Access Type	Read, update, delete	Read
Usage	Predictable, repetitive	Ad hoc, random, heuristic
Response Time	Sub-seconds	Several seconds to minutes
Users	Large number	Relatively small number

รูปที่ 1-5 การเปรียบเทียบกันระหว่างระบบการดำเนินงานและระบบสารสนเทศ

SECTION 5

การสร้างคลังข้อมูลเพื่อจัดเตรียมข้อมูลเชิงกลยุทธ์

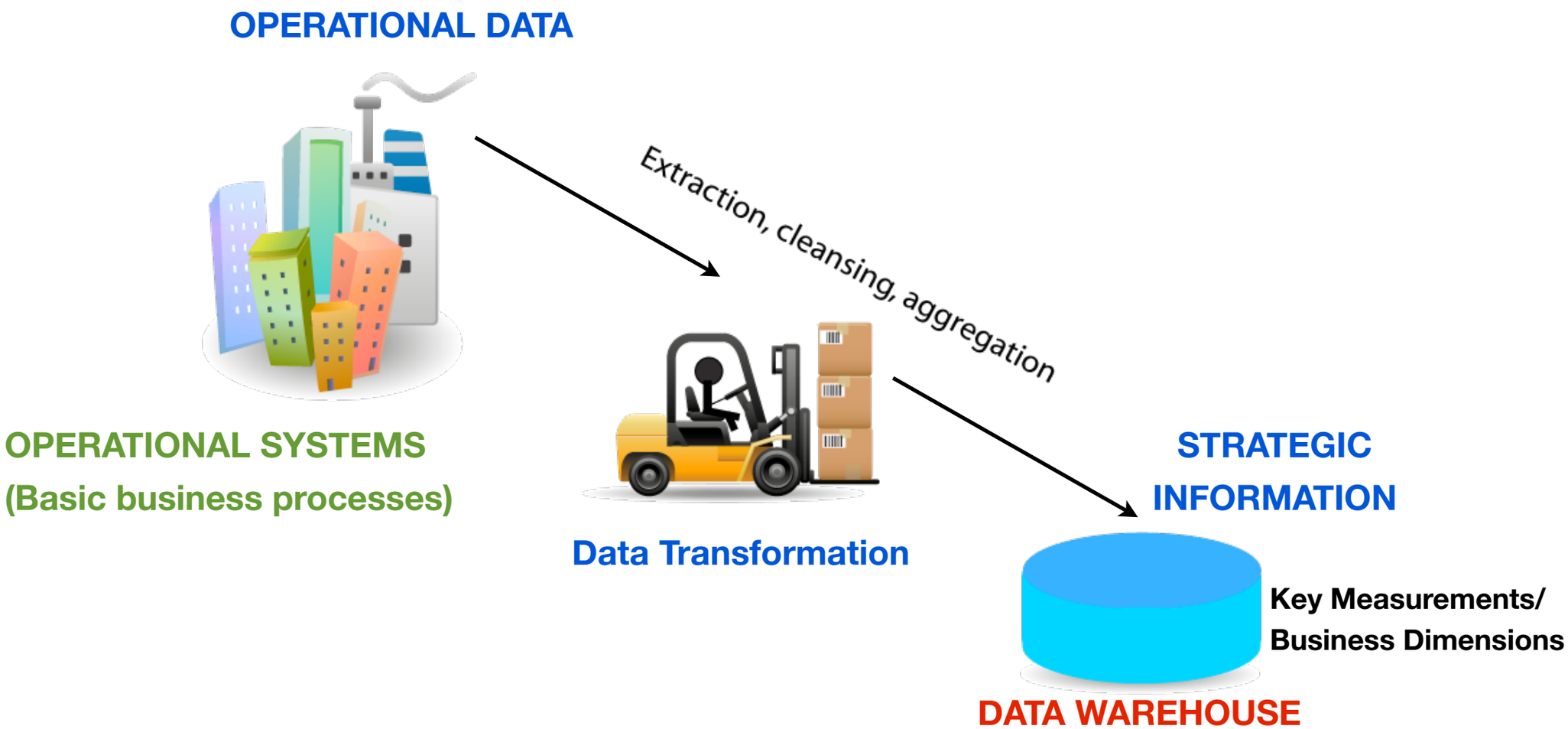
ในการจัดเตรียมข้อมูลเชิงกลยุทธ์ เราไม่ได้ต้องการระบบสารเทศหรือระบบสนับสนุนการตัดสินใจที่แตกต่างกันหลายระบบ แต่เราจะต้องการระบบใหม่เพียงแค่ระบบเดียวที่สามารถจัดจัดเตรียมข้อมูลเชิงกลยุทธ์สำหรับวิเคราะห์ข้อมูลให้แนวโน้มหรือวิสัยทัศน์และสามารถเฝ้าดูประสิทธิภาพของฟังก์ชันการทำงานต่างๆ ในการดำเนินธุรกิจ โดยระบบสารสนเทศที่จะทำการสร้างขึ้นควรที่จะต้องมีคุณลักษณะและการประมวลผลที่แตกต่างจากระบบการดำเนินงาน โดยระบบสารสนเทศสำหรับจัดเตรียมข้อมูลเชิงกลยุทธ์ควรจะประกอบด้วยคุณลักษณะดังต่อไปนี้

- ☑ มีฐานข้อมูลที่ถูกออกแบบมาสำหรับการวิเคราะห์ข้อมูล
- ☑ มีข้อมูลที่มาจากแอพลิเคชันที่หลากหลาย
- ☑ มีฟังก์ชันที่ใช้งานง่ายและเอื้อต่อการใช้งานที่ค่อนข้างยาวนานในแต่ละครั้ง
- ☑ มีการตระหนักถึงลักษณะของข้อมูลที่ถูกใช้งาน
- ☑ ทำให้ผู้ใช้สามารถใช้งานกับระบบได้โดยตรงโดยไม่ต้องร้องขอความช่วยเหลือจากฝ่ายไอที
- ☑ มีการอัพเดทข้อมูล/เนื้อหา เป็นระยะๆ ที่เป็นไปอย่างมีเสถียรภาพ
- ☑ มีการจัดเก็บข้อมูลทั้งในปัจจุบันและข้อมูลย้อนหลังก่อนหน้า
- ☑ สามารถให้ผู้ใช้สามารถเรียกใช้คิวรี (queries) และได้รับผลลัพธ์ทางออนไลน์
- ☑ สามารถให้ผู้ใช้ทำการกำหนดค่าเริ่มต้นให้กับรายงานที่ต้องการได้

จากคุณลักษณะต่างๆ ข้างต้นจะเป็นคุณสมบัติพื้นฐานที่ระบบสนับสนุนการตัดสินใจควรจะมี ดังนั้นการประมวลผลและใช้งานระบบสนับสนุนการตัดสินใจเพื่อให้ได้มาซึ่งข้อมูลเชิงกลยุทธ์จะมีความต้องการต่างๆ ในการทำงาน ซึ่งสามารถแบ่งระดับได้เป็น 4 ระดับด้วยกัน คือ

1. การใช้คิวรีและการสร้างรายงานอย่างง่ายจากข้อมูลปัจจุบันและข้อมูลย้อนหลัง
2. ระบบควรที่จะมีความสามารถในการวิเคราะห์แบบ **“what if”** ในหลายๆ วิธี
3. มีความสามารถใช้คิวรีย้อนกลับ (step back) วิเคราะห์และดำเนินการต่อตามระยะเวลาที่ต้องการ
4. มีความสามารถในการมองเห็นแนวโน้มของข้อมูลย้อนหลังและประยุกต์ใช้แนวโน้มเหล่านั้นในอนาคต

ดังนั้น ในการสร้างระบบสารสนเทศเพื่อจัดเตรียมข้อมูลเชิงกลยุทธ์สำหรับสนับสนุนการตัดสินใจ โดยการสร้างตามความต้องการระดับต่างๆ ข้างต้น เราจะสามารถทำการสร้าง **“คลังข้อมูล *(Data warehouse)*”** ซึ่งเป็นระบบที่ถูกประยุกต์ในวงกว้างและแพร่หลายในปัจจุบัน โดยคลังข้อมูลจะเป็นระบบที่แยกตัวมาจากระบบการดำเนินงานที่ซึ่งจะเน้นที่กระบวนการจัดเตรียมข้อมูลเชิงกลยุทธ์ที่ได้ข้อมูลมาจากการดำเนินธุรกิจในแต่ละวัน (ดังแสดงในรูปที่ 1-6)



รูปที่ 1-6 ภาพรวมกว้างๆ ของคลังข้อมูล

เมื่อเรารู้แหล่งข้อมูลหรือรู้ระบบการดำเนินงานที่มีข้อมูลที่เราต้องการแล้ว เราจะต้องทำการเลือกข้อมูลบางส่วนที่เราต้องการจากระบบการดำเนินงานมาเก็บไว้ในที่พักข้อมูลที่เรียกว่า **“staging area”** ที่จะทำหน้าที่พักข้อมูล และทำการประมวลผลข้อมูล ซึ่งอาจจะเป็นการทำความสะอาดข้อมูล การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลให้อยู่ในรูปแบบที่เหมาะสมต่อการใช้งานได้ง่าย และอื่นๆ จากนั้นค่อยทำการย้าย/ถ่ายโอนข้อมูลที่ถูกประมวลผลแล้วเข้าสู่คลังข้อมูลต่อไป โดยข้อมูลที่จะทำการจัดเก็บลงในคลังข้อมูลนั้นจะประกอบไปด้วย *(1)* ข้อมูลทั่วๆไปจากระบบการดำเนินงาน เช่น ข้อมูลรายการสินค้า ข้อมูลลูกค้า เป็นต้น และ *(2)* ข้อมูลที่เป็นตัวชี้วัดหรือมาตรวัดประสิทธิภาพและประสิทธิผลที่เกี่ยวเนื่องกับการดำเนินธุรกิจ

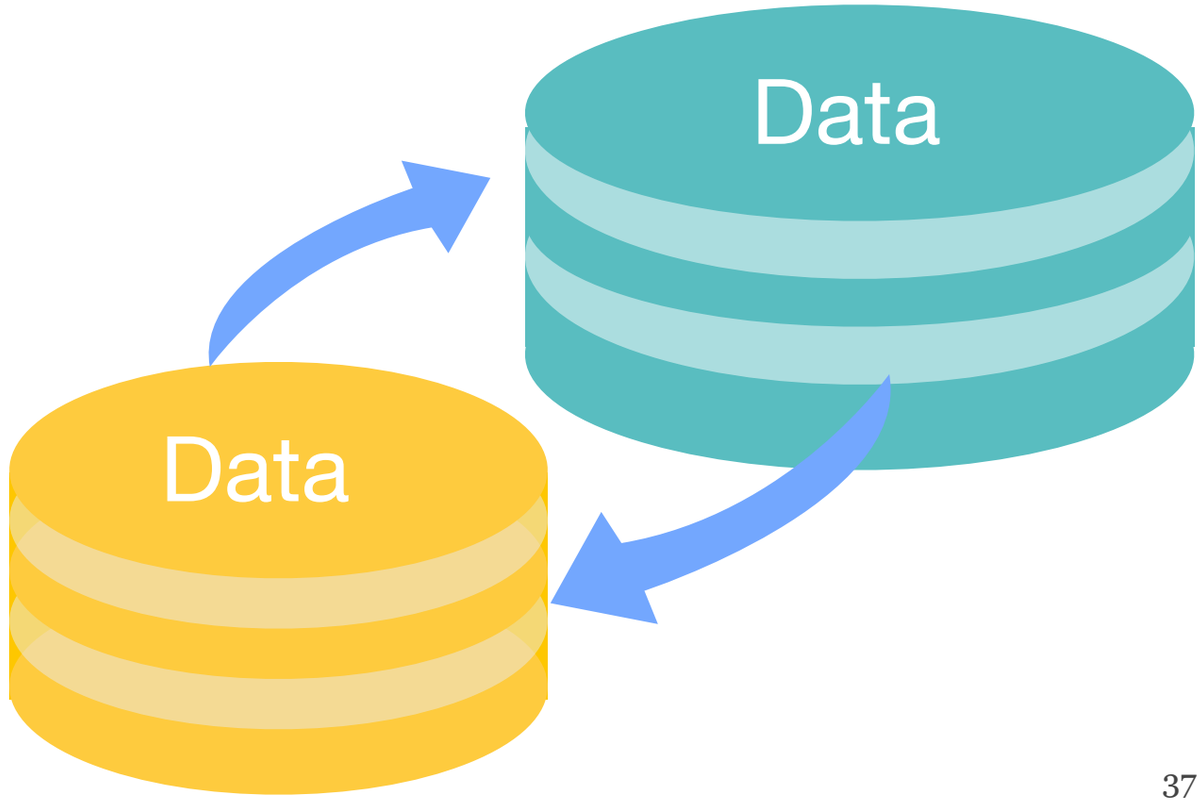
โดยข้อมูลตัวชี้วัดเหล่านี้จะเกี่ยวเนื่องกับข้อมูลทางธุรกิจต่างๆ ตัวอย่างเช่น คลังข้อมูลจะมีการจัดเก็บข้อมูลยอดการขายต่อรายการสินค้าหนึ่งๆ ต่อกลุ่มลูกค้าหนึ่งๆ ต่อพื้นที่ของการขายหนึ่งๆ ต่อโปรโมชั่นหนึ่งๆ เป็นต้น ซึ่งยอดการขายสินค้าจะหมายถึงมาตรวัดความสำเร็จของการดำเนินธุรกิจในแง่ของการขายสินค้าที่มีความเกี่ยวข้องกับมิติทางธุรกิจต่างๆ อาทิ เช่น รายการสินค้า กลุ่มลูกค้า พื้นที่ที่ตั้งร้าน และโปรโมชั่น เมื่อเราทราบถึงข้อมูลยอดขายสินค้าจะทำให้เราสามารถนำข้อมูลเหล่านั้นไปประกอบการตัดสินใจเพื่อทำการปรับปรุงเปลี่ยนแปลงกระบวนการดำเนินธุรกิจ หรือการพัฒนารายการสินค้าและบริการได้

SECTION 6

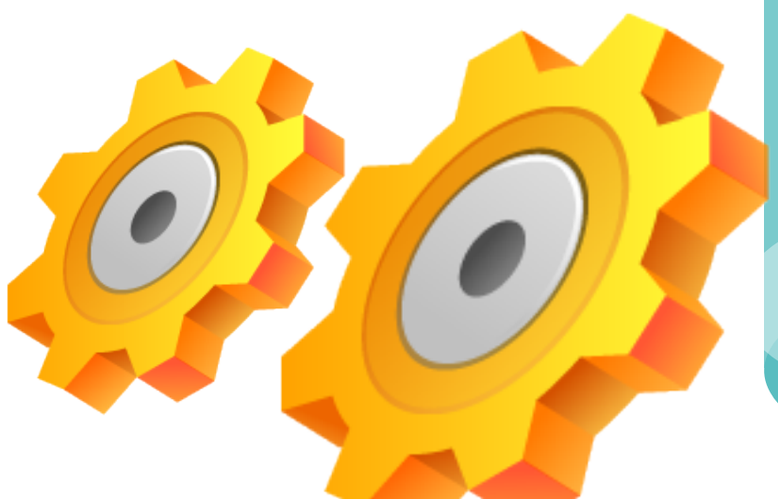
การให้นิยามเกี่ยวกับ"คลังข้อมูล"

จากที่กล่าวมาข้างต้น เราสามารถกล่าวถึง**“คลังข้อมูล”** ได้ว่าเป็นระบบสารสนเทศหรือระบบสนับสนุนการตัดสินใจที่มีการคัดเลือกข้อมูลบางส่วนจากระบบการดำเนินงาน จากนั้นทำการประมวลผลข้อมูลที่ถูกคัดเลือกไว้เพื่อนำไปจัดเก็บไว้ในคลังข้อมูลเพื่อที่จะให้บริการในการตอบคำถามเกี่ยวกับการดำเนินธุรกิจของบริษัท/องค์กรผ่านคิวรีที่สร้างขึ้นจากผู้ใช้ ให้บริการการเรียกดูแนวโน้มของการดำเนินธุรกิจ จัดเตรียมข้อมูลเพื่อช่วยในการปรับปรุงประสิทธิภาพของการดำเนินธุรกิจ อนุญาตให้ผู้ใช้งานสามารถเข้าถึงข้อมูลได้โดยตรงและหลายมุมมอง/หลายมิติ จัดเตรียมตัวชี้วัดประสิทธิภาพของการดำเนินงานที่สำคัญต่อการดำเนินธุรกิจ และมีการจัดเก็บข้อมูลได้อย่างถูกต้อง จากขอบเขตการทำงานของคลังข้อมูลข้างต้น เราจะสามารถสรุปได้อย่างคร่าวๆ ว่า คลังข้อมูลจะนำข้อมูลที่มีประโยชน์ทั้งหมดที่มีอยู่แล้วในองค์กรมาทำความสะอาดและเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลเพื่อให้ได้เป็นข้อมูลเชิงกลยุทธ์

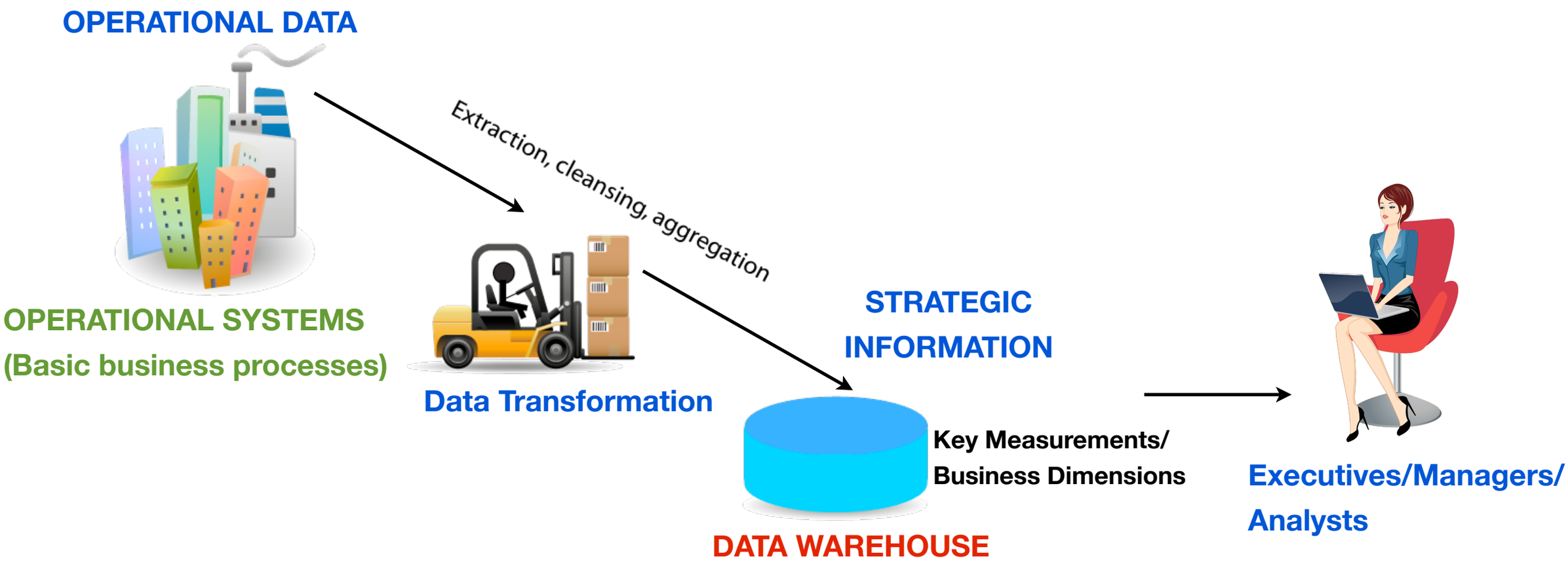
คลังข้อมูลไม่ได้เป็นเพียงซอร์ฟแวร์หนึ่งๆหรือฮาร์ดแวร์หนึ่งๆ ที่เราสามารถซื้อมาเพื่อจัดเตรียมข้อมูลเชิงกลยุทธ์เท่านั้น แต่จะเปรียบเสมือนระบบคอมพิวเตอร์ที่ผู้ใช้สามารถค้นหาข้อมูลเชิงกลยุทธ์ได้โดยตรง โดยที่คลังข้อมูลจะมีลักษณะเด่นอยู่ที่ความสามารถในการวิเคราะห์ข้อมูล ความสามารถเปลี่ยนแปลงได้ง่าย มีความยืดหยุ่น และมีการโต้ตอบกับผู้ใช้งาน โดยมีการตอบสนองและสื่อถึงรูปแบบการถาม-ตอบและถามใหม่ไปเรื่อยๆ และมีความสามารถในการค้นหาคำตอบจากคำถามที่ซับซ้อนหรือคำถามที่ไม่สามารถคาดเดา

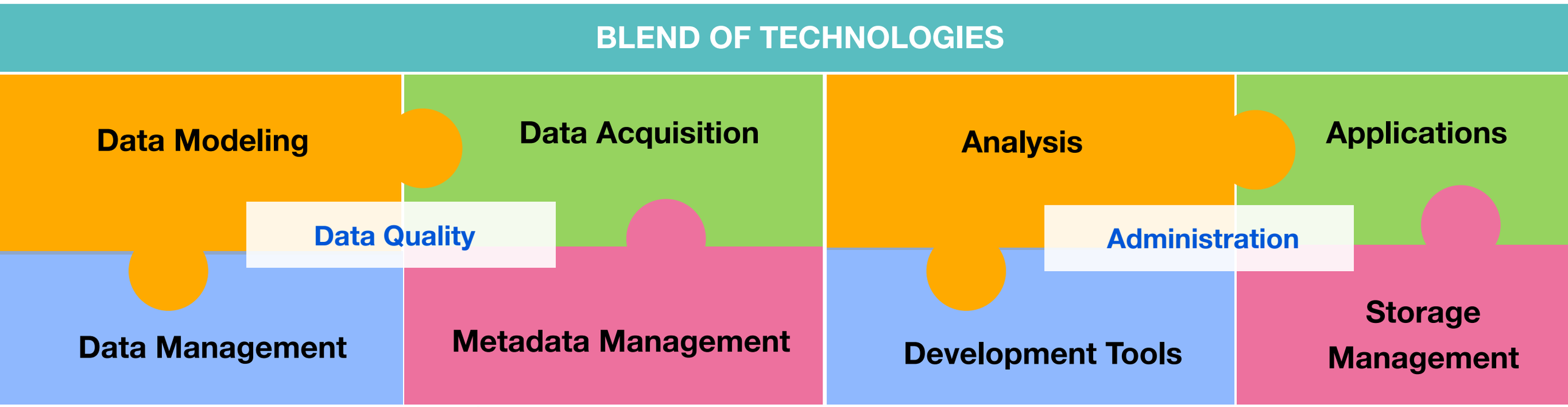


จากฟังก์ชันการให้บริการที่ค่อนข้างจะหลากหลายของคลังข้อมูล เป็นเหตุให้การสร้างคลังข้อมูลจำเป็นที่จะต้องใช้เทคโนโลยีต่างๆ ที่ค่อนข้างหลายเพื่อมาช่วยในการทำงานแต่ละขั้นตอน โดยเทคโนโลยีที่จะถูกนำมาใช้จะถูกแสดงในรูปที่ 1-7 ที่จะประกอบไปด้วยการทำงานต่างๆ



อาทิเช่น การสร้างแบบจำลองข้อมูล (Data modeling) การได้มาซึ่งข้อมูล (Data acquisition) การตรวจสอบคุณภาพของข้อมูล (Data quality) การจัดการข้อมูล (Data management) และการจัดการเมตาดาต้า/การจัดการเกี่ยวกับข้อมูลของข้อมูล (Metadata management) การวิเคราะห์ต่างๆ (Analysis) การจัดการเกี่ยวกับการจัดเก็บข้อมูล (Storage management) เครื่องมือในการพัฒนาฟังก์ชันต่างๆ (Development tools) และแอพลิเคชันต่างๆ เป็นต้น ซึ่งรายละเอียดของแต่ละส่วนจะถูกอธิบายในบทถัดๆ ไป ตามลำดับ





รูปที่ 1-7 การผสมผสานของเทคโนโลยีเพื่อทำการสร้างคลังข้อมูล

SECTION 7

วิวัฒนาการของการสร้างคลังข้อมูล

การสร้างคลังข้อมูลเริ่มได้รับการยอมรับจากบริษัท/องค์กรต่างๆ ในช่วงปลายของทศวรรษ 1980 ซึ่งนับแต่นั้นเป็นต้นมาแนวความคิด/รายละเอียดของการสร้างคลังข้อมูลได้ถูกเปลี่ยนแปลงและพัฒนาขึ้นอย่างต่อเนื่อง ตัวอย่างเช่น

1983

บริษัท Teradata ได้เริ่มคิดค้นระบบการจัดการฐานข้อมูลสำหรับระบบสนับสนุนการตัดสินใจ

1988

Barry Devlin และ Paul Murphy ได้ตีพิมพ์บทความวิจัยใน IBM Systems Journal ที่มีเนื้อหาเกี่ยวกับสถาปัตยกรรมสำหรับระบบสารสนเทศในแง่มุมของคลังข้อมูลทางธุรกิจ

1990

ผู้พัฒนา Red Brick Systems ได้ทำการเพิ่ม Red Brick Warehouse ซึ่งเป็นระบบจัดการฐานข้อมูลสำหรับการสร้างคลังข้อมูล

1991

Bill Inmon ได้เขียนหนังสือ “Building the Data Warehouse” ซึ่งทำให้เขาถูกเรียกว่าเป็นบิดาผู้ให้กำเนิดการสร้างคลังข้อมูล และในปีเดียวกันบริษัท Prism Solutions ได้คิดค้น Prism Warehouse Manager software สำหรับสร้างคลังข้อมูล

1995

Data warehousing instituer ได้ก่อกำเนิดขึ้น โดยเน้นที่การสร้างคลังข้อมูล การพัฒนากระบวนการทำธุรกิจอย่างชาญฉลาด

1996

Ralph Kimbal ได้เขียนหนังสือชื่อ “The Data Warehousing Toolkit” ซึ่งทำให้เขาเป็นหนึ่งในนักเขียนที่ดีที่สุดในการสร้างคลังข้อมูลและการสร้างระบบสนับสนุนการตัดสินใจ

1997

Oracle 8 ที่มีการทำคิวรีกับ Star schema ได้ออกวางจำหน่าย

ความท้าทายและอุปสรรคของการสร้างคลังข้อมูล

หลังจากที่บริษัทต่างๆเริ่มที่จะคำนึงถึงหรือตัดสินใจที่จะประยุกต์ใช้คลังข้อมูลเพื่อที่จะได้รับข้อมูล/ข่าวสารที่สามารถประยุกต์ใช้ในการตัดสินใจแล้ว เมื่อบริษัทเหล่านั้นเริ่มโครงการสำหรับสร้างระบบคลังข้อมูลอาจจะต้องพบเจออุปสรรคหรือความท้าทายต่างๆ มากมายดังต่อไปนี้

- ลูกค้ามีความรู้ความสามารถและประสบการณ์มากขึ้น ซึ่งจะพยายามกดดันให้ผู้สร้างต้องทำการสร้างคลังข้อมูลที่มีการบริการดีขึ้น ปรับปรุงคุณภาพ และทำการปรับแต่งคลังข้อมูลใหม่

- กฎระเบียบของภาครัฐเกี่ยวกับการเปิดเสรีทางอุตสาหกรรมทำให้บริษัทต่างๆ มีการแข่งขันที่รุนแรงมากขึ้น ซึ่งจะส่งผลถึงความต้องการข้อมูลจากคลังข้อมูลมากขึ้นตามไปด้วย

- กฎระเบียบใหม่ๆ ของภาครัฐเกี่ยวกับความเป็นส่วนตัวของบุคคลหรือข้อมูลจะทำให้เราต้องทำการปรับเปลี่ยนวิธีในการเก็บรวบรวมข้อมูลและวิธีในการใช้งานด้วย

- บางคลังข้อมูลที่มีสถาปัตยกรรมที่ไม่เหมาะสมจะผลิตข้อมูลหรือมุมมองเกี่ยวกับข้อมูลที่กระจัดกระจายไม่เป็นชิ้นเป็นอัน
- คิวรี รายงาน และเครื่องมือสำหรับวิเคราะห์ที่ได้สร้างขึ้นในช่วงแรกของการสร้างคลังข้อมูลนั้นมีความซับซ้อนเกินไปและระบบคลังข้อมูลมีการใช้งานที่มากเกินกว่าที่คาดการไว้
- โครงสร้างของคลังข้อมูลที่วางไว้ตอนแรกนั้นไม่เพียงพอกับความต้องการของผู้ใช้ในปัจจุบัน

จากปัญหาที่ได้กล่าวข้างต้น เมื่อเราเริ่มการสร้างหรือกำลังสร้างคลังข้อมูล เราจะต้องมองย้อนไปพิจารณาถึงปัญหาเหล่านี้ เพื่อที่จะทำให้คลังข้อมูลนั้นมีความสามารถสูงที่สุดและมีความน่าเชื่อถือมากที่สุด ซึ่งขั้นตอนการสร้างคลังข้อมูลจะอธิบายในบทถัดๆ ไป

SECTION 8

วิวัฒนาการของการทำธุรกิจอย่างชาญฉลาด

จากอุปสรรคของการประยุกต์ใช้คลังข้อมูลในช่วงต้น ทำให้บริษัทย้อนกลับมามองที่การสนับสนุนการตัดสินใจ โดยหลายๆ บริษัทเริ่มที่จะเข้าใจว่าเป้าหมายที่แท้จริงของระบบสนับสนุนการตัดสินใจนั้นมีความซับซ้อนคล้ายกับจานที่วางซ้อนกัน 2 ชั้น ซึ่งสามารถกล่าวได้คือ

ในการทำธุรกิจอย่างชาญฉลาด (Business intelligence, BI) ควรที่จะต้องทำการพิจารณาและดำเนินการตามกระบวนการทั้งสองข้างต้น ซึ่งโดยแท้จริงแล้ว BI จะใช้แอพลิเคชันและเทคโนโลยีต่างๆเข้ามาเป็นตัวช่วยในการทำธุรกิจ โดยแอพพลิเคชันที่ใช้จะหมายถึงระบบหรือเทคโนโลยีสำหรับการเก็บรวบรวมข้อมูล การทำความสะอาดข้อมูล การรวมข้อมูลให้เป็นหนึ่งเดียว และการจัดเก็บข้อมูล เป็นต้น นอกจากนั้น BI จะเกี่ยวข้องกับเครื่องมือ เทคนิค และแอพพลิเคชันต่างๆ ที่ใช้สำหรับการวิเคราะห์ข้อมูลที่ทำการเก็บไว้ก่อนหน้า ซึ่งจากความหมายที่ค่อนข้างกว้างของ BI จึงได้มีผู้คนมากมายได้ให้คำนิยามเกี่ยวกับ BI ไว้มากมายเช่น Garter Group ได้ให้นิยามว่า BI เปรียบเสมือนร่มที่รวมแนวความคิดและวิธีการที่จะปรับปรุงการตัดสินใจทางธุรกิจ และ Data warehouse institute เปรียบเทียบ BI ว่าเป็นโรงกลั่นข้อมูลที่คล้ายคลึงกับโรงกลั่นน้ำมันที่นำข้อมูลมาเป็นวัตถุดิบ จากนั้นทำการกลั่นข้อมูลโดยการประมวลผลตามขั้นตอนต่างๆเพื่อให้ได้มาซึ่งสารสนเทศ เป็นต้น

จากที่กล่าวข้างต้นเราสามารถมองว่า BI สำหรับองค์กรใดๆ ก็ตามจะประกอบไปด้วย 2 ส่วนด้วยกัน คือ

การแปลงข้อมูลให้เป็นสารสนเทศ

(Data to information)

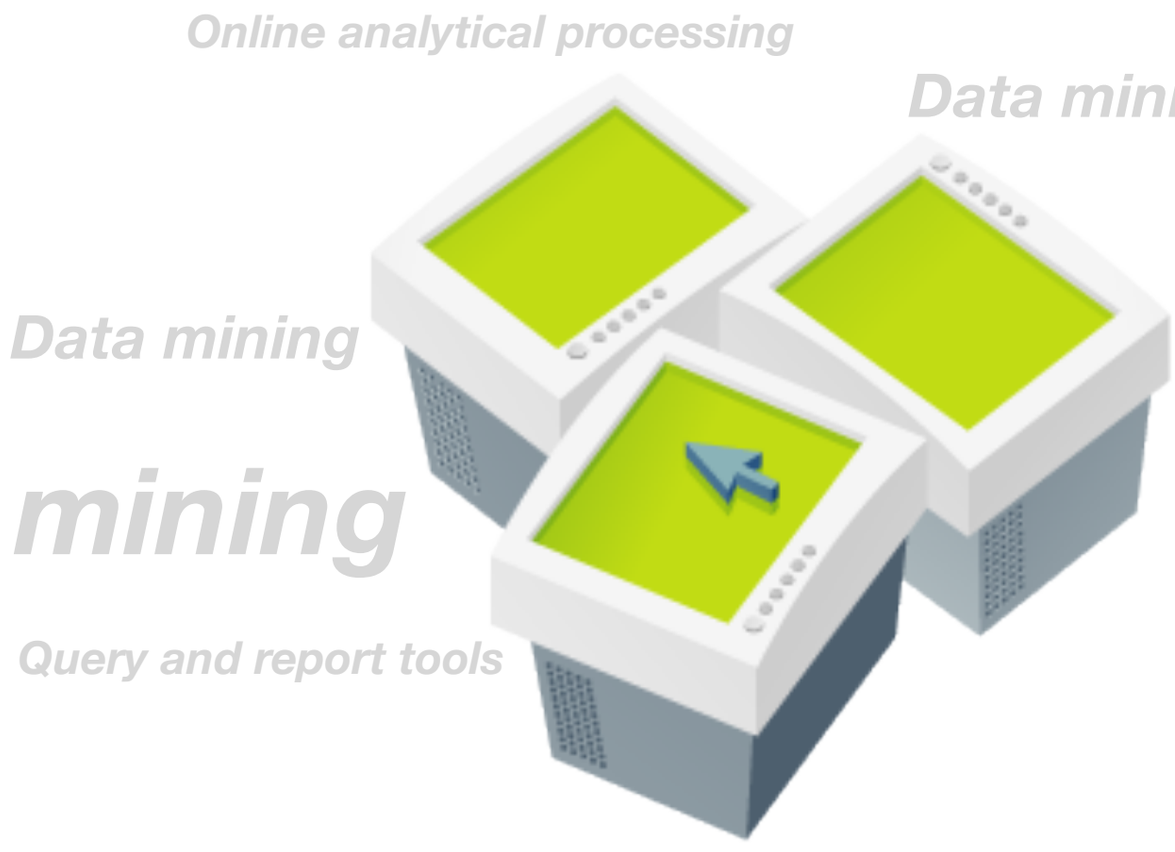
จะเป็นการสกัดข้อมูลจากหลายๆระบบการดำเนินงานแล้วนำข้อมูลเหล่านั้นรวมเข้าด้วยกัน จากนั้นทำการประมวลผลข้อมูลเบื้องต้น เช่น การทำความสะอาด การเปลี่ยนแปลง/เปลี่ยนรูป และทำการจัดเก็บข้อมูลเหล่านั้นไว้ในที่เก็บข้อมูลพิเศษที่แตกต่างจากระบบการดำเนินงานเดิม

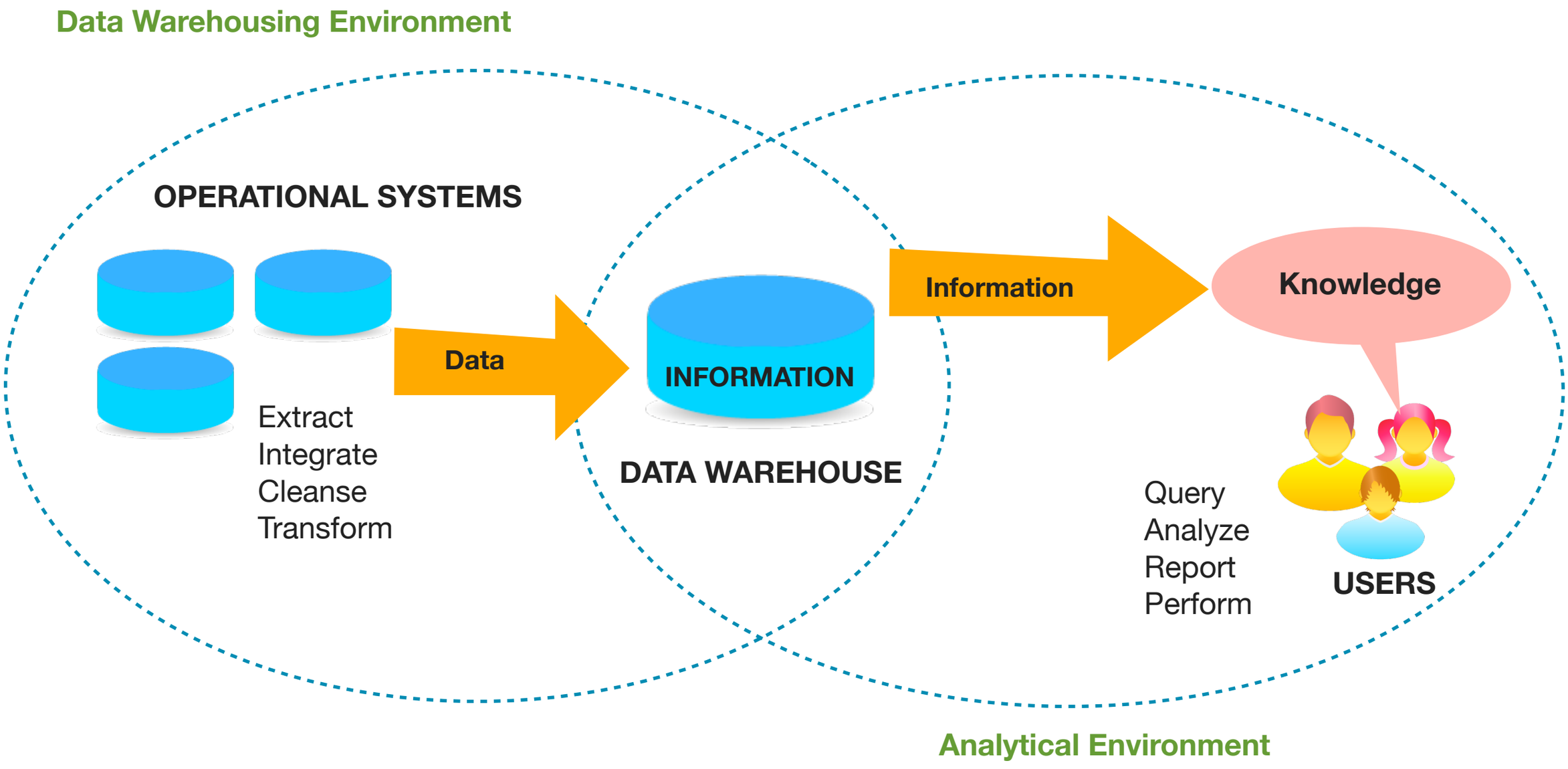
การค้นหาความรู้จากสารสนเทศ

(Information to knowledge)

จะเกี่ยวข้องกับเครื่องมือในการวิเคราะห์ข้อมูลที่อนุญาตให้ผู้ใช้สามารถเข้าถึงและวิเคราะห์เนื้อหาของสารสนเทศแล้วแปลสภาพเนื้อหาเหล่านั้นให้เป็นองค์ความรู้ต่อไป

BI ที่แสดงในรูปที่ 1-8 จะแสดงถึง 2 ส่วนประกอบหลักที่ต้องทำงานด้วยกัน ซึ่งการสร้างคลังข้อมูลจะอยู่ในส่วนแรกที่เป็นกระบวนการเปลี่ยนรูปข้อมูลให้เป็นสารสนเทศ ในส่วนที่สองนั้นจะเป็นทำการวิเคราะห์ข้อมูลเหล่านั้นให้เป็นองค์ความรู้ โดยอาจใช้เครื่องมือต่างๆ เข้ามาช่วย อาทิเช่น การทำเหมืองข้อมูล (Data mining) การประมวลผลข้อมูลแบบออนไลน์ (Online analytical processing) และการประมวลผลคิวรีต่างๆ (Query and report tools)





รูปที่ 1-8 การทำธุรกิจอย่างชาญฉลาด: การสร้างคลังข้อมูลและการวิเคราะห์ข้อมูลในแง่มุมอื่นๆ

SECTION 9

คำถามท้ายบท

1. ข้อมูลเชิงกลยุทธ์คืออะไร จงยกตัวอย่างของข้อมูลเชิงกลยุทธ์ที่เกี่ยวข้องกับธุรกิจธนาคารมา 3 ข้อ

2. ขั้นตอนการทำงานหลักของคลังข้อมูลเป็นอย่างไรบ้าง

3. เพราะเหตุใดระบบการดำเนินงานจึงไม่เหมาะกับการจัดเตรียมข้อมูลเชิงกลยุทธ์

4. จงอธิบายความแตกต่างระหว่างระบบการดำเนินงานและระบบสารสนเทศ

5. จงยกตัวอย่างของโอกาสที่จะได้รับเมื่อธุรกิจหนึ่งๆ มีการใช้ข้อมูลเชิงกลยุทธ์มาประกอบการตัดสินใจ

6. จงยกตัวอย่างของการเสียโอกาสทางธุรกิจ เมื่อธุรกิจหนึ่งๆ ไม่มีการประยุกต์ใช้ข้อมูลเชิงกลยุทธ์มาประกอบการตัดสินใจ

7. คุณสมบัติพื้นฐานที่พึงมีของระบบสนับสนุนการตัดสินใจประกอบไปด้วยอะไรบ้าง

8. ข้อมูลที่ถูกจัดเก็บอยู่ในคลังข้อมูลมีอะไรบ้าง

9. การสร้างคลังข้อมูลประกอบไปด้วยการผสมผสานกันระหว่างเทคโนโลยีใดบ้าง

10. การทำธุรกิจอย่างชาญฉลาดมีลักษณะเป็นอย่างไร ประกอบไปด้วยกี่ขั้นตอน อะไรบ้าง

CHAPTER 2

นิยามและส่วนประกอบของคลังข้อมูล

SECTION 1

แผนการสอนประจำบท

วัตถุประสงค์ของบทเรียน

- ทบทวนคำจำกัดความของคลังข้อมูล
- อธิบายเกี่ยวกับคุณลักษณะต่างๆ ของคลังข้อมูล
- การแยกความแตกต่างระหว่างคลังข้อมูลและดาต้ามาร์ท (Data marts)
- ศึกษาเกี่ยวกับส่วนประกอบของคลังข้อมูล
- ศึกษาเกี่ยวกับวิธีการสร้างคลังข้อมูล

เนื้อหาของบทเรียน

เนื้อหาในบทนี้จะประกอบด้วยคุณลักษณะของคลังข้อมูล ส่วนประกอบของคลังข้อมูล นิยามของดาต้าร์มาร์ท ความแตกต่างของคลังข้อมูลและดาต้ามาร์ท วิธีการสร้างคลังข้อมูล และแนวปฏิบัติในการสร้างคลังข้อมูล

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารประกอบการสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

กิจกรรมการเรียน-การสอน

- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารประกอบการสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

การวัดและประเมินผล

- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

SECTION 2

บทนำ

การทำธุรกิจอย่างชาญฉลาด **(Business intelligence, BI)** จะประกอบไปด้วยการทำงานหลัก 2 ขั้นตอน คือ การเปลี่ยนข้อมูลให้เป็นข้อมูลสารสนเทศ และการเปลี่ยนข้อมูลสารสนเทศไปเป็นองค์ความรู้ โดยการสร้างคลังข้อมูลนั้นจะเป็นส่วนหนึ่งของ BI ที่จะทำการเปลี่ยนข้อมูลดิบให้เป็นข้อมูลสารสนเทศที่มีประโยชน์ อาทิ การสร้างข้อมูลที่เป็นผลสรุป การสร้างข้อมูลที่ถูกอธิบายหรือแสดงในเชิงเปรียบเทียบ และ อื่นๆ โดยข้อมูลสารสนเทศที่ได้จะสามารถนำไปเป็นเครื่องมือในการประกอบการตัดสินใจในการดำเนินธุรกิจต่างๆ ได้

การทำงานหลักของคลังข้อมูลจะประกอบไปด้วย 4 ฟังก์ชันการทำงานหลักด้วยกัน คือ (1) การเลือกข้อมูลดิบมาบางส่วน โดยทำการเลือกเฉพาะข้อมูลที่สำคัญหรือเป็นข้อมูลที่ผู้ใช้สนใจ (2) การประมวลผลกับข้อมูลดิบเหล่านั้น อาทิ การทำให้ข้อมูลต่างๆ เป็นมาตราฐาน การทำให้ข้อมูลมีความถูกต้อง สมบูรณ์ ครบถ้วน เป็นต้น (3) การจัดเก็บข้อมูลที่สำคัญเหล่านั้นไว้ในคลังข้อมูล และ (4) การเรียกใช้งานข้อมูลที่สำคัญที่ถูกจัดเก็บอยู่ในคลังข้อมูล

SECTION 3

คุณลักษณะของข้อมูล ในคลังข้อมูล

ก่อนที่จะทำการสร้างคลังข้อมูล เราควรจะต้องทราบถึงคุณลักษณะของคลังข้อมูลว่ามีลักษณะเด่น หรือลักษณะพิเศษของคลังข้อมูลว่าเป็นอย่างไร มีแหล่งข้อมูลเป็นอย่างไรบ้าง และปัจจัยอื่นๆ อีกมากมาย นอกจากนั้นเราจำเป็นจะที่ต้องพิจารณาถึงส่วนประกอบต่างๆ ที่แตกต่างกันที่จะทำให้การทำงานสามารถตอบสนองความต้องการของผู้ใช้ได้ดีที่สุด รวมถึงการศึกษาเกี่ยวกับคุณลักษณะพื้นฐานของคลังข้อมูลด้วย ซึ่งจากบทที่ผ่านมาจะทำให้เราทราบถึงความต้องการของการสร้างคลังข้อมูล แต่เรายังไม่ทราบว่าคลังข้อมูลคืออะไร? มีลักษณะเป็นอย่างไร? เพื่อให้เข้าใจนิยามพื้นฐานของคลังข้อมูล ลองพิจารณานิยามที่ได้จาก **“Bill Inmon”** ที่ซึ่งเป็นผู้ที่ได้รับการขนานนามว่าเป็น **“บิดาของคลังข้อมูล”** ได้ทำการนิยามคลังข้อมูลไว้ว่า

อ้างอิงภาพ http://formacioncontinua.medellin.upb.edu.co/2013/BI/Bill-Inmon.php

นิยามคลังข้อมูล

"***Data warehouse is a collection of data in support of management's decision that have 4 characteristics : (1) subject-oriented (2) integrated (3) nonvolatile and (4) time variant***"

ในส่วนของ **"Sean Kelly"** ซึ่งเป็น **"ผู้ทรงอิทธิพลต่อการสร้างคลังข้อมูล"**
ก็ได้ให้คำจำกัดความเกี่ยวกับคลังข้อมูลไว้เช่นกัน ดังนี้

"***Data in data warehouse is (1) separate (2) available (3) integrated (4) time stamped (5) subject oriented (6) nonvolatile and (7) accessible***"

จากนิยามที่ใกล้เคียงกันของบุคคลที่มีชื่อเสียงเกี่ยวกับการสร้างคลังข้อมูลทั้งสอง เราจะสามารถสรุปและทำการอธิบายถึงนิยามของคลังข้อมูลได้ แต่ก่อนที่จะทำการศึกษาถึงนิยามของคลังข้อมูล ลองพิจารณาคำถามเบื้องต้นต่อไปนี้เพื่อที่จะทราบถึงคุณลักษณะหลักของคลังข้อมูล ก่อนที่จะทำความเข้าใจหรือทำการสร้างคลังข้อมูลคุณเกิดข้อสงสัยเหล่านี้หรือไม่

จากคำถามเหล่านี้ เราลองพิจารณาถึงคุณลักษณะของข้อมูลที่สำคัญในคลังข้อมูลดังนี้

1

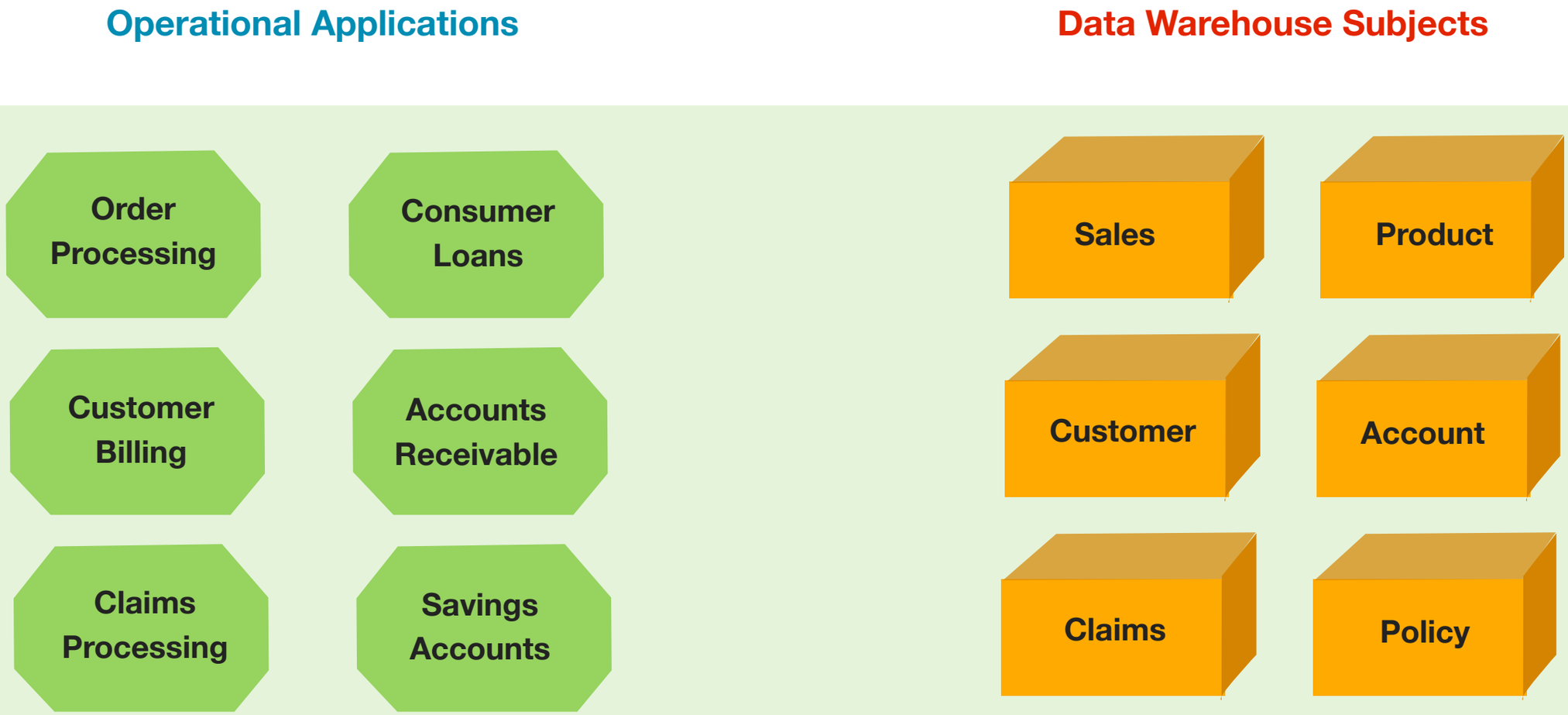
ข้อมูลที่ถูกจัดเก็บตามหัวข้อที่สนใจ (Subject-oriented data)

ก่อนที่เราจะพิจารณาข้อมูลในคลังข้อมูล ลองพิจารณาข้อมูลในระบบการดำเนินงานซึ่งเป็นข้อมูลที่เราคุ้นเคยเป็นอย่างดี โดยข้อมูลที่ถูกจัดเก็บอยู่ในระบบการดำเนินงานจะถูกจัดเก็บโดยแยกตามแอพพลิเคชันหรือฟังก์ชันการใช้งานต่างๆ อาทิ

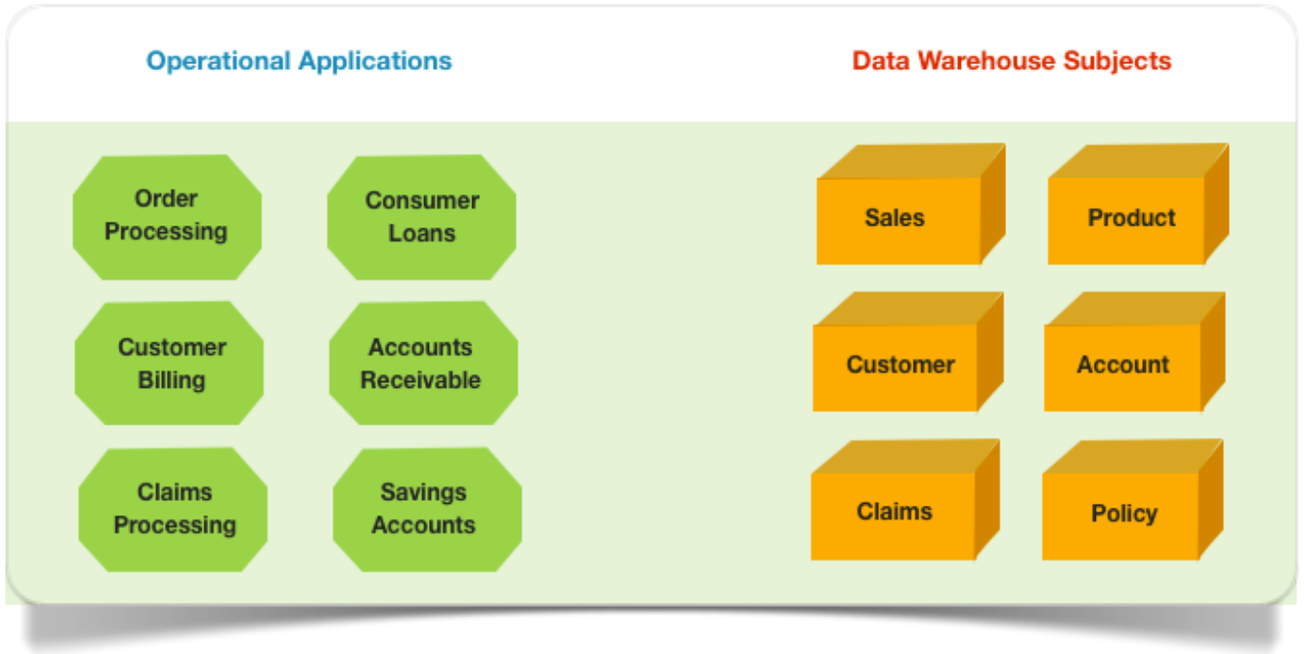
- ข้อมูลการสั่งสินค้าจากลูกค้าที่ประกอบไปด้วยข้อมูลลูกค้า ข้อมูลเกี่ยวกับสินค้า และจำนวนที่ต้องการสั่งซื้อสินค้า
- ข้อมูลคลังสินค้าจะประกอบไปด้วย รหัสสินค้า ชื่อสินค้า หมวดหมู่สินค้า และจำนวนสินค้าคงเหลือในคลังสินค้า
- การตรวจสอบเครดิตของลูกค้าจะเก็บข้อมูล ลูกค้า และธนาคาร เป็นต้น

จากตัวอย่างข้างต้น จะทำให้เราได้เห็นภาพกว้างๆ ของระบบการดำเนินงานของธุรกิจต่างๆ ที่มีการจัดเก็บข้อมูลที่สอดคล้องกับการทำธุรกรรมหนึ่งๆ กับธุรกิจนั้นๆ และยังสอดคล้องกับฟังก์ชันการทำงานต่างๆ แต่ในส่วนของคลังข้อมูลนั้นจะมีการเก็บข้อมูลที่แตกต่างจากระบบดำเนินงาน ข้อมูลในคลังข้อมูลจะถูกเก็บและเชื่อมโยงด้วย “***หัวข้อทางธุรกิจ (Business subject)***” ที่เกี่ยวเนื่องและมีความสำคัญต่อการดำเนินธุรกิจ เช่น “บริษัทผู้ผลิตสินค้า” จะมีความเกี่ยวเนื่องกับข้อมูลการผลิตสินค้า การขายสินค้า การส่งสินค้า การจัดเก็บสินค้าเข้าสู่คลังสินค้า และอื่นๆ ข้อมูลเหล่านี้จะเป็นข้อมูลที่เป็นหัวข้อทางธุรกิจที่สำคัญของบริษัทที่จะส่งผลต่อผลกำไร-ขาดทุนของธุรกิจนั้นๆ ลองพิจารณาอีกตัวอย่างหนึ่ง คือ “บริษัทค้าปลีก” จะมีข้อมูลการขายณ จุดขายสินค้าเป็นหัวข้อทางธุรกิจที่สำคัญ เป็นต้น จากตัวอย่างธุรกิจทั้งสอง เราจะสามารถสรุปเกี่ยวกับ “หัวข้อทางธุรกิจ” ได้ว่าเป็น “ฟังก์ชัน กิจกรรม หรือการดำเนินธุรกิจที่สำคัญและส่งผลถึงผลกำไรหรือขาดทุนของบริษัทได้ รวมถึงเป็น**ข้อมูลที่สามารถวัดหรือประเมินผลสัมฤทธิ์ได้**”

In the data warehouse, data is not stored by operational applications, but by business subjects.



รูปที่ 2-1 การเปรียบเทียบการจัดเก็บข้อมูลระหว่างระบบการดำเนินงานและคลังข้อมูล



รูปที่ 2-1 การเปรียบเทียบการจัดเก็บข้อมลระหว่างระบบการดำเนินงานและคลังข้อมล

ในการที่จะเข้าใจถึงความแตกต่างระหว่างข้อมูลที่ถูกจัดเก็บไว้ในระบบการดำเนินงานและคลังข้อมูล ลองพิจารณารูปที่ 2-1 ที่แสดงการเปรียบเทียบการจัดเก็บข้อมูลจากทั้งสองระบบ โดยจากรูปเราจะสังเกตุได้ว่าข้อมูลที่ถูกเก็บไว้ในระบบการดำเนินงานจะถูกเก็บแยกตามแต่ละแอพพลิเคชัน อาทิ ข้อมูลการสั่งสินค้า ข้อมูลการยืมเงินของลูกค้า ข้อมูลการออกใบเสร็จให้กับลูกค้า ข้อมูลการเรียกเคลมประกัน เป็นต้น ซึ่งจากแอพพลิเคชันต่างๆ เราจะเห็นว่าได้ว่าระบบทั้งสองมีวัตถุประสงค์ต่างกัน โดยระบบการดำเนินงานจะเก็บข้อมูลเพื่อเป็นการลงบันทึกการทำธุรกรรมทางธุรกิจเสียเป็นส่วนใหญ่ แต่การจัดเก็บข้อมูลในคลังข้อมูลจะเป็นการเก็บข้อมูลเพื่อสร้างเป็นข้อมูลเชิงกลยุทธ์สำหรับประกอบการตัดสินใจในการดำเนินการต่างๆ

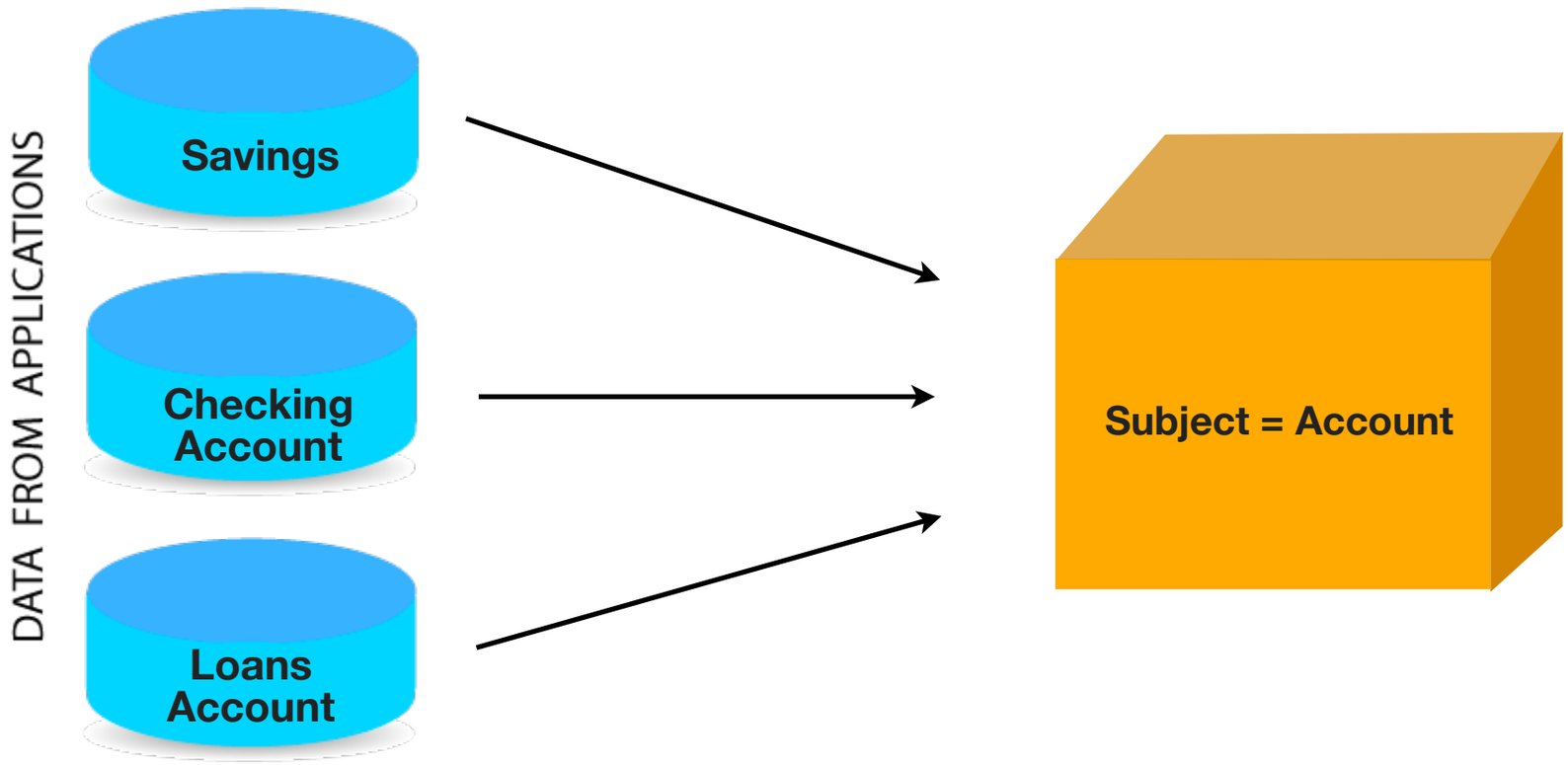
เพื่อให้เห็นความแตกต่าง ลองพิจารณาข้อมูลการเรียกเคลมประกันซึ่งจะเป็นข้อมูลที่ถูกจัดเก็บในบริษัทที่ทำธุรกิจประกัน โดยระบบการดำเนินงานจะทำการจัดเก็บข้อมูลการเคลมประกันของลูกค้าแต่ละราย เช่น ณ วันหนึ่งๆ มีลูกค้าที่ซื้อกรมธรรม์หมายเลยใดได้ทำการติดต่อมาเพื่อขอรับการเคลมประกันบ้าง ซึ่งวัตถุประสงค์หลักจะเป็นการเก็บข้อมูลเพื่อเป็นหลักฐานการทำธุรกิจกันระหว่างลูกค้าและบริษัท แต่สำหรับคลังข้อมูลจะเป็นการเก็บข้อมูลการเคลมประกันของลูกค้า โดยจะมุ่งเน้นที่ข้อมูลที่เป็นข้อเท็จจริงและตัวชี้วัดที่เกี่ยวข้องกับหัวข้อนั้นๆ เช่น สาเหตุของการเคลมประกัน จำนวนที่ต้องจ่ายค่าชดเชย และอื่นๆ ข้อมูลเหล่านี้จะเป็นข้อมูลที่เกี่ยวข้องกับการดำเนินธุรกิจโดยตรง เมื่อผู้บริหารทราบถึงค่าชดเชยที่ต้องจ่ายในแต่ละเดือน จะทำให้ทราบถึงผลกำไรและจะสามารถตัดสินใจที่จะคิดหรือดำเนินกลยุทธ์ต่างๆ ที่เพิ่มขึ้นจากเดิมได้

2

ข้อมูลที่ถูกรวมมาจากหลายแหล่งข้อมูล
(Integrated data)

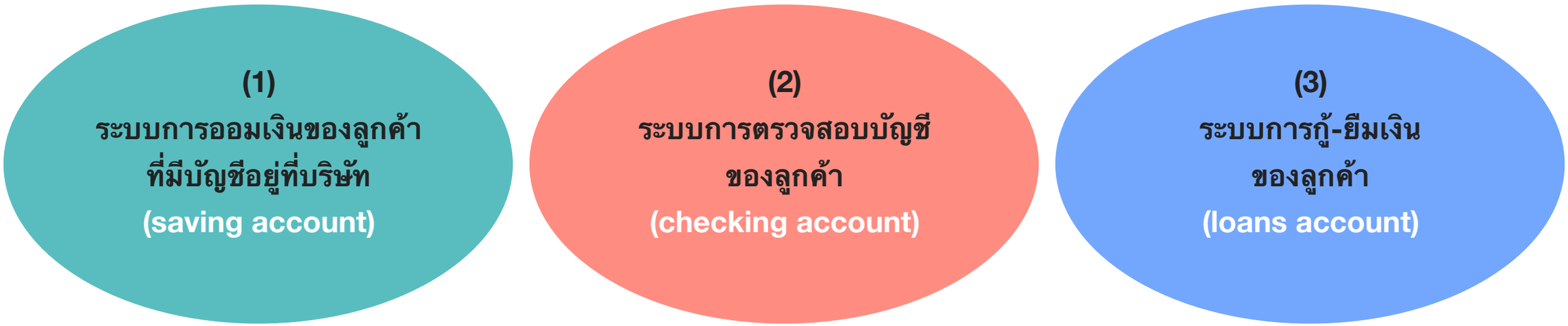
ในการสร้างข้อมูลเชิงกลยุทธ์จากคลังข้อมูลให้มีความครบถ้วนสมบูรณ์ เราอาจจำเป็นที่จะต้องเรียกใช้ข้อมูลจากหลายๆ แอพพลิเคชัน (หลายระบบการดำเนินงานหรือหลายแหล่งข้อมูล) โดยข้อมูลที่มาจากหลายระบบอาจจะมีความแตกต่างกันในเรื่องของระบบการจัดการฐานข้อมูลที่ใช้ (Database management system, DBMS) รูปแบบของแฟ้มข้อมูล (File format) หรือการจัดเก็บข้อมูลส่วนย่อยๆ (Data segment) และแอพพลิเคชันต่างๆ ที่มีความแตกต่างในเรื่องของเค้าโครงของแฟ้มข้อมูล (File layout) การแทนข้อมูลที่เป็นอักขระ (Character code representation) และการตั้งชื่อฟิลด์ต่างๆ ที่สื่อถึงข้อมูลเดียวกัน (Field naming convention) ตัวอย่างเช่น ในการสร้างคลังข้อมูลหนึ่งๆอาจมีการเรียกใช้ข้อมูลจากบริษัท Metro Mail, A.C. Nielsen และ IRI ซึ่งเป็นบริษัทผู้ให้บริการข้อมูลในการดำเนินธุรกิจต่างๆ ดังนั้นเมื่อคลังข้อมูลมีการใช้ข้อมูลทั้งจากระบบการดำเนินงานที่มีจำนวนหลายระบบและจากแหล่งข้อมูลภายนอกจะทำให้เราจะต้องทำการรวบรวมข้อมูลเหล่านั้นแล้วจัดเก็บไว้ในคลังข้อมูลให้อยู่ในรูปแบบที่เหมาะสมต่อการใช้งานต่อไป

Data inconsistencies are removed; data from diverse operational applications is integrated.



รูปที่ 2-2 การรวมกันของข้อมูล

รูปที่ 2-2 แสดงถึงกระบวนการรวบรวมข้อมูลอย่างง่ายของบริษัท/สถาบันทางการเงิน โดยข้อมูลนั้นจะถูกรวบรวมจาก 3 แอพพลิเคชัน แล้วเก็บไว้ในหัวข้อทางธุรกิจเกี่ยวกับบัญชี โดยแอพพลิเคชันที่เป็นแหล่งข้อมูลอินพุตของคลังข้อมูลจะประกอบไปด้วย



ซึ่งจากข้างต้นระบบทั้งสามอาจมีการตั้งชื่อต่างๆ ให้กับฟิลด์หรือแอทริบิวของข้อมูลที่แตกต่างกัน หรืออาจมีรูปแบบ (format) ของข้อมูลที่แตกต่างกัน อาทิ หมายเลขบัญชีสำหรับระบบ saving account ควรจะมีด้วยกัน 8 หลักด้วยกัน แต่สำหรับระบบ checking account จะใช้หมายเลขบัญชีเพียง 6 หลัก ซึ่งจากความแตกต่างข้างต้น เราจะต้องกำจัดความไม่สอดคล้องกันของข้อมูล โดยการสร้างมาตราฐาน ให้กับข้อมูลเหล่านั้นเพื่อทำ ให้ข้อมูลที่เหมือนกันที่ถูกเก็บ โดยแอพลิเคชันที่แตกต่างกันและถูกจัดเก็บลง ในฐานข้อมูลที่แตกต่างกัน ให้สื่อความหมายเดียวกัน ดังนั้นก่อนที่จะทำการเคลื่อนย้ายข้อมูลเข้าสู่คลังข้อมูลเราจะต้องทำการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล ให้เป็นมาตรฐานเสียก่อน

3

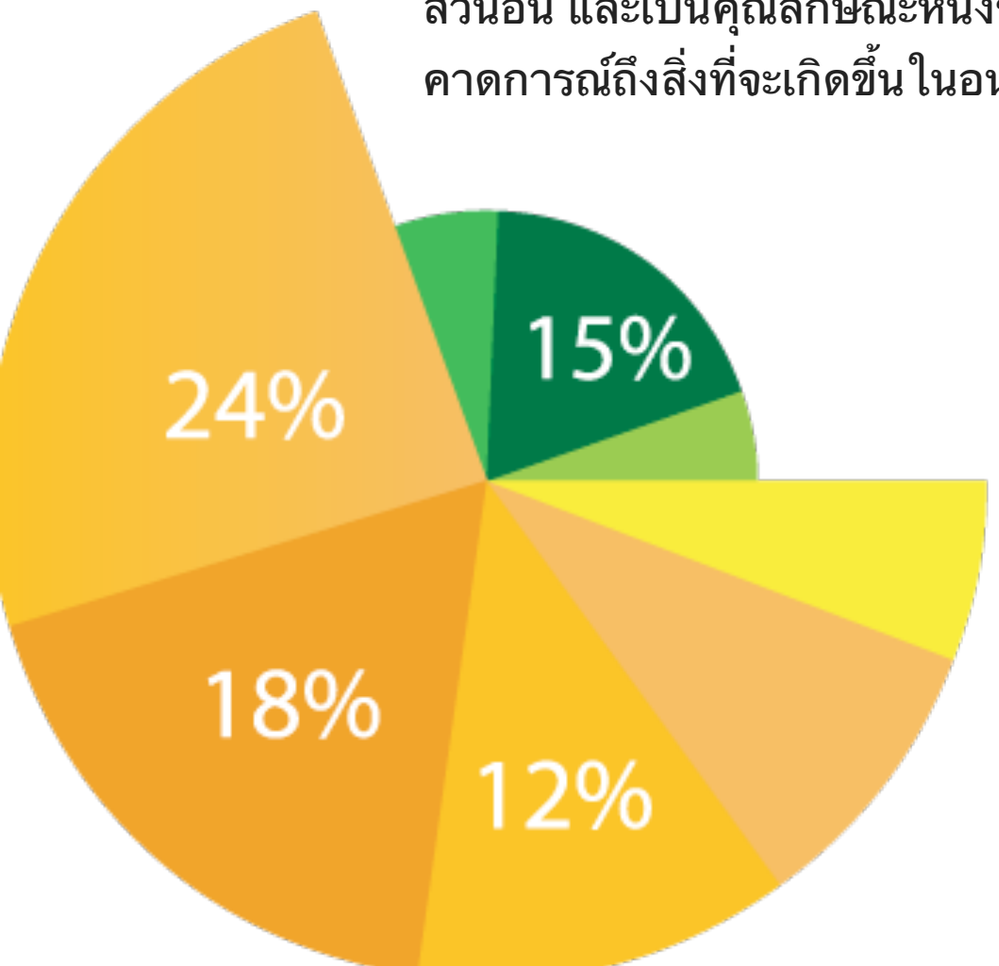
ข้อมูลที่เกี่ยวเนื่องกับช่วงเวลาต่างๆ (Time-variant data)

การจัดเก็บข้อมูลของระบบการดำเนินงานจะทำการเก็บ**ข้อมูลที่เป็นปัจจุบัน**เท่านั้น เช่น ระบบบัญชีของธนาคารที่ทำการเก็บข้อมูลยอดเงินคงเหลือปัจจุบันของบัญชีลูกค้า ระบบการสั่งสินค้าจะทำการเก็บข้อมูลการสั่งสินค้าครั้งล่าสุดของลูกค้ารายหนึ่งๆ เป็นต้น

แต่อย่างไรก็ตามระบบการดำเนินงานบางระบบอาจจะทำการเก็บข้อมูลในอดีตบ้างแต่ก็เป็นการเก็บข้อมูลเพื่อสนับสนุนการทำธุรกิจในแต่ละวันเท่านั้น

แต่ในส่วนของข้อมูลในคลังข้อมูลที่ทำการสร้าง/จัดเตรียมข้อมูลเชิงกลยุทธ์เพื่อช่วยเหลือผู้ใช้ในการวิเคราะห์ข้อมูลในหลายๆ แง่มุม เช่น ผู้ใช้คลังข้อมูลอาจจะต้องการรูปแบบการซื้อสินค้า (buying pattern) ของลูกค้าแต่ละรายโดยข้อมูลที่ผู้ใช้สนใจจะไม่ได้เป็นเพียงแค่ข้อมูลการซื้อสินค้าครั้งล่าสุดของลูกค้าแต่ละรายเท่านั้น แต่จะสนใจข้อมูลการซื้อครั้งก่อนๆ หน้าด้วย หรือในอีกกรณีหนึ่งที่ผู้ใช้คลังข้อมูลอาจต้องการทราบถึงเหตุผลที่ยอดขายสินค้าลดลงในแถบตะวันออกเฉียงเหนือ ด้วยความต้องการดังกล่าวผู้ใช้อาจจะต้องการข้อมูลยอดขายทั้งหมดที่เกิดขึ้นในเขตตะวันออกเฉียงเหนือที่เกิดขึ้นในช่วงเวลาที่ผ่านมา และในอีกกรณีหนึ่งนักการตลาดของบริษัทที่ทำธุรกิจร้านค้าปลีกจะต้องการที่จะโปรโมทสินค้า 2 รายการหรือมากกว่านั้น นักวิเคราะห์อาจต้องการยอดขายของรายการสินค้าที่ต้องการโปรโมทเทียบกับรายการสินค้าอื่นๆ ในแต่ละช่วงไตรมาสที่ผ่านๆมา เป็นต้น ซึ่งจากความต้องการที่ค่อนข้างจะหลากหลายของผู้ใช้ **คลังข้อมูลจะต้องทำการเก็บข้อมูลที่เป็นปัจจุบัน และข้อมูลย้อนหลัง** โดยมีข้อมูลแกนเวลาเข้ามาเกี่ยวข้อง ซึ่งการเก็บข้อมูลลักษณะนี้จะช่วยให้ผู้ใช้สามารถทราบถึงความเปลี่ยนแปลงของข้อมูลต่อช่วงเวลาต่างๆ ได้

ตัวอย่างเช่น ในคลังข้อมูลที่ประกอบไปด้วยข้อมูลยอดขายที่เป็นจำนวนชิ้นสินค้า ซึ่งในการจัดเก็บข้อมูลการขายสินค้าอาจจะมีการเก็บเวลาที่เกี่ยวข้องกับข้อมูลการขายสินค้านั้นๆ แนบไปกับข้อมูลจริงที่ต้องทำการเก็บ โดยเวลาที่ถูกจัดเก็บจะมีความละเอียดที่หลากหลาย เช่น ช่วงเวลา วัน เดือน ปี ที่มีการขายสินค้า เป็นต้น โดยในการจัดเก็บข้อมูลเราอาจจำเป็นต้องทำการจัดเก็บข้อมูลที่เป็นจำนวนชิ้นสินค้าที่ขายได้กับหน่วยของเวลา เพื่อบ่งบอกถึงยอดขายรายวัน ยอดขายแต่ละสัปดาห์ ยอดขายแต่ละเดือน หรือยอดขายแต่ละไตรมาส เป็นต้น ซึ่งจากตัวอย่างข้างต้นเราจะเห็นว่า **"*time-variant*"** นั้นเป็นจะเป็นการจัดเก็บข้อมูลเชิงเวลาแนบไว้กับข้อมูลส่วนอื่น และเป็นคุณลักษณะหนึ่งของคลังข้อมูลที่ทำให้ผู้ใช้สามารถวิเคราะห์ข้อมูลในอดีตและปัจจุบันเพื่อคาดการณ์ถึงสิ่งที่จะเกิดขึ้นในอนาคตได้



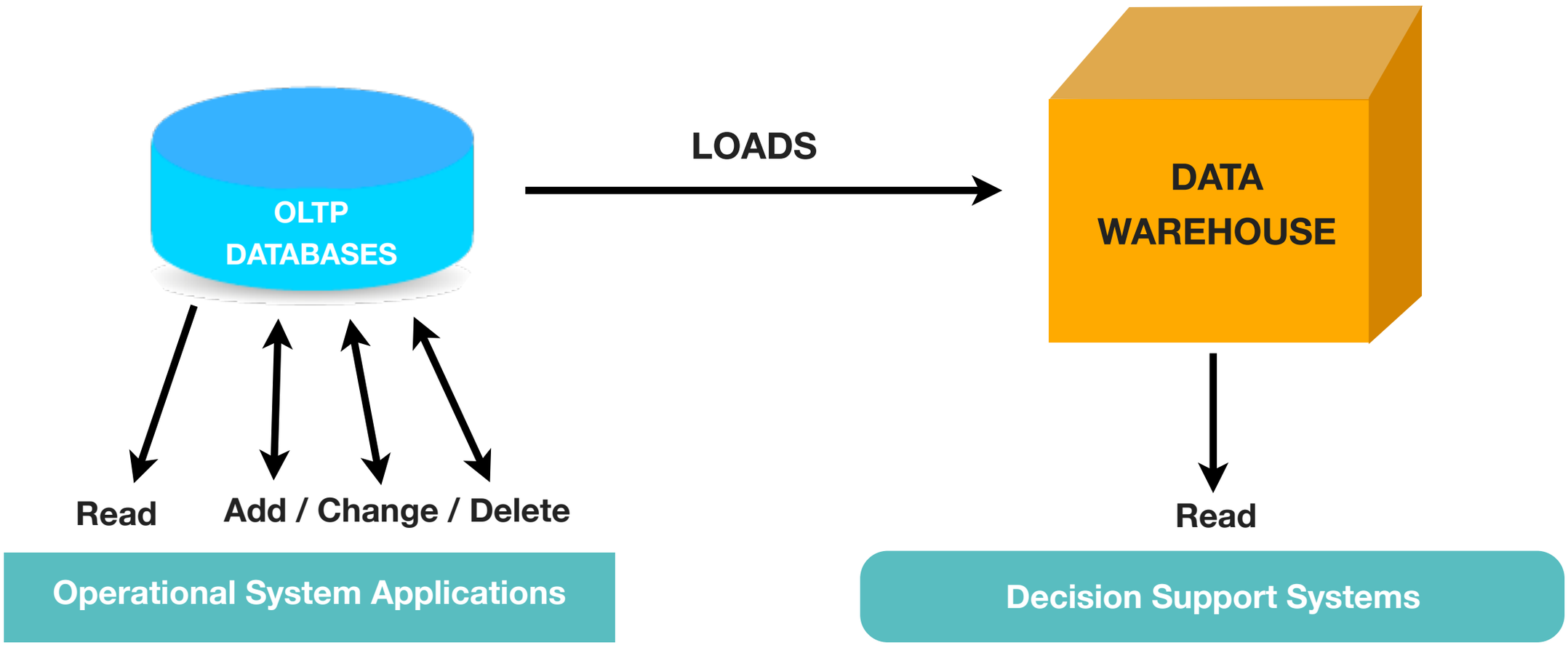
4

ข้อมูลที่ไม่เปลี่ยนแปลง (Nonvolatile data)

อย่างที่เราทราบจากบทที่แล้วว่า ระบบการดำเนินงานจะอนุญาตให้ผู้ใช้สามารถทำการเรียกดูข้อมูล (Select) เพิ่มข้อมูลลงในฐานข้อมูล (Insert) ลบข้อมูลลงในฐานข้อมูล (Delete) และอัพเดทข้อมูลต่างๆ (Update) ได้ ซึ่งเราจะสามารถดำเนินการได้ทุกๆ การทำงาน แต่ในส่วนของคลังข้อมูล การทำงานจะเริ่มจากการเลือกหรือสกัดข้อมูลที่ต้องการเพียงบางส่วนจากระบบการดำเนินงานและแหล่งข้อมูลอื่นๆ จากนั้นทำการรวบรวมข้อมูลเข้าด้วยกันและทำการประมวลผลข้อมูล จากนั้นค่อยทำการจัดเก็บข้อมูลลงในฐานข้อมูลเพื่อให้ผู้ใช้สามารถเรียกใช้ข้อมูลสำหรับการวิเคราะห์ต่างๆ ได้

จากขั้นตอนการทำงานของคลังข้อมูลดังกล่าว ข้อมูลจากระบบการดำเนินงานจะถูกเลือกออกมาเพียงบางส่วนและถูกเคลื่อนย้ายเข้าสู่คลังข้อมูล ในช่วงเวลาที่กำหนด โดยขึ้นอยู่กับความต้องการทางธุรกิจ เช่น ทำการเคลื่อนย้ายข้อมูลวันละสองครั้ง วันละครั้ง อาทิตย์ละครั้ง หรือสองอาทิตย์หนึ่งครั้ง เป็นต้น ซึ่งจากการทำงานดังกล่าว จะทำให้เห็นภาพได้ว่าขั้นตอนการทำงานแทบทั้งหมดจะถูกดำเนินการในระบบ ผู้ใช้งานคลังข้อมูลจะไม่สามารถทำการเพิ่ม ลบ หรืออัพเดทข้อมูลในคลังข้อมูลได้ ผู้ใช้งานจะสามารถเรียกดูข้อมูลได้เท่านั้น เพื่อให้เข้าใจมากขึ้น ลองพิจารณารูปที่ 2-3 ที่การทำธุรกรรมทางธุรกิจในแต่ละครั้งจะมีการอัพเดทระบบการดำเนินงานแบบทันที และผู้ใช้ระบบการดำเนินงานสามารถเพิ่ม เปลี่ยนแปลง และลบข้อมูลออกจากระบบการดำเนินงานได้ แต่สำหรับคลังข้อมูล จะไม่ทำการอัพเดทข้อมูลแบบทันท่วงที แต่จะทำการอัพเดทตามเวลาที่กำหนด และผู้ใช้ไม่สามารถลบข้อมูลออกจากคลังข้อมูลได้

Data inconsistencies are removed; data from diverse operational applications is integrated.



รูปที่ 2-3 คุณสมบัติการไม่เปลี่ยนแปลงข้อมูลในคลังข้อมูล

5

ข้อมูลที่มีรายละเอียดหลายระดับ (Data granularity)

การเก็บข้อมูลในระบบการดำเนินงานมักจะทำการเก็บข้อมูลในลักษณะที่มีความละเอียดค่อนข้างสูง อาทิเช่น การขายของร้านค้าปลีกจะทำการเก็บจำนวนสินค้าที่ขายได้ในแต่ละรายการที่จุดแคชเชียร์คิดเงิน หรือการสั่งสินค้าจะทำการเก็บจำนวนสินค้าที่สั่งในแต่ละครั้ง เป็นต้น โดยส่วนใหญ่ของระบบการดำเนินงานจะไม่ทำการเก็บข้อมูลที่เป็นผลสรุปแต่จะเน้นที่การเก็บข้อมูลแต่ละรายการ (transaction) เพื่อการดำเนินธุรกิจในแต่ละวัน

แต่เมื่อไรก็ตามที่เราต้องการข้อมูลที่เป็นผลสรุปจากระบบการดำเนินงาน เราจะต้องทำการรวมข้อมูลแต่ละรายการเข้าด้วยกัน เช่น ถ้าเราต้องการยอดขายของสินค้าชนิดหนึ่งในเดือนมิถุนายน เราจะต้องทำการอ่านข้อมูลการขายสินค้าทั้งหมดของเดือนมิถุนายนจากนั้นทำการรวมข้อมูลการขายสินค้าเหล่านั้นเพื่อให้ได้เป็นข้อมูลที่เป็นผลสรุปที่เราต้องการ แต่ในกรณีของคลังข้อมูลที่ผู้ใช้มักจะทำการเรียกดูข้อมูลที่เป็นผลสรุป ซึ่งผู้ใช้อาจทำการเรียกดูข้อมูลจำนวนสินค้ารายการหนึ่งๆ ที่ขายได้ในภาคตะวันออก จากนั้นค่อยเพิ่มความละเอียดของข้อมูลที่ต้องการได้รับขึ้น ซึ่งข้อมูลที่ต้องการอาจเป็นยอดขายของสินค้ารายการหนึ่งที่ได้ขายได้ในแต่ละจังหวัดของภาคตะวันออก และอาจเจาะลึกไปถึงข้อมูลยอดขายของสินค้ารายการหนึ่งที่ขายได้ในแต่ละสาขาที่อยู่ในภาคตะวันออก เป็นต้น ซึ่งโดยส่วนใหญ่แล้วผู้ใช้มักจะเริ่มจากการเรียกดูข้อมูลที่มีรายละเอียดน้อยแล้วค่อยๆเพิ่มรายละเอียดขึ้นเรื่อย ๆ

จากการใช้งานคลังข้อมูลข้างต้น การจัดเก็บข้อมูลในคลังข้อมูลโดยส่วนใหญ่จะทำการเก็บข้อมูลที่เป็นแบบผลรวมหรือผลสรุปที่มีความละเอียดแตกต่างกันตามความต้องการของผู้ใช้ ซึ่งถ้าคลังข้อมูลมีการจัดเก็บข้อมูลในลักษณะที่มีความละเอียดสูงจะทำให้เราต้องทำการเก็บข้อมูลเป็นจำนวนมาก ดังนั้นในการจัดเก็บข้อมูล เราจะต้องทำการกำหนดระดับความละเอียดของข้อมูลตามชนิดข้อมูลในคลังข้อมูล และทำการพิจารณาถึงประสิทธิภาพที่คาดหวังจากการค้นคืนผลลัพธ์จากคลังข้อมูล

ในการที่จะทำความเข้าใจเกี่ยวกับระดับความละเอียดของข้อมูลในคลังข้อมูล ลองพิจารณารูปที่ 2-4 ที่ประกอบไปด้วยการจัดเก็บข้อมูลในคลังข้อมูลของธนาคารหนึ่งๆ ที่มีความละเอียดแตกต่างกัน 3 ระดับนั่นคือ การเก็บข้อมูลรายวัน รายเดือน และรายไตรมาส ตามลำดับ

THREE DATA LEVELS IN A BANKING DATA WAREHOUSE

Daily Detail	Monthly Summary	Quarterly Summary
Account	Account	Account
Activity Date	Month	Quarter
Amount	Number of transactions	Number of transactions
Deposit/Withdrawal	Withdrawals	Withdrawals
	Deposits	Deposits
	Beginning Balance	Beginning Balance
	Ending Balance	Ending Balance

Data granularity refers to the level of detail. Depending on the requirements, multiple levels of detail may be present. Many data warehouses have at least dual levels of granularity.

รูปที่ 2-4 ความละเอียดของข้อมูลในคลังข้อมูล

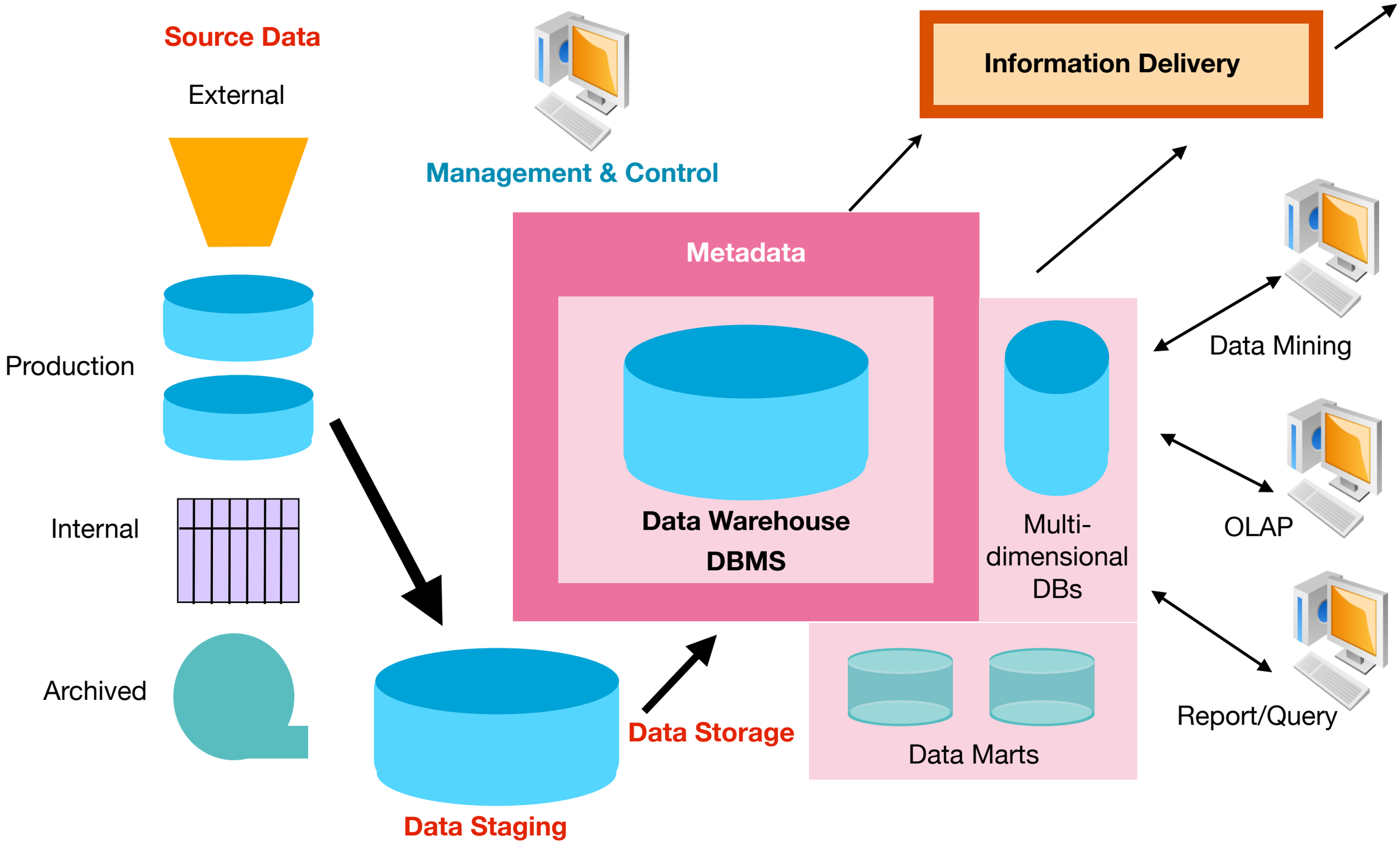
SECTION 4

ส่วนประกอบของคลังข้อมูล

หลังจากทราบถึงคุณลักษณะต่างๆ รวมถึงฟังก์ชันการทำงานหลักของคลังข้อมูล ในส่วนนี้จะพิจารณาถึงส่วนประกอบต่างๆของคลังข้อมูล ซึ่ง ในการสร้างคลังข้อมูลจะคิด พิจารณา และเกี่ยวข้องกับส่วนประกอบต่างๆ ทั้ง ในส่วนของซอร์ฟแวร์และฮาร์ดแวร์ โดยเราจะต้องทำการรวมส่วนประกอบเหล่านี้เข้าด้วยกันและทำการปรับแต่งส่วนประกอบเหล่านี้เพื่อ ให้คลังข้อมูลมีการทำงานที่มีประสิทธิภาพและประโยชน์สูงสุด

การเลือกส่วนประกอบที่จะใช้ในคลังข้อมูลจะขึ้นอยู่กับข้อจำกัดและความต้องการของแต่ละองค์กรเป็นหลัก โดยส่วนประกอบพื้นฐานของคลังข้อมูลจะถูกแสดง ในรูปที่ 2-5 ซึ่งจากทางซ้ายสุดคือแหล่งข้อมูล (Source data) และ ถัดมาคือ “data staging” หรือ “staging area” ที่เป็นตัวกลางหรือเป็นที่พักข้อมูลที่ถูกสกัด/ถูกเลือกมาจากแหล่งข้อมูล จากนั้นจะเป็นส่วนของพื้นที่ ในการจัดเก็บข้อมูลและเมตาดาต้า (Data and metadata storage) และทางด้านขวาสุดจะเป็นส่วนของระบบที่ ใช้สำหรับเข้าถึงหรือส่งผ่านข้อมูลไปยังผู้ใช้งาน (Information delivery) ที่ประกอบไปด้วยวิธีการต่างๆ สำหรับการส่งข้อมูลให้กับผู้ใช้ และยังรวมถึงเครื่องมือต่างๆที่ ใช้ในการวิเคราะห์ข้อมูลที่ซับซ้อน อาทิ เครื่องมือสำหรับการทำเหมืองข้อมูล (Data mining) เครื่องมือสำหรับวิเคราะห์ข้อมูลแบบออนไลน์ (Online analytical processing, OLAP) และ เครื่องมือ ในการสร้างคิวรีและรายงานต่างๆ (Query and report tools) เป็นต้น

Architecture is the proper arrangement of the components.



รูปที่ 2-5 ส่วนประกอบของคลังข้อมูล

ในการที่จะทำความเข้าใจส่วนประกอบหลักทั้ง 4 ส่วน ดังแสดงในรูปที่ 2-5 เราควรที่จะต้องทำการศึกษาถึงรายละเอียดของแต่ละส่วนที่จะใช้ในการสร้างคลังข้อมูล โดยรายละเอียดของแต่ละส่วนจะสามารถอธิบายได้ดังนี้

- แหล่งข้อมูลของคลังข้อมูล
- พื้นที่พักข้อมูล
- พื้นที่สำหรับจัดเก็บข้อมูล
- ระบบเข้าถึงและส่งผ่านข้อมูลไปยังผู้ใช้
- ส่วนงานการจัดการและการควบคุมต่างๆ
- การจัดเก็บเมตาดาต้า

แหล่งข้อมูลของคลังข้อมูล

แหล่งข้อมูลของคลังข้อมูลสามารถแบ่งได้เป็น 4 ประเภท ดังนี้

Production data

Internal data

Archieved data

External data

Production data จะเป็นข้อมูลที่มาจากระบบการดำเนินงานหลายระบบด้วยกัน เช่น ระบบการเงิน ระบบการผลิต ระบบการสั่งซื้อสินค้า ระบบตลอดห่วงโซ่อุปทาน และระบบการจัดการความสัมพันธ์ลูกค้า เป็นต้น

ข้อมูลเหล่านี้จะถูกเลือกหรือสกัดมาจากระบบการดำเนินงาน โดยทำการเลือกจากความต้องการข้อมูลในคลังข้อมูล แต่ในการเลือกข้อมูลจากระบบการดำเนินงานหลายๆ ระบบอาจทำให้เราต้องพบกับข้อมูลที่มีรูปแบบที่หลายหลาย เช่น ข้อมูลอาจมาจากฮาร์ดแวร์ที่แตกต่างกัน ระบบฐานข้อมูลที่แตกต่างกัน ระบบปฏิบัติการที่ต่างกัน เป็นต้น โดยข้อมูลที่ได้รับจากระบบการดำเนินงานที่แตกต่างกันอาจมีความไม่สอดคล้องกันของข้อมูลเจือปนอยู่ด้วย ดังนั้นเมื่อเราได้รับข้อมูลจากระบบการดำเนินงานแล้วเราจะต้องทำให้ข้อมูลเหล่านั้นเป็นมาตราฐานเดียวกันด้วย

แหล่งข้อมูลของคลังข้อมูล

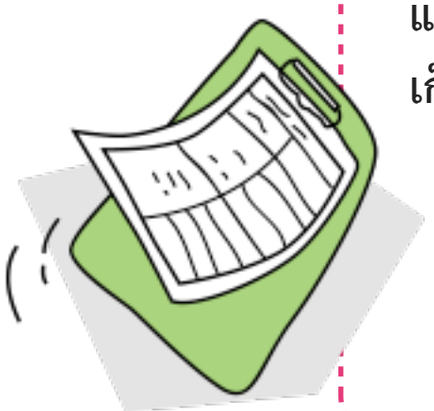
แหล่งข้อมูลของคลังข้อมูลสามารถแบ่งได้เป็น 4 ประเภท ดังนี้

Production data

Internal data

Archieved data

External data



Internal data จะเป็นข้อมูลประเภทสเปรดชีท เอกสารต่างๆ ที่แสดงรายละเอียดของลูกค้าหรือ ฐานข้อมูลของแผนกที่ถูกสร้างขึ้นไว้ใช้งานส่วนตัว ซึ่งถูกเก็บไว้ใช้ในการดำเนินงานบางอย่างของการดำเนินธุรกิจในแต่ละแผนก ข้อมูลเหล่านี้จะเป็นข้อมูลที่อาจมีความสำคัญที่เราไม่สามารถละเลยได้ ในการเก็บรวบรวมข้อมูลที่เป็น ***"internal data"*** จากผู้ใช้ เราอาจต้องทำการตัดสินใจว่าเราควรเก็บ internal data เป็นจำนวนเท่าใด โดยข้อมูลเหล่านี้จะทำให้ข้อมูลในคลังข้อมูลมีปริมาณเพิ่มขึ้นและยังเป็นการเพิ่มความซับซ้อนให้กับขั้นตอนการทำงานของคลังข้อมูลอีกด้วย (จะทำให้ขั้นตอนการรวบรวมข้อมูล และขั้นตอนการทำให้ข้อมูลเป็นมาตราฐานจะมีความซับซ้อนและยุ่งยากมากขึ้น)

ดังนั้น ในการที่จะพิจารณาที่จะจัดเก็บข้อมูลที่เป็น ***"internal data"*** เราควรที่จะต้องหาหรือออกแบบวิธีการเลือกข้อมูลเหล่านี้ ค้นหาวิธีในการเข้าถึงและสกัดข้อมูลจากเอกสารต่างๆ และ พิจารณาการรวมฐานข้อมูลย่อยๆ ที่ถูกจัดเก็บไว้ในแต่ละแผนกเข้าด้วยกัน

แหล่งข้อมูลของคลังข้อมูล

แหล่งข้อมูลของคลังข้อมูลสามารถแบ่งได้เป็น 4 ประเภท ดังนี้

Production data

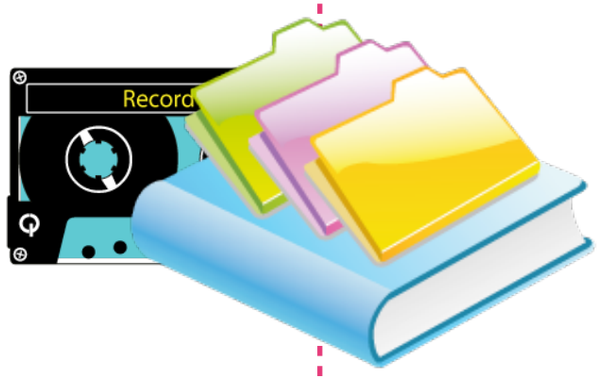
Internal data

Archieved data

External data

Archieved data จะเป็นข้อมูลเก่าที่อาจมีอายุมากกว่า 1-5 ปีขึ้นไป โดยข้อมูลเหล่านี้อาจจะไม่ได้ถูกจัดเก็บอยู่ในฐานข้อมูลที่จะมีข้อมูลใหม่ๆ แต่จะถูกจัดเก็บไว้ในแฟ้มข้อมูล ดิสก์ หรือเทป เป็นต้น

ข้อมูลที่เป็น **“archieved data”** จะเป็นข้อมูลชนิดหนึ่งที่มีความสำคัญกับคลังข้อมูล เนื่องจากคลังข้อมูลจะมีคุณลักษณะหนึ่งที่มีการพิจารณาข้อมูลย้อนหลังเพื่อใช้ในการวิเคราะห์รูปแบบของข้อมูลและวิเคราะห์แนวโน้มของข้อมูล ดังนั้น เราอาจจำเป็นต้องทำการเก็บข้อมูลที่ค่อนข้างเก่าเหล่านี้ไว้ในคลังข้อมูลด้วยเช่นกัน



แหล่งข้อมูลของคลังข้อมูล

แหล่งข้อมูลของคลังข้อมูลสามารถแบ่งได้เป็น 4 ประเภท ดังนี้

Production data

Internal data

Archieved data

External data

External data จะเป็นข้อมูลเกี่ยวกับสถิติในภาคอุตสาหรรมที่สร้างขึ้นจากบริษัทภายนอก และ หน่วยงานราชการต่างๆ ที่ผู้บริหารจะใช้ข้อมูลเหล่านี้เพื่อใช้ประกอบการตัดสินใจบางอย่าง ด้วยเหตุนี้ในการสร้างคลังข้อมูลเราอาจจำเป็นต้องทำการเก็บข้อมูลจากแหล่งข้อมูลภายนอกไว้ เพื่อทำการวิเคราะห์แนวโน้มของอุตสาหกรรม และเปรียบเทียบประสิทธิภาพระหว่างบริษัทของเรากับองค์กรอื่นๆ เป็นต้น

แต่อย่างไรก็ดีข้อมูลจากภายนอกจะมีการจัดรูปแบบของข้อมูลที่ไม่เหมือนกับข้อมูลภายใน เราจะต้องทำการออกแบบวิธีในการแปลงข้อมูลให้อยู่ในรูปแบบเดียวกับข้อมูลภายใน และเราอาจจำเป็นต้องจัดการกับการส่งผ่านข้อมูลจากแหล่งข้อมูลภายนอกอีกด้วย

พื้นที่พักข้อมูล

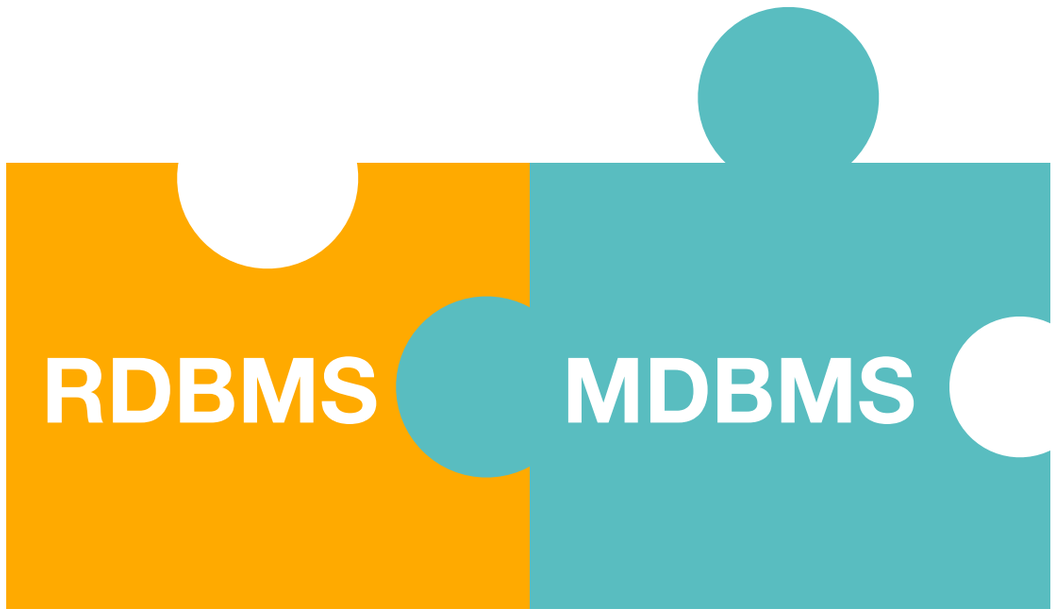
หลังจากที่เราทำการเลือกหรือสกัดข้อมูลที่ต้องการจากระบบการดำเนินงานต่างๆ และจากแหล่งข้อมูลภายนอกแล้ว เราจะต้องทำการประมวลผลข้อมูลเหล่านั้นก่อนที่จะทำการนำข้อมูลเหล่านั้นไปจัดเก็บไว้ในคลังข้อมูล โดยในการประมวลผลข้อมูลเราจะต้องทำการเปลี่ยน แปลง และจัดข้อมูลให้อยู่ในรูปแบบเหมาะสมต่อโครงสร้างข้อมูลคลังข้อมูล แต่เนื่องจากข้อมูลในระบบการดำเนินงานและคลังข้อมูลมีความแตกต่างกัน และถูกแยกออกจากกัน

กล่าวคือ ข้อมูลในระบบการดำเนินงานจะถูกจัดเก็บตามแอพลิเคชันเพื่อสนับสนุนการดำเนินธุรกิจในแต่ละวัน แต่ข้อมูลในคลังข้อมูลจะถูกจัดเก็บตามหัวข้อทางธุรกิจ ซึ่งข้อมูลอาจมาจากการสรุปข้อมูลในแง่มุมต่างๆ เมื่อข้อมูลของทั้งสองระบบมีความแตกต่างกัน เราจึงจำเป็นต้องสร้างที่พักข้อมูล ที่อยู่ระหว่างระบบการดำเนินงานและคลังข้อมูล ซึ่งก็คือ “data stagning” หรือ “staging area” ที่จะประกอบไปด้วยพื้นที่สำหรับจัดเก็บข้อมูลที่สกัดได้จากระบบการดำเนินงาน และฟังก์ชันการทำงานต่างๆ เช่น การทำความสะอาดข้อมูล การเปลี่ยนแปลงข้อมูล การรวมข้อมูลเข้าด้วยกัน เพื่อทำการเตรียมข้อมูลสำหรับจัดเก็บในคลังข้อมูลต่อไป

พื้นที่สำหรับจัดเก็บข้อมูล

ในส่วนของพื้นที่สำหรับจัดเก็บข้อมูลในคลังข้อมูลจะเป็นส่วนที่แยกออกมาจากระบบการดำเนินงาน โดยในการจัดเก็บข้อมูลเราอาจเรียกใช้เครื่องมือต่างๆ ที่มีวางจำหน่าย หรือทำการสร้างฟังก์ชันการจัดเก็บข้อมูลขึ้นเอง ซึ่งโดยส่วนใหญ่แล้วคลังข้อมูลจะใช้ **“Relational DBMS, RDBMS”** ในการจัดเก็บข้อมูล แต่ก็มีบางคลังข้อมูลใช้ **“Multidimensional DBMS, MDBMS”** เพื่อเก็บข้อมูลด้วยเช่นกัน

ในการจัดเก็บข้อมูล ผู้สร้างคลังข้อมูลควรจะเน้นย้ำที่การจัดเก็บข้อมูลทั้งในปัจจุบันและข้อมูลย้อนหลัง รวมถึงการวางแผนหรือออกแบบโครงสร้างของข้อมูลที่จะใช้ในการวิเคราะห์ข้อมูลเหล่านั้น นอกจากนี้ยังต้องคำถึงประสิทธิภาพในการเรียกใช้ข้อมูลอีกด้วย

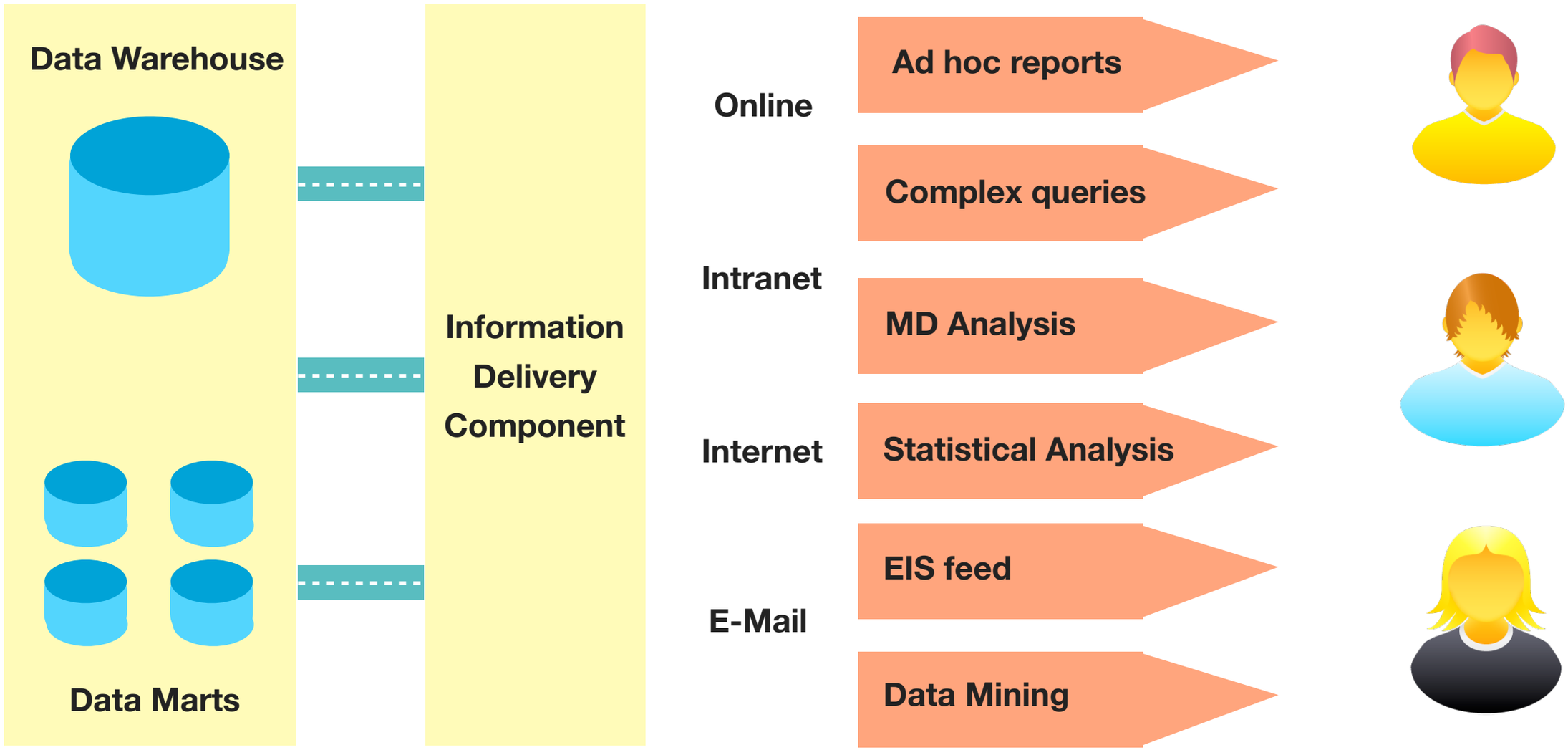


ระบบเข้าถึงและส่งผ่านข้อมูลไปยังผู้ใช้

ในการพิจารณาเกี่ยวกับการส่งผ่านข้อมูลให้กับผู้ใช้ เราจะต้องทราบก่อนว่าคลังข้อมูลที่เราสร้างขึ้นนั้นมีผู้ใช้กี่ประเภท? แต่ละประเภทเป็นใครบ้าง? และผู้ใช้แต่ละประเภทหรือแต่ละรายต้องการข้อมูลประเภทใด? โดยส่วนใหญ่ของผู้ใช้ที่เพิ่งเริ่มใช้งานที่ยังไม่ได้ผ่านการอบรมและยังไม่มีประสบการณ์ในการใช้งานคลังข้อมูลมาก่อน รวมถึงผู้ใช้งานคลังข้อมูลแบบเป็นครั้งคราวมักจะต้องการรายงานและการประมวลผลคิวรีที่ระบบคลังข้อมูลกำหนดไว้ให้หรือจัดเตรียมไว้ให้อยู่แล้ว ในขณะที่นักวิเคราะห์ทางธุรกิจต้องการที่จะวิเคราะห์ข้อมูลที่มีความซับซ้อน และผู้ใช้ที่มีอำนาจในการตัดสินใจมักจะต้องการที่จะเรียกดูข้อมูลที่น่าสนใจ เป็นต้น

เมื่อผู้ใช้งานมีหลายกลุ่มและมีความต้องการที่หลากหลาย เราอาจจำเป็นต้องทำการออกแบบหรือกำหนดฟังก์ชันการส่งข้อมูลที่แตกต่างกันเพื่อตอบสนองต่อการส่งข้อมูลที่หลากหลายให้กับผู้ใช้แต่ละประเภท ดังแสดงในรูปที่ 2-6 ที่แสดงฟังก์ชันการส่งข้อมูล 4 วิธีด้วยกัน

ซึ่งโดยส่วนใหญ่การส่งข้อมูลที่เป็นคิวรีและรายงานต่างๆ จะเป็นแบบออนไลน์ที่อนุญาตให้ผู้ใช้สามารถรับข้อมูลได้อย่างทันทีที่มีการส่งคิวรีที่ต้องการไปประมวลผลที่คลังข้อมูล อีกวิธีการหนึ่งที่ได้รับความนิยมลดหลั่นลงมาคือการตั้งเวลาในการส่งคิวรีไปยังคลังข้อมูลเพื่อประมวลผลและการเรียกดูรายงานตามช่วงเวลาที่กำหนด โดยหลังจากทำการตั้งเวลาแล้วผู้ใช้จะได้รับรายงานต่างๆ ผ่านทางอีเมล์ หรือเราอาจใช้อินเทอร์เน็ตของบริษัทในการเรียกดูข้อมูลได้เช่นกัน ซึ่งในปัจจุบันเทคโนโลยีอินทราเน็ตและอินเทอร์เน็ตจะเป็นวิธีที่ได้รับความนิยมอย่างมากในองค์กรใหญ่ๆ



รูปที่ 2-6 ระบบการเข้าถึงและส่งผ่านข้อมูลไปยังผู้ใช้งาน

ส่วนงานการจัดการและการควบคุมต่าง ๆ

ส่วนงานการจัดการและควบคุมจะทำหน้าที่ประสานงานในกิจกรรมและการให้บริการต่างๆ ภายในคลังข้อมูล การทำงานของส่วนงานนี้จะประกอบไปด้วยการควบคุมการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล การถ่ายโอนข้อมูลไปยังคลังข้อมูล และการควบคุมการส่งข้อมูลให้กับผู้ใช้ นอกจากนี้ยังเป็นการประกอบการทำงานร่วมกับระบบจัดการฐานข้อมูล (DBMS) จัดการให้ข้อมูลถูกจัดเก็บในที่เก็บข้อมูลอย่างเหมาะสม คอยเฝ้าดูการเคลื่อนที่ของข้อมูลไปยัง staging area และข้อมูลที่ออกจาก staging Area ไปยังที่เก็บข้อมูลของคลังข้อมูลอีกด้วย

การจัดเก็บเมตาดาต้า

เมตาดาต้าในคลังข้อมูลเปรียบเสมือนพจนานุกรมข้อมูลหรือแคทตาล็อกของข้อมูลในระบบจัดการฐานข้อมูล (Data dictionary/data catalog of DBMS) ซึ่งภายในพจนานุกรมข้อมูลจะมีการจัดเก็บข้อมูลที่เกี่ยวข้องกับโครงสร้างการจัดเก็บข้อมูลต่างๆ เช่น การจัดเก็บข้อมูลเกี่ยวกับแฟ้มที่ทำการเก็บข้อมูล รวมถึง address ต่างๆ ข้อมูลเกี่ยวกับ index และ อื่นๆ

นอกจากนั้น เรายังสามารถเปรียบเทียบเมตาดาต้าในเชิงกว้างๆ ได้ ซึ่งเมตาดาต้านั้นสามารถเปรียบได้กับข้อมูลตำแหน่งของเมืองที่เราอยู่ซึ่งในบางครั้งอาจต้องการข้อมูลเกี่ยวกับห้างร้านต่างๆ ในเมืองของคุณว่าร้านแห่งนั้นอยู่ที่ใด ชื่ออะไรบ้าง มีสินค้าอะไรในร้านเหล่านั้นบ้าง เมื่อเราต้องการข้อมูลเราควรเปิดสมุดหน้าเหลือง เมตาดาต้าทำหน้าที่เหมือนกับไดเร็กทอรี่ของข้อมูลสำหรับคลังข้อมูลของคุณ ดังนั้น เมตาดาต้าจึงมีความสำคัญสำหรับการสร้างและการใช้คลังข้อมูลเป็นอย่างมาก เช่น

(1) เมตาดาต้าทำหน้าที่เสมือนการเชื่อมส่วนต่างๆ ของคลังข้อมูลเข้าด้วยกัน

(2) เมตาดาต้าจะช่วยให้ผู้พัฒนาคลังข้อมูลเข้าใจถึงเนื้อหา/ข้อมูล และโครงสร้างของฐานข้อมูล

(3) เมตาดาต้าจะช่วยให้ผู้ใช้สามารถจำเนื้อหา/ข้อมูลเฉพาะทาง/คำศัพท์ของพวกเขาได้ เป็นต้น

ข้อมูลที่เป็นเมตาดาต้าจะสามารถแบ่งกลุ่มได้เป็น 3 ชนิดหลักๆ ดังนี้

1

เมตาดาต้าที่ได้มาจากระบบดำเนินงาน (Operational metadata)

2

เมตาดาต้าที่ได้มากจากการเลือกข้อมูล และการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล (Extract and transformation metadata)

3

เมตาดาต้าสำหรับผู้ใช้ในการใช้งานคลังข้อมูล (End-user metadata)

M A T A D A T A

1

เมตาดาต้าที่ได้มาจากระบบดำเนินงาน (Operational metadata)

เมตาดาต้าที่ได้มาจากระบบดำเนินงาน (Operational metadata) จะเป็นข้อมูลรายละเอียดต่างๆ ที่เกี่ยวข้องกับระบบการดำเนินงาน เช่น ชื่อฐานข้อมูลของระบบ ชื่อตารางต่างๆ รวมถึง ชื่อฟิลด์หรือแอทริบิวที่เราสนใจ โครงสร้างข้อมูล ชนิดของข้อมูล ความยาวของข้อมูลในแต่ละฟิลด์ที่เราสนใจ และอื่นๆ

2

เมตาดาต้าที่ได้มากจากการเลือกข้อมูล และการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล (Extract and transformation metadata)

เมตาดาต้าที่ได้มากจากการเลือกข้อมูล และการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล (Extract and transformation metadata) จะเป็นข้อมูลที่เกี่ยวกับการเลือกข้อมูลจากแหล่งข้อมูล เมตาดาต้าลักษณะนี้จะประกอบไปด้วย ชื่อต่างๆ ของข้อมูล/ตาราง หรืออื่นๆ ที่เกี่ยวข้องกับการเลือกข้อมูล ความถี่ในการสกัดข้อมูล วิธีการสกัดข้อมูล และกฎทางธุรกิจ (Business Rule) สำหรับการสกัดข้อมูล เป็นต้น

3

เมตาดาต้าสำหรับผู้ใช้ในการใช้งานคลังข้อมูล (End-user metadata)

เมตาดาต้าสำหรับผู้ใช้ในการใช้งานคลังข้อมูล (End-user metadata) จะเปรียบเสมือนแผนที่ของคลังข้อมูลที่ช่วยให้ผู้ใช้สามารถค้นหาข้อมูลและสารสนเทศจากคลังข้อมูลได้ โดย “End-User metadata” จะอนุญาตและยอมให้ผู้ใช้ทำการใช้คำศัพท์เฉพาะที่เกี่ยวกับธุรกิจในการเรียกดูข้อมูลจากคลังข้อมูลได้อีกด้วย

เมตาดาต้าทั้ง 3 ชนิดข้างต้นจะมีประโยชน์และมีวัตถุประสงค์ของการจัดเก็บข้อมูลที่แตกต่างกัน โดยในการเรียกใช้ข้อมูลเมตาดาต้านั้นสามารถทำได้หลายแง่มุม เช่น เมื่อผู้ใช้ต้องการใช้งานคลังข้อมูล ผู้ใช้จะสามารถเรียดดูข้อมูลเมตาดาต้าเพื่อที่จะทราบถึงชื่อตารางหรือชื่อฟิลด์ที่เก็บข้อมูลที่พวกเขาต้องการได้ หรือ ในส่วนของส่วนงานการจัดการและการควบคุมต่างๆ (ดังแสดงในรูปที่ 2-5) จะมีการเรียกใช้เมตาดาต้าเพื่อทำการจัดการและควบคุมการทำงานต่างๆ เนื่องจากเมตาดาต้าจะมีส่วนของข้อมูลเกี่ยวกับแหล่งข้อมูล พารามิเตอร์ต่างๆ ขั้นตอนการทำงานต่าง ชื่อของฐานข้อมูล และอื่นๆ โดยที่ข้อมูลเหล่านี้จะใช้ในการจัดการ และควบคุมกระบวนการทำงานของคลังข้อมูลทั้งหมด

จากที่กล่าวมาทั้งหมดข้างต้น เราจะเห็นภาพกว้างๆ ว่าเมตาดาต้านั้นมีความสำคัญต่อการสร้างการทำงาน และการใช้งานคลังข้อมูล ดังนั้นเราจะทำการศึกษาถึงรายละเอียดของเมตาดาต้าอีกครั้งหนึ่งในบทที่ 9

SECTION 5

คลังข้อมูลและดาต้ามาร์ท

หลังจากที่เราทราบถึงคุณลักษณะและส่วนประกอบต่างๆ ของคลังข้อมูลแล้ว เราจะสามารถกล่าวได้ว่า “คลังข้อมูลนั้นเป็นระบบสำหรับสร้างหรือจัดเตรียมข้อมูลที่เป็นผลสรุป” โดยข้อมูลที่ถูกจัดเก็บอยู่ในคลังข้อมูลจะถูกจัดเก็บตามหัวข้อทางธุรกิจต่างๆ ที่ผู้ใช้งานสนใจ ข้อมูลจะถูกรวบรวมมาจากระบบการดำเนินงานต่างๆ ที่หลากหลายรวมถึงแหล่งข้อมูลภายนอกด้วย ข้อมูลในคลังข้อมูลจะมีแกนเวลาเข้ามาเกี่ยวข้องเสมอ มีความละเอียดหลายระดับ และข้อมูลในคลังข้อมูลจะไม่ถูกเปลี่ยนแปลงจากผู้ใช้งาน แต่อย่างไรก็ดี สำหรับคนที่ยังไม่มีประสบการณ์เกี่ยวกับคลังข้อมูลมากนักอาจได้ยินหรือรับรู้อีกสิ่งหนึ่ง นั่นคือ **“ดาต้ามาร์ท (data mart)”** และอาจเกิดความสับสนเกี่ยวกับความสัมพันธ์และความสอดคล้องระหว่าง**“คลังข้อมูล”** และ **“ดาต้ามาร์ท”** ก็เป็นได้หลายๆ คนจะมองว่าสองคำนี้เหมือนกันหรือสื่อถึงสิ่งเดียวกัน แต่แท้จริงแล้วทั้งสองคำนี้ไม่เหมือนกัน ซึ่งจากผู้ที่มีประสบการณ์กับคลังข้อมูลจะมองว่า

ซึ่งจากความแตกต่างในเรื่องของฟังก์ชันการทำงาน จำนวนข้อมูล ผู้ใช้งาน และปัจจัยอื่นๆ ระหว่างระบบทั้งสอง เราจะเป็นต้องพิจารณาสิ่งต่างๆ ดังต่อไปนี้

- ในการสร้างคลังข้อมูลเราควรใช้วิธีใดระหว่างการสร้างคลังข้อมูลแบบ top-down หรือ bottom-up? กล่าวคือเราจะทำการสร้างคลังข้อมูลโดยพิจารณารายละเอียดทั้งหมดก่อน แล้วจึงทำการสร้างข้อมูลแต่ละส่วน หรือเราจะทำการสร้างคลังข้อมูลส่วนย่อยๆ ก่อน แล้วค่อยทำการรวมคลังข้อมูลเหล่านั้นให้เป็นคลังข้อมูลสำหรับทั้งองค์กร
    - เราควรจะทำการสร้างคลังข้อมูลประเภทใดระหว่างคลังข้อมูลสำหรับทั้งองค์กรหรือคลังข้อมูลของแต่ละแผนก?
        - เราควรสร้างอย่างใดอย่างหนึ่ง<u>ก่อน</u>ระหว่างคลังข้อมูลสำหรับทั้งองค์กร หรือคลังข้อมูลของแต่ละแผนก?
            - ถ้าเราทำการสร้างคลังข้อมูลของแต่ละแผนก เราควรสร้างคลังข้อมูลในลักษณะเป็นแบบที่เป็นดาต้ามาร์ทที่เชื่อมต่อกันหรือเป็นอิสระต่อกัน (Dependent or Independent data mart)?

จากปัจจัยข้างต้น จะมีคำถามอื่นๆ ตามมาอีกมากมาย เช่น เราจะต้องมองภาพรวมกว้างๆ ของทั้งองค์กรเพื่อทำการสร้างคลังข้อมูลแบบ top-down ใช่หรือไม่? หรือ เราควรจะเริ่มจากการสร้างคลังข้อมูลแบบ bottom-up โดยทำการพิจารณาความต้องการของแต่ละส่วนงาน/แผนกใช่หรือไม่? เราควรจะสร้างคลังข้อมูลขนาดใหญ่แล้วทำการจัดเก็บข้อมูลลงในแต่ละดาต้ามาร์ทหรือไม่? หรือเราควรจะสร้างแต่ละดาต้ามาร์ทแล้วค่อยทำการรวมดาต้ามาร์ทที่สร้างขึ้นให้เป็นคลังข้อมูลขนาดใหญ่? เราควรสร้างดาต้ามาร์ทให้เป็นอิสระต่อกันหรือไม่? หรือเราควรสร้างดาต้ามาร์ทให้มีความเกี่ยวเนื่องกันไหม? คำถามเหล่านี้เป็นคำถามที่สำคัญและส่งผลต่องบประมาณการสร้างคลังข้อมูล ส่งผลต่อระยะเวลาในการสร้างคลังข้อมูล และส่งผลต่อรูปแบบการใช้งานคลังข้อมูล

ดังนั้นก่อนที่เราจะทำการสร้างคลังข้อมูลเราควรจะต้องพิจารณาให้รอบคอบว่าเราควรจะสร้างคลังข้อมูลในลักษณะใด โดยใช้วิธีการอะไร เพื่อควบคุมค่าใช้จ่าย เวลาในการดำเนินงาน และฟังก์ชันการทำงานที่ครบครันที่สุด แต่ก่อนที่เราจะตอบปัญหาต่างๆ ข้างต้น ลองพิจารณารูปที่ 2-7 เพื่อทราบถึงความแตกต่างระหว่าง **“data warehouse”** และ **“data mart”** เพื่อที่จะได้ทำการออกแบบหรือกำหนดการสร้างคลังข้อมูลต่อไป

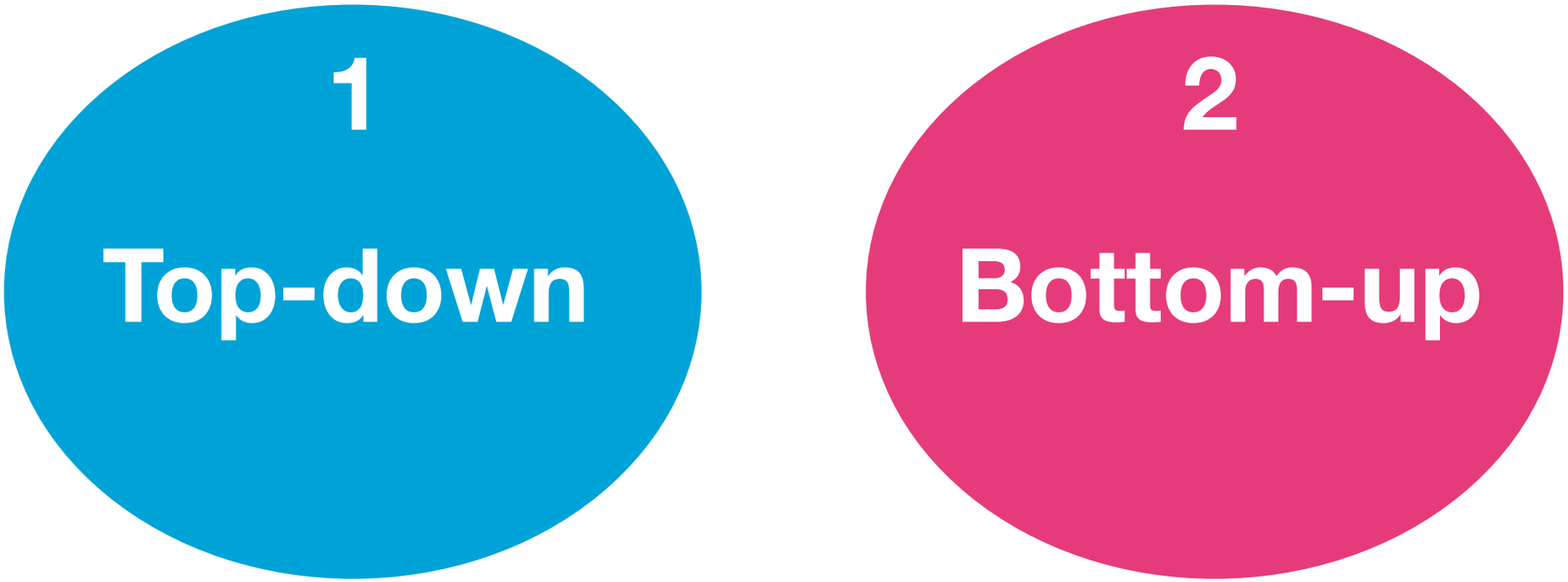
Monthly Summary	Quarterly Summary
Corporate/Enterprise-wide	Departmental
Union of all data marts	A single business process
Data received from staging area	STARjoin (facts & dimensions)
Queries on presentation resource	Technology optimal for data access and analysis
Structure for corporate view of data	Structure to suit the departmental view of data
Organized on E-R model	

รูปที่ 2-7 ความแตกต่างระหว่างคลังข้อมูลและดาต้ามาร์ท

SECTION 6

วิธีการสร้างคลังข้อมูล

หลังจากที่เราทราบถึงความแตกต่างของ **“Data warehouse”** และ **“Data mart”** แล้ว เราควรที่จะพิจารณาถึงวิธีการสร้างทั้งสองระบบ ที่จะสามารถจำแนกวิธีได้สร้างได้ 2 วิธี ดังนี้



1 Top-down

เป็นวิธีการสร้างคลังข้อมูลที่ถูกเสนอโดย “Bill Inmon” ซึ่งได้นิยามคลังข้อมูลที่ถูกสร้างโดยวิธีการนี้ว่าเป็น“ศูนย์กลางคลังข้อมูลสำหรับองค์กร” ที่มีการจัดเก็บข้อมูลที่มีความละเอียดค่อนข้างสูงและมีการทำนอร์มอลไลซ์กับข้อมูล โดยคลังข้อมูลที่สร้างขึ้นจากวิธีการ top-down จะอยู่ที่ศูนย์กลางที่มีการสร้าง“logical framework”สำหรับสนับสนุนการทำธุรกิจอย่างชาญฉลาดขององค์กร

ข้อดี

ข้อดีของวิธีการ *Top-down* จะประกอบไปด้วย

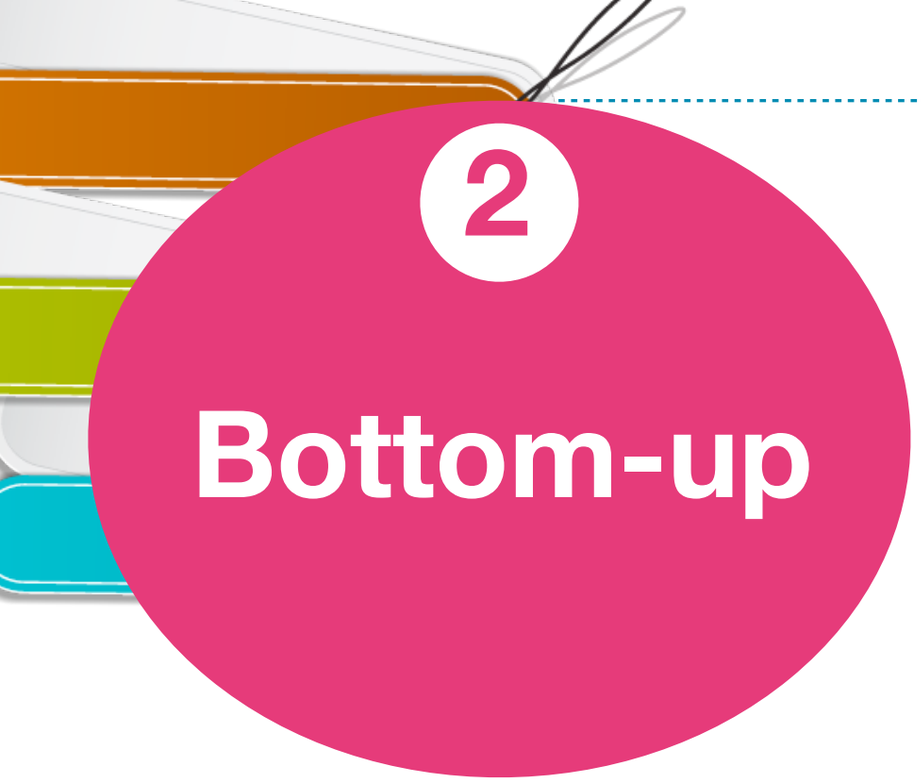
- สามารถมองข้อมูลได้ทั่วทั้งองค์กร
- สถาปัตยกรรมเป็นเนื้อเดียวกันและไม่ได้เป็นแบบการรวมกันของหลายๆ ดาต้ามาร์ท
- ทำการเก็บข้อมูลไว้ที่เดียว
- มีการควบคุมและกำหนดกฎเกณฑ์ต่างๆ จากศูนย์กลาง

ข้อเสีย

ข้อเสียของวิธีการ *Top-down* จะประกอบไปด้วย

- ใช้เวลาในการสร้างค่อนข้างนาน
- มีความเสี่ยงที่เกิดความล้มเหลวค่อนข้างสูง
- ต้องการผู้สร้างที่มีความรู้ ความสามารถสูง ในการที่จะสร้างการเชื่อมโยงฟังก์ชันการทำงานที่มีการข้ามสายงาน
- เสียค่าใช้จ่ายค่อนข้างมาก

จากข้างต้นเราจะสามารถเห็นภาพกว้างๆ ของวิธีการสร้างคลังข้อมูลแบบ Top-down ที่จะทำให้เราได้ข้อมูลเป็นกลุ่มก้อนเป็นชิ้นเดียวกัน แต่อย่างไรก็ดีการสร้างวิธีนี้จะใช้เวลานาน เนื่องจากต้องทำความเข้าใจเกี่ยวกับการดำเนินธุรกิจทั้งองค์กร มีความเสี่ยงสูงที่จะเกิดความล้มเหลวหากทีมผู้สร้างคลังข้อมูลยังไม่มีประสบการณ์เกี่ยวกับการสร้างคลังข้อมูลเพียงพอ และมีความต้องการผู้สร้างที่มีความเชี่ยวชาญทั้งในเชิงเทคนิคและเชิงธุรกิจสูง



2 Bottom-up

เป็นวิธีการสร้างคลังข้อมูลที่ถูกเสนอโดย Ralph Kimball ซึ่งได้นิยามคลังข้อมูลที่ถูกสร้างโดยวิธีการนี้ว่าเป็น “กลุ่มของดาต้ามาร์ทที่สอดคล้องกัน” โดยปัจจัยหลักของวิธีการนี้จะอยู่ที่ความสอดคล้องกันของดาต้ามาร์ทที่สร้างขึ้นเพื่อสนับสนุนการวิเคราะห์ข้อมูลในแต่ละส่วนงาน การสร้างคลังข้อมูลด้วยวิธีการ bottom-up จะเริ่มจากการสร้างดาต้ามาร์ทของแต่ละส่วนงานเพื่อให้ผู้ใช้หรือพนักงานในแต่ละแผนกสามารถทำการวิเคราะห์ข้อมูลและสร้างรายงานในแง่มุมต่างๆ ที่สอดคล้องกับการดำเนินธุรกิจของแผนกนั้นๆ ได้ ดาต้ามาร์ทแต่ละส่วนจะมีการเก็บข้อมูลที่มีความละเอียดสูง รวมถึงข้อมูลที่เป็นผลสรุปตามความต้องการในการวิเคราะห์ข้อมูลของผู้ใช้ เมื่อทำการสร้างดาต้ามาร์ทหลายๆ ดาต้ามาร์ทแล้ว จากนั้นจะทำการเชื่อมโยงทุกๆ ดาต้ามาร์ทเข้าด้วยกันโดยคำนึงถึงความสอดคล้องของข้อมูลเป็นหลัก

ข้อดี

ข้อดีของวิธีการ *Bottom-up* จะประกอบไปด้วย

- สามารถการดำเนินการได้เร็วขึ้นและง่ายขึ้น โดยการพิจารณาข้อมูลแต่ละส่วนงาน
- มีความเสี่ยงของความล้มเหลวน้อย
- สามารถกำหนดให้ส่วนงาน/แผนกที่มีความสำคัญค่อนข้างมากสามารถทำการสร้างดาต้ามาร์ทได้ก่อน
- ช่วยให้ผู้สร้างสามารถทำการเรียนรู้ทีละส่วนงาน

ข้อเสีย

ข้อเสียของวิธีการ *Bottom-up* จะประกอบไปด้วย

- แต่ละดาต้ามาร์ทจะมีข้อมูลของตัวเองเท่านั้นซึ่งเป็นข้อมูลที่ค่อนข้างแคบ
- อาจทำให้เกิดความซ้ำซ้อนของข้อมูล อาจต้องทำการเก็บข้อมูลหนึ่งๆ ไว้ในทุกดาต้ามาร์ท
- การสร้างแต่ละดาต้ามาร์ทก่อน แล้วค่อยรวมกันอาจทำให้ข้อมูลไม่สอดคล้องกัน
- อาจทำให้อินเทอร์เฟซ (interface) ของทุกๆ ดาต้ามาร์ทนั้นไม่สอดคล้องกัน และเมื่อทำการรวมดาต้ามาร์ทเข้าด้วยกันอาจไม่สามารถจัดการให้อินเทอร์เฟซต่างๆ ให้เป็นมาตราฐานเดียวกันได้

วิธีการสร้างคลังข้อมูลแบบ Bottom-up จะทำการสร้างดาต้ามาร์ททีละส่วน โดยที่เราสามารถกำหนดความสำคัญของส่วนงานเพื่อกำหนดว่าดาต้ามาร์ทใดควรจะทำการสร้างก่อน แต่ข้อเสียที่ชัดเจนที่สุดคือ ข้อมูลจะกระจายออกเป็นส่วนๆ ไม่รวมเป็นกลุ่มก้อน ซึ่งเมื่อดาต้ามาร์ทเป็นอิสระต่อกันจะทำให้ไม่สามารถมองเห็นความต้องการทั้งหมดทั่วทั้งองค์กรได้

SECTION 7

แนวปฏิบัติสำหรับการสร้างคลังข้อมูล

จากวิธีการสร้างคลังข้อมูลทั้งสองวิธีข้างต้น เราจะทราบว่าแต่ละวิธีจะมีข้อดีและข้อเสียที่แตกต่างกัน ดังนั้นในการสร้างคลังข้อมูลเราจะต้องพิจารณาว่าสิ่งที่เราต้องการคืออะไร? องค์กรของเราต้องการที่จะมองหาผลลัพธ์ระยะยาวหรือดาต้ามาร์ทที่มีข้อมูลไม่มากที่สามารถสร้างได้รวดเร็ว ณ ปัจจุบัน? องค์กรของเราต้องการใช้เวลาในการสร้างระยะสั้นหรือไม่? หรือองค์กรของเรากำลังมองหาวิธีการที่สามารถใช้งานได้จริง ซึ่งจากวิธีการสร้างแบบ Top-down และ Bottom-up ต่างก็มีข้อเสีย ดังนั้นวิธีการที่น่าจะดีที่สุดก็คือ“การรวมกันของทั้งสองวิธี” ที่จะทำให้เราสามารถเห็นภาพกว้างๆ ของทั้งองค์กร โดยในการวางแผนการสร้างสำหรับทั้งองค์กรจะใช้วิธีการแบบ top-down แต่เราจะประยุกต์ใช้วิธีการแบบ bottom-up ในการสร้างแต่ละดาต้ามาร์ทที่มีความเหมาะสม โดยทำการกำหนดลำดับความสำคัญของแต่ละส่วนงานที่จะทำการสร้างคลังข้อมูล

ซึ่งขั้นตอนวิธีในการสร้างคลังข้อมูลที่นำวิธีการ top-down และ bottom-up มารวมกันจะประกอบไปด้วย 4 ขั้นตอน ดังนี้

1. วางแผนและกำหนดความต้องการของทั้งองค์กรทุกๆ ระดับ ตั้งแต่ความต้องการของนักวิเคราะห์ข้อมูล ผู้จัดการ ผู้บริหาร กรรมการผู้จัดการ และ อื่นๆ

2. สร้างสถาปัตยกรรมโดยรวมสำหรับคลังข้อมูลที่สมบูรณ์

3. กำหนดวิธีการที่จะทำให้ข้อมูลที่ถูกเก็บอยู่ในคลังข้อมูลและแต่ละดาต้ามาร์ทมีความสอดคล้องกัน และเป็นมาตราฐานเดียวกัน

4. ทำการสร้างคลังข้อมูลทีละส่วนงานเรียงต่อกัน โดยทำการสร้างดาต้ามาร์ทหนึ่งๆ ณ ช่วงเวลาหนึ่งๆ เท่านั้น

จากแนวปฏิบัติทั้ง 4 ข้อจะทำให้เราสามารถมองได้ว่าคลังข้อมูลคือ “กลุ่มของดาต้ามาร์ทที่สอดคล้องกัน” โดยที่แต่ละดาต้ามาร์ทจะให้บริการการดำเนินธุรกิจของแต่ส่วนงานและภาพรวมของทั้งองค์กรด้วย ซึ่งเราสามารถเรียกกลุ่มของดาต้ามาร์ทที่ทำการสร้างขึ้นว่าเป็น “Enterprise data warehouse” โดยแนวปฏิบัติในการสร้างคลังข้อมูลนั้นจะมีหัวใจหลักอยู่ที่การวางแผนและกำหนดความต้องการของทั้งองค์กร ซึ่งเราจะต้องเก็บรวมรวมความต้องการให้ได้ทั้งหมดจากผู้ใช้ทุกระดับ และการทำให้ข้อมูลมีความสอดคล้องกัน ทั้งในส่วนของ ชนิดของข้อมูล ความยาวของฟิลด์ ความถูกต้องแม่นยำ และความหมายที่สื่อไปในทางเดียวกัน เพื่อหลีกเลี่ยงความแตกต่างของข้อมูล หรือความไม่สอดคล้องของข้อมูลระหว่างดาต้ามาร์ทที่อาจจะทำให้เกิดความไม่ถูกต้องของข้อมูลเกิดขึ้น

SECTION 8

คำถามท้ายบท

1. นิยามหรือคำจำกัดความของ “คลังข้อมูล” คืออะไร
2. จงอธิบายและแจกแจงคุณลักษณะเด่นของคลังข้อมูล
3. เพราะเหตุใดทุกๆ คลังข้อมูลจะต้องมีแกนเวลาเข้ามาเกี่ยวข้อง
4. จงอธิบายหรือให้คำจัดความของ “ข้อมูลที่มีความละเอียดหลายระดับ”
5. คลังข้อมูลประกอบไปด้วยส่วนประกอบอะไรบ้าง
6. คลังข้อมูลและดาต้ามาร์ท เหมือนหรือแตกต่างกันอย่างไร
7. เพราะเหตุใดคลังข้อมูลจึงจำเป็นต้องมีหลายแหล่งข้อมูล และแหล่งข้อมูลของคลังข้อมูลมีกี่ชนิดอะไรบ้าง
8. เพราะเหตุใดเราจึงต้องเรียกใช้“data staging” หรือ “staging area”
9. การสร้างคลังข้อมูลแบบ top-down และ bottom-up มีความแตกต่างกันอย่างไร
10. ในการสร้างคลังข้อมูล เราควรเลือกใช้วิธีการสร้างแบบใด เพราะเหตุใด

CHAPTER 3

สถาปัตยกรรมของคลังข้อมูล

SECTION 1

แผนการสอนประจำบท

วัตถุประสงค์ของบทเรียน

- เข้าใจเกี่ยวกับสถาปัตยกรรมคลังข้อมูล
- ศึกษาเกี่ยวกับสถาปัตยกรรมคลังข้อมูลกับส่วนประกอบต่าง ๆ
- ศึกษาเกี่ยวกับความสามารถของสถาปัตยกรรมที่สามารถสนับสนุนการเคลื่อนที่ของข้อมูลได้
- ศึกษาเกี่ยวกับฟังก์ชันและบริการของส่วนประกอบต่างๆ ของสถาปัตยกรรม
- ศึกษาเกี่ยวกับสถาปัตยกรรมชนิดต่างๆ ของคลังข้อมูล

เนื้อหาของบทเรียน

เนื้อหาในบทนี้จะประกอบด้วย ปัจจัยที่เกี่ยวข้องกับการออกแบบสถาปัตยกรรมของคลังข้อมูล กรอบสถาปัตยกรรมของคลังข้อมูล สถาปัตยกรรมของคลังข้อมูลเชิงเทคนิค สถาปัตยกรรมชนิดต่างๆ ที่มักถูกประยุกต์ในการสร้างคลังข้อมูล

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารประกอบการสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

กิจกรรมการเรียน-การสอน

- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารประกอบการสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

การวัดและประเมินผล

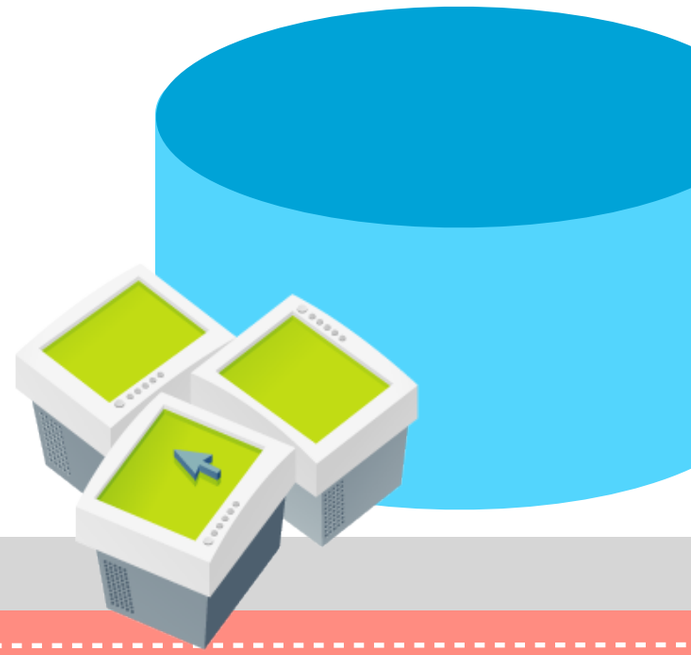
- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

SECTION 2

บทนำ

จากทั้งสองบทก่อนหน้าจะทำให้เราทราบถึงความต้องการและความจำเป็นในการสร้างคลังข้อมูล ทราบว่าคลังข้อมูลคืออะไร คุณลักษณะเด่นของคลังข้อมูล ส่วนประกอบของคลังข้อมูล ชนิดของคลังข้อมูล และท้ายสุดคือวิธีการในการสร้างคลังข้อมูล แต่อย่างไรก็ดี เรายังไม่ทราบว่าเราจะสามารถทำการเก็บข้อมูลไว้ในคลังข้อมูลได้อย่างไร? เพื่อที่จะตอบคำถามดังกล่าว

ในบทนี้เราทำการศึกษาเกี่ยวกับสถาปัตยกรรมของคลังข้อมูลที่เป็นโครงสร้างที่จะนำทุกๆ ส่วนประกอบของคลังข้อมูลมารวมกันเพื่อสนับสนุนการทำงานของคลังข้อมูลให้มีประสิทธิภาพสูงสุด ในการออกแบบสถาปัตยกรรมของคลังข้อมูลเราจะต้องทำการพิจารณาปัจจัยต่างๆ ที่เกี่ยวข้องกับการจัดเตรียมข้อมูล การจัดเก็บข้อมูลลงในฐานข้อมูล และการส่งผ่านข้อมูลไปยังผู้ใช้งาน โดยสถาปัตยกรรมของคลังข้อมูลจะประกอบไปด้วย กฎ ขั้นตอน/กระบวนการ และฟังก์ชันการทำงานต่างๆ ที่จะทำให้คลังข้อมูลสามารถขับเคลื่อนไปข้างหน้า และสามารถตอบสนองความต้องการธุรกิจได้



การสร้างหรือออกแบบสถาปัตยกรรมสำหรับคลังข้อมูลจะมีวัตถุประสงค์เพื่อจัดเตรียม/สร้างขอบข่ายการทำงานสำหรับการพัฒนาและการประยุกต์ใช้คลังข้อมูล โดยที่การออกแบบสถาปัตยกรรมคลังข้อมูลจะเป็นตัวกำหนดมาตราฐาน ตัวชี้วัด การออกแบบการทำงานให้สอดคล้องกัน และยังรวมถึงเทคนิคต่างๆ ซึ่งโดยทั่วๆ ไปแล้ว การออกแบบสถาปัตยกรรมคลังข้อมูลจะขึ้นอยู่กับฟังก์ชันการทำงานหลักของคลังข้อมูล ที่ประกอบไปด้วย 3 ฟังก์ชัน คือ

1 **Data acquisition**

การได้มาซึ่งข้อมูล (Data acquisition) จะเป็นการทำงานที่อยู่บนพื้นฐานของกฎต่างๆ ที่จะทำให้ได้ข้อมูลที่เป็นอินพุตจากแหล่งข้อมูล

2 **Data storage**

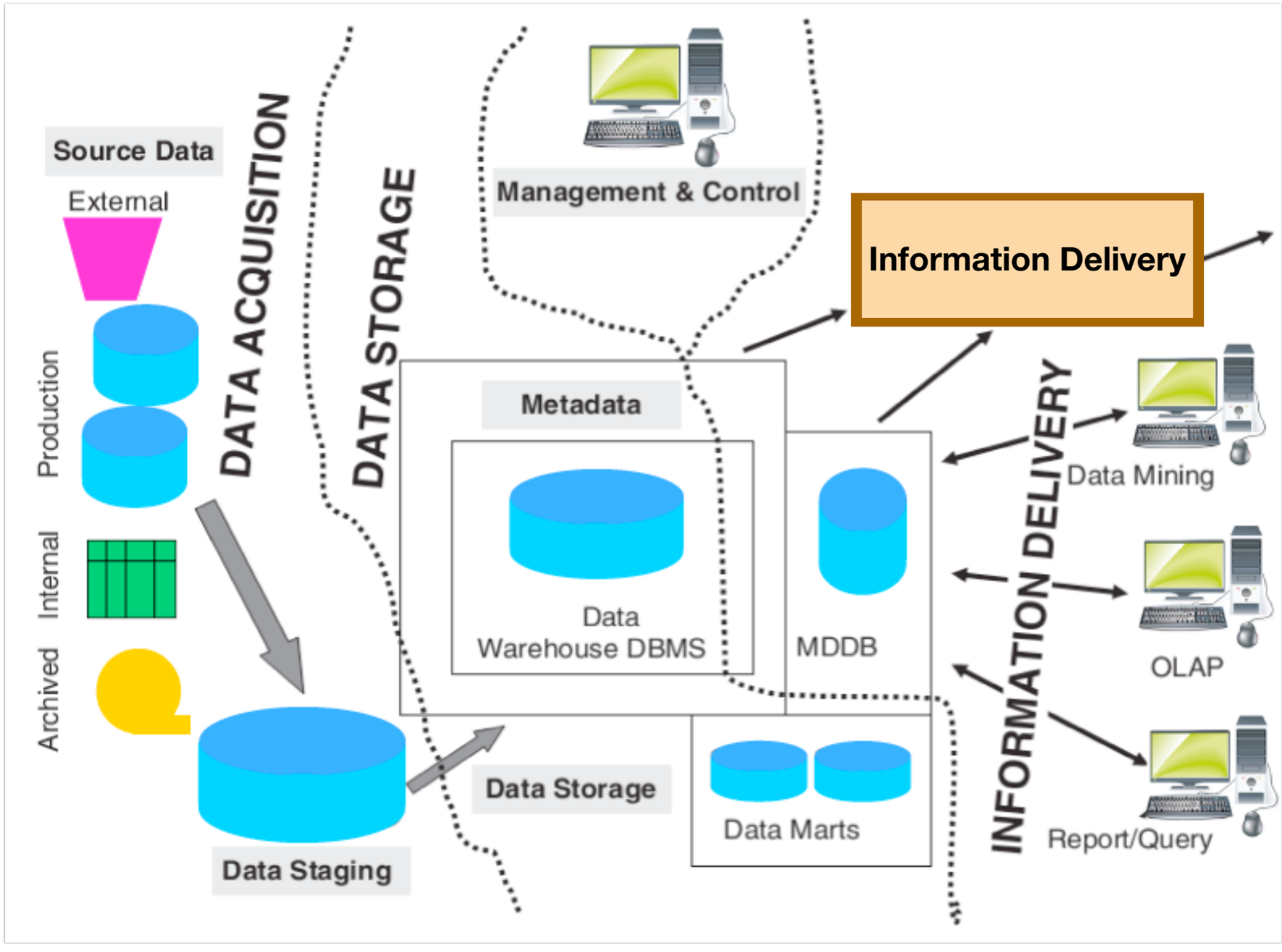
การจัดเก็บข้อมูลในคลังข้อมูล (Data storage) จะเป็นขั้นตอนหรือกระบวนการสำหรับจัดเก็บข้อมูลที่ได้มาจากขั้นตอนแรกไว้ในฐานข้อมูลหรือพื้นที่สำหรับจัดเก็บข้อมูล

3 **Information delivery**

การเข้าถึงข้อมูลหรือส่งผ่านข้อมูลไปยังผู้ใช้ (Information delivery) จะเป็นการเข้าถึง/เรียกดู/ใช้งานข้อมูลที่ถูกจัดเก็บอยู่ในคลังข้อมูล และเป็นส่วนที่ใช้ติดต่อสื่อสารกับผู้ใช้งาน

จากฟังก์ชันการทำงานหลักทั้ง 3 ฟังก์ชันข้างต้น เราจะสามารถแบ่งส่วนประกอบของคลังข้อมูลให้สอดคล้องกับทั้ง 3 ฟังก์ชันได้ ดังแสดงในรูปที่ 3-1 ที่จะแสดงถึงส่วนประกอบทั้งหมดที่สอดคล้องหรือสนับสนุนการทำงานของแต่ละฟังก์ชัน

แต่อย่างไรก็ตาม จะมีส่วนประกอบหนึ่งที่ไม่ขึ้นกับฟังก์ชันใดเลย นั่นคือ ส่วนของการจัดการและการควบคุมกระบวนการทำงานที่จะเป็นส่วนพิเศษที่มีหน้าที่ในการจัดการและควบคุมกระบวนการทำงานทั้งหมดของคลังข้อมูล



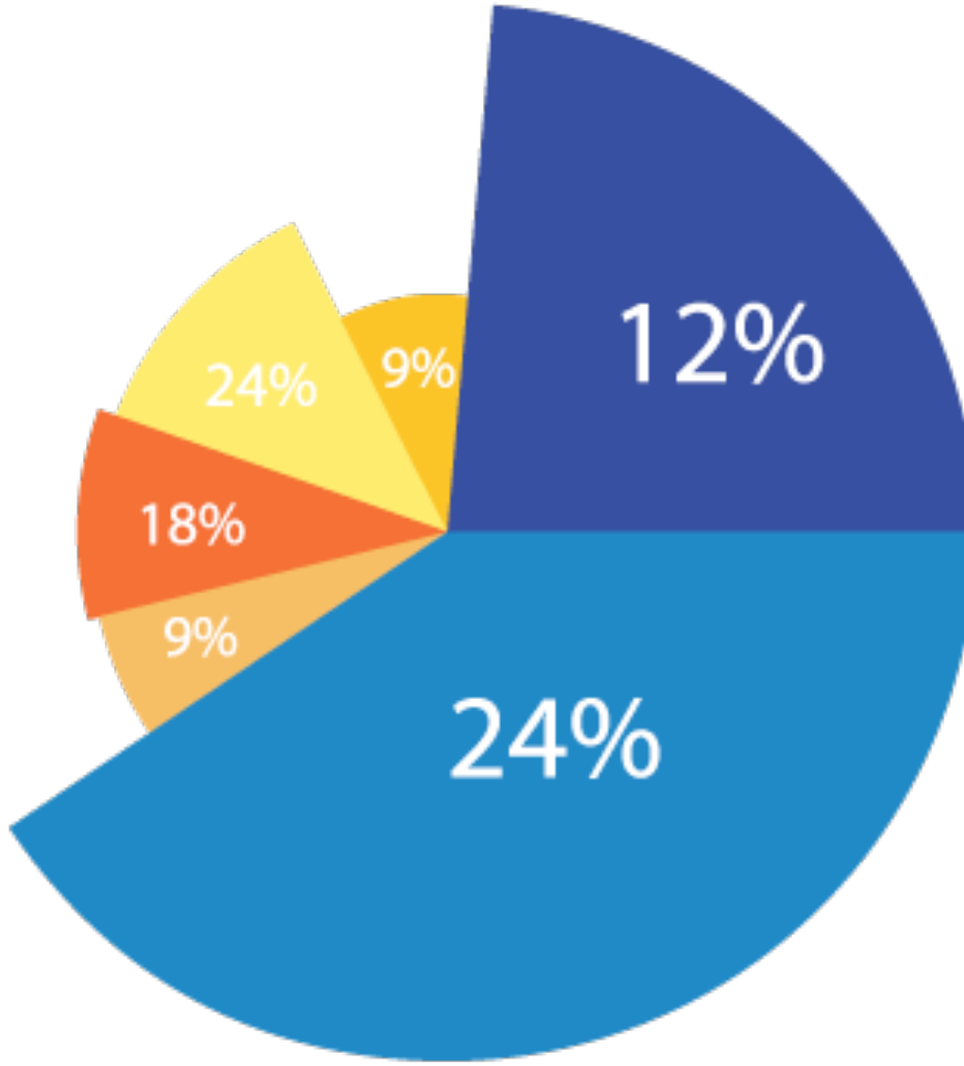
รูปที่ 3-1 สถาปัตยกรรมของข้อมูลที่สอดคล้องกับฟังก์ชันการทำงานหลักของคลังข้อมูล

SECTION 3

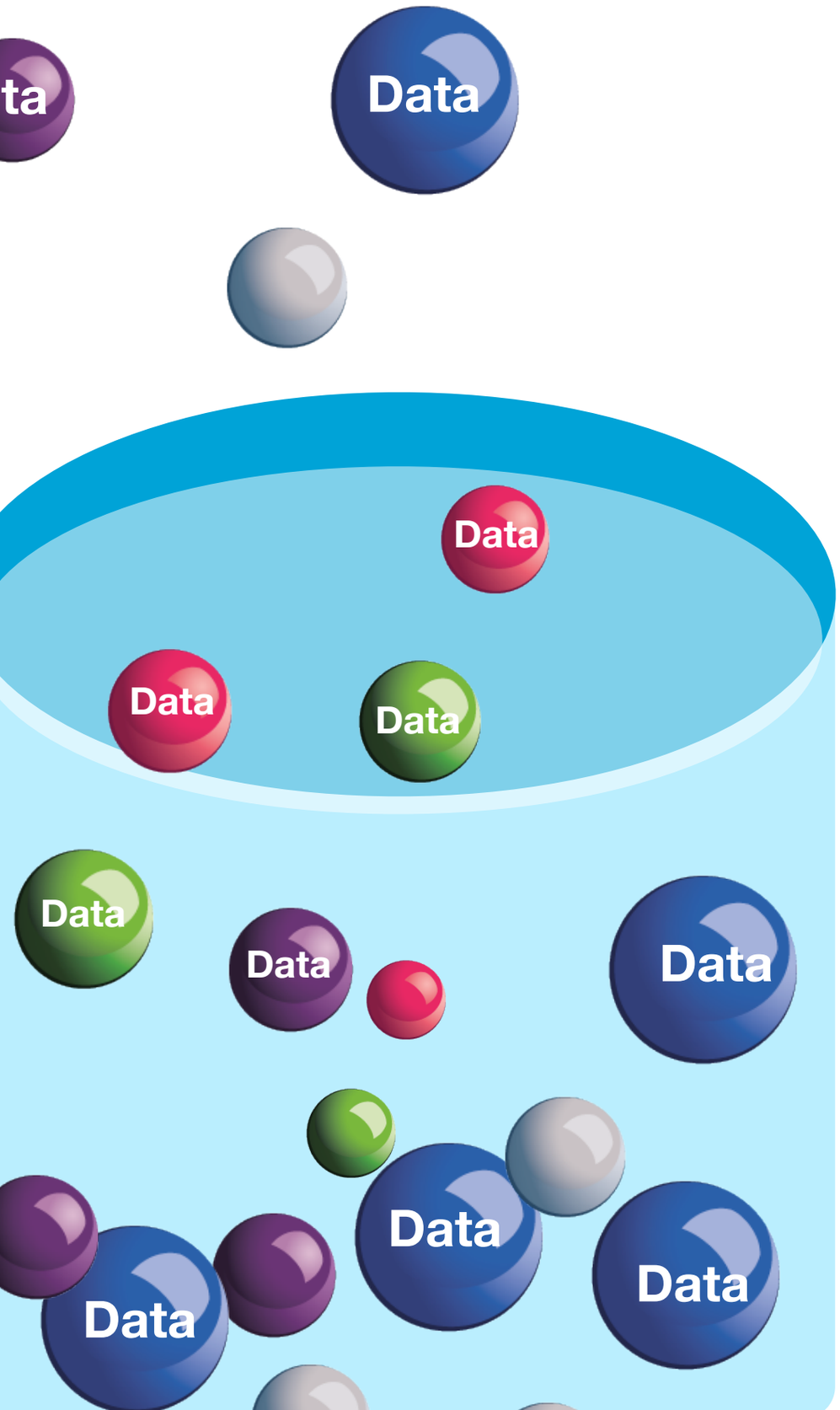
ป้จจัยที่เกี่ยวข้องกับการออกแบบสถาปัตยกรรมของคลังข้อมูล

สถาปัตยกรรมของคลังข้อมูลจะถูกออกแบบมาเพื่อสนับสนุนการทำงานต่างๆ ของคลังข้อมูลเพื่อทำการสร้างหรือจัดเตรียมข้อมูลเชิงกลยุทธ์ที่สามารถตอบสนองความต้องการของผู้ใช้งาน ซึ่งโดยส่วนใหญ่ของผู้ใช้มักจะสนใจหรือมีความต้องการที่จะได้ผลลัพธ์ในหลายๆ แง่มุม เช่น ผู้ใช้อาจจะต้องการข้อมูลยอดขายสินค้าของในปีหนึ่งๆ โดยทำการแบ่งยอดขายในปีต้องการออกตามไตรมาส ตามรายการสินค้าที่มีการวางจำหน่าย และตามพื้นที่ที่มีการวางจำหน่ายสินค้า เป็นต้น

ซึ่งจากความต้องการดังกล่าว เราจะต้องออกแบบสถาปัตยกรรมเพื่อสนับสนุนการประมวลผลข้อมูลในแต่มุมต่างๆ ที่มีปริมาณมหาศาล โดยในการออกแบบสถาปัตยกรรมเราจะต้องพิจารณาปัจจัยต่างๆ อาทิเช่น การพิจารณาถึงจำนวนและขอบเขตของแหล่งข้อมูลที่จะใช้เป็นอินพุตของคลังข้อมูล ซึ่งในการพิจารณาเราจะต้องเข้าใจถึงแพลตฟอร์มของแหล่งข้อมูลต่างๆ ลักษณะของข้อมูลต่างๆ ฟังก์ชันการเลือกหรือการสกัดข้อมูลจากแหล่งข้อมูลเหล่านั้น ขอบเขตของข้อมูล และอื่นๆ



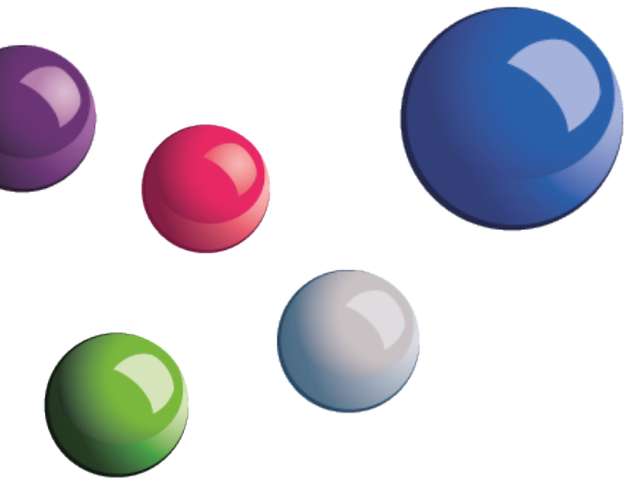
นอกจากนั้นเรายังต้องพิจารณาถึงการทำข้อมูลให้เป็นมาตราฐาน การรวบรวมข้อมูลเข้าด้วยกัน ความละเอียดของข้อมูล และปริมาณข้อมูลที่ต้องจัดเก็บในคลังข้อมูล เป็นต้น จากข้างต้นเราจะเห็นว่าปัจจัยเหล่านี้เป็นปัจจัยที่เกี่ยวข้องทั้งหมด โดยจะเกี่ยวข้องกับข้อมูล และฟังก์ชันการทำงานเสียเป็นส่วนใหญ่ แต่อย่างไรก็ดีในการทำงานของคลังข้อมูลจะมีอยู่ 3 ปัจจัยหลักที่เราต้องทำการพิจารณา ดังนี้



จากคุณสมบัติของข้อมูลในคลังข้อมูลที่อธิบายในบทที่ 2 จะทำให้เราทราบว่า เมื่อเราทำการจัดเก็บข้อมูลลงในฐานข้อมูลของคลังข้อมูลแล้ว ผู้ใช้จะสามารถอ่านหรือเรียกดูข้อมูลเหล่านั้นได้เพียงอย่างเดียว ผู้ใช้จะไม่สามารถทำการเพิ่ม ลบ หรืออัพเดทข้อมูลเหล่านั้นได้ ซึ่งจากคุณสมบัติดังกล่าว เราจะสามารถเรียกข้อมูลในคลังข้อมูลว่าเป็น “***ข้อมูลที่สามารถอ่านได้อย่างเดียว (Read-only data)***”

ดังนั้นในการได้มาซึ่งข้อมูลดังกล่าวจะต้องมีฟังก์ชันต่างๆ มากมายถูกดำเนินการหรือประมวลผล เมื่อเราทำการออกแบบสถาปัตยกรรมของคลังข้อมูล เราจะต้องรวมฟังก์ชันเหล่านั้นไว้ในคลังข้อมูลด้วย นอกจากข้อมูลที่สามารถอ่านได้อย่างเดียวแล้ว คลังข้อมูลยังมีอีกคุณสมบัติ นั่นคือ การรวบรวมข้อมูล (Integrated data) โดยการรวบรวมข้อมูลจะเกิดขึ้นก็ต่อเมื่อข้อมูลของคลังข้อมูลนั้นมาจากหลายแหล่งข้อมูล ซึ่ง ณ ปัจจุบัน องค์กรต่าง ๆ จะมีการใช้ระบบการดำเนินงานที่หลากหลายเพื่อสนับสนุนการทำงานหลาย ๆ งานที่แตกต่างกัน

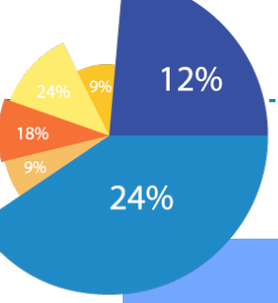
ดังนั้นเมื่อเราทำการเลือกหรือสกัดข้อมูลจากแหล่งข้อมูลแล้ว เราจะต้องทำการรวบรวมข้อมูลต่างๆ เข้าด้วยกัน ถ้าข้อมูลที่ได้มาจากระบบที่มีแพลตฟอร์มที่แตกต่างกันจะทำให้เราต้องทำการประมวลผลข้อมูลเหล่านั้น เช่น การทำความสะอาดข้อมูล การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล และ อื่นๆ ซึ่งจากคุณสมบัติและการทำงานดังกล่าว สถาปัตยกรรมของคลังข้อมูลควรที่จะต้องรวมฟังก์ชันการทำงานเหล่านี้เข้าไปด้วย



นอกจากคุณสมบัติทั้งสองข้างต้น การออกแบบสถาปัตยกรรมของคลังข้อมูลจะต้องสนับสนุนการจัดเก็บข้อมูลตามหัวข้อทางธุรกิจต่างๆ (Business subject) และการจัดเก็บข้อมูลที่เป็นปัจจุบันและข้อมูลย้อนหลัง ซึ่งโดยปกติของคลังข้อมูลจะมีการเก็บข้อมูลย้อนหลังตั้งแต่ 5-15 ปี เป็นต้น ซึ่งจากการเก็บข้อมูลอย่างต่อเนื่องตั้งแต่อดีตจนถึงปัจจุบันจะทำให้คลังข้อมูลมีปริมาณข้อมูลค่อนข้างมาก ดังนั้นการออกแบบสถาปัตยกรรมของคลังข้อมูลจะต้องสนับสนุนปริมาณข้อมูลที่ค่อนข้างมากและการเก็บข้อมูลย้อนหลังอีกด้วย

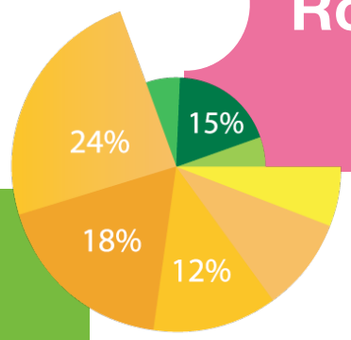
การวิเคราะห์ความซับซ้อน และเวลาที่ใช้ใน การประมวลผลที่รวดเร็ว

นอกเหนือจากการพิจารณาเนื้อหาของคลังข้อมูลตามคุณสมบัติต่างๆ ของข้อมูลในคลังข้อมูลแล้ว ในการออกแบบสถาปัตยกรรมของคลังข้อมูล เราจะต้องทำการออกแบบให้สถาปัตยกรรมนั้น ๆ สนับสนุนการวิเคราะห์ข้อมูลที่ซับซ้อนเพื่อที่จะสามารสร้างข้อมูลเชิงกลยุทธ์ให้แก่ผู้ใช้ได้ นอกจากจะมีการวิเคราะห์ที่ซับซ้อนแล้ว สถาปัตยกรรมที่สร้างขึ้นจะต้องมีการประมวลผลที่รวดเร็ว โดยการใช้งานครั้งหนึ่งๆ ของผู้ใช้อาจจะใช้งานในลักษณะที่เป็นการโต้ตอบ ซึ่งคลังข้อมูลจะต้องมีความสามารถค้นหาข้อมูลได้อย่างรวดเร็ว เช่น ผู้บริหารทางการตลาดต้องการที่จะทราบเหตุผลอย่างรวดเร็วว่า เพราะเหตุใดยอดขายสินค้าในเขตภาคกลางลดลงอย่างต่อเนื่องเป็นเวลา 3 สัปดาห์ ซึ่งจากความต้องการดังกล่าว คลังข้อมูลอาจจะสามารถสร้างหรือจัดเตรียมข้อมูลยอดขายสินค้าโดยแสดงการเปรียบเทียบยอดขายของภาคกลางโดยเทียบกับภาคอื่นๆ หรือข้อมูลการเปรียบเทียบยอดขายสินค้าในช่วง 3 สัปดาห์ที่ผ่านมากับยอดขายสินค้าในช่วงเวลาอื่นๆ เพื่อช่วยให้ผู้บริหารสามารถทราบถึงสาเหตุที่แท้จริงได้ เมื่อผู้บริหารทราบเหตุผลแล้ว ผู้บริหารอาจจะทำการตัดสินใจดำเนินการบางอย่างเพื่อทำให้ยอดขายเพิ่มขึ้น เป็นต้น



Drill down

Roll up



Slice

Dice

นอกเหนือจากความต้องการข้างต้น สถาปัตยกรรมของคลังข้อมูลควรที่จะสนับสนุนการวิเคราะห์ในแง่มุมต่างๆ ทั้งการวิเคราะห์ข้อมูลแบบเจาะลึก (Drill down) การวิเคราะห์ข้อมูลที่เป็นผลสรุป (Roll up) การวิเคราะห์ข้อมูลเพียงบางส่วนที่ต้องการ (Slice) และการปรับเปลี่ยนมุมมองของข้อมูล (Dice)

ซึ่งจากความต้องการในการวิเคราะห์ข้อมูลที่ค่อนข้างจะหลากหลาย ผู้ใช้อาจต้องการรูปแบบผลลัพธ์ที่มีความหลากหลายด้วยเช่นกัน เช่น การแสดงผลลัพธ์ในรูปแบบตาราง กราฟ หรือชาร์ตในลักษณะต่างๆ

ความยืดหยุ่นและความเปลี่ยนแปลงได้ง่าย

ในการออกแบบและพัฒนาคลังข้อมูลเราอาจไม่ทราบถึงความต้องการทั้งหมด หรืออาจเก็บรวมรวมความต้องการจากผู้ใช้ได้ในปริมาณที่จำกัด โดยที่เมื่อเราเริ่มทำการสร้างคลังข้อมูลและมีการเริ่มใช้คลังข้อมูล ผู้ใช้อาจจะสามารถบอกความต้องการที่มากขึ้นหรือสามารถบอกได้ถึงแนวทางการทำธุรกิจที่เปลี่ยนแปลงไปตามกาลเวลาได้มากขึ้น ซึ่งจากกรณีดังกล่าวทำให้การออกแบบสถาปัตยกรรมของคลังข้อมูลควรที่จะต้องมีความยืดหยุ่นกับการเปลี่ยนแปลงที่จะเกิดขึ้นได้ ซึ่งถ้าสถาปัตยกรรมของคลังข้อมูลสามารถรองรับความเปลี่ยนแปลงดังกล่าวได้ จะทำให้คลังข้อมูลนั้นสามารถทำงานและถูกใช้งานได้อย่างเต็มที่ และมีคุณค่ากับองค์กรเป็นอย่างมาก แม้ว่าความต้องการหรือแนวทางการดำเนินธุรกิจจะเปลี่ยนแปลงไปตามกาลเวลาก็ตาม

SECTION 4

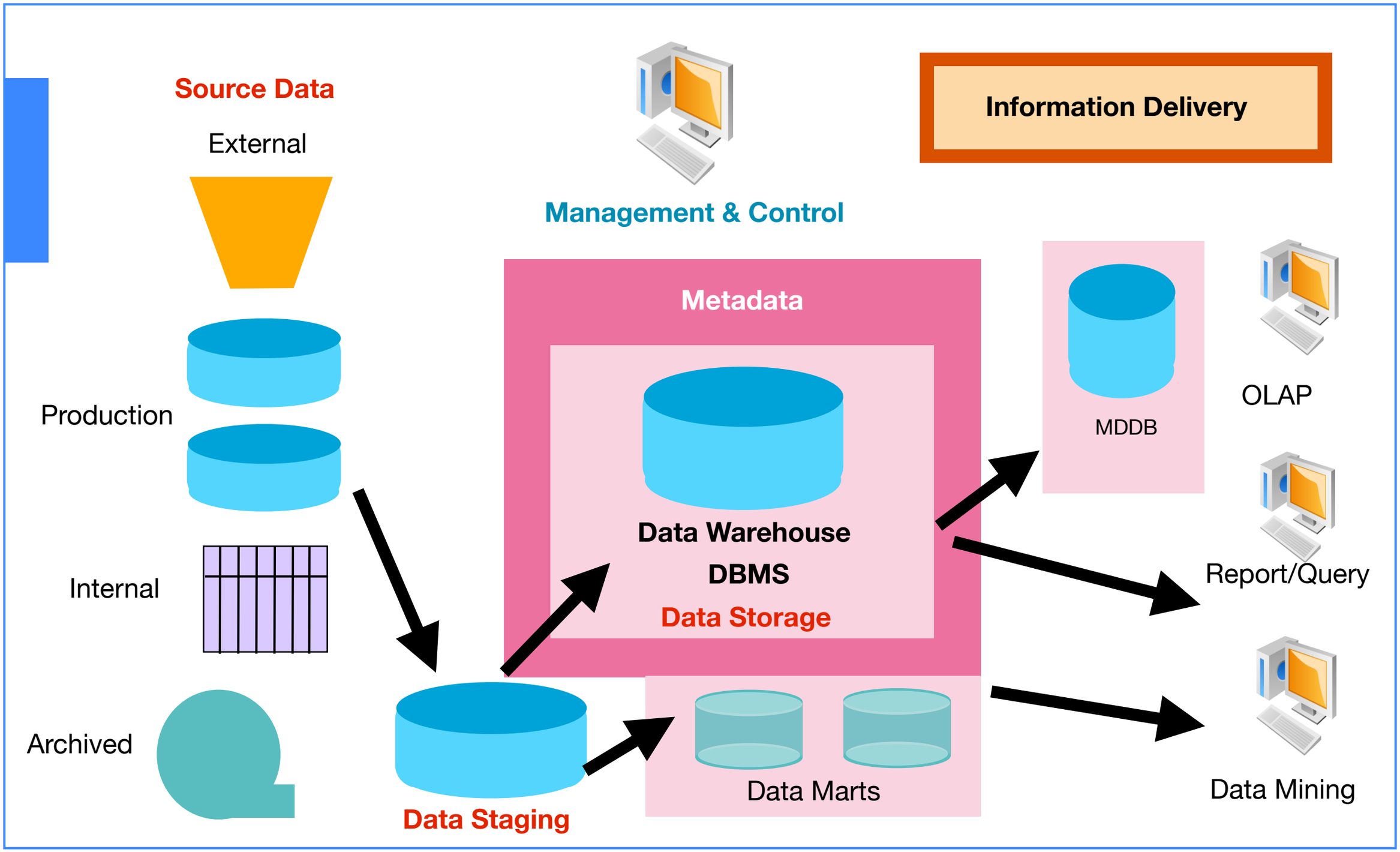
กรอบสถาปัตยกรรมของคลังข้อมูล

จากบทที่ 2 และเนื้อหาในส่วนที่แล้ว เราสามารถแบ่งกลุ่มของส่วนประกอบของคลังข้อมูลได้เป็น 3 กลุ่มตามฟังก์ชันการทำงานหลักของคลังข้อมูลซึ่งก็คือ การได้มาซึ่งข้อมูล การจัดเก็บข้อมูล และการส่งผ่านข้อมูล ในส่วนนี้เราจะทำการศึกษาถึงกรอบของสถาปัตยกรรมของคลังข้อมูลที่สามารถตอบสนองวัตถุประสงค์ต่างๆ ดังนี้

สถาปัตยกรรมที่สนับสนุนการเคลื่อนที่ของข้อมูล

จากการทำงานทั้ง 3 ฟังก์ชันหลักของคลังข้อมูลเราจะทราบว่าข้อมูลในคลังข้อมูลจะมีการเคลื่อนที่อยู่หลายครั้งด้วยกัน โดยการเคลื่อนที่ของข้อมูลนั้นจะเริ่มจากข้อมูลที่อยู่ในแหล่งข้อมูลที่เป็นระบบการดำเนินงาน หรือแหล่งข้อมูลภายนอก และยังรวมไปถึงข้อมูลที่เป็นสเปรดชีทและฐานข้อมูลย่อยของแต่ละแผนก ซึ่งจากข้อมูลดังกล่าว ระบบคลังข้อมูลจะทำการสกัด/เลือกข้อมูลเพียงบางส่วนจากข้อมูลที่อยู่ตามที่ต่างๆ ข้างต้น แล้วทำการจัดเก็บข้อมูลเหล่านั้นไว้ในพื้นที่พักข้อมูลเพื่อรอที่จะทำการประมวลผลเบื้องต้นกับข้อมูลเหล่านั้น แล้วจึงค่อยทำการถ่ายโอนข้อมูลเข้าสู่คลังข้อมูลและส่งต่อไปยังผู้ใช้งาน (ดังแสดงในรูปที่ 3-2)

ซึ่งการเคลื่อนที่ของข้อมูลจะต้องเกี่ยวข้องกับส่วนประกอบของคลังข้อมูลหลายส่วนด้วยกัน เช่น แหล่งข้อมูล พื้นที่พักข้อมูล พื้นที่สำหรับจัดเก็บข้อมูล และระบบการส่งผ่านข้อมูล ดังนั้นเพื่อให้ข้อมูลสามารถเคลื่อนที่ได้อย่างสะดวกและราบรื่น เราควรที่จะต้องทำการออกแบบสถาปัตยกรรมของคลังข้อมูลให้สอดคล้องการเคลื่อนที่ของข้อมูล โดยทำการเชื่อมโยงส่วนประกอบต่างๆ ที่เกี่ยวเนื่องกับการเคลื่อนที่ของข้อมูลเข้าด้วยกัน



รูปที่ 3-2 สถาปัตยกรรมที่สนับสนุนการเคลื่อนที่ของข้อมูล

การจัดการและการควบคุมขั้นตอนการทำงานต่างๆ

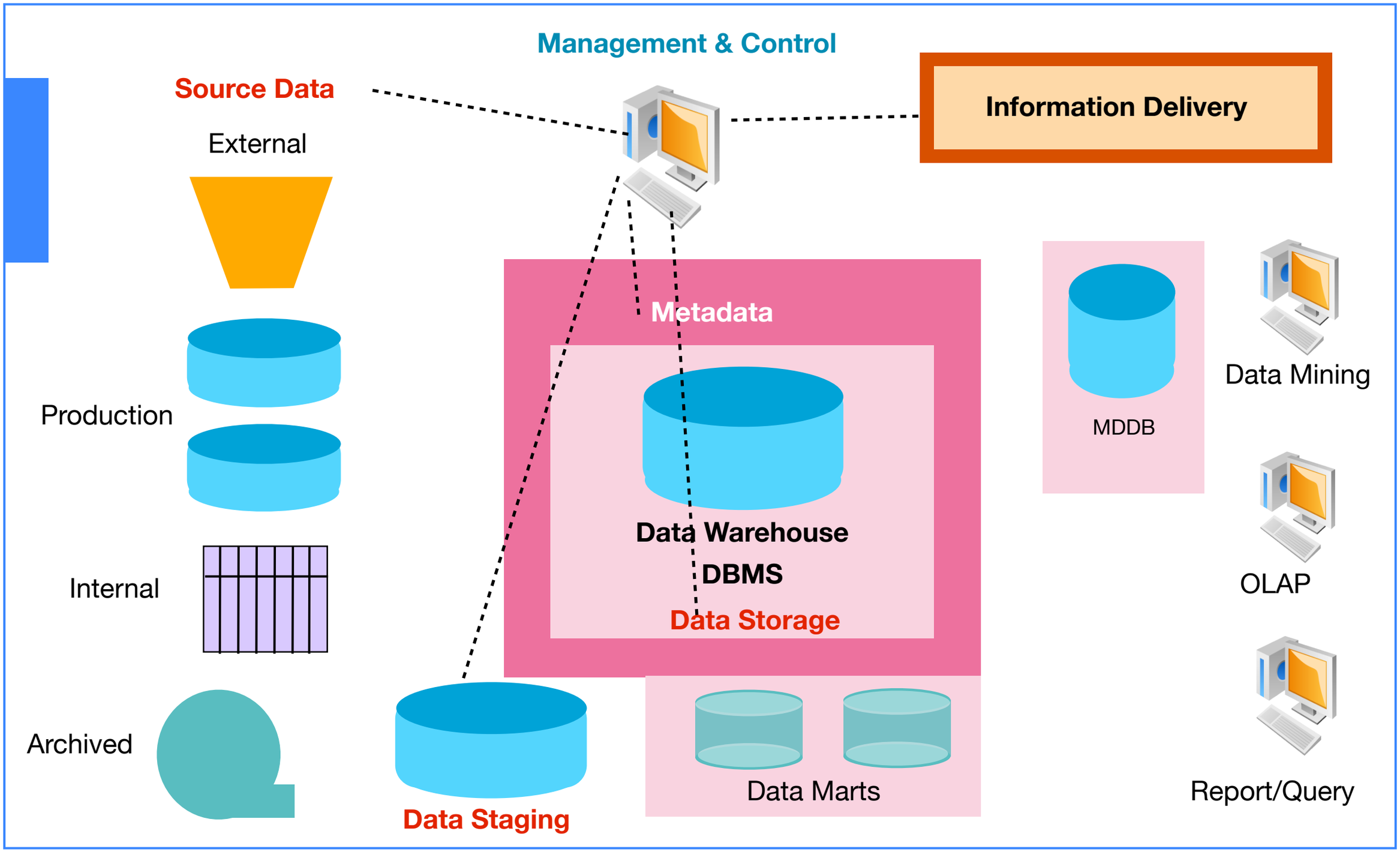
ในสถาปัตยกรรมของคลังข้อมูลจะมีส่วนประกอบหนึ่งที่มีความสำคัญค่อนข้างมากและมีความสำคัญไม่น้อยไปกว่าส่วนประกอบอื่นๆ เลย นั่นคือ ส่วนงานการจัดการและการควบคุมขั้นตอนการทำงาน ซึ่งจะลักษณะเป็นเหมือนร่มที่ปกคลุมส่วนประกอบและฟังก์ชันการทำงานต่างๆ ดังแสดงในรูปที่ 3-3 โดยส่วนงานนี้ประกอบไปด้วยฟังก์ชันการทำงานหลัก 2 ฟังก์ชันด้วยกัน คือ

(1) การเฝ้าดูหรือเฝ้าตรวจสอบการทำงานทุกๆขั้นตอน

(2) การแก้ไขปัญหาเมื่อฟังก์ชันการทำงานหนึ่งๆ เกิดปัญหาหรือมีข้อผิดพลาดเกิดขึ้น

โดยในส่วนของการเฝ้าดูหรือเฝ้าตรวจสอบจะทำการตรวจสอบตั้งแต่ขั้นตอนการสกัดข้อมูลจากแหล่งข้อมูล ขั้นตอนการเคลื่อนย้ายข้อมูลเข้าสู่พื้นที่พักข้อมูล ขั้นตอนการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล และอื่นๆ ขั้นตอนการตรวจสอบเหล่านี้จะเป็นการทำให้แน่ใจว่าขั้นตอนการทำงานต่างๆ ข้างต้นนั้นสามารถทำงานได้อย่างถูกต้องและสามารถทำงานได้ตามเวลาที่กำหนด

นอกจากทำการเฝ้าตรวจสอบและควบคุมการทำงานต่างๆแล้ว ส่วนงานนี้ยังทำการจัดการเกี่ยวกับการสำรองและการกู้คืนข้อมูลเมื่อมีความผิดพลาดหรือล้มเหลวเกิดขึ้น และยังรวมไปถึงการเฝ้าดูการเจริญเติบโตของปริมาณข้อมูลในคลังข้อมูล และการเคลื่อนย้ายข้อมูลที่ค่อนข้างเก่ามากและไม่ถูกใช้งาน ออกจากคลังข้อมูลอีกด้วย

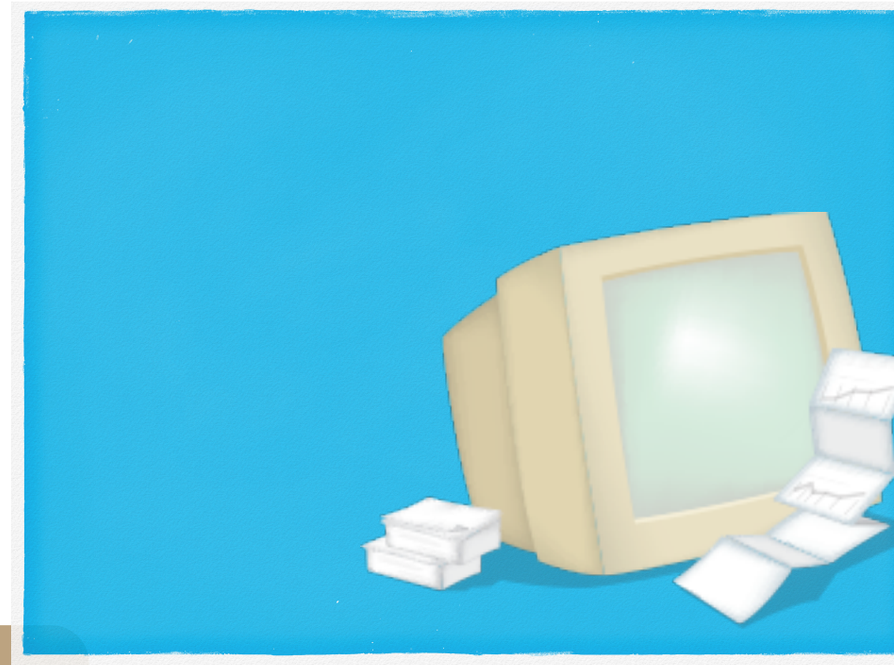


รูปที่ 3-3 สถาปัตยกรรมของคลังข้อมูลส่วนการจัดการและการควบคุมขั้นตอนการทำงานต่างๆ

SECTION 5

สถาปัตยกรรมของคลังข้อมูลเชิงเทคนิค

ในการที่จะออกแบบสถาปัตยกรรมของคลังข้อมูลนั้น เราควรจะออกแบบให้มีความสอดคล้องกับ 3 ฟังก์ชันการทำงานหลัก โดยสถาปัตยกรรมที่ถูกออกแบบจะต้องมีความยืดหยุ่น สามารถปรับเปลี่ยนได้ง่าย และยังต้องรองรับการวิเคราะห์ที่ซับซ้อน รวมถึงการทำงานได้อย่างรวดเร็ว นอกจากนั้นสถาปัตยกรรมของคลังข้อมูลจะต้องประกอบไปด้วยฟังก์ชันที่ใช้สำหรับกำหนดคิวรีหรือรายงานไว้ก่อนหน้าอีกด้วย (Function for predefining query and report) และยังคงต้องมีการให้บริการต่างๆ ซึ่งจากการทำงานที่เพิ่มเข้ามาจะเป็นส่วนช่วยให้คลังข้อมูลนั้นสามารถเป็นส่วนเติมเต็มเป้าหมายและความต้องการทางธุรกิจได้

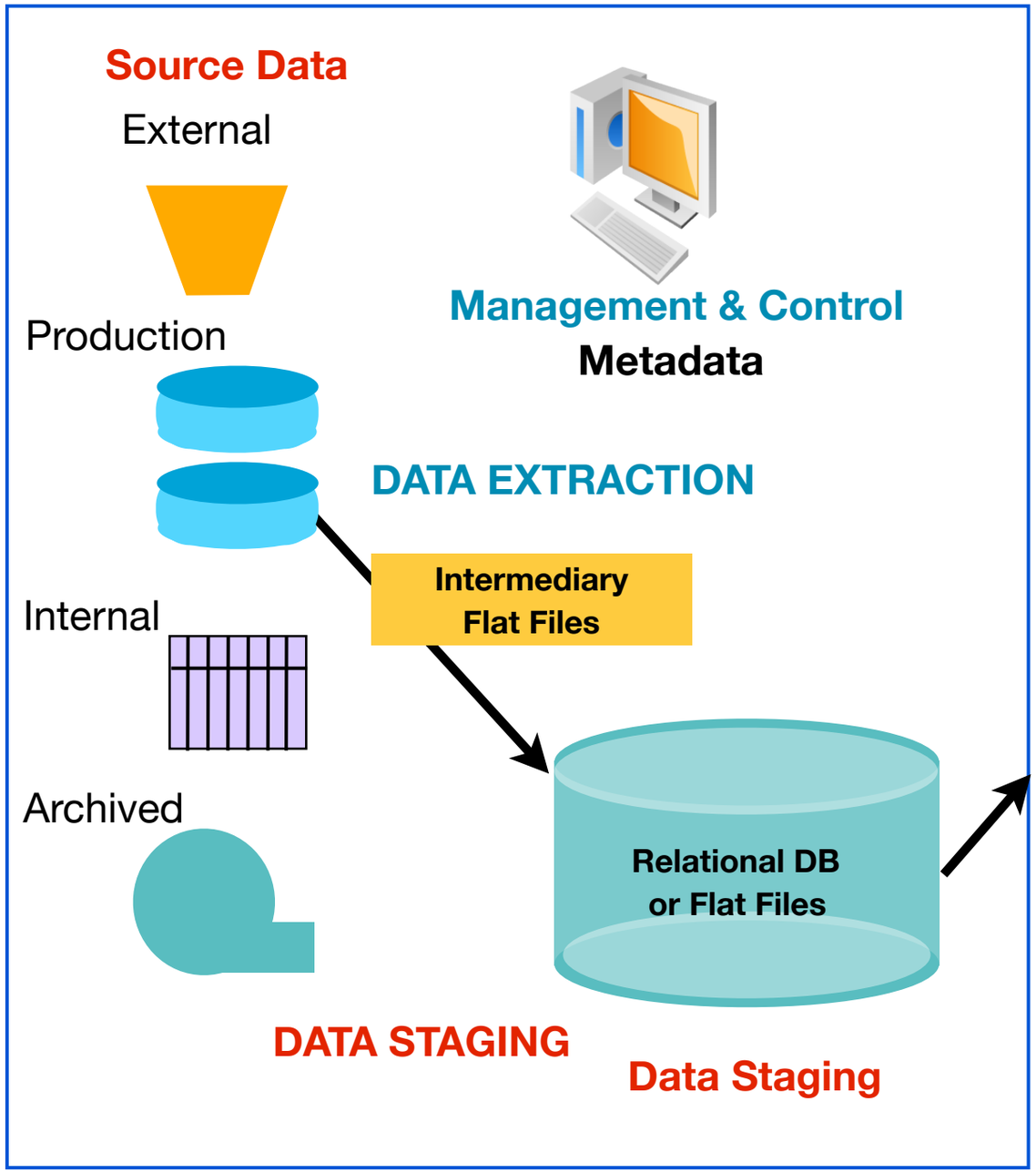


ในการออกแบบสถาปัตยกรรมคลังข้อมูลเชิงเทคนิคนั้น เราจะต้องออกแบบให้คลังข้อมูลนั้นมีฟังก์ชันและการให้บริการที่สมบูรณ์ โดยสถาปัตยกรรมของคลังข้อมูลนั้นจะไม่ได้เป็นกลุ่มของเครื่องมือที่จะดำเนินการหรือกระทำการประมวลผลต่างๆ แต่จะเป็นส่วนประกอบต่างๆ ที่ทำให้คลังข้อมูลสามารถทำงานได้อย่างสะดวกราบรื่น ซึ่งตัวเครื่องมือสำหรับดำเนินการต่างๆ จะเป็นสิ่งที่ถูกใช้ในการสร้างคลังข้อมูลตามสถาปัตยกรรมที่ได้ออกแบบไว้เท่านั้น ดังนั้นในการสร้างคลังข้อมูลเราควรที่จะต้องให้ความสำคัญกับสถาปัตยกรรมของคลังข้อมูลก่อนที่จะให้ความสำคัญกับเครื่องมือในการดำเนินการต่างๆ ดังนั้นในส่วนนี้จะทำการอธิบายถึงฟังก์ชัน การบริการ ขั้นตอนวิธีต่างที่เกี่ยวข้องกับส่วนประกอบต่างๆของคลังข้อมูล โดยฟังก์ชันเหล่านี้จะอิงกับฟังก์ชันการทำงานหลักของคลังข้อมูล ดังนี้

สถาปัตยกรรมสำหรับการได้มาซึ่งข้อมูล

ในส่วนของการได้มาซึ่งข้อมูลนั้นจะเริ่มการทำงานจากการสกัดข้อมูลหรือการเลือกข้อมูลที่ต้องการเพียงบางส่วนจากแหล่งข้อมูล จากนั้นทำการถ่ายโอนข้อมูลไปยังพื้นที่พักข้อมูล เพื่อทำการประมวลผลเบื้องต้น แล้วจึงทำการถ่ายโอนไปเก็บไว้ในคลังข้อมูลต่อไป ซึ่งจากการทำงานดังกล่าว ฟังก์ชันต่างๆ จะเกี่ยวเนื่องกับส่วนประกอบของคลังข้อมูล 2 ส่วนหลักๆด้วยกันคือ แหล่งข้อมูล/ระบบการดำเนินงาน และพื้นที่พักข้อมูล ดังแสดงในรูปที่ 3-4 ซึ่งจากทั้งสองส่วนประกอบนี้จะมีฟังก์ชันการทำงานต่างๆ ที่เกี่ยวเนื่องภายใต้ส่วนประกอบเหล่านี้มากมาย เช่น

- การเลือกแหล่งข้อมูล
- การสร้างแฟ้มข้อมูลสำหรับการสกัดข้อมูล
- การส่งผ่านแฟ้มข้อมูลที่ถูกสกัดจากหลายๆแพลตฟอร์ม
- การเปลี่ยนรูปแบบข้อมูลจากแหล่งข้อมูลภายนอก
- การเปลี่ยนรูปแบบข้อมูลจากข้อมูลที่เป็นสเปรดชีทหรือระบบฐานข้อมูลย่อยของแต่ละแผนก
- การทำให้ข้อมูลจากหลายๆแหล่งข้อมูลสอดคล้องกัน
- การเชื่อมโยงระหว่างข้อมูลจากพื้นที่พักข้อมูลกับฐานข้อมูลของคลังข้อมูล
- การทำความสะอาดข้อมูลและการรวมข้อมูลเข้าด้วยกัน
- การแปลงชนิดของข้อมูล
- การคำนวณค่าในแอทริบิวต่างๆ
- การแก้ปัญหาเกี่ยวกับการขาดหายไปของข้อมูล (Resolve missing values) และอื่นๆ

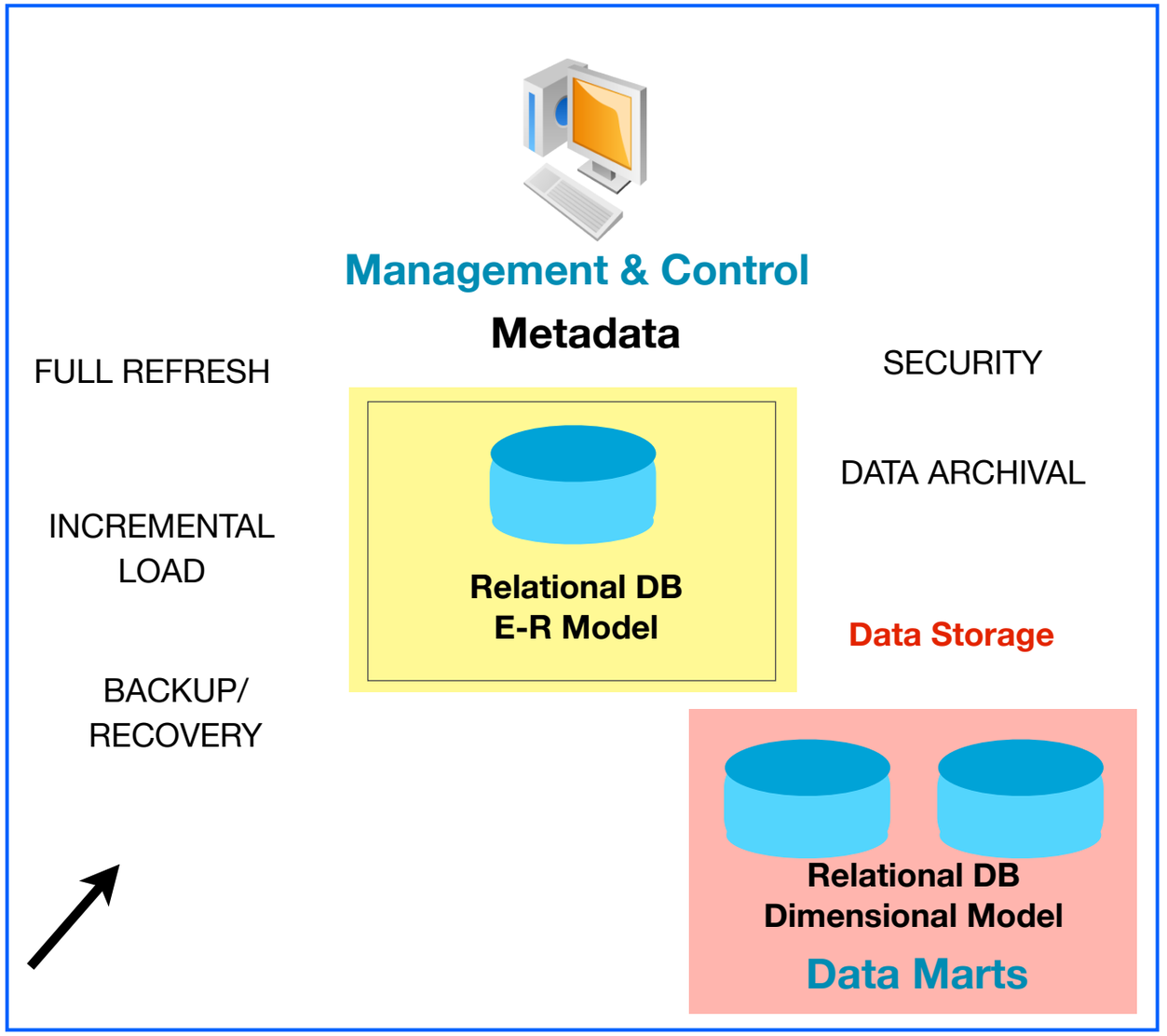


จากฟังก์ชันการทำงานข้างต้น เราพอที่จะมองเห็นภาพกว้างๆ ของการทำงานของการได้มาซึ่งข้อมูลคือ เริ่มจากการสกัดข้อมูลจากแหล่งข้อมูลต่างๆ โดยข้อมูลที่ถูกสกัดจะถูกเก็บไว้ในแฟ้มข้อมูลชั่วคราว (Temporary files) ถ้าแหล่งข้อมูลเหล่านั้นใช้แพลตฟอร์มที่แตกต่างกัน ข้อมูลหนึ่งที่สกัดได้จากแพลตฟอร์มหนึ่งๆ จะต้องถูกเขียนลงในแต่ละแฟ้มข้อมูล ซึ่งจากแพลตฟอร์มที่หลากหลายเราอาจจะได้แฟ้มข้อมูลสำหรับข้อมูลหนึ่งๆ หลายแฟ้มข้อมูลด้วยกัน เมื่อเราได้ข้อมูลจากการสกัดข้อมูลแล้ว เราอาจจะทำการรวมข้อมูลในเบื้องต้นก่อน แล้วจึงค่อยส่งข้อมูลเหล่านั้นไปยังพื้นที่พักข้อมูล ซึ่ง ณ ที่พักข้อมูลจะทำการรับข้อมูลที่ได้มาเก็บไว้ในแฟ้มข้อมูลหรือระบบจัดการฐานข้อมูลก็ได้ ซึ่ง ณ ปัจจุบันคลังข้อมูลสมัยใหม่มักจะใช้ระบบการจัดการฐานข้อมูลมาช่วยในการจัดเก็บข้อมูลลงในพื้นที่พักข้อมูล ซึ่งการใช้เครื่องมือดังกล่าวจะช่วยเพิ่มความสะดวก ประสิทธิภาพ และความถูกต้องการทำงานได้เป็นอย่างดี

รูปที่ 3-4 สถาปัตยกรรมของคลังข้อมูลที่สนับสนุนการได้มาซึ่งข้อมูล

สถาปัตยกรรมสำหรับการจัดเก็บข้อมูล

การจัดเก็บข้อมูลจะเริ่มจากการถ่ายโอนข้อมูลจากพื้นที่พักข้อมูล เพื่อนำไปเก็บไว้ในพื้นที่สำหรับจัดเก็บข้อมูลในคลังข้อมูล การดำเนินการดังกล่าวจะต้องเกี่ยวเนื่องกับส่วนประกอบของคลังข้อมูลสองส่วนด้วยกันคือ พื้นที่สำหรับจัดเก็บข้อมูล และพื้นที่สำหรับจัดเก็บเมตาดาต้า ดังแสดงในรูปที่ 3-5 ซึ่งจากรูปเราจะเห็นว่าไม่ว่าเราจะใช้วิธีการสร้างคลังข้อมูลแบบ top-down หรือ bottom-up ที่เป็นการสร้างคลังข้อมูลสำหรับทั้งองค์กรหรือการสร้างคลังข้อมูลย่อยเพื่อสนับสนุนการทำงานแต่ละแผนก ก็มักจะนิยมใช้ระบบการจัดการฐานข้อมูลมาช่วยในการจัดเก็บข้อมูล ที่ช่วยในเรื่องของความสะดวกและความถูกต้อง



รูปที่ 3-5 สถาปัตยกรรมของคลังข้อมูลที่สนับสนุนการจัดเก็บข้อมูล

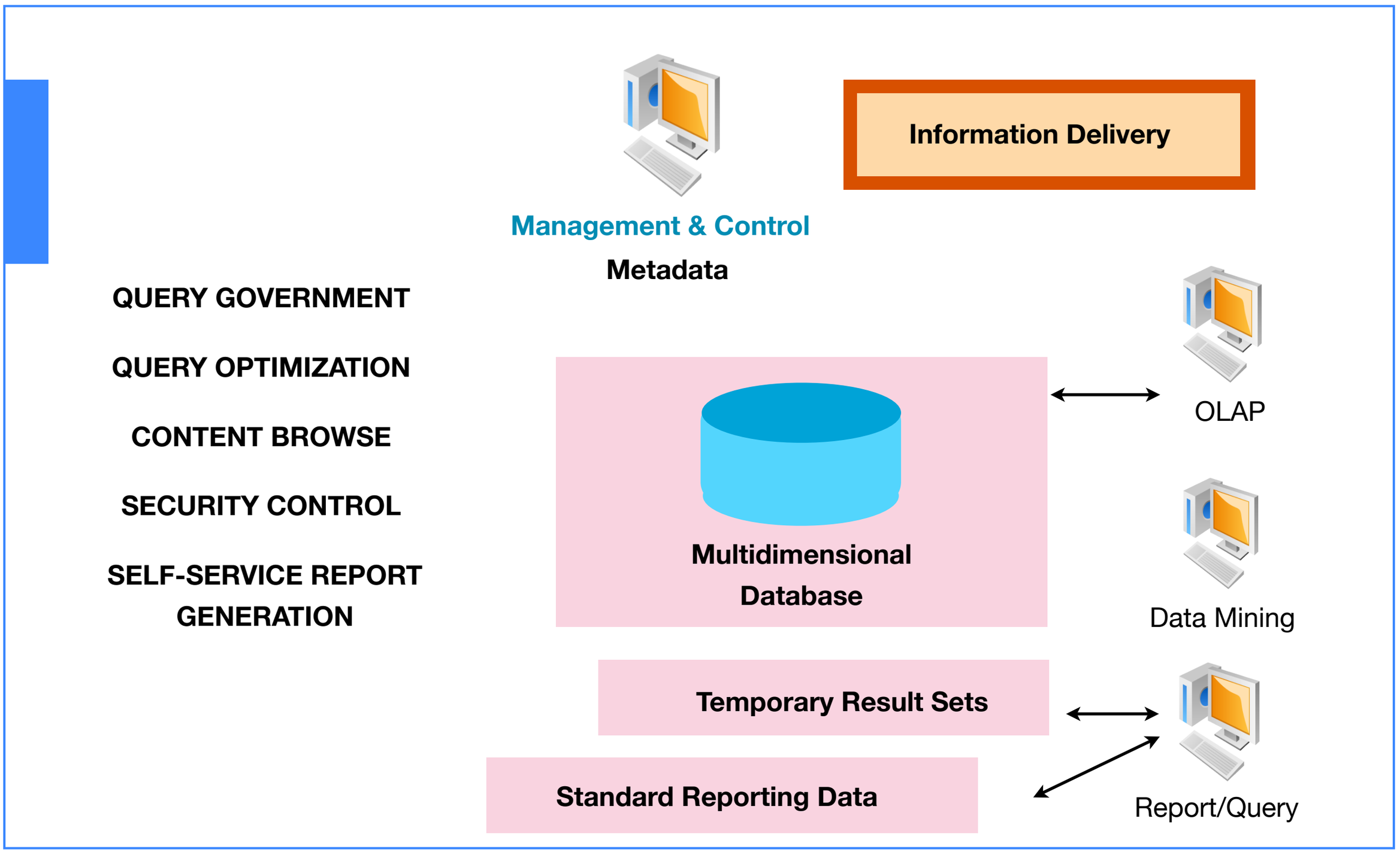
นอกเหนือจากการจัดเก็บข้อมูลลงในคลังข้อมูลแล้ว เราจะต้องทำการเก็บข้อมูลที่เป็นเมตาดาต้า ซึ่งเป็นข้อมูลของข้อมูลหรือดิกชันนารีข้อมูล ที่เป็นส่วนสำคัญมากต่อการสร้างและการใช้งานคลังข้อมูล (โดยรายละเอียดของเมตาดาต้าจะอธิบายในบทที่ 9 ต่อไป) โดยในการจัดเก็บข้อมูลและเมตาดาต้านั้นจะประกอบไปด้วยฟังก์ชันการทำงานเป็นจำนวนมาก ตัวอย่างเช่น

- การเพิ่มเติมข้อมูลเข้าสู่คลังข้อมูลตามช่วงเวลาที่กำหนด
- การถ่ายโอนข้อมูลไปยังตารางต่างๆ ทั้งในส่วนของข้อมูลที่เป็นรายละเอียดและข้อมูลที่เป็นผลสรุป
- การเฝ้าตรวจสอบขั้นตอนการถ่ายโอนข้อมูล
- ฟังก์ชันสำหรับการสำรองข้อมูลและกู้คืนข้อมูลที่ถูกจัดเก็บอยู่ในฐานข้อมูล
- ฟังก์ชันที่ดำเนินการรักษาความปลอดภัยกับข้อมูล
- การเฝ้าดูการทำงานและการปรับแก้การทำงานของฐานข้อมูล และ อื่นๆ

สถาปัตยกรรมสำหรับการส่งผ่านข้อมูล

ระบบหรือฟังก์ชันการส่งผ่านข้อมูลจะเป็นส่วนประกอบเพียงส่วนเดียวที่สามารถเชื่อมต่อกับผู้ใช้งานได้ ซึ่ง ในมุมมองของผู้ใช้จะมองว่าระบบการส่งผ่านข้อมูลก็คือคลังข้อมูล หรืออาจมองว่าสถาปัตยกรรมของข้อมูลนั้นจะเป็นเน้นที่ความคงทนต่อความผิดพลาดและความยืดหยุ่นในการส่งผ่านข้อมูล เป็นต้น การส่งผ่านข้อมูลที่ดีนั้นจะต้องทำ ให้ผู้ใช้สามารถเข้าถึงข้อมูลได้อย่างง่าย ไม่ว่าจะเป็นการเข้าถึงข้อมูล ในคลังข้อมูลของทั้งองค์กรหรือคลังข้อมูลย่อยของแต่ละแผนก นอกจากนั้นการส่งผ่านข้อมูลจะต้องมีความสามารถคืนค่าผลลัพธ์ที่ผู้ใช้ต้องการได้อย่างทันท่วงที ซึ่งการใช้งานคลังข้อมูล โดยส่วนใหญ่จะเป็นการเรียกดูข้อมูลที่เป็นแบบโต้ตอบ กล่าวคือ เมื่อผู้ใช้กรอกคิวรีแล้วได้ผลลัพธ์ออกมา ผู้ใช้ก็จะทำการถามซ้ำแต่จะถาม ในเชิงเจาะลึก หรือถาม ในเชิงหาผลสรุป จากพฤติกรรมการใช้งาน ดังกล่าว ระบบส่งผ่านข้อมูลที่สร้างขึ้นจะต้องสามารถตอบสนองความต้องการ ในลักษณะต่างๆ ได้

ณ ปัจจุบัน คลังข้อมูลสมัยใหม่มักนิยม ใช้เครื่องมือมาช่วย ในการส่งผ่านหรือวิเคราะห์ข้อมูล ซึ่งก็คือ OLAP (Online analytical processing) (จะอธิบาย ในบทที่ 11) ที่จะช่วยลดเวลา ในการประมวลผลและสามารถปรับเปลี่ยนมุมมองของการวิเคราะห์ข้อมูลได้หลายมุมมองด้วยกัน นอกจากนั้น ในกรณีที่ผู้ใช้ต้องการวิเคราะห์ข้อมูลที่มีความซับซ้อนค่อนข้างมาก เราสามารถเชื่อมต่อระบบการส่งผ่านข้อมูลเข้ากับการวิเคราะห์ข้อมูล ในรูปแบบอื่นๆ ได้ เช่น การทำเหมืองข้อมูล และเครื่องมือ ในการสร้างคิวรีหรือรายงาน เป็นต้น โดยตัวอย่างของระบบการส่งผ่านข้อมูลจะสามารถแสดงได้ดังรูปที่ 3-6

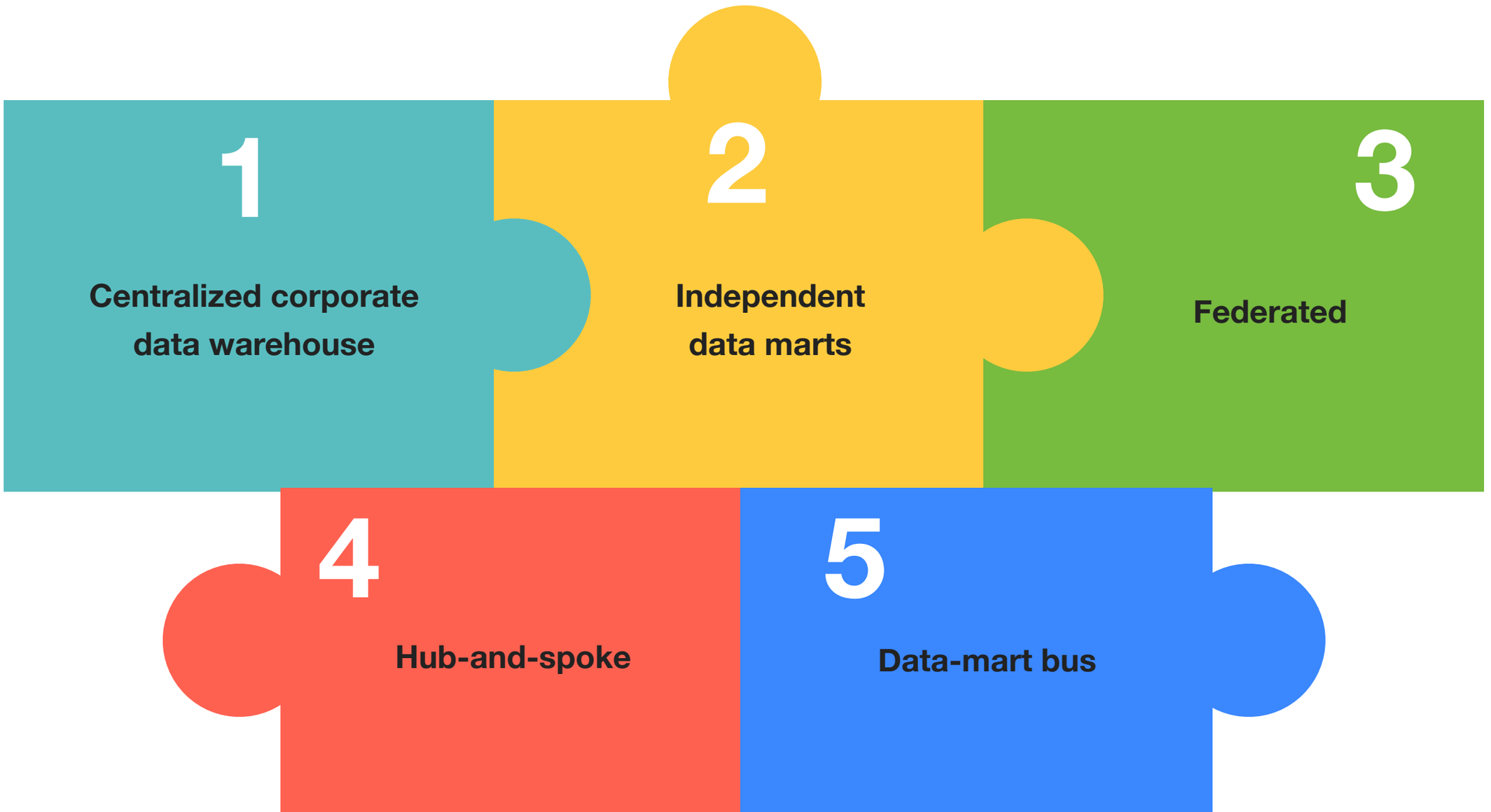


รูปที่ 3-6 สถาปัตยกรรมของคลังข้อมูลที่สนับสนุนการส่งผ่านข้อมูล

SECTION 6

สถาปัตยกรรมชนิดต่าง ๆ ของคลังข้อมูล

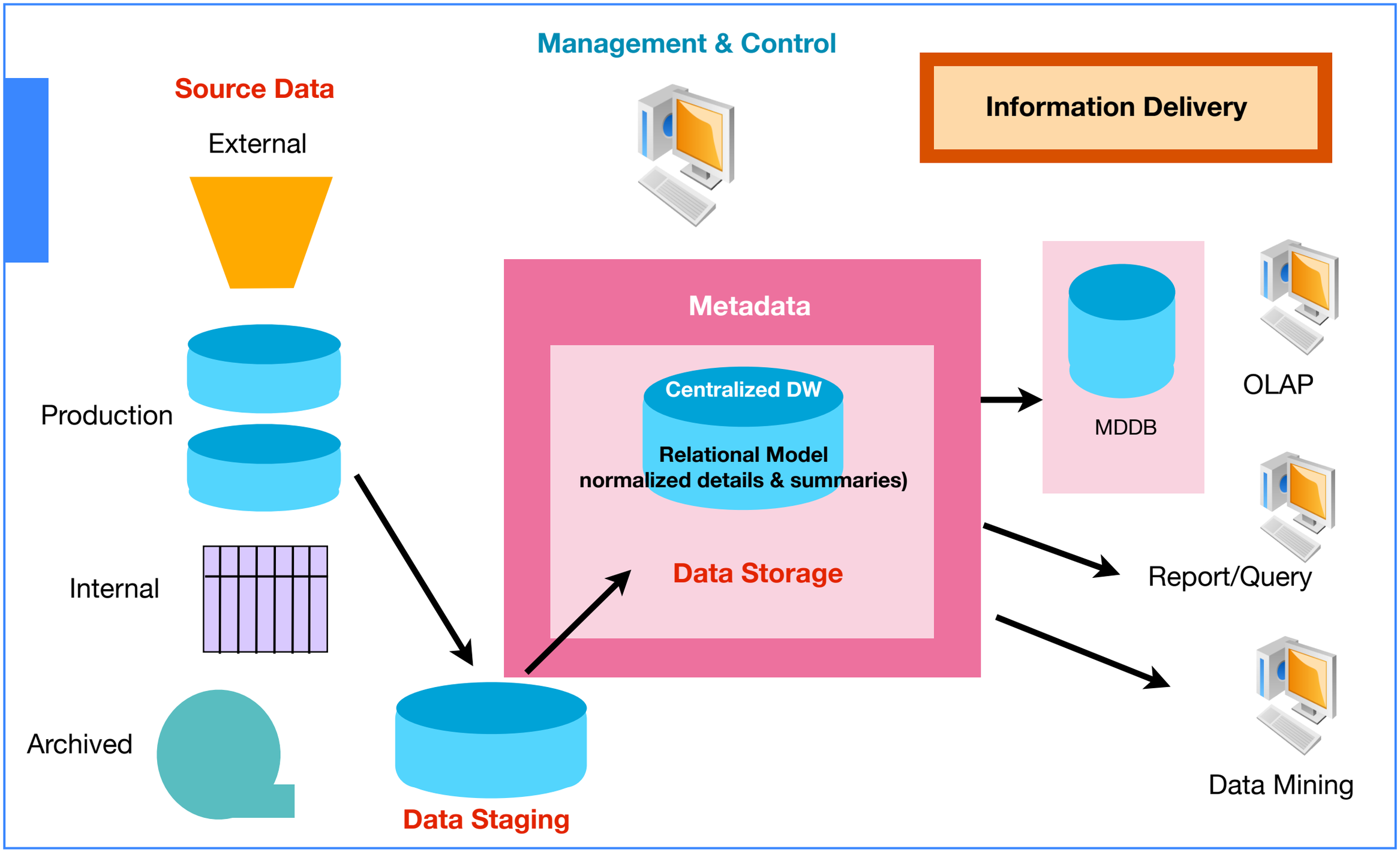
จากเนื้อหาในส่วนก่อนหน้าและบทที่ 2 เราจะทราบถึงส่วนประกอบของคลังข้อมูล รวมถึงวิธีในการสร้างคลังข้อมูลสำหรับทั้งองค์กร (Enterprise data warehouse) และคลังข้อมูลย่อยที่สนับสนุนการทำงานของแต่ละแผนก (Data marts) ซึ่งจากวิธีการสร้างทั้ง 2 วิธีจะทำให้คลังข้อมูลมีการจัดเก็บข้อมูลที่แตกต่างกัน ซึ่งจากการเก็บข้อมูลชนิดต่างๆ ทำให้เราสามารถจำแนกชนิดหรือประเภทของสถาปัตยกรรมของคลังข้อมูลออกเป็น 5 ประเภทที่มีความแตกต่างกันทั้งในด้านการรวบรวมข้อมูล การจัดเก็บข้อมูล และการเชื่อมโยงกันระหว่างคลังข้อมูลและดาต้ามาร์ท ซึ่งสามารถอธิบายได้ดังนี้



1

Centralized corporate data warehouse

Centralized corporate data warehouse จะเป็นสถาปัตยกรรมที่มีการจัดเก็บข้อมูลไว้ในฐานข้อมูลเพียงที่เดียว โดยจะไม่มีดาต้ามาร์ทเลยไม่ว่าจะเป็นดาต้ามาร์ทแบบอิสระต่อกันหรือแบบเชื่อมโยงกันก็ตาม ในการเข้าถึง/ส่งผ่านข้อมูลไปยังผู้ใช้จากสถาปัตยกรรมนี้สามารถทำได้โดยตรงกับคลังข้อมูลที่เป็นศูนย์กลาง เพื่อให้เข้าใจเกี่ยวกับสถาปัตยกรรมแบบ centralized corporate data warehouse มากขึ้น ลองพิจารณารูปที่ 3-7 ที่จะแสดงการเคลื่อนที่ของข้อมูล โดยเริ่มจากแหล่งข้อมูลไปยัง staging area จากนั้นจะถูกส่งต่อไปจัดเก็บไว้ในคลังข้อมูลที่เป็น central data warehouse แล้วจึงอนุญาตให้ผู้ใช้สามารถเรียกใช้ข้อมูลได้

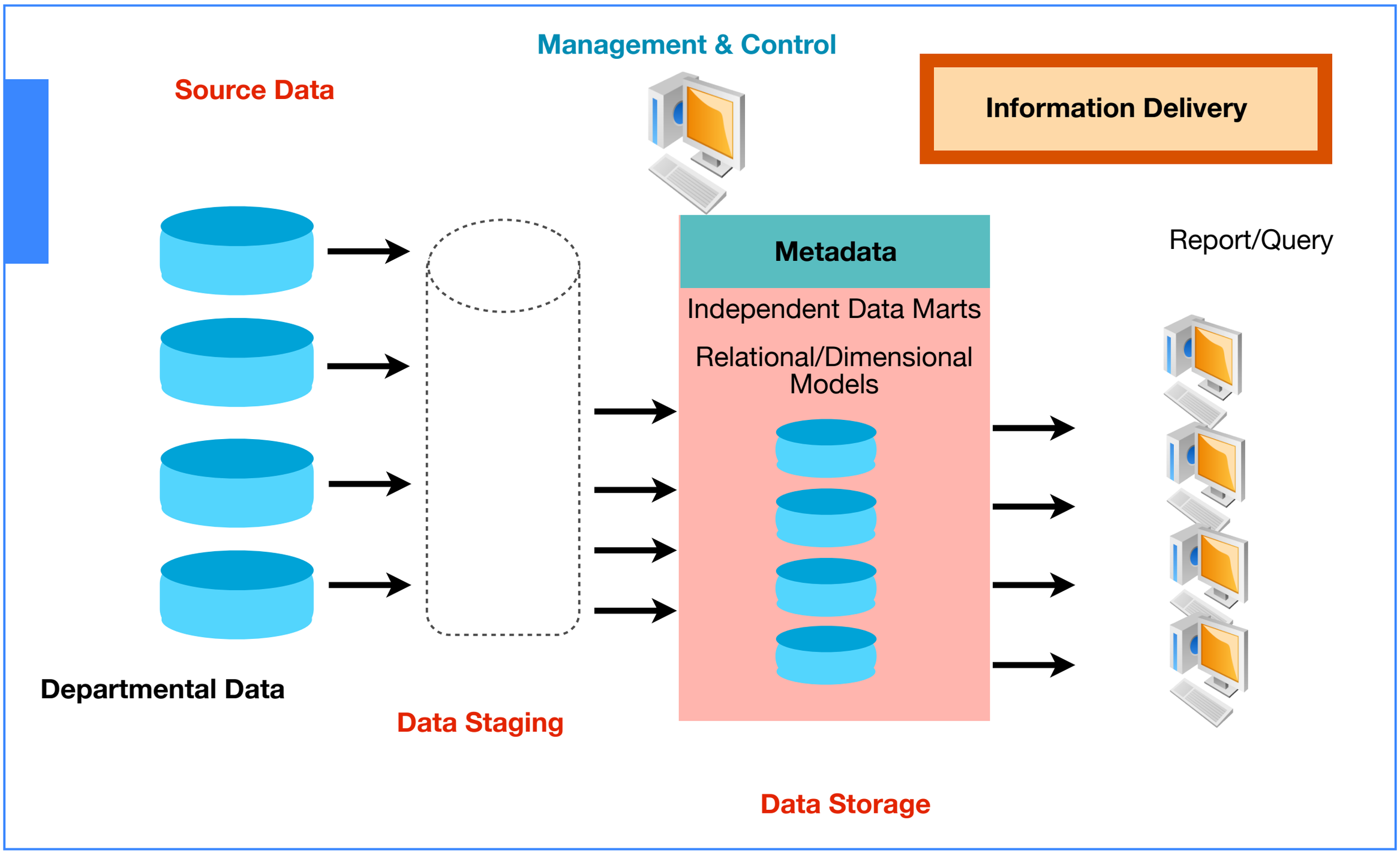


รูปที่ 3-7 ตัวอย่างสถาปัตยกรรมของคลังข้อมูลแบบ centralized data warehouse

2

Independent data marts

Independent data marts จะเป็นสถาปัตยกรรมที่ประกอบไปด้วยกลุ่มของดาต้ามาร์ท ที่ไม่มีการเชื่อมโยงกัน (ดังแสดงในรูปที่ 3-8) โดยแต่ละดาต้ามาร์ทจะให้บริการเฉพาะส่วน หรือแต่ละแผนกเท่านั้น ซึ่งจะทำให้ดาต้ามาร์ทเหล่านั้นไม่สอดคล้องกัน อาจมีข้อมูลที่ไม่ได้เป็นมาตราฐานเดียวกัน และอาจมีข้อมูลที่ไม่ได้สื่อความความเดียวกัน ตัวอย่างเช่น ดาต้ามาร์ทของการขายและการส่งสินค้าอาจมีการทำงานแยกออกจากกันและไม่มีความเกี่ยวเนื่องกัน ซึ่งโดยแท้จริงแล้วทั้งสองแผนกจะมีความเกี่ยวเนื่องกันเป็นอย่างมาก ซึ่งจากความที่ไม่เกี่ยวเนื่องกันอาจทำให้ผู้ใช้เกิด ความยากลำบากในการวิเคราะห์ยอดขายและการส่งของพร้อมๆ กันได้

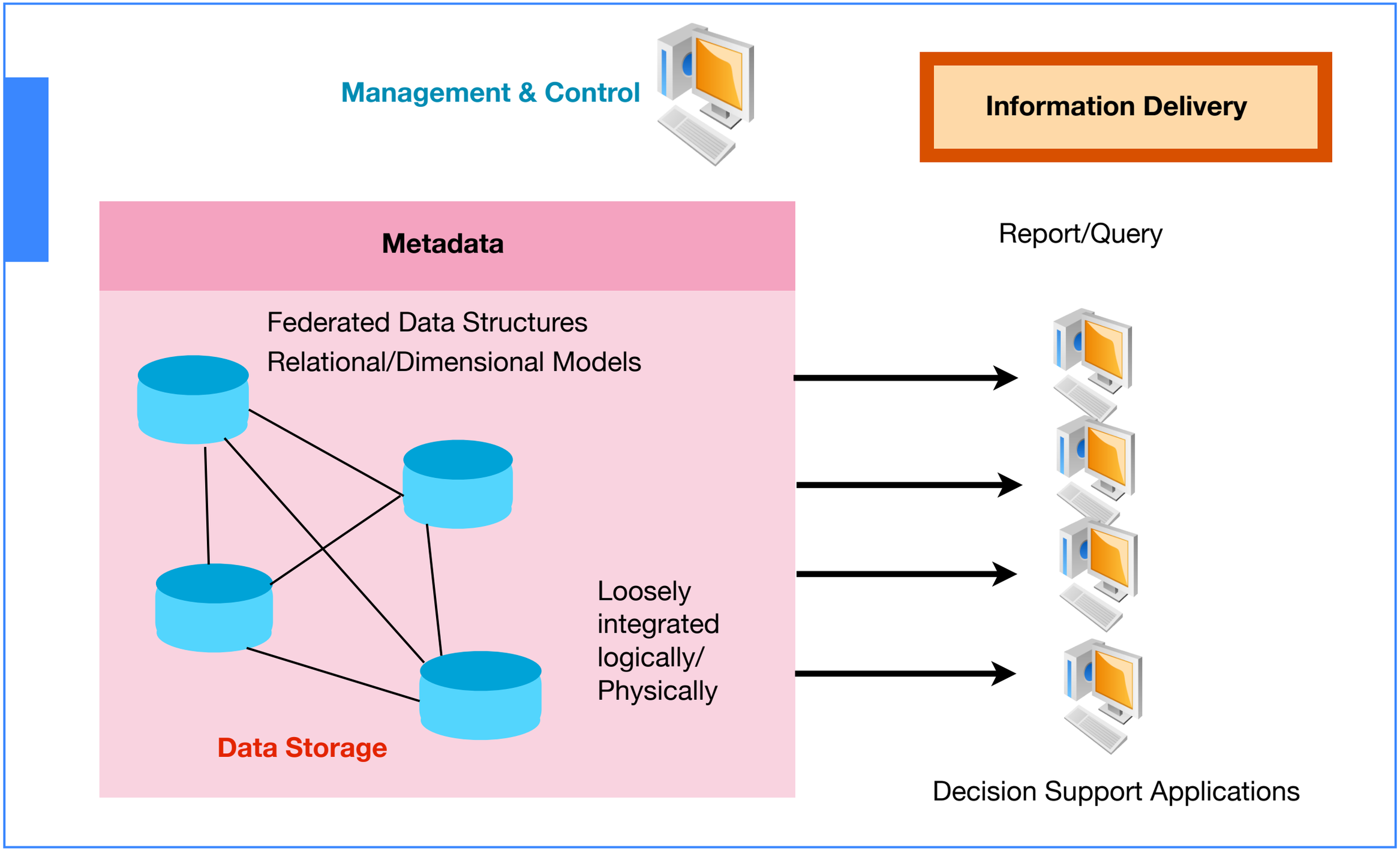


รูปที่ 3-8 ตัวอย่างสถาปัตยกรรมของคลังข้อมูลแบบ independent data marts

3

Federated

Federated จะเป็นสถาปัตยกรรมที่มีลักษณะคล้ายกับ independent data marts ตรงที่สถาปัตยกรรมนี้จะประกอบไปด้วยกลุ่มของดาต้ามาร์ท แต่จะมีความแตกต่างตรงที่ดาต้ามาร์ทเหล่านี้จะถูกรวมเข้าด้วยกัน ซึ่งในการรวมดาต้ามาร์ทเข้าด้วยกันจะทำการรวมทั้งในส่วนของ logical และ physical models ที่จะทำการรวมดาต้ามาร์ทที่มีข้อมูลเหมือนๆ กันเข้าด้วยกัน ซึ่งการรวมดาต้ามาร์ทเข้าด้วยกันจะทำให้ดาต้ามาร์ทต่างๆสามารถใช้ข้อมูลฟิลด์ที่เหมือนกันร่วมกันได้ และยังทำให้สามารถเรียกใช้ข้อมูลจากดาต้ามาร์ทส่วนอื่นๆ ได้อีกด้วย (ตัวอย่างของสถาปัตยกรรมคลังข้อมูลแบบ federated จะแสดงดังรูปที่ 3-9)

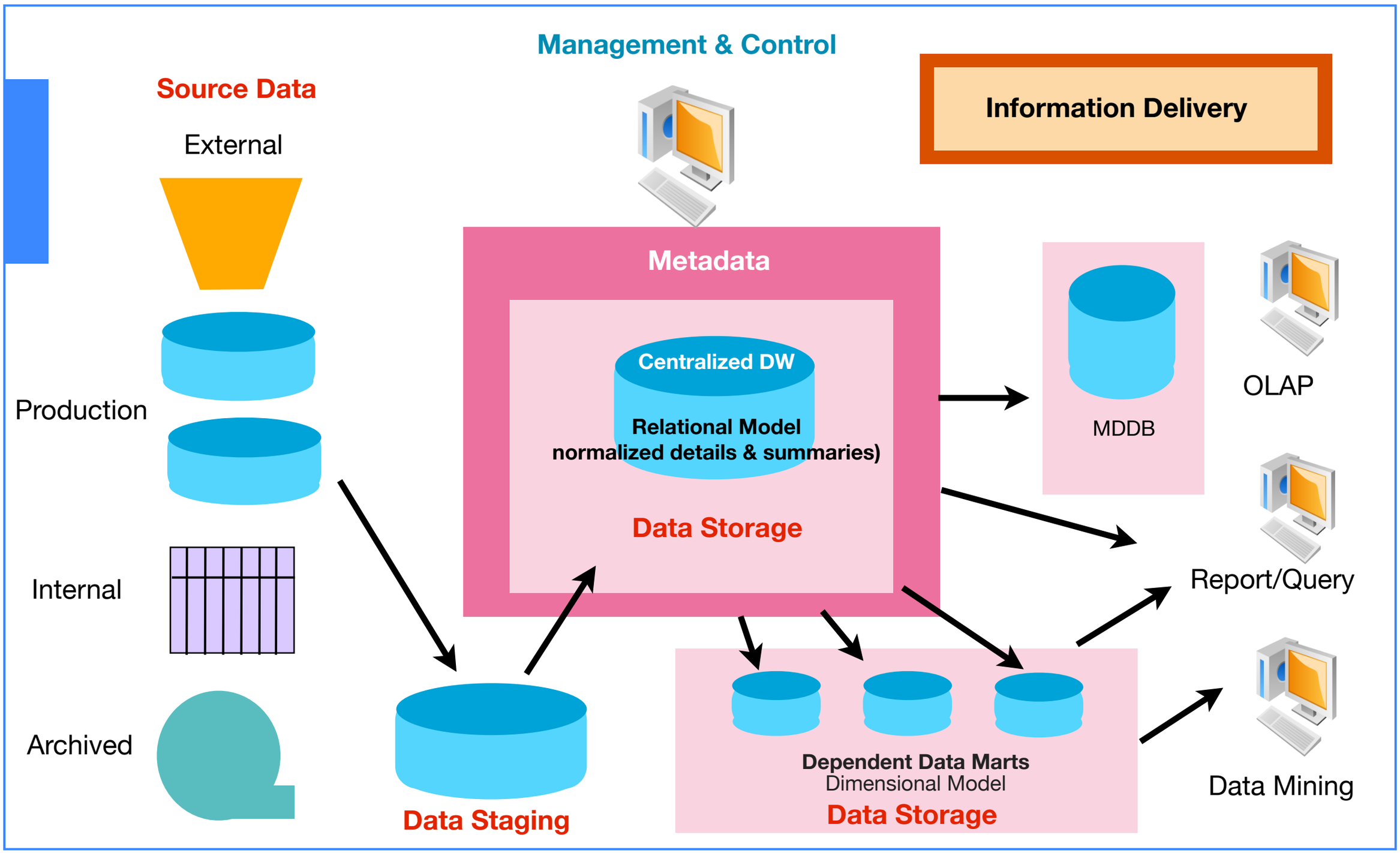


รูปที่ 3-9 ตัวอย่างสถาปัตยกรรมของคลังข้อมูลแบบ federated

4

Hub-and-spoke

Hub-and-spoke จะเป็นสถาปัตยกรรมที่เกิดจากแนวคิดของ Inmon สถาปัตยกรรมนี้จะมีลักษณะคล้ายกับสถาปัตยกรรมแบบ centralized data warehouse ที่สามารถทราบถึงข้อมูลทั้งองค์กร แต่จะมีความแตกต่างตรงที่สถาปัตยกรรมนี้จะมีดาต้ามาร์ทเข้ามาเกี่ยวข้องด้วย ซึ่งแต่ละดาต้ามาร์อาจมีความเกี่ยวข้องกันหรือไม่เกี่ยวข้องกันก็ได้ ข้อมูลในแต่ละดาต้ามาร์ทจะถูกถ่ายโอนมาจาก centralized data warehouse โดยการสร้าง hub จาก centralized data warehouse เพื่อทำการส่งข้อมูลเข้าสู่ดาต้ามาร์ททที่เป็นลักษณะเหมือนปลายทาง โดยที่ดาต้ามาร์ททที่มีความเกี่ยวเนื่องกันจะถูกพัฒนาขึ้นเพื่อใช้ในการวิเคราะห์สำหรับส่วนงาน คิวรีพิเศษ การทำเหมืองข้อมูล และ อื่นๆ เมื่อผู้ใช้ทำการสร้างคิวรีจะสามารถติดต่อกับดาต้ามาร์ทได้โดยตรงหรือจะติดต่อกับ centralized data warehouse ก็ได้ ตัวอย่างของสถาปัตยกรรมแบบ hub-and-spoke จะถูกแสดงดังรูปที่ 3-10

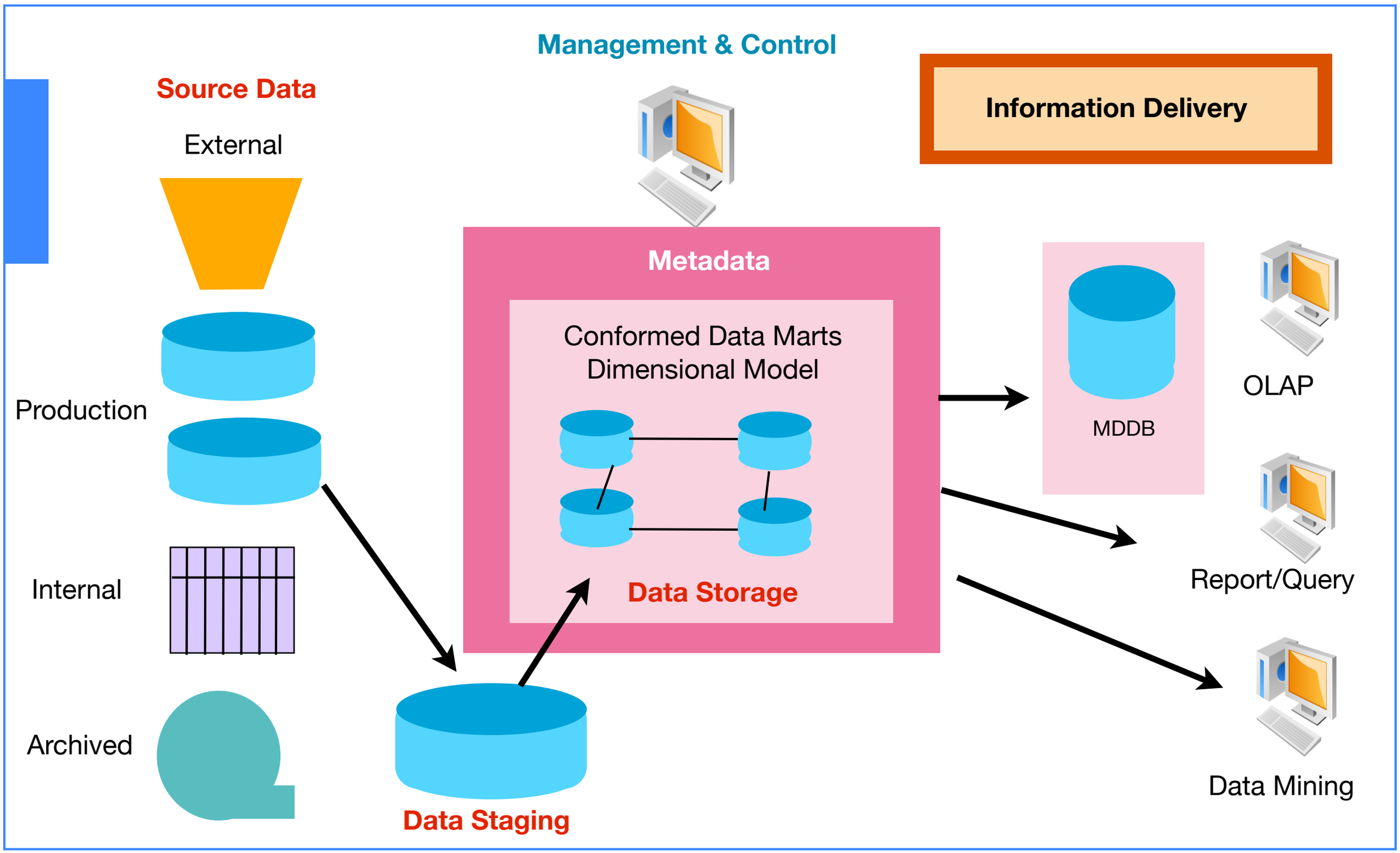


รูปที่ 3-10 ตัวอย่างสถาปัตยกรรมของคลังข้อมูลแบบ hub-and-spoke

5

Data-mart bus

Data-mart bus จะเป็นสถาปัตยกรรมที่เกิดจากแนวคิดของ Kimball ที่ทำการสร้างความสอดคล้องให้กับดาต้ามาร์ท ในการสร้างคลังข้อมูลโดยสถาปัตยกรรมนี้เราจะเริ่มจากการวิเคราะห์ความต้องทางธุรกิจแต่ละหัวข้อ จากนั้นทำการสร้างดาต้ามาร์ทแรกขึ้น เมื่อได้ดาต้ามาร์ทแรกแล้วในส่วนของมิติทางธุรกิจ (business dimension) ของดาต้ามาร์ทแรกจะถูกแบ่งสรรให้กับดาต้ามาร์ทอื่นๆที่จะสร้างขึ้นในอนาคต หัวใจหลักของสถาปัตยกรรมนี้คือการทำให้ดาตามาร์ทสอดคล้องกันซึ่งจะทำให้คลังข้อมูลที่ถูกรวมเข้าด้วยกันตอบสนองการมองภาพรวมของทั้งองค์กรได้



รูปที่ 3-11 ตัวอย่างสถาปัตยกรรมของคลังข้อมูลแบบ data-mart bus

SECTION 7

คำถามท้ายบท

1. จงอธิบายการแบ่งส่วนประกอบของข้อมูลตามฟังก์ชันการทำงานหลักของคลังข้อมูลว่าเราสามารถแบ่งได้กี่กลุ่ม แต่ละกลุ่มประกอบไปด้วยส่วนประกอบอะไรบ้าง และแต่ละส่วนใช้ทำอะไร
2. จงอธิบายถึงฟังก์ชันสำหรับการจัดการและการควบคุมขั้นตอนการทำงานต่างๆของคลังข้อมูล
3. จงอธิบายถึงการเคลื่อนที่ของข้อมูลจากจุดเริ่มต้นไปจุดสิ้นสุด
4. จงแจกแจงฟังก์ชันการทำงานที่เกี่ยวเนื่องกับการสกัดข้อมูล 5 ฟังก์ชัน
5. จงอธิบายถึงการจัดเก็บข้อมูลในพื้นที่พักข้อมูล
6. จงแจกแจงฟังก์ชันการทำงานที่เกี่ยวเนื่องกับการจัดเก็บข้อมูล 5 ฟังก์ชัน
7. จงอธิบายถึงชนิดของสถาปัตยกรรมของคลังข้อมูลทั้ง 5 ชนิด และอธิบายถึงความแตกต่างของแต่ละชนิด

CHAPTER 4

โครงสร้างพื้นฐานของคลังข้อมูล

SECTION 1

แผนการสอนประจำบท

วัตถุประสงค์ของบทเรียน

- เข้าใจถึงความแตกต่างระหว่างสถาปัตยกรรมและโครงสร้างพื้นฐานของคลังข้อมูล
- ศึกษาเกี่ยวกับความสามารถของโครงสร้างพื้นฐานของคลังข้อมูลในการสนับสนุนการทำงานสถาปัตยกรรมของคลังข้อมูล
- ศึกษาเกี่ยวกับส่วนประกอบของโครงสร้างพื้นฐาน
- ศึกษาเกี่ยวกับทางเลือกของโครงสร้างพื้นฐานที่สนับสนุนการประมวลผลแบบขนาน
- ศึกษาเกี่ยวกับการเลือกระบบจัดการฐานข้อมูล

เนื้อหาของบทเรียน

เนื้อหาในบทนี้จะประกอบด้วย ประเภทของโครงสร้างพื้นฐาน (โครงสร้างพื้นฐานการดำเนินงานและทางกายภาพ) ปัจจัยที่ต้องคำนึงถึงในการเลือกฮาร์ดแวร์และระบบปฏิบัติการสำหรับคลังข้อมูล การเลือกแพลตฟอร์มการคำนวณ สถาปัตยกรรมการคำนวณและการประมวลผลคิวรีแบบขนานเบื้องต้น

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารประกอบการสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

กิจกรรมการเรียน-การสอน

- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารประกอบการสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

การวัดและประเมินผล

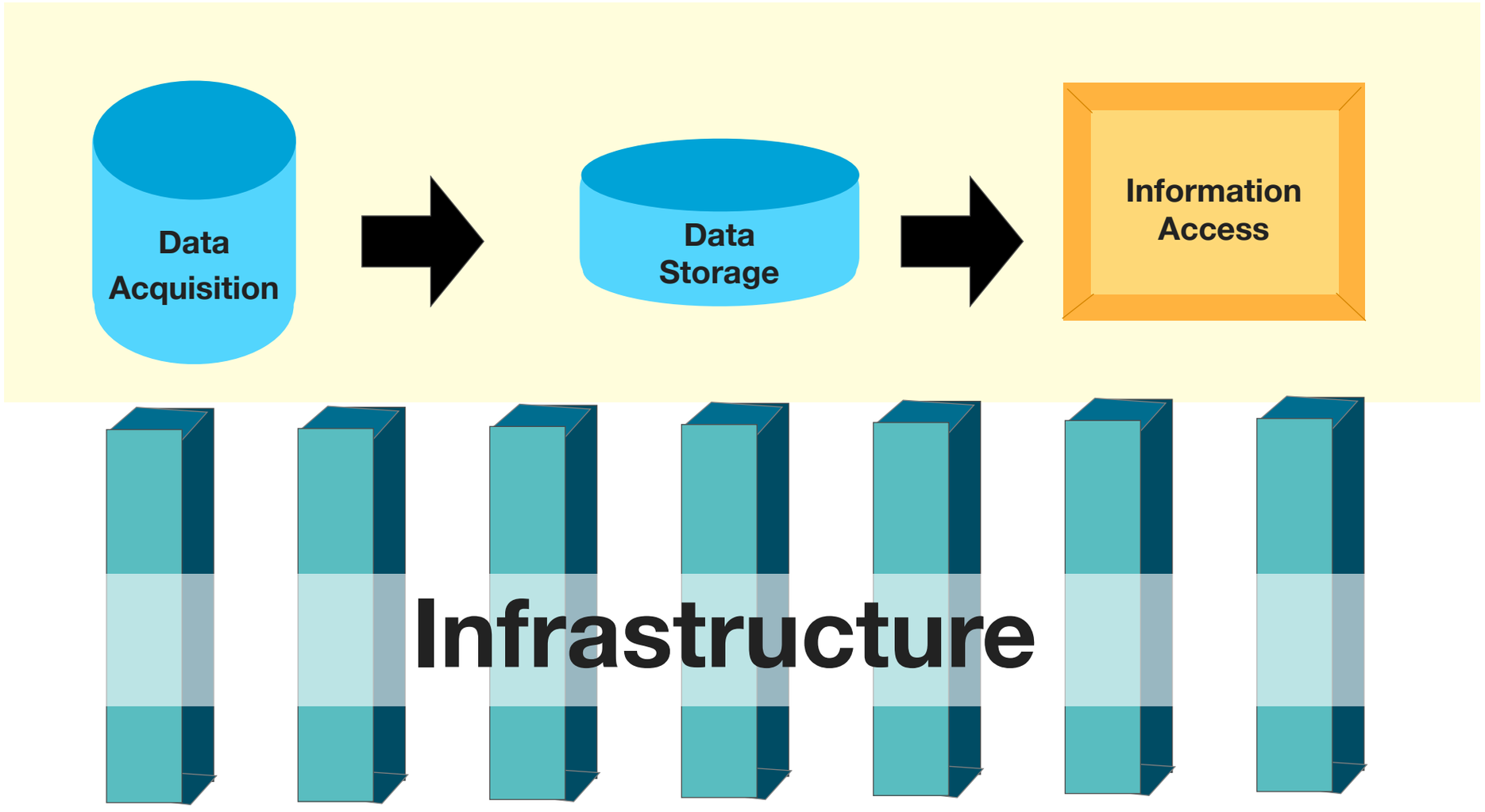
- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

SECTION 2

บทนำ

ในบทก่อนหน้าเราได้ศึกษาเกี่ยวกับสถาปัตยกรรมของคลังข้อมูลที่เอียนเองตามฟังก์ชันการทำงานต่างๆ ที่สำคัญของคลังข้อมูล ซึ่งได้แก่ การได้มาซึ่งข้อมูล (Data acquisition) การจัดเก็บข้อมูล (Data storage) และการส่งผ่านข้อมูลหรือการเข้าถึงข้อมูลในคลังข้อมูล (Information access) ซึ่งจากสถาปัตยกรรมดังกล่าวจะเป็นสิ่งที่แสดงถึงโครงสร้างและความสัมพันธ์ของฟังก์ชันการทำงานต่างๆ ของคลังข้อมูล แต่ในบทนี้เราจะทำการศึกษาเกี่ยวกับโครงสร้างพื้นฐานของคลังข้อมูล (Infrastructure of data warehouse) และทำความเข้าใจถึงบทบาท ความสำคัญ รวมถึงเทคนิค/วิธีต่างๆ สำหรับการออกแบบโครงสร้างพื้นฐานให้มีความเหมาะสมกับคลังข้อมูลที่เราจะทำการสร้างขึ้น

ในการออกแบบหรือสร้างโครงสร้างพื้นฐานของคลังข้อมูล เราจะต้องออกแบบให้สนับสนุนหรือส่งเสริมฟังก์ชันการทำงานต่างๆ ที่เป็นส่วนประกอบของสถาปัตยกรรมของคลังข้อมูล โดยเราสามารถมองโครงสร้างพื้นฐานว่าเป็นรากฐานของสถาปัตยกรรมของคลังข้อมูลก็เป็นได้ ลองพิจารณารูปที่ 4-1 ซึ่งแสดงถึงความสัมพันธ์ระหว่างสถาปัตยกรรมของคลังข้อมูลและโครงสร้างพื้นฐานของคลังข้อมูล โดยโครงสร้างพื้นฐานจะเปรียบเสมือนรากฐานของคลังข้อมูลและมีสถาปัตยกรรมของคลังข้อมูลมาซ้อนทับอยู่

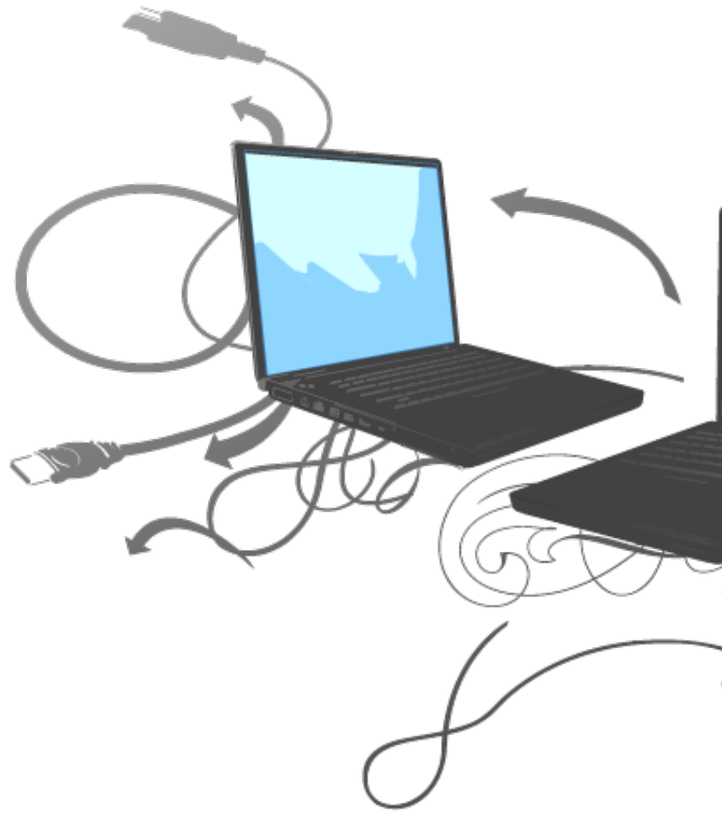


รูปที่ 4-1 ตัวอย่างโครงสร้างพื้นฐานที่สนับสนุนสถาปัตยกรรมของคลังข้อมูล

เมื่อเราทำการพิจารณาเกี่ยวกับโครงสร้างพื้นฐานของคลังข้อมูล เราจะทราบว่าโครงสร้างพื้นฐานจะประกอบไปด้วยส่วนประกอบต่างๆ มากมาย เช่น

- ฮาร์ดแวร์ (Hardware)
- ระบบปฏิบัติการ (Operating system)
- ระบบการจัดการฐานข้อมูล (Database management system, DBMS) LAN
- WAN ซอร์ฟแวร์และเครื่องมือต่างๆ (Software and tools)

ซึ่งในตัวของซอร์ฟแวร์ที่ใช้อาจจะรวมไปถึงซอฟต์แวร์ระบบเครือข่าย (Network software) ซอร์ฟแวร์ฐานข้อมูล (Database software) และอื่นๆ นอกจากนั้นโครงสร้างพื้นฐานจะรวมถึงทรัพยากรมนุษย์ ขั้นตอนต่างๆ และการอบรมอีกด้วย ซึ่งจากส่วนประกอบที่หลากหลายของโครงสร้างพื้นฐาน เราสามารถแบ่งส่วนประกอบออกเป็น 2 หมวดหมู่ที่มีความแตกต่างกัน คือ 1) โครงสร้างพื้นฐานการดำเนินงาน (Operational infrastructure) และ 2) โครงสร้างพื้นฐานทางกายภาพ (Physical infrastructure) ซึ่งเราจะทำการศึกษาส่วนประกอบของแต่ละหมวดหมู่เพื่อให้เข้าใจถึงส่วนประกอบทั้งหมดของโครงสร้างพื้นฐานของคลังข้อมูล



SECTION 3

โครงสร้างพื้นฐานการดำเนินงาน

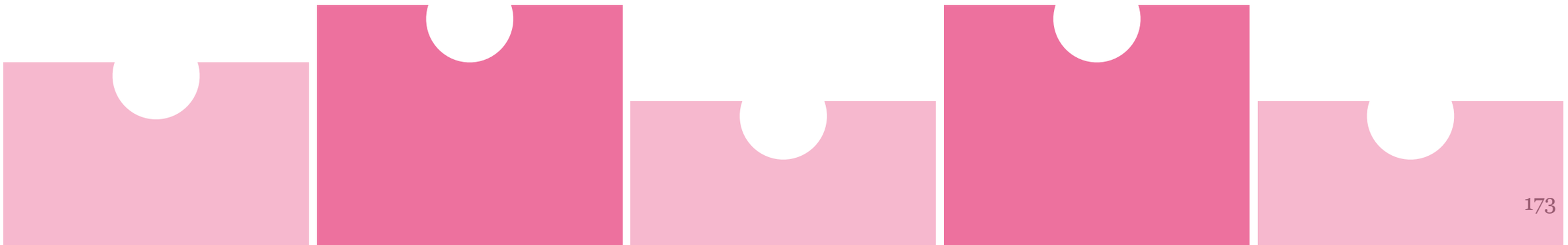
โครงสร้างพื้นฐานการดำเนินงาน

จะประกอบไปด้วยทรัพยากรมนุษย์ กระบวนการต่างๆ การอบรม และซอร์ฟแวร์การจัดการ (Management software) ที่สนับสนุนฟังก์ชันการทำงานต่างๆ ในสถาปัตยกรรมของคลังข้อมูล ซึ่งจากส่วนประกอบของโครงสร้างพื้นฐานการดำเนินงาน “ทรัพยากรมนุษย์” และ “กระบวนการ” จะไม่ได้หมายถึง กลุ่มคนหรือกระบวนการที่มีหน้าที่ในการสร้างคลังข้อมูลแต่จะหมายถึง “กลุ่มคนหรือกระบวนการที่จะทำให้คลังข้อมูลดำเนินต่อไปได้” ส่วนประกอบเหล่านี้จะเป็นส่วนที่สนับสนุนในเรื่องของการจัดการและการดูแลรักษาคลังข้อมูลให้ดำเนินต่อไปได้อย่างมีประสิทธิภาพ

โดยส่วนใหญ่ของการออกแบบโครงสร้างพื้นฐาน ผู้สร้างคลังข้อมูลมักจะให้ความสนใจกับฮาร์ดแวร์และซอร์ฟแวร์ค่อนข้างมาก และไม่ค่อยสนใจกับส่วนประกอบอื่นของโครงสร้างพื้นฐานการดำเนินงานเท่าที่ควร การให้ความสนใจกับฮาร์ดแวร์และซอร์ฟแวร์นั้นเป็นสิ่งที่ถูก ถ้าเรามีฮาร์ดแวร์และซอร์ฟแวร์ที่ถูกต้องและเหมาะสมกับฟังก์ชันการทำงานต่าง ๆ ในสถาปัตยกรรมของคลังข้อมูล จะช่วยให้เราสามารถสร้างและใช้งานคลังข้อมูลได้อย่างมีประสิทธิภาพ

แต่อย่างไรก็ดีในการออกแบบส่วนประกอบของโครงสร้างพื้นฐานของคลังข้อมูลก็ยังต้องการโครงสร้างพื้นฐานการดำเนินงานในการดูแลและจัดการสิ่งต่าง ๆ อยู่ดี ดังนั้นเราสามารถกล่าวได้ว่าโครงสร้างพื้นฐานการดำเนินงานก็มีความสำคัญเทียบเท่ากับฮาร์ดแวร์และซอร์ฟแวร์ที่ทำให้คลังข้อมูลสามารถทำงานได้อย่างมีประสิทธิภาพ ซึ่งถ้าเราไม่มีโครงสร้างพื้นฐานการดำเนินงานที่เหมาะสมแล้ว คลังข้อมูลที่เราสร้างขึ้นจะไม่สามารถทำงานได้อย่างเต็มประสิทธิภาพและไม่สามารถคงอยู่ยืนยง

ดังนั้นเราควรให้ความใส่ใจหรือสนใจเกี่ยวกับรายละเอียดต่าง ๆ ของโครงสร้างพื้นฐานการดำเนินงานในการออกแบบโครงสร้างพื้นฐานของคลังข้อมูลด้วย



SECTION 4

โครงสร้างพื้นฐานทางกายภาพ

จะประกอบไปด้วยฮาร์ดแวร์ต่างๆ ระบบปฏิบัติการ ระบบฐานข้อมูล เครื่องข่าย ซอร์ฟแวร์เครือข่าย และอื่นๆ ดังแสดงในรูปที่ 4-2 จะแสดงส่วนประกอบต่างๆ ของโครงสร้างพื้นฐานทางกายภาพ โดยในการเลือกส่วนประกอบต่างๆ เราจะมีหลายทางเลือกด้วยกัน

เช่น ในท้องตลาดจะมีฮาร์ดแวร์ให้เลือกใช้หลายผลิตภัณฑ์ด้วยกัน การเลือกใช้ฮาร์ดแวร์ที่จะรองรับคลังข้อมูลที่เราจะสร้างขึ้นนั้นจะสามารถทำได้ค่อนข้างยาก เนื่องจากเราต้องพิจารณาหลายๆ ปัจจัยที่อาจส่งผลกระทบต่อการทำงานของฮาร์ดแวร์นั้นๆ แต่ก่อนที่เราจะทำการเลือกซื้อหรือเลือกใช้ฮาร์ดแวร์ใด เราต้องไม่ลืมว่าฟังก์ชันการทำงานของคลังข้อมูลจะประกอบไปด้วย การสกัดข้อมูล (data extraction) การเปลี่ยนแปลงเปลี่ยนรูปข้อมูล (data transformation) การรวบยอดข้อมูล (data integration) และการจัดการกับ staging area ซึ่งฟังก์ชันการทำงานทั้งหมดจะต้องทำงานบนฮาร์ดแวร์และระบบปฏิบัติการที่เราเลือก

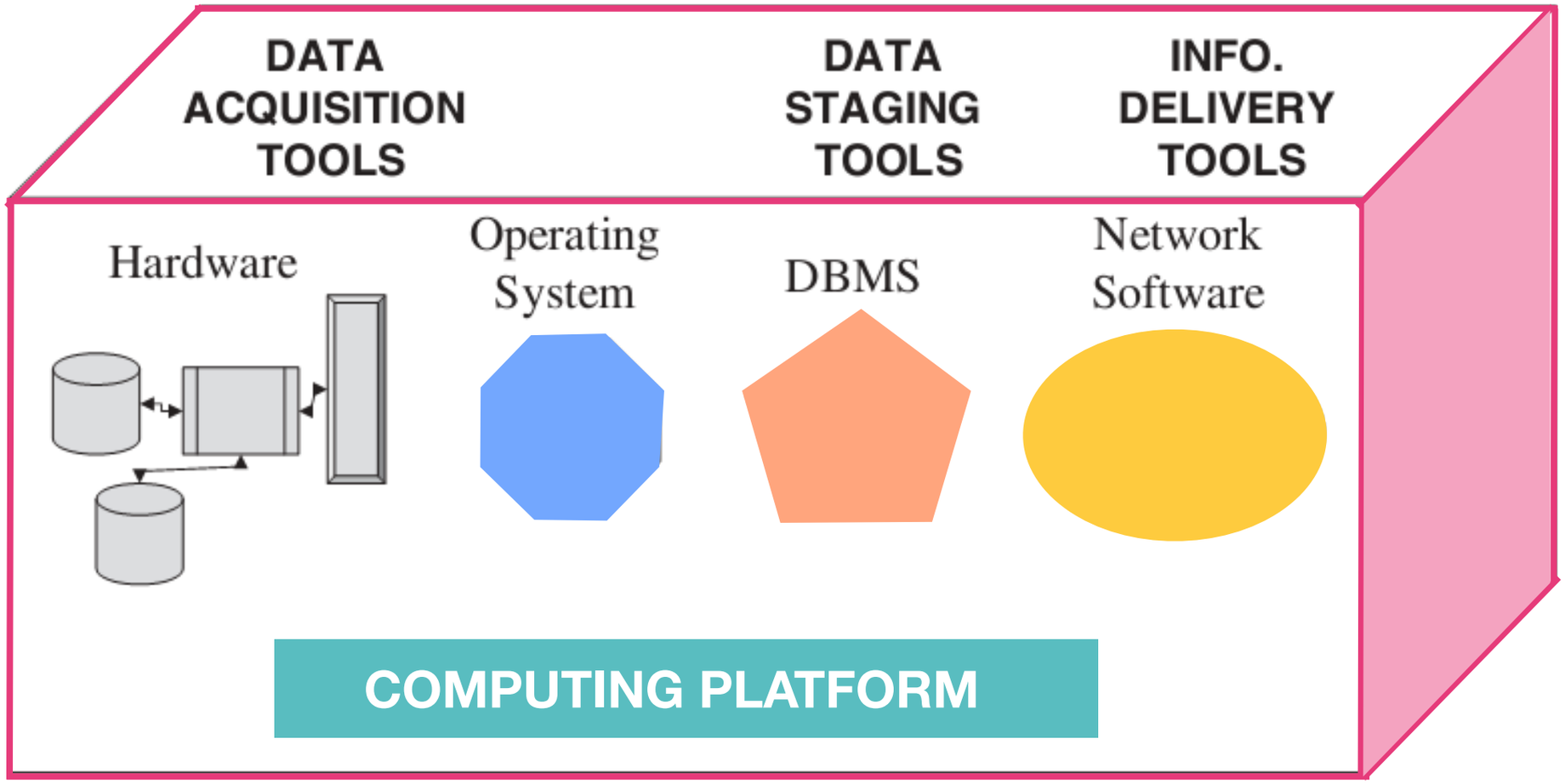
ดังนั้นในการเลือกฮาร์ดแวร์และระบบปฏิบัติการ เราจะต้องเลือกสิ่งที่สนับสนุนการทำงานของฟังก์ชันต่างๆ ด้วย โดยในการเลือกฮาร์ดแวร์จะมีคำแนะนำดังนี้

1 เราต้องมั่นใจได้ว่าฮาร์ดแวร์ที่เราเลือกใช้สำหรับรองรับการทำงานของคลังข้อมูลจะสามารถเพิ่มขยายหรือต่อเติมได้เมื่อคลังข้อมูลมีการเติบโตเพิ่มขึ้นโดยดูจากจำนวนผู้ใช้ จำนวนคิวรีที่ทำการสืบค้นข้อมูล และความซับซ้อนของคิวรีที่เพิ่มขึ้น

2 เราต้องมั่นใจได้ว่าฮาร์ดแวร์ที่เราจะซื้อนั้นมีการบำรุงรักษาหรือซ่อมบำรุงจากผู้ขายในระดับสูงที่สุดเท่าที่จะเป็นไปได้ ซึ่งเมื่อเกิดปัญหาขึ้นที่ฮาร์ดแวร์เราจะสามารถขอความช่วยเหลือจากผู้ขายได้

3 เราต้องทำการตรวจสอบข้อคิดเห็นที่เกี่ยวข้องกับผลิตภัณฑ์/ฮาร์ดแวร์ที่เรากำลังจะซื้อจากแหล่งข้อมูลต่างๆ ว่ามีใครพูดถึงการทำงานที่ผิดพลาด หรือการเสียหายของฮาร์ดแวร์ที่เราสนใจหรือไม่

4 เราต้องตรวจสอบความมั่นคงของบริษัทผู้ขายฮาร์ดแวร์ว่ายังคงมีสถานะทางการเงินและการค้าที่ดีหรือไม่ เนื่องจากเราต้องต้องแน่ใจว่าเมื่อเราทำการซื้อฮาร์ดแวร์จากบริษัทนั้นๆ แล้ว บริษัทเหล่านั้นจะไม่ล้มเลิกหรือปิดกิจการ ซึ่งจะทำให้เราไม่สามารถเรียกใช้บริการซ่อมบำรุงจากผู้ขายได้



รูปที่ 4-2 ส่วนประกอบของโครงสร้างพื้นฐานทางกายภาพ

ในการเลือกระบบปฏิบัติการก็มีข้อแนะนำสำหรับการเลือกเช่นกัน ซึ่งสิ่งแรกที่ระบบปฏิบัติที่เราเลือกจะต้องมีคือ ระบบปฏิบัติการจะต้องทำงานร่วมกันกับฮาร์ดแวร์ที่เราเลือกได้ นอกจากนั้นเราจะต้องคำนึงถึงปัจจัยอื่นๆ ดังต่อไปนี้

1

ระบบปฏิบัติการที่เลือกจะต้องมีความสามารถในการยืดขยายต่อเติมได้ (Scalability) เนื่องจากคลังข้อมูลที่เราสร้างขึ้นนั้นจะเติบโตขึ้นทุกวัน และเติบโตอย่างรวดเร็ว ทั้งในแง่ของจำนวนข้อมูลและจำนวนผู้ใช้ รวมถึงจำนวนคิวรีที่ต้องทำการประมวลผลเพิ่มขึ้น ดังนั้นระบบปฏิบัติการสำหรับคลังข้อมูลจะต้องสนับสนุนความเติบโตเหล่านี้ด้วย

2

ระบบปฏิบัติการที่เลือกจะต้องมีความสามารถในการป้องกันหรือปกป้องทรัพยากรของระบบเมื่อมีผู้ใช้ต้องการที่จะเข้าถึงข้อมูลหรือใช้ทรัพยากรของคลังพร้อมกันหลายราย ระบบปฏิบัติการจะต้องมีความสามารถในการป้องกันการคุกคามหรือป้องกันการใช้ทรัพยากรที่มากเกินความจำเป็นได้ และรวมถึงเตรียมความปลอดภัยให้กับผู้ใช้แต่ละคน

3

ระบบปฏิบัติการที่เลือกควรจะต้องมีความน่าเชื่อถือ (Reliability) ไม่เกิดข้อผิดพลาดในการทำงาน หรือมีการป้องกันส่วนอื่นๆเมื่อเกิดข้อผิดพลาดของการทำงานเกิดขึ้น

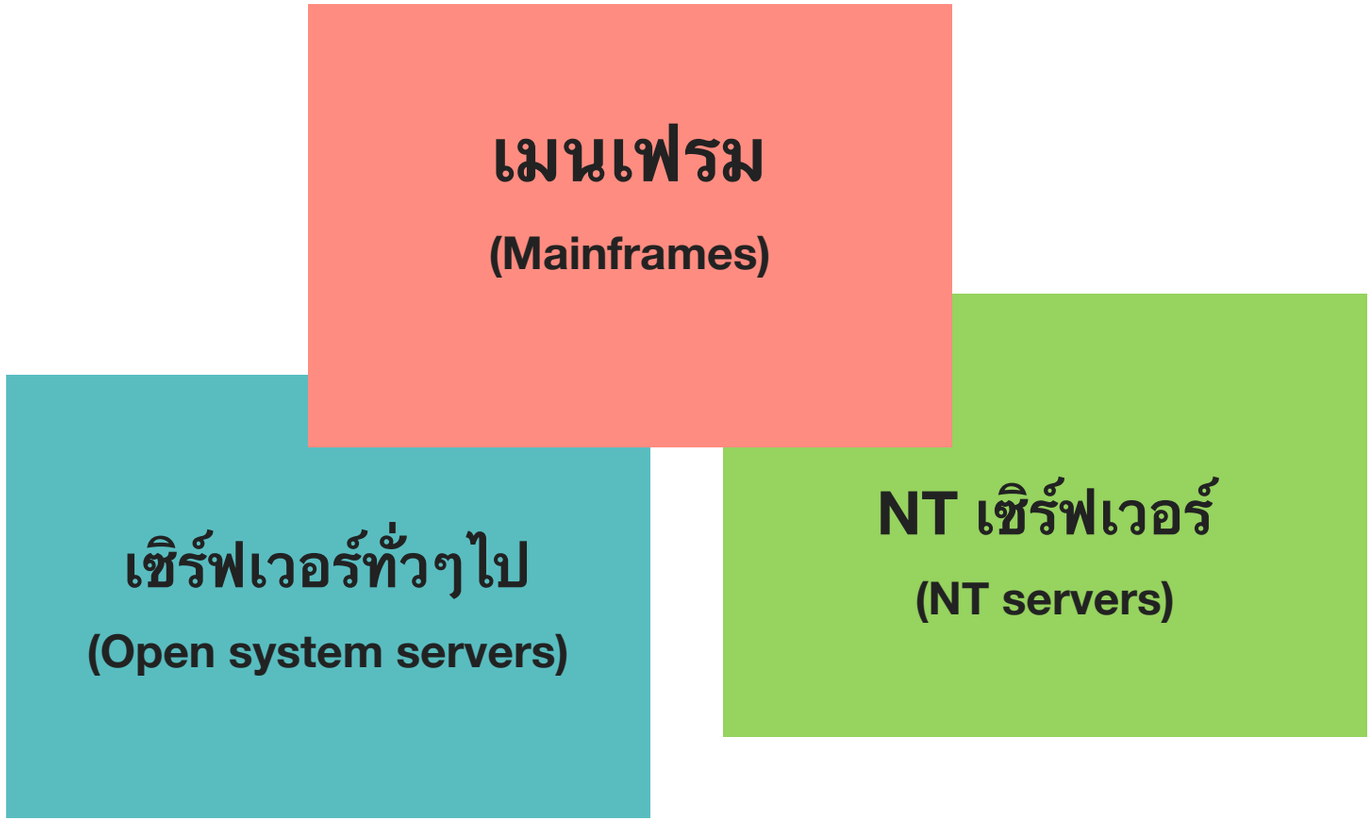
4 ระบบปฏิบัติการที่เลือกควรจะต้องสามารถทำงานได้ (Availability) เมื่อมีการยกเลิกการทำงานบางโปรเซสของแอพลิเคชันหนึ่งๆ

5 ระบบปฏิบัติการที่เลือกจะต้องมีความสามารถในการกำหนดลำดับความสำคัญของโปรเซส และสามารถทำการเปลี่ยนการคำนวณไปยังโปรเซสที่มีความสำคัญสูงกว่าในกรณีที่ต้องการได้

6 ระบบปฏิบัติการที่เลือกจะต้องมีความสามารถในการให้บริการหลายๆการร้องขอรับบริการ โดยการแบ่ง thread ไปยังหลายๆ โปรเซสเซอร์ได้

7 ระบบปฏิบัติการที่เลือกจะต้องมีความสามารถในการป้องกันการล่วงละเมิดการใช้หน่วยความจำของงานอื่นๆ เมื่อมีการประมวลผลคิวรีพร้อมๆ กันหลายงาน

จากคำแนะนำและเกณฑ์ในการเลือกฮาร์ดแวร์และระบบปฏิบัติการสำหรับคลังข้อมูลเราจะต้องเลือกสิ่งที่เราจะใช้ให้ตรงกับความต้องการหรือสิ่งแวดล้อมที่เรามีอยู่ให้มากที่สุด ซึ่งในการเลือกฮาร์ดแวร์นั้นเราจะมีทางเลือกไม่ค่อยมาก ซึ่งโดยส่วนใหญ่ของคลังข้อมูลจะทำงานบนฮาร์ดแวร์ 3 ประเภทหลักๆ ด้วยกันคือ



เมนเฟรม
(Mainframes)

- ส่วนมากจะเป็นฮาร์ดแวร์ที่เหลือมาจากการสร้างระบบการดำเนินงานก่อนหน้าหรือระบบดั้งเดิม
- นิยมใช้สำหรับระบบการดำเนินงาน
- ไม่คุ้มค่าสำหรับคลังข้อมูล
- สามารถเพิ่มต่อหรือยืดขยายได้ยาก
- ไม่ค่อยมีใครใช้สำหรับคลังข้อมูล

เซิร์ฟเวอร์ทั่วๆไป
(Open system servers)

- เป็น UNIX เซิร์ฟเวอร์ที่ได้รับความนิยมและเหมาะสมกับคลังข้อมูลขนาดกลาง
- ค่อนข้างเสถียร แข็งแรง ทนทาน
- สามารถประยุกต์ใช้กับการคำนวนแบบขนานได้

NT เซิร์ฟเวอร์
(NT servers)

- สนับสนุนคลังข้อมูลขนาดกลาง
- ความสามารถในการคำนวณแบบขนานมีจำกัด
- ค่าใช้จ่ายคุ้มค่ากับคลังข้อมูลขนาดกลางและขนาดเล็ก

จากทั้ง 3 ทางเลือกข้างต้น เราต้องทำการตัดสินใจเลือกเซิร์ฟเวอร์ที่สามารถต่อเติมหรือยืดขยายได้ (Scalability) และมีประสิทธิภาพที่ดีในการค้นคืนข้อมูลให้กับคิวรีต่างๆ (Optimal query performance) ที่เฉพาะเจาะจง เป็นคิวรีที่มีความซับซ้อนและไม่สามารถคาดเดาได้ และมีประสิทธิภาพในการทำงานที่ดีเมื่อจำนวนผู้ใช้มีจำนวนเพิ่มขึ้น (โดยส่วนใหญ่จะเพิ่มขึ้นเป็น 2 เท่าภายใน 6 เดือน) คลังข้อมูลที่เราทำการสร้างขึ้นอาจจำเป็นต้องเพิ่มเนื้อหาหรือผลสรุปของข้อมูล โดยทำการเพิ่มหัวข้อต่างๆ ทางธุรกิจ (Business subject) หรือเพิ่มดาต้ามาร์ทให้กับคลังข้อมูลที่เราสร้างขึ้นซึ่งอาจจะทำให้การใช้พื้นที่จากเดิมประมาณ 200-300 GB ในตอนเริ่มต้นจะเพิ่มขั้นไปเป็นมากกว่า 1 เทราไบต์ภายใน 18-24 เดือน

นอกเหนือจากฮาร์ดแวร์และระบบปฏิบัติการที่เราจะต้องทำการพิจารณาแล้ว เรายังต้องพิจารณาแพลตฟอร์มการคำนวณ (Computing platform) ซึ่งเป็นส่วนที่ใช้สำหรับทำการประมวลผลในหลายๆ ฟังก์ชันการทำงานของคลังข้อมูล แพลตฟอร์มการคำนวณนั้นจะเป็นกลุ่มของฮาร์ดแวร์ ระบบปฏิบัติการ เครือข่าย และ ซอร์ฟแวร์เครือข่าย ที่สนับสนุนการทำงานต่างๆ ของคลังข้อมูล เช่น

สนับสนุนการได้มาซึ่งข้อมูล (data acquisition) เช่น การสกัดข้อมูล การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล การทำความสะอาดข้อมูล และการรวมข้อมูลเข้าด้วยกัน เป็นต้น

สนับสนุนการจัดเก็บข้อมูล (data storage) เช่น การถ่ายโอนข้อมูล การจัดเก็บข้อมูล และการจัดการต่างๆ กับข้อมูล เป็นต้น

สนับนุนการเข้าถึง/ส่งผ่านข้อมูล (information access/delivery) เช่น การสร้างรายงาน การประมวลผลคิวรี และการวิเคราะห์ที่ซับซ้อน เป็นต้น

จากฟังก์ชันการทำงานข้างต้นที่แพลตฟอร์มการคำนวณต้องสนับสนุนหรือส่งเสริมการทำงาน เราจะต้องทำการเลือกแพลตฟอร์มการคำนวณให้มีความเหมาะสมกับสภาวะแวดล้อมที่เรามีอยู่ ซึ่งทางเลือกสำหรับการเลือกแพลตฟอร์มการคำนวณจะมีหลายทางเลือกด้วยกัน ดังนี้

การเลือกใช้แพลตฟอร์มเดียว

จะเป็นแพลตฟอร์มที่ค่อนข้างตรงไปตรงมาและเป็นทางเลือกที่ง่ายที่สุดในการดำเนินการสร้างคลังข้อมูล การใช้เพียงแพลตฟอร์มเดียวจะทำให้ทุกๆ ฟังก์ชันที่เป็นการคำนวณส่วนหลัง (back-end functions) เช่น การสกัดข้อมูล การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล และทุกฟังก์ชันที่เป็นการคำนวณเบื้องหน้า (front-end functions) นั้นมีการดำเนินงานบนแพลตฟอร์มเดียว อาทิเช่น การใช้เมนเฟรม มินิคอมพิวเตอร์ หรือ การใช้เซิร์ฟเวอร์ที่เป็น UNIX เพียงตัวเดียว

การเลือกใช้แพลตฟอร์มเดียวมี **ข้อดี** ที่ว่าเราจะไม่ต้องพบเจอกับปัญหาเกี่ยวกับความสอดคล้องหรือเข้ากันได้ระหว่างแพลตฟอร์มต่างๆ ซึ่งถ้าเราใช้หลายแพลตฟอร์มเราอาจเจอกับปัญหาความไม่สอดคล้องกันของแพลตฟอร์มซึ่งจะทำให้ไม่สามารถทำงานได้

อีกข้อดีหนึ่งของการใช้แพลตฟอร์มเดียวคือ การไหลเวียนของข้อมูลตั้งแต่เริ่มต้นจนสิ้นสุดกระบวนการทำงานจะค่อนข้างนุ่มนวล เราไม่ต้องทำการเปลี่ยนแปลงรูปแบบของข้อมูลใดๆ และไม่ต้องใช้ซอร์ฟแวร์ตัวกลาง (middleware) มาช่วยในการทำงานแต่อย่างใด

แต่อย่างไรก็ดีการใช้แพลตฟอร์มเดียว เช่น การใช้เมนเฟรม และ มินิคอมพิวเตอร์ ก็มี**ข้อเสีย**คือ อาจจะทำให้ไม่สามารถปรับเปลี่ยนอุปกรณ์หรือเพิ่มขยายต่อเติมได้มากนัก (upgrade) ซึ่งการเพิ่มขยายต่อเติมนั้นเป็นสิ่งหนึ่งที่จำเป็นมากเมื่อคลังข้อมูลมีปริมาณข้อมูลและปริมาณผู้ใช้งานเพิ่มขึ้น

อีกข้อเสียหนึ่งของการใช้แพลตฟอร์มเดียวก็คือ ซอร์ฟแวร์ต่างๆที่เราเลือกใช้ในบางฟังก์ชันการทำงานจะไม่สนับสนุนแพลตฟอร์มเมนเฟรมหรือมินิคอมพิวเตอร์ ซึ่งถ้าเราไม่ใช้ซอร์ฟแวร์เหล่านั้นในการทำงาน คลังข้อมูลก็ไม่สามารถทำงานได้

การเลือกใช้แพลตฟอร์มแบบผสมผสาน

จะเป็นการผสมผสานการใช้แพลตฟอร์มต่างๆ หลายแพลตฟอร์มเข้าด้วยกัน ซึ่งการผสมผสานจะเป็นไปตามฟังก์ชันก์การทำงานต่างๆ เช่น

การสกัดข้อมูล ซึ่งโดยส่วนใหญ่จะทำการสกัดข้อมูลที่แหล่งข้อมูล ตัวอย่างเช่น ข้อมูลการใช้โทรศัพท์ของลูกค้าจากบริษัทผู้ให้บริการเครือข่ายโทรศัพท์จะทำการสกัดข้อมูลที่แหล่งข้อมูลหรือระบบการดำเนินงานที่อาจใช้มินิคอมพิวเตอร์ในการดำเนินงาน จากนั้นจะทำการสร้างแฟ้มข้อมูลสำหรับข้อมูลที่ถูกสกัดออกมา แล้วจึงค่อยถ่ายโอนไปยังพื้นที่พักข้อมูลต่อไป

หรือในอีกระบบหนึ่งคือ ระบบสั่งสินค้าทางอีเมลล์ที่ทำงานบนเครื่องเมนเฟรม จะทำการสกัดข้อมูลแล้วเก็บข้อมูลไว้ในแฟ้มข้อมูลหนึ่งๆ ก่อนที่จะทำการถ่ายโอนข้อมูลไปยังพื้นที่พักข้อมูล จากตัวอย่างทั้งสองเราจะเห็นว่าจะมีการสกัดข้อมูลที่แหล่งข้อมูล แล้วจึงค่อยทำการถ่ายโอนข้อมูลที่ถูกสกัดแล้วไปยังพื้นที่พักข้อมูล ซึ่งจะไม่มีการคัดลอกข้อมูลทั้งหมดไปยังพื้นที่พักข้อมูล

ดังนั้นเมื่อเราต้องมีฟังก์ชันการสกัดข้อมูลเก็บไว้ที่แหล่งข้อมูลหรือระบบการดำเนินงานจึงทำให้เราต้องผสมผสานหลายแพลตฟอร์มเข้าด้วยกัน

การรวมข้อมูลและเปลี่ยนรูปแบบของข้อมูล จะเป็นการทำงานหลังจากขั้นตอนการสกัดข้อมูล โดยจะนำแฟ้มข้อมูลที่ถูกสกัดแล้วมาทำการเปลี่ยนรูปแบบข้อมูล และรวมข้อมูลหลายๆ แฟ้มเข้าด้วยกันเพื่อลดจำนวนแฟ้มข้อมูล ซึ่งโดยส่วนใหญ่ของกระบวนการทำงานดังกล่าวมักจะทำงานที่แหล่งข้อมูล/ระบบการดำเนินงาน ซึ่งจะช่วยให้เราสามารถลดจำนวนแฟ้มข้อมูลที่ต้องทำการถ่ายโอนไปยังพื้นที่พักข้อมูลได้

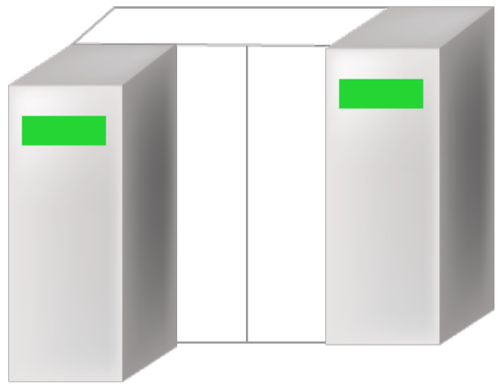
การทำความสะอาดข้อมูลเบื้องต้น จะเป็นการเติมค่าของข้อมูลที่ขาดหายไปให้กับข้อมูลที่ถูกสกัดมาจากแหล่งข้อมูล รวมถึงการกำหนดค่า default value และการเปลี่ยนแปลงข้อมูลในรูปแบบอื่นๆ ซึ่งการทำงานกระบวนการนี้จะเหมือนกับกระบวนการทำงานก่อนหน้าที่จะดำเนินการที่ระบบการดำเนินงานหรือแหล่งข้อมูล

การเปลี่ยนแปลงเปลี่ยนรูปข้อมูลและการรวมข้อมูลเข้าด้วยกัน จะเป็นขั้นตอนการทำงานที่ฟังก์ชันการเปลี่ยนแปลงและการเปลี่ยนรูปข้อมูล และการรวบรวมหรือรวบยอดข้อมูลเข้าด้วยกัน ซึ่งโดยส่วนใหญ่แล้วทั้ง 2 ขั้นตอนนี้จะใช้ซอร์ฟแวร์หรือเครื่องมือต่างๆ มาช่วยในการดำเนินการ ซึ่งในการที่จะติดตั้งซอร์ฟแวร์สำหรับทั้ง 2 ฟังก์ชันเราควรจะติดตั้งและกำหนดการทำงานให้อยู่ที่พื้นที่พักข้อมูล

การตรวจสอบและการควบคุมคุณภาพ จะเป็นการตรวจสอบคุณภาพของข้อมูลที่จะทำการจัดเก็บเข้าสู่ฐานข้อมูลของคลังข้อมูล ซึ่งฟังก์ชันการทำงานนี้จะอยู่ที่พื้นที่พักข้อมูล

การสร้างแฟ้มสำหรับการถ่ายโอนข้อมูล (load image) จะเป็นการสร้างแฟ้มสำหรับการถ่ายโอนข้อมูลสำหรับข้อมูลที่ถูกสกัดมาจากแหล่งข้อมูลแฟ้มหนึ่งๆ ซึ่งขั้นตอนนี้จะดำเนินการที่พื้นที่พักข้อมูล

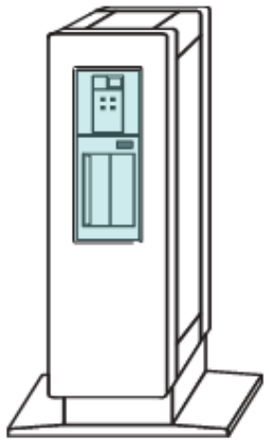
จากขั้นตอนการทำงานทั้งหมดที่กล่าวมาข้างต้น เราสามารถสรุปได้ว่า ฟังก์ชันการทำงานทั้งหมดจะดำเนินการจาก 2 แพลตฟอร์มเป็นอย่างน้อย ซึ่งก็คือ แหล่งข้อมูล/ระบบการดำเนินงาน และพื้นที่พักข้อมูล ดังแสดงในรูปที่ 4-3



MAINFRAME

UNIX

UNIX or OTHER



MINI

Data Extraction

Initial Reformatting/Merging

Preliminary Data Cleansing

Preliminary Data Cleansing

Transformation/ Consolidation

Validation/Quality Check

Load Image Creation

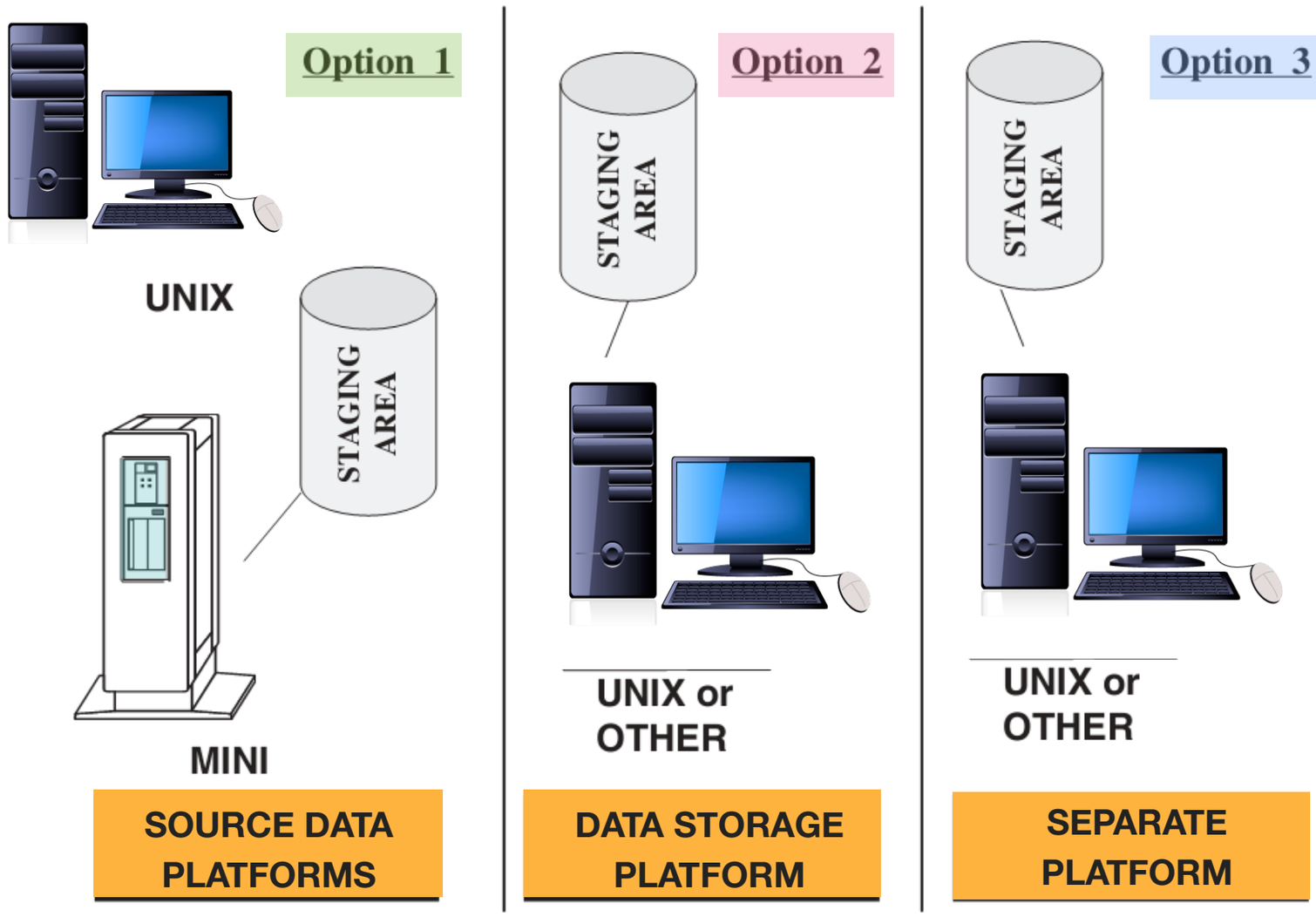
SOURCE DATA PLATFORMS

STAGING AREA PLATFORM

รูปที่ 4-3 แพลตฟอร์มสำหรับการได้มาซึ่งข้อมูล (Data acquisition)

การเลือกแพลตฟอร์มสำหรับพื้นที่พักข้อมูล

จะเป็นการเลือกการติดตั้งหรือการเลือกที่อยู่ของฟังก์ชันการทำงานต่างๆ ที่อยู่ในพื้นที่พักข้อมูลซึ่งมีทางเลือกหลักๆ อยู่ 3 ทางเลือก ด้วยกัน ดังแสดงในรูปที่ 4-4



รูปที่ 4-4 ทางเลือกแพลตฟอร์มสำหรับพื้นที่พักข้อมูล

กำหนดให้พื้นที่พักข้อมูลอยู่ที่แพลตฟอร์มเดียวกันกับแหล่งข้อมูล ซึ่งจะช่วยให้ลดเวลาและกระบวนการทำงานในการเคลื่อนย้ายข้อมูลจากแหล่งข้อมูลไปยังพื้นที่พักข้อมูลที่อยู่กันคนละแพลตฟอร์ม แต่การที่จะกำหนดให้พื้นที่พักข้อมูลอยู่ที่เดียวกันกับแหล่งข้อมูลหรือระบบการดำเนินงานได้นั้น เราจะต้องแน่ใจว่าทรัพยากรต่างๆ ของระบบการดำเนินงาน เช่น หน่วยประมวลผล หน่วยความจำ และพื้นที่ในดิสก์นั้นมีเพียงพอสำหรับการประมวลผลต่างๆ ในพื้นที่พักข้อมูล

กำหนดให้พื้นที่พักข้อมูลอยู่ที่แพลตฟอร์มเดียวกันกับฐานข้อมูลของคลังข้อมูล ซึ่งจะสามารถช่วยลดขั้นตอนการทำงานในการถ่ายโอนข้อมูลจากพื้นที่พักข้อมูลไปยังฐานข้อมูลได้ โดยเราสามารถจัดเก็บข้อมูลจากแฟ้มข้อมูลที่ถูกสกัดและประมวลผลต่างๆ (เปลี่ยนแปลง/เปลี่ยนรูปและรวบรวม/รวบยอดข้อมูลแล้ว) ไปยังฐานข้อมูลของคลังข้อมูลได้โดยตรง

กำหนดให้พื้นที่พักข้อมูลอยู่ในอีกแพลตฟอร์มหนึ่งๆ ที่เป็นอิสระจากแพลตฟอร์มอื่นๆ อาจจะเป็นแพลตฟอร์มที่เหมาะกับฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลที่มีความซับซ้อน และกระบวนการทำความสะอาดข้อมูลที่ประกอบไปด้วยการทำงานหลายขั้นตอน ซึ่งจากการกำหนดให้พื้นที่พักข้อมูลอยู่ในแพลตฟอร์มที่เป็นอิสระจะมีประโยชน์ต่างๆ ที่เห็นได้ชัดดังนี้

เราสามารถจัดการกับฟังก์ชันต่างๆ ที่มีการทำงานซับซ้อนได้โดยง่าย เช่น ฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล และการทำความสะอาดข้อมูลที่มีความซับซ้อน ซึ่งอาจจำเป็นต้องใช้เครื่องมือหรือซอร์ฟแวร์ต่างๆ เราสามารถติดตั้งเครื่องมือเหล่านั้นที่แพลตฟอร์มของพื้นที่พักข้อมูลได้โดยง่าย

เราสามารถจัดการกับข้อมูลที่ต้องจัดเก็บระหว่างการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล และการทำความสะอาดข้อมูลได้โดยง่าย โดยที่ในระหว่างการเปลี่ยนแปลงข้อมูลและการทำความสะอาดข้อมูล เราจำเป็นต้องเก็บข้อมูลเดิมและข้อมูลใหม่ที่มีการเปลี่ยนแปลงหรือถูกทำความสะอาดแล้วไว้ทั้งหมดเพื่อป้องกันการสูญหายหรือความผิดพลาดที่อาจจะเกิดขึ้นระหว่างการทำงาน รวมถึงเราอาจจะใช้แฟ้มข้อมูลหรือตารางสำหรับการตรวจสอบข้อมูลก็เป็นได้ ซึ่งการกระทำดังกล่าวจะเป็นการจัดการเกี่ยวกับการเคลื่อนที่หรือการเคลื่อนย้ายข้อมูล ซึ่งเราจะสามารถดำเนินการและจัดการได้โดยง่าย ถ้าเราทำการแยกพื้นที่พักข้อมูลไว้ในแพลตฟอร์มที่เป็นอิสระ

การเลือกวิธีการในการถ่ายโอนข้อมูลระหว่างแหล่งข้อมูลและพื้นที่พักข้อมูล

หลังจากที่เราทำการเลือกแพลตฟอร์มสำหรับพื้นที่พักข้อมูลซึ่งมีหลายๆ ครั้งที่เราเลือกที่จะกำหนดให้พื้นที่พักข้อมูลไม่ได้อยู่ในแพลตฟอร์มของแหล่งข้อมูล เราจะต้องพิจารณาถึงทางเลือกในการถ่ายโอนข้อมูลระหว่างแหล่งข้อมูลและพื้นที่พักข้อมูลซึ่งมีทางเลือกหลักๆ อยู่ 4 ทางเลือก

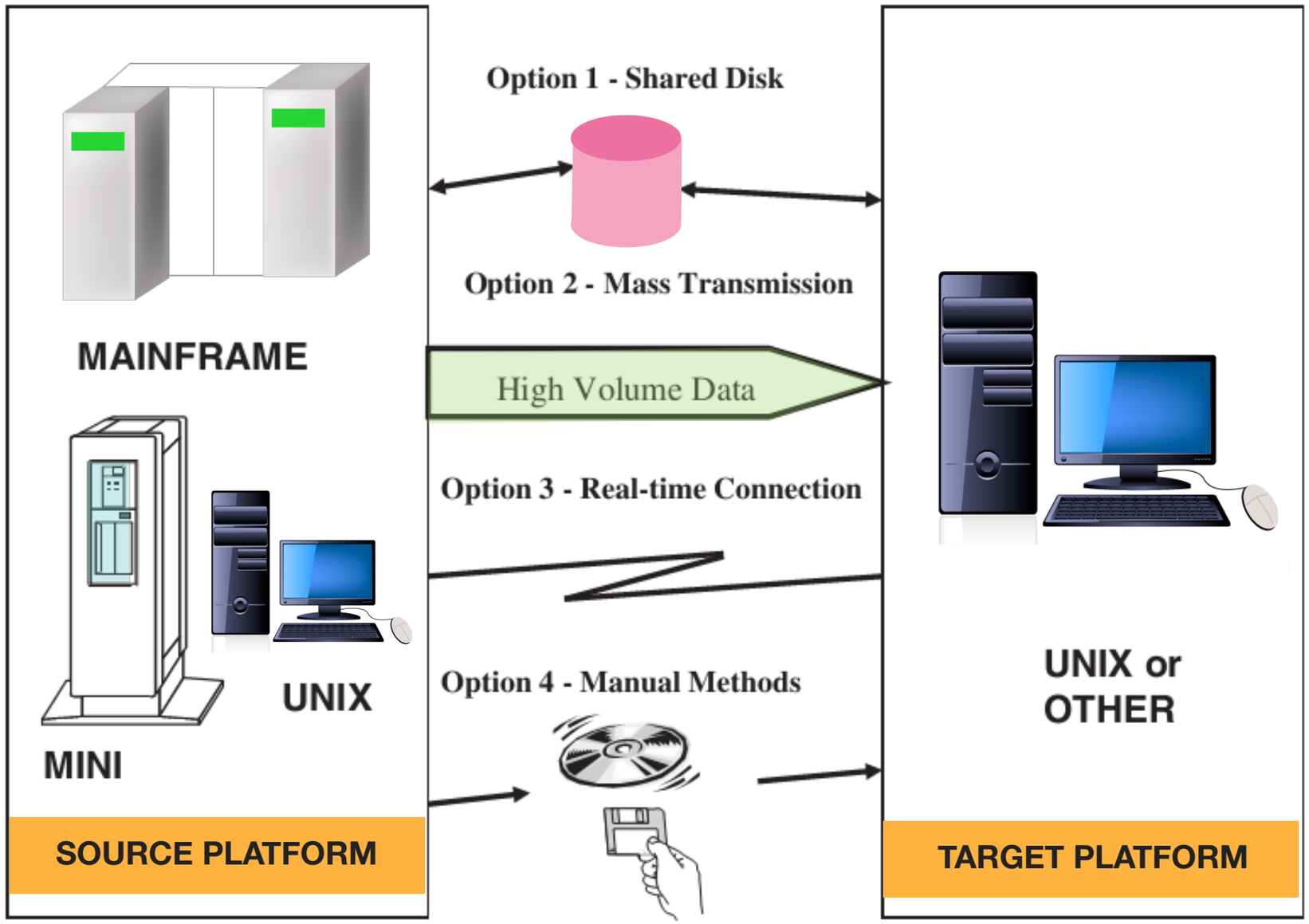
Option 1 Shared Disk

Option 2 Mass Transmission

Option 3 Real-time Connection

Option 4 Manual Methods

ดังแสดงในรูปที่ 4-5 ซึ่งมีรายละเอียดดังนี้



รูปที่ 4-5 ทางเลือกในการถ่ายโอนข้อมูลระหว่างแหล่งข้อมูลและ staging area

การกำหนดให้แหล่งข้อมูลและพื้นที่พักข้อมูลมีการใช้ดิสก์ด้วยกันบางส่วน (Shared disk) โดยการกำหนดให้ดิสก์หนึ่งๆ สามารถถูกเข้าถึงได้จากทั้งแพลตฟอร์มของแหล่งข้อมูลและแพลตฟอร์มของพื้นที่พักข้อมูล

การส่งผ่านข้อมูล (Data transmission) ข้ามแพลตฟอร์มโดยใช้พอร์ตในการส่งผ่าน โดยพอร์ตที่ใช้ในการส่งผ่านข้อมูลจะเป็นฮาร์ดแวร์ที่มีความสามารถในการเชื่อมต่อระหว่างแพลตฟอร์มและมีความสามารถในการถ่ายโอนข้อมูลในปริมาณที่มาก แต่ก่อนที่จะทำการถ่ายโอนข้อมูลแต่ละแพลตฟอร์มจะต้องถูกปรับแต่ง (configure) ให้รู้จักพอร์ตที่ใช้ในการส่งผ่านข้อมูล

การส่งผ่านข้อมูลแบบเรียลไทม์ (real time) ซึ่งจะต้องทำการเชื่อมต่อทั้ง 2 แพลตฟอร์มเข้าด้วยกันแบบเรียลไทม์ด้วย โดยที่หลังจากทำการเชื่อมต่อจะทำให้โปรแกรมที่อยู่ในแพลตฟอร์มหนึ่งสามารถใช้ทรัพยากรของอีกแพลตฟอร์มหนึ่งได้ เช่น การอ่านข้อมูลและการเขียนข้อมูลลงในอีกแพลตฟอร์มหนึ่งได้ วิธีการส่งผ่านข้อมูลแบบเรียลไทม์นั้นจะเหมาะกับคลังข้อมูลที่ต้องการข้อมูลแบบเรียลไทม์หรือใกล้ๆ เรียลไทม์

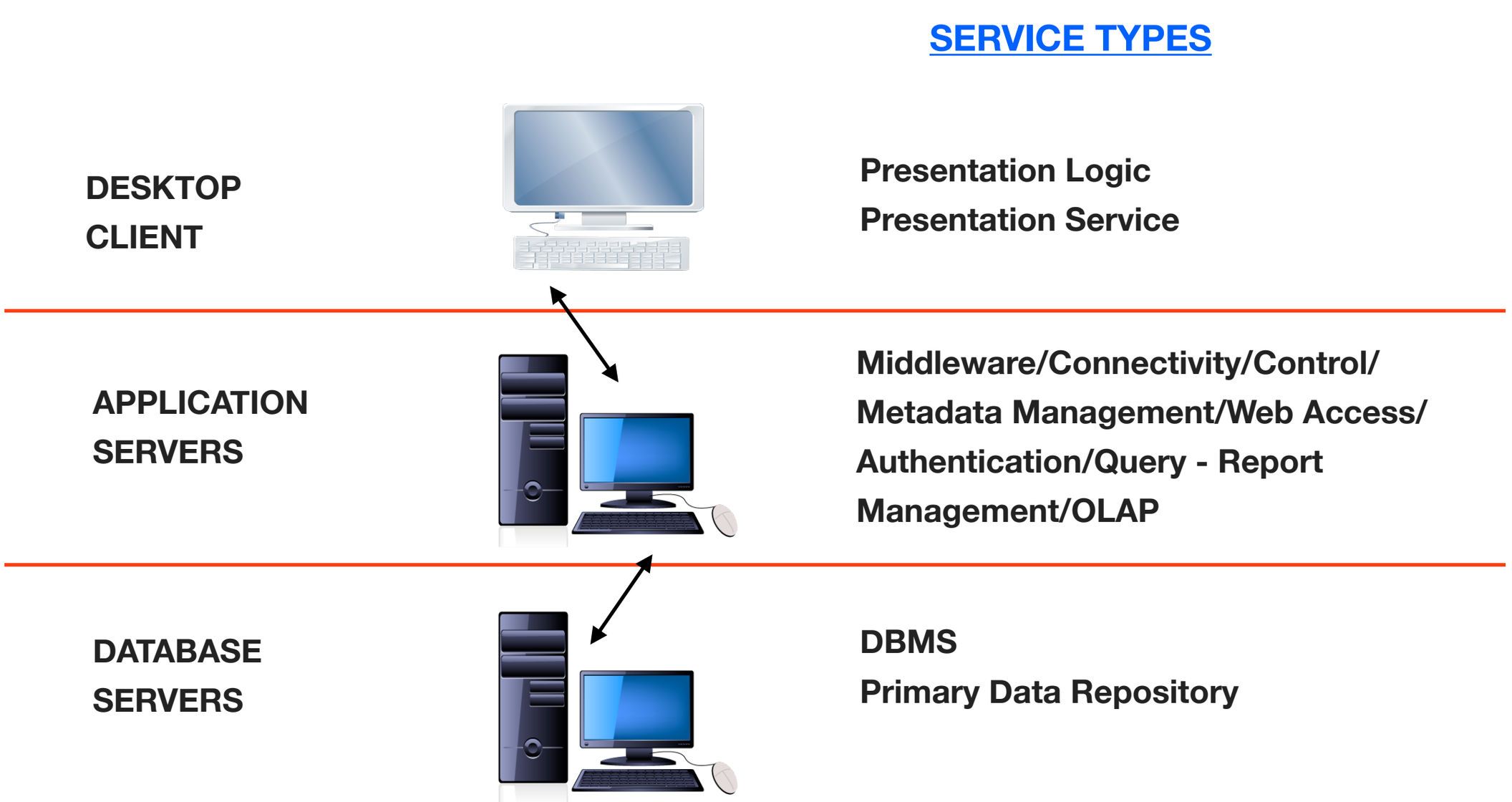
การส่งผ่านข้อมูลด้วยมือ (manual) โดยการเขียนข้อมูลที่ได้จากแพลตฟอร์มหนึ่งๆ ลงไปในเทป (tape) หรือ ดิสก์ (disk) จากนั้นทำการส่งผ่านเทปหรือดิสก์เหล่านั้นไปยังอีกแพลตฟอร์มหนึ่งเพื่อทำการอ่านข้อมูลและประมวลผลข้อมูลเหล่านั้นต่อไป

การเลือกสถาปัตยกรรมแบบ Client/server สำหรับคลังข้อมูล

แม้ว่าหลายๆคลังข้อมูลจะมีการใช้เมนเฟรมและมินิคอมพิวเตอร์ แต่ในปัจจุบันการสร้างคลังข้อมูลจะถูกสร้างโดยตั้งอยู่บนพื้นฐานของสถาปัตยกรรม Client/server ดังแสดงในรูปที่ 4-6 ที่จะใช้แพลตฟอร์มทั้งสิ้น 3 แพลตฟอร์มด้วยกัน คือ **client แอพลิเคชัน และฐานข้อมูล** ตามลำดับ

ซึ่งการแบ่งแยกแพลตฟอร์มในลักษณะนี้จะช่วยให้เราสามารถดำเนินการต่างๆ ได้มากมาย เช่น

- การสั่งให้ middleware ทำงาน และการสร้างการเชื่อมต่อต่าง ๆ
- การใช้ซอร์ฟแวร์ในการจัดการและการควบคุมต่างๆ
- การจัดการกับการเข้าถึงข้อมูลผ่านเว็บ
- การจัดการกับเมตาดาต้า
- การกำหนดสิทธิ์การใช้งานและการยืนยันตัวตนเพื่อใช้งาน
- การจัดการและดำเนินการกับรายงานที่เป็นมาตรฐาน
- การจัดการกับการประมวลผลคิวรีที่มีความซับซ้อน
- การใช้งาน OLAP เพื่อสร้างรายงานต่างๆ

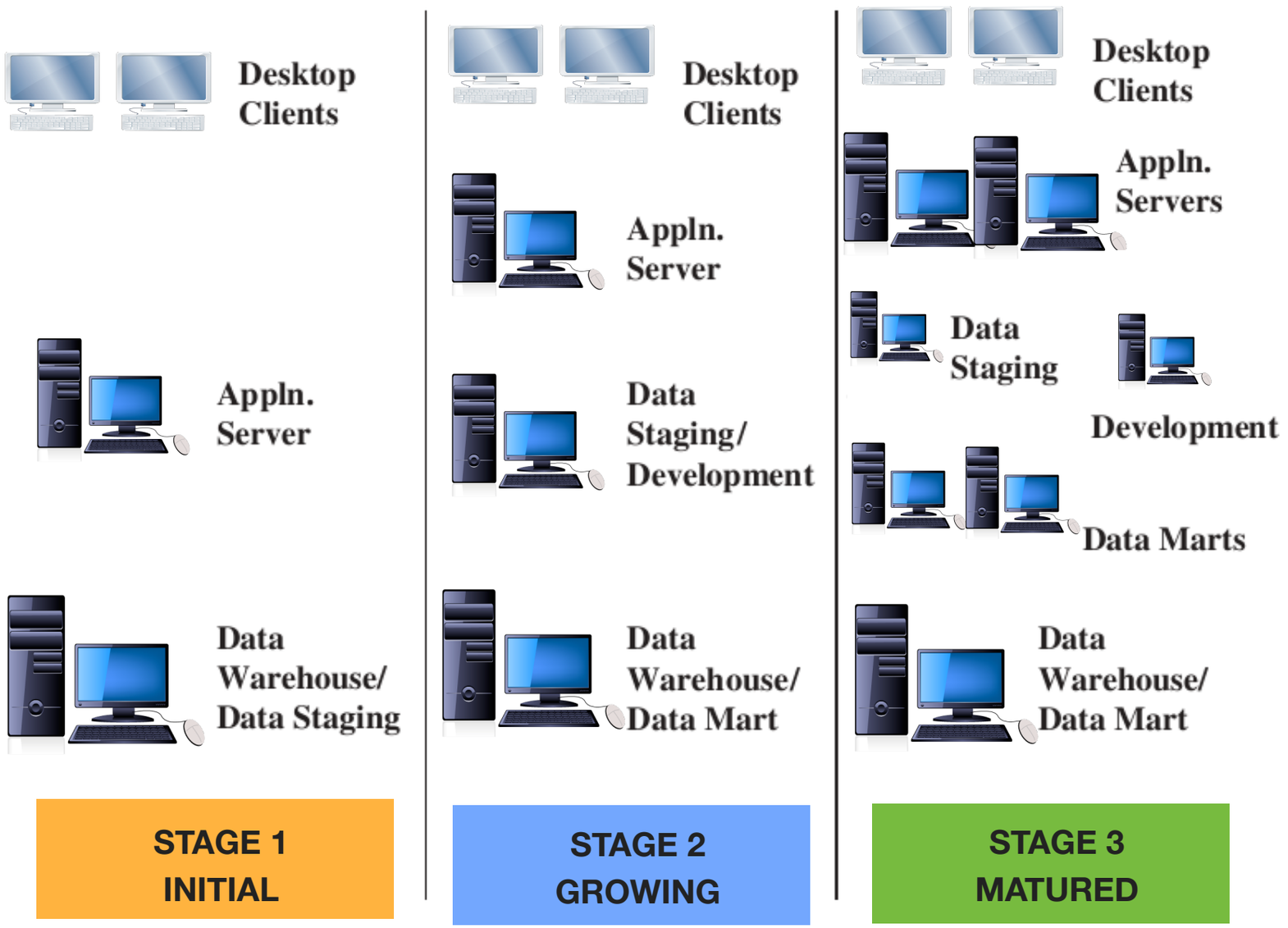


รูปที่ 4-6 สถาปัตยกรรม client/server สำหรับคลังข้อมูล

การเลือกแพลตฟอร์มเมื่อคลังข้อมูลเสร็จสมบูรณ์

จากทางเลือกก่อนหน้าจะเป็นการเลือกหรือการกำหนด
แพลตฟอร์มให้กับคลังข้อมูลก่อนการเริ่มทำงาน แต่อย่างไรก็ดี
เมื่อการสร้างคลังข้อมูลเสร็จสมบูรณ์และเริ่มใช้งาน แพลตฟอร์มของคลังข้อมูล
อาจจะมีการเปลี่ยนแปลงเกิดขึ้น ดังแสดงในรูปที่ 4-7 ที่แสดงถึงแพลตฟอร์ม
ในตอนเริ่มต้นที่พื้นที่พักข้อมูล และฐานข้อมูลของคลังข้อมูลที่อยู่ในแพลตฟอร์ม
เดียวกัน

ต่อมาเมื่อเวลาผ่านไป เมื่อมีผู้ใช้มากขึ้นและจำนวนคิวรีที่ต้องทำการ
ประมวลผลมากขึ้น แพลตฟอร์มอาจมีการเปลี่ยนแปลง โดยอาจจะ
ทำการแยกพื้นที่พักข้อมูลให้ไม่อยู่แพลตฟอร์มเดียวกันกับฐานข้อมูล
และเมื่อคลังข้อมูลเริ่มที่จะมั่นคงและคงตัว เราอาจจะแยกคลังข้อมูล
จากที่เป็นคลังข้อมูลของทั้งองค์กรออกเป็นดาต้ามาร์ทของแต่ละแผนก
แล้วทำการแยกแต่ละดาต้ามาร์ทไว้ในอีกแพลตฟอร์มหนึ่ง เป็นต้น



รูปที่ 4-7 การเลือกแพลตฟอร์มเมื่อคลังข้อมูลเสร็จสมบูรณ์

หลังจากที่เราทราบถึงการทำงานของคลังข้อมูลที่ต้องยุ่งเกี่ยวกับข้อมูลที่มีความซับซ้อนและมีจำนวนมาก ในการทำงานเราอาจจำเป็นต้องอาศัยการคำนวณแบบขนานที่จะใช้หลายๆโปรเซสเซอร์มาช่วยในการคำนวณ ซึ่งฮาร์ดแวร์ที่สนับสนุนการคำนวณแบบขนานจะมีสถาปัตยกรรมที่แตกต่างกัน ดังนั้นเราควรจะพิจารณาถึงสถาปัตยกรรมของการคำนวณแบบขนานในปัจจุบันที่ประกอบไปด้วย

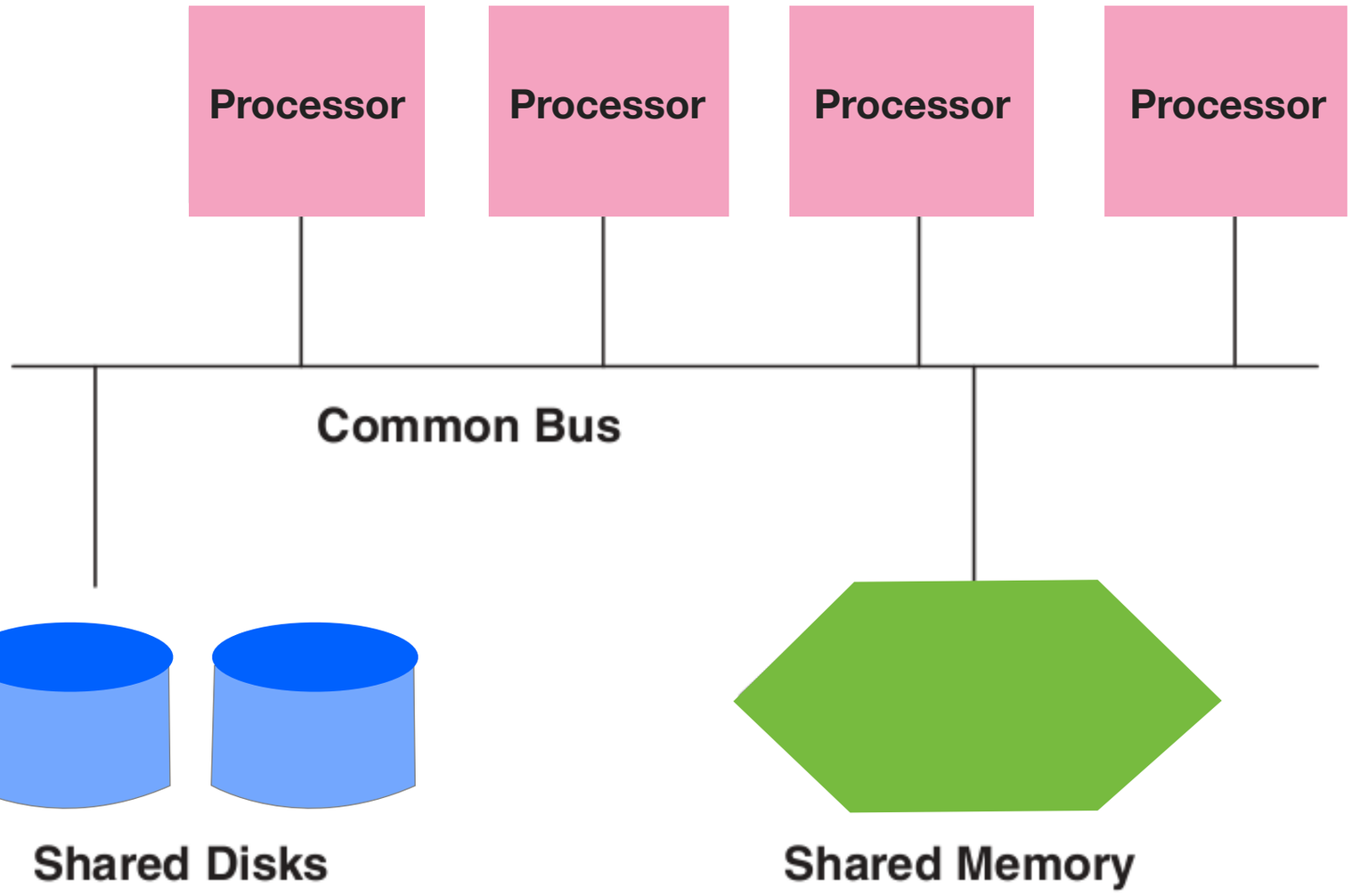
(1) SMP (Symmetric multiprocessing) (2) Clusters
(3) MPP (Massively parallel processing) และ (4) ccNUMA หรือ NUMA (Cache-coherent nonuniform memory architecture)

ดังแสดงในรูปที่ 4-8, 4-9, 4-10 และ 4-11 ตามลำดับ โดยในการพิจารณาแต่ละสถาปัตยกรรม เราควรจะพิจาณาคุณลักษณะ ประโยชน์ และข้อจำกัดต่างๆ ดังนี้

SMP

(Symmetric multiprocessing)

1 SMP (Symmetric multiprocessing)



รูปที่ 4-8 ตัวอย่างสถาปัตยกรรม SMP

คุณลักษณะ

- จะเป็นสถาปัตยกรรมที่ง่ายที่สุด ที่แต่ละโปรเซสเซอร์จะใช้งานหน่วยความจำหลักและดิสก์ร่วมกัน
- แต่ละโปรเซสเซอร์จะสามารถ ใช้งานหน่วยความจำหลักได้ทั้งหมดได้โดยผ่านบัส (bus)
- โปรเซสเซอร์จะสามารถติดต่อสื่อสารกันได้ผ่านทางหน่วยความจำหลัก
- ทุกโปรเซสเซอร์จะสามารถเข้าถึง disk controller ได้

ประโยชน์

- เป็นเทคโนโลยีที่ถูกใช้มาตั้งแต่ยุค 1970s และได้มีการพิสูจน์ประสิทธิภาพแล้ว
- มีความสามารถทำงานพร้อมๆ กันได้หลายๆ งาน ซึ่งจะสามารถทำการรันคิวรีพร้อมๆ กันได้
- สามารถปรับสมดุลของภาระงานของแต่ละโปรเซสเซอร์ได้ค่อนข้างดี
- สามารถต่อเติมยืดขยายได้ โดยที่เราสามารถเพิ่มโปรเซสเซอร์ได้โดยทำการเชื่อมต่อกับบัส
- ทำให้เราสามารถดูแลและจัดการกับเซิร์ฟเวอร์ได้ค่อนข้างง่าย

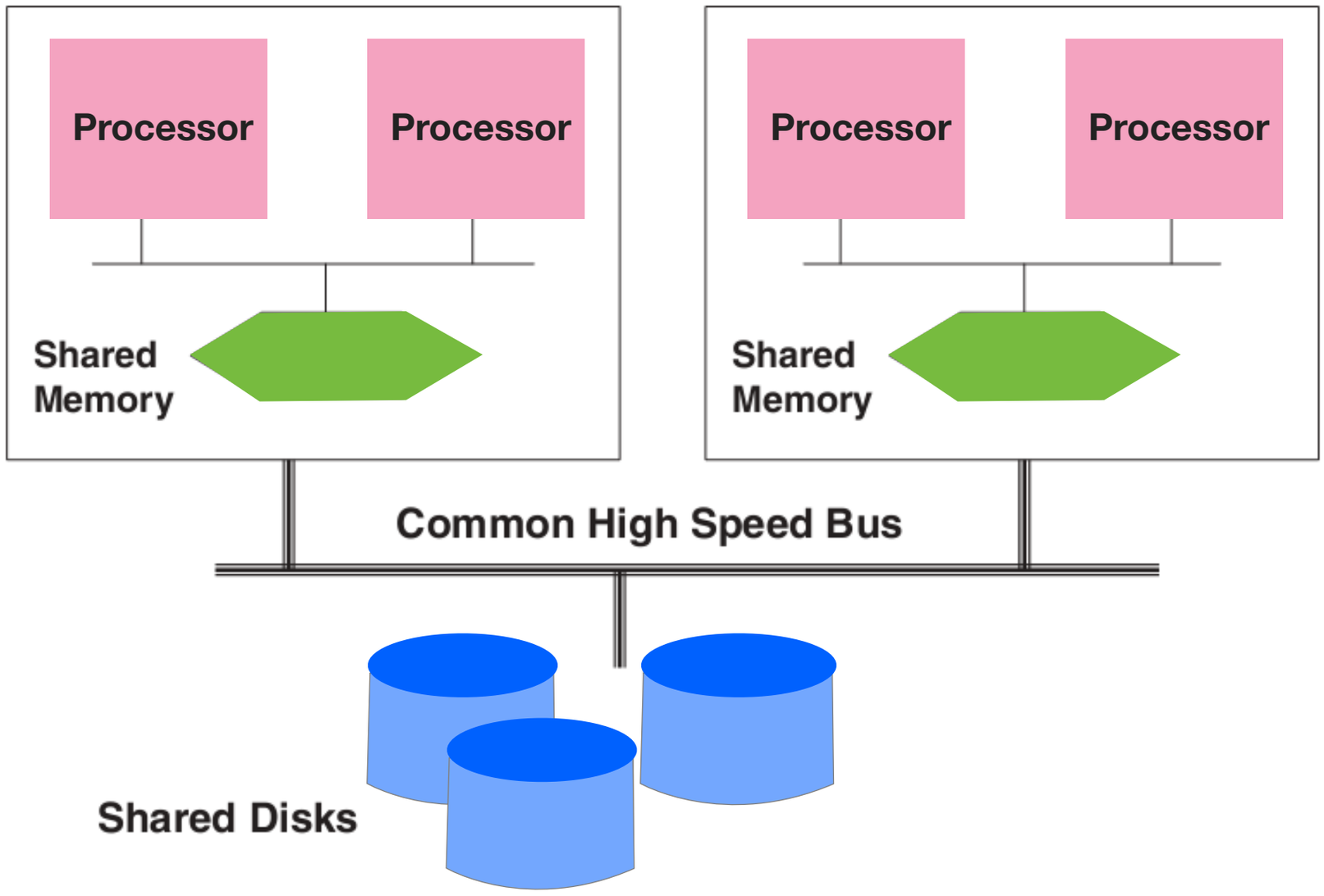
ข้อจำกัด

- หน่วยความจำจะค่อนข้างจำกัด เนื่องจากมีอยู่ที่เดียวและถูกใช้โดยโปรเซสเซอร์ทั้งหมด
- ประสิทธิภาพการทำงานอาจจะถูกจำกัดโดย bandwidth ตัวอย่างเช่น การติดต่อสื่อสารระหว่างโปรเซสเซอร์ และ I/O เป็นต้น

จากที่กล่าวข้างต้น สถาปัตยกรรม SMP จะเป็นตัวเลือกที่น่าสนใจกับคลังข้อมูลที่มีข้อมูลไม่มากประมาณ 200-300 GB ที่อาจต้องการทำงานหลายงานพร้อมกัน


2
Clusters

2 Clusters



รูปที่ 4-9 ตัวอย่างสถาปัตยกรรม Cluster

คุณลักษณะ

- กลุ่มของโปรเซสเซอร์จะถูกแบ่งเป็นกลุ่มๆ ที่เรียกว่าโหนด โดยที่แต่ละโหนดจะมีโปรเซสเซอร์อย่างน้อย 1 หรือมากกว่านั้น และแต่ละโหนดจะมีหน่วยความจำเป็นของตนเอง
- หน่วยความจำในแต่ละโหนดถูกใช้โดยโปรเซสเซอร์ในโหนดนั้นๆ เท่านั้น โดยสามารถใช้ร่วมกันได้ แต่จะไม่มีการหน่วยความจำร่วมกันระหว่างโปรเซสเซอร์ที่ไม่ได้อยู่ในโหนดเดียวกัน
- โปรเซสเซอร์จะสามารถติดต่อสื่อสารกันได้ผ่านบัสความเร็วสูง
- แต่ละโหนดจะมีการใช้ดิสก์ร่วมกัน
- สถาปัตยกรรมนี้จะมีลักษณะเป็น cluster ของโหนดต่างๆ

ประโยชน์

- เป็นสถาปัตยกรรมที่มีประโยชน์สูง เนื่องจากเราสามารถเข้าถึงข้อมูลทั้งหมดได้ตลอดแม้ว่าจะมีโหนดใดโหนดหนึ่งเสียหาย
- ยังคงรักษาการจัดเก็บข้อมูลไว้ที่เดียว
- เป็นทางเลือกที่ดีสำหรับการยืดขยายต่อเติม ซึ่งเราสามารถเพิ่มโหนดเข้าไปได้

ข้อจำกัด

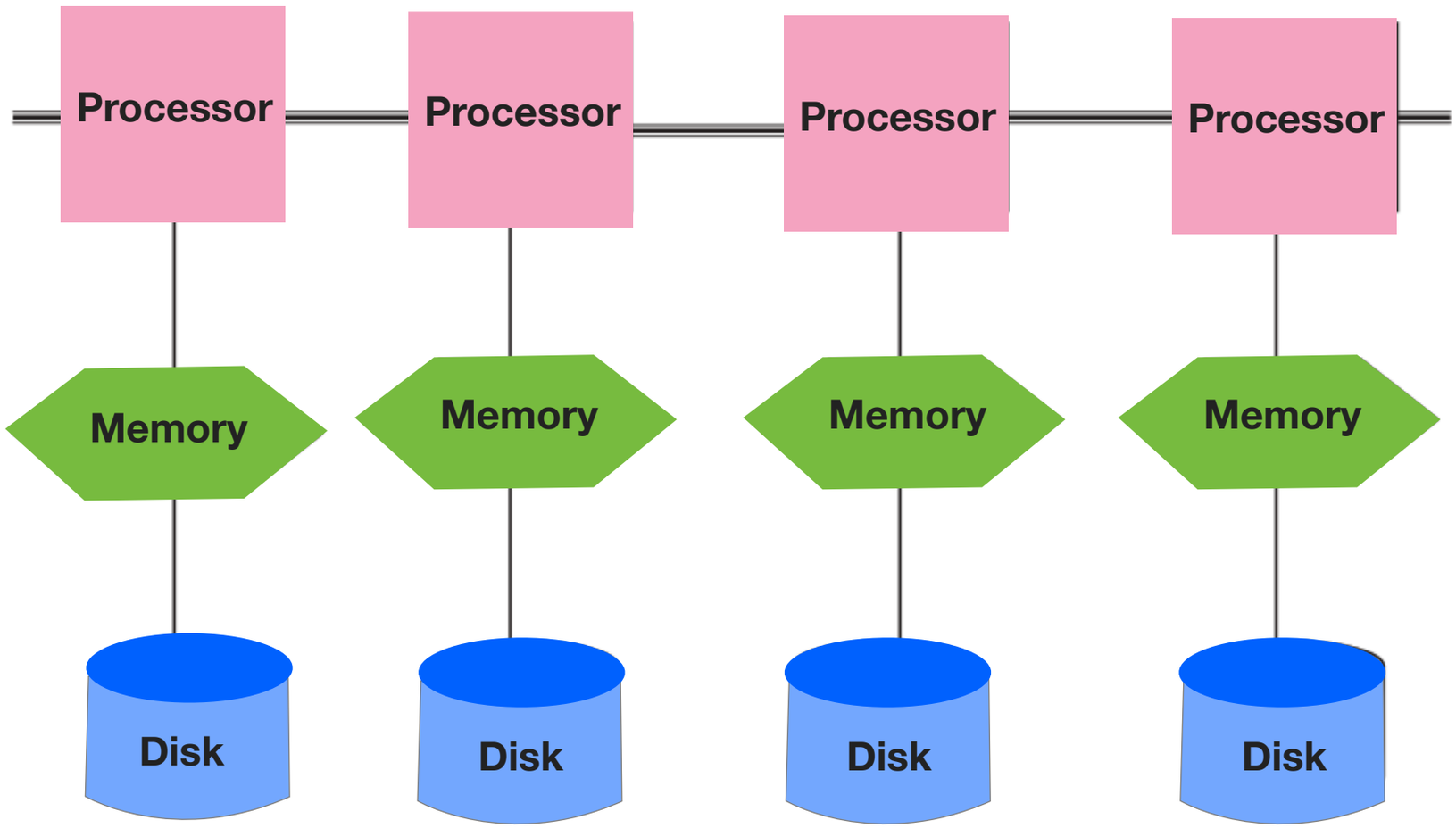
- Banwidth ของบัสอาจจะเป็นตัวจำกัดความสามารถในการต่อเติมเพิ่มขยายโหนดใหม่ ๆ ให้กับระบบได้
- จะมีค่าใช้จ่าย (overhead) เกี่ยวกับระบบปฏิบัติการค่อนข้างสูง
- แต่ละโหนดจะมีหน่วยความจำ (เรียกว่า data cache) ดังนั้นเราจะต้องดูแลเรื่องความสอดคล้องของ cache ในการ synchornize กันระหว่างโหนดด้วย

จากข้างต้น สถาปัตยกรรมแบบ Cluster จะเป็นทางเลือกที่ดีสำหรับคลังข้อมูลที่คาดหวังว่าจะค่อยๆ เติบโตขึ้นเรื่อย ๆ

MPP

(Massively parallel processing)

3 MPP (Massively parallel processing)



รูปที่ 4-10 ตัวอย่างสถาปัตยกรรม MPP

คุณลักษณะ

- เป็นสถาปัตยกรรมที่ไม่มีการใช้อุปกรณ์ใด ๆ ร่วมกัน
- เป็นสถาปัตยกรรมที่ให้ความใส่ใจกับการเรียกใช้ข้อมูลจากหน่วยความจำและดิสก์ โดยการกำหนดให้แต่ละโปรเซสเซอร์มีหน่วยความจำและดิสก์เป็นของตนเองไม่ต้องยุ่งเกี่ยวกับโปรเซสเซอร์อื่นๆ
- สามารถทำงานได้ดีกับระบบปฏิบัติการที่สนับสนุนการเข้าถึงดิสก์โดยตรง
- การติดต่อสื่อสารจะสามารถติดต่อได้ผ่านทางโปรเซสเซอร์

ประโยชน์

- เป็นสถาปัตยกรรมที่มีความสามารถในการยืดขยายต่อเติมสูง (highly scalable)
- สามารถเข้าถึงข้อมูลได้อย่างรวดเร็ว
- ค่าใช้จ่ายต่อ 1 โหนดจะค่อนข้างต่ำ

ข้อจำกัด

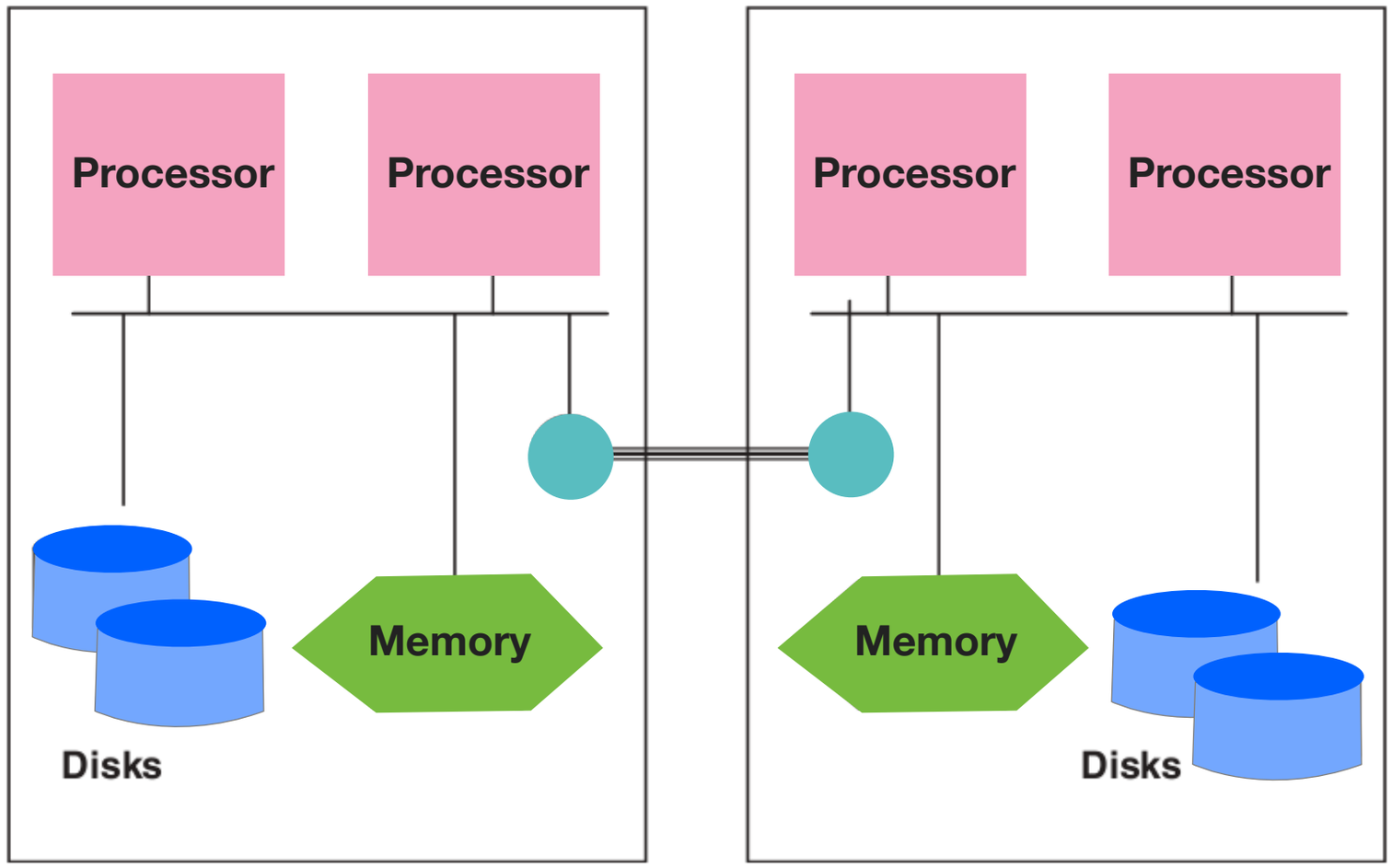
- ต้องทำการแบ่งส่วนข้อมูลอย่างชัดเจน
- การเข้าถึงข้อมูลเป็นไปอย่างจำกัด
- มีข้อจำกัดในเรื่องของการปรับสมดุลของงานในแต่ละโปรเซสเซอร์
- ต้องทำการดูแลรักษาเกี่ยวกับ cache consistency

จากข้างต้น สถาปัตยกรรม MPP จะเป็นทางเลือกที่ดีสำหรับการสร้างคลังข้อมูลขนาดกลางหรือขนาดใหญ่ที่มีข้อมูลประมาณ 400-500 GB แต่สำหรับคลังข้อมูลที่มีขนาด 1 เทราไบต์ขึ้นไป เราจะต้องมองหาสถาปัตยกรรมอื่นที่ผสมผสานข้อดีของสถาปัตยกรรมข้างต้น

ccNUMA หรือ NUMA

(Cache-coherent nonuniform memory architecture)

4 ccNUMA หรือ NUMA (Cache-coherent nonuniform memory architecture)



รูปที่ 4-11 ตัวอย่างสถาปัตยกรรม NUMA

คุณลักษณะ

- เป็นสถาปัตยกรรมใหม่ที่สร้างขึ้นในช่วงทศวรรษ 1990s
- NUMA จะมีสถาปัตยกรรมคล้ายกับ SMP ที่แตกส่วนออกเป็น SMP ย่อยๆ ที่มีการเชื่อมต่อกัน
- ในระบบจะมีหน่วยความจำจริงๆอยู่ที่เดียว แต่ในแต่ละโหนดจะมีโควต้าการใช้หน่วยความจำเป็นของตัวเอง ซึ่งแต่ละโหนดจะมีไดเรกทอรีของ memory address สำหรับโหนดนั้นๆเก็บอยู่ด้วย
- เวลาที่ใช้ในการเข้าถึงข้อมูลในหน่วยความจำจะค่อนข้างหลากหลายเนื่องจากโหนดแรกอาจจะต้องการข้อมูลที่ถูกเก็บอยู่ในพื้นที่หน่วยความจำของโหนดที่ 3 ก็เป็นได้ ซึ่งนี่คือเหตุผลที่เราเรียกสถาปัตยกรรมนี้ว่าเป็น nonuniform memory access architecture

ประโยชน์

- มีความยืดหยุ่นสูงสุด
- แก้ไขปัญหาเกี่ยวกับหน่วยความจำของ SMP
- มีความสามารถในการยืดขยายต่อเติมได้มากกว่า SMP
- สามารถใส่ OLAP ไว้ในเซิร์ฟเวอร์นี้ได้

ข้อจำกัด

- การเขียนโปรแกรมเพื่อจัดการสิ่งต่างๆ ในสถาปัตยกรรม NUMA จะมีความซับซ้อนมากกว่าสถาปัตยกรรมอื่นๆ
- ซอร์ฟแวร์ที่สนับสนุน NUMA นั้นมีค่อนข้างน้อย

จากข้างต้น สถาปัตยกรรม NUMA จะเป็นทางเลือกที่ดีสำหรับคุณ โดยที่คุณอาจจะเลือกใช้ NUMA ที่มี 1 หรือ 2 โหนด แต่ถ้าบริษัทของคุณยังไม่มีประสบการณ์ทางด้านฮาร์ดแวร์เลย ทางเลือกนี้อาจจะยุ่งยากและซับซ้อนเกินไป

SECTION 5

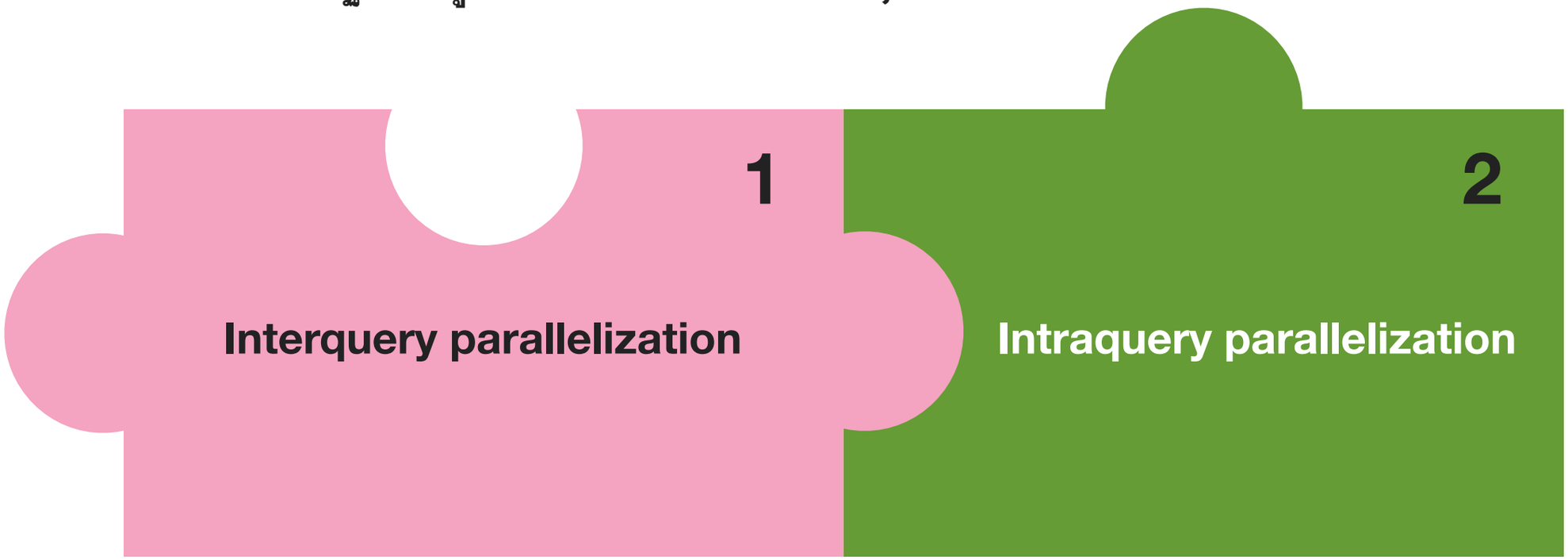
ซอฟต์แวร์ระบบฐานข้อมูล

RDBMS

ในปัจจุบันซอร์ฟแวร์ระบบฐานข้อมูลได้ถูกพัฒนาอย่างต่อเนื่อง หลายๆ ซอร์ฟแวร์ของ RDBMS ได้มีการเพิ่มเติมฟังก์ชันการได้มาซึ่งข้อมูล (data acquisition) สำหรับคลังข้อมูล ซึ่งจะทำให้การเข้าถึงข้อมูลหรือการถ่ายโอนข้อมูลจากระบบการดำเนินงานไปยังคลังข้อมูลสามารถทำงานได้ง่ายขึ้น ในหลายๆ ซอร์ฟแวร์ได้มีการเพิ่มฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล (data transformation) อีกด้วย แต่อย่างไรก็ดี นอกเหนือจากฟังกัชันที่ถูกเพิ่มขึ้นมาในซอร์ฟแวร์ระบบฐานข้อมูลแล้ว การปรับสมดุลของภาระงาน (load balancing) และประสิทธิภาพของการประมวลผลคิวรี (query performance) ก็เป็นสิ่งสำคัญมากสำหรับคลังข้อมูล เนื่องจากการทำงานของคลังข้อมูลจะเน้นที่การประมวลผลคิวรีเป็นหลัก ซึ่งจะเน้นหนักไปที่การปรับปรุงประสิทธิภาพการประมวลผลคิวรีให้มีประสิทธิภาพมากที่สุดเท่าที่จะเป็นไปได้ ในการที่จะพัฒนาการประมวลผลคิวรีได้อย่างมีประสิทธิภาพนั้น เราอาจเลือกทำการประมวลผลคิวรีแบบขนานเพื่อเพิ่มประสิทธิภาพในการประมวลผลที่ซึ่งจะเป็นการใช้ประโยชน์จากฮาร์ดแวร์ที่รองรับการคำนวณแบบขนานได้อย่างเต็มที่

การประมวลผลแบบขนานนั้นเป็นทางเลือกหนึ่งสำหรับฮาร์ดแวร์ที่มีหลายโปรเซสเซอร์ ซึ่งในปัจจุบันซอร์ฟแวร์ระบบฐานข้อมูลโดยส่วนใหญ่จะสามารถรองรับการประมวลผลคิวรีแบบขนานได้ โดยระบบฐานข้อมูลจะทำการแบ่งงานออกเป็นหลายๆ ส่วนแล้วแจกจ่ายให้กับแต่ละโปรเซสเซอร์เพื่อทำการประมวลผล ซึ่งในการแบ่งงานและแจกจ่ายงานนั้นระบบฐานข้อมูลจะต้องพิจารณาถึงความสมดุลของภาระงานที่แต่ละโปรเซสเซอร์จะได้รับด้วย เมื่อแต่ละโปรเซสเซอร์ทำงานที่ได้รับมอบหมายเสร็จสิ้น ระบบฐานข้อมูลจะต้องทำการรวบรวมผลลัพธ์จากที่มีอยู่หลายๆ ส่วนเข้าด้วยกัน เพื่อคืนค่าผลลัพธ์ให้กับผู้ใช้ต่อไป

การประมวลผลแบบขนานในระบบฐานข้อมูลจะมีด้วยกัน 2 ประเภทหลักๆ คือ



ที่จะมีข้อดีข้อเสีย และรายละเอียดการทำงานแตกต่างกัน ที่ซึ่งสามารถแสดงรายละเอียดได้ดังนี้

1

Interquery parallelization

Interquery parallelization

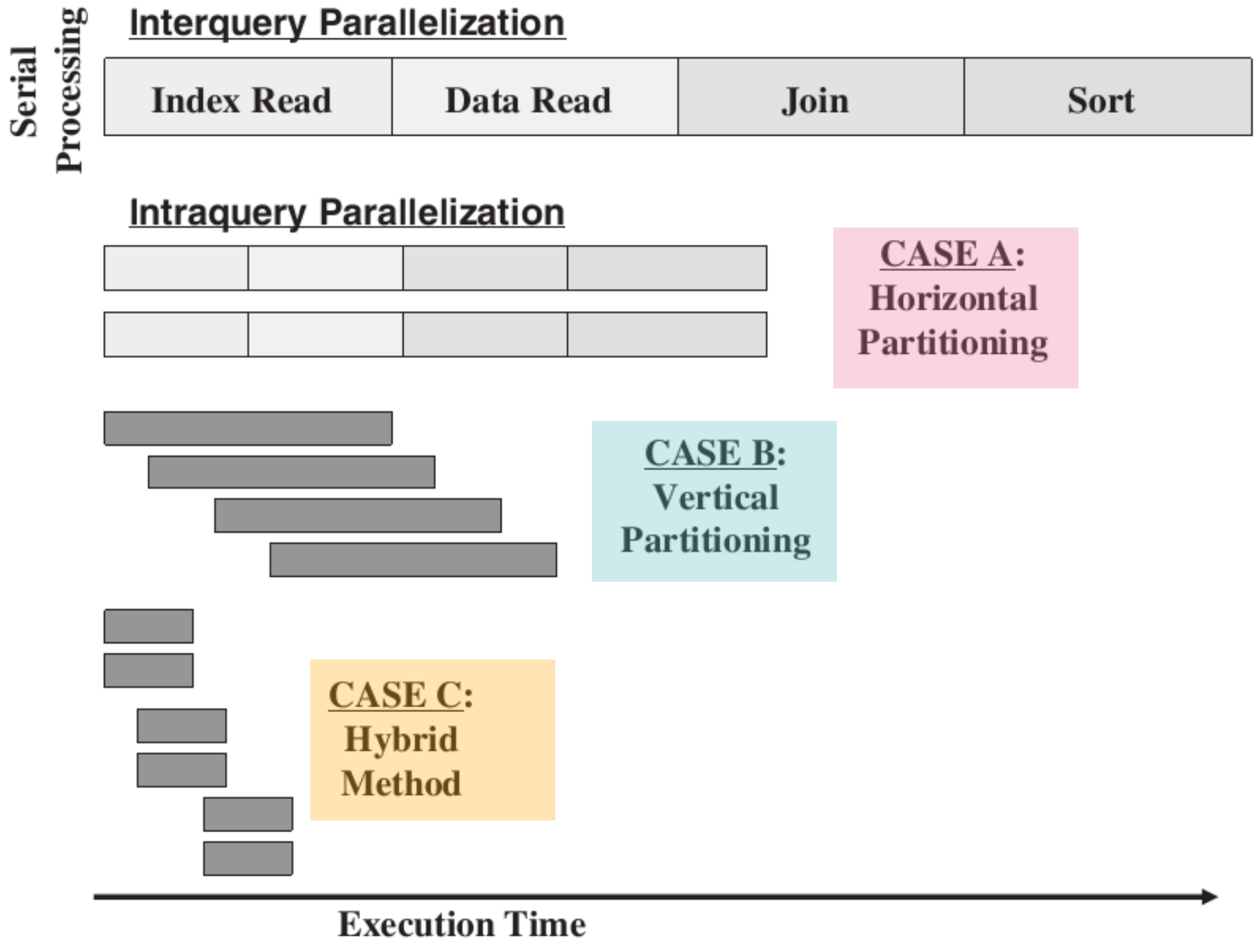
จะเป็นการประมวลผลคิวรีหลายๆ คิวรีพร้อมๆ กันแบบขนานโดยใช้หลายโปรเซสเซอรทำงานพร้อมๆ กัน โดยที่จำนวนคิวรีที่จะทำการประมวลผลพร้อมกันอาจจะเป็นจำนวนที่เรากำหนดไว้หรืออาจจะเท่ากับจำนวนโปรเซสเซอร์ที่ว่างงานอยู่ในขณะนั้นก็เป็นได้ แต่อย่างไรก็ดีประสิทธิภาพการทำงานของ interquery parallelization จะมีข้อจำกัดตรงที่--ถึงแม้ว่าหลายๆ คิวรีจะถูกประมวลผลพร้อมๆกัน แต่ในการประมวลผลคิวรีหนึ่งๆยังคงต้องทำการคำนวณแบบตามลำดับ โดยทำการประมวลผลคิวรีหนึ่งๆ บนโปรเซสเซอร์เพียงตัวเดียวเท่านั้น สมมติว่าคิวรีหนึ่งๆ จะประกอบไปด้วยขั้นตอน index read, data read, data join และ data sort ตามลำดับ ซึ่งจากลำดับการทำงานดังกล่าว การทำงานของแต่ละขั้นตอนจะเริ่มต้นขึ้นได้ก็ต่อเมื่อขั้นตอนก่อนหน้าสิ้นสุดลงเท่านั้น ดังแสดงตัวอย่างในส่วนบนของรูปที่ 4-12

2

Intraquery parallelization

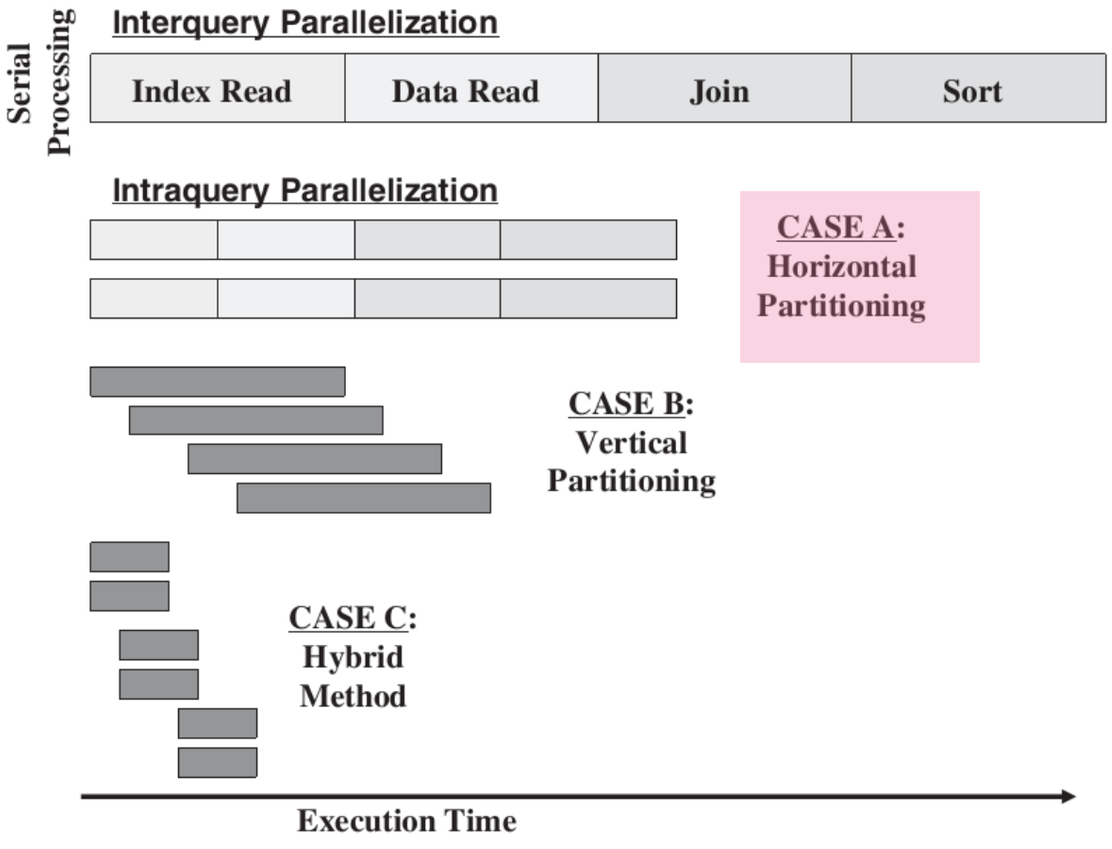
Intraquery parallelization

จะเป็นการประมวลผลคิวรีหนึ่งๆ ที่ทำการแบ่งแยกขั้นตอนต่างๆออกจากกันแล้วทำการคำนวณหรือประมวลผลแบบขนาน ตัวอย่างเช่นการประมวลผลคิวรีหนึ่งๆ จะประกอบด้วยการทำงาน index read, data read, data join และ data sort ถ้าเราทำการประมวลผลแบบ intraquery parallelization ขั้นตอนต่างๆ จะถูกทำงานพร้อมๆกันโดยหลายๆโปรเซสเซอร์ ซึ่งโดยปกติแล้ว intraquery parallelization จะสามารถแบ่งการทำงานได้เป็น 3 วิธีหลัก ดังแสดงในรูปที่ 4-12



รูปที่ 4-12 การทำ interqueryและ intraquery parallelization

1 Horizontal parallelism

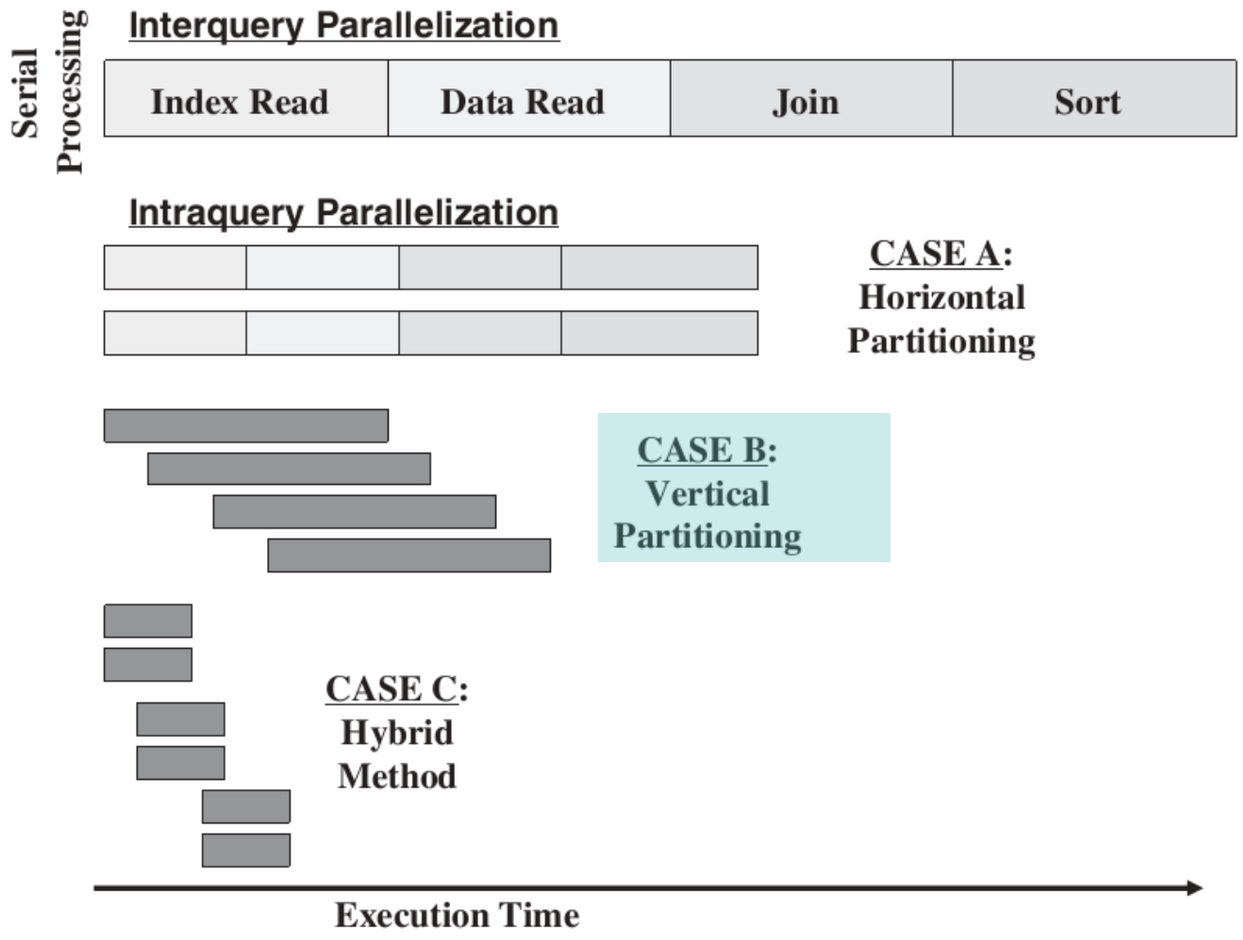


รูปที่ 4-12 การทำ interqueryและ intraquery parallelization

จะเป็นการคำนวณแบบขนานกับข้อมูลที่มีการแบ่งส่วนซึ่งถูกจัดเก็บอยู่ในหลายดิสก์ โดยที่การคำนวณแบบขนานจะเกิดขึ้นที่ขั้นตอนการทำงานหนึ่งๆ ของคิวรีหนึ่งๆ ตัวอย่างเช่น เมื่อคิวรีหนึ่งต้องทำการอ่านข้อมูล (data read) จากหลายดิสก์ Horizontal parallelism จะทำการกำหนดให้แต่ละโปรเซสเซอร์ทำการอ่านข้อมูลแต่ละดิสก์พร้อมๆ กัน หลังจากทำการอ่านข้อมูลเสร็จแล้วจะเปลี่ยนการทำงานไปยังขั้นตอนถัดไป

แต่อย่างไรก็ดี การที่จะกำหนดให้แต่ละโปรเซสเซอร์ทำการอ่านข้อมูลในแต่ละดิสก์ ถ้าแต่ละดิสก์มีข้อมูลไม่เท่ากันจะทำให้บางโปรเซสเซอร์ที่ทำการอ่านข้อมูลจำนวนน้อยต้องเสียเวลาในการรอให้โปรเซสเซอร์ที่ต้องทำการอ่านข้อมูลจำนวนมากทำการอ่านข้อมูลจนหมดเสียก่อน จึงค่อยเริ่มการทำงานขั้นตอนต่อไป

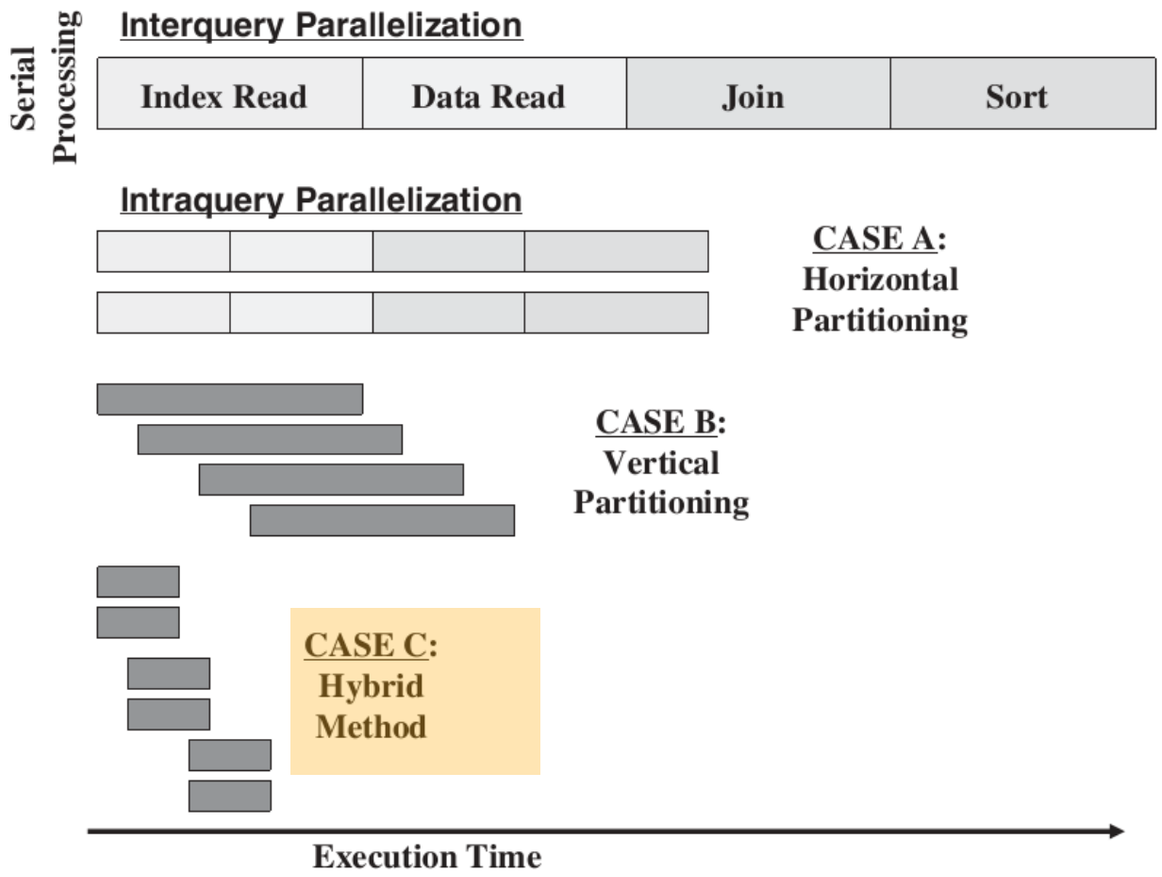
2 Vertical parallelism



รูปที่ 4-12 การทำ interqueryและ intraquery parallelization

จะเป็นการคำนวณแบบขนานกับหลายๆงานด้วยกัน เช่น ในคิวรีหนึ่งๆ จะประกอบไปด้วยหลายขั้นตอนการทำงาน และขั้นตอนการทำงานแหล่านั้นจะถูกคำนวณพร้อมๆกันในลักษณะที่เป็นแบบ pipeline เมื่อเราทำการอ่านข้อมูลเสร็จแล้วหนึ่งขั้นตอน จะทำการส่งผลลัพธ์ที่ได้จากการอ่านข้อมูลไปยังขั้นตอนการทำงานถัดไปทันที การทำงานด้วยวิธีนี้จะช่วยให้สามารถลดเวลาในการรอให้ขั้นตอนหนึ่งๆทำงานจนเสร็จสิ้นจึงค่อยเริ่มการทำงานขั้นตอนถัดไป แต่การที่จะคำนวณในลักษณะนี้ได้ DBMS จะต้องมีความสามารถในการแบ่งขั้นตอนการทำงานและส่งผ่านข้อมูลระหว่างขั้นตอนการทำงานค่อนข้างมาก ดังแสดงในตัวอย่าง B ในรูปที่ 4-12

3 Hybrid method



รูปที่ 4-12 การทำ interqueryและ intraquery parallelization

จะเป็นการผสมผสานระหว่าง horizontal และ vertical parallelism โดยการแบ่งการประมวลผลทั้งในแบบ horizontal และ vertical ซึ่งจะทำให้เราสามารถใช้ทรัพยากรได้อย่างคุ้มค่าที่สุด มีประสิทธิภาพในการคำนวณสูงที่สุดและมีความยืดหยุ่น สามารถยืดขยายต่อเติมได้ ดังแสดงในตัวอย่าง C ในรูปที่ 4-12

หลังจากที่เราทราบถึงสถาปัตยกรรมของเซิร์ฟเวอร์ที่เราจะเลือกใช้ และทางเลือกในการประมวลผลคิวรีแบบขนานแล้ว สิ่งเหล่านี้จะเป็นสิ่งที่ช่วยให้เราตัดสินใจในการเลือกใช้ระบบจัดการฐานข้อมูลซึ่งในทางปฏิบัติเราอาจจะเลือกฮาร์ดแวร์ที่เกี่ยวข้องกับเซิร์ฟเวอร์ที่มีความสามารถในการคำนวณแบบขนานก่อน จากนั้นเราค่อยทำการเลือกระบบจัดการฐานข้อมูลที่เหมาะกับเซิร์ฟเวอร์ที่เราเลือก โดยที่เราจะต้องพิจารณาถึงการปรับความสมดุลของภาระงานที่แต่ละโปรเซสเซอร์จะต้องรับผิดชอบ (load balancing) และทางเลือกในการคำนวณแบบขนาน (parallel processing options) นอกจากนี้เราจะต้องทำการพิจารณาปัจจัยสิ่งต่างๆเหล่านี้ประกอบการเลือกระบบจัดการฐานข้อมูล

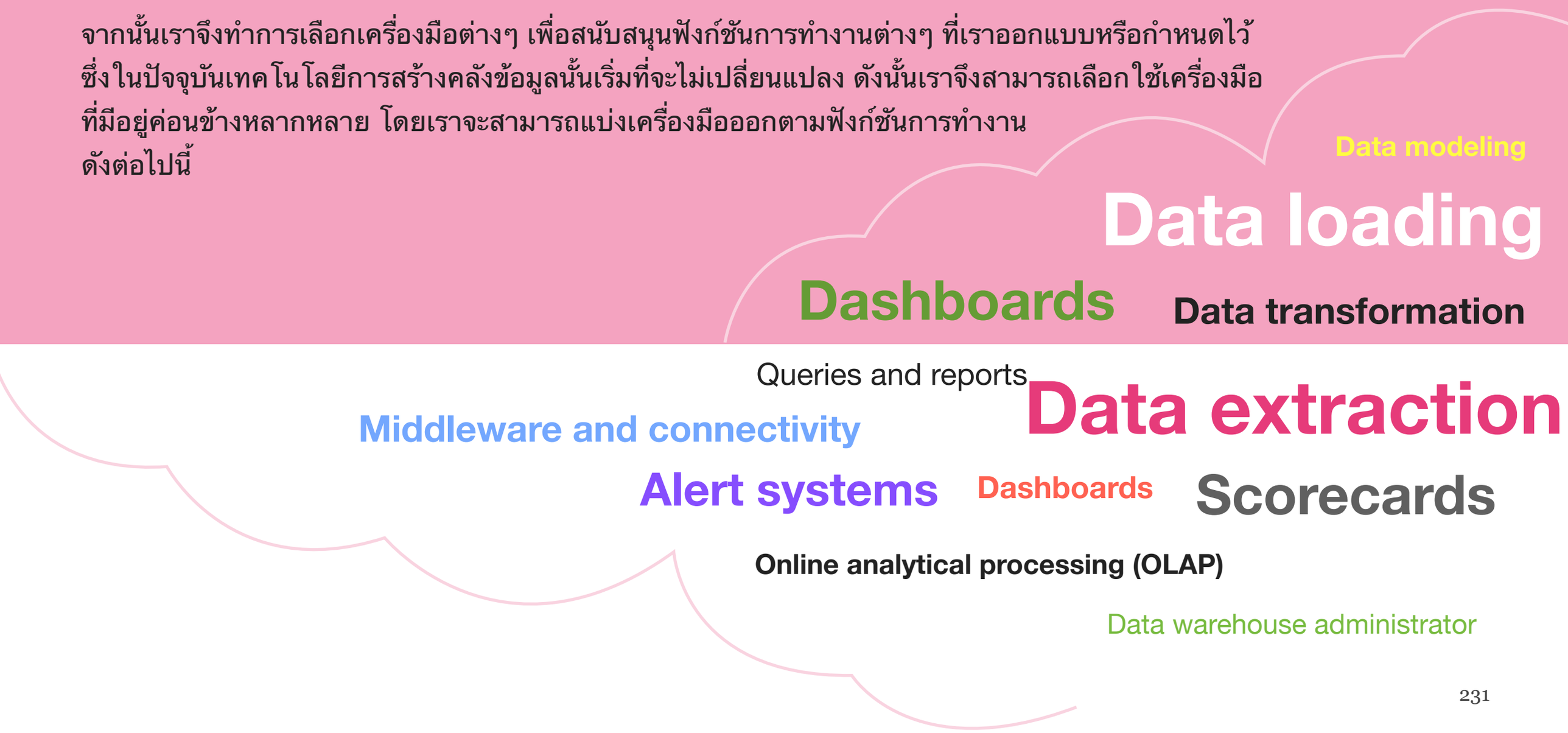
Query governor	ใช้สำหรับคาดการณ์และยกเลิกคิวรีที่ไม่สามารถควบคุมได้
Query optimizer	ใช้สำหรับวิเคราะห์และเพิ่มประสิทธิภาพให้กับการทำคิวรี
Query management	ใช้สำหรับปรับสมดุลของการคำนวณสำหรับคิวรีที่แตกต่างกัน
Load utility	ใช้สำหรับช่วยเพิ่มประสิทธิภาพของการโหลดข้อมูล (data loading) และการกู้คืนข้อมูล (data recovery)
Metadata management	ใช้สำหรับจัดการหรือจัดเก็บแคตตาล็อกข้อมูลหรือดาต้าดิกชันนารี (data catalog or data dictionary)

Scalability	ความยืดหยุ่นของฐานข้อมูลที่สามารถรองรับจำนวนผู้ใช้และจำนวนข้อมูลที่เพิ่มขึ้น
Extensibility	ความสามารถในการเพิ่มความสามารถไปเป็น OLAP database
Portability	ความสามารถในการทำงานข้ามแพลตฟอร์ม
Query tool application program interfaces (APIs)	มีเครื่องมือที่เป็นอินเทอร์เฟซ (interface) ที่ใช้ติดต่อกับผู้ใช้หรือไม่ เราจำเป็นต้องใช้เครื่องมือเหล่านี้หรือไม่
Administration	มีฟังก์ชันสำหรับการดูแลรักษาฐานข้อมูล

เครื่องมือต่างๆ ที่จำเป็นสำหรับคลังข้อมูล

หลังจากที่เราพิจารณาถึงฮาร์แวร์/เซิร์ฟเวอร์ ระบบปฏิบัติการ และระบบฐานข้อมูลแล้ว เราจำเป็นต้องพิจารณาถึงเครื่องมือต่างๆ ที่จะใช้สำหรับการสร้างคลังข้อมูล โดยที่ก่อนที่เราจะทำการเลือกเครื่องมือต่างๆ เราต้องออกแบบหรือกำหนดสถาปัตยกรรมของคลังข้อมูลที่เราจะทำการสร้างขึ้นเสียก่อน (โดยสถาปัตยกรรมที่เราออกแบบจะประกอบด้วยฟังก์ชันการทำงานต่างๆ มากมาย)

จากนั้นเราจึงทำการเลือกเครื่องมือต่างๆ เพื่อสนับสนุนฟังก์ชันการทำงานต่างๆ ที่เราออกแบบหรือกำหนดไว้ ซึ่งในปัจจุบันเทคโนโลยีการสร้างคลังข้อมูลนั้นเริ่มที่จะไม่เปลี่ยนแปลง ดังนั้นเราจึงสามารถเลือกใช้เครื่องมือที่มีอยู่ค่อนข้างหลากหลาย โดยเราจะสามารถแบ่งเครื่องมือออกตามฟังก์ชันการทำงาน ดังต่อไปนี้



Data modeling

- เป็นเครื่องมือที่ใช้สำหรับสร้างและจัดการกับแบบจำลองข้อมูล (data model) ที่ทำการเชื่อมโยงข้อมูลระหว่างข้อมูลจากแหล่งข้อมูลและข้อมูลที่จะจัดเก็บอยู่ในคลังข้อมูล โดยที่แบบจำลองข้อมูลที่จะทำการสร้างขึ้นอาจจะนำไปใช้ใน staging area
  - เป็นเครื่องมือที่มีความสามารถในการสร้าง database schema
    - เป็นเครื่องมือที่ใช้สำหรับสร้างแบบจำลองข้อมูลจากดาต้าดิกชันนารีที่ได้จากฐานข้อมูลของแหล่งข้อมูลหรือระบบการดำเนินงาน
      - เป็นเครื่องมือที่ช่วยในการออกแบบ diemsional model ที่ใช้สำหรับการสร้าง star schema

Data extraction

- เป็นเครื่องมือที่ช่วยในการสกัดข้อมูล โดยมีฟังก์ชันการสกัดข้อมูลสำหรับ full refresh และการเพิ่มเติมข้อมูลที่มีการเปลี่ยนแปลง
  - การเลือกเครื่องมือในการสกัดข้อมูลจะขึ้นอยู่กับแพลตฟอร์มของแหล่งข้อมูลและแพลตฟอร์มของฐานข้อมูลที่ใช้

Data transformation

- เป็นเครื่องมือสำหรับการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลให้อยู่ในรูปแบบและโครงสร้างที่เหมาะสม
- มีการกำหนดค่าที่เป็น default value ให้กับข้อมูลที่ขาดหายไป
- จะประกอบไปด้วยฟังก์ชันหลักๆ คือ การแยกหรือแตกข้อมูลจากฟิลด์หนึ่งๆ (fied splitting) การรวมข้อมูลเข้าด้วยกัน (consolidation) และการทำข้อมูลให้เป็นมาตราฐานเดียวกัน (standardization)

Data loading

- เป็นเครื่องมือสำหรับการถ่ายโอนข้อมูลที่ทำการเปลี่ยนแปลง/เปลี่ยนรูปแล้วที่อยู่ในรูปของ load image ไปยังพื้นที่สำหรับจัดเก็บข้อมูลในคลังข้อมูล
  - เครื่องมือที่ทำการถ่ายโอนข้อมูลอาจมีความสามารถในการสร้างคีย์หลักให้กับข้อมูลที่ทำการถ่ายโอน

Data quality

- เป็นเครื่องมือที่ช่วยในการค้นหาและแก้ไขความผิดพลาดที่เกิดขึ้นกับข้อมูล
- เป็นเครื่องมือที่ช่วยปรับปรุงความสอดคล้องของข้อมูลให้มีความสอดคล้องมากขึ้น
- เป็นเครื่องมือที่อาจจะใช้ในพื้นที่พักข้อมูลหรือใช้ปรับปรุงคุณภาพของข้อมูลในแหล่งข้อมูลโดยตรง

Queries and reports

- เป็นเครื่องมือที่ช่วยให้ผู้ใช้สามารถสร้างรายงานที่ซับซ้อน เป็นกราฟิก และรายงานสำเร็จรูปได้
- เป็นเครื่องมือที่ช่วยผู้ใช้ในการสร้างและรันคิวรีต่างๆ

Dashboards

- เป็นเครื่องมือที่ช่วยในการจัดเตรียมข้อมูลข่าวสารให้กับผู้ใช้แบบทันท่วงที ซึ่งเป็นการให้ข้อมูลที่มีการโต้ตอบกันระหว่างผู้ใช้กับคลังข้อมูล
- เป็นเครื่องมือที่อนุญาตให้ผู้ใช้สามารถทำงานต่างๆ ได้ เช่น การค้นหาข้อมูลแบบเจาะลึก ทำการเปลี่ยนแปลงค่าพารามิเตอร์ต่างๆ ได้

Scorecards

- เป็นเครื่องมือที่อนุญาตให้ผู้ใช้เลือกตัวชี้วัดประสิทธิภาพ (key performance indicator) สำหรับการสร้างรายงานต่างๆ ได้โดยง่าย
  - เป็นเครื่องมือสำหรับเปรียบเทียบระหว่างประสิทธิภาพ ณ ปัจจุบันและประสิทธิภาพในอดีต
    - เป็นเครื่องมือที่เน้นในเรื่องความชัดเจนและความง่ายในการใช้งาน

Online analytical processing (OLAP)

- เป็นเครื่องมือที่ช่วยให้ผู้ใช้รันคิวรีที่มีความซับซ้อน
- เป็นเครื่องมือที่ช่วยสร้างคิวรีสำเร็จรูป
- เครื่องมือทางด้าน OLAP จะสามารถแบ่งได้เป็น 2 ประเภทคือ MOLAP (Multidimensional online analytical processing) และ ROLAP (relational online analytical processing) ที่ซึ่ง MOLAP จะทำงานกับ multidimensional databases ที่รับข้อมูลมาจากคลังข้อมูลหลัก ในขณะที่ ROLAP จะทำงานกับ relational database ของคลังข้อมูล

Alert systems

- เป็นเครื่องมือที่จะแสดงข้อผิดพลาดที่เกิดขึ้นในคลังข้อมูล โดยสามรถทำการกำหนด exceptions ต่างๆ ได้

Middleware and connectivity

- เป็นเครื่องมือที่ช่วยในการเข้าถึงข้อมูลที่ประกอบไปด้วยหลายแพลตฟอร์ม

Data warehouse administrator

- เป็นเครื่องมือที่ช่วยผู้ดูแลคลังข้อมูลที่สามารถดูแลและจัดการงานในแต่ละวัน
- เป็นเครื่องมือที่เน้นในกระบวนการถ่ายโอนข้อมูลและติดตามประวัติของการถ่ายโอนข้อมูล
- เป็นเครื่องมือที่สามารถติดตามชนิดและจำนวนคิวรีที่ผู้ใช้เรียกดูข้อมูล

SECTION 6

คำถามท้ายบท

1. จงแจกแจงส่วนประกอบหลักของโครงสร้างพื้นฐานและหน้าที่ของแต่ละส่วนประกอบ
2. จงอธิบายถึงปัจจัยในการเลือกฮาร์ดแวร์สำหรับคลังข้อมูล
3. จงอธิบายถึงปัจจัยและทางเลือกในการเลือกระบบปฏิบัติการ
4. จงอธิบายถึงทางเลือกของสถาปัตยกรรมที่ประกอบไปด้วยหลายๆ โปรเซสเซอร์ โดยอธิบายถึงคุณลักษณะ ประโยชน์ และข้อจำกัด
5. จงอธิบายถึงการประมวลผลคิวรีแบบขนาน ว่ามีกระบวนการทำงานอย่างไรบ้าง
6. จงแจกแจงเครื่องมืออื่นๆ (นอกเหนือจาก ฮาร์ดแวร์ ระบบปฏิบัติการ และระบบจัดการฐานข้อมูล) ที่จำเป็นต่อการสร้างคลังข้อมูล

CHAPTER 5

การวางแผนและการจัดการสร้างคลังข้อมูล

SECTION 1

แผนการสอนประจำบท

วัตถุประสงค์ของบทเรียน

- ทบทวนสาระสำคัญของการวางแผนสำหรับคลังข้อมูล
- ศึกษาเกี่ยวกับความแตกต่างระหว่าง โครงการสร้างคลังข้อมูล และโครงการสร้างระบบ OLTP
- เรียนรู้วิธีการปรับ ใช้วงจรสำหรับการพัฒนาซอร์ฟแวร์เข้ากับ โครงการสร้างคลังข้อมูล
- ศึกษาและพิจารณาสัญญาณเตือนและปัจจัยความสำเร็จของ โครงการสร้างคลังข้อมูล

เนื้อหาของบทเรียน

เนื้อหาในบทนี้จะประกอบด้วย การวางแผนในการสร้างคลังข้อมูล โครงการสร้างคลังข้อมูล การจัดการโครงการสร้างคลังข้อมูล วงจรการพัฒนาคลังข้อมูล

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารประกอบการสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

กิจกรรมการเรียน-การสอน

- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารประกอบการสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

การวัดและประเมินผล

- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

SECTION 2

บทนำ

ในการสร้างคลังข้อมูลครั้งหนึ่งๆ นั้นอาจจะประสบความสำเร็จหรืออาจประสบความล้มเหลวก็เป็นได้ จากการเก็บรวบรวมสถิติที่ในการสร้างคลังข้อมูลให้แก่บริษัทหรือองค์กรต่าง ๆ จะทำให้เราทราบว่า ร้อยละ 50 ของการสร้างคลังข้อมูลจะไม่ประสบความสำเร็จ ในหลาย ๆ ครั้งของการสร้างคลังข้อมูลจะทำการสร้างคลังข้อมูลแต่ไม่แล้วเสร็จหรือคลังข้อมูลยังไม่สมบูรณ์ก็มีการยกเลิกการสร้างไป

นอกจากนั้นความล้มเหลวของการสร้างคลังข้อมูลอาจจะเกิดขึ้นหลังจากการสร้างคลังข้อมูลเสร็จสิ้นแต่ถูกปล่อยละเลยทิ้งไว้ไม่มีคนใช้ เนื่องจากข้อมูลในคลังข้อมูลไม่สอดคล้องกับการดำเนินธุรกิจหรืออาจเกิดจากปัจจัยอื่น ๆ จากปัญหาที่กล่าวข้างต้นเมื่อบริษัทหรือองค์กรหนึ่ง ๆ มีความประสงค์ที่จะทำการสร้างคลังข้อมูล เราในฐานะผู้สร้างคลังข้อมูลจะต้องทำการศึกษาถึงความเป็นไปได้ในการสร้างคลังข้อมูลของธุรกิจนั้น เช่น ความต้องการของผู้ใช้งาน ความจำเป็น งบประมาณ ความพร้อมของทีมผู้สร้างและอื่นๆ หลังจากทำการศึกษาและประเมินเบื้องต้นแล้ว เราจะต้องทำการวางแผนและเตรียมการจัดการที่ดีเพื่อให้คลังข้อมูลที่สร้างขึ้นประสบความสำเร็จ

เนื้อหาในบทนี้จะทำการอธิบายเกี่ยวกับการวางแผนและการจัดการโครงการสร้างคลังข้อมูล และ ปัจจัยที่มีผลกระทบต่อการสร้างคลังข้อมูล เพื่อทำให้เราสามารถดำเนินการสร้างคลังข้อมูลได้อย่างมีประสิทธิภาพและประสบความสำเร็จตามที่ตั้งเป้าไว้

PLAN

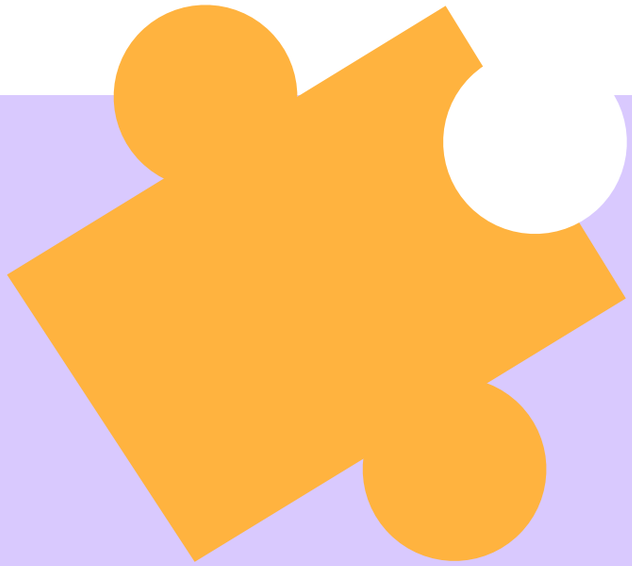
SECTION 3

การวางแผนในการสร้างคลังข้อมูล

ในการสร้างคลังข้อมูล ถ้าเรามีการวางแผนที่ไม่เหมาะสมและการจัดการต่าง ๆ ที่ไม่ดีพออาจก่อให้เกิดความล้มเหลวในการสร้างคลังข้อมูลได้ สิ่งแรกที่เราต้องทำการพิจารณาคือ “บริษัทหรือองค์กรนั้นๆต้องการคลังข้อมูลจริงๆหรือไม่?”—เพื่อที่จะตอบคำถามนี้เราจะต้องประเมินว่าสิ่งที่บริษัท นักวิเคราะห์ข้อมูล ผู้จัดการ และผู้บริหารคาดหวังจากคลังข้อมูลคืออะไร?

จากนั้นเราต้องทำการพิจารณาว่าเราควรจะสร้างคลังข้อมูลในลักษณะใดหรือแบบใด? เราควรจะต้องตรวจสอบให้แน่ใจว่าข้อมูลที่เราต้องการมากจากที่ใด และ แหล่งข้อมูลที่เรากำหนดมีข้อมูลที่เราต้องการหรือไม่? เราต้องเสาะหาว่าผู้ใช้คลังข้อมูลคือใคร? ผู้ใช้จะใช้คลังข้อมูลอย่างไรและเมื่อใด? คำถามข้างต้นนั้นเป็นคำถามที่มีความสำคัญมากต่อการสร้างคลังข้อมูล

ดังนั้นก่อนที่เราจะทำการสร้างคลังข้อมูล เราจะต้องทำการตอบคำถามเหล่านี้ให้ได้เสียก่อน เพื่อให้การสร้างคลังข้อมูลประสบความสำเร็จและคุ้มค่ากับการลงทุนในที่เป็นจำนวนเงินและเวลาที่เสียไป โดยในการตอบคำถามข้างต้น เราควรที่จะเริ่มทำการพิจารณาปัจจัยที่สำคัญดังต่อไปนี้



Value and Expectations

Value and Expectations

ในหลาย ๆ บริษัทที่ผ่านมามีการเริ่มทำการสร้างคลังข้อมูลโดยไม่มีการประเมินคุณค่าที่จะได้รับจากการสร้างคลังข้อมูล แต่อย่างไรก็ตาม ในการที่จะสร้างคลังข้อมูลให้ประสบความสำเร็จได้นั้น ขั้นตอนแรกเริ่ม เราจะต้อง**ประเมินให้ได้ว่ามีเพียงการสร้างคลังข้อมูลเท่านั้นหรือไม่ที่สามารถตอบสนองต่อความต้องการในการดำเนินธุรกิจหรือยังมีระบบหรือวิธีการอื่นๆที่ดีกว่าอีกบ้าง?**
—ถ้าเราได้คำตอบว่า**ใช่** ขั้นตอนต่อไปเราจะต้องทำการ**แจกแจงประโยชน์และคุณค่าที่จะได้รับจากการสร้างข้อมูล** จากนั้นเราจะต้องตอบคำถามเหล่านี้ให้ได้ เช่น การสร้างคลังข้อมูลสามารถช่วยให้ผู้บริหารสามารถตัดสินใจได้ดีขึ้นหรือไม่ คลังข้อมูลมีส่วนช่วยหรือสนับสนุนให้การดำเนินธุรกิจดีขึ้นหรือไม่ คลังข้อมูลสามารถช่วยเพิ่มส่วนแบ่งทางการตลาดได้หรือไม่—ถ้าได้เป็นจำนวนเท่าไร ความคาดหวังที่คาดว่าจะได้รับจากคลังข้อมูลมีอะไรบ้าง เป็นต้น เมื่อเราสามารถตอบคำถามเรานี้ได้ เราจะสามารถทราบถึงคุณประโยชน์ของการสร้างคลังข้อมูลที่มีต่อองค์กรและบริษัทนั้นๆ

Risk Assessment

Risk Assessment

การวางแผนที่ไม่ดีอาจส่งผลให้เกิดความล้มเหลวกขึ้นกับการสร้างคลังข้อมูล ถ้าการสร้างคลังข้อมูลล้มเหลว จะมีคำถามตามมาว่า เราสูญเสียเงินไปเป็นจำนวนเท่าไร? แต่โดยแท้จริงแล้วการประเมินความเสี่ยงนั้นไม่ได้มีแค่การคำนวณถึงค่าใช้จ่ายที่สูญเสียไป แต่จะมีส่วนอื่น ๆ ด้วย เช่น (1) อะไรคือความเสี่ยงที่จะได้รับเมื่อบริษัทไม่ได้รับประโยชน์จากคลังข้อมูล?—(เนื่องจากสร้างไม่เสร็จหรือสร้างแล้วไม่ได้คุณภาพเพียงพอที่จะใช้งาน) (2) จะมีความสูญเสียอะไรเกิดขึ้นบ้างจากการที่บริษัทไม่มีคลังข้อมูล? (3) บริษัทสูญเสียโอกาสอะไรบ้าง? และอื่นๆ จากที่กล่าวข้างต้น เราจะต้องประเมินความเสี่ยงในแง่มุมต่างๆ เพื่อให้ทราบถึงประโยชน์ที่ซ่อนเร้นของการสร้างข้อมูล ซึ่งก็คือความสูญเสียที่อาจจะเกิดขึ้นจากการไม่มีคลังข้อมูลนั่นเอง

Top-down and bottom-up

Top-down and bottom-up

ในบทที่ 2 ได้มีการอธิบายถึงการสร้างคลังข้อมูลทั้งในแบบ top-down และ bottom-up รวมถึงวิเคราะห์ข้อดี-ข้อเสีย ของทั้งสองวิธี เมื่อเราทราบรายละเอียดของวิธีการสร้างคลังข้อมูลแล้ว เราจะต้องทำการวิเคราะห์ถึงวิธีการสร้างที่เหมาะสมกับความต้องการและความจำเป็นของบริษัท เช่น ถ้าบริษัทมีความต้องการที่จะรีบใช้คลังข้อมูล เราควรจะสร้างคลังข้อมูลโดยใช้วิธี bottom-up ซึ่งจะทำการสร้างดาต้ามาร์ทแต่ละส่วนก่อนเพื่อให้ผู้บริหารหรือผู้จัดการแต่ละแผนกสามารถวิเคราะห์ข้อมูลในส่วนนั้น ๆ ได้ แต่ถ้าบริษัทของเราต้องการความสอดคล้องของการดำเนินธุรกิจในแต่ละส่วนงาน หรือต้องการข้อมูลที่เป็นกลุ่มก้อนเดียวกัน เราควรจะใช้วิธีการสร้างแบบ top-down ซึ่งจะทำให้ผู้สร้างเห็นภาพรวมของความต้องการทั่วทั้งองค์กร สามารถวางโครงสร้างข้อมูลที่เชื่อมต่อกันได้ แต่ก็ต้องแลกกับเวลาที่เพิ่มขึ้นในการสร้างข้อมูล ดังนั้นเมื่อถึงเวลาที่เราจะต้องเลือกวิธีในการสร้างคลังข้อมูล เราควรจะเลือกกลวิธีที่เหมาะสมกับสิ่งแวดล้อม ความต้องการ และข้อจำกัดที่เรามี ตามลำดับ

Build or buy

Build or buy

หลังจากที่มีการเริ่มการใช้คลังข้อมูลอย่างจริงมาตั้งแต่ปี 1990 จึงเป็นเหตุให้เทคโนโลยีทางด้านคลังข้อมูลเริ่มสุกงอม ในปัจจุบันได้มีบริษัทมากมายได้ผลิตซอร์ฟแวร์สำหรับสร้างคลังข้อมูลออกวางจำหน่าย ด้วยซอร์ฟแวร์จำนวนมากที่วางขายอยู่ในท้องตลาด จึงทำให้บริษัทที่จะทำการสร้างคลังข้อมูลเริ่มมีทางเลือกในการประยุกต์ใช้ซอร์ฟแวร์ต่างๆ ในแต่ละขั้นตอนการทำงานแทนที่จะทำการสร้างคลังข้อมูลขึ้นเองทั้งหมด เมื่อเรามีทางเลือกในการสร้างคลังข้อมูล อันดับแรกเราจะต้องทำการตัดสินใจว่าเราจะทำการสร้างคลังข้อมูลขึ้นเองหรือจะใช้ซอร์ฟแวร์สำเร็จในบางขั้นตอนการทำงานหรือทั้งหมด โดยในการตัดสินใจเราจะต้องพิจารณาสิ่งเหล่านี้ เช่น ดาต้ามาร์ทที่เราควรจะสร้างมีจำนวนเท่าใด (อาจจะเกิดจากไม่ต้องการให้ความลับลั่วไหลหรือเหตุผลอื่นๆ) และอื่นๆ

ในการทำงานของคลังข้อมูลจะมีฟังก์ชันการทำงานอยู่ที่ การสกัดข้อมูล การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล และการถ่ายโอนข้อมูล ซึ่งจากฟังก์ชันการทำงานทั้ง 3 เราจะต้องพิจารณาว่าเราจะทำการสร้างฟังก์ชันการทำงานทั้งหมดขึ้นเองหรือไม่ เราจะต้องการใช้ซอร์ฟแวร์สำเร็จสำหรับการส่งผ่านข้อมูลให้กับผู้ใช้หรือไม่ ถ้าเราประยุกต์ใช้ซอร์ฟแวร์สำหรับฟังก์ชันการทำงานต่าง ๆ จะช่วยให้เราสามารถดำเนินการได้อย่างรวดเร็วหรือไม่ ถ้าเรามีการวางแผนและจัดการที่ค่อนข้างดีและจากทางเลือกที่เรามี เราอาจทำการสร้างฟังก์ชันการทำงานขึ้นเองบางส่วนและประยุกต์ใช้ซอร์ฟแวร์บางส่วนในคลังข้อมูลที่จะทำการสร้างขึ้น

Single vendor

or best-of-breed

ถ้าบริษัทของเราตัดสินใจที่จะใช้ซอร์ฟแวร์สำเร็จรูปในการสร้างคลังข้อมูล เราจะต้องทำการพิจารณาถึงข้อดีข้อเสียของแต่ละซอร์ฟแวร์ที่มีอยู่ที่ท้องตลาด เมื่อเราทราบถึงข้อดีข้อเสียของแต่ละซอร์ฟแวร์ที่สนับสนุนแต่ละฟังก์ชันการทำงานแล้ว เราจะต้องพิจารณาว่าเราควรจะใช้ซอร์แวร์เพียงซอร์ฟแวร์เดียวจากบริษัทเพียงบริษัทเดียวหรือจะใช้หลายซอร์ฟแวร์จากหลายๆ บริษัท โดยทำการเลือกซอร์ฟแวร์ที่ดีที่สุดสำหรับแต่ละฟังก์ชันการทำงาน ลองพิจารณาข้อดีของแต่ละทางเลือกที่เรามีดังนี้

- ประโยชน์ของการใช้ซอร์ฟแวร์เพียงซอร์ฟแวร์เดียว
- การผสมผสานระหว่างฟังก์ชันการทำงานต่างๆ ค่อนข้างลงตัวราวกับการเชื่อมต่อของแต่ละส่วนงานนั้นไร้รอยต่อ
- การแลกเปลี่ยนข้อมูลสามารถจัดการจากส่วนกลาง
- สามารถต่อรองราคากับผู้ขายได้

การใช้ซอร์ฟแวร์จากบริษัทเดียวจะทำให้เราผสานฟังก์ชันการทำงานต่าง ๆ ได้ค่อนข้างดี แต่มีข้อเสียตรงที่มีตัวเลือกของซอร์ฟแวร์ค่อนข้างน้อย เช่น IBM และ NCR ที่มีการพัฒนาซอร์ฟแวร์ที่ครบถ้วนสมบูรณ์ จากข้อจำกัดที่มีของการใช้ซอร์ฟแวร์จากบริษัทเดียว ลองพิจารณาถึงการใช้ซอร์ฟแวร์จากหลาย ๆ บริษัท โดยพิจารณาถึงข้อดี-ข้อเสียของการใช้ดังนี้

ข้อดี

ประโยชน์ของการใช้หลายซอร์ฟแวร์

- สามารถสร้างสภาพแวดล้อมให้เหมาะสมกับองค์กรของคุณ
- เราสามารถเลือกผลิตภัณฑ์ที่เหมาะกับงานเฉพาะทางได้

จากข้อดีของการใช้หลายซอร์ฟแวร์จะทำให้เราได้คลังข้อมูลที่สนับสนุนการทำงานได้อย่างเต็มที่ แต่ปัญหาของทางเลือกนี้จะอยู่ที่การทำให้หลายๆซอร์ฟแวร์นั้นทำงานเข้ากันได้ดี ซึ่งแต่ละซอร์ฟแวร์ตามท้องตลาดจะถูกสร้างขึ้นเพื่อสนับสนุนแต่ละงานและยังไม่มีมาตราฐานส่วนกลาง ดังนั้นในการเชื่อมต่อระหว่างซอร์ฟแวร์ให้ทำงานร่วมกันอาจจะทำได้ยากหรือไม่อาจทำได้ ดังนั้นเมื่อเราทำการเลือกซอร์ฟแวร์ต่างๆที่จะใช้เราต้องตรวจสอบให้แน่ใจว่าซอร์ฟแวร์เหล่านั้นสามารถทำงานร่วมกันได้

ข้อเสีย

ข้อเสียอีกข้อหนึ่งของการใช้หลายซอร์ฟแวร์คืออำนาจการต่อรองของบริษัทกับบริษัทผู้ผลิตซอร์ฟแวร์จะลดลง ซึ่งจะทำให้เราอาจจะต้องเสียงบประมาณที่สูงขึ้น วิธีการนี้อาจจะไม่เหมาะกับบริษัทที่ไม่ต้องการเทคนิคที่มากมายนัก

ดังนั้นเราควรที่จะวางแผนและประเมินถึงความต้องการของบริษัทของเราเป็นอันดับแรกจากนั้นค่อยทำการตัดสินใจที่จะเลือกใช้ซอร์ฟแวร์จากความต้องการที่มีอยู่

PLAN

การคำนึงถึงความต้องการในเชิงธุรกิจก่อนเทคโนโลยีในการสร้างคลังข้อมูล

ในการวางแผนสำหรับการสร้างคลังข้อมูล เราจะต้องคำนึงถึงความต้องการจากผู้ใช้มากกว่าเทคโนโลยีที่จะใช้ในการสร้างคลังข้อมูล ซึ่งในหลายๆครั้งของการสร้างคลังข้อมูล ผู้สร้างมักจะหลงลืมไปว่า “คลังข้อมูลนั้นถูกสร้างขึ้นเพื่อตอบสนองความต้องการข้อมูลเชิงกลยุทธ์ของผู้ใช้” โดยผู้สร้างหลายรายจะสนใจที่วิธีการในการสร้างมากกว่าข้อมูลที่ผู้ใช้ต้องการ ดังนั้นในการสร้างคลังข้อมูล เราไม่ควรจะทำการสร้างคลังข้อมูลทั้งที่เรายังไม่มีความเข้าใจถึงความต้องการที่แท้จริง เราควรจะเริ่มจากการพิจารณาว่าในการสร้างคลังข้อมูลนั้นต้องการข้อมูลอะไรบ้าง ไม่ใช่ว่าเราจะจัดหาข้อมูลเรานั้นได้อย่างไร
จากนั้นเราจึงทำการพิจารณาถึงโครงสร้างของข้อมูลและสถาปัตยกรรมของคลังข้อมูลที่สนับสนุนความต้องการของผู้ใช้

ในการที่จะทราบถึงความต้องการ เราควรจะทำแบบสำรวจเบื้องต้นเพื่อทราบถึงความต้องการจากผู้ใช้ทุกกลุ่ม ซึ่งจะทำให้เราเข้าใจถึงการดำเนินธุรกิจมากขึ้นและทำให้เราสามารถวางแผนที่จะสร้างคลังข้อมูลได้รัดกุมมากขึ้น นอกจากนี้เรายังสามารถจัดลำดับความสำคัญและกำหนดลำดับการสร้างดาต้ามาร์ทได้อีกด้วย ตัวอย่างเช่น หลังจากทำแบบสำรวจเบื้องต้นแล้วเราจะต้องทำการสร้างดาต้ามาร์ทสำหรับการขายสินค้าเป็นลำดับแรก จากนั้นค่อยสร้างดาต้ามาร์ทสำหรับการเงิน และอื่นๆ ตามลำดับ

จากตัวอย่างข้างต้น ยังมีข้อมูลอื่น ๆ ที่เราอาจจะได้รับจากสำรวจเบื้องต้นจากผู้ใช้แต่ละกลุ่มดังนี้

- ภารกิจและหน้าที่ของผู้ใช้แต่ละกลุ่ม
- ระบบคอมพิวเตอร์ที่ใช้โดยผู้ใช้แต่ละกลุ่ม
- ตัวชี้วัดประสิทธิภาพที่สำคัญของแต่ละฟังก์ชัน
- ปัจจัยที่มีผลต่อความสำเร็จของกลุ่มผู้ใช้นั้นๆ
- ผลิตภัณฑ์ที่ผลิตหรือขาย
- หมวดหมู่ของสินค้าและบริการ
- สถานที่ที่ใช้ในการดำเนินธุรกิจ
- ระดับของการวัดผลประกอบการ/กำไร-ต่อลูกค้า ต่อรายการสินค้า ต่ออาณาเขต/บริเวณ
- ระดับของรายละเอียดค่าใช้จ่ายและรายได้
- คิวรีและรายงานสำหรับข้อมูลเชิงกลยุทธ์ที่ใช้ในปัจจุบัน
- อื่นๆ

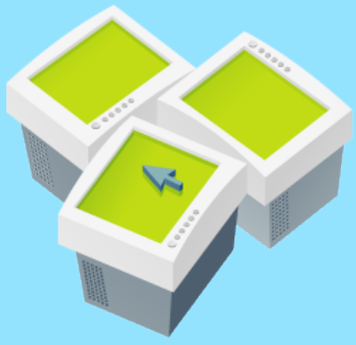
นอกจากเราจะทำการสำรวจความต้องการจากผู้ใช้กลุ่มต่างๆ แล้ว เราจะต้องทำการสำรวจหรือสอบถามผู้ดูแลระบบการดำเนินงานที่ซึ่งจะทำให้เราทราบถึงข้อมูลที่เราต้องการเก็บอยู่ที่ใด ทราบว่าระบบการดำเนินงานมีสถาปัตยกรรมเป็นอย่างไร ทราบถึงความสัมพันธ์ระหว่างโครงสร้างของข้อมูลและคุณภาพของข้อมูล ทราบถึงเอกสาร/คู่มือสำหรับระบบการดำเนินงาน และอื่นๆ สิ่งเหล่านี้จะเป็นข้อมูลทั้งหมดที่เกี่ยวกับระบบการดำเนินงานที่มีส่วนช่วยในการวางแผนและประเมินการสร้างคลังข้อมูลด้วยเช่นกัน

การอธิบายเกี่ยวกับการสร้างคลังข้อมูล

ก่อนที่จะทำการสร้างคลังข้อมูลเราสามารถประมาณค่าใช้จ่ายอย่างคร่าวๆ ได้ ซึ่งโดยส่วนใหญ่ของค่าใช้จ่ายในการสร้างคลังข้อมูลจะประกอบไปด้วย **ค่าฮาร์ดแวร์ 31% ค่าซอร์ฟแวร์และฐานข้อมูล 24% ค่าใช้จ่ายในการรวมฟังก์ชันการทำงานต่างๆเข้าด้วยกัน 35% และค่าบริหารจัดการ 10%** ซึ่งจากค่าใช้จ่ายที่ประเมินไว้ เราจะสามารถคำนวณ ROI และ ROA ได้อย่างไร?—การที่จะคำนวณความคุ้มค่าหรือผลประโยชน์จากการใช้คลังข้อมูลนั้นค่อนข้างทำได้ยาก เนื่องจากเราอาจจะไม่ทราบถึงประโยชน์ที่แท้จริงจนกระทั่งทำการสร้างและนำคลังข้อมูลไปใช้อย่างเต็มที่

แต่อย่างไรก็ตามเราจะต้องทำการวิเคราะห์เกี่ยวกับค่าใช้จ่ายและประโยชน์ที่จะได้รับก่อนทำการตัดสินใจสร้างคลังข้อมูลเพื่อประเมินผลประโยชน์ที่บริษัทจะได้รับ เมื่อเราเริ่มทำการวิเคราะห์ต่างๆ เราสามารถวิเคราะห์หลายวิธี ดังตัวอย่างดังต่อไปนี้

การคำนวณค่าใช้จ่ายเกี่ยวกับเทคโนโลยีที่มีอยู่ในปัจจุบันสำหรับสร้างแอพลิเคชันและรายงานต่างๆ ที่สนับสนุนการตัดสินใจเชิงกลยุทธ์ จากนั้นทำการเปรียบเทียบค่าใช้จ่ายที่คำนวณไว้กับค่าใช้จ่ายโดยประมาณของการสร้างคลังข้อมูลโดยทำการหาอัตราส่วนระหว่างค่าใช้จ่ายทั้งสอง



การคำนวณมูลค่าทางธุรกิจที่จะได้รับหลังจากทำการสร้างคลังข้อมูล โดยทำการประมาณเป็นค่าเงินสำหรับผลกำไร เงินปันผล การเติบโตของรายได้ และส่วนแบ่งการตลาดที่เติบโตขึ้น จากนั้นทำการพิจารณาถึงมูลค่าทางธุรกิจเทียบกับค่าใช้จ่ายในการสร้างคลังข้อมูล

การระบุเกี่ยวกับทุกส่วนประกอบที่จะมีผลกระทบกับคลังข้อมูล เริ่มจากต้นทุนในส่วนต่างๆ เช่น การซื้อหรือเช่าฮาร์ดแวร์ ซอร์ฟแวร์จากบริษัท ซอร์ฟแวร์ที่สร้างขึ้นเอง การดูแลรักษาคลังข้อมูล จากนั้นคำนวณเกี่ยวกับมูลค่าทางธุรกิจที่จะได้รับ ซึ่งประกอบไปด้วย ค่าใช้จ่ายที่ลดลง รายได้ที่เพิ่มขึ้น และประสิทธิภาพในการดำเนินธุรกิจ จากนั้นทำการวิเคราะห์เงินหมุนเวียนและคำนวนเกี่ยวกับ ROI

การวางแผนทั้งหมดสำหรับสร้างคลังข้อมูล

จากที่กล่าวมาก่อนหน้าจะเป็นการวางแผนหรือพิจารณาในแต่ละแง่มุมก่อนที่จะเริ่มทำการสร้างคลังข้อมูล ซึ่งการวางแผนที่ดีนั้นจะช่วยให้เห็นข้อดี-ข้อเสียของการสร้างคลังข้อมูล สร้างแรงกระตุ้นหรือแรงบันดาลใจให้กับการทำงานต่างๆ ในการวางแผนเราอาจจะมองหาทางเลือกและเหตุผลในการเลือกขั้นตอนหรือกระบวนการทำงาน ชนิดของคลังข้อมูล และทำการแจงแจงความคาดหวังจากผู้ใช้ การวางแผนการสร้างคลังข้อมูลนั้นประกอบด้วยขั้นตอนต่างๆ มากมายซึ่งเราสามารถจัดลำดับของขั้นตอนต่างๆ ได้ดังรูปที่ 5-1

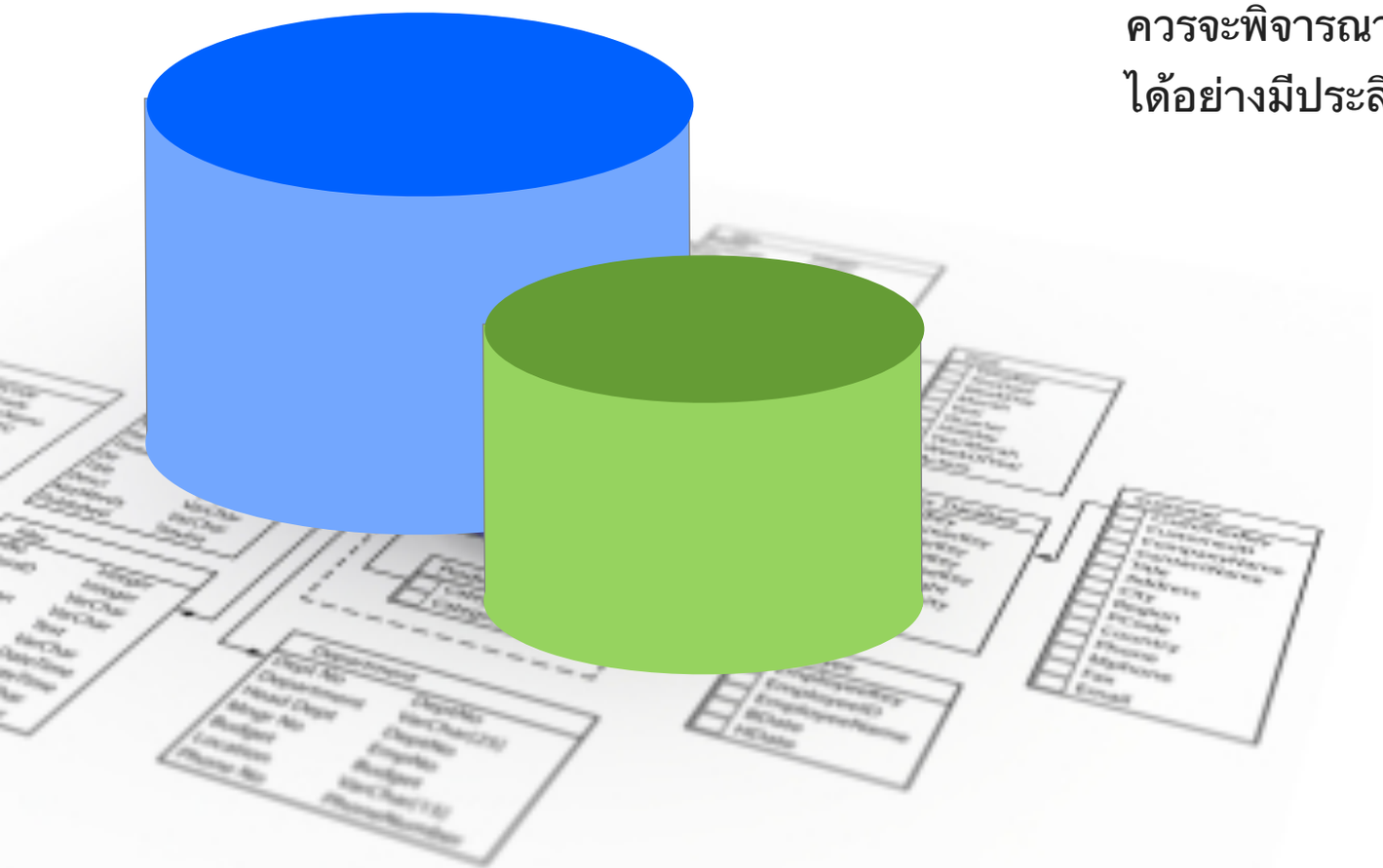
รูปที่ 5-1 แผนการเบื้องต้นสำหรับสร้างคลังข้อมูล

SECTION 4

โครงการสร้างคลังข้อมูล

โครงการสร้างคลังข้อมูล

การสร้างคลังข้อมูลจะมีขั้นตอนการทำงานที่แตกต่างจากการสร้างระบบการดำเนินงาน ถ้าเรามีประสบการณ์ในการสร้างระบบการดำเนินงาน แต่ยังไม่มีประสบการณ์เกี่ยวกับคลังข้อมูล เราควรจะพิจารณาถึงความแตกต่างเหล่านั้นเพื่อให้สามารถสร้างคลังได้อย่างมีประสิทธิภาพ



ความแตกต่างของระบบคลังข้อมูลและระบบการดำเนินงาน

คลังข้อมูลจะมีลักษณะเฉพาะที่แตกต่างจากระบบการดำเนินงาน ถ้าเราทำการเปรียบเทียบคลังข้อมูลกับระบบการดำเนินงานได้ จะช่วยให้เราเข้าใจถึงความแตกต่างของทั้งสองระบบ และเข้าใจคุณลักษณะเฉพาะของคลังข้อมูลได้ จากบทที่ 2 ทำให้เราทราบถึงส่วนประกอบของคลังข้อมูลที่จะประกอบด้วยฟังก์ชันการทำงาน 3 ฟังก์ชันหลักด้วยกันคือ

(1) การเก็บรวบรวมข้อมูล/การได้มาซึ่งข้อมูล (data acquisition)

(2) การจัดเก็บข้อมูล (data storage)

(3) การส่งผ่านข้อมูลสารสนเทศให้กับผู้ใช้งาน (information delivery)

ดังนั้นเมื่อเราจะทำการเปรียบเทียบหาความแตกต่างของทั้ง 2 ระบบเราควรจะพิจารณาผ่านฟังก์ชันการทำงานหลักทั้ง 3 ฟังก์ชันนี้ ดังแสดงในรูปที่ 5-2

Data Warehouse: Distinctive Features and Challenges for Project Management

DATA ACQUISITION	DATA STORAGE	INFO. DELIVERY
Large number of sources	Storage of large data volumes	Several user types
Many disparate sources	Rapid growth	Queries stretched to limits
Different computing platforms	Need for parallel processing	Multiple query types
Outside sources	Data storage in staging area	Web-enabled
Huge initial load	Multiple index types	Multidimensional analysis
Ongoing data feeds	Several index files	OLAP functionality
Data replication considerations	Storage of newer data types	Metadata management
Difficult data integration	Archival of old data	Interfaces to DSS apps.
Complex data transformations	Compatibility with tools	Feed into Data Mining
Data cleansing	RDBMS & MDDBMS	Multi-vendor tools

รูปที่ 5-2 ความท้าทายที่พบระหว่างการสร้างคลังข้อมูล

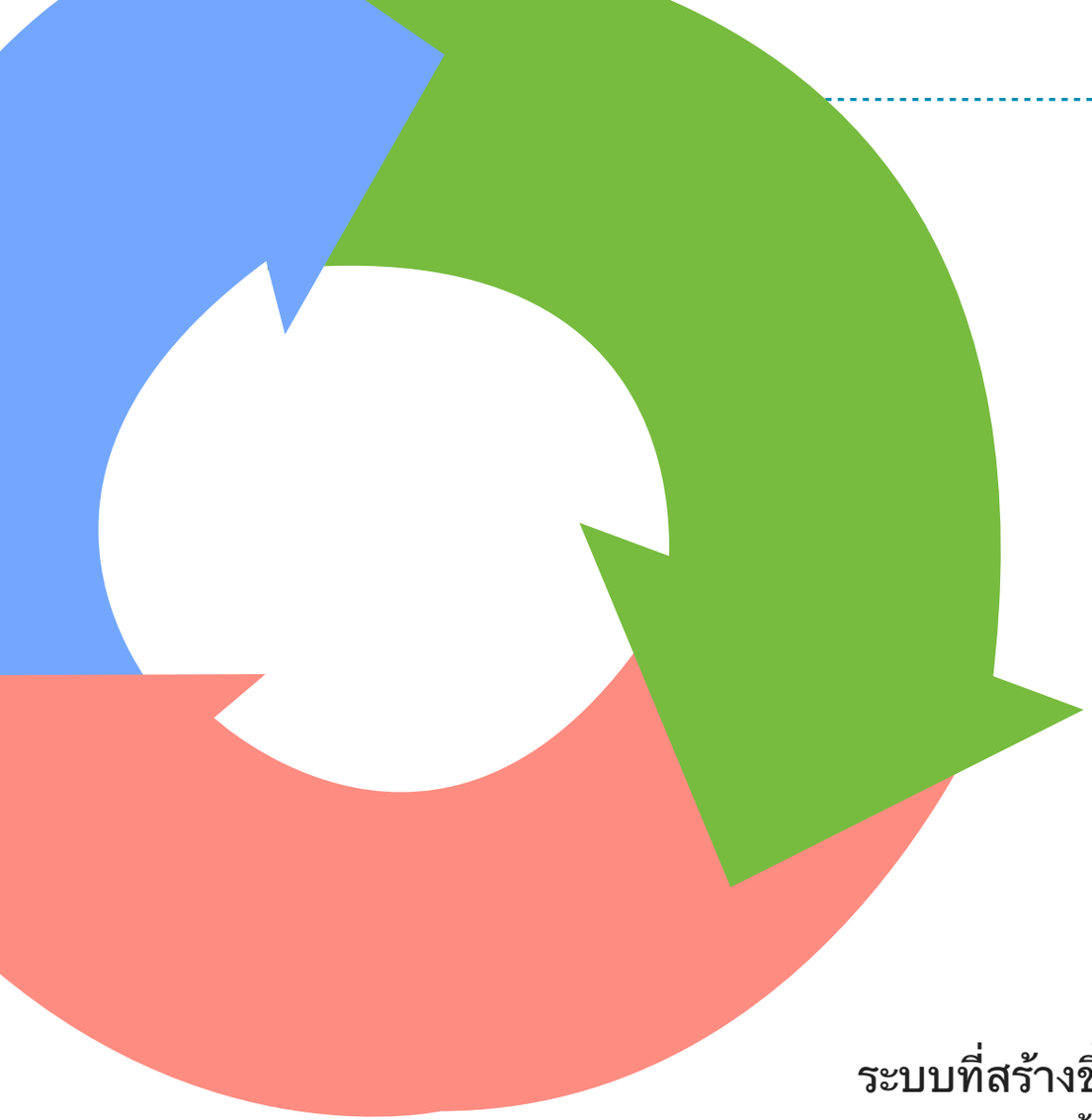
หลังจากเราทราบถึงความแตกต่างแล้ว เราควรพิจารณาถึงผลที่ตามมาจากความแตกต่างเหล่านั้น เพื่อที่จะกำหนดทิศทางการทำงานให้กับฟังก์ชันต่างๆ แต่ก่อนอื่น ลองพิจารณาความแตกต่างของทั้งสองระบบในหัวข้อสำคัญๆ ดังต่อไปนี้

- คลังข้อมูลมีขอบเขตที่กว้าง มีความซับซ้อน และเกิดจากการรวมกันของหลายเทคโนโลยี
- คลังข้อมูลจะอนุญาตให้มีการเพิ่มชนิดของกิจกรรมใหม่ๆ
- เมตาดาต้าที่เก็บอยู่ในคลังข้อมูลนั้นมีความสำคัญมาก เราจำเป็นต้องให้ความใส่ใจกับเมตาดาต้าเป็นพิเศษ
- การสร้างคลังข้อมูลมักจะนำซอร์ฟแวร์หรือเครื่องมือที่วางจำหน่ายอยู่ตามท้องตลาดมาช่วยในการสร้างฟังก์ชันการทำงานต่างๆ ดังนั้นเราต้องเผื่อเวลาไว้สำหรับการประเมินคุณภาพและเลือกซอร์ฟแวร์หรือเครื่องมือที่จะนำมาประยุกต์ใช้กับคลังข้อมูล
- การสร้างคลังข้อมูลจะให้ความสำคัญในการสร้างและดำเนินการเกี่ยวกับโครงสร้างพื้นฐาน (infrastructure) สถาปัตยกรรม (architecture) และการให้ความรู้แก่ผู้ใช้ (training) ในการสร้างคิวรีและรายงาน ซึ่งขั้นตอนเหล่านี้อาจใช้เวลาค่อนข้างมาก
- คลังข้อมูลจะประยุกต์ใช้การคำนวณแบบขนาน (parallel computing) กับฟังก์ชันการทำงานที่ค่อนข้างซับซ้อน และมีภาระงานค่อนข้างมาก

การประเมินความพร้อมในการสร้างคลังข้อมูล

ก่อนที่จะทำการสร้างคลังข้อมูล เราควรจะมีการประเมินความพร้อมของทีมงานที่จะทำการสร้างคลังข้อมูลและกลุ่มผู้ใช้งานคลังข้อมูล ซึ่งจะเป็นการประเมินความพร้อมและกำหนดเป้าหมายสำคัญที่จะช่วยให้หัวหน้าโปรเจค (Project manager) ทราบถึงช่องว่างหรือข้อบกพร่องต่างๆของทีมงานและแผนการสร้างคลังข้อมูล นอกจากนี้การประเมินความพร้อมยังช่วยในด้านต่างๆ เช่น

- ลดความเสี่ยงของการเกิดปัญหาระหว่างการดำเนินการ
- สามารถควบคุมการแก้ปัญหาด้วยกระบวนการที่วางแผนไว้
- มีการประเมินข้อตกลงและความรับผิดชอบอีกครั้งหนึ่ง
- มีการตรวจสอบและระบุขอบเขต/ขนาดของคลังข้อมูลอีกครั้งหนึ่ง
- มีการระบุถึงปัจจัยสำคัญที่จะนำไปสู่ความสำเร็จ
- มีการตรวจสอบความคาดหวังของผู้ใช้
- มีการตรวจสอบความต้องการในการได้รับความรู้
- อื่นๆ

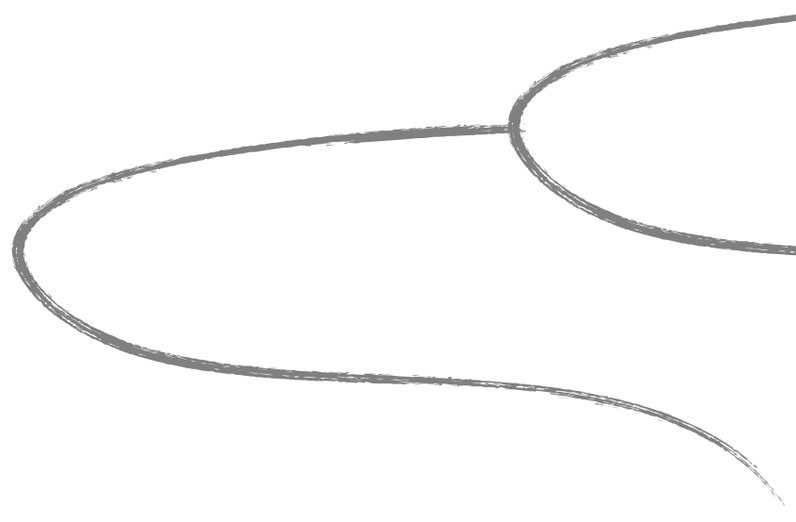


วงจรการพัฒนาคลังข้อมูล

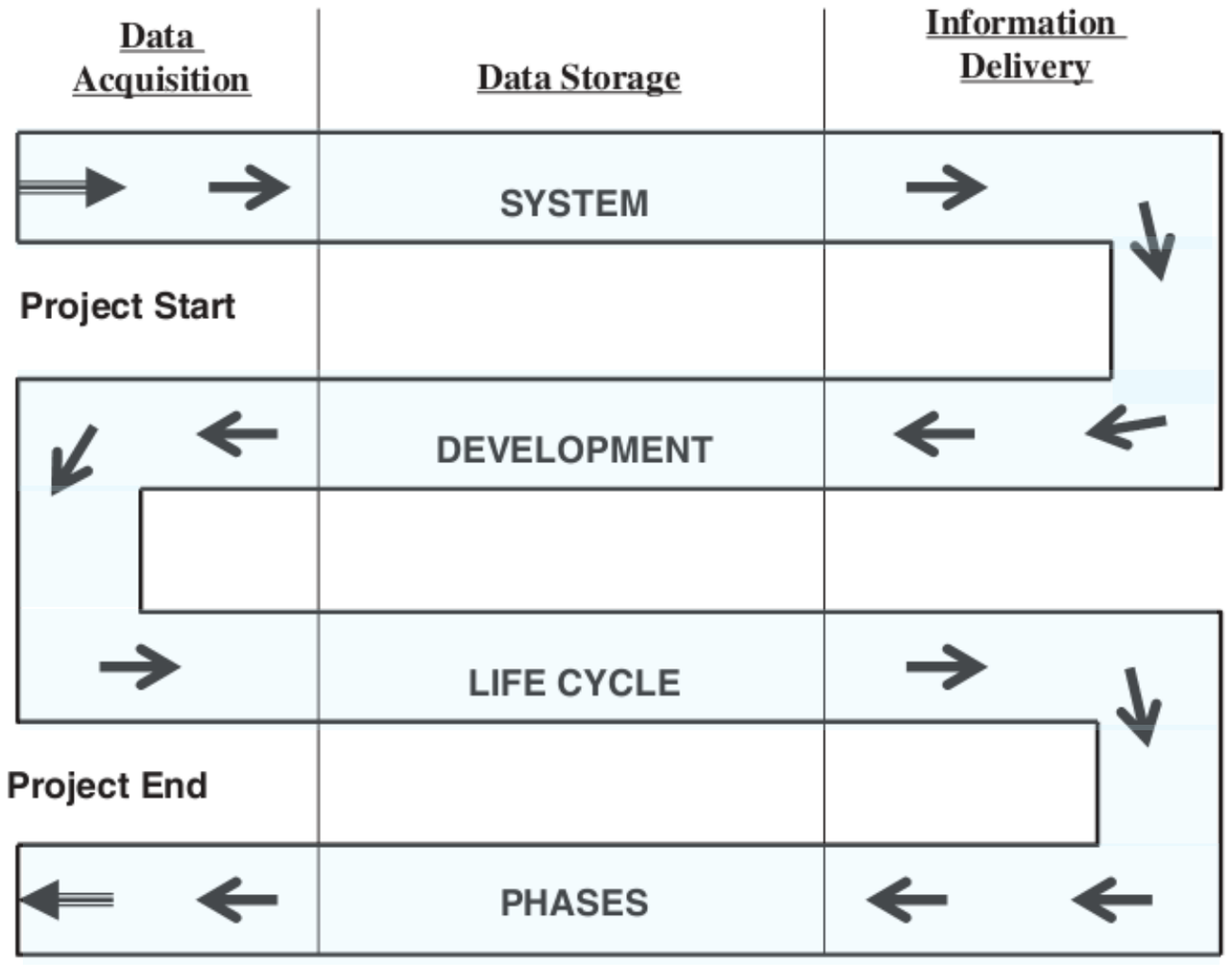
ในการสร้างระบบการดำเนินงานเราจะคุ้นเคยกับกระบวนการสร้างระบบหรือที่เรียกว่า system development life cycle (SDLC) ซึ่งกระบวนการสร้างระบบจะเกี่ยวกับขั้นตอนการทำงานต่าง ๆ มากมาย เช่น **การวางแผน การวิเคราะห์ความต้องการ การออกแบบ การดำเนินการ และการทดสอบ** ระบบที่สร้างขึ้น กระบวนการสร้างระบบจะเป็นส่วนช่วยให้การสร้างระบบนั้นเป็นไปอย่างมีลำดับเป็นขั้นตอน ลดความซ้ำซ้อนของโปรเจคและลดความกำกวมของหน้าที่ของแต่ละส่วนของทีม แต่อย่างไรก็ตาม ในการสร้างคลังข้อมูลที่มีความซับซ้อนทั้งฟังก์ชันการทำงานและการรวมกันของเทคโนโลยี เราไม่สามารถประยุกต์ใช้กลยุทธ์หรือกระบวนการสร้างระบบปฏิบัติการอันใดอันหนึ่งได้โดยตรง เช่น การใช้ waterfall ในการสร้างคลังข้อมูลเป็นต้น เราอาจจะต้องใช้วิธีที่แตกต่างไปจากเดิมที่จะสามารถเข้ากันได้ดีกับฟังก์ชันการทำงานต่าง ๆ ของคลังข้อมูล

โดยส่วนใหญ่ของการสร้างคลังข้อมูลมักจะประยุกต์ใช้กระบวนการทำซ้ำ *(iterative method)* ที่จะช่วยให้เราสามารถปรับแต่งฟังก์ชันการทำงานต่างๆ ได้ ตัวอย่างเช่น ถ้าเราต้องการระบุถึงแหล่งที่มาของข้อมูล เราอาจเริ่มจากการตรวจสอบแหล่งข้อมูลทั้งหมดและทำการแจกแจงหรือพิจารณาถึงโครงสร้างข้อมูลของแหล่งข้อมูลเหล่านั้น ในรอบการทำงานถัดไปจะทำการตรวจสอบองค์ประกอบของข้อมูลที่อยู่ในแต่ละแหล่งข้อมูลกับผู้ใช้และผู้ดูแลระบบ

จากนั้นทำการตรวจสอบองค์ประกอบของข้อมูลอีกครั้งหนึ่ง เป็นต้น กระบวนการทำซ้ำนั้นเป็นสิ่งจำเป็นสำหรับการสร้างคลังข้อมูล เนื่องจากความซับซ้อนของการทำงานและขอบเขตที่กว้าง ซึ่งเราอาจจำเป็นต้องมีกระบวนการในการพิจารณาหลายครั้งสำหรับฟังก์ชันการทำงานต่างๆ



เมื่อเราเลือกที่จะประยุกต์ใช้กระบวนการทำซ้ำในการสร้างคลังข้อมูล เราต้องไม่ลืมว่าฟังก์ชันการทำงานหลักของคลังข้อมูลประกอบด้วย การได้มาซึ่งข้อมูล การจัดเก็บข้อมูล และการส่งผ่านข้อมูล ดังนั้นเราจะต้องพิจารณาว่ากระบวนการทำซ้ำนั้นครอบคลุมการทำงานของทั้ง 3 ฟังก์ชันหรือไม่ ลองพิจารณารูปที่ 5-3 ที่แสดงถึงความสัมพันธ์ของทั้ง 3 ฟังก์ชันกับกระบวนการในการสร้างระบบ

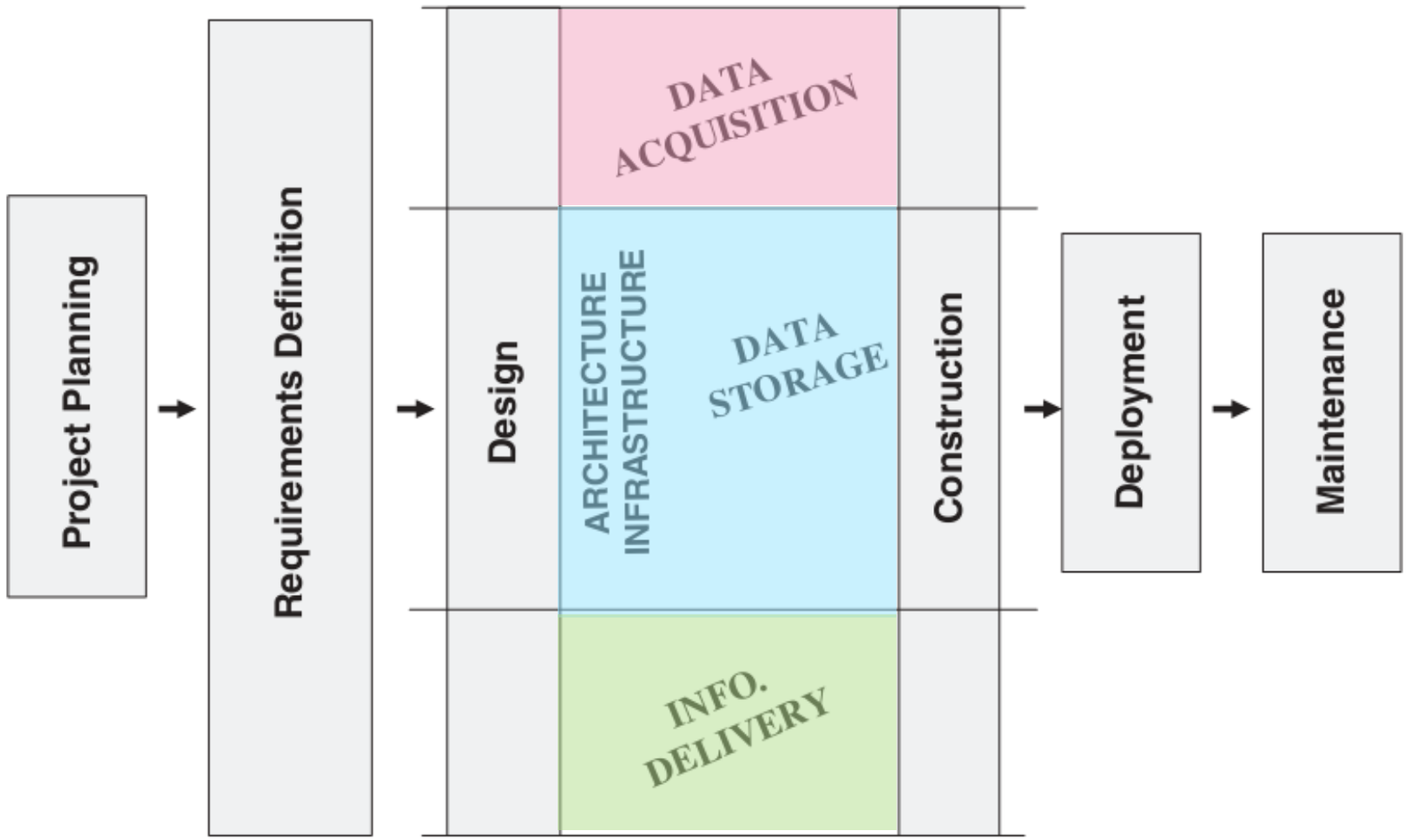


รูปที่ 5-3 ขั้นตอนการสร้างคลังข้อมูลที่สอดคล้องกับ 3 ฟังก์ชันการทำงานหลัก

ขั้นตอนการพัฒนาคลังข้อมูล

ในการกำหนดกระบวนการสร้างคลังข้อมูล เราจะต้องแน่ใจว่าแต่ละขั้นตอนจะสนับสนุนฟังก์ชันการเก็บรวบรวมข้อมูล การจัดเก็บข้อมูล และการส่งผ่านข้อมูล รวมถึงไม่ทำงานขัดกับโครงสร้างพื้นฐานและ สถาปัตยกรรมที่ได้ออกแบบไว้ ลองพิจารณารูปที่ 5-4 ที่แสดงกระบวนการสร้างคลังข้อมูลที่ประกอบไปด้วยการวางแผน การกำหนดความต้องการ การออกแบบฟังก์ชันการทำงาน (การเก็บรวบรวมข้อมูล การจัดเก็บข้อมูล และการส่งผ่านข้อมูล) การสร้างฟังก์ชันการทำงานต่างๆข้างต้น การติดตั้งระบบที่สร้างขึ้นเพื่อใช้งาน และการดูแลบำรุงรักษาคลังข้อมูลที่สร้างขึ้น

จากกระบวนการต่างๆในรูป 5-4 จะไม่ได้ทำการเน้นที่ส่วนใดส่วนหนึ่งเป็นพิเศษและเป็นกระบวนการทำงานอย่างคร่าวๆ ดังนั้นถ้าเราจะประยุกต์ใช้กระบวนการขั้นตอนวิธีดังกล่าว เราจะต้องพิจารณาถึงสถานการณ์ที่เราพบเจออยู่ เช่น ถ้าองค์กรของเรามีปัญหาเกี่ยวกับคุณภาพของข้อมูล เราควรจะเน้นย้ำที่กระบวนการออกแบบและสร้างฟังก์ชันการเก็บรวบรวมข้อมูล

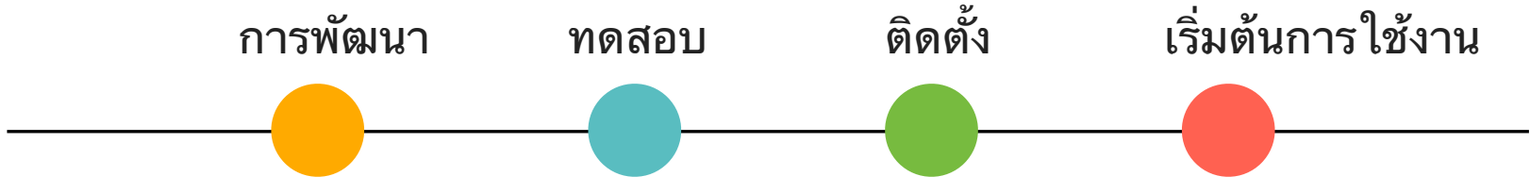


รูปที่ 5-4 ขั้นตอนการสร้างคลังข้อมูล

SECTION 5

การจัดการโครงการสร้างคลังข้อมูล

การจัดการโครงการสร้างคลังข้อมูล



เมื่อกระบวนการพัฒนา ทดสอบ ติดตั้งและเริ่มต้นการใช้งานเสร็จสิ้น เราจะได้คลังข้อมูลที่สมบูรณ์ซึ่งถือว่าเป็นความสำเร็จก้าวแรกของการสร้างคลังข้อมูล แต่ไม่ใช่ว่าทุกครั้งของการสร้างคลังข้อมูลจะประสบผลสำเร็จ หลาย ๆ ครั้ง โปรเจคการสร้างคลังข้อมูลจะพบกับความล้มเหลวด้วยปัญหาที่แตกต่างกันออกไป ซึ่งความล้มเหลวอาจเกิดจากความไม่มีคุณภาพของข้อมูลหรือการเข้าถึงข้อมูลที่ไม่เหมาะสม เป็นต้น ถ้าเรามีการจัดการโปรเจคที่ดีจะช่วยให้การสร้างคลังข้อมูลประสบความสำเร็จได้

ดังนั้นเราจำเป็นต้องพิจารณาประเด็นเกี่ยวกับหลักการในการจัดการโครงสร้างที่ใช้กับการสร้างคลังข้อมูลที่จะช่วยให้โครงการของเราประสบความสำเร็จได้

หลักการในการจัดการโครงการสร้างคลังข้อมูล

เมื่อเราพิจารณาถึงหลักการในการจัดการโปรเจคการสร้างคลังข้อมูล เราควรจะต้องพิจารณาหลักการดังต่อไปนี้

Sponsorship

ไม่มีโครงการใดๆในการสร้างคลังข้อมูลที่จะประสบความสำเร็จได้ โดยปราศจากการสนับสนุนที่แข็งแกร่งจากผู้บริหาร

Project manager

การสร้างคลังข้อมูลอาจประสบความล้มเหลวได้ ถ้าในการสร้างนั้น ๆ มีผู้จัดการโปรเจคที่สนใจเทคนิคการสร้างคลังข้อมูลมากกว่าความต้องการของผู้ใช้และมากกว่ามุมมองทางธุรกิจ

Data quality

ในการสร้างคลังข้อมูลจะมี 3 สิ่งที่สำคัญคือ คุณภาพ คุณภาพ และคุณภาพ โดยคุณภาพของข้อมูลในคลังข้อมูลจะส่งผลต่อความเชื่อมั่นและการใช้งานคลังข้อมูลของผู้ใช้งาน

User requirement

ความต้องการจากผู้ใช้จะเป็นสิ่งบังคับการทำงานและกำหนดการของโปรเจค

Building for growth

เมื่อมีการติดตั้งและเริ่มใช้งานคลังข้อมูลจะทำให้มีจำนวนคิวรีที่ต้องเข้าถึงข้อมูลในคลังข้อมูลและจำนวนผู้ใช้เพิ่มขึ้นอย่างรวดเร็ว ถ้าการสร้างคลังข้อมูลไม่ได้คำนึงถึงจำนวนคิวรีและจำนวนผู้ใช้ที่เพิ่มขึ้นแล้วจะทำให้เกิดปัญหากับคลังข้อมูลได้

Realistic expectation

ในการสร้างคลังข้อมูลเราจะต้องสร้างความเข้าใจให้กับผู้ใช้ให้คาดหวังในสิ่งที่เป็นไปได้

Dimensional data modeling

การออกแบบแบบจำลองเชิงมิติของข้อมูล (dimension data model) ที่ดีเป็นสิ่งจำเป็นสำหรับการสร้างคลังข้อมูล

External data

การใช้เพียงข้อมูลภายในองค์กรสำหรับการสร้างคลังข้อมูลนั้นไม่เพียงพอ เราจำเป็นต้องใช้ข้อมูลจากแหล่งข้อมูลภายนอกด้วย

Project politic

การสร้างคลังข้อมูลอาจก่อให้เกิดอุปสรรคต่างๆมากมาย เมื่อต้องทำการยุ่งเกี่ยวกับผู้ใช้หลายกลุ่มและหลายระดับที่อาจมีความขัดแย้งกัน ดังนั้นเราควรจะต้องระมัดระวังให้มากเมื่อต้องติดต่อกับบุคคลที่มีความขัดแย้งกัน

Training

ในหลาย ๆ ครั้งผู้ใช้คลังข้อมูลจะไม่ทราบถึงวิธีการใช้คลังข้อมูล จากการไม่รู้หรือใช้งานไม่เป็นอาจทำให้ผู้ใช้ไม่ยอมใช้คลังข้อมูล ดังนั้นเราควรจะมีวิธีการให้ความรู้ที่ทำให้ผู้ใช้สามารถใช้งานคลังข้อมูลได้โดยง่าย

สัญญาณเตือนต่าง ๆ ในการสร้างคลังข้อมูล

เมื่อเราทำการสร้างหรือเริ่มใช้คลังข้อมูล เราควรจะต้องเฝ้าติดตามถึงสัญญาณเตือนต่าง ๆ ที่อาจจะทำให้เกิดความล้มเหลวขึ้นในการสร้างคลังข้อมูล ซึ่งเมื่อเราพบเจอสัญญาณเตือนหรือลางบอกเหตุแล้ว เราควรจะทำการวิเคราะห์ถึงปัญหาที่อาจจะเกิดขึ้น และทำการแก้ไขโดยเร่งด่วน ซึ่งโดยส่วนใหญ่ของการสร้างคลังข้อมูลอาจพบเจอลางบอกเหตุดังแสดงในรูปที่ 5-5

WARNING SIGN	INDICATION	ACTION
The Requirements Definition phase is well past the target date.	Suffering from "analysis paralysis."	Stop the capturing of unwanted information. Remove any problems by meeting with users. Set firm final target date.
Need to write too many in-house programs.	Selected third party tools running out of steam.	If there is time and budget, get different tools. Otherwise increase programming staff.
Users not cooperating to provide details of data.	Possible turf concerns over data ownership.	Very delicate issue. Work with executive sponsor to resolve the issue.
Users not comfortable with the query tools.	Users not trained adequately.	First, ensure that the selected query tool is appropriate. Then provide additional training.
Continuing problems with data brought over to the staging area.	Data transformation and mapping not complete.	Revisit all data transformation and integration routines. Ensure that no data is missing. Include the user representative in the verification process.

รูปที่ 5-5 สัญญาณเตือนในการสร้างคลังข้อมูล

ปัจจัยความสำเร็จของการสร้างคลังข้อมูล

เมื่อคลังข้อมูลถูกสร้างเสร็จอย่างสมบูรณ์ เรายังคงไม่สามารถบอกได้ว่าการสร้างคลังข้อมูลนั้นประสบความสำเร็จ แต่เราจะรู้ได้อย่างไรว่าคลังข้อมูลที่สร้างขึ้นนั้นประสบความสำเร็จหรือไม่? เราจะต้องตรวจสอบว่าคลังข้อมูลนั้นว่าประสบความสำเร็จจริงหรือไม่ โดยพิจารณาถึงสิ่งต่าง ๆ ที่สามารถบ่งบอกได้ เพื่อให้ได้คำตอบที่บ่งบอกว่าคลังข้อมูลที่สร้างขึ้นประสบความสำเร็จจริง เราจะต้องทำการตรวจสอบซึ่งก็คือการเฝ้าดู ROI ของคลังข้อมูลที่สร้างขึ้น ถ้าเราใช้วิธีนี้เราอาจจะต้องใช้เวลา 3 ปีหรือ 5 ปีเลยทีเดียว แต่ถ้าเราต้องการวัดความสำเร็จของการสร้างคลังข้อมูลที่สร้างขึ้นโดยใช้เวลาไม่นานนัก เราควรที่จะต้องใช้ตัวบ่งใช้อื่น ๆ โดยเราอาจจะทำการเฝ้าดูสิ่งต่อไปนี้ เพื่อทราบถึงความสำเร็จของการสร้างคลังข้อมูลเบื้องต้นได้

Queries and report—
จำนวนที่เพิ่มขึ้นของคิวรีและรายงานที่ถูกเรียกดูข้อมูลจากคลังข้อมูล

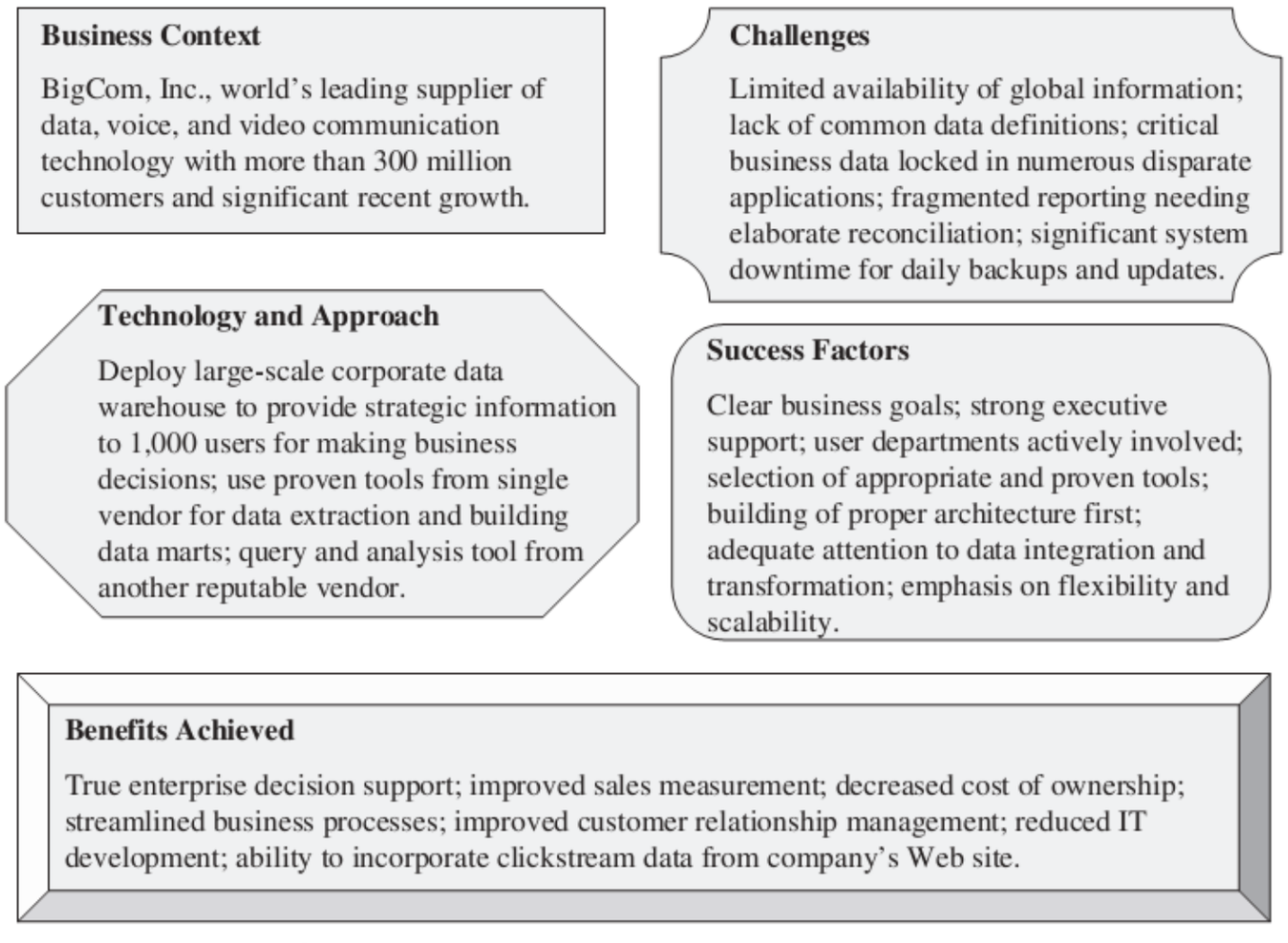
Queries types—
คิวรีมีความซับซ้อนมากขึ้น

Active uers—
จำนวนผู้ใช้ที่เพิ่มขึ้น

Usage—
เวลาที่ผู้ใช้ทำการใช้คลังข้อมูลเพิ่มขึ้น

ตัวอย่างโครงการสร้างคลังข้อมูลที่ประสบความสำเร็จ

การตรวจสอบปัจจัยความสำเร็จของคลังข้อมูลอาจช่วยให้เราทราบถึงความสำเร็จเบื้องต้นที่ได้รับ แต่ถ้าเราทำการวิเคราะห์ถึงรายละเอียดของสิ่งที่ทำให้การสร้างคลังข้อมูลนั้นประสบความสำเร็จอย่างแท้จริง จะทำให้เราเข้าใจถึงความสำเร็จได้ง่ายขึ้น ลองพิจารณากรณีตัวอย่างธุรกิจที่มีการสร้างคลังข้อมูลที่ประสบความสำเร็จเป็นอย่างดี ดังแสดงในรูปที่ 5-6 ซึ่งจากรูปเราจะทราบถึงปัจจัยของความสำเร็จและประโยชน์ที่ได้รับจากการสร้างคลังข้อมูลเพื่อสนับสนุนการดำเนินธุรกิจ



รูปที่ 5-6 การวิเคราะห์ความสำเร็จของคลังข้อมูล

SECTION 6

คำถามท้ายบท

1. จงอธิบายประเด็นสำคัญที่จะพิจารณาในขณะที่การวางแผนสำหรับการคลังข้อมูล
2. จงอธิบายความแตกต่างระหว่างวิธีการสร้างคลังข้อมูลแบบ top-down และ bottom-up คุณคิดว่าแบบไหนดีกว่ากัน เพราะอะไร
3. จงแจกแจงข้อดีของการใช้ซอร์ฟแวร์จากผู้ผลิตเพียงรายเดียวและจากผู้ผลิตหลายรายมารวมกัน
4. โดยส่วนใหญ่ของการสร้างคลังข้อมูล จะใช้วิธีการสร้างแบบใด
5. การประเมินความพร้อมในการสร้างคลังข้อมูล จะสามารถมีส่วนช่วยในแง่มุมใดบ้าง
6. จงแจกแจงและอธิบายขั้นตอนการพัฒนาคลังข้อมูลตามวงจรการพัฒนาคลังข้อมูล
7. จงแจกแจงหลักการในการจัดตั้งโครงการสร้างคลังข้อมูล
8. จงยกตัวอย่างสัญญาณเตือนที่อาจเกิดขึ้นระหว่างการสร้างคลังข้อมูล และวิธีในการแก้ไขปัญหา อย่างน้อย 3 ตัวอย่าง

CHAPTER 6

การกำหนดความต้องการทางธุรกิจ

SECTION 1

แผนการสอนประจำบท

วัตถุประสงค์ของบทเรียน

- ศึกษาเกี่ยวกับวิธีการในการกำหนดความต้องการของคลังข้อมูล
- ศึกษาเกี่ยวกับบทบาทหน้าที่ของมิติทางธุรกิจ
- เรียนรู้เกี่ยวกับแพคเกจข้อมูลและการใช้แพคเกจข้อมูลในการกำหนดความต้องการ
- ศึกษาเกี่ยวกับวิธีการในการเก็บรวมความต้องการจากผู้ใช้

เนื้อหาของบทเรียน

เนื้อหาในบทนี้จะประกอบด้วย การวิเคราะห์ข้อมูลมิติต่าง ๆ ทางธุรกิจ แพคเกจข้อมูล มิติทางธุรกิจ ลำดับชั้นและหมวดหมู่ของข้อมูลในมิติทางธุรกิจ วิธีในการรวบรวมความต้องการ ขอบเขตของความต้องการ

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารประกอบการสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

กิจกรรมการเรียน-การสอน

- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารประกอบการสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

การวัดและประเมินผล

- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

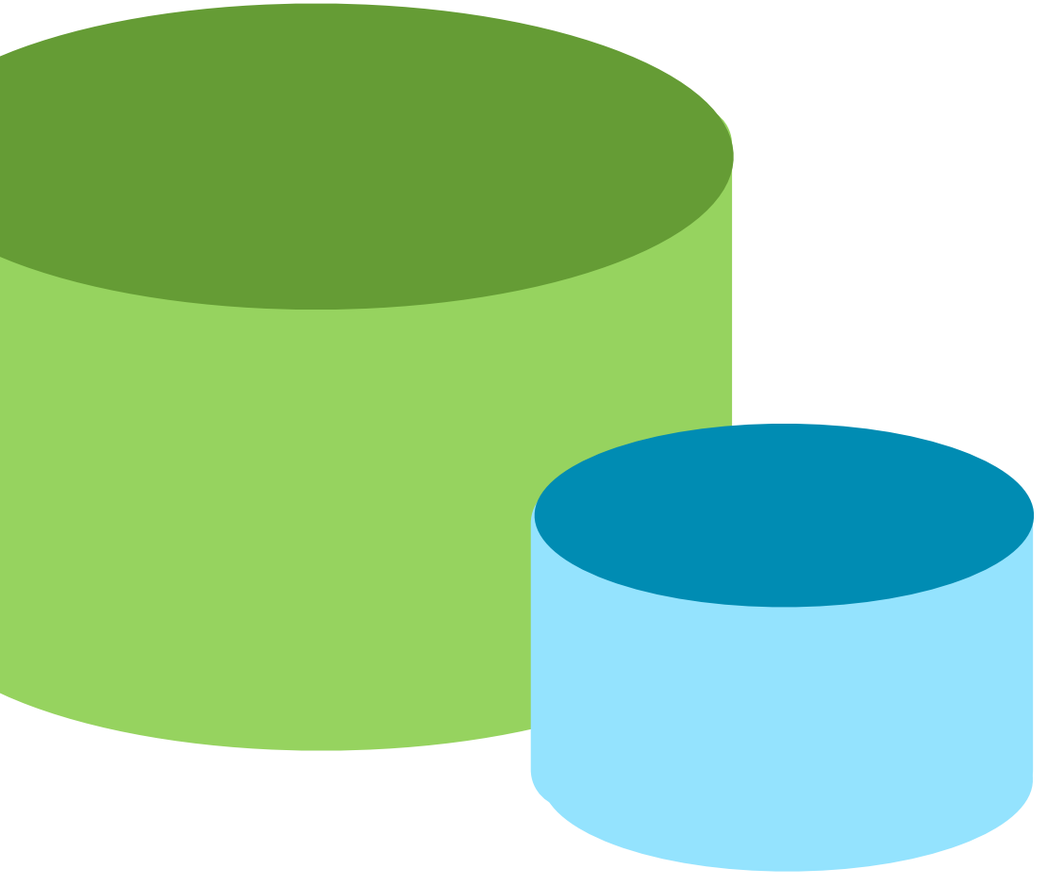
SECTION 2

บทนำ

จากบทก่อน ๆ หน้าจะทำให้เราทราบว่า **“คลังข้อมูล”** คือ ระบบที่สร้างหรือจัดเตรียมข้อมูล/สารสนเทศสำหรับการทำธุรกิจอย่างชาญฉลาดและเป็นสิ่งที่ช่วยแก้ปัญหาต่าง ๆ ในการดำเนินธุรกิจให้แก่ผู้ใช้งานคลังข้อมูล โดยเริ่มแรกของการสร้างคลังข้อมูลจะต้องมีการกำหนดความต้องการของผู้ใช้คลังข้อมูลที่จะเน้นย้ำที่ข้อมูลที่ผู้ใช้ต้องการจากคลังข้อมูลมากกว่าที่จะมองว่าเราจะทำการสร้างหรือค้นหาข้อมูลที่ผู้ใช้ต้องการได้อย่างไร ถ้าเรามีกระบวนการกำหนดความต้องการที่ไม่ดีจะส่งผลถึงคุณภาพของข้อมูลที่ได้จากคลังข้อมูลโดยตรง

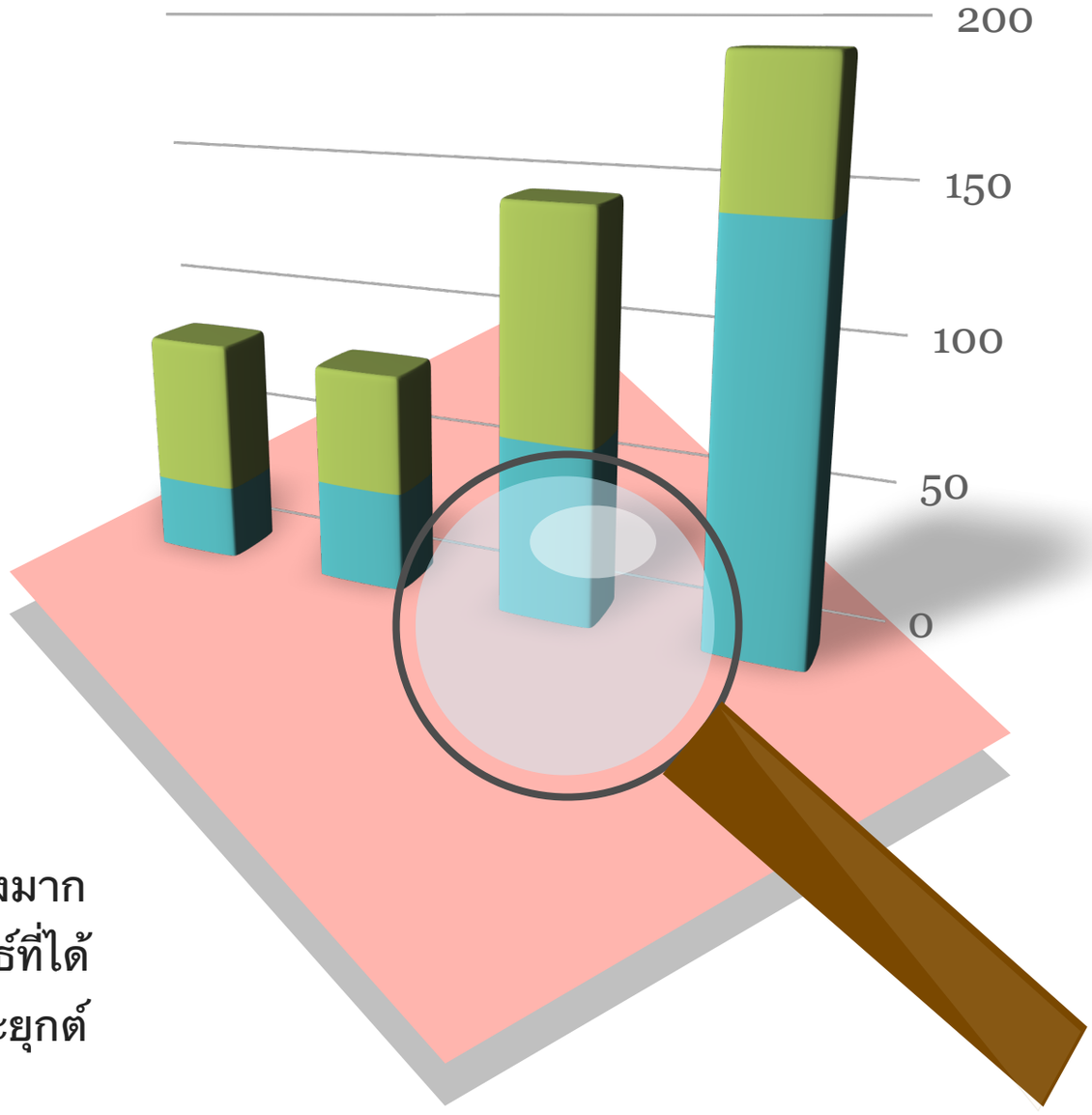
ดังนั้น ในการเก็บรวบรวมความต้องการจากผู้ใช้ เราควรจะเริ่มจากการเรียนรู้กระบวนการทำธุรกิจในแต่ละวัน (day-to-day business) จากนั้นค่อยพิจารณาถึงข้อมูล/รายงานเชิงกลยุทธ์ที่ผู้ใช้ต้องการ

ดังนั้น เนื้อหาในบทนี้จะเกี่ยวข้องกับการกำหนดความต้องการของผู้ใช้งาน รวมถึงกระบวนการในการจัดเก็บรวบรวมความต้องการซึ่งมีอยู่หลายวิธีด้วยกัน



SECTION 3

การวิเคราะห์ข้อมูลมิติต่างๆ ทางธุรกิจ



การสร้างคลังข้อมูลจะแตกต่างจากการสร้างระบบการดำเนินงานค่อนข้างมาก เนื่องจากทั้งสองระบบมีวัตถุประสงค์ของการสร้าง การทำงาน และผลลัพธ์ที่ได้จากแต่ละระบบ จากความแตกต่างของทั้งสองระบบ เราจะไม่สามารถประยุกต์ใช้วิธีที่ใช้ในการเก็บรวบรวมความต้องการของระบบการดำเนินงานได้โดยตรง โดยเราจะต้องทำการเปลี่ยนแปลงกระบวนการหรือแนวคิดในการกำหนดหรือรวบรวมความต้องการที่จะต้องพิจารณาสิ่งต่อไปนี้

ผู้ใช้ไม่สามารถคาดเดาการใช้ข้อมูลจากคลังข้อมูลได้

ในการทำความเข้าใจเกี่ยวกับการกำหนดหรือรวบรวมความต้องการจากผู้ใช้ ลองพิจารณาตัวอย่างการสร้างระบบการดำเนินงานของการสั่งซื้อสินค้า ซึ่งเราจะสามารถทำการเก็บรวบรวมความต้องการจากผู้ใช้โดยการสัมภาษณ์ผู้ที่ทำงานอยู่ในแผนกการสั่งซื้อสินค้าที่จะสามารถเล่าถึงฟังก์ชันการทำงานทั้งหมดที่ต้องการได้ เช่น การรับการสั่งซื้อสินค้า การตรวจสอบคลังสินค้า การตรวจสอบเครดิตของลูกค้า การต่อรองราคา กำหนดการส่งของ และอื่นๆ นอกเหนือจากการบ่งบอกถึงฟังก์ชันที่ต้องการแล้ว ผู้ใช้จะสามารถบอกถึงอินเทอร์เฟซที่ต้องการใช้แสดงผลลัพธ์ และบ่งบอกถึงรายงานต่าง ๆ ที่ต้องการสำหรับการสั่งซื้อสินค้า จากสิ่งที่ผู้ใช้บอกกล่าวจากการสัมภาษณ์จะทำให้เราจะสามารถทราบถึงความต้องการของผู้ใช้ว่าจะใช้ระบบที่สร้างขึ้น อย่างไร เมื่อไหร่ และที่ใด เพื่อสนับสนุนการทำงานในแต่ละวัน

จากการสัมภาษณ์ผู้ใช้เราจะทราบถึงความต้องการที่เกี่ยวข้องกับฟังก์ชันการทำงาน เนื้อหาของข้อมูลและรูปแบบการใช้งานระบบการดำเนินงานที่จะทำการสร้างขึ้น แต่สำหรับคลังข้อมูล ผู้ใช้อาจจะไม่สามารถระบุหรือกำหนดความต้องการได้อย่างชัดเจน ผู้ใช้อาจไม่สามารถกำหนดถึงข้อมูล/สารสนเทศ ที่เป็นที่ต้องการ และไม่สามารถแสดงถึงวิธีการใช้/ดำเนินการกับข้อมูลสารสนเทศได้อย่างสมบูรณ์ ซึ่งผู้ใช้โดยส่วนใหญ่มักจะไม่คุ้นชินกับการให้ข้อมูล/ความต้องการสำหรับการสร้างคลังข้อมูล ดังนั้นเมื่อความต้องการจากผู้ใช้ไม่ชัดเจน เราจะสามารถแก้ปัญหานี้ได้อย่างไร?—วิธีการหนึ่งที่ค้อนข้างจะปลอดภัยก็คือ การเก็บข้อมูลทุก ๆ ชิ้นที่เราคิดว่ามีประโยชน์สำหรับผู้ใช้ไว้ในคลังข้อมูล แต่อย่างไรก็ดีสิ่งที่เราคิดอาจจะไม่ตรงความต้องการของผู้ใช้ก็เป็นได้ เนื่องจากเราขาดความรู้ ความเข้าใจในกระบวนการดำเนินธุรกิจ

นอกจากนั้นเราอาจจะทำการเก็บข้อมูลที่เกี่ยวข้องกับการดำเนินธุรกิจทั้งหมดทั่วทั้งองค์กร โดยทำการตรวจสอบเกี่ยวกับการปฏิบัติการที่ดีที่สุดในธุรกิจนั้น ๆ หรือเราอาจจะทำการรวบรวมกฎเกณฑ์ทางธุรกิจบางอย่างที่สามารถชี้นำในการตัดสินใจแบบวันต่อวัน เราอาจจะหาวิธีการที่จะสามารถพัฒนาและขายสินค้าเพื่อช่วยเหลือในการกำหนดความต้องการในการใช้คลังข้อมูล จากวิธีการที่กล่าวมาทั้งหมดข้างต้น จะเป็นวิธีการพื้นฐานเท่านั้นซึ่งอาจยังไม่เพียงพอต่อการระบุถึงรายละเอียดของความต้องการที่ผู้ใช้ต้องการได้

ดังนั้นเราจึงจำเป็นต้องพิจารณาถึงปัจจัยอื่นๆ เพื่อช่วยให้สามารถเก็บรวบรวมความต้องการให้ตรงกับความต้องการจริงมากที่สุด

นิยามของข้อมูลทางธุรกิจ

นิยามของข้อมูลทางธุรกิจ

ถึงแม้ว่าผู้ใช้จะไม่สามารถอธิบายถึงความต้องการให้กับผู้สร้างคลังข้อมูลได้ทั้งหมด แต่พวกเขาสามารถบอกถึงรายละเอียดความเข้าใจเกี่ยวกับสิ่งที่พวกเค้าคิดเกี่ยวกับการดำเนินธุรกิจได้ ผู้ใช้อาจสามารถบอกถึงมาตรวัดความสำเร็จที่สำคัญและจำเป็นต่อการทำธุรกิจ สำหรับแต่ละแผนกของบริษัท พนักงานในแผนกนั้นๆ จะสามารถบอกให้เราทราบถึงสิ่งที่ใช้วัดความสำเร็จของแต่ละแผนกได้ หรือพนักงานจะสามารถบอกได้ว่าพวกเขาจะทำการรวมข้อมูลชิ้นเล็กๆ ให้เป็นข้อมูลเพื่อส่งเสริมการตัดสินใจเชิงกลยุทธ์ได้อย่างไร ซึ่งโดยส่วนใหญ่แล้วเราจะทำการสัมภาษณ์ผู้จัดการแต่ละแผนกถึงความต้องการหรือแนวคิดเกี่ยวกับมิติของการทำธุรกิจ (Business dimensions)

Marketing Vice President

How much did my new product generate
month by month, in the southern division, by customer demographic, by sales office, relative to the previous version, and compared to plan?

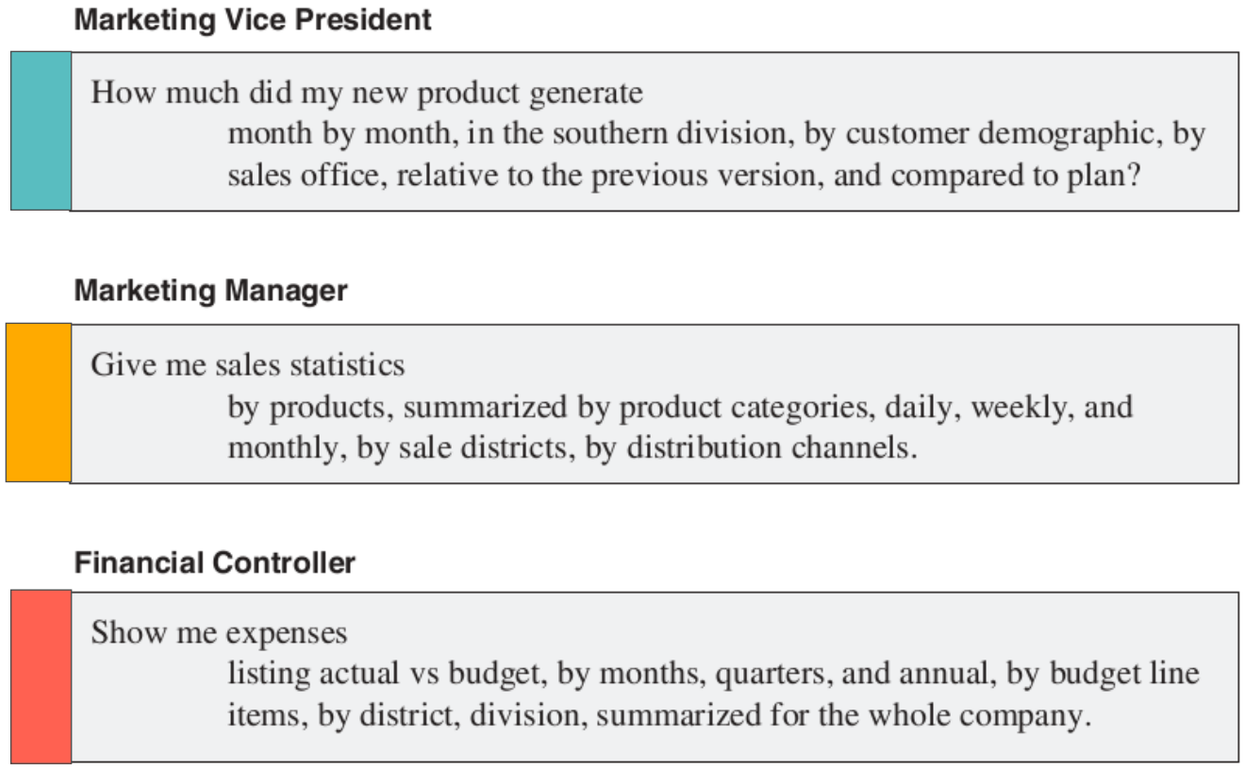
Marketing Manager

Give me sales statistics
by products, summarized by product categories, daily, weekly, and monthly, by sale districts, by distribution channels.

Financial Controller

Show me expenses
listing actual vs budget, by months, quarters, and annual, by budget line items, by district, division, summarized for the whole company.

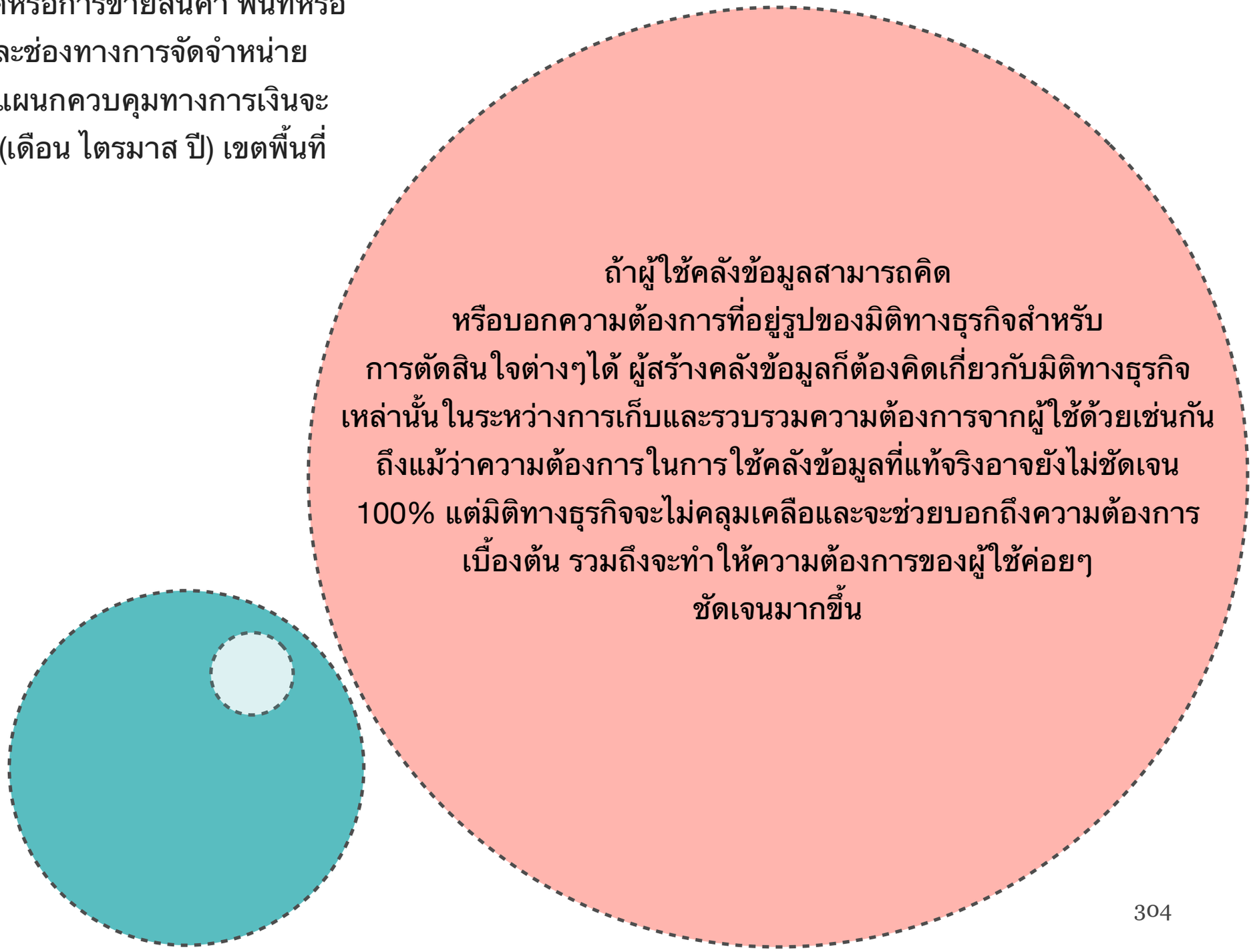
รูปที่ 6-1 สิ่งที่ผู้ใช้ในตำแหน่งต่าง ๆ คิดเกี่ยวกับมิติต่าง ๆ ทางธุรกิจ

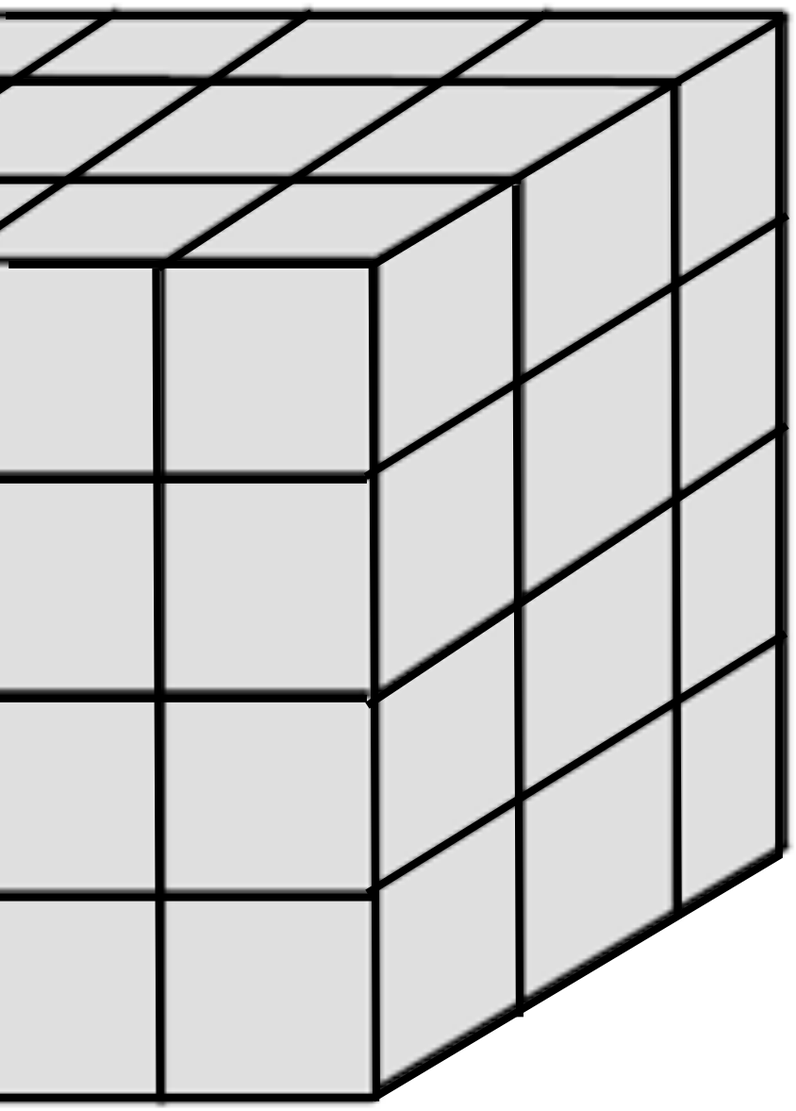


รูปที่ 6-1 สิ่งที่ผู้ใช้ในตำแหน่งต่าง ๆ คิดเกี่ยวกับมิติต่าง ๆ ทางธุรกิจ

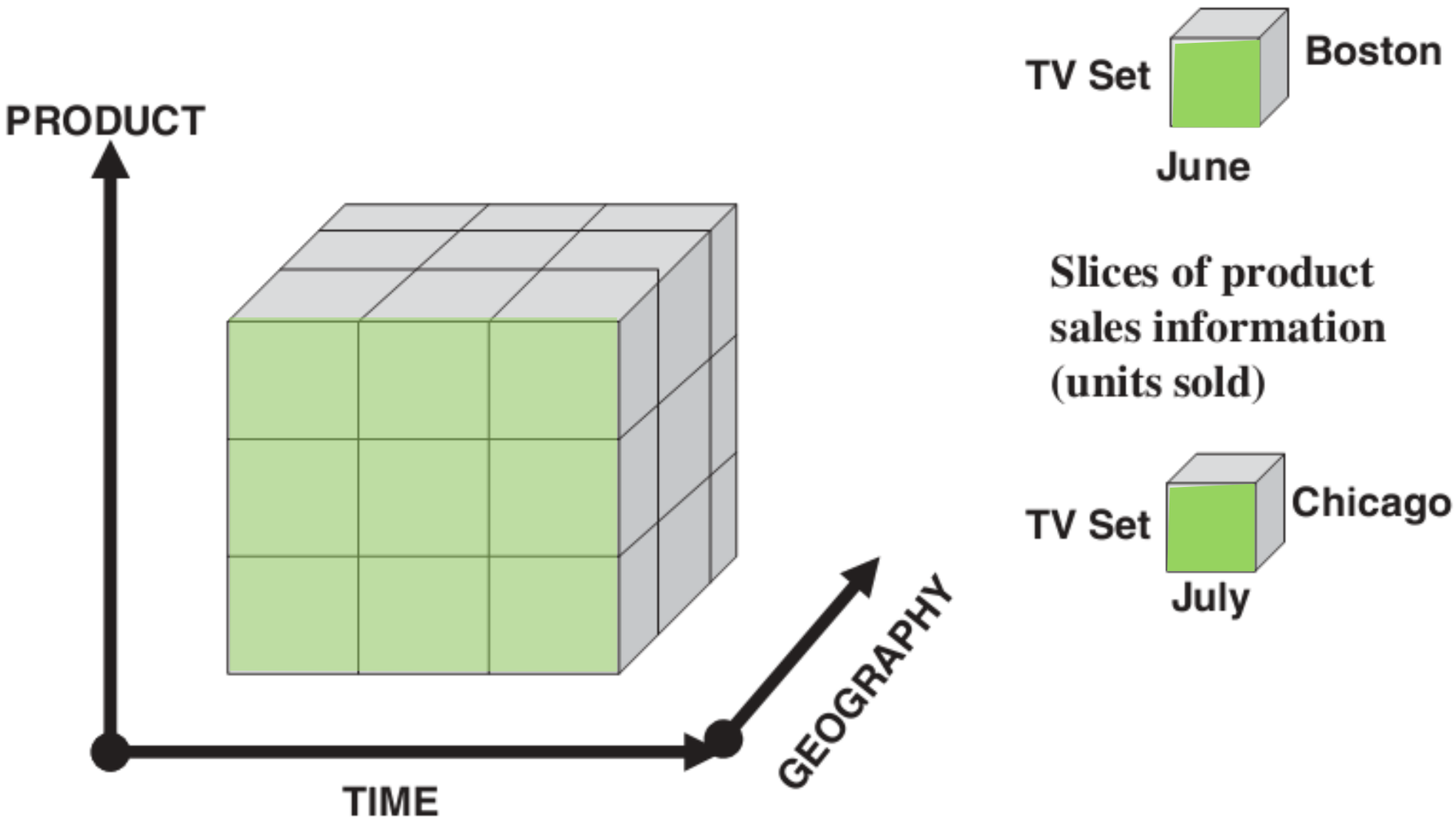
รูปที่ 6-1 แสดงถึงชนิดของคำถามหรือข้อมูลต่างๆ ที่ผู้จัดการในแต่ละระดับหรือแต่ละแผนกมักถามเพื่อทำการตัดสินใจ ซึ่งจากรูป รองประธานฝ่ายการตลาด (Marketing vice president) จะสนใจเกี่ยวกับรายได้หรือยอดขายของสินค้าใหม่ที่เพิ่งว่างจำหน่าย โดยอาจจะสนใจเกี่ยวกับยอดขายในแต่ละเดือน ยอดขายในแต่ละพื้นที่ หรือยอดขายในแต่ละตัวแทนจำหน่าย ความสัมพันธ์และการเปรียบเทียบยอดขายระหว่างสินค้าใหม่และสินค้าเก่าเพื่อใช้ในการวางกลยุทธ์ต่างๆ เป็นต้น สิ่งเหล่านี้จะเป็นมิติของการทำธุรกิจที่รองประธายฝ่ายการตลาดจะต้องรับผิดชอบในการทำงาน

ในทำนองเดียวกัน มิติทางธุรกิจของผู้จัดการฝ่ายการตลาดจะเกี่ยวข้องกับรายการสินค้า หมวดหมู่ของสินค้า ช่วงเวลา (วัน อาทิตย์ เดือน) ของการผลิตหรือการขายสินค้า พื้นที่หรือที่ตั้งของห้างร้านที่ขายสินค้า และช่องทางการจัดจำหน่ายสินค้า สำหรับมิติทางธุรกิจของแผนกควบคุมทางการเงินจะเกี่ยวข้องกับ งบประมาณ เวลา (เดือน ไตรมาส ปี) เขตพื้นที่ และแผนก





ลองพิจารณาการเก็บข้อมูลมิติทางธุรกิจ ดังแสดงในรูปที่ 6-2 ที่จะแสดงถึงการวิเคราะห์ยอดขายกับมิติทางธุรกิจ 3 มิติด้วยกัน คือ รายการสินค้า เวลา และพื้นที่ที่มีการขายสินค้า โดยทั้ง 3 มิติที่ได้กล่าวข้างต้นจะสามารถแสดงผลอยู่ในรูปแบบของกราฟสามมิติที่มีลักษณะเป็นลูกบาศก์ได้ โดยแต่ละลูกบาศก์จะสามารถบอกถึงยอดขายต่อรายการสินค้าต่อเวลาและต่อพื้นที่ที่มีการขายสินค้า จากรูปที่ 6-2 เราจะเห็นว่ามิติทางธุรกิจจะประกอบไปด้วยสามมิติด้วยกัน แต่เมื่อไรก็ตามที่จำนวนมิติเพิ่มขึ้นเราจะสามารถใช้แนวความคิดของการขยายมิติให้เป็นหลายมิติมากขึ้นและทำการแสดงผลมิติที่มีการขยายเหล่านั้นให้อยู่ในรูปของลูกบาศก์หลายมิติ ที่เรียกว่า **“hypercubes”**



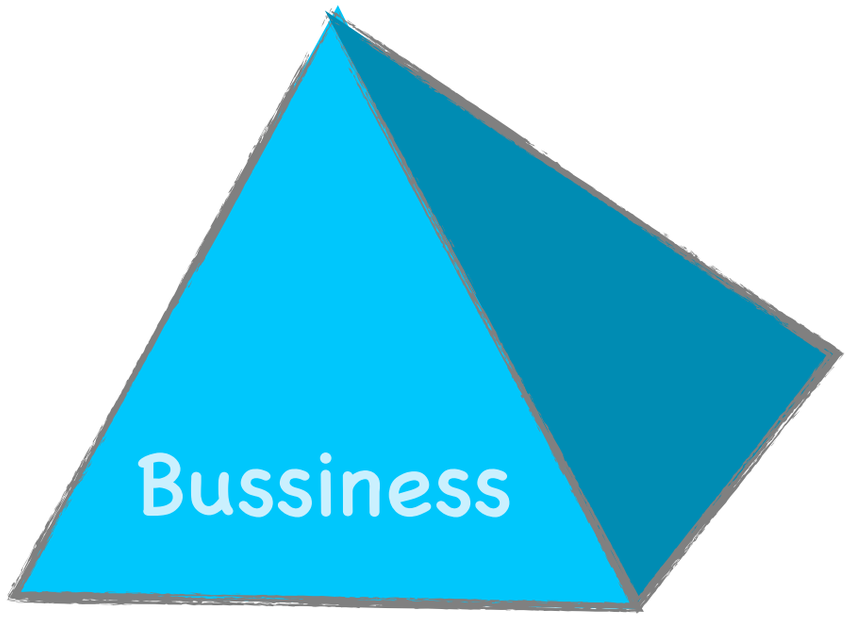
รูปที่ 6-2 ลักษณะข้อมูลที่เป็นมิติทางธุรกิจ

ตัวอย่างมิติทางธุรกิจ

แนวคิดเกี่ยวกับมิติ/มุมมองทางธุรกิจนั้นเป็นแนวคิดพื้นฐานเกี่ยวกับความต้องการเบื้องต้นสำหรับการสร้างคลังข้อมูล ดังนั้น เพื่อให้ทราบถึงแนวคิดของมิติ/มุมมองทางธุรกิจ ลองพิจารณาตัวอย่างที่มีความแตกต่างกันดังรูปที่ 6-3 ซึ่งจะประกอบไปด้วยธุรกิจ 4 ประเภทด้วยกัน คือ

1. ธุรกิจซุปเปอร์มาร์เก็ตจะใช้จำนวนชิ้นสินค้าที่ขายได้เป็นมาตรวัดผลสัมฤทธิ์ซึ่งจะเกี่ยวเนื่องกับมิติทางธุรกิจที่เป็น hypercube 4 มิติด้วยกัน คือ เวลา โปรโมชั่น รายการสินค้า และสาขาห้างร้าน

2. ธุรกิจประกันภัยจะมีมิติทางธุรกิจแตกต่างจากธุรกิจอื่นๆที่มีลักษณะการขายสินค้า โดยที่ธุรกิจประกันภัยส่วนใหญ่จะวิเคราะห์การเคลมประกัน (เนื่องจากเป็นค่าใช้จ่ายของบริษัทที่ก่อให้เกิดผลกำไรหรือขาดทุน) ที่เกี่ยวข้องกับการเคลมกับตัวแทน การเคลมแต่ละครั้ง ช่วงเวลา บริษัทประกัน กลยุทธ์/นโยบายของแต่ละประกัน และสถานะของการเคลม



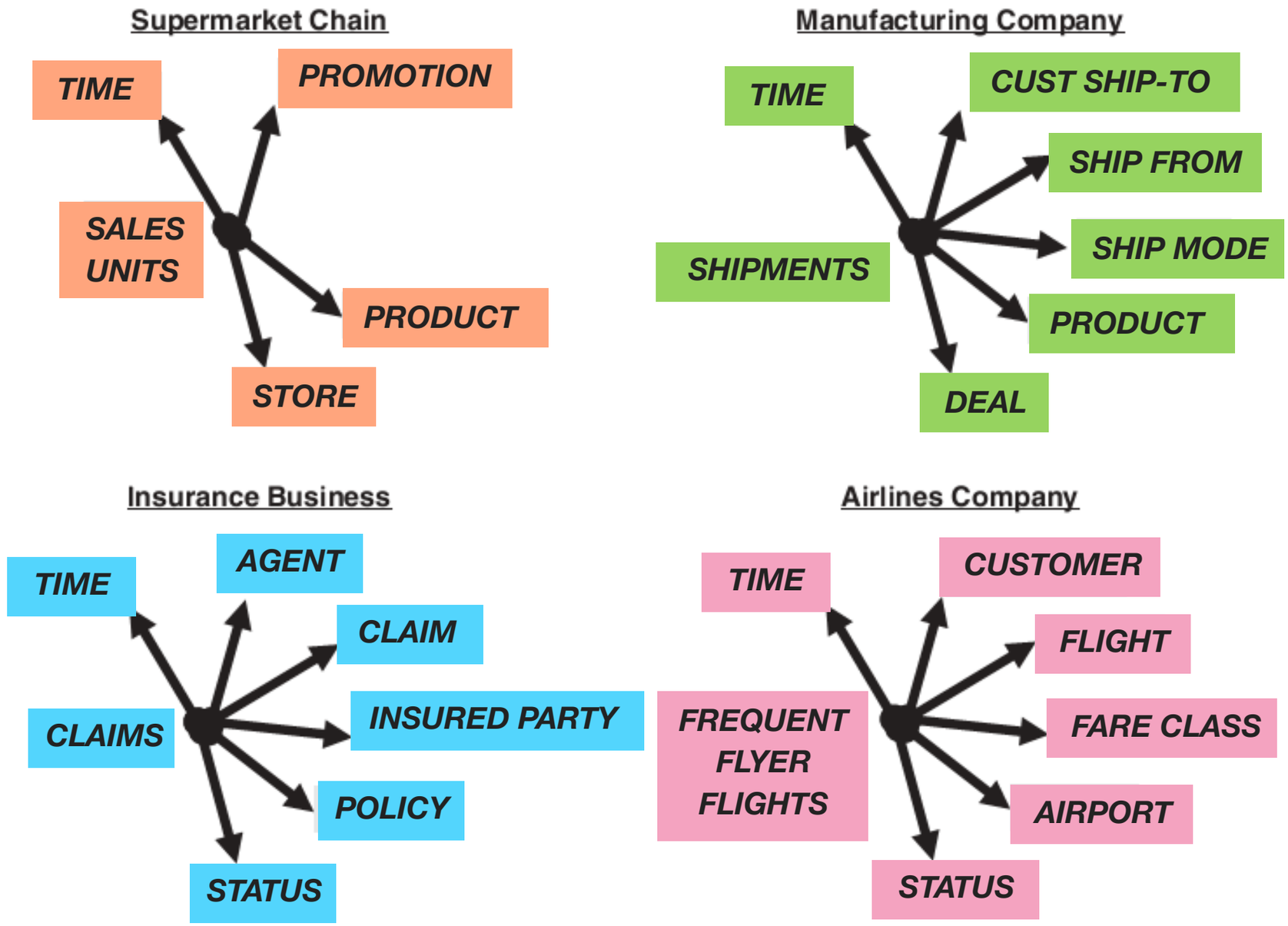
3

ธุรกิจสายการบินจะพิจารณาข้อมูลต่างๆ ที่เกี่ยวกับการบิน ซึ่งจะมีมิติทางธุรกิจหลายมิติด้วยกัน เช่น เวลา ลูกค้า เที่ยวบิน fare class สนามบิน และสถานะความถี่ในการบิน

4

ธุรกิจการขนส่งสินค้าสำหรับบริษัทผลิตสินค้าจะมีมิติทางธุรกิจที่แตกต่างจากธุรกิจอื่นๆ ซึ่งมิติทางธุรกิจที่ใช้ในการวิเคราะห์จะเกี่ยวข้องกับแกนเวลา ที่อยู่ปลายทาง ที่อยู่ต้นทาง โหมดของการขนส่ง รายการสินค้า และข้อเสนอพิเศษอื่นๆ

จากตัวอย่างทั้งสี่ธุรกิจข้างต้น เราจะเห็นว่ามิติเชิงธุรกิจของแต่ละธุรกิจจะมีความแตกต่างกันไปตามหัวข้อที่ต้องการวิเคราะห์ แต่อย่างไรก็ตามจะมีมิติหนึ่งที่มีอยู่เหมือนกันในทุกๆธุรกิจ นั่นคือ มิติของเวลา ซึ่งทุก ๆ ธุรกิจจะวิเคราะห์ผลสัมฤทธิ์โดยขึ้นอยู่กับแกนเวลาเป็นส่วนใหญ่

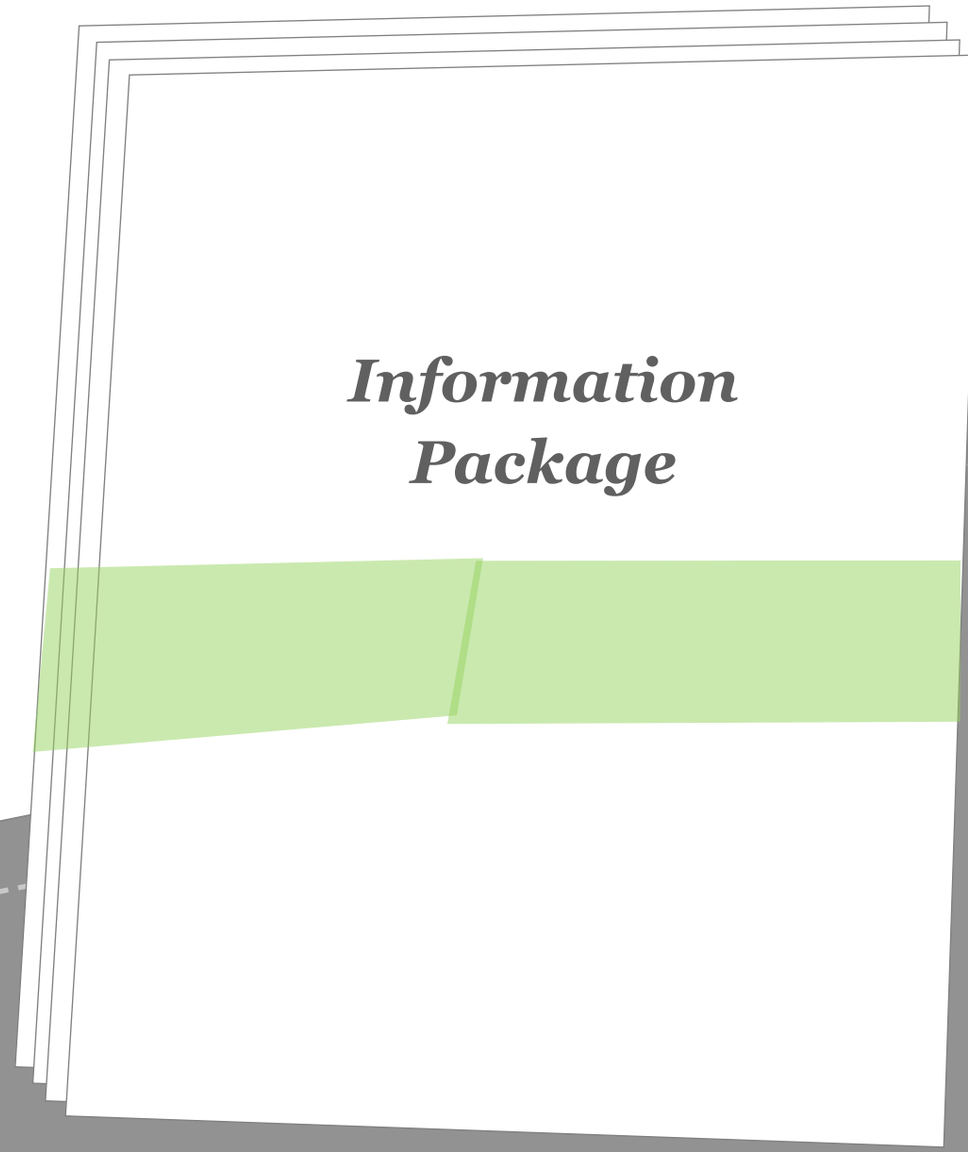


รูปที่ 6-3 ตัวอย่างมิติทางธุรกิจ

SECTION 4

แพคเกจข้อมูล

ในส่วนนี้จะเป็นการเสนอแนวคิดสำหรับการกำหนดและจัดเก็บความต้องการสำหรับการสร้างคลังข้อมูล แนวคิดที่จะเสนอจะช่วยให้ทราบถึงความเข้าใจในด้านต่าง ๆ ความคิดที่คลุมเคลือ และการแสดงความคิดเห็นระหว่างการเก็บรวบรวมความต้องการต่าง ๆ ซึ่งจะถูกเก็บไว้เป็น**แพคเกจข้อมูล *(Information package)*** ต่างๆ เพื่อใช้ในการพัฒนาคลังข้อมูลสืบไป



การกำหนดความต้องการที่ยังไม่สมบูรณ์

จากที่กล่าวก่อนหน้า ผู้ใช้อาจไม่สามารถอธิบายถึงสิ่งที่พวกเขาคาดหวังว่าจะเห็นหรือได้รับจากคลังข้อมูล ที่ซึ่งจะทำให้ผู้สร้างคลังข้อมูลจะไม่สามารถระบุชิ้นส่วนของข้อมูลที่ต้องเก็บไว้ในคลังข้อมูล เราจึงต้องการวิธีการหรือแนวคิดใหม่ ๆ ในการจัดเก็บและรวบรวมความต้องการต่าง ๆ ที่แตกต่างจากวิธีการเก็บรวบรวมความต้องการของระบบการดำเนินงานที่จะถามผู้ใช้ถึง ฟังก์ชันการทำงาน หน้าจอ และรายงานต่าง ๆ ที่ผู้ใช้ต้องการ โดยในการสร้างคลังข้อมูลนั้นเราไม่สามารถเริ่มต้นด้วยการออกแบบโครงสร้างของข้อมูลว่าประกอบไปด้วยข้อมูลอะไรบ้าง เราจะสามารถทำได้เพียงถามถึงมิติทางธุรกิจและการวิเคราะห์/วัดผลสัมฤทธิ์ของการดำเนินงานต่างๆ

วิธีการใหม่สำหรับการกำหนดความต้องการสำหรับคลังข้อมูลจะตั้งอยู่บนพื้นฐานของมิติทางธุรกิจ โดยวิธีการนี้จะทำการรวมมาตรวัดต่าง ๆ และมิติทางธุรกิจเข้าด้วยกัน ซึ่งจะช่วยให้เราสามารถเก็บมาตราวัดผลสัมฤทธิ์และมิติต่างๆ ที่เกี่ยวข้องไว้ในคลังข้อมูลได้ ซึ่งทั้งสองสิ่งนี้เราจะถูกสร้างเป็นแพคเกจข้อมูลตามหัวข้อที่ผู้ใช้สนใจ ลองพิจารณาตัวอย่างแพคเกจข้อมูลสำหรับการวิเคราะห์ของขายในธุรกิจหนึ่ง

ดังแสดงในรูปที่ 6-4 เนื่องจากหัวข้อของการเก็บข้อมูลจะเกี่ยวกับยอดขาย ดังนั้นมาตรวัดที่สะท้อนความเป็นจริงหรือมาตรวัดที่น่าสนใจจะเกี่ยวข้องกับยอดขายจริง ยอดขายที่คาดหวัง และการขายตามงบประมาณ ในส่วนของมิติทางธุรกิจจะประกอบไปด้วย เวลา สถานที่ รายการสินค้า และกลุ่มอายุของลูกค้า ซึ่งแต่ละมิติจะมีข้อมูลที่เป็นลำดับชั้นหรือเป็นระดับ เช่น มิติเวลาที่มีลำดับชั้นจะมีข้อมูลที่มีรายละเอียดแตกต่างกันตั้งแต่ปีลดหลั่นไปเรื่อย ๆ จนถึงแต่ละวันเลยทีเดียว หรือในอีกกรณีหนึ่งของแกนเวลาอาจมีลำดับชั้นเป็น ไตรมาส เดือน และอาทิตย์ ตามลำดับ ซึ่งระดับหรือลำดับชั้นของแต่ละมิติจะถูกแสดงรายละเอียดไว้ในไดอะแกรมสำหรับแพคเกจข้อมูลด้วย

Information Subject: Sales Analysis

Dimensions

Time Periods	Locations	Products	Age Groups		
Year	Country	Class	Group 1		

Hierarchies

Measured Facts: Forecast Sales, Budget Sales, Actual Sales

รูปที่ 6-4 ตัวอย่างแพคเกจข้อมูล

ดังนั้นเป้าหมายในการเก็บรวบรวมความต้องการคือการรวบรวมแพคเกจข้อมูลสำหรับทุกๆ หัวข้อที่จะทำการสร้างคลังข้อมูล เมื่อเราได้แพคเกจข้อมูลเราจะสามารถทำงานกระบวนการต่างๆ ต่อไปได้ เช่น

- กำหนดหัวข้อ/หัวเรื่องที่พบบ่อย
- กำหนดมาตรวัดทางธุรกิจที่เป็นกุญแจสำคัญ
- กำหนดว่าจะแสดงข้อมูลอย่างไร
- กำหนดว่าผู้ใช้จะสามารถสรุปรวบยอดข้อมูลได้อย่างไร
- กำหนด/ระบุปริมาณข้อมูลสำหรับแต่ละการวิเคราะห์หรือคิวรีจากผู้ใช้
- กำหนดว่าจะสามารถเข้าถึงข้อมูลได้อย่างไร
- จัดทำ/แจกแจงรายละเอียดของข้อมูล
- ประมาณขนาดของคลังข้อมูล
- ระบุ/กำหนดความถี่ในการอัพเดทข้อมูล
- การตรวจสอบ/ทำให้แน่ใจว่าข้อมูลจะถูกรวบรวมเป็นแพคเกจได้อย่างไร

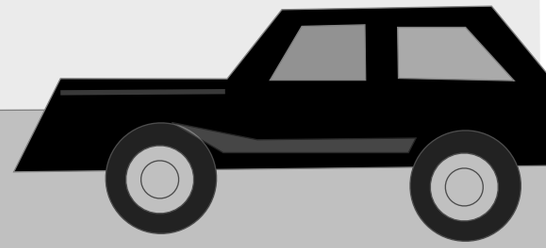
มิติทางธุรกิจ

เมื่อเราทราบถึงรายละเอียดของมิติต่าง ๆ ทางธุรกิจ ซึ่งเป็นวิธีการหนึ่งในการกำหนดความต้องการให้เป็นรูปร่างขึ้นและสามารถทราบถึงความต้องการพื้นฐานที่จะนำไปวิเคราะห์และสร้างคลังข้อมูลต่อไป แต่ก่อนอื่นเราต้องทำการพิจารณาหรือระบุมิติทางธุรกิจและระดับชั้นของมิตินั้นๆ โดยต้องการเลือกมิติให้ดีที่สุดและมีความสอดคล้องกับมาตราวัดประสิทธิภาพของการดำเนินธุรกิจมากที่สุด

ลองพิจารณาตัวอย่างมิติทางธุรกิจของบริษัทผู้ผลิตรถยนต์ ซึ่งต้องการที่จะวิเคราะห์ยอดขายรถยนต์ของบริษัท ดังนั้นทางบริษัทต้องการที่จะสร้างคลังข้อมูลที่อนุญาตให้ผู้ใช้สามารถวิเคราะห์ยอดขายในแง่มุมต่างๆ

จากความต้องการดังกล่าวเราสามารถวิเคราะห์มิติทางธุรกิจที่เกี่ยวข้องกับการวิเคราะห์ยอดขายรถยนต์ได้ดังนี้

- มิติของสินค้า
- มิติของผู้ใช้
- มิติของลูกค้า
- มิติของการจ่ายเงิน
- มิติแกนเวลา



มิติของสินค้า (Product) ช่วยให้ผู้ใช้ทราบถึงยอดขายรถยนต์แต่ละรุ่น รุ่นใดขายดี ขายไม่ดี

มิติของผู้ใช้ (Dealer) จะช่วยให้เราทราบได้ว่าผู้ค้าใดสามารถขายรถยนต์ได้จำนวนมาก

มิติของลูกค้า ซึ่งจะแสดงถึงใครเป็นซื้อรถยนต์จากบริษัทเราบ้างซึ่งมิตินี้จะทำให้เราทราบถึงอาชีพ ที่อยู่ของลูกค้า และอื่นๆ

มิติของการจ่ายเงิน จะช่วยบอกข้อมูลเกี่ยวกับการจ่ายเงินซื้อรถยนต์ของลูกค้า เช่น การจ่ายเงินทั้งหมดหรือการผ่อนจ่าย เป็นต้น ที่ซึ่งจะช่วยให้ผู้ใช้ทราบว่าการกู้ยืมเงินจากสถาบันการเงินมีผลต่อการซื้อรถยนต์หรือไม่

มิติแกนเวลา ทุก ๆ การวิเคราะห์จะมีเวลาเข้ามาเกี่ยวข้องด้วยเสมอเช่น ผลการขายรถยนต์ในแต่ละเดือน แต่ละไตรมาส เป็นต้น

เพื่อทำความเข้าใจเกี่ยวกับมิติทางธุรกิจมากขึ้น ลองพิจารณาอีกตัวอย่างหนึ่งที่เกี่ยวกับ**ธุรกิจโรงแรม**ที่ต้องการวิเคราะห์ข้อมูลการเข้าพักของลูกค้าในแต่ละเวลาของโรงแรม เราต้องการที่จะวิเคราะห์ข้อมูลการเข้าพักของลูกค้าในสาขาหนึ่งๆ และชนิดของห้องหนึ่งๆ จากความต้องการในการวิเคราะห์ดังกล่าวจะสามารถกำหนดมิติทางธุรกิจได้อย่างน้อย 3 มิติด้วยกัน คือ มิติของสาขาโรงแรม มิติของห้องพักซึ่งรวมถึงข้อมูลชนิดของห้องพัก และมิติแกนเวลาที่เป็นมิติที่จำเป็นในทุกการสร้างคลังข้อมูล เป็นต้น

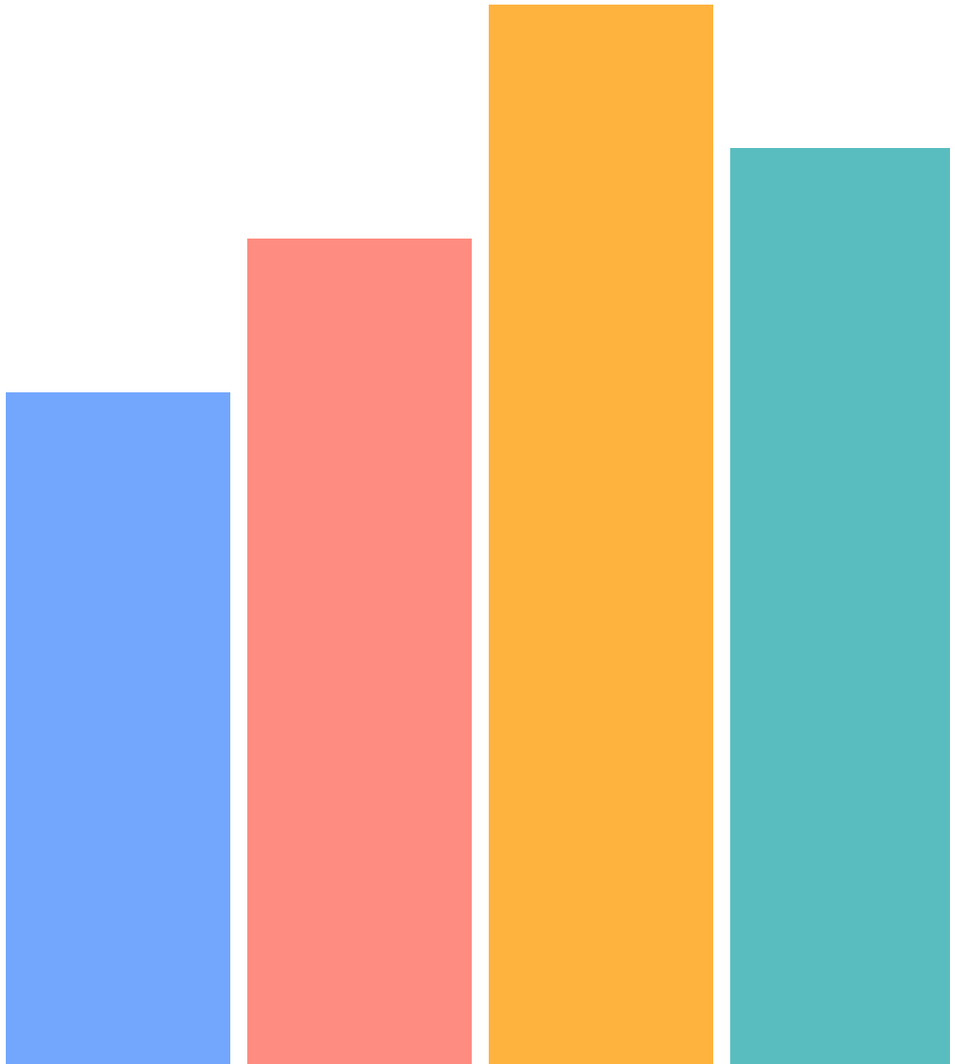
ลำดับชั้นและหมวดหมู่ของข้อมูลในมิติทางธุรกิจ

เมื่อเราทำการวิเคราะห์เกี่ยวกับมาตรวัดความสำเร็จทางธุรกิจกับมิติทางธุรกิจ สิ่งแรกที่เห็นจะเป็นข้อมูลเชิงตัวเลขที่เป็นผลสรุปและข้อมูลเชิงตัวเลขตามระดับความละเอียดของข้อมูล ซึ่งจากข้อมูลทั้งที่เป็นผลสรุปและข้อมูลที่แสดงถึงรายละเอียดต่างๆ ผู้ใช้สามารถเรียกดูข้อมูลได้โดยการท่องไปตามลำดับขั้นความละเอียด (hierarchical levels) ของมิติต่างๆ ตัวอย่างเช่น

ตอนเริ่มต้นผู้ใช้อาจต้องการเรียกดูข้อมูลยอดขายสินค้าทั้งปีของบริษัท ข้อมูลนี้จะเป็นข้อมูลที่เป็นผลสรุป ต่อมาผู้ใช้อาจต้องการข้อมูลที่มีรายละเอียดเพิ่มขึ้นหรือลึกมากขึ้น กล่าวคือ ข้อมูลยอดขายของแต่ละไตรมาสหรือไตรมาสหนึ่ง ๆ จากนั้นผู้ใช้อาจเจาะลึกลงไปที่ยอดขายของแต่ละเดือน หรือเดือนหนึ่งๆ หรืออาจเป็นข้อมูลยอดขายในแต่ละวัน หรือวันหนึ่งๆ เป็นต้น

จากตัวอย่างเราจะเห็นว่ามิติแกนเวลาส่งผลถึงความละเอียดของมาตรวัดความสำเร็จ ซึ่งจากความละเอียดของแกนเวลาจะทำให้ผู้ใช้ได้ข้อมูลที่เกี่ยวกับมาตรวัดความสำเร็จที่แตกต่างกันตามความละเอียดหรือลำดับชั้นของแกนเวลาที่ประกอบไปด้วย ปี ไตรมาส เดือน และวัน เป็นต้น

ซึ่งลำดับชั้นมิติแกนเวลาจะเป็นเหมือนเส้นทางในการเข้าถึงข้อมูลแบบเจาลึกลงรายละเอียด (drilling down) และแบบสรุปผลของข้อมูล (rolling up) ตัวอย่างเช่น ถ้าเรากำลังทำการเรียกดูข้อมูลยอดขายในแต่ละวันซึ่งเป็นข้อมูลที่ละเอียดที่สุดของแกนเวลา เราสามารถทำการเรียกดูข้อมูลที่มีรายละเอียดน้อยกว่า เช่น ยอดขายรายเดือน (rolling up) เป็นต้น แต่ถ้าเรากำลังทำการเรียกดูข้อมูลยอดขายในแต่ละไตรมาสหรือไตรมาสหนึ่งๆ เราจะสามารถทำการเรียกดูข้อมูลแบบเจาะลึกลงไปในแต่ละไตรมาสได้ (drilling down) ซึ่งเราจะได้ผลยอดขายแต่ละเดือนได้ เป็นต้น



นอกจากลำดับชั้นของข้อมูล (ความละเอียดของข้อมูล) ในแต่ละมิติธุรกิจแล้ว แต่ละมิติทางธุรกิจยังอาจประกอบไปด้วยข้อมูลที่บ่งบอกถึงหมวดหมู่ของข้อมูล (categories) หลายหมวดหมู่ด้วยกัน ตัวอย่างเช่น ในมิติแกนเวลาอาจมีแอทริบิวหรือส่วนประกอบหนึ่งๆ ที่บ่งบอกว่าวันหนึ่งๆนั้นเป็นวันหยุดหรือไม่ ซึ่งส่วนประกอบของข้อมูลเราจะสามารถแบ่งกลุ่มข้อมูลในแกนเวลาได้เป็น 2 กลุ่มคือ วันที่เป็นวันหยุดและวันธรรมดา ที่ซึ่งจะทำให้ผู้ใช้สามารถวิเคราะห์ยอดขายในกลุ่มของวันหยุดและวันธรรมดาได้

นอกจากนี้ยังสามารถที่จะเปรียบเทียบความแตกต่างของยอดขายในวันหยุดและวันธรรมดาได้อีกด้วย ลองพิจารณาอีกตัวอย่างหนึ่งนั่นคือ มิติรายการสินค้า ซึ่งเราอาจมีการวิเคราะห์ยอดขายต่อชนิดของแพคเกจสินค้า ซึ่งจากตัวอย่างเราสามารถบอกได้ว่าชนิดแพคเกจของสินค้า จะเป็นตัวที่ใช้ในการแบ่งกลุ่มข้อมูลในมิติรายการสินค้า จากทั้งสองตัวอย่างข้างต้น ข้อมูลที่บ่งบอกว่าเป็นวันหยุดหรือไม่และข้อมูลชนิดของแพคเกจสินค้าไม่จำเป็นต้องถูกเก็บ หรือวางไว้ในลำดับชั้นของมิติแกนเวลา และมิติรายการสินค้า แต่ข้อมูลส่วนนั้นยังคงถูกเก็บไว้ในแต่ละมิติ ที่เราสามารถเรียกข้อมูลเหล่านั้นว่าข้อมูลหมวดหมู่

ในการออกแบบหรือกำหนดแพคเกจข้อมูลของแต่ละมิติทางธุรกิจเราควรต้องรวมลำดับชั้นและหมวดหมู่ของข้อมูลเข้าไปด้วย เพื่อทำให้ผู้ใช้ได้ข้อมูลหลากหลายมุมมอง เพื่อให้เข้าใจได้ง่ายขึ้น ลองพิจารณา 2 ตัวอย่างจากส่วนที่แล้วนั่นคือ บริษัทผู้ผลิตรถยนต์ และธุรกิจโรงแรมที่มีหลายสาขา แล้วทำการวิเคราะห์ลำดับชั้นและหมวดหมู่ในแต่ละมิติทางธุรกิจ

ธุรกิจผู้ผลิตรถยนต์ จะประกอบไปด้วย มิติรายการสินค้า มิติผู้ค้า มิติการจ่ายเงิน และมิติแกนเวลา เราสามารถกำหนดรายละเอียดของแต่ละมิติได้ดังนี้

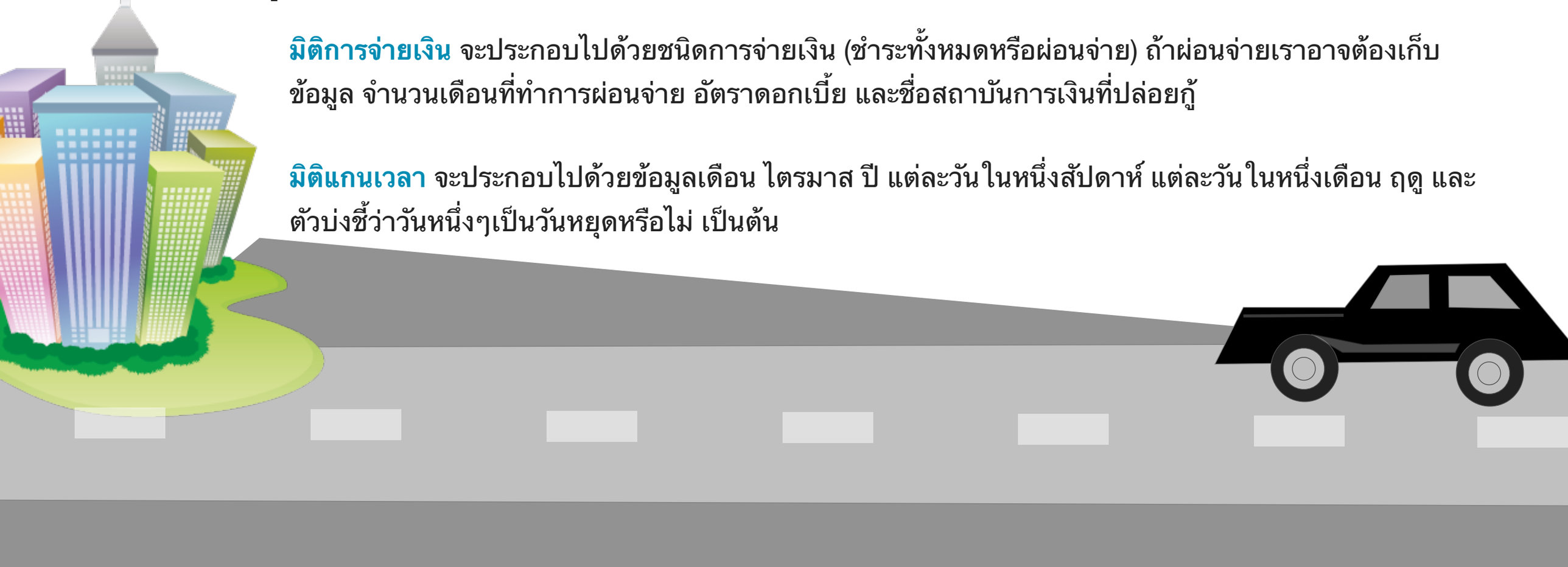
มิติรายการสินค้าจะประกอบด้วยชื่อรุ่น ปีของรุ่น ลักษณะรูปลักษณ์ รายการผลิต กลุ่มของรายการสินค้า สีภายนอก และสีภายใน เป็นต้น

มิติผู้ค้า จะประกอบไปด้วยชื่อผู้ค้า อำเภอ จังหวัด ตัวบ่งชี้ผู้ค้าว่า มีสินค้าแค่ยี่ห้อเดียวหรือไม่ และวันแรกของการประกอบการ เป็นต้น

มิติลูกค้า ประกอบไปด้วยอายุ เพศ ช่วงของเงินเดือน สถานะภาพ จำนวนสมาชิกในครอบครัว มีบ้านเป็นของตัวเองหรือไม่ มูลค่าของบ้าน มีรถเป็นของตัวเองหรือไม่ เป็นต้น

มิติการจ่ายเงิน จะประกอบไปด้วยชนิดการจ่ายเงิน (ชำระทั้งหมดหรือผ่อนจ่าย) ถ้าผ่อนจ่ายเราอาจต้องเก็บข้อมูล จำนวนเดือนที่ทำการผ่อนจ่าย อัตราดอกเบี้ย และชื่อสถาบันการเงินที่ปล่อยกู้

มิติแกนเวลา จะประกอบไปด้วยข้อมูลเดือน ไตรมาส ปี แต่ละวันในหนึ่งสัปดาห์ แต่ละวันในหนึ่งเดือน ฤดู และตัวบ่งชี้ว่าวันหนึ่งๆเป็นวันหยุดหรือไม่ เป็นต้น



ธุรกิจโรงแรมที่มีหลายสาขาที่ประกอบไปด้วยมิติสาขาของโรงแรม มิติห้องพัก และมิติแกนเวลา ซึ่งสามารถกำหนดลำดับขั้นของแต่ละหมวดหมู่ของแต่ละมิติได้ดังนี้

มิติสาขาของโรงแรมจะประกอบไปด้วยชื่อสาขา รหัสสาขา เขต ที่อยู่ อำเภอ เมือง รหัสไปรษณีย์ ผู้จัดการ ปีที่สร้าง ปีที่ทำการปรับปรุง เป็นต้น

มิติห้องพักจะประกอบไปด้วยข้อมูลชนิดของห้องพัก ขนาดของห้องพัก จำนวนเตียงในห้องพักชนิดของเตียง จำนวนผู้เข้าพักสูงสุด ตู้เย็น และครัว เป็นต้น

มิติแกนเวลาจะประกอบไปด้วยข้อมูลแต่ละวันใน 1 เดือน แต่ละวันใน 1 สัปดาห์ เดือน ไตรมาส ปี และตัวบ่งชี้ว่าวันหนึ่งๆเป็นวันหยุดหรือไม่ เป็นต้น

จากทั้งสองตัวอย่างธุรกิจข้างต้น เราจะเห็นว่ามิติแกนเวลาของทั้งสองธุรกิจจะเหมือนกัน ซึ่งโดยส่วนใหญ่แล้วแต่ละคลังข้อมูลจะมีลำดับชั้นและหมวดหมู่ของข้อมูลในมิติแกนเวลาไม่แตกต่างกันมากนัก ในส่วนของรายละเอียดในมิติอื่น ๆ จะขึ้นกับความต้องการของผู้ใช้หรือสิ่งที่สนใจในแต่ละองค์กร ซึ่งเมื่อเราทำการพิจารณามิติต่าง ๆ ทางธุรกิจแล้ว เราจะสามารถใช้มิติทางธุรกิจในการวิเคราะห์ได้ก็ต่อเมื่อมีมาตรวัดความสำเร็จหรือตัวบ่งชี้ประสิทธิภาพของการดำเนินธุรกิจ ซึ่งมาตรวัดเหล่านี้จะเป็นความจริง (Fact) ที่สะท้อนถึงการดำเนินการของส่วนงานหรือแผนกที่สามารถดำเนินการตามที่ตั้งเป้าไว้หรือไม่ จากตัวอย่างทั้งสองข้างต้น ลองพิจารณาตลาดรถยนต์ที่มาตรวัดที่ใช้ส่วนใหญ่จะเกี่ยวกับการขาย เช่น ราคาขายจริง ราคาที่ผู้ผลิตเสนอให้ผู้ขายปลีก ราคาเต็ม เครดิตของตัวแทนจำหน่าย ใบแจ้งหนี้ตัวแทนจำหน่าย จำนวนเงินดาวน์รถ และจำนวนเงินคงคลัง เป็นต้น สำหรับธุรกิจโรงแรมจะมีมาตรวัดที่แตกต่างออกไป ซึ่งโดยปกติแล้วมาตรวัดจะเกี่ยวเนื่องกับสิ่งที่เราต้องการวิเคราะห์ ซึ่งสำหรับธุรกิจโรงแรมมาตรวัดจะเกี่ยวกับ การเข้าพักของลูกค้าซึ่งประกอบไปด้วย ห้องที่มีผู้เข้าพัก ห้องว่าง ห้องพักที่ไม่สามารถใช้งานได้ จำนวนผู้เข้าพัก และรายได้ เป็นต้น

หลังจากการวิเคราะห์มิติทางธุรกิจที่ประกอบไปด้วยลำดับชั้นและหมวดหมู่ของข้อมูลในแต่ละมิติ และมาตรวัดประสิทธิภาพต่างๆ เราสามารถนำข้อมูลทั้งหมดมารวมกันเพื่อสร้างแพคเกจข้อมูล (information package) ของแต่ละธุรกิจ ดังแสดงในรูปที่ 6-5 และ 6-6

Information Subject: Automaker Sales

Dimensions

Time	Product	Payment Method	Customer Demo-graphics	Dealer	
Year	Model Name	Finance Type	Age	Dealer Name	
Quarter	Model Year	Term (Months)	Gender	City	
Month	Package Styling	Interest Rate	Income Range	State	
Date	Product Line	Agent	Marital Status	Single Brand Flag	
Day of Week	Product Category		House-hold Size	Date First Operation	
Day of Month	Exterior Color		Vehicles Owned		
Season	Interior Color		Home Value		
Holiday Flag	First Year		Own or Rent		

Hierarchies/Categories

Facts: Actual Sale Price, MSRP, Options Price, Full Price, Dealer Add-ons, Dealer Credits, Dealer Invoice, Down Payment, Proceeds, Finance

รูปที่ 6-5 ตัวอย่างแพคเกจข้อมูลสำหรับบริษัทผู้ผลิตรถยนต์

Information Subject: Hotel Occupancy

Dimensions

Time	Hotel	Room Type			
Year	Hotel Line	Room Type			
Quarter	Branch Name	Room Size			
Month	Branch Code	Number of Beds			
Date	Region	Type of Bed			
Day of Week	Address	Max. Occupants			
Day of Month	City/State /Zip	Suite			
Holiday Flag	Construction Year	Refrigerator			
	Renovation Year	Kichennette			

Facts: Occupied Rooms, Vacant Rooms, Unavailable Rooms, Number of Occupants, Revenue

Hierarchies/Categories

รูปที่ 6-6 ตัวอย่างแพคเกจข้อมูลสำหรับธุรกิจโรงแรม

SECTION 5

วิธีในการเก็บรวบรวมความต้องการ

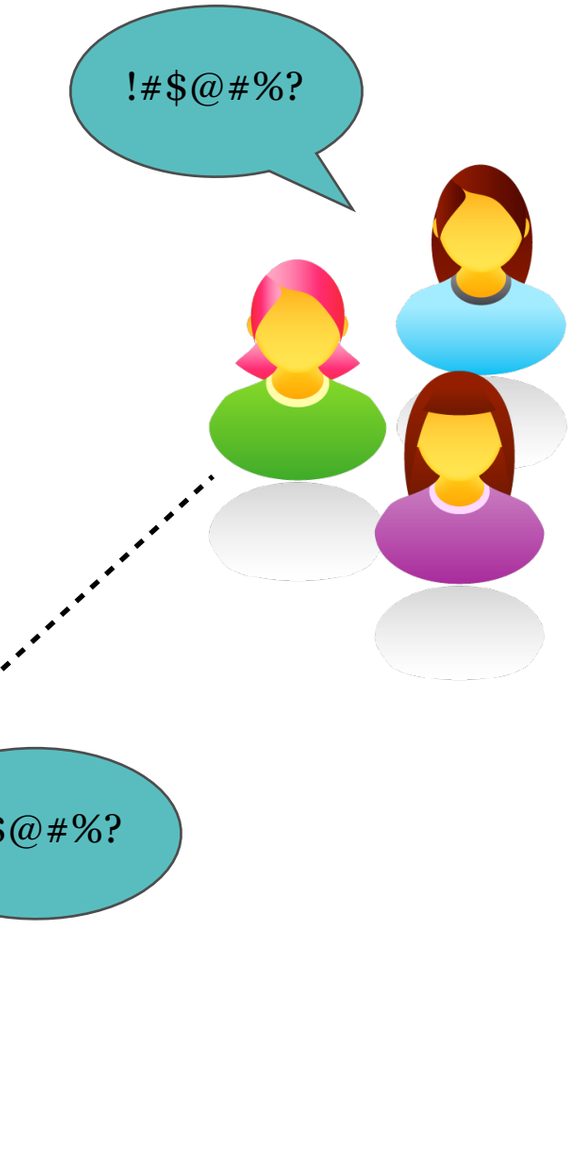
ในขณะที่เราทราบถึงวิธีการที่จะทำให้ผู้ใช้คิดและแสดงความต้องการให้เป็นรูปเป็นร่างโดยการสร้างไดอะแกรมแพกเกจข้อมูล แต่ก่อนที่จะได้มาซึ่งไดอะแกรมแพกเกจข้อมูล เราจะต้องทำการเก็บรวบรวมข้อมูลและความต้องการจากผู้ใช้เบื้องต้นเสียก่อน ซึ่งการเก็บรวบรวมความต้องการสามารถทำได้หลายวิธีตามแต่สถานการณ์ที่เราประสบพบเจออยู่ อย่างที่เราทราบกันดีว่าผู้ใช้คลังข้อมูลจะประกอบด้วยผู้ใช้หลายประเภท เช่น ผู้บริหารระดับสูง ผู้จัดการแผนกสำคัญ ๆ นักวิเคราะห์เชิงธุรกิจ ผู้ดูแลระบบข้อมูล และอื่น ๆ

เมื่อคลังข้อมูลมีผู้ใช้ที่ค่อนข้างหลากหลาย แต่ละกลุ่มของผู้ใช้อาจมีความต้องการที่แตกต่างกันตามหน้าที่การทำงานที่รับผิดชอบ ซึ่งอาจจะทำให้เราได้รับข้อมูลที่หลากหลายในการเก็บรวบรวมข้อมูล อาทิเช่น ผู้บริหารจะเป็นผู้กำหนดแนวทางและขอบเขตของคลังข้อมูลที่จะสร้างขึ้น ซึ่งข้อมูลดังกล่าวสามารถบ่งบอกถึงสิ่งที่ผู้บริหารเหล่านั้นสนใจหรือข้อมูลที่พวกเขาเหล่านั้นต้องการ ในส่วนของผู้จัดการแผนกและนักวิเคราะห์ข้อมูลที่เป็นผู้รายงานและเตรียมรายงานเกี่ยวกับสิ่งที่ผู้บริหารสนใจ ผู้ใช้ทั้งสองกลุ่มนี้อาจต้องการการเรียกดูข้อมูลต่าง ๆ เพื่อจัดทำรายงาน และท้ายสุดคือผู้ดูแลระบบฐานข้อมูลอาจให้ข้อมูลหรือยุ่งเกี่ยวกับฐานข้อมูลและแหล่งข้อมูล เป็นต้น

จากหน้าที่และความรับผิดชอบที่แตกต่างกันของผู้ใช้คลังข้อมูล เราจะต้องทำการเก็บรวบรวมข้อมูล ความต้องการและแนวทางในการปฏิบัติงานของแต่ละกลุ่มผู้ใช้โดยการใช้เทคนิคต่างๆ โดยแนวทางในการสร้างระบบการดำเนินงานและการสร้างคลังข้อมูลจะประกอบไปด้วย

- การสัมภาษณ์แต่ละบุคคลแบบหนึ่งต่อหนึ่งที่คาดว่าจะใช้คลังข้อมูลหรือการสัมภาษณ์เป็นกลุ่มเล็กๆ
- การประชุมโดยรวมกลุ่มคนที่เกี่ยวข้องกับการสร้างคลังข้อมูล (Joint Application Development, JAD)
- การใช้แบบสอบถามข้อมูล
- อื่นๆ

จากทั้งสามเทคนิคการเก็บรวบรวมข้อมูลข้างต้น วิธีการสัมภาษณ์จะเป็นวิธีการที่มีการติดต่อสื่อสารกับผู้ใช้มากที่สุด ซึ่งจะช่วยให้เราได้ข้อมูลค่อนข้างละเอียดและได้รับการยืนยันเกี่ยวกับข้อมูลอีกด้วย JAD หรือการประชุมแบบรวมกลุ่มจะมีการติดต่อสื่อสารกับผู้ใช้แต่ละงานไม่มากนักและการใช้แบบสอบถามจะไม่มีการพูดคุยกับผู้ใช้เลย ลองพิจารณาข้อดีและแนวปฏิบัติของแต่ละวิธีดังต่อไปนี้ ซึ่งอาจจะช่วยให้เราเลือกเก็บรวบรวมข้อมูลและความต้องการได้ถูกต้อง ครบถ้วนและสมบูรณ์



การสัมภาษณ์

การสัมภาษณ์

- ทำการสัมภาษณ์ผู้ใช้ 1-3 คนที่มีหน้าที่เหมือนกันหรือใกล้เคียงกันในการสัมภาษณ์แต่ละครั้ง เพื่อช่วยให้เราสามารถได้รับข้อมูลความต้องการหลายๆมุมมองของผู้ใช้
- ง่ายในการนัดหมาย
- เป็นวิธีการที่ดีในการเก็บรวบรวมความต้องการที่มีความซับซ้อน
- ผู้ใช้บางคนจะรู้สึกผ่อนคลายและสะดวกสบายเมื่อเราทำการสัมภาษณ์แบบหนึ่งต่อหนึ่ง
- ต้องการการเตรียมการที่ดีที่จะทำให้การเก็บรวบรวมข้อมูลได้อย่างมีประสิทธิภาพ
- ต้องทำการศึกษาข้อมูลเบื้องต้นหรือทำวิจัยเบื้องต้นก่อนการสัมภาษณ์
- ต้องมีการกำหนดวัตถุประสงค์ในการสัมภาษณ์แต่ละครั้ง
- ควรกำหนดชนิดของคำถาม
- ช่วยกระตุ้นให้ผู้ใช้มีการเตรียมตัว/เตรียมข้อมูลสำหรับการสัมภาษณ์

การประชุมแบบรวมกลุ่ม

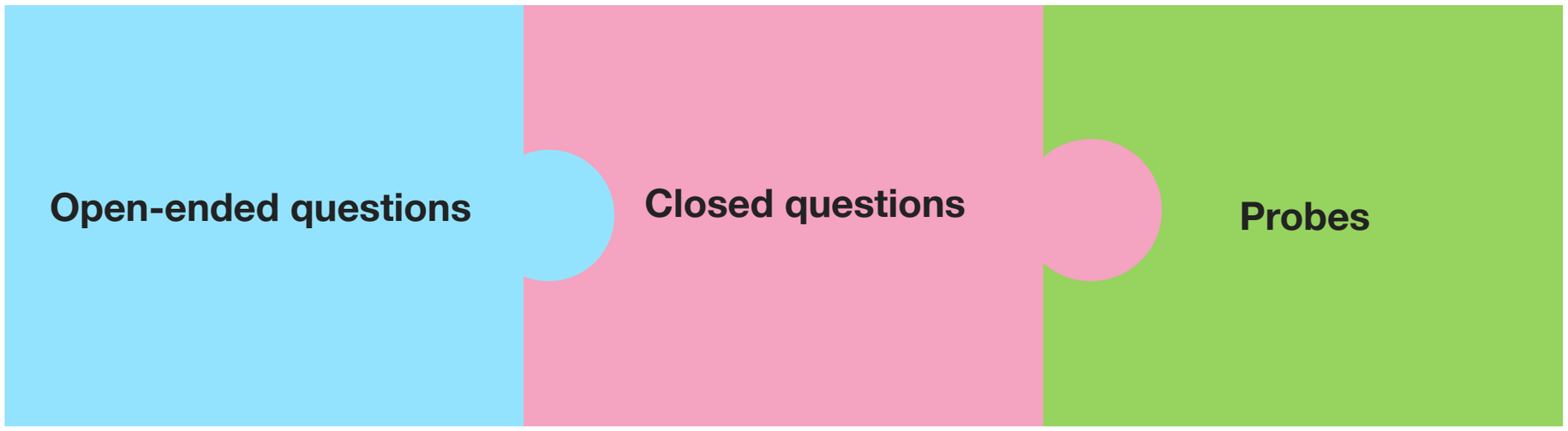
- จะมีคนเข้าร่วมประมาณ 20 คนในการประชุมแต่ละครั้ง
- ควรจะใช้วิธีการนี้หลังจากเราได้ข้อมูลความต้องการเบื้องต้นจากผู้ใช้ต่างๆ
- ไม่เหมาะกับการเริ่มต้นการเก็บรวบรวมข้อมูลมีประโยชน์ในการยืนยันและทำข้อตกลงเกี่ยวกับความต้องการต่างๆ
- จะเป็นวิธีที่มีประสิทธิภาพเมื่อผู้เข้าร่วมประชุมมีตำแหน่งหน้าที่การงานที่แตกต่างกัน
- ต้องมีการจัดการที่ดีจึงจะประสบความสำเร็จในการจัดเก็บข้อมูล

แบบสอบถาม

- สามารถทราบถึงความต้องการเป็นจำนวนมากได้อย่างรวดเร็ว
- อาจมีประโยชน์เมื่อผู้กรอกแบบสอบถามมีความแตกต่างกัน
- อาจมีประโยชน์กับผู้ใช้บางคนที่ค่อนข้างยุ่ง ซึ่งจะไม่มีเวลามาเข้าประชุมหรือเข้าร่วมสัมภาษณ์
- ไม่มีการติดต่อสื่อสารกับผู้ใช้

ชนิดของคำถาม

ในการเก็บรวบรวมข้อมูลภายใต้เทคนิคข้างต้นจะใช้การซักถามเพื่อให้ทราบถึงความต้องการของผู้ใช้แต่ละหมวดหมู่ ถ้าเรามีความเข้าใจเกี่ยวกับชนิดของคำถามและมีการเตรียมคำถามที่ดี อาจช่วยให้เราสามารถเก็บรวบรวมความต้องการจากผู้ใช้งานได้อย่างถูกต้องและครบถ้วน ซึ่งโดยส่วนใหญ่ของการเก็บรวบรวมข้อมูลจะใช้คำถามอยู่ 3 ประเภทดังนี้



Open-ended questions

Open-ended questions: การใช้คำถามในลักษณะนี้ช่วยเปิดทางเลือกในการตอบคำถาม ซึ่งจะช่วยให้ผู้ถูกสัมภาษณ์รู้สึกสบายใจสามารถตอบคำถามได้โดยง่าย ผู้ถูกสัมภาษณ์สามารถบอกกล่าวได้ถึงความเชื่อและความคิดเห็น ช่วยเปิดโอกาสที่จะถามคำถามต่อๆไป หรือการถามคำถามเพิ่มเติม จากคำถามที่เตรียมมาก่อนหน้าได้

แต่อย่างไรก็ดี ก็มีข้อเสียหลายข้อด้วยกัน เช่น ทำให้เรารับรู้ข้อมูลที่ไม่จำเป็นมากเกินไป มีความเสี่ยงที่จะไม่สามารถควบคุมการสัมภาษณ์ได้ อาจใช้เวลานาน เป็นต้น

Closed questions

Closed questions: การใช้คำถามในลักษณะนี้จะทำให้ช่องทางการตอบคำถามนั้นแคบลง ซึ่งในบางคำถามอาจมีคำตอบเพียงแค่ “ใช่” หรือ “ไม่” เท่านั้น การใช้คำถามปลายปิดจะช่วยประหยัดเวลา ได้ข้อมูลอย่างรวดเร็วและง่าย ช่วยให้สามารถควบคุมการสัมภาษณ์ได้ ช่วยให้สามารถ ได้ข้อมูลเบื้องต้นได้อย่างรวดเร็วและช่วยให้ได้ข้อมูลที่ตรงประเด็น เป็นต้น แต่ข้อเสียของการใช้คำถามปลายปิดคือ ไม่สามารถที่จะเก็บรายละเอียดข้อมูลได้ทั้งหมด ลดโอกาสในการสร้างความไว้วางใจและความสามัคคีระหว่างผู้ให้สัมภาษณ์และผู้ถูกสัมภาษณ์ ท้ายสุดอาจทำให้การสัมภาษณ์นั้นน่าเบื่อ เป็นต้น

Probes

Probes: คำถามในลักษณะนี้เป็นคำถามแบบต่อเนื่อง โดยเราอาจจะใช้คำถามเป็นแบบปลายเปิดหรือปลายปิดไปก่อนหน้า วัตถุประสงค์หลักของคำถามแบบต่อเนื่องคือ ต้องการที่จะถามคำถามขยายหรือต่อเติมจากคำถามคำตอบของปลายเปิดหรือปลายปิดที่ถามไปก่อนหน้า ที่จะช่วยให้ได้รับรายละเอียดและรวมถึงมุมมองของผู้ถูกสัมภาษณ์ด้วย

วิธีในการถามคำถาม

ในการเก็บรวบรวมข้อมูลโดยใช้คำถามชนิดต่างๆ ที่เหมาะสมอาจไม่เพียงพอ เราต้องมีการจัดลำดับคำถามที่ดีที่เหมาะกับผู้ที่มีส่วนร่วมในการเก็บข้อมูล และมีความสอดคล้องกับวัตถุประสงค์ที่ต้องการด้วย ลองพิจารณาโครงสร้างการจัดลำดับคำถามดังต่อไปนี้เพื่อนำไปประยุกต์ใช้ในการเก็บรวบรวมข้อมูลต่อไป

Pyramid structure : วิธีการนี้เริ่มด้วยคำถามปลายปิดที่ค่อนข้างเฉพาะเจาะจงจากนั้นค่อยๆ ทำการถามคำถามปลายเปิดที่เกี่ยวข้องกับเรื่องเดิมที่ถามไว้จากคำถามแรก

Funnel structure: วิธีการนี้จะเริ่มจากคำถามทั่วๆ ไปที่เป็นคำถามปลายเปิด จากนั้นค่อยๆ ถามคำถามปลายปิดที่ทำให้หัวข้อที่กำลังสนทนาแคบลงเรื่อยๆเพื่อสรุปคำถาม เราควรใช้วิธีการนี้เพื่อให้ได้รายละเอียดที่เราต้องการอย่างค่อยเป็นค่อยไป

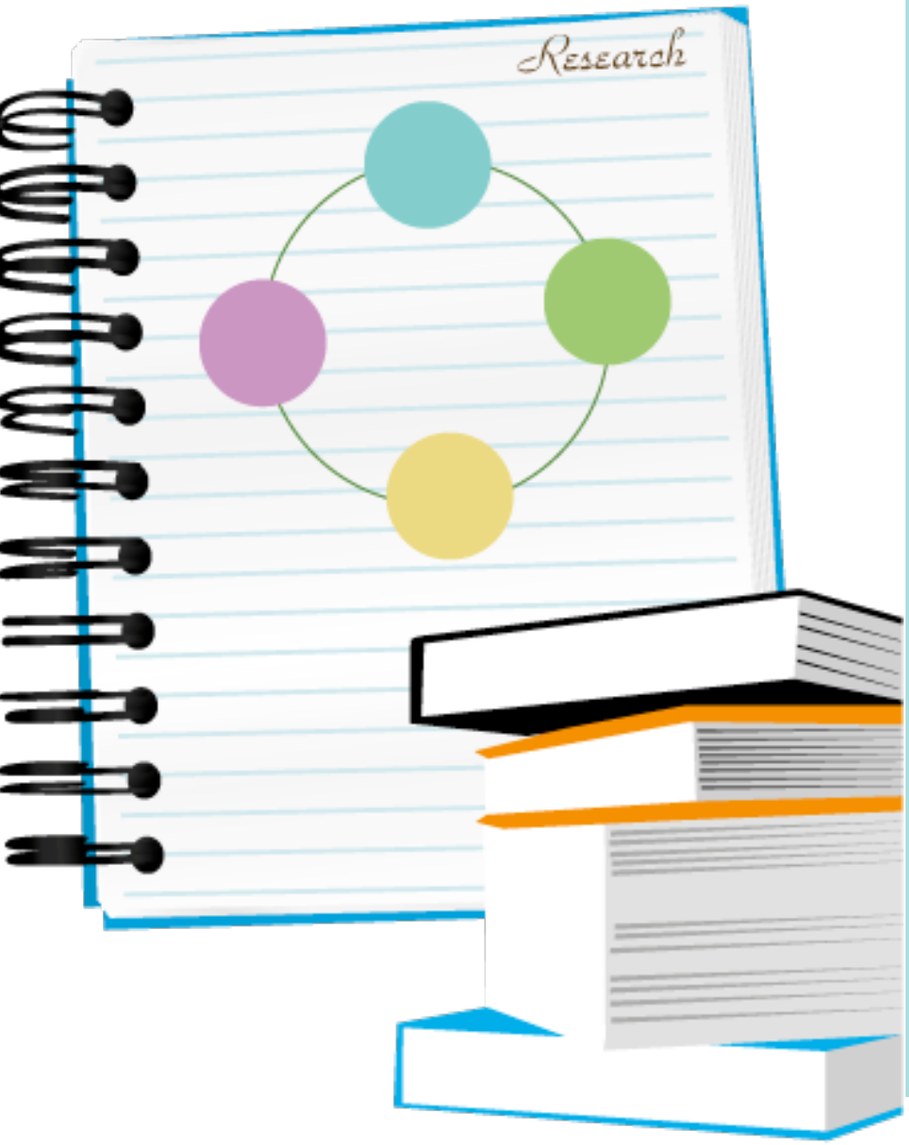
Diamond-shaped structure: วิธีนี้จะทำการอุ่นเครื่องการสัมภาษณ์ด้วยคำถามปลายปิดที่ค่อนข้างเฉพาะเจาะจง จากนั้นทำการถามแบบกว้างๆ ด้วยคำถามปลายเปิด และจากนั้นทำให้การสัมภาษณ์เข้าสู่บทสรุปด้วยคำถามปลายปิดที่เฉพาะเจาจงอีกครั้งหนึ่ง ซึ่งจากลำดับของคำถามดังกล่าวเราจะเห็นว่าลำดับของคำถามในลักษณะนี้จะดีกว่า 2 ลำดับข้างต้น แต่อย่างไรก็ดีลำดับในลักษณะนี้อาจทำให้ใช้เวลาใน

เทคนิคในการสัมภาษณ์

ในการเก็บรวบรวมข้อมูลและความต้องการ วิธีการสัมภาษณ์ผู้ที่มีส่วนเกี่ยวข้องหลายระดับและหลายส่วนงานอาจทำให้การเก็บรวบรวมข้อมูลนั้นใช้เวลาค่อนข้างมาก เราควรจะมีการเตรียมการและจัดการที่ดีด้วยการเตรียมความพร้อมต่างๆ ดังต่อไปนี้

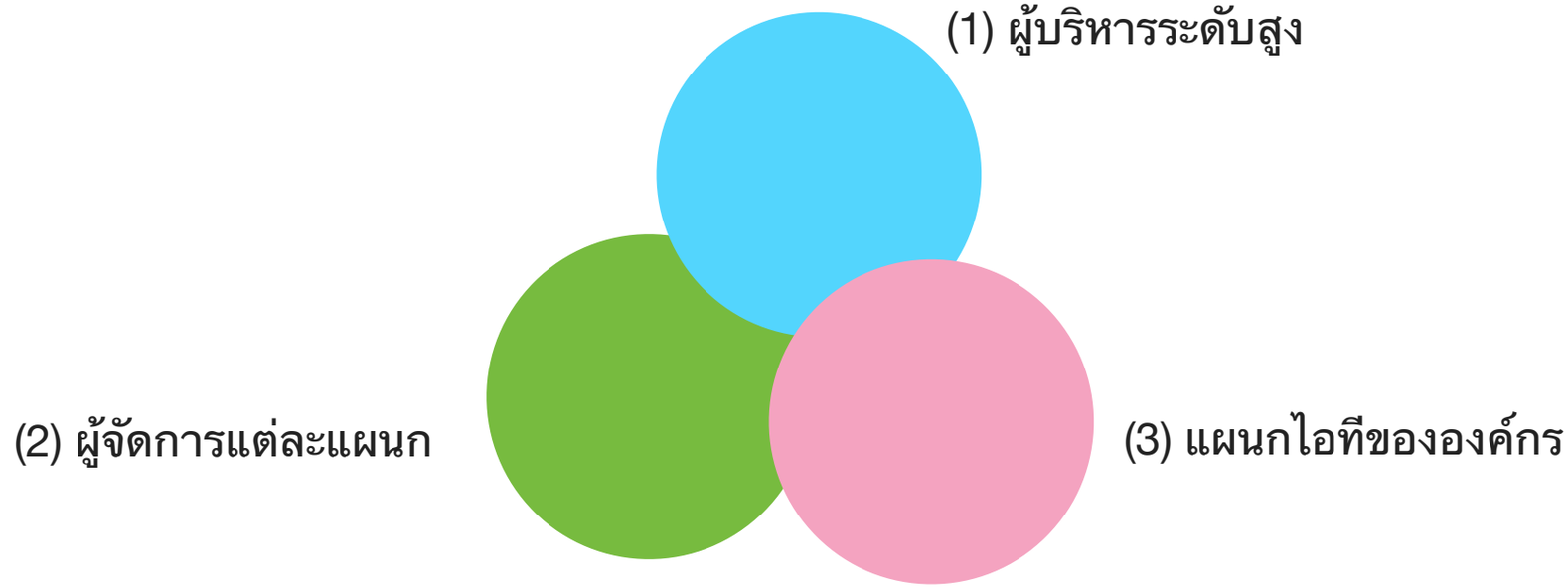
- ทำการเลือกและให้ความรู้ที่เกี่ยวกับการสัมภาษณ์ เพื่อทำการรวบรวมข้อมูลและความต้องการให้กับสมาชิกในทีมงานในการสร้างคลังข้อมูล
- ทำการกำหนดบทบาทของแต่ละสมาชิกผู้สร้างให้ชัดเจน เช่น ผู้นำในการสัมภาษณ์ ผู้จดบันทึกข้อมูล และผู้สรุปประเด็นต่างๆ
- เตรียมลิสต์รายชื่อของผู้ที่ต้องถูกสัมภาษณ์และทำการกำหนดตารางเวลาอย่างคร่าวๆ
- เตรียมลิสต์ของสิ่งที่ผู้สร้างคาดหวังว่าจะได้รับจากการสัมภาษณ์แต่ละครั้ง
- ทำการวิจัยเบื้องต้นก่อนการสัมภาษณ์
- เตรียมคำถามที่เกี่ยวข้องกับการสัมภาษณ์
- จัดทำการประชุมผู้ที่ต้องถูกสัมภาษณ์ทั้งหมดเพื่อชี้แจงรายละเอียดต่างๆ

จากหัวข้อที่กำหนดข้างต้น ทุกหัวข้อยกเว้นการทำวิจัยเบื้องต้นก่อนการสัมภาษณ์นั้นเป็นกระบวนการที่ค่อนข้างเข้าใจได้ง่ายและสามารถดำเนินการได้ตามเทคนิคของทีมผู้สร้าง ดังนั้นเพื่อให้เข้าใจเกี่ยวกับความสำคัญของการทำวิจัยเบื้องต้นก่อนการสัมภาษณ์ที่มีการเก็บรวบรวมข้อมูลและความต้องการ โดยพิจารณาถึงหัวข้อต่าง ๆ ที่สำคัญในการทำวิจัยเบื้องต้นดังนี้



- โครงสร้างทางธุรกิจขององค์กรทั้งในอดีตและปัจจุบัน
- จำนวนพนักงาน รวมถึงหน้าที่และความรับผิดชอบของแต่ละตำแหน่ง
- ที่ตั้ง/ที่อยู่ของผู้ใช้
- วัตถุประสงค์หลักของแต่ละส่วนงานของภาคธุรกิจในองค์กร
- ความสัมพันธ์ของส่วนงานภาคธุรกิจกับกลยุทธ์ขององค์กร
- ผลงานของแต่ละส่วนงานภาคธุรกิจที่เกี่ยวข้องกับรายได้ และค่าใช้จ่าย
- การตลาดของบริษัท
- การแข่งขันทางการตลาด
- อื่นๆ

หลังจากการทำวิจัยเบื้องต้น เราจะทราบถึงข้อมูลความเป็นมาและการดำเนินธุรกิจขององค์กรเบื้องต้น รวมถึงบุคคลหรือกลุ่มคนที่เราจะทำการสัมภาษณ์ที่ซึ่งสามารถแบ่งได้เป็น 3 กลุ่มกว้างๆ คือ



ซึ่งจากกลุ่มของผู้ใช้หรือผู้ที่เราจะทำการสัมภาษณ์เราควรจะต้องทำการสร้างลิสต์รายการข้อมูลที่เราต้องการที่จะได้รับจากแต่ละกลุ่มของผู้ใช้เสียก่อน ดังแสดงในรูปที่ 6-7

Senior Executives

- Organization objectives
- Criteria for measuring success
- Key business issues, current & future
- Problem identification
- Vision and direction for the organization
- Anticipated usage of the DW

Dept. Managers / Analysts

- Departmental objectives
- Success metrics
- Factors limiting success
- Key business issues
- Products & Services
- Useful business dimensions for analysis
- Anticipated usage of the DW

IT Dept. Professionals

- Key operational source systems
- Current information delivery processes
- Types of routine analysis
- Known quality issues
- Current IT support for information requests
- Concerns about proposed DW

รูปที่ 6-7 ความคาดหวังจากการสัมภาษณ์ผู้ใช้

เกร็ดเล็กเกร็ดน้อย

ในการตั้งคำถามเพื่อใช้ในการสัมภาษณ์จะมีเคล็ดลับหรือเกร็ดเล็กน้อยต่างๆ มากมายดังนี้

Current information sources: คำถามที่ใช้ถามแผนกไอที เกี่ยวกับแหล่งข้อมูลที่ใช้ในการสร้างข้อมูล เช่น

- ระบบการดำเนินงานใดที่มีข้อมูลที่สำคัญเกี่ยวกับหัวข้อธุรกิจต่างๆ?
- ประเภทของระบบคอมพิวเตอร์ที่จัดเก็บข้อมูลที่เกี่ยวกับหัวข้อทางธุรกิจมีอะไรบ้าง
- คิวรีและรายงานที่ได้จากแหล่งข้อมูล (ระบบการดำเนินงาน) ประกอบไปด้วยข้อมูลสารสนเทศอะไรบ้าง?
- ความละเอียดของข้อมูลในระบบเป็นอย่างไร?

Subset areas: เป็นคำถามเกี่ยวกับหัวข้อทางธุรกิจต่างๆ เช่น

- หัวข้อทางธุรกิจใดที่มีคุณค่าต่อการวิเคราะห์มากที่สุด?
- มิติทางธุรกิจ (Business dimension) คืออะไร?
- ส่วนงานใดของการดำเนินธุรกิจที่ต้องมีการตัดสินใจบ้าง?

Key performance matrices: จะเป็นคำถามเกี่ยวกับข้อมูลเชิงเวลาสำหรับกิจกรรมต่างๆ เช่น

- ต้องมีการอัพเดตข้อมูลเพื่อสนับสนุนการตัดสินใจบ่อยเพียงใด?
- กรอบเวลาเป็นอย่างไร?

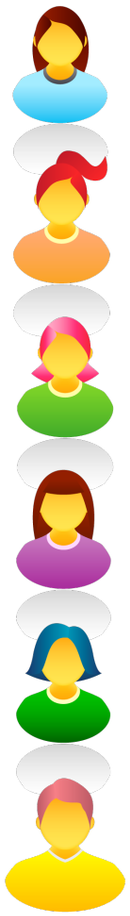
จากตัวอย่างข้างต้น เราจะทราบถึงมุมมองของแต่ละวัตถุประสงค์ รวมถึงข้อมูลที่ได้รับจากแต่ละคำถาม ซึ่งจะช่วยให้เราสามารถเก็บรวบรวมความต้องการในแง่มุมต่างๆ ดังนั้น ในการสัมภาษณ์เพื่อเก็บรวบรวมข้อมูล เราควรจะบันทึกข้อมูลไว้ต่างๆ ดังนี้

1. รายละเอียดของผู้ถูกสัมภาษณ์
2. ประวัติความเป็นมาและวัตถุประสงค์
3. รายละเอียดความต้องการของข้อมูลสารสนเทศ
4. รายละเอียดและความต้องการของการวิเคราะห์
5. เครื่องมือที่ใช้ในปัจจุบัน
6. เกณฑ์ความสำเร็จ
7. ตัวชี้วัดทางธุรกิจที่เป็นประโยชน์
8. มิติของธุรกิจที่เกี่ยวข้อง

หลังจากทำการบันทึกข้อมูล/ความต้องการที่มีรายละเอียดข้างต้น เราสามารถนำความต้องการของแต่ละบุคคลไปพิจารณาเพื่อกำหนดความต้องการที่แท้จริงของการสร้างคลังข้อมูลต่อไป

การประชุมแบบรวมกลุ่ม

JAD (Join Application Development) จะเป็นการนำกลุ่มคนที่มีหน้าที่ที่แตกต่างกันมารวมกันเพื่อพูดคุยตัดสินใจ เก็บรวบรวมข้อมูล/ความต้องการจากการสร้างระบบการดำเนินงาน ซึ่งข้อดีของ JAD คือประหยัดเวลาในการเก็บรวบรวมข้อมูลความต้องการและจะได้ข้อมูลที่ครบถ้วนสมบูรณ์ แต่อย่างไรก็ดี JAD ที่นำกลุ่มคนมาประชุมกันอาจใช้เวลามาก อาจเกิดข้อขัดแย้งขึ้นระหว่างการประชุม ดังนั้นเราจึงต้องมีการควบคุมการประชุมให้ดี โดยที่กลุ่มของผู้ใช้ที่จะมีส่วนร่วมในการประชุมจะประกอบไปด้วย



Executive sponsor คือ บุคคลที่ควบคุมเรื่องการเงิน และเป็นผู้ที่มีความสามารถกำหนดทิศทางและมีอำนาจในการตัดสินใจ

Facilitation เป็นบุคคลที่ให้คำแนะนำต่าง ๆ ระหว่างการประชุม

Scribe คือ บุคคลที่ทำการจดบันทึกทุก ๆ การตัดสินใจ

Full-time participants คือ ทุก ๆ คนที่เกี่ยวกับการตัดสินใจเกี่ยวกับการสร้างคลังข้อมูล

On-call participants คือ บุคคลที่มีส่วนเกี่ยวข้องกับโปรเจค แต่มีส่วนร่วมแค่ในบางเรื่องเท่านั้น

Observers เป็นบุคคลที่นั่งอยู่ในที่ประชุมแต่ไม่ออกความเห็นในการตัดสินใจ

เมื่อเราสามารถรวบรวมข้อมูลผู้ที่ส่วนเกี่ยวข้องกับการประชุมได้แล้ว เราลองพิจารณาถึงขั้นตอนการทำงานของ JAD ที่ประกอบไปด้วย 5 ขั้นตอนดังนี้

1

Project definition จะทำการสัมภาษณ์เพื่อเก็บข้อมูลแบบผิวเผิน จัดการเกี่ยวกับการประชุมและคำแนะนำสำหรับการจัดการต่างๆ

2

Research คือ การทำความคุ้นเคยกับการดำเนินการธุรกิจ เอกสารต่างๆ ในการดำเนินธุรกิจ เอกสารต่างๆ ที่เกี่ยวกับความต้องการในการได้รับข้อมูล การจัดเก็บข้อมูลเบื้องต้นและจัดเตรียมวาระการประชุม

3

Preparation คือ การสร้างเอกสารการดำเนินงานจากขั้นตอนก่อนหน้า ฝึกอบรมผู้จดบันทึกข้อมูล เตรียมเครื่องมืออำนวยความสะดวก จัดเตรียมสถานที่สำหรับการประชุมและเตรียมรายการต่าง ๆ สำหรับแต่ละหัวข้อ

4

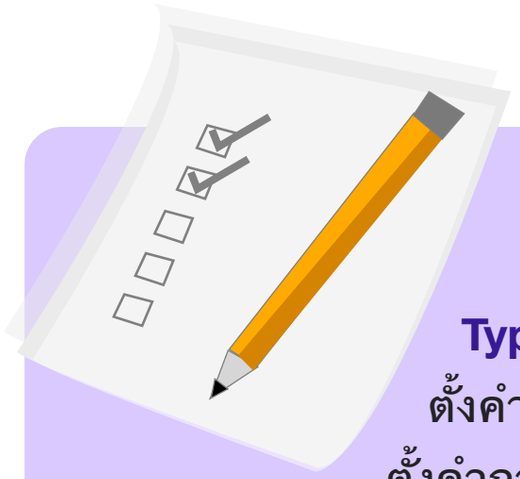
JAD session เริ่มจากการทบทวนวาระและวัตถุประสงค์ของการประชุม ทบทวนเกี่ยวกับสมมติฐาน ทบทวนเกี่ยวกับความต้องการข้อมูลต่างๆ ทบทวนมาตราวัดประสิทธิภาพในการดำเนินธุรกิจและมิติต่างๆ ทางธุรกิจ

5

Final Document ทำการเขียนเอกสารทำการเชื่อมโยงข้อมูลที่เก็บรวบรวมไว้เขาด้วยกัน แจกแจงแหล่งข้อมูล ระบุถึงมาตราวัดความสำเร็จของการดำเนินธุรกิจทั้งหมด แจกแจงมิติทางธุรกิจต่างๆรวมถึงลำดับชั้นของรายละเอียดข้อมูลและการอนุมัติ/ทำข้อตกลง ในขั้นตอนสุดท้าย

การใช้แบบสอบถาม

การเก็บข้อมูลโดยใช้แบบสอบถามจะไม่มีการพูดคุยหรือไม่มีการโต้ตอบกันระหว่างผู้ให้และผู้รับข้อมูล ดังนั้น ถ้าเราจะประยุกต์ใช้แบบสอบถามในการเก็บรวบรวมข้อมูลเราจะต้องให้ความใส่ใจและให้ความระมัดระวังในการออกแบบแบบสอบถามเพื่อให้ได้ข้อมูลที่ถูกต้องและครบถ้วน ซึ่งการออกแบบแบบสอบถามควรที่จะต้องพิจารณาสิ่งต่อไปนี้



Type and Choice of questions: ในการตั้งคำถามสำหรับทำแบบสอบถาม เราสามารถตั้งคำถามได้ทั้งแบบปลายเปิดและปลายปิดตามความเหมาะสม สิ่งสำคัญเป็นอันดับแรกของการตั้งคำถามคือภาษา ซึ่งจะหมายถึงเราควรจะตั้งคำถามโดยใช้ภาษาเดียวกับภาษาบ้านเกิดของผู้ตอบ เพื่อลดความเข้าใจไม่ตรงกัน ของผู้ถามและผู้ตอบ เราไม่ควรใช้ศัพท์เทคนิคเข้าใจยาก เราควรใช้ภาษาหรือคำที่เฉพาะเจาะจงไม่คลุมเคลือ
นอกจากเรื่องของภาษาที่ใช้แล้ว คำถามที่ตั้งขึ้นควรจะสั้นและกระชับ เป็นคำถามที่ไม่ได้สื่อความหมายไปในทางคัดค้านหรือท้วงติง และท้ายสุดคือเราควรเลือกคำถามที่ให้เหมาะสมกับผู้ตอบ (ผู้ใช้) แต่ละกลุ่ม

Questionnaire Design: ลำดับของคำถามเป็นสิ่งที่สำคัญมาก สำหรับการออกแบบแบบสอบถาม เราควรเริ่มจากคำถามที่มีการโต้เถียง/ขัดแย้งน้อย แต่เป็นคำถามที่สำคัญ แล้วค่อยเพิ่มความเข้มข้นของคำถามเพิ่มขึ้นเรื่อยๆ นอกจากนี้เราควรจะแบ่งกลุ่มคำถามที่มีความคล้ายคลึงกันหรือคำถามที่มีเนื้อหาสาระเหมือนกันให้อยู่ในกลุ่มเดียวกันและลำดับที่ใกล้ๆกัน นอกจากเรื่องที่เกี่ยวกับคำถามแล้ว เราจะต้องพยายามทำให้แบบสอบถามนั้นมีความดึงดูด และเพลิดเพลินในการตอบด้วย ด้วยเราอาจจะต้องทำการเตรียมเนื้อที่ว่างไว้สำหรับคำตอบไม่น้อยจนเกินไป หรือไม่มากจนเกินไป

Administering questionnaire: ในการออกแบบสอบถามให้กับผู้ใช้ เราควรต้องพิจารณาว่าใครควรได้รับแบบสอบถามบ้าง โดยที่ในการกำหนดเราจะต้องแน่ใจว่าเราไม่แจกแบบสอบถามขาดตกบกพร่อง ในบางครั้งเราอาจจัดกลุ่มของผู้ใช้และทำการแจกแบบสอบถามที่ละรายในแต่ละกลุ่ม หรือเราอาจใช้อีเมล์ในการตอบแบบสอบถามก็เป็นได้

การทบทวนเอกสารที่มีอยู่

โดยส่วนใหญ่การเก็บรวบรวมข้อมูลจะใช้วิธีการสัมภาษณ์ การประชุมแบบรวมกลุ่มและการใช้แบบสอบถามแต่อย่างไรก็ดี ยังมีวิธีการหนึ่งในการเก็บรวบรวมข้อมูล คือ การตรวจสอบหรือทบทวนเอกสารต่างๆ ที่ไม่ต้องยุ่งเกี่ยวหรือเกี่ยวข้องกับผู้ใช้มากนัก เราจะยุ่งเกี่ยวเพียงแค่สมาชิกในทีมงานผู้สร้างคลังข้อมูลเสียเป็นส่วนใหญ่ โดยปกติของเอกสารที่เราต้องทบทวนหรือตรวจสอบจะประกอบไปด้วยเอกสาร 2 ประเภท หลัก ๆ ดังนี้

Document from user departments:

เป็นเอกสารที่ได้มาจากผู้ใช้ ซึ่งเป็นเอกสารเกี่ยวกับขั้นตอนการทำงานและดำเนินธุรกิจ จากเอกสารดังกล่าว สิ่งแรกที่ต้องมองหาคือ รายงาน และหน้าจอ (Screen) ที่ผู้ใช้ใช้ในการดำเนินธุรกิจ ณ ปัจจุบัน ซึ่งอาจเป็นสิ่งที่ผู้ใช้ต้องการในการใช้คลังข้อมูล จากนั้นเราต้องมองทุกๆรายละเอียดที่เกี่ยวข้องกับฟังก์ชันการทำงานทั้งหมด เพื่อที่จะได้ทราบว่าสิ่งใด/กระบวนการใด/ข้อมูลใดเป็นสิ่งสำคัญและสิ่งจำเป็นสำหรับผู้ใช้ จากนั้นลองมองหาข้อมูลที่เกี่ยวกับการใช้ข้อมูลที่ได้จากรายงานต่างๆ ไปวิเคราะห์ และอื่นๆ จากการประยุกต์ใช้วิธีการตรวจสอบ/ทบทวนเอกสารที่ได้จากผู้ใช้จะช่วยให้ทีมผู้สร้างเข้าใจในฟังก์ชันการทำงานหรือการดำเนินธุรกิจมากขึ้น และอาจทำให้เราได้ข้อมูลเพิ่มเติมจากการสัมภาษณ์หรือเราอาจหาข้อมูลจากเอกสารเพื่อใช้สำหรับการเตรียมคำถาม เพื่อสัมภาษณ์ผู้ใช้ เป็นต้น

Documentation from IT:

เมื่อเราทำการตรวจสอบ/ทบทวนเอกสารที่ได้จากผู้ใช้ จะทำให้เราทราบถึงกระบวนการในการดำเนินธุรกิจ แต่สำหรับเอกสารที่ได้จากฝ่ายไอทีจะทำให้เราทราบถึงรายละเอียดต่างๆของระบบการดำเนินงาน ซึ่งถือว่าเป็นแหล่งข้อมูลของการสร้างคลังข้อมูล ข้อมูลที่จะได้รับจากเอกสารจะเกี่ยวกับองค์ประกอบของข้อมูลแต่ละส่วน พจนานุกรมข้อมูล (data dictionary) หรือแคตตาล็อกของข้อมูลจากแหล่งข้อมูล ซึ่งจะทำให้เรารู้และเข้าใจในโครงสร้างข้อมูล ซึ่งจากข้อมูลที่ได้รับจากเอกสารจะทำให้เรามีความเข้าใจเกี่ยวกับแหล่งข้อมูลทั้งหมด ทั้งในเรื่องของแพลตฟอร์ม และระบบปฏิบัติการที่ใช้ในระบบดำเนินงาน เป็นต้น

ถ้าเราทำการตรวจสอบ/ทบทวนเอกสารแล้วไปพูดคุยหรือสัมภาษณ์ผู้ดูแลหรือผู้ที่มีความเชี่ยวชาญในระบบไอที จะทำให้เราเข้าใจเกี่ยวกับกฎต่าง ๆ ทางธุรกิจและคุณค่าของแต่ละส่วนประกอบของข้อมูล นอกจากนั้นเรายังทราบเกี่ยวกับความเป็นเจ้าของข้อมูล (Data ownership) บุคคลที่มีหน้าที่จัดการเกี่ยวกับคุณภาพของข้อมูล รวมถึงขั้นตอนการทำงานและการเก็บรวบรวมข้อมูลของระบบดำเนินการอื่นอีกด้วย

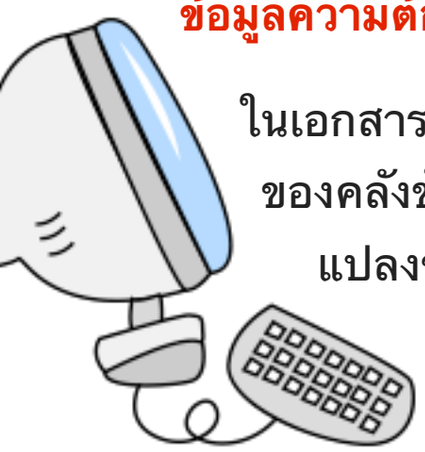
SECTION 6

ขอบเขตของความต้องการ

หลังจากการเก็บรวบรวมความต้องการของผู้ใช้ด้วยวิธีการสัมภาษณ์ นัดกลุ่มประชุม ใช้แบบสอบถาม และตรวจสอบ/ทบทวนเอกสารต่าง ๆ แล้ว ทีมผู้สร้างส่วนใหญ่จะละเลยการจัดทำเอกสารที่เป็นการสรุปความต้องการของผู้ใช้ โดยจะเปลี่ยนไปขั้นตอนของการทำงานถัดไปต่อในทันที ซึ่งโดยแท้จริงแล้วการลงรายละเอียดหรือเขียนรายละเอียดเกี่ยวกับความต้องการของผู้ใช้นั้นเป็นสิ่งสำคัญมาก

เช่น เมื่อเราได้บทสรุปของความต้องการที่เป็นเอกสารแล้ว เราสามารถส่งต่อให้กับบุคคลที่จะดำเนินในขั้นตอนต่อไปได้ ซึ่งถ้าในทีมผู้สร้างมีการเปลี่ยนแปลงสมาชิกโดยทำการถอดถอนบุคคลที่ทำการเก็บรวบรวมความต้องการออกจากทีม ถ้าเรามีเอกสารที่แสดงถึงรายละเอียดความต้องการจากผู้ใช้ที่จะทำให้ เราสามารถดำเนินการขั้นตอนต่อไปได้ โดยทำการอ่านข้อมูลที่ได้จากเอกสารเหล่านั้น นอกจากนั้นเราสามารถตรวจสอบความสมบูรณ์ของรายละเอียดของความต้องการกับผู้ใช้โดยใช้เอกสารที่สร้างขึ้นเป็นสื่อกลางได้

จากประโยชน์ที่เราจะได้รับจากการทำเอกสารที่แสดงรายละเอียดของความต้องการจากผู้ใช้ที่มีต่อการสร้างคลังข้อมูล เราควรจะทำเอกสารที่มีข้อมูลที่ครบถ้วนสมบูรณ์ซึ่งควรจะต้องประกอบไปด้วยรายละเอียดต่าง ๆ ดังต่อไปนี้



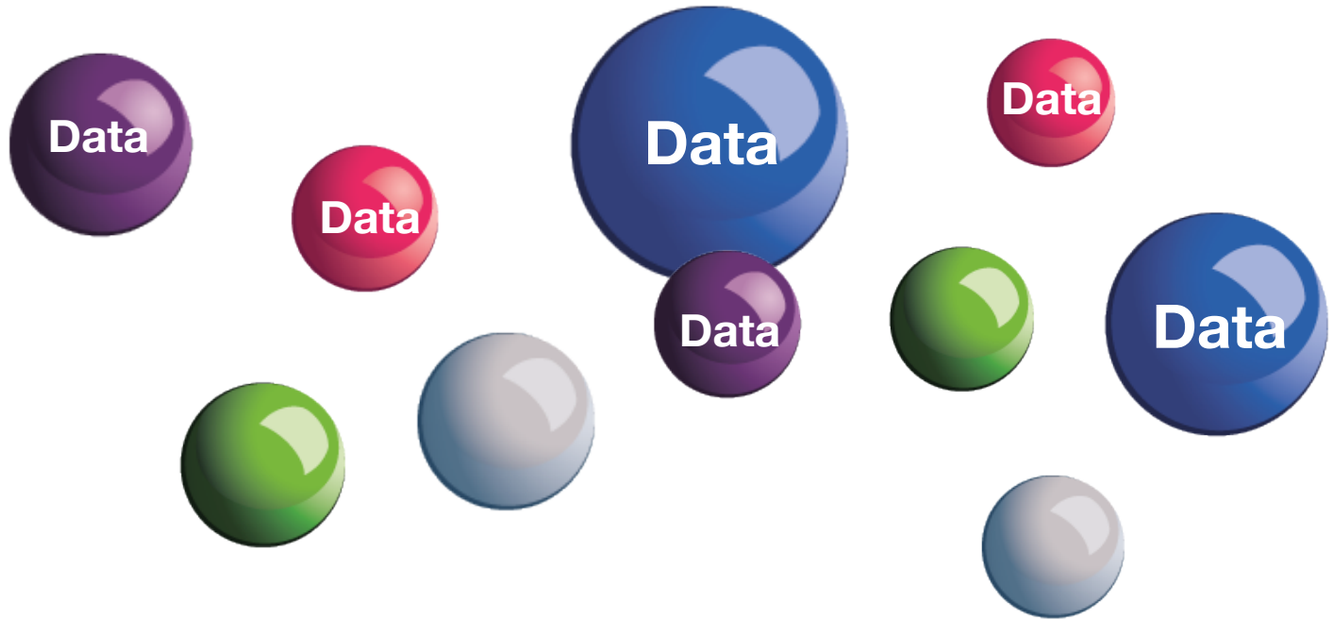
ข้อมูลความต้องการที่เกี่ยวข้องกับแหล่งข้อมูล

ในเอกสารควรจะมีรายละเอียดเกี่ยวกับแหล่งข้อมูล เนื่องจากในการสร้างคลังข้อมูลเราจะใช้ข้อมูลจากแหล่งข้อมูลเป็นอินพุตของคลังข้อมูล โดยเราจะทำการเก็บรวบรวมข้อมูลจากแหล่งข้อมูลต่างๆ ทำการรวบรวมข้อมูลเหล่านั้นเข้าด้วยกัน ทำการแปลงข้อมูลให้เป็นมาตราฐานหรือให้อยู่ในรูปแบบที่เหมาะสม และทำการถ่ายโอนข้อมูลเหล่านั้นเข้าสู่คลังข้อมูล ซึ่งจากความสำคัญของรายละเอียดของแหล่งข้อมูลเอกสารที่เก็บรวบรวมความต้องการควรจะต้องประกอบไปด้วย

- แหล่งข้อมูลที่ยังคงทำงานอยู่ และสามารถเก็บรวบรวมข้อมูลจากแหล่งข้อมูลนั้น ๆ ได้
- โครงสร้างข้อมูลของแต่ละแหล่งข้อมูล
- ที่ตั้งของแหล่งข้อมูล
- ระบบปฏิบัติการ เครือข่าย โปรโตคอล และสถาปัตยกรรมแหล่งข้อมูล
- ขั้นตอนการสกัดข้อมูล
- ข้อมูลย้อนหลังที่สามารถหาได้

ข้อมูลความต้องการที่เกี่ยวข้องกับการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล

หลังจากที่เราทำการเก็บรวบรวมข้อมูลเกี่ยวกับแหล่งข้อมูลแล้ว เราควรจะทำการกำหนดวิธีการในการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล (data transformation) ให้อยู่ในรูปแบบที่เหมาะสมต่อการจัดเก็บข้อมูลในคลังข้อมูล ดังนั้นเอกสารที่เกี่ยวกับรายละเอียดของความต้องการ เราควรใส่รายละเอียดของการเปลี่ยนรูปข้อมูลลงไปด้วย เพื่อช่วยในการเชื่อมโยงความสัมพันธ์ของข้อมูลจากแหล่งข้อมูลกับข้อมูลที่ถูกเก็บอยู่ในคลังข้อมูลนอกจากนั้นยังทำให้เราทราบถึงรายละเอียดของการรวมข้อมูลเข้าด้วยกัน (merging) การแปลงข้อมูล (conversion) และการแบ่งข้อมูลเป็นส่วนๆ (Splitting) อีกด้วย



ข้อมูลความต้องการที่เกี่ยวข้องกับการจัดเก็บข้อมูล

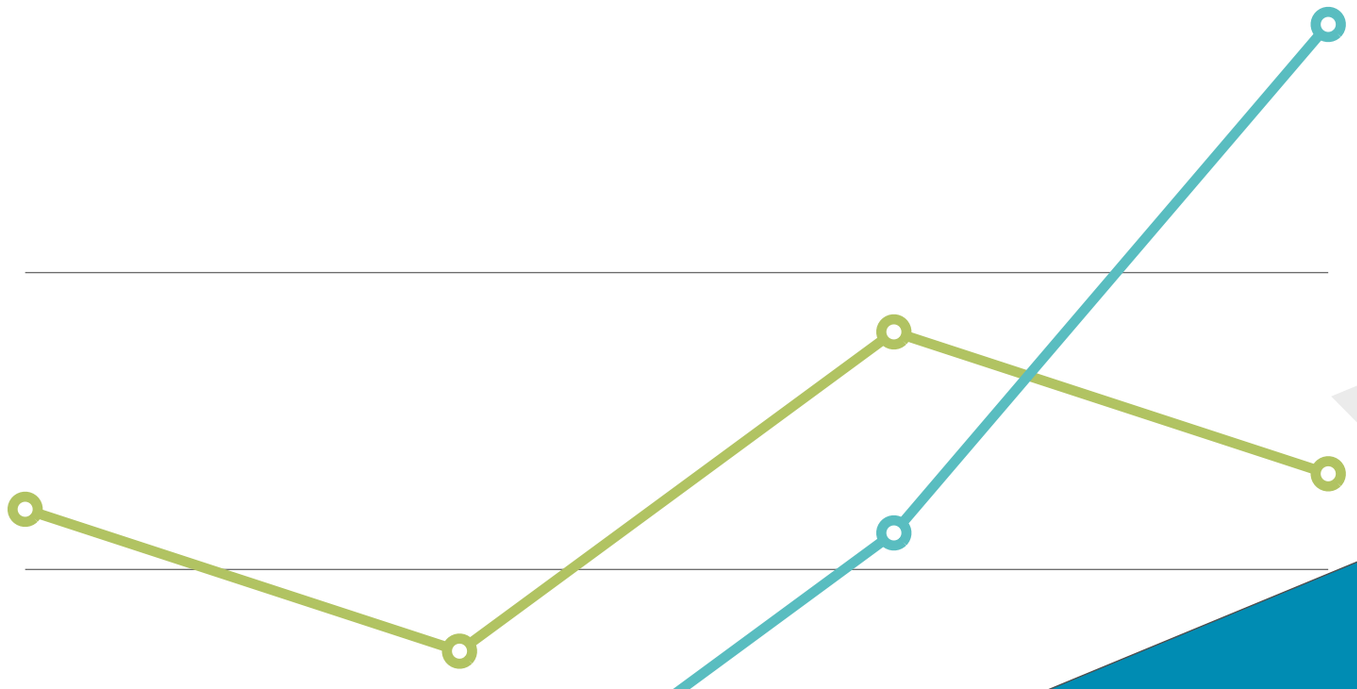
หลังจากที่ทำการสัมภาษณ์ผู้ใช้ที่เกี่ยวข้องกับความต้องการต่าง ๆ แล้ว เมื่อเราทำการพิจารณาถึงรายงานที่ต้องการ คิวรีที่ต้องการทำ ให้เราทราบถึงระดับของความละเอียดของข้อมูลที่จะเก็บไว้ในคลังข้อมูล ทราบถึงจำนวนดาต้ามาร์ทที่เราต้องสร้างขึ้นเพื่อสนับสนุนการทำงานของผู้ใช้ ทราบถึงรายละเอียดมาตรวัดผลสัมฤทธิ์ของแต่ละส่วนงานหรือแต่ละขั้นตอนการดำเนินธุรกิจ ทราบถึงมิติทางธุรกิจ และวิธีการจัดเก็บข้อมูล ไว้ในคลังข้อมูลและอาจรวมไปถึงพื้นที่สำหรับเก็บข้อมูลที่เพิ่มขึ้น

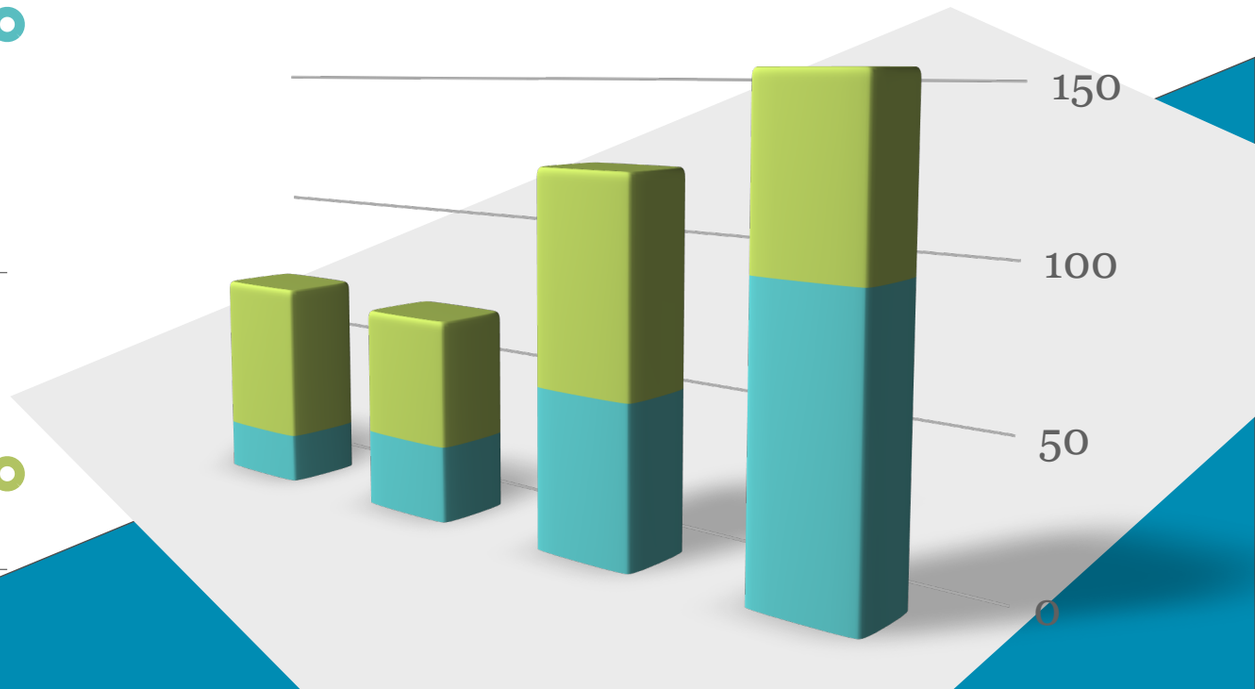
จากรายละเอียดข้างต้นเราควรต้องเก็บรายละเอียดเกี่ยวกับความต้องการในการจัดเก็บข้อมูลไว้ในเอกสารที่แสดงรายละเอียดเกี่ยวกับความต้องการด้วย รวมถึงเราควรทำการประมาณและประเมินเบื้องต้นเกี่ยวกับปริมาณข้อมูลแล้วทำการบันทึกลงไปในเอกสารเพื่อแสดงถึงความต้องการทางด้านฮาร์ดแวร์ที่ใช้ในการจัดเก็บข้อมูล

ข้อมูลความต้องการที่เกี่ยวข้องกับการส่งผ่านข้อมูล

เอกสารเกี่ยวกับรายละเอียดความต้องการควรจะแสดงถึงความต้องการในการเรียกใช้งานข้อมูลลักษณะต่างๆ เช่น

- การวิเคราะห์ข้อมูลแบบเจาะลึก (drill- down analysis)
- การวิเคราะห์ข้อมูลโดยการหาผลสรุป (roll- up analysis)
- การวิเคราะห์ข้อมูลโดยการตัดส่วน เพื่อเปลี่ยนมุมมองไปยังรูปแบบต่างๆ (slicing and dicing analysis)
- การวิเคราะห์ข้อมูลแบบเฉพาะเจาะจง (ad hoc analysis)





ข้อมูลความต้องการที่เกี่ยวข้องกับไดอะแกรมแพคเกจข้อมูล

อย่างที่เราทราบกันดีว่า ไดอะแกรมที่แสดงแพคเกจข้อมูลจะช่วยให้เราสามารถแยกความแตกต่างระหว่างระบบดำเนินการและคลังข้อมูลได้ รวมถึงช่วยให้เราสามารถกำหนดความต้องการต่างๆ ให้เป็นรูปธรรมมากขึ้น ไดอะแกรมแพคเกจข้อมูลจะบรรจุไปด้วยมาตราวัดความสำเร็จที่สำคัญ ซึ่งมาตราวัดเหล่านี้จะถูกใช้ในการวัดประสิทธิภาพของการดำเนินธุรกิจ ไดอะแกรมจะประกอบด้วยมิติทางธุรกิจต่างๆ รายละเอียดของการวิเคราะห์ข้อมูลในสิ่งต่างๆ ซึ่งจากข้อมูลที่ถูกบรรจุอยู่ในไดอะแกรมแพกเกจข้อมูล และประโยชน์ที่ได้รับจากไดอะแกรมแพกเกจข้อมูล เราควรจะเก็บข้อมูลรายละเอียดเกี่ยวกับไดอะแกรมแพกเกจข้อมูลไว้ในเอกสารที่แสดงถึงความต้องการด้วย

จากข้อมูลในแง่มุมหรือขั้นตอนต่าง ๆ ที่ถูกเก็บไว้ในเอกสารที่แสดงถึงความต้องการจากผู้ใช้ในการใช้งานคลังข้อมูล เราสามารถจัดเรียงข้อมูลต่างๆ ให้เป็นลำดับที่สามารถเข้าใจง่าย และมีข้อมูลที่สมบูรณ์ได้ดังนี้

1. Introduction จะประกอบด้วยวัตถุประสงค์และขอบเขตการสร้างคลังข้อมูล

2. General requirement descriptions จะประกอบด้วยข้อมูลความต้องการจากแหล่งข้อมูลต่างๆ บทสรุปจากบทสัมภาษณ์ และระบุถึงชนิดของข้อมูลที่ต้องการหรือถูกเรียกใช้จากคลังข้อมูล

3. Specific requirements จะเป็นข้อมูลเกี่ยวกับแหล่งข้อมูลที่ต้องการว่าเราต้องการที่จะใช้ข้อมูลจากแหล่งข้อมูลใดบ้างรายละเอียดความต้องการเกี่ยวกับการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลและความต้องการเกี่ยวกับการจัดเก็บข้อมูล รวมถึงการส่งผ่าน/เรียกใช้ข้อมูลตามที่ผู้ใช้ต้องการ

4. Information package ข้อมูลเกี่ยวกับแพคเกจข้อมูล

5. Other requirements เป็นข้อมูลความต้องการทั่วไป เช่น ความถี่ในการสกัดข้อมูลจากแหล่งข้อมูล วิธีการถ่ายโอนข้อมูล เป็นต้น

6. User expectations เกี่ยวกับความคาดหวังเกี่ยวกับโอกาสต่างๆที่จะได้รับจากการใช้คลังข้อมูล

7. User participations and sign off เป็นลิสต์เกี่ยวกับงานและกิจกรรมต่างๆที่ผู้ใช้คาดหวังว่าจะมีส่วนร่วมในกระบวนการทั้งหมดของการสร้างคลังข้อมูล

8. General implementation plan จะเป็นข้อมูลเกี่ยวกับแผนในการดำเนินการสร้างคลังข้อมูล

SECTION 7

คำถามท้ายบท

1. อะไรคือความแตกต่างระหว่างความต้องการในการสร้างระบบการดำเนินงานและระบบคลังข้อมูล
2. จงอธิบายเหตุผลว่าเพราะเหตุใดมิติทางธุรกิจจึงเป็นสิ่งที่มีประโยชน์ในการกำหนดความต้องการสำหรับการสร้างคลังข้อมูล
3. จงอธิบายเกี่ยวกับสิ่งที่ถูกบรรจุอยู่ในแพคเกจข้อมูล
4. ลำดับชั้นและหมวดหมู่ของข้อมูลในแต่ละมิติทางธุรกิจมีลักษณะเป็นอย่างไร จงยกตัวอย่าง 3 ตัวอย่างเพื่ออธิบาย
5. จงยกตัวอย่างมาตรวัดหรือตัวบ่งชี้ในการวัดประสิทธิภาพในการดำเนินธุรกิจมา 5 ตัวอย่าง
6. จงยกตัวอย่างของผู้ใช้หรือผู้ที่มีส่วนร่วมในการสัมภาษณ์เพื่อทำการเก็บรวบรวมความต้องการ
7. วิธีในการถามคำถามเพื่อเก็บรวบรวมความต้องการมีกี่ประเภท อะไรบ้าง แต่ละวิธีมีข้อดีและข้อเสียอย่างไรบ้าง
8. ในการทำวิจัยเบื้องต้นก่อนการสัมภาษณ์ เราจะต้องพิจารณาถึงสิ่งใดบ้าง
9. เพราะเหตุใดการทบทวนเอกสารเก่า ๆ ถึงมีความสำคัญ และในการดำเนินการดังกล่าวเราจะสามารถทราบถึงข้อมูลอะไรบ้าง
10. จงแจกแจงข้อมูลหรือเนื้อหาของข้อมูลที่ควรถูกบันทึกอยู่ในเอกสารสำหรับเก็บรวบรวมข้อมูล

CHAPTER 7

การสร้างแบบจำลองมิติต่าง ๆ

SECTION 1

แผนการสอนประจำบท

วัตถุประสงค์ของบทเรียน

- ทำความเข้าใจการกำหนดความต้องการที่จะส่งผลต่อการออกแบบโครงสร้างข้อมูลในคลังข้อมูล
- แนะนำการสร้างแบบจำลองมิติต่างๆ
- ศึกษาเกี่ยวกับพื้นฐานของ STAR-schema Snowflake-schema และ Family of STARs schema
- ศึกษาเกี่ยวกับการสร้างตารางข้อมูลสำหรับการรวบยอดข้อมูล
- ศึกษาเกี่ยวกับความเปลี่ยนแปลงของข้อมูลที่จะเกิดขึ้นกับการสร้างแบบจำลองมิติต่างๆ

เนื้อหาของบทเรียน

เนื้อหาในบทนี้จะประกอบด้วย การสร้างแบบจำลองข้อมูล Star-schema Snowflake-schema การรวบยอดข้อมูลใน fact table การสร้างแบบจำลองข้อมูล Family of stars ความเปลี่ยนแปลงของข้อข้อมูลที่จะเกิดขึ้นกับการสร้างแบบจำลองมิติต่าง ๆ ทั้งในรูปแบบการเปลี่ยนแปลงอย่างช้าๆ และอย่างรวดเร็ว

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารประกอบการสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

กิจกรรมการเรียน-การสอน

- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารประกอบการสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

การวัดและประเมินผล

- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

SECTION 2

บทนำ

หลังจากทำการเก็บรวบรวมความต้องการจากผู้ใช้แล้ว เราจะทำการสร้างโครงสร้างและสร้างความสัมพันธ์ของข้อมูลสำหรับคลังข้อมูลโดยการรวมข้อมูลต่าง ๆ เข้าด้วยกัน โดยเริ่มจากการเก็บรวบรวมความต้องการจากผู้ใช้ที่ซึ่งเราจะได้เอกสารที่บ่งบอกถึงรายละเอียดของความต้องการและไดอะแกรมแพคเกจข้อมูล จากนั้นเราจะทำการออกแบบและสร้าง ***“แบบจำลองมิติต่างๆ (Dimensional model)”*** ที่สามารถตอบคำถามต่างๆของผู้ใช้ จัดการเกี่ยวกับมุมมองต่างๆทางธุรกิจและสามารถแสดงแนวโน้มทางธุรกิจได้

เมื่อเราทำการพิจารณาไดอะแกรมแพคเกจข้อมูลภายใต้หัวข้อทางธุรกิจหนึ่ง ๆ ที่ประกอบไปด้วย

1

ตัวชี้วัดต่างๆ
(measurements or metrics)

2

มิติทางธุรกิจ
(business dimensions)
ที่ประกอบไปด้วยแอทริบิวต่างๆ

เราจะต้องทำการตรวจสอบหรือกำหนดข้อมูลที่จะใช้สำหรับสร้างแบบจำลองมิติต่าง ๆ ที่ประกอบไปด้วยองค์ประกอบต่าง ๆ ดังนี้

1 ทำการเลือกหัวข้อต่างๆ (subjects) จากไดอะแกรมแพคเกจข้อมูลที่มี เพื่อทำการออกแบบโครงสร้างทางตรรกะ

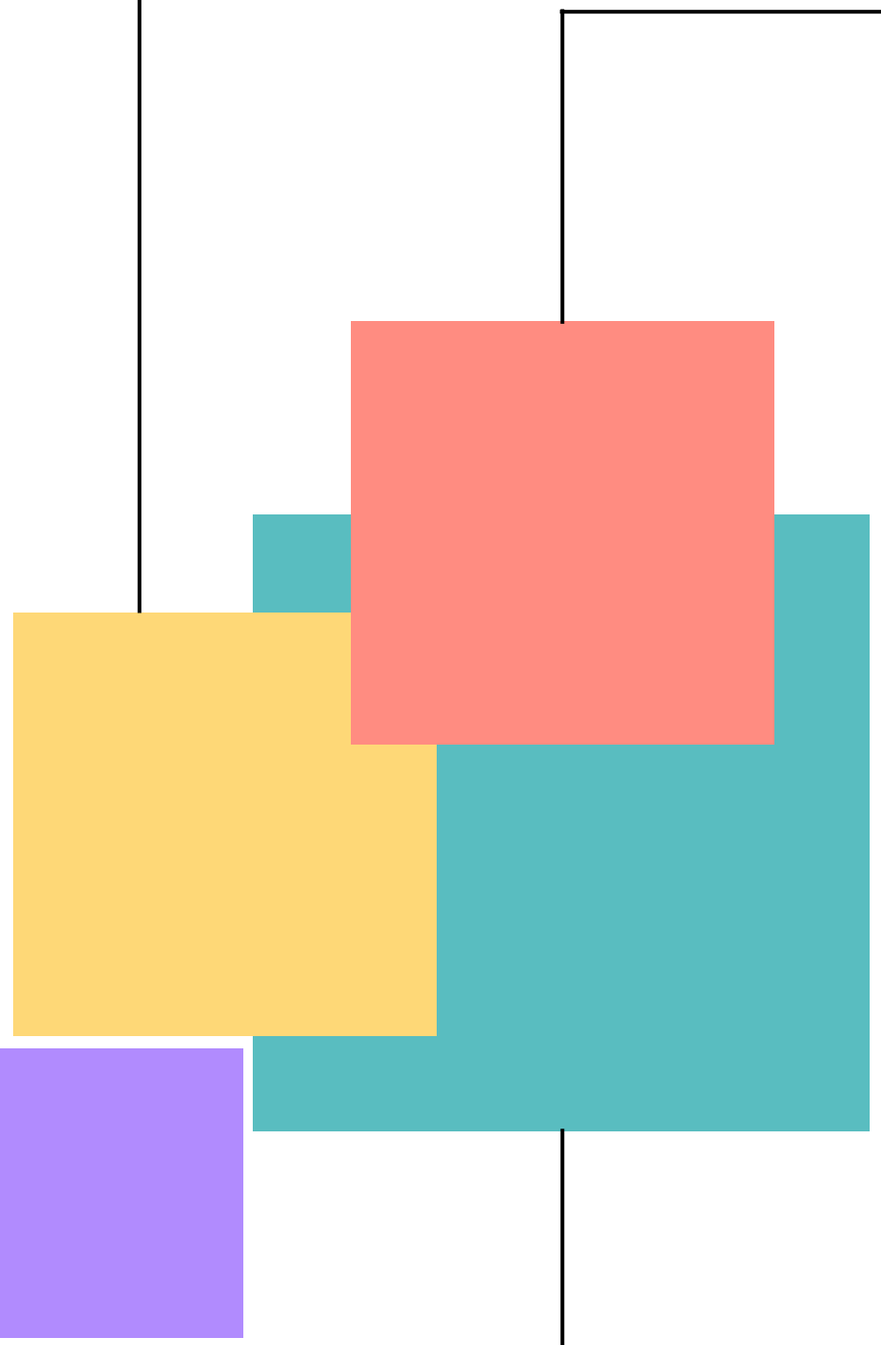
2 ทำการกำหนดระดับของความละเอียดของข้อมูลที่จะถูกเก็บอยู่ในแบบจำลองมิติต่างๆ

3 ทำการเลือกมิติทางธุรกิจ ที่จะถูกรวมไว้ในแบบจำลองมิติต่าง ๆ (เช่น รายการสินค้า ลูกค้า เวลา เป็นต้น) จากนั้นทำการตรวจสอบว่าข้อมูลแต่ละส่วนที่มาจากแต่ละมิติทางธุรกิจนั้นมีความสอดคล้องกับข้อมูลส่วนอื่น ๆ หรือไม่

4 ทำการเลือกตัวชี้วัดหรือหน่วยของการวัดที่จะถูกรวมไว้ในแบบจำลองมิติต่างๆ

5 ทำการกำหนดระยะเวลาที่จะใช้เก็บข้อมูลย้อนหลัง

หลังจากที่เราทำการเลือกและกำหนดสิ่งต่าง ๆ ในขั้นตอนเริ่มต้นแล้ว เราจะทำการพิจารณาไดอะแกรมแพคเกจข้อมูลอีกครั้งหนึ่ง ลองพิจารณาไดอะแกรมแพคเกจข้อมูลการขายรถยนต์ของบริษัทผู้ผลิตรถยนต์ในรูปที่ 7-1 ที่ประกอบไปด้วยตัวชี้วัดต่าง ๆ เช่น ราคาขายสินค้า (actual sale price) ที่เป็นค่าความจริงที่บ่งบอกถึงราคาขายจริงๆ ซึ่งจากตัวชี้วัด (measure) หรือค่าความจริง (fact) จากไดอะแกรมแพคเกจข้อมูลเราจะสามารถสร้างตารางสำหรับเก็บข้อมูลตัวชี้วัดเหล่านั้นที่เรียกว่า ***"fact table"*** (ดังแสดงในรูปที่ 7-1) หลังจากทำการพิจารณาเกี่ยวกับตัวชี้วัดทั้งหมดแล้ว ขั้นตอนต่อไปจะทำการพิจารณาแต่ละมิติทางธุรกิจที่มีการเก็บรายละเอียดของข้อมูลไว้ในแต่ละมิติ โดยรายละเอียดของข้อมูลที่ถูกเก็บอาจอยู่ในรูปแบบของลำดับชั้นของข้อมูล *(hierarchy)* หรือหมวดหมู่ของข้อมูล *(category)* เมื่อพิจารณาแต่ละมิติทางธุรกิจ เราจะต้องทำการสร้างตารางสำหรับจัดเก็บข้อมูลสำหรับแต่ละมิติทางธุรกิจที่เรียกว่า ***"dimension table"***



Automaker Sales

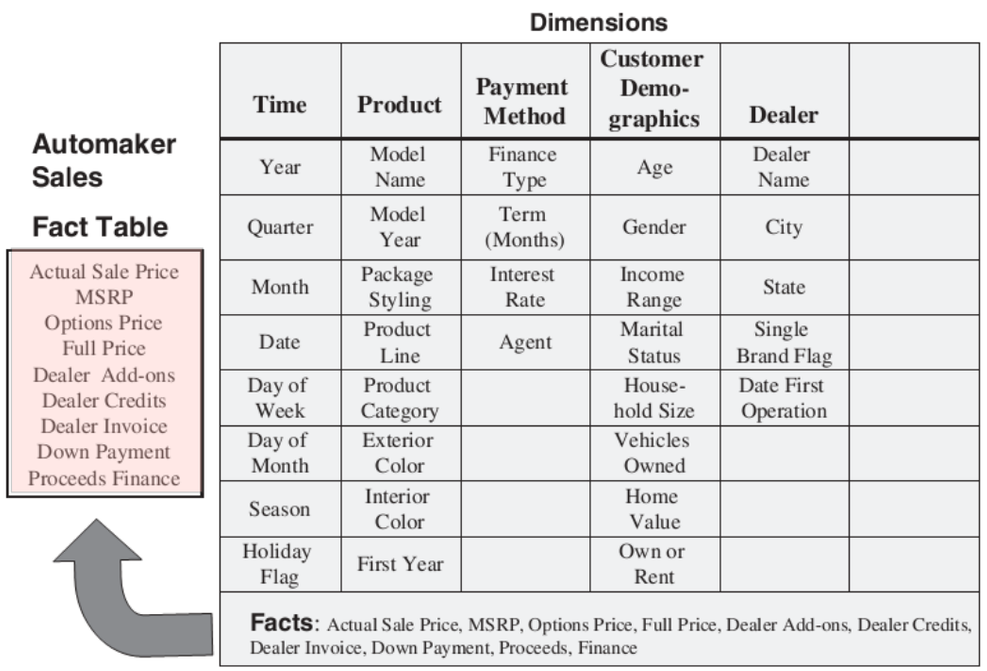
Fact Table

Actual Sale Price
MSRP
Options Price
Full Price
Dealer Add-ons
Dealer Credits
Dealer Invoice
Down Payment
Proceeds Finance

Dimensions

Time	Product	Payment Method	Customer Demo-graphics	Dealer	
Year	Model Name	Finance Type	Age	Dealer Name	
Quarter	Model Year	Term (Months)	Gender	City	
Month	Package Styling	Interest Rate	Income Range	State	
Date	Product Line	Agent	Marital Status	Single Brand Flag	
Day of Week	Product Category		House-hold Size	Date First Operation	
Day of Month	Exterior Color		Vehicles Owned		
Season	Interior Color		Home Value		
Holiday Flag	First Year		Own or Rent		
Facts: Actual Sale Price, MSRP, Options Price, Full Price, Dealer Add-ons, Dealer Credits, Dealer Invoice, Down Payment, Proceeds, Finance					

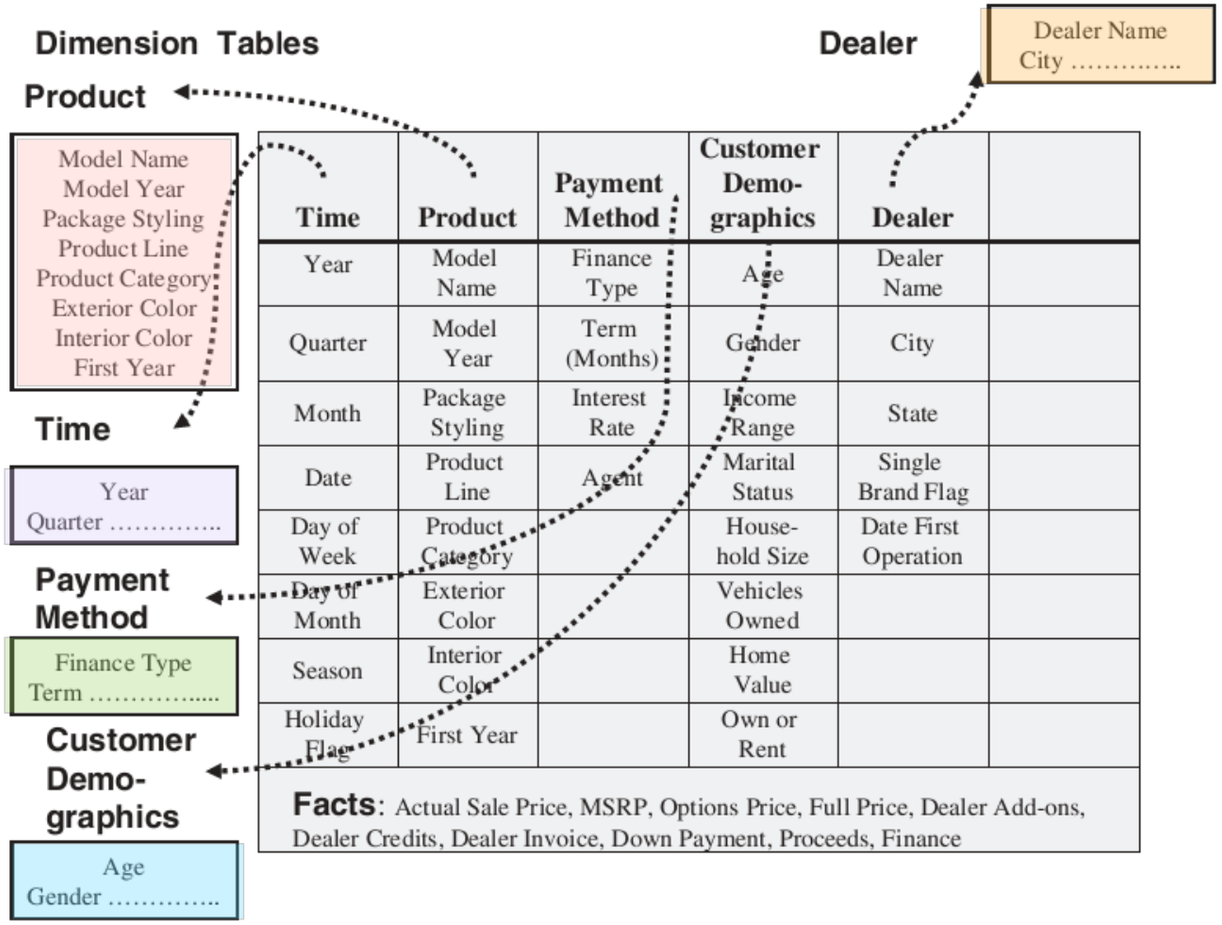
รูปที่ 7- 1 การสร้าง fact table สำหรับการขายรถยนต์ของบริษัทผู้ผลิตรถยนต์จากแพคเกจข้อมูล



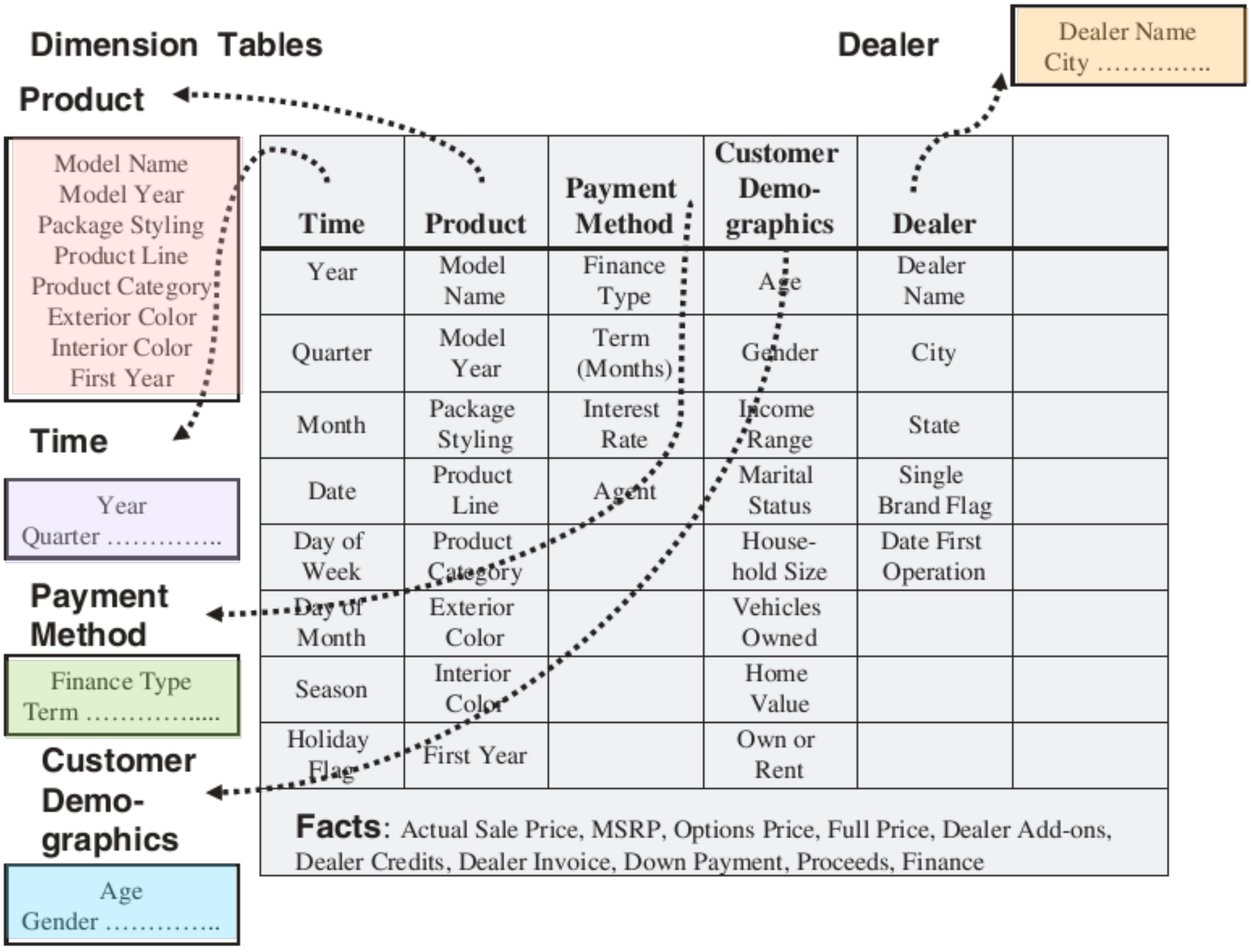
รูปที่ 7- 1 การสร้าง fact table สำหรับการขายรถยนต์ของบริษัทผู้ผลิตรถยนต์จากแพคเกจข้อมูล

จากรูปที่ 7-1 เมื่อเราทำการพิจารณามิติรายการสินค้า (product dimension) เราจะสามารถสร้างตารางสำหรับจัดเก็บข้อมูลในมิติรายการสินค้าที่เรียกว่า **“product dimension table”** ได้ดังแสดงในรูปที่ 7-2 ซึ่งจากรูปจะแสดงตารางที่สร้างขึ้นสำหรับมิติรายการสินค้าที่จะมีข้อมูลรายละเอียดเหมือนกันกับรายละเอียดของมิติรายการสินค้าในไดอะแกรมแพคเกจข้อมูล จากนั้นเราจะทำการสร้างตารางของมิติทางธุรกิจอื่นๆ เมื่อเราทำการสร้างตารางของทุกมิติทางธุรกิจในไดอะแกรมแพคเกจข้อมูลแล้ว เราจะได้ 1 fact table ที่ใช้ในการจัดเก็บข้อมูลตัวชี้วัดจากไดอะแกรมแพคเกจข้อมูล และกลุ่มของ dimension table ที่ใช้ในการจัดเก็บข้อมูลรายละเอียดสำหรับแต่ละมิติทางธุรกิจ จากนั้นเราจะทำการประกอบหรือเชื่อมต่อตารางทั้ง 2 ชนิดเข้าด้วยกันเพื่อสร้างเป็นแบบจำลองมิติต่างๆ เมื่อทำการเชื่อมต่อระหว่างตารางทั้งสองชนิดเข้าด้วยกัน เราจะสามารถวิเคราะห์ข้อมูลในมิติทางธุรกิจหนึ่งๆหรือข้ามมิติก็ได้ โดยใช้แอทริบิวต่าง ๆ ใน dimension table

เพื่อให้เข้าใจเกี่ยวกับคิวรีที่ใช้ในการถามคำถามมากขึ้น ลองพิจารณาตัวอย่างคิวรีสำหรับถามคำถามเกี่ยวกับยอดขายรถยนต์ของบริษัทผู้ผลิตรถยนต์ที่จะถามว่า “ยอดขายรถ Jeep Cherokee รุ่นปี 2007 ที่ขายได้ที่ BigSam Auto dealer โดยทำการขายให้กับลูกค้าที่มีบ้านเป็นของตัวเอง และซื้อแบบผ่อนชำระเพียงแค่ 3 ปี กับบริษัท Daimler-Chrysler” ซึ่งผู้ใช้จะต้องการยอดเกี่ยวกับ actual sale price, MSRP และ Full price ด้วย ซึ่งจากคิวรีข้างต้นเราต้องทำการวิเคราะห์ข้อมูลตัวชี้วัดจาก fact table ทั้ง 3 ค่าข้างต้นคือ actual sale price, MSRP และ Full price ที่เกี่ยวข้องกับแอทริบิวต่างๆ ในหลายๆมิติทางธุรกิจด้วยกัน
ซึ่งแต่ละแอทริบิวจะทำหน้าที่เสมือนเงื่อนไขที่ใช้สำหรับกรองข้อมูลที่เกี่ยวข้องกับคำถามจากคิวรี



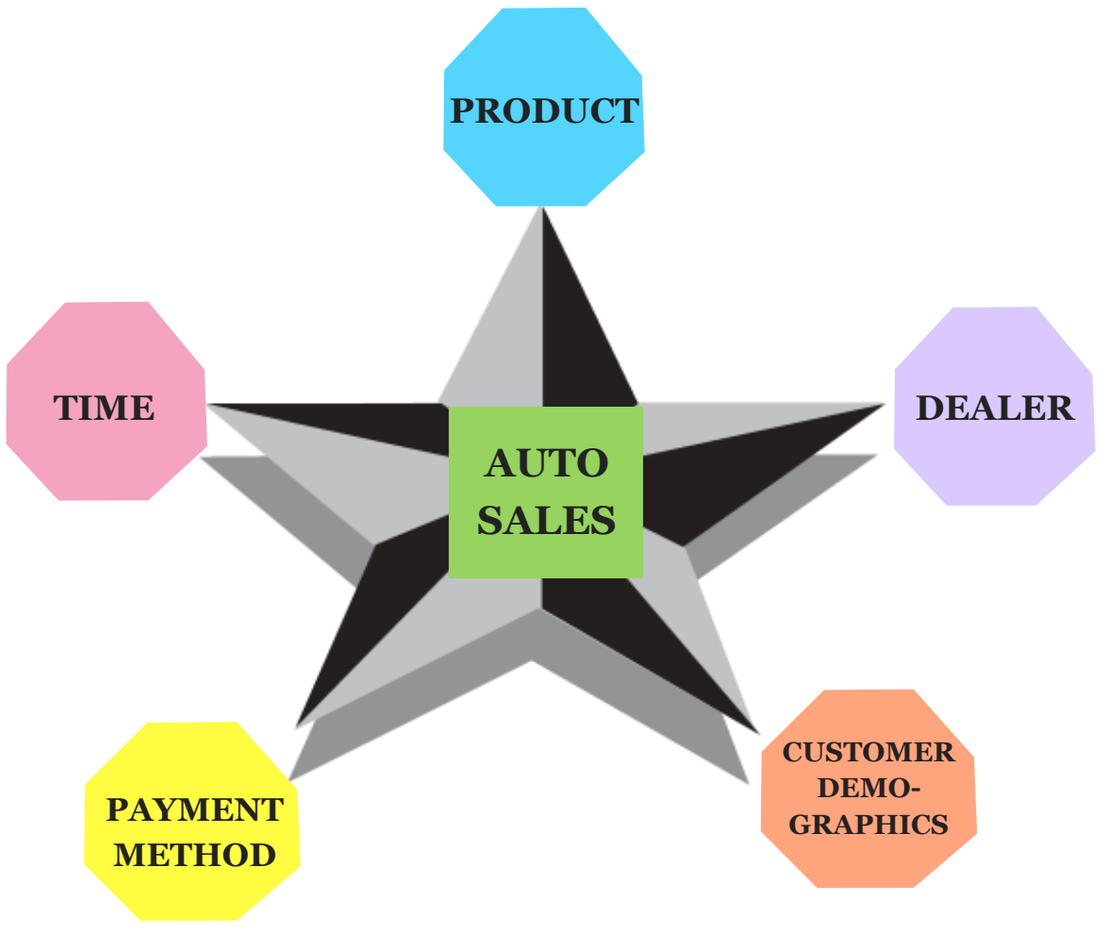
รูปที่ 7- 2 การสร้าง dimension table สำหรับมิติต่าง ๆ ทางธุรกิจที่เกี่ยวข้องกับการขายรถยนต์



รูปที่ 7- 2 การสร้าง dimension table สำหรับมิติต่าง ๆ ทางธุรกิจที่เกี่ยวข้องกับการขายรถยนต์

จากการวิเคราะห์ดังกล่าว จะเห็นว่าเราจะต้องทำการจัดการเชื่อมความสัมพันธ์ระหว่าง fact และ dimension table เข้าด้วยกัน โดยการพิจารณาเกณฑ์ต่าง ๆ ดังต่อไปนี้

- หลังจากการเชื่อมต่อความสัมพันธ์แล้ว ผู้ใช้ควรจะสามารถเข้าถึงข้อมูลในแบบจำลองมิติต่าง ๆ ได้อย่างรวดเร็วที่สุด
- หลังจากการเชื่อมต่อความสัมพันธ์แล้วแบบจำลองมิติต่างๆจะต้องสามารถให้บริการเกี่ยวกับการวิเคราะห์และการค้นคืนข้อมูลให้กับคิวรีได้อย่างดีที่สุด
- หลังจากการเชื่อมต่อความสัมพันธ์แล้ว เราจะสามารถเห็นความสัมพันธ์ของระหว่าง dimension table กับ fact table ได้
- หลังจากการเชื่อมต่อความสัมพันธ์แล้ว จะทำให้ dimension table ในแบบจำลองมิติต่างๆ นั้นสามารถเชื่อมต่อกับ fact table ได้อย่างเท่าเทียมกัน
- หลังจากการเชื่อมต่อความสัมพันธ์แล้ว จะทำให้ผู้ใช้สามารถเรียกดูข้อมูลแบบเจาะลึก (drill down) และแบบสรุปรวบยอด (roll up) ได้ผ่านทางลำดับชั้นต่างๆทางลำดับชั้นของแต่ละ dimension table



รูปที่ 7-3 ตัวอย่าง Star schema

จากเงื่อนไขข้างต้น เราจะเห็นว่าการวาง fact table ไว้ตรงกลางแล้วนำ dimension table ต่าง ๆ มาล้อมรอบ fact table จะทำให้เราได้แบบจำลองมิติต่างๆที่ตรงตามเงื่อนไขทั้งหมด การเชื่อมต่อดังกล่าวจะทำให้แต่ละ dimension table จะมีความสัมพันธ์โดยตรงกับ fact table และมีรูปร่างคล้ายกับดาวที่มี fact table อยู่ในตำแหน่งใจกลางดาว โดยที่แต่ละ dimension table จะอยู่ที่แต่ละมุมของดาว จากโครงสร้างลักษณะดังกล่าว เราจะสามารถเรียกแบบจำลองมิติต่าง ๆ ที่มีลักษณะคล้ายดาวว่า ***"Star schema"*** ดังแสดงในรูปที่ 7-3

SECTION 3

Star Scehma

Star schema

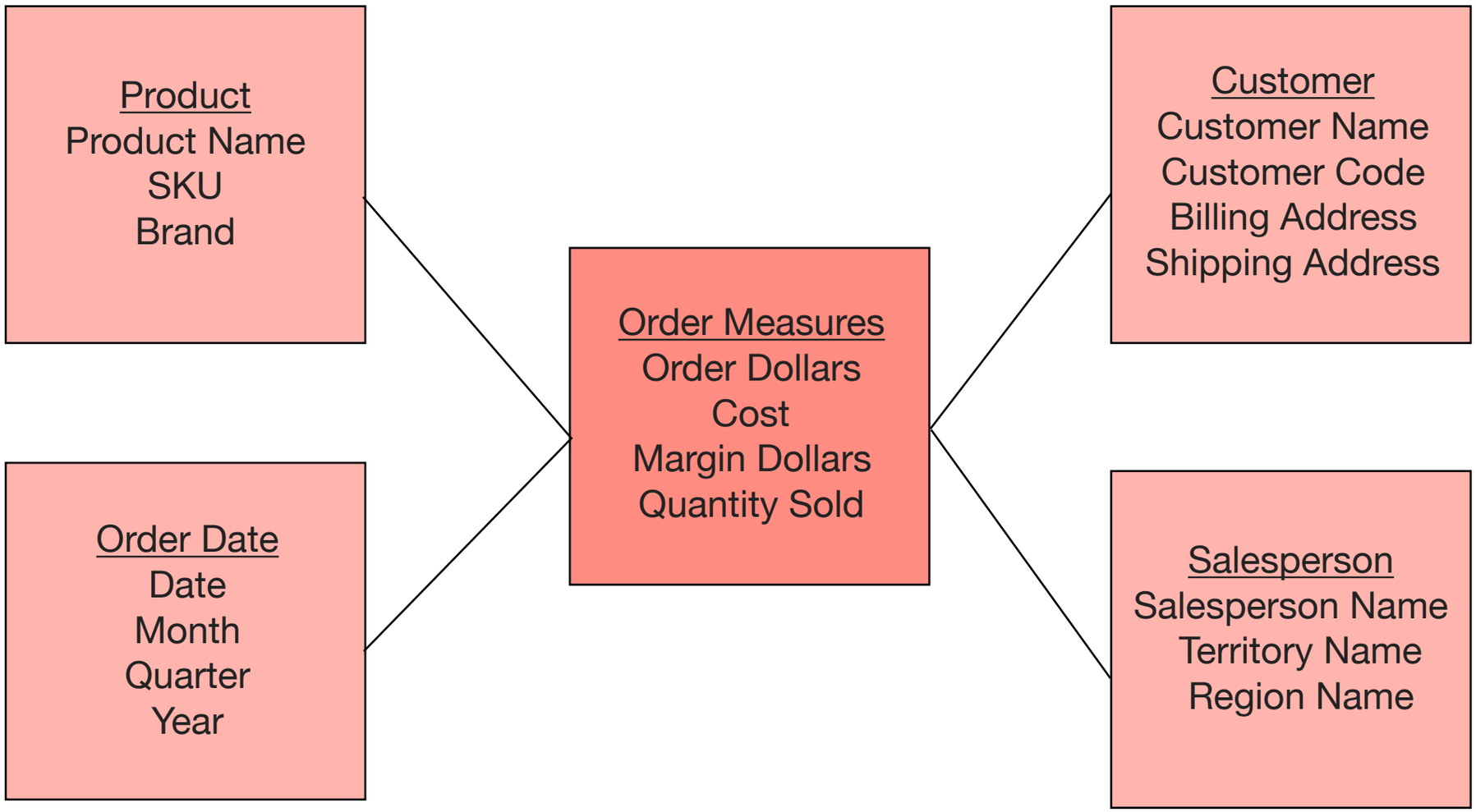
จากส่วนที่แล้วเราจะเข้าใจเกี่ยวกับแบบจำลองมิติต่างๆ และส่วนประกอบของแบบจำลองมิติต่างๆ Star schema ก็เป็นแบบจำลองมิติต่าง ๆ ชนิดหนึ่งที่ไม่ซับซ้อนและง่ายที่จะเข้าใจ Star schema จะประกอบไปด้วย 1 fact table ที่ประกอบไปด้วยค่าความจริง ตัวบ่งชี้หรือมาตรวัดผลสัมฤทธิ์ที่เกี่ยวเนื่องกับหัวข้อทางธุรกิจที่เราสนใจ และหลาย dimension table ที่สอดคล้องกับ fact table นั้นๆ จากส่วนประกอบของ star schema เราสามารถสร้างแบบจำลองมิติต่าง ๆ ได้โดยจัดวาง fact table ไว้ตรงกลางแล้วทำการล้อมรอบ fact table ด้วย dimension table ต่างๆ ดังแสดงในรูปที่ 7-4 ซึ่งแสดงถึง star schema อย่างง่ายสำหรับการวิเคราะห์การสั่งซื้อสินค้าที่ประกอบไปด้วยหนึ่ง fact table ที่มีมาตรวัดเป็นจำนวนเงินในการสั่งซื้อสินค้า ค่าใช้จ่าย ผลกำไร และจำนวนชิ้นสินค้าที่ถูกสั่งซื้อ และประกอบไปด้วย 4 dimension table ที่เกี่ยวข้องกับรายการสินค้า (product dimension table) ลูกค้า (customer dimension table) วันที่ทำการสั่งซื้อสินค้า (order date dimension table) และพนักงานขาย (sales person dimension table)

product dimension table

customer dimension table

order date dimension table

sales person dimension table

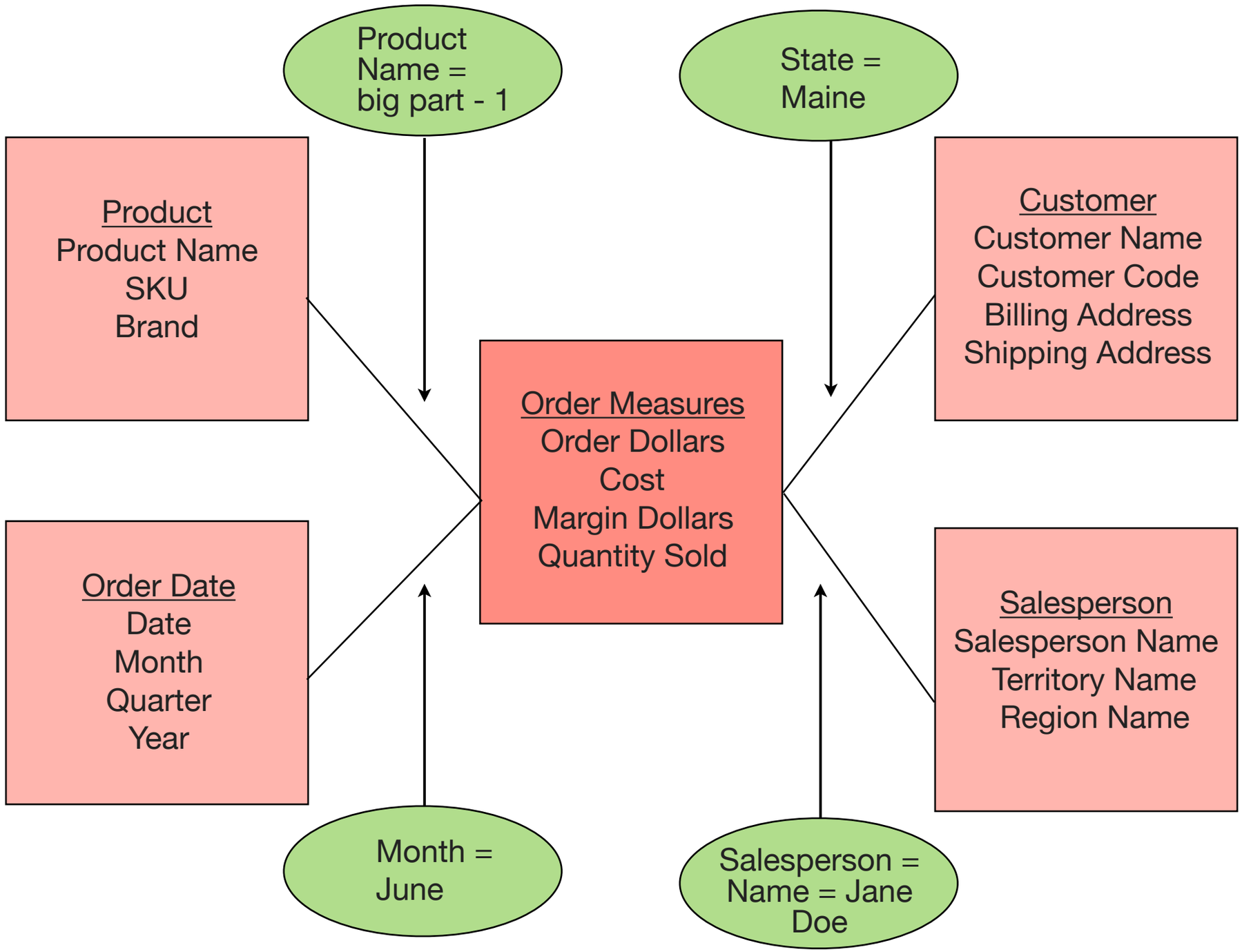


รูปที่ 7-4 ตัวอย่าง Star schema อย่างง่ายสำหรับการวิเคราะห์การสั่งซื้อสินค้า

จากส่วนประกอบต่างๆของ star schema ดังรูปที่ 7-4 ผู้ใช้สามารถวิเคราะห์ข้อมูลผ่านทางมาตรวัดความสำเร็จต่างๆ (จำนวนเงินในการสั่งซื้อสินค้า ค่าใช้จ่าย ผลกำไร และจำนวนชิ้นสินค้าที่ถูกสั่งซื้อ) โดยทำการแบ่งแยกความสนใจตามลักษณะของข้อมูลลูกค้า รายการสินค้า วันเวลาที่สั่งซื้อสินค้า และพนักงานขาย ที่จะทำให้ผู้ใช้สามารถทราบถึงการเกิดขึ้นของการสั่งซื้อสินค้าว่าอะไรถูกสั่งซื้อไป ซื้อไปเมื่อไหร่ โดยใครเป็นคนซื้อ และใครเป็นคนขาย เป็นต้น

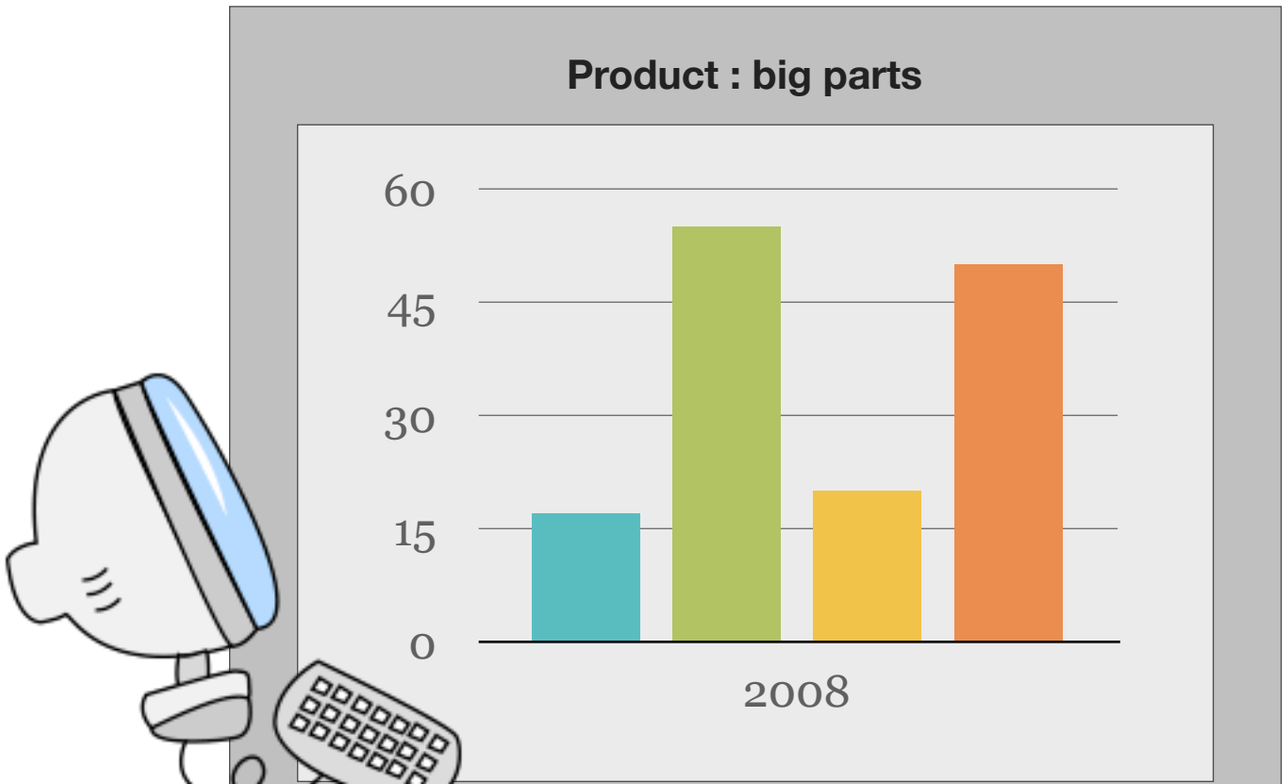
การที่จะทราบถึงข้อมูลต่าง ๆ ข้างต้น **ผู้ใช้จะต้องทำการกำหนด/สร้างคิวรีที่เกิดจากการรวมกันหรือเชื่อมต่อกันของ *dimension* ต่าง ๆ เข้ากับ *fact table* ซึ่งจะเป็นการค้นหาแถวของข้อมูลใน *fact table* ที่มีความสัมพันธ์กับแถวของข้อมูลต่าง ๆ ในแต่ละ *dimension table***

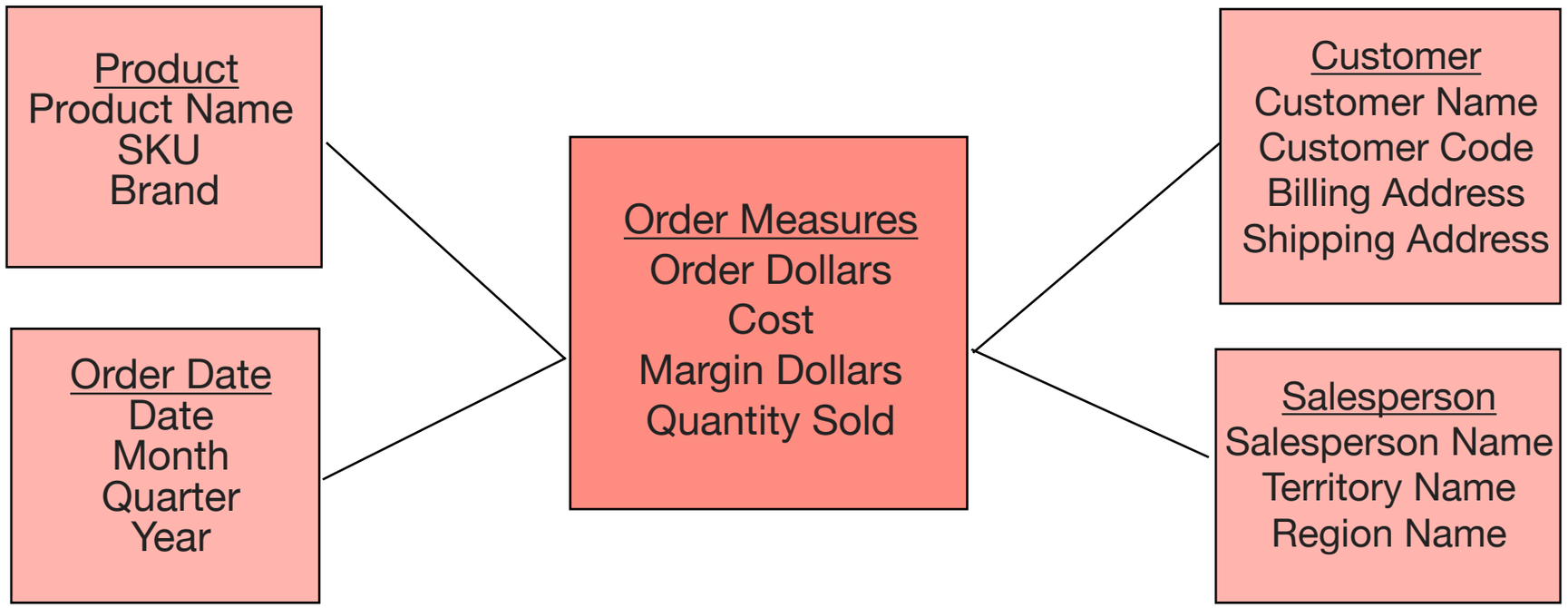
ลองพิจารณาตัวอย่างของคิวรีอย่างง่ายที่สามารถค้นหาข้อมูลใน star schema จากรูปที่ 7-4 ที่ซึ่งพนักงานฝ่ายการตลาดอาจต้องการจำนวนชิ้นสินค้าที่ถูกสั่งซื้อ และจำนวนเงินของการสั่งซื้อสินค้าชนิด **“bigpart-1”** ที่ถูกซื้อโดยลูกค้าที่อยู่ในรัฐ “Maine” ขายโดยนาย “Jane Doe” ในช่วงเดือนมิถุนายน จากคิวรีดังกล่าวผู้ใช้ได้ทำการกำหนดเงื่อนไขต่างๆ ที่เกี่ยวข้องกับแต่ละ dimension table (ดังแสดงในรูปที่ 7-5) และทำการเรียกดูข้อมูลที่เกี่ยวกับมาตรวัดจำนวนชิ้นสินค้าที่ถูกสั่งซื้อ และจำนวนเงินของการสั่งซื้อสินค้า

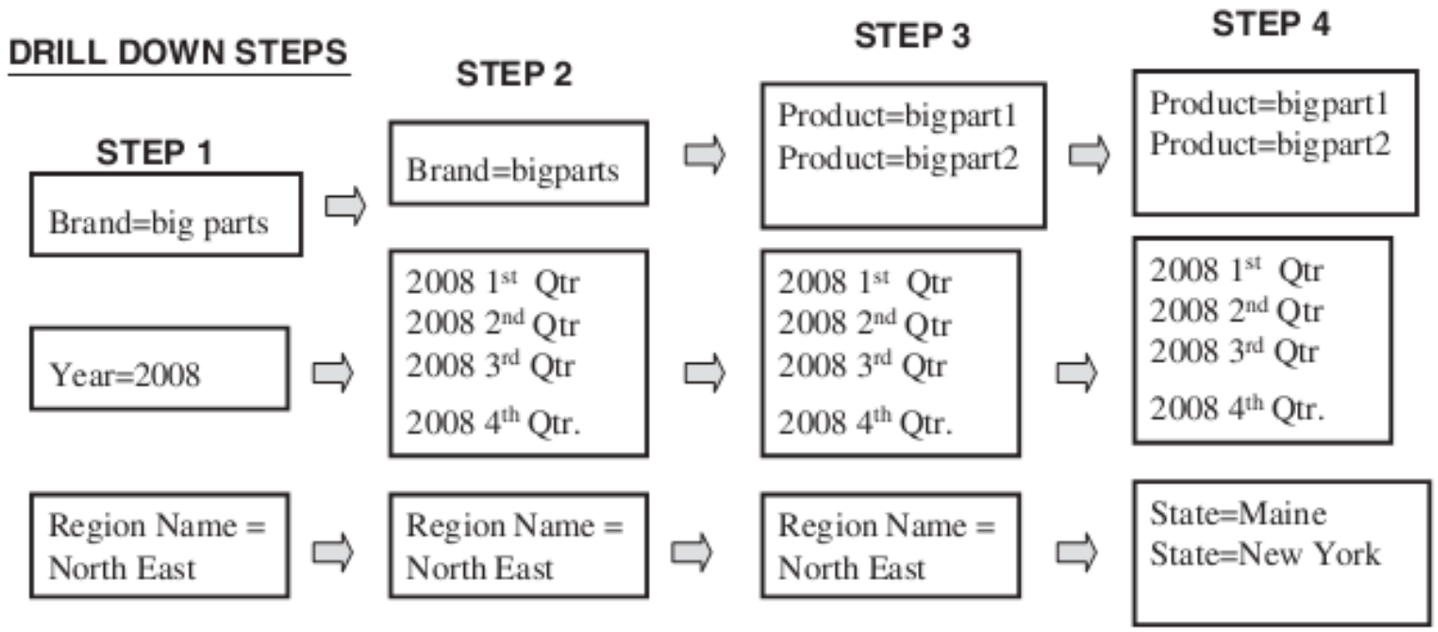


รูปที่ 7-5 ตัวอย่างการกำหนดคิวรีเพื่อเรียกดูข้อมูลจาก star schema

นอกจากการสร้างคิวรีแบบปกติแล้ว ผู้ใช้ยังสามารถเรียกดูข้อมูลแบบเจาะลึก (drilling down) ได้ด้วยการเรียกดูข้อมูลแบบเจาะลึกจะเป็นการแยกข้อมูลที่เป็นผลสรุปออกเป็นส่วนๆ โดยเริ่มแรกผู้ใช้อาจทำการเรียกดูข้อมูลที่เป็นผลสรุป จากนั้นจำการแยกส่วนของข้อมูลตามที่ผู้ใช้กำหนด ซึ่งจะได้ข้อมูลที่มีรายละเอียดมากขึ้นตามที่ผู้ใช้ต้องการ ลองพิจารณาตัวอย่างที่แสดงในรูปที่ 7-6 ที่แสดงถึงการเรียกดูข้อมูลแบบเจาะลึกจากผู้ใช้ โดยเริ่มแรกผู้ใช้อาจจะต้องการเรียกดูข้อมูลจำนวนสินค้ายี่ห้อ “big parts” ที่ขายได้ในปี 2008 ซึ่งเป็นยอดขายในภาคตะวันออกเฉียงเหนือ ต่อมาผู้ใช้จะต้องการข้อมูลที่เจาะลึกมากขึ้น โดยทำการแยกยอดขายในปี 2008 ออกเป็นแต่ละไตรมาส และยังสามารถแยกรายการสินค้าจากยี่ห้อสินค้าไปเป็นแต่ละรายการสินค้าของยี่ห้อ “big parts” ได้อีกด้วย และท้ายสุดผู้ใช้อาจเจาะลึกไปยังยอดขายในแต่ละเมืองทางแถบตะวันออกเฉียงเหนือ เป็นต้น จากความต้องการในเรียกดูข้อมูลดังกล่าว เราจะเห็นว่าข้อมูลที่ทำการเรียกดูจะมีความละเอียดของข้อมูลในแต่ละมิติที่มีความแตกต่างกัน ซึ่งการเรียกดูในลักษณะนี้เราจะเรียกว่า **การเรียกดู/วิเคราะห์ข้อมูลแบบเจาะลึก *(drill down analysis)***







รูปที่ 7- 6 ตัวอย่างการเรียกดูข้อมูลแบบเจาะลึกจาก star schema

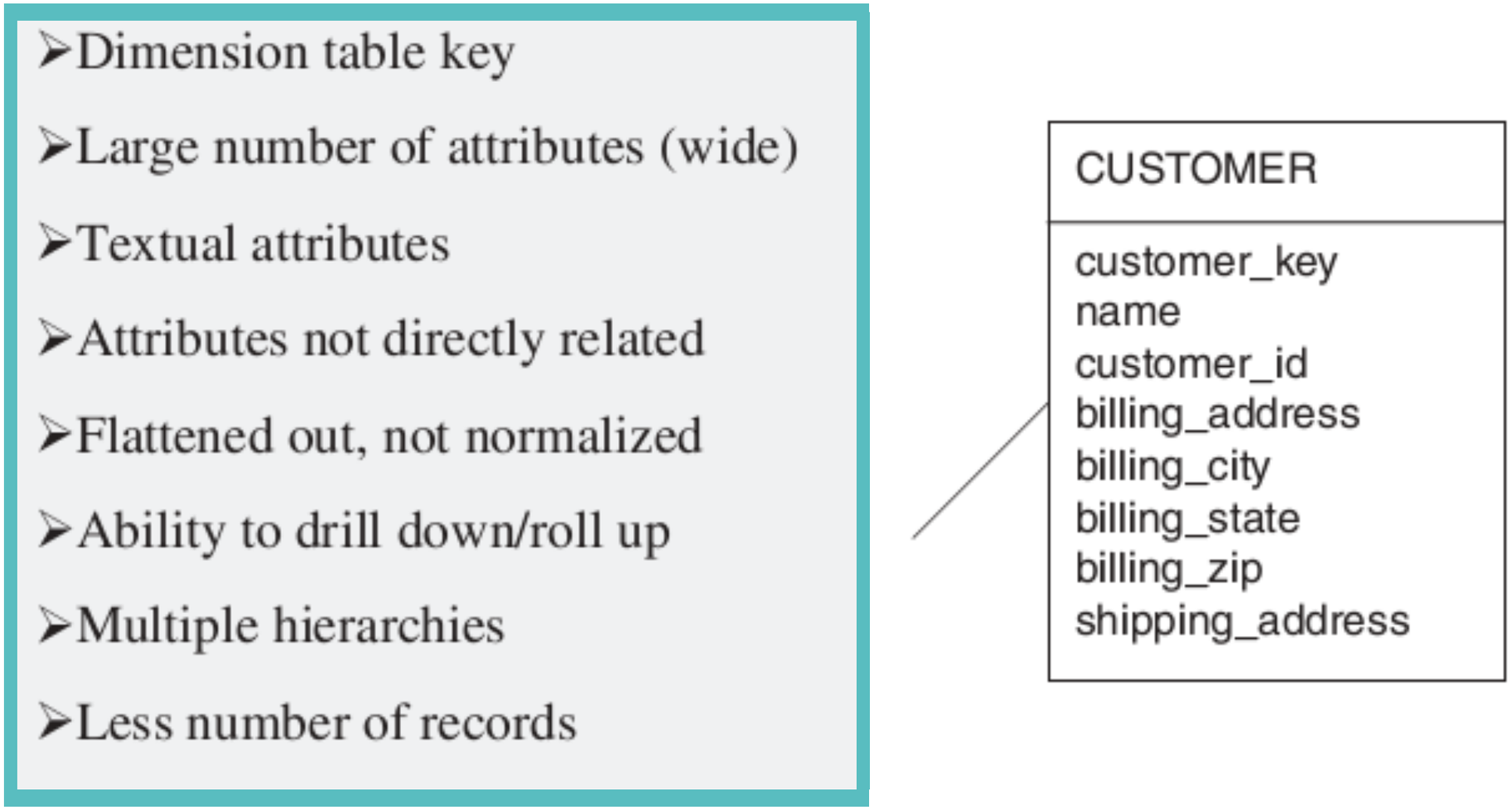
หลังจากทราบโครงสร้างพื้นฐานของ star schema ที่ประกอบไปด้วย fact table และ dimension table ต่างๆ ที่เกี่ยวข้องกับ fact table รวมถึงประเภทของคิวรีที่ใช้ในการเรียกดูข้อมูลจาก star schema แล้ว เพื่อให้เข้าใจในรายละเอียดของ star schema ลองพิจารณาถึงส่วนประกอบและรายละเอียดของ fact table และ แต่ละ dimension table ที่มีส่วนประกอบต่าง ๆ ดังนี้

Dimension table

star schema จะประกอบไปด้วย dimension table ต่าง ๆ ซึ่งแต่ละ dimension จะแสดงถึงมิติทางธุรกิจที่เกี่ยวเนื่องกับหัวข้อในการดำเนินธุรกิจที่สนใจ โดยแต่ละ dimension จะประกอบไปด้วยส่วนต่าง ๆ ดังแสดงในรูปที่ 7-7 ที่จะประกอบไปด้วย

(1) คีย์หลักที่เป็นตัวบ่งชี้ที่เป็นเอกลักษณ์ (uniquely identifiers) — ใช้สำหรับแยกความแตกต่างระหว่างข้อมูลแต่ละแถวใน dimension table

(2) คอลัมน์/แอทริบิวต่าง ๆ ซึ่งโดยส่วนใหญ่แล้วอยู่ในรูปแบบของตัวอักษรหรือตัวเลขที่ไม่ใช้สำหรับการคำนวณ และมีจำนวนไม่เกิน 50 คอลัมน์ด้วยกัน โดยที่แต่ละแอทริบิวอาจจะไม่ได้เกี่ยวเนื่องกับแอทริบิวอื่น ๆ โดยตรงก็ได้ เช่น ขนาดแพคเกจของสินค้า จะไม่ได้เกี่ยวข้องโดยตรงกับยี่ห้อสินค้าที่เก็บอยู่ใน dimension table



รูปที่ 7-7 ส่วนประกอบของ customer dimension table

แอทริบิวต่าง ๆ ใน dimension table จะมีคุณสมบัติหรือคุณลักษณะเด่น ๆ ดังนี้

- แอทริบิวหนึ่ง ๆ จะไม่ได้เกี่ยวเนื่องกับแอทริบิวอื่น ๆ โดยตรง เช่น ขนาดของแพคเกจสินค้าอาจจะไม่ได้เกี่ยวข้องกับยี่ห้อสินค้าโดยตรง แต่ทั้งสองแอทริบิวจะถูกเก็บอยู่ใน dimension table

- แอทริบิวใน dimension table จะไม่ถูกนอร์มอลไลซ์ (normalized) เนื่องจากในการค้นหาคำตอบให้กับคิวรีของผู้ใช้จะมีการเรียกใช้ข้อมูลจากแอทริบิวซ้ำ ๆ ดังนั้นเพื่อให้การค้นคืนผลลัพธ์มีประสิทธิภาพ เราควรจะสามารถเรียกใช้ข้อมูลจากแอทริบิวต่าง ๆ ของ dimension table ได้โดยตรง แล้วจึงนำผลลัพธ์ที่ได้ไปทำการค้นหาข้อมูลที่เกี่ยวข้องใน fact table ต่อไป ถ้าเราทำการนอร์มอลไลซ์ข้อมูลในแอทริบิวต่าง ๆ ของ dimension table จะเป็นการลดทอนประสิทธิภาพของการค้นคืนข้อมูลให้กับคิวรีต่าง ๆ ดังนั้นเราไม่ควรจะทำการนอร์มอลไลซ์ข้อมูลในแต่ละแอทริบิวของ dimension table เพื่อให้การค้นคืนผลลัพธ์ให้กับคิวรีจากผู้ใช้มีประสิทธิภาพ

แอทริบิวใน dimension table จะต้องถูกเก็บข้อมูลเป็นลำดับชั้น (hierarchy) เพื่อทำให้ผู้ใช้สามารถเรียกดูรายละเอียดของข้อมูลแบบเจาะลึก (drilling down) และแบบสรุปรายละเอียด (rolling up) ได้ ตัวอย่างเช่น

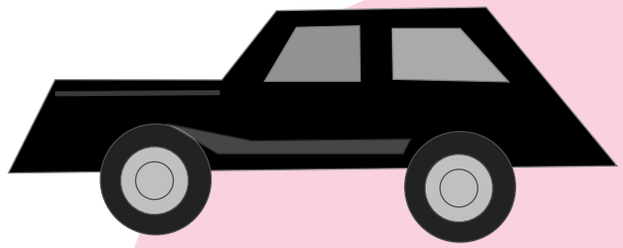
เราทำการจัดเก็บ 3 แอทริบิว คือ รหัสไปรษณีย์ ชื่อเมือง และ ชื่อรัฐ ไว้ใน dimension table โดยทั้ง 3 แอทริบิวมีความเกี่ยวเนื่องกันโดยอยู่ในลำดับชั้นเดียวกันซึ่งการเก็บข้อมูลในลักษณะนี้จะยอมให้ผู้ใช้สามารถเรียกดูข้อมูลยอดขาย (ข้อมูลจาก fact table ที่เกี่ยวเนื่องกับ dimension table) แบบเจาะลึก โดยการเรียกดูยอดขายในรัฐหนึ่งๆ จากนั้นทำการเจาะลึกไปเป็นยอดขายในเมืองหนึ่งๆ และยอดขายในเขตรหัสไปรษณีย์หนึ่งๆ เป็นต้น

สำหรับการเรียกดูข้อมูลแบบสรุปรายละเอียด โดยเริ่มจากการเรียกดูข้อมูลที่มีความละเอียดที่สุดคือ ยอดขายในเขตรหัสไปรษณีย์หนึ่ง ๆ แล้วทำการรวมยอดขายเป็นยอดขายในเมืองหนึ่ง ๆ ที่มีหลายรหัสไปรษณีย์ และ ยอดขายในรัฐหนึ่ง ๆ ที่มีหลายเมืองได้

ใน dimension table อาจมีแอทริบิวที่เป็นลำดับชั้นมากกว่าหนึ่งหรือหลายลำดับชั้นด้วยกัน เช่น ใน product dimension table อาจมี 2 ลำดับชั้น คือ

(1) ลำดับชั้นของรายการสินค้าที่ประกอบไปด้วยหมวดหมู่สินค้าในมุมมองของการตลาด (รายการสินค้า A หมวดหมู่ X เป็นต้น)
(2) ลำดับชั้นของรายการสินค้าที่ประกอบไปด้วยหมวดหมู่สินค้าในมุมมองของการเงิน (รายการสินค้า A หมวดหมู่สินค้าสร้างรายได้ หมวดหมู่สินค้าขายดี เป็นต้น)

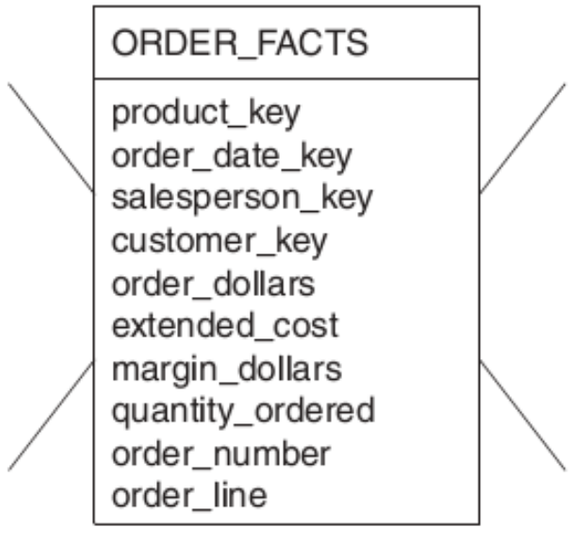
จำนวนเรคคอร์ดหรือแถวของข้อมูลใน dimension table จะมีจำนวนน้อยกว่าใน fact table ค่อนข้างมาก ตัวอย่าง เช่น product dimension table ของบริษัทผู้ผลิตรถยนต์อาจมีข้อมูลรายการรถยนต์ที่ทำการผลิตเพียง 500 เรคคอร์ด แต่ในขณะที่ fact table อาจมียอดขายรถยนต์เป็นจำนวนหลายล้านเรคคอร์ด เป็นต้น



หลังจากทราบรายละเอียดของ dimension table ที่อยู่ใน star schema หนึ่ง ๆ แล้ว ลองพิจารณารายละเอียดและคุณลักษณะของ fact table ที่จะประกอบไปด้วยมาตรวัดความสำเร็จต่างๆของการดำเนินธุรกิจในหัวข้อที่เราสนใจ และส่วนประกอบอื่น ๆ ดังรูปที่ 7-8 จากรูปเราสามารถแจกแจงส่วนประกอบและคุณสมบัติของ fact table ได้ดังนี้

Fact table

- Concatenated fact table key
- Grain or level of data identified
- Fully additive measures
- Semi-additive measures
- Large number of records
- Only a few attributes
- Sparsity of data
- Degenerate dimensions



รูปที่ 7-8 ส่วนประกอบของ fact table ที่เกี่ยวข้องกับการสั่งซื้อสินค้า

คีย์หลักของข้อมูลเรคคอร์ดหนึ่ง ๆ ใน fact table จะมาจาก**การเรียงต่อกันของคีย์หลักของ *dimension table*** — เนื่องจากข้อมูลในแต่ละเรคคอร์ดของ fact table จะเกิดจากการรวมกันของทุก dimension table ซึ่งจากรูปที่ 7-8 จะประกอบไปด้วย 4 dimension ด้วยกัน คือ **รายการสินค้า วันที่สั่งสินค้า ลูกค้า และพนักงานขาย** ถ้า dimension เหล่านี้มีการเก็บข้อมูลที่มีความละเอียดสูงสุด นั่นคือ ข้อมูลแต่ละรายการสินค้า ข้อมูลแต่ละวัน ข้อมูลแต่ละรายชื่อลูกค้า ข้อมูลแต่ละพนักงานขายสินค้า ตามลำดับ

จากความละเอียดของข้อมูลดังกล่าว ข้อมูลแถวหนึ่งๆ ใน *fact table* จะเกี่ยวข้องกับรายการสินค้าหนึ่งที่ถูกซื้อในวันหนึ่งโดยลูกค้าคนหนึ่งและขายโดยพนักงานขายคนหนึ่ง ซึ่งเราสามารถระบุถึงข้อมูลเรคคอร์ดนั้นได้ โดยนำคีย์หลักของทุก ๆ dimension table

ข้อมูลมาตรวัด (measurement/metrics) ใน fact table นั้นจะมีความละเอียดที่แตกต่างกัน ซึ่งจะแสดงถึงข้อมูลที่แตกต่างกันออกไปด้วย ตัวอย่างเช่น ถ้าข้อมูลมาตรวัดจำนวนสินค้าที่ขายได้มีความละเอียดสูง จะทำให้เราสามารถทราบถึงจำนวนชิ้นสินค้าชนิดหนึ่ง ๆ ที่ขายได้ในวันหนึ่ง ๆ โดยลูกค้าคนหนึ่ง ๆ และพนักงานขายคนหนึ่ง ๆ แต่ถ้าเราเปลี่ยนความละเอียดของข้อมูลให้น้อยลง เช่น เราทำการเก็บข้อมูลยอดขายของรายการสินค้าหนึ่งที่ขายได้ในเดือนหนึ่ง ๆ จะทำให้เราได้ข้อมูลที่เป็นผลสรุปมากขึ้น เมื่อเราทำการเก็บข้อมูลที่มีรายละเอียดสูงจะทำให้ผู้ใช้สามารถเรียกดูข้อมูลแบบเจาะลึก (drill down) และแบบสรุปรายละเอียด (roll up) ได้อย่างมีประสิทธิภาพ

ข้อดีอีกข้อหนึ่งของการเก็บข้อมูลที่มีความละเอียดสูงคือ ความคงทนต่อการเปลี่ยนแปลงของข้อมูลในรูปแบบต่างๆ เช่น ถ้าเราต้องการเพิ่มแอทริบิวใหม่ที่เกี่ยวกับเขตที่พนักงานขายสังกัดอยู่เข้าไปใน sale representative dimension การเพิ่มนี้ไม่ก่อให้เกิดการเปลี่ยนแปลงต่อ fact table เนื่องจากข้อมูลแต่ละเรคคอร์ดใน fact table สามารถสื่อถึงพนักงานขายแต่ละรายได้อยู่แล้ว

ดังนั้นในการกำหนดคิวรีโดยผู้ใช้จะไม่มีการเปลี่ยนแปลงตามไปด้วย อีกกรณีหนึ่งคือ ถ้าเราต้องการที่จะเพิ่มมิติทางธุรกิจใหม่เข้าไปใน star schema คือ promotion dimension table จะทำให้เราจะต้องทำการปรับแต่งเรคคอร์ดใน fact table ใหม่เพื่อให้แต่ละเรคคอร์นั้นรวมข้อมูลเกี่ยวกับโปรโมชั่นเข้าไปด้วย ซึ่งการแก้ไขเรคคอร์ดต่าง ๆ ใน fact table จะไม่ทำให้ความละเอียดของข้อมูลเปลี่ยนไปแต่อย่างใด ความละเอียดของข้อมูลใน fact table ยังคงเป็นรายละเอียดสูงสุดเหมือนเดิม และการกำหนดคิวรียังคงเดิมถ้าเราไม่ได้สนใจโปรโมชั่นในการสืบค้นข้อมูล

นอกจากนี้เรายังสามารถนำเอาข้อมูลที่มีรายละเอียดสูงไปเป็นอินพุตของโมเดลทางด้านการทำเหมืองข้อมูล (data mining) ที่ต้องการข้อมูลที่มีรายละเอียดสูงได้อีกด้วย

จากที่กล่าวมาทั้งหมดข้างต้นจะเป็นข้อดีของการที่ fact table มีข้อมูลที่มีรายละเอียดสูง แต่อย่างไรก็ดีการที่มีรายละเอียดสูงก็แลกมาด้วยการสิ้นเปลืองเนื้อที่สำหรับจัดเก็บข้อมูลและความยุ่งยากในการดูแลรักษาข้อมูลใน *fact table* ที่จะเพิ่มขึ้น การที่ fact table มีรายละเอียดสูงจะทำให้ fact table มีจำนวนเรคคอร์ดเป็นจำนวนมาก และเมื่อผู้ใช้ต้องการข้อมูลที่เป็นผลสรุปจะทำให้การสืบค้นข้อมูลอาจใช้เวลานาน ดังนั้นเราอาจจำเป็นที่จะต้องสร้างตาราง fact table ที่เป็นผลสรุปของข้อมูลขึ้นมาใหม่เพื่อทำให้ผู้ใช้สามารถค้นหาข้อมูลที่เป็นผลสรุปได้รวดเร็วมากขึ้น แต่ก็จะทำให้สิ้นเปลืองพื้นที่สำหรับจัดเก็บข้อมูลมากขึ้นไปอีก

แอทริบิว/มาตรวัดบางตัวจะมีคุณสมบัติเป็น fully additive measures ซึ่งเป็นมาตรวัดที่สามารถทำการรวบยอดข้อมูลได้ ตัวอย่างเช่น

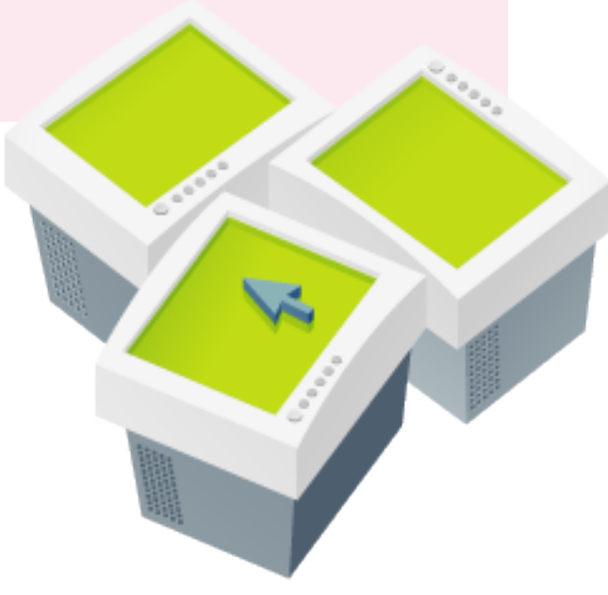
มาตรวัดยอดเงินการสั่งซื้อสินค้า ค่าใช้จ่าย และจำนวนชิ้นสินค้าที่ถูกสั่งซื้อ จะเกี่ยวข้องกับ รายการสินค้าหนึ่ง ๆ ที่ถูกซื้อในวันหนึ่ง ๆ โดยลูกค้าคนหนึ่ง โดยซื้อจากผู้ขายคนหนึ่ง แต่ในความเป็นจริงแล้วผู้ใช้จะต้องการข้อมูลสรุปรวบยอดของยอดเงินการสั่งซื้อสินค้า ค่าใช้จ่าย และจำนวนชิ้นสินค้าที่ถูกสั่งซื้อของรายการสินค้าหนึ่งที่ถูกขายในวันหนึ่ง ๆ ถูกซื้อโดยลูกค้าที่อาศัยอยู่ในเขตหนึ่ง ๆ เป็นต้น

เพื่อให้ได้ข้อมูลที่ผู้ใช้ต้องการ เราต้องทำการค้นหาเรคคอร์ดข้อมูลใน fact table ที่เกี่ยวข้องกับลูกค้าที่อาศัยอยู่ในเขตที่เราต้องการ แล้วทำการรวมยอดเงินการสั่งซื้อสินค้า ค่าใช้จ่าย และจำนวนชิ้นสินค้าที่ถูกสั่งซื้อ จากการที่เราสามารถทำการสรุปรวบยอดมาตรวัดเหล่านี้ได้โดยใช้ฟังก์ชันการบวกธรรมดา ๆ เราจะเรียกมาตรวัดเหล่านี้ว่าเป็น fully additive measure

ดังนั้นในการค้นหาข้อมูลจากคิวรีที่ต้องการข้อมูลที่เป็นผลสรุป เราจะต้องแน่ใจว่ามาตรวัดที่เรากำลังพิจารณานั้นเป็นแบบ fully additive ไม่เช่นนั้นแล้วเราอาจจะได้ผลลัพธ์ที่ไม่ถูกต้อง

แอทริบิว/มาตรวัดบางตัวจะมีคุณสมบัติเป็น semiadditive measures กล่าวคือ เป็นมาตรวัดที่สามารถคำนวณได้ แต่ไม่สามารถคำนวณสรุปรวบยอดข้อมูลจากหลาย ๆ เรคคอร์ดของ fact table ได้โดยตรง ตัวอย่างเช่น

มาตรวัดผลกำไรของการสั่งซื้อสินค้าที่สามารถคำนวณได้จากยอดเงินการสั่งซื้อสินค้า และค่าใช้จ่าย ถ้าในการสั่งสินค้ารายการหนึ่งในวันหนึ่งโดยลูกค้าคนหนึ่งมียอดเงินการสั่งซื้อสินค้าเป็น 120 และค่าใช้จ่ายเท่ากับ 100 เราจะสามารถคำนวณเปอร์เซ็นต์ของผลกำไรได้เท่ากับ 20 แต่ถ้าต้องการเปอร์เซ็นต์ของกำไรที่ได้จากการขายสินค้ารายการหนึ่งในวันหนึ่งให้กับลูกค้าที่อาศัยอยู่ในเขตหนึ่ง ๆ เราไม่สามารถสรุปรวบยอดข้อมูลได้โดยตรงจากเรคคอร์ดต่าง ๆ ใน fact table แต่เราจะสามารถคำนวณได้จากผลรวมของยอดเงินการสั่งซื้อสินค้าและผลรวมของค่าใช้จ่ายที่ได้มาจากการบวกกันของผลลัพธ์แถวต่าง ๆ แล้วค่อยทำการหาค่าผลกำไรเป็นร้อยละ ซึ่งจากการที่ไม่สามารถรวมผลข้อมูลได้โดยตรง มาตรวัดเปอร์เซ็นต์ผลกำไรจะไม่เป็น fully additive measure แต่จะเป็นแบบ semiadditive measure ซึ่งความแตกต่างของมาตรวัดทั้ง 2 ชนิดจะเกิดขึ้นเมื่อเราทำการรวบยอดข้อมูลเพื่อคืนค่าผลลัพธ์ให้กับคิวรีจากผู้ใช้

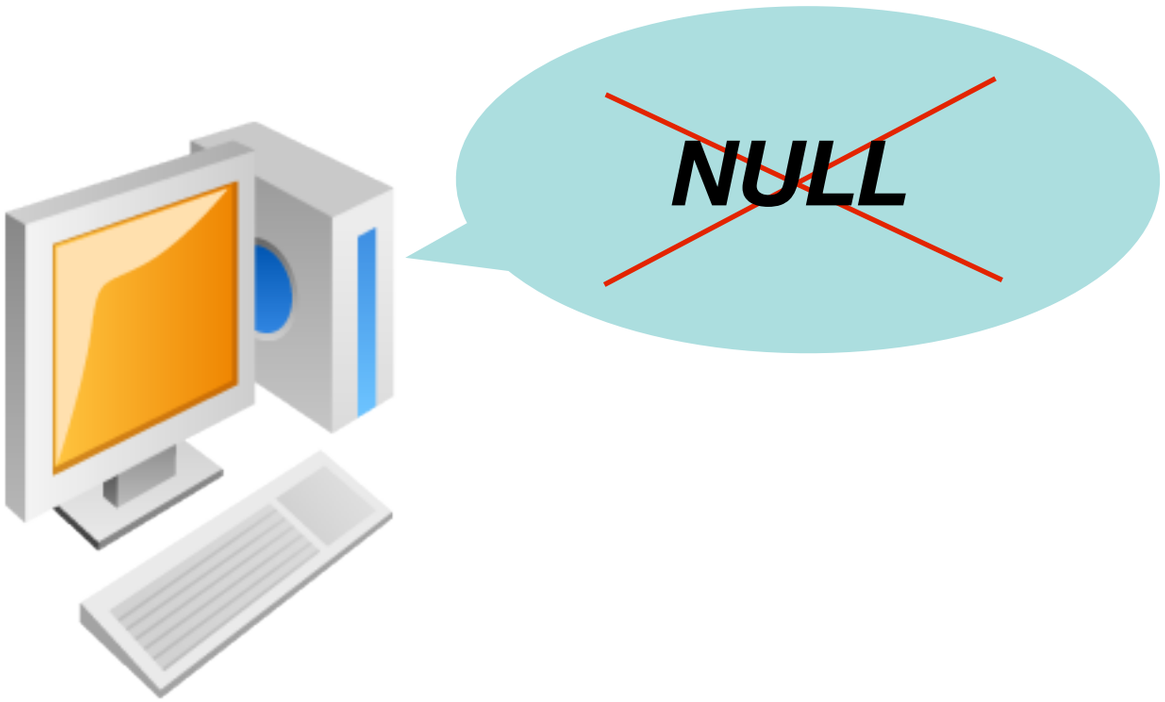


โดยปกติ fact table จะมีจำนวนแอทริบิวประมาณ 10 แอทริบิวหรือน้อยกว่านั้น ซึ่งเป็นจำนวนที่น้อยกว่า dimension table แต่จำนวนเรคคอร์ดใน fact table จะมีมากกว่าใน dimension table เป็นจำนวนหลายเท่าด้วยกัน สมมติว่าใน star schema ประกอบไปด้วย 4 dimension table ดังนี้

1. Product dimension table ที่มีข้อมูลรายการสินค้า 3 รายการ
2. Customer dimension table ที่มีข้อมูลลูกค้าจำนวน 5 ราย
3. Order date dimension table ที่มีข้อมูลวันที่มีการสั่งสินค้าเป็นจำนวน 30 วัน
4. Sale representative dimension table ที่มีข้อมูลพนักงานขายจำนวน 10 คน จากข้อมูลทั้งหมด เราสามารถคำนวณจำนวนเรคคอร์ด/แถวของข้อมูลใน fact tableได้เท่ากับ 4,500 เรคคอร์ด (=3×5×30×10) ซึ่งเป็นเรคคอร์ดจำนวนมากเมื่อเทียบกับ dimension table สมมติว่าถ้าใน star schema ประกอบไปด้วย 2 dimension table จะทำให้ fact table นั้นค่อนข้างแคบ เนื่องจากจะมีจำนวนแอทริบิวเพียง 2 แอทริบิวเท่านั้น แต่อาจจะยังคงมีเรคคอร์ดของข้อมูลเป็นจำนวนมาก

ใน fact table อาจจะมีข้อมูลแบบเบาบาง (sparse data) ถ้าใน star schema ประกอบไปด้วย 4 dimension table นั่นคือ **product, order date, customer และ sale representative** ที่เก็บข้อมูลที่มีความละเอียดสูงสุด จะทำให้ข้อมูลแถว/เรคคอร์ดหนึ่ง ๆ ใน fact table จะเกี่ยวข้องกับ รายการสินค้าชนิดหนึ่ง ที่ถูกซื้อในวันหนึ่ง โดยลูกค้าคนหนึ่ง และขายโดยพนักงานคนหนึ่ง ถ้าบริษัทมีวันหยุดทำการ เราก็จะไม่มีข้อมูลการสั่งซื้อสินค้าในวันนั้น ๆ ซึ่งเราจะไม่ทำการเก็บข้อมูลอยู่ใน fact table ด้วย

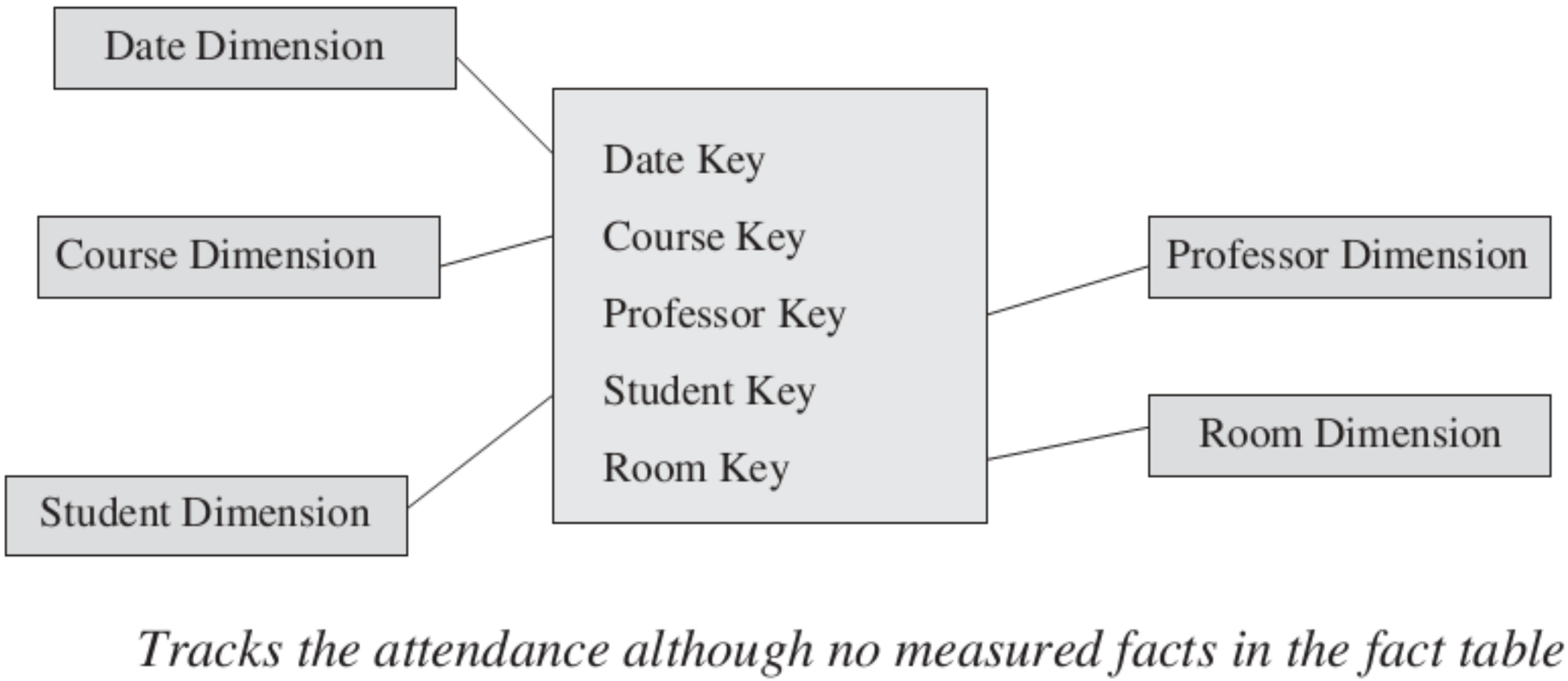
ดังนั้นเราสามารถสรุปได้ว่าเรา**จะทำการเก็บข้อมูลที่เกิดขึ้นไว้ใน *fact table* เท่านั้น สำหรับข้อมูลที่ไม่เกิดขึ้น เราจะไม่ทำการเก็บข้อมูลไว้เป็น *NULL* หรือเป็นค่าอื่น ๆ โดยเด็ดขาด** ซึ่งการเก็บข้อมูลที่เกิดขึ้นจริงเท่านั้น อาจจะทำให้ข้อมูลใน fact table มีความเบาบางเนื่องจากการขาดหายไปหรือการไม่เกิดขึ้นของข้อมูลในบางช่วงเวลา



ใน fact table อาจจะมีข้อมูลที่ไม่ใช่มาตรวัด (metrics) ต่าง ๆ อยู่ด้วย จากตัวอย่างในรูปที่ 7-8 จะมีข้อมูล order_number และ order_line เป็นแอทริบิวใน fact table ที่ไม่ใช่มาตรวัด แต่เป็นข้อมูลจำนวนสินค้าที่ถูกสั่งซื้อ จำนวนครั้งที่สั่งซื้อสินค้า เป็นต้น ถึงแม้ว่าข้อมูลเหล่านี้จะไม่ใช่มาตรวัด แต่ในบางกรณีข้อมูลเหล่านี้จะมีประโยชน์ในการวิเคราะห์ต่าง ๆ เช่น เราอาจจะต้องการหาจำนวนเฉลี่ยของสินค้าต่อการสั่งซื้อสินค้าในแต่ละครั้ง การที่จะได้มาซึ่งค่าเฉลี่ยเราจำเป็นต้องใช้ข้อมูลเหล่านี้ในการคำนวณ ดังนั้นเราจึงจำเป็นต้องเก็บข้อมูลเหล่านี้ไว้ใน fact table ซึ่งการเก็บข้อมูลในลักษณะนี้จะเรียกว่า “***degenerate dimensions***”

นอกจากลักษณะและคุณสมบัติของข้อมูลที่ถูกเก็บอยู่ใน fact table ที่อธิบายก่อนหน้าแล้ว เรายังต้องพิจารณาว่า fact table มีมาตรวัดที่มีประโยชน์หรือไม่ ลองพิจารณาตัวอย่างในรูปที่ 7-9 ซึ่งแสดง fact table ที่ไม่มีมาตรวัด (metrics) หรือค่าความจริง (facts) ใดๆ ซึ่งจากรูปจะเป็น star schema ที่ต้องการจะตรวจสอบการเข้าชั้นเรียนของนักเรียน ที่ประกอบไปด้วย dimension ต่างๆ อาทิเช่น วันที่ วิชาที่เรียน ห้องเรียน นักเรียน และอาจารย์ ถ้านักเรียนคนหนึ่ง ๆ เข้าเรียนวิชาหนึ่ง ณ ห้องเรียนหนึ่ง ที่สอนโดย อ. หนึ่งๆ และเหตุการณ์เกินขึ้นในวันหนึ่งๆ เราจะเก็บข้อมูลการเข้าเรียนของนักเรียนด้วยค่าเลข 1 ดังนั้นทุกครั้งที่มีการเข้าเรียนของนักเรียนคนใดก็ตาม fact table จะต้องเก็บเลข 1 ไว้สำหรับข้อมูลแถว/เรคคอร์ดนั้น ๆ เสมอ แต่แท้จริงแล้วเราไม่ต้องทำการเก็บเลข 1 เพื่อแสดงถึงการเข้าเรียนของนักเรียน เนื่องจากถ้ามีข้อมูลเกิดขึ้นใน fact table นั้นก็หมายถึงมีนักเรียนเข้าเรียน ดังนั้นเราจึงไม่ต้องทำการเก็บค่าเลข 1 ที่เป็นมาตรวัดหรือค่าความจริงแต่อย่างใด ดังนั้นเมื่อ fact table ไม่มีการเก็บค่าความจริงหรือมาตรวัดใด ๆ เราจะเรียก fact table นั้นว่า “***Factless fact table***”

Measures or facts are represented in a fact table. However, there are business events or coverage that could be represented in a fact table, although no measures or facts are associated with these.



รูปที่ 7-9 ตัวอย่าง factless fact table ที่ไม่มีมาตรวัดหรือค่าความจริงใดๆ

แบบจำลองมิติต่าง ๆ จะประกอบไปด้วย 2 ส่วนหลัก ๆ คือ **fact table** และ **dimension table** ซึ่งอย่างที่เราทราบดีว่า dimension table จะประกอบข้อมูลต่าง ๆ ที่มีลักษณะเป็นแบบลำดับชั้น (hierarchy) และแบบแบ่งหมวดหมู่ (category) ในส่วนของ fact table จะประกอบไปด้วยคีย์ต่าง ๆ จาก dimension table ที่เกี่ยวเนื่องกับ fact table และมาตรวัดผลสัมฤทธิ์ที่เราสนใจ ซึ่งจากองค์ประกอบทั้งสองส่วนแต่ละส่วนจะต้องมีคีย์ไว้ใช้สำหรับแยกความแตกต่างระหว่างข้อมูลที่มีลักษณะแตกต่างกัน ซึ่งใน star schema จะประกอบไปด้วยคีย์ 3 ประเภทด้วยกันดังนี้

คีย์หลัก (Primary keys)

คีย์ตัวแทน (Surrogate keys)

คีย์รอง (Foreign keys)

คีย์หลัก
(Primary keys)

คีย์หลักจะเป็นข้อมูลแอทริบิวหนึ่งที่ใช้สำหรับระบุความแตกต่างของข้อมูลแต่ละแถวใน dimension table เช่น คีย์หลักของ product dimension table จะสามารถระบุหนึ่งรายการสินค้าหนึ่ง ๆ ได้อย่างชัดเจน แต่อย่างไรก็ดี เราควรจะต้องระมัดระวังในการกำหนดคีย์หลักให้กับแต่ละแถวของข้อมูลของแต่ละ dimension ด้วย ลองพิจารณาตัวอย่างการกำหนดคีย์หลักที่ก่อให้เกิดปัญหาได้ ดังนี้—ถ้าเราเก็บคีย์หลักของ product dimension table เป็นรหัสสินค้าที่ประกอบไปด้วยตัวเลข 8 หลักด้วยกัน แต่ละหลักอาจจะสื่อความหมายต่าง ๆ เช่น อาจมีเลข 2 หลักที่แสดงถึงรหัสหรือหมายเลขคลังสินค้าที่รายการสินค้านั้น ๆ ถูกเก็บอยู่ อาจมีเลขอีก 2 หลักที่แสดงถึงประเภทของรายการสินค้า เป็นต้น แต่ถ้าเราทำการเคลื่อนย้ายสินค้าไปจัดเก็บไว้ในอีกคลังสินค้าหนึ่งจะทำให้รหัสสินค้าที่มีความเกี่ยวเนื่องกับรหัสคลังสินค้าต้องเปลี่ยนแปลงตามรหัสคลังสินค้าไปด้วย ซึ่งถ้ารหัสสินค้านั้นเป็นส่วนหนึ่งของคีย์หลักของ product dimension table จะทำให้เราต้องทำการเก็บข้อมูลรายการสินค้าชนิดเดียวกันซ้ำ 2 ครั้ง เนื่องจากคีย์หลักที่ไม่เหมือนกัน (เนื่องจากบริษัททำการเปลี่ยนคลังสินค้าที่ใช้เก็บสินค้า ทำให้คีย์หลักเปลี่ยนไป) ซึ่งการเก็บข้อมูลซ้ำซ้อนกันอาจทำให้เกิดปัญหาในกรณีที่เราต้องทำการรวบรวม/รวบยอดข้อมูลที่อาจไม่สอดคล้องกัน ดังนั้นจากเหตุการณ์ดังกล่าว เราไม่ควรจะใช้คีย์หลักของข้อมูลที่ได้จากระบบการดำเนินงานมาใช้เป็นคีย์หลักของ dimension table เนื่องจากข้อมูลในระบบการดำเนินงานนั้นมีการเปลี่ยนแปลงค่อนข้างบ่อย เราจึงควรที่จะสร้างคีย์หลักขึ้นมาใหม่ที่ไม่สอดคล้องกับคีย์หลักของระบบการดำเนินงาน

คีย์ตัวแทน
(Surrogate keys)

เพื่อที่จะแก้ปัญหาข้างต้น เราอาจใช้คีย์ตัวแทน *(surrogate keys)* เป็นคีย์หลักของแต่ละ dimension table แทน คีย์ตัวแทนจะเป็นคีย์ที่สร้างมาจากลำดับของตัวเลขที่ไม่มีความหมายใด ๆ แอบแฝง (อาทิเช่นไม่มีหมายเลขคลังสินค้าที่จัดเก็บสินค้านั้น ๆ เป็นส่วนประกอบของคีย์หลัก เป็นต้น)

ดังนั้นคีย์ตัวแทนสามารถอ้างถึงรายการสินค้าได้โดยง่าย แต่ในบางกรณีที่เราต้องการความหมายแอบแฝงของคีย์หลักจากระบบการดำเนินงาน เช่น เราต้องการทราบถึงหมายเลขคลังสินค้าที่จัดเก็บรายการสินค้านั้น ๆ เราสามารถเก็บข้อมูลหมายเลขคลังสินค้าแยกไว้เป็นอีกแอทริบิวหนึ่งของข้อมูลใน dimension table ได้

คีย์รอง
(Foreign keys)

แต่ละ dimension table จะเกี่ยวข้องกับ fact table โดยมีความสัมพันธ์เป็นแบบ one-to-many โดยที่คีย์หลักของแต่ละ dimension table จะเป็นคีย์รองใน fact table ถ้า fact table มีความสัมพันธ์กับ 4 dimension table เช่น รายการสินค้า วันและเวลา ลูกค้า และตัวแทนขาย แต่ละคีย์หลักของทั้ง 4 dimension table จะถูกเก็บเป็นคีย์รองอยู่ใน fact table ทั้งหมด หลังจากเก็บคีย์รองทั้งหมดไว้ใน fact table แล้ว ลองพิจารณาทางเลือกในการเก็บในการกำหนดคีย์หลักให้กับ fact table ซึ่งสามารถทำได้ดังนี้

1 นำคีย์หลักของแต่ละ dimension table มาเรียงต่อกันเพื่อสร้างเป็นคีย์หลักของ fact table แต่ยังคงเก็บคีย์หลักของแต่ละ dimension table ไว้เป็นคีย์รองของ fact table ด้วย วิธีการนี้จะทำให้เปลืองเนื้อที่ในการจัดเก็บข้อมูลแต่ละเรคคอร์ดใน fact table

2 นำคีย์หลักของแต่ละ dimension table มาเรียงต่อกันเพื่อสร้างเป็นคีย์หลักของ fact table แต่ไม่ทำการเก็บคีย์หลักของแต่ละ dimension table ไว้ใน fact table เลย เนื่องจากเราสามารถมองได้ว่าแต่ละส่วนของคีย์หลักของ fact table สามารถสื่อถึงคีย์หลักของแต่ละ dimension table

3 สร้างคีย์หลักขึ้นใหม่ที่ไม่ได้ขึ้นกับคีย์หลักของ dimension table วิธีการนี้ยังคงเก็บเก็บคีย์หลักของแต่ละ dimension table ไว้เป็นคีย์รองของ fact table ด้วย วิธีการนี้จะทำให้เปลืองเนื้อที่ในการจัดเก็บข้อมูลแต่ละเรคคอร์ดใน fact table

จากทั้ง 3 วิธีที่ได้กล่าวมาข้างต้น วิธีที่ 2 เป็นวิธีที่ได้รับความนิยมเป็นอย่างมาก เนื่องจากทางเลือกนี้ประหยัดพื้นที่ในการจัดเก็บข้อมูลและยังสามารถเชื่อมโยงถึงคีย์หลักของแต่ละ dimension table ได้อีกด้วย

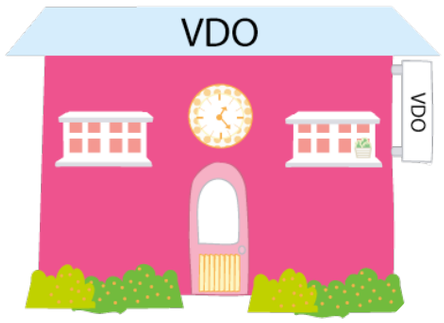
ตัวอย่าง star schema สำหรับธุรกิจต่างๆ

จากส่วนก่อนหน้าเราทราบว่า star schema จะประกอบไปด้วย

(1) มาตราวัดหรือข้อเท็จจริง (facts)

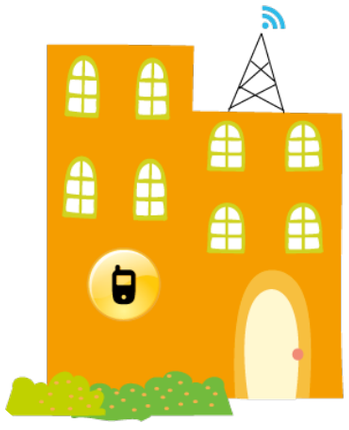
(2) มิติทางธุรกิจต่าง ๆ ที่แต่ละมิติจะประกอบไปด้วยข้อมูลที่เป็นลำดับชั้น (hierarchy) และข้อมูลที่เป็นหมวดหมู่ (category) ที่ใช้สำหรับการวิเคราะห์

(3) คีย์หลัก (primary key) และ คีย์รอง (foreign keys) ซึ่งจากส่วนประกอบต่าง ๆ ของ star schema เราอาจยังไม่เห็นภาพโดยละเอียดหรืออาจยังไม่เข้าใจรายละเอียดข้างใน star schema มากนัก ดังนั้น เพื่อให้เรามีความเข้าใจเกี่ยวกับ star schema มากขึ้นลองพิจารณาตัวอย่าง star schema ของธุรกิจที่แตกต่างกัน 4 ประเภทดังต่อไปนี้



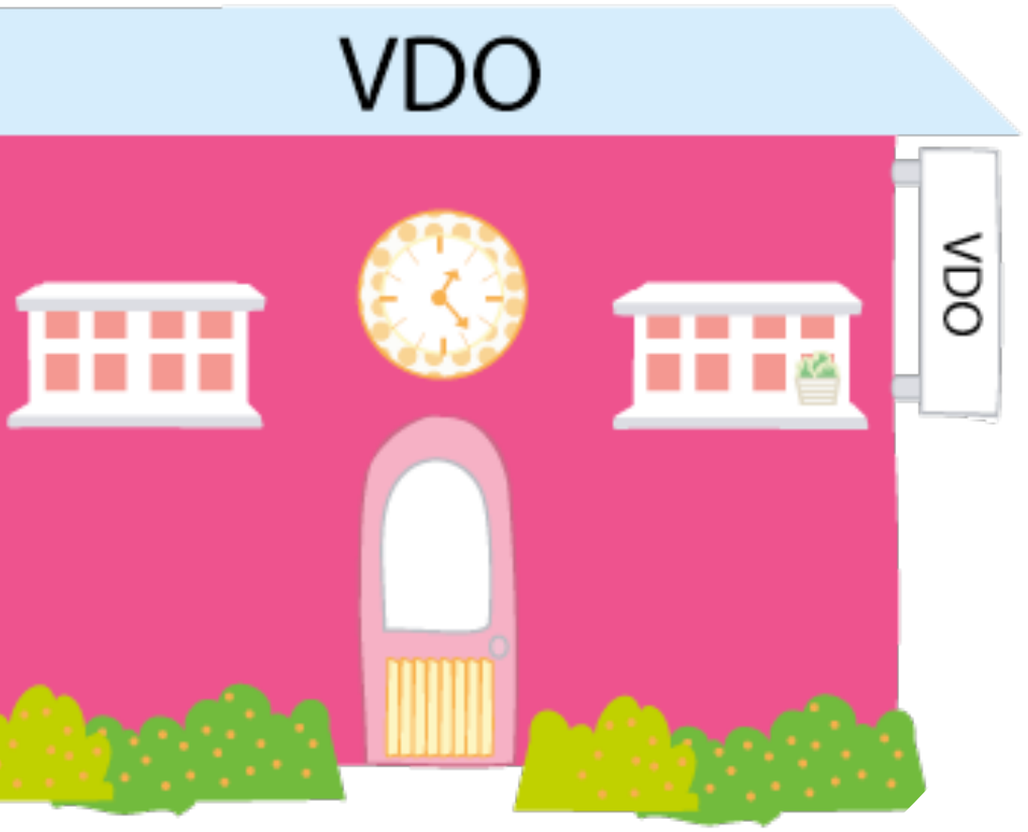
บริษัทให้เช่าวีดีโอ

ธุรกิจซุปเปอร์มาเก็ต



บริษัทผู้ให้บริการระบบโทรศัพท์เคลื่อนที่

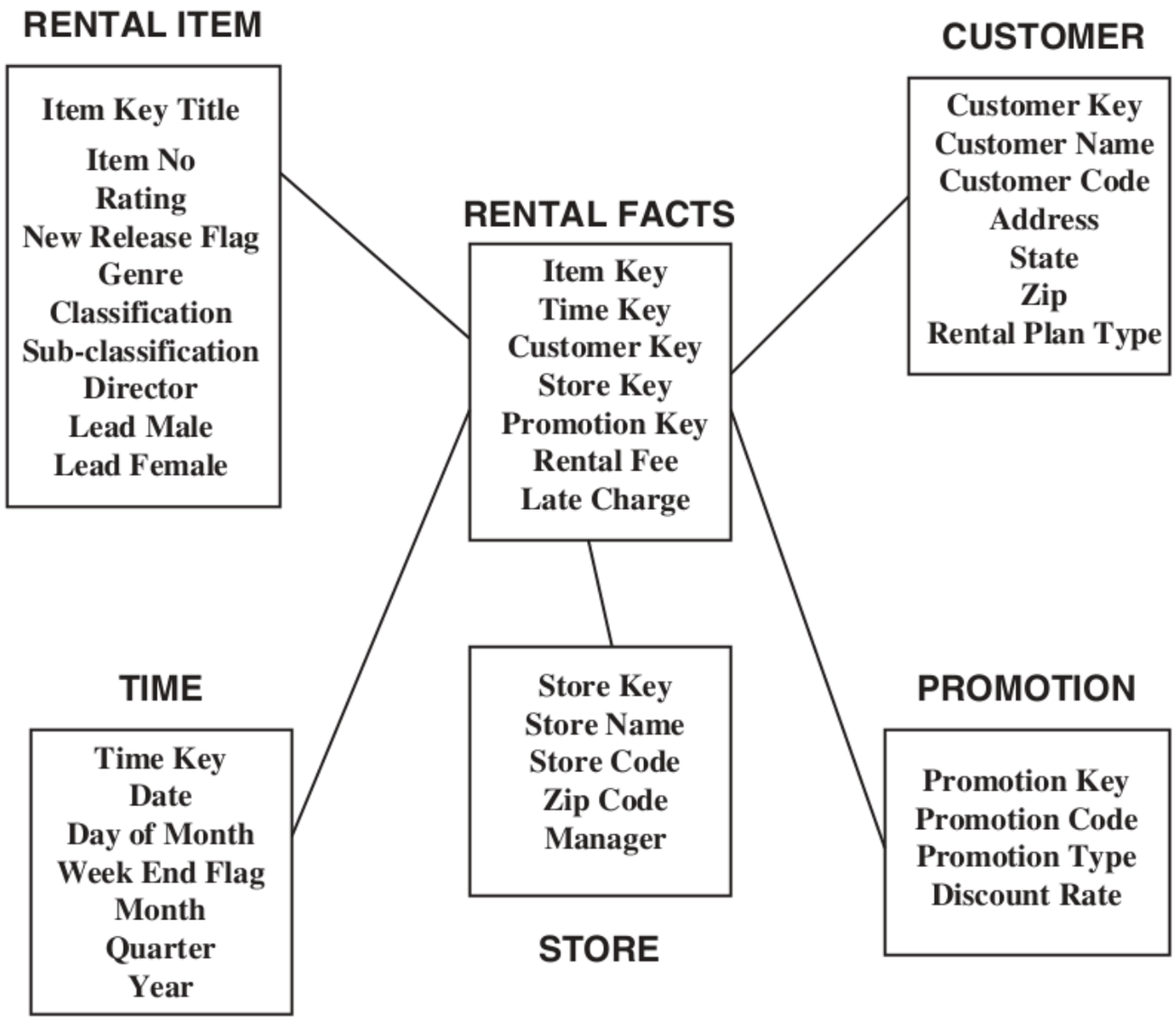
บริษัทประมูล



บริษัทให้เช่าวีดีโอ

บริษัทให้เช่าวีดีโอ—จะสนใจเกี่ยวกับหัวข้อ (subject) การเช่าวีดีโอเป็นหลัก มาตรวัดความสำเร็จของการเช่าวีดีโออาจะประกอบไปด้วยจำนวนเงินที่ได้รับจากค่าเช่าและจำนวนค่าปรับในกรณีที่ส่งคืนวีดีโอช้าเป็นต้น

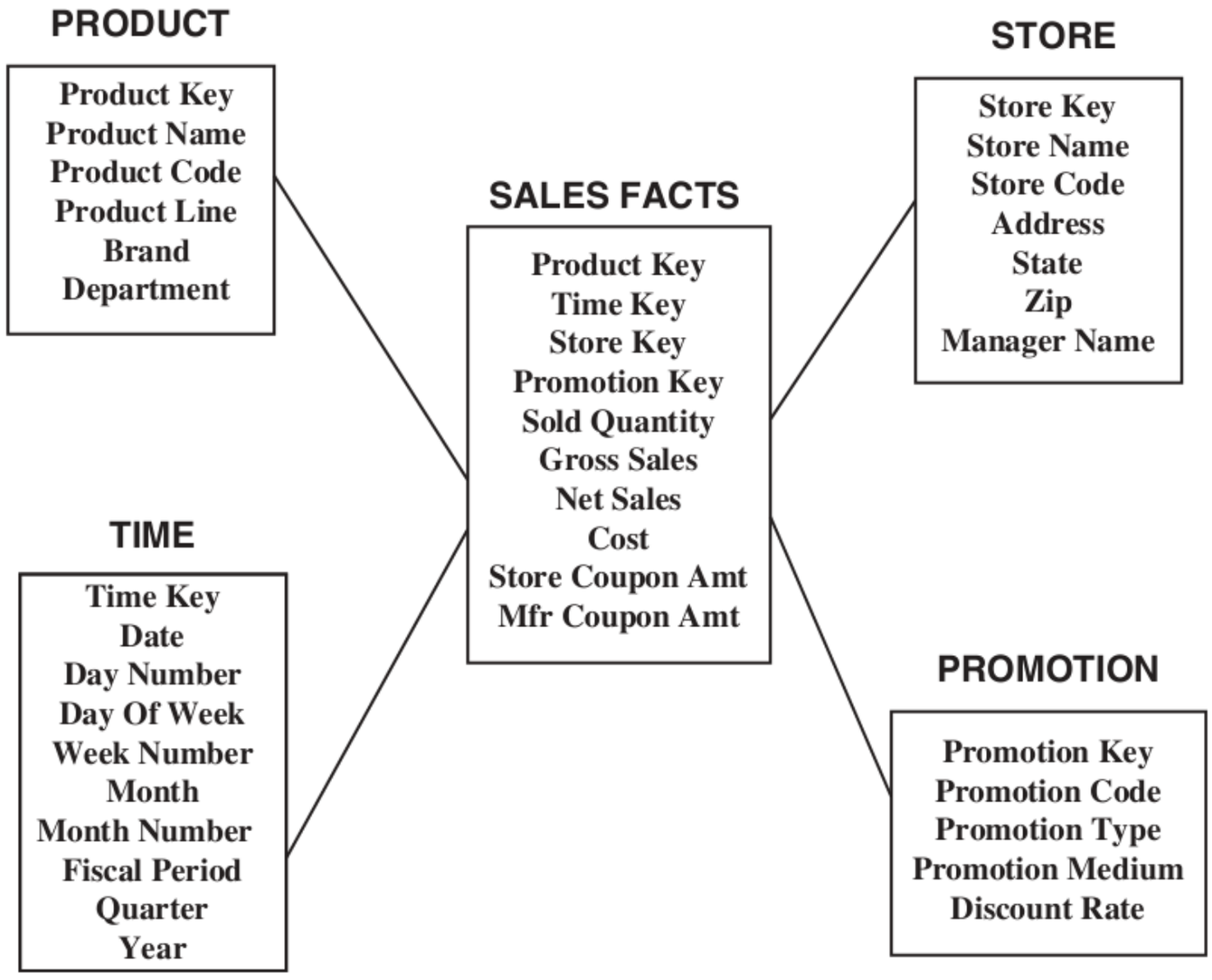
นอกจากนี้มิติทางธุรกิจที่เกี่ยวข้องกับการเช่าวีดีโออาจจะประกอบไปด้วย รายการวีดีโอ (rental item) ลูกค้า (customer) แกนเวลา (time) และโปรโมชั่น (promotion) สำหรับการเช่าวีดีโอเมื่อเราทำการกำหนดมาตรวัดความสำเร็จของการเช่าวีดีโอ และมิติทางธุรกิจที่เกี่ยวข้องกับการเช่าวีดีโอแล้ว เราสามารถสร้าง star schema ของบริษัทให้เช่าวีดีโอได้ดังรูปที่ 7-10



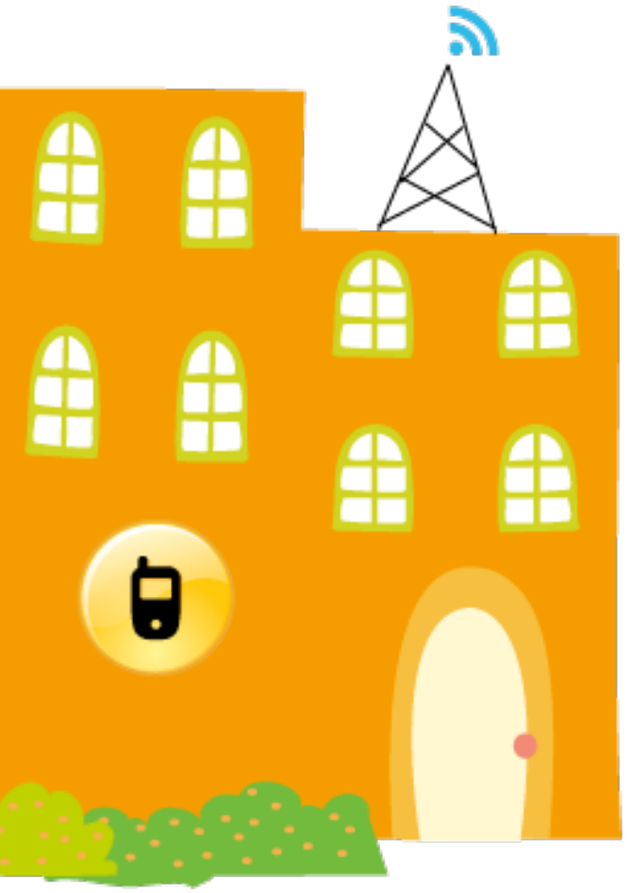
รูปที่ 7-10 ตัวอย่าง star schema การเช่าวีดีโอสำหรับบริษัทให้เช่าวีดีโอ

ธุรกิจซุปเปอร์มาเก็ต

ธุรกิจซุปเปอร์มาเก็ต—จะสนใจเกี่ยวกับการขายสินค้าเป็นหลัก ซึ่งมาตรวัดความสำเร็จของการขายสินค้าจะประกอบไปด้วย จำนวนชิ้นสินค้าแต่ละชนิดที่ขายได้ (sold quantity) ยอดขายที่ได้ขายทั้งหมดเป็นจำนวนเงิน (gross sales) ยอดขายทั้งหมดเป็นจำนวนเงินหลังจากหักค่าใช้จ่ายแล้ว (net sales) ค่าใช้จ่ายต่างๆ (cost) ยอดการใช้คูปองส่วนลดจากทางร้านค้า (store coupon amount) และยอดการใช้คูปองส่วนลดจากบริษัทผู้ผลิต (manufacturer coupon amount) เป็นต้น ในส่วนของมิติทางธุรกิจของธุรกิจซุปเปอร์มาร์เก็ตจะประกอบไปด้วย แกนเวลา (time) รายการสินค้า (product) โปรโมชั่นการขายสินค้า (promotion) และ สาขาต่างๆ ที่มีการจำหน่ายสินค้าของบริษัท (store) เมื่อทราบรายละเอียดเกี่ยวกับมาตรวัดความสำเร็จของการสินค้า และมิติทางธุรกิจต่าง ๆ ที่เกี่ยวข้องแล้ว เราสามารถสร้าง star schema สำหรับการขายสินค้าของธุรกิจซุปเปอร์มาร์เก็ตได้ดังรูปที่ 7-11



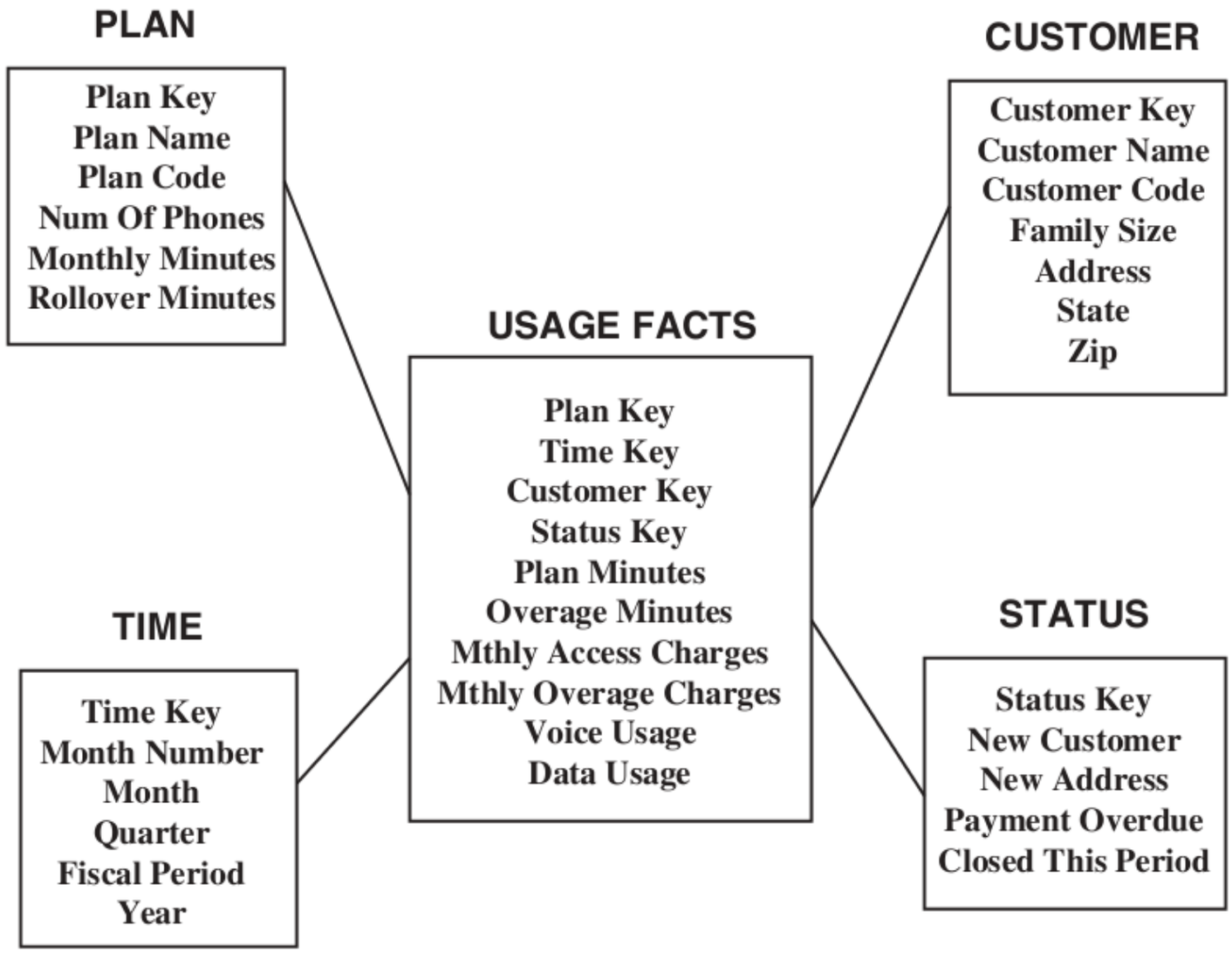
รูปที่ 7-11 ตัวอย่าง star schema สำหรับการขายสินค้าของธุรกิจซุปเปอร์มาร์เก็ต



บริษัทผู้ให้บริการระบบโทรศัพท์เคลื่อนที่

บริษัทผู้ให้บริการระบบโทรศัพท์เคลื่อนที่—จะสนใจเกี่ยวกับการใช้งานระบบโทรศัพท์เคลื่อนที่ของลูกค้าเป็นหลัก ซึ่งมาตรวัดความความสำเร็จของการใช้งานระบบโทรศัพท์เคลื่อนที่ของลูกค้าจะประกอบไปด้วย แพคแกจของบริษัทที่ลูกค้าเลือกใช้ (plan minutes) จำนวนนาทีที่ผู้ใช้ทำการโทรศัพท์ไปหาผู้อื่น (overall minutes) จำนวนปริมาณข้อมูลที่ผู้ใช้เรียกดูผ่านอินเทอร์เน็ตของโทรศัพท์เคลื่อนที่ (data usage) เป็นต้น

ในส่วนของมิติทางธุรกิจที่เกี่ยวข้องกับการใช้งานโทรศัพท์เคลื่อนที่จะประกอบด้วย แพคเกจหรือโปรโมชั่นจากบริษัทผู้ให้บริการ (plan) แกนเวลา (time) สถานะของลูกค้า (status) และข้อมูลลูกค้า (customer) เป็นต้น จากข้อมูลทั้งในส่วนของมาตรวัดความสำเร็จและมิติทางธุรกิจของการใช้งานระบบโทรศัพท์เคลื่อนที่ของลูกค้า เราสามารถสร้าง star schema ได้ดังรูปที่ 7-12

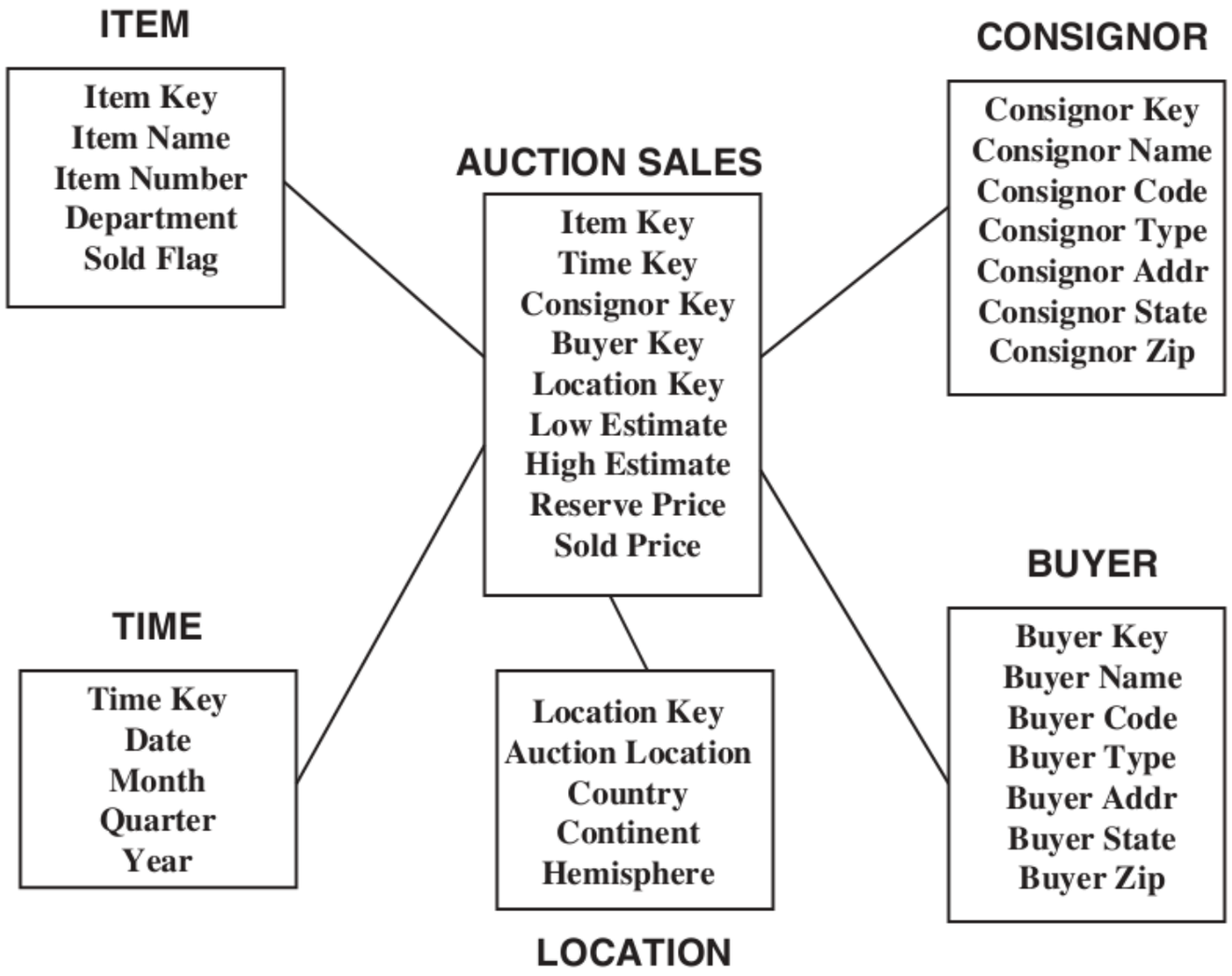


รูปที่ 7-12 ตัวอย่าง star schema สำหรับการใช้โทรศัพท์เคลื่อนที่ของลูกค้าจากบริษัทผู้ให้บริการระบบโทรศัพท์เคลื่อนที่

บริษัทประมูล

บริษัทประมูล—จะสนใจที่การประมูลสินค้าเป็นหลัก ซึ่งมาตรวัดความสำเร็จของการประมูลจะประกอบไปด้วย จำนวนครั้งของการประมูลที่ได้ราคาต่ำกว่าที่ประเมินไว้ (low estimate) จำนวนครั้งของการประมูลที่ได้ราคาสูงกว่าที่ประเมินไว้ (high estimate) ราคาที่จองไว้ (reserve price) และราคาที่ขายสินค้าหลังจากการประมูล (sold price) เป็นต้น

ในส่วนของมิติทางธุรกิจที่เกี่ยวข้องกับการประมูลสินค้าจะเกี่ยวข้องกับรายการสินค้าที่นำมาประมูล (item) ลูกค้าที่ทำการประมูลสินค้า (buyer) ผู้ดำเนินการประมูล (consignor) และ แกนเวลา (time) จากข้อมูลทั้งมาตรวัดความสำเร็จของการประมูลและมิติต่างๆทางธุรกิจ เราสามารถสร้าง star schema สำหรับการประมูลสินค้าได้ดังรูปที่ 7-13



รูปที่ 7-13 ตัวอย่าง star schema สำหรับการประมูลสินค้าของบริษัทประมูล

ประโยชน์ของ star schema

เมื่อเราพิจารณา star schema แบบเจาะลึก เราจะเห็นว่า star schema จะเป็นแบบจำลองความสัมพันธ์ (relational model) ของ fact table และ dimension table ที่อยู่ในรูปแบบของ one-to-many โดยตารางทั้งสองประเภทจะไม่มีการทำนอร์มอลไลซ์ (normalized) ข้อมูลเพื่อลดความซ้ำซ้อนของข้อมูลที่เกิดขึ้น

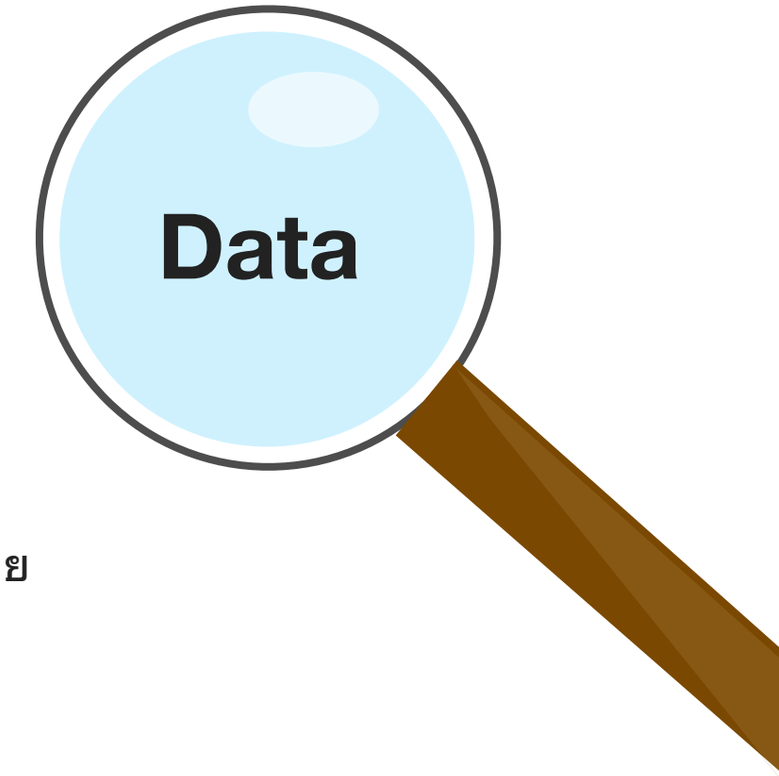
ดังนั้นข้อมูลในแต่ละ dimension table ของ star schema อาจเกิดความซ้ำซ้อนเกิดขึ้นได้ แต่อย่างไรก็ดี star schema ก็มีประโยชน์ต่าง ๆ มากมายดังต่อไปนี้

ผู้ใช้สามารถเข้าใจได้ง่าย

ผู้ใช้สามารถเข้าใจได้ง่าย—อย่างที่เราทราบกันมาแล้วในบทก่อนหน้าว่าผู้ใช้คลังข้อมูลสามารถกำหนดหรือสร้างคิวรีเองเพื่อทำการเรียกดูข้อมูลในแง่มุมต่าง ๆ ได้ โดยที่ก่อนเริ่มการใช้งานคลังข้อมูล ผู้ใช้จะต้องทราบว่าพวกเขาสามารถเรียกดูข้อมูลอะไรได้บ้าง รายละเอียดของข้อมูลเป็นอย่างไรบ้าง ความสัมพันธ์ระหว่างชิ้นส่วนต่าง ๆ ของข้อมูลเป็นอย่างไรบ้าง เป็นต้น

ดังนั้นเพื่อให้ผู้ใช้สามารถใช้งานคลังข้อมูลผ่านการกำหนดหรือเรียกใช้คิวรี ผู้ใช้จะต้องสามารถทำความเข้าใจเกี่ยวกับโครงสร้างของข้อมูลในคลังข้อมูลผ่าน star schema ที่สะท้อนถึงสิ่งที่ผู้ใช้คิด รูปแบบรายงาน และการวิเคราะห์ต่างๆที่ผู้ใช้ต้องการได้ ซึ่งจากโครงสร้างของ star schema หนึ่งๆ เช่น star schema ที่เกี่ยวข้องกับการขายสินค้า เราจะอธิบายผู้ใช้ให้เข้าใจได้ว่ายอดขายของรายการสินค้า A จะถูกเก็บอยู่ใน fact table และยังสามารถชี้ให้เห็นถึงความสัมพันธ์ของส่วนประกอบต่าง ๆ ของข้อมูลผ่านทาง dimension table ต่าง ๆ ได้อีกด้วย ซึ่งจากการอธิบายโดยการอ้างถึง star schema จะทำให้ผู้ใช้เข้าใจถึงการเชื่อมโยงของข้อมูลได้ในทันที

**เพิ่มประสิทธิภาพใน
การเข้าถึงหรือค้นหาข้อมูล**



เพิ่มประสิทธิภาพในการเข้าถึงหรือค้นหาข้อมูล—แม้ว่าคิวรีที่สร้างขึ้นดูเหมือนจะซับซ้อน แต่การค้นหาข้อมูลนั้นสามารถทำได้โดยง่ายและตรงไปตรงมา เพื่อให้เข้าใจถึงประโยชน์ที่ได้จาก star schema ในเรื่องของการเพิ่มประสิทธิภาพการค้นหาข้อมูล ลองพิจารณาตัวอย่างในรูปที่ 7-14 ที่แสดงตัวอย่าง star schema สำหรับการวิเคราะห์ข้อบกพร่องของรถยนต์ในขั้นตอนการผลิต โดยจาก star schema ดังกล่าว ถ้าคุณเป็นผู้จัดการฝ่ายบริการที่ศูนย์ตัวแทนจำหน่ายรถยนต์แห่งหนึ่ง คุณอาจสังเกตุเห็นได้ว่ามีปัญหาของการพ่นสีเกิดขึ้นบนรถ Corvettes ที่ผลิตขึ้นในเดือนมกราคมปี 2012 เป็นจำนวนมาก

ดังนั้น คุณอาจจะต้องการเครื่องมือในการวิเคราะห์ข้อบกพร่องเหล่านั้น เพื่อให้ทราบถึงสาเหตุที่ซ่อนอยู่และทำการแก้ไขปัญหาเหล่านั้น

ถ้าเราทำการสร้าง star schema สำหรับการตรวจจับข้อบกพร่องของการผลิตรถยนต์ โดยทำการเก็บรวบรวมจำนวนข้อบกพร่องที่เป็นมาตรวัดไว้ใน fact table ที่อยู่ตรงกลาง จากนั้นรายล้อม fact table ด้วย dimension table ต่างๆ ดังนี้:

(1) time dimension table ที่เก็บข้อมูลปีของรถรุ่นหนึ่ง ๆ

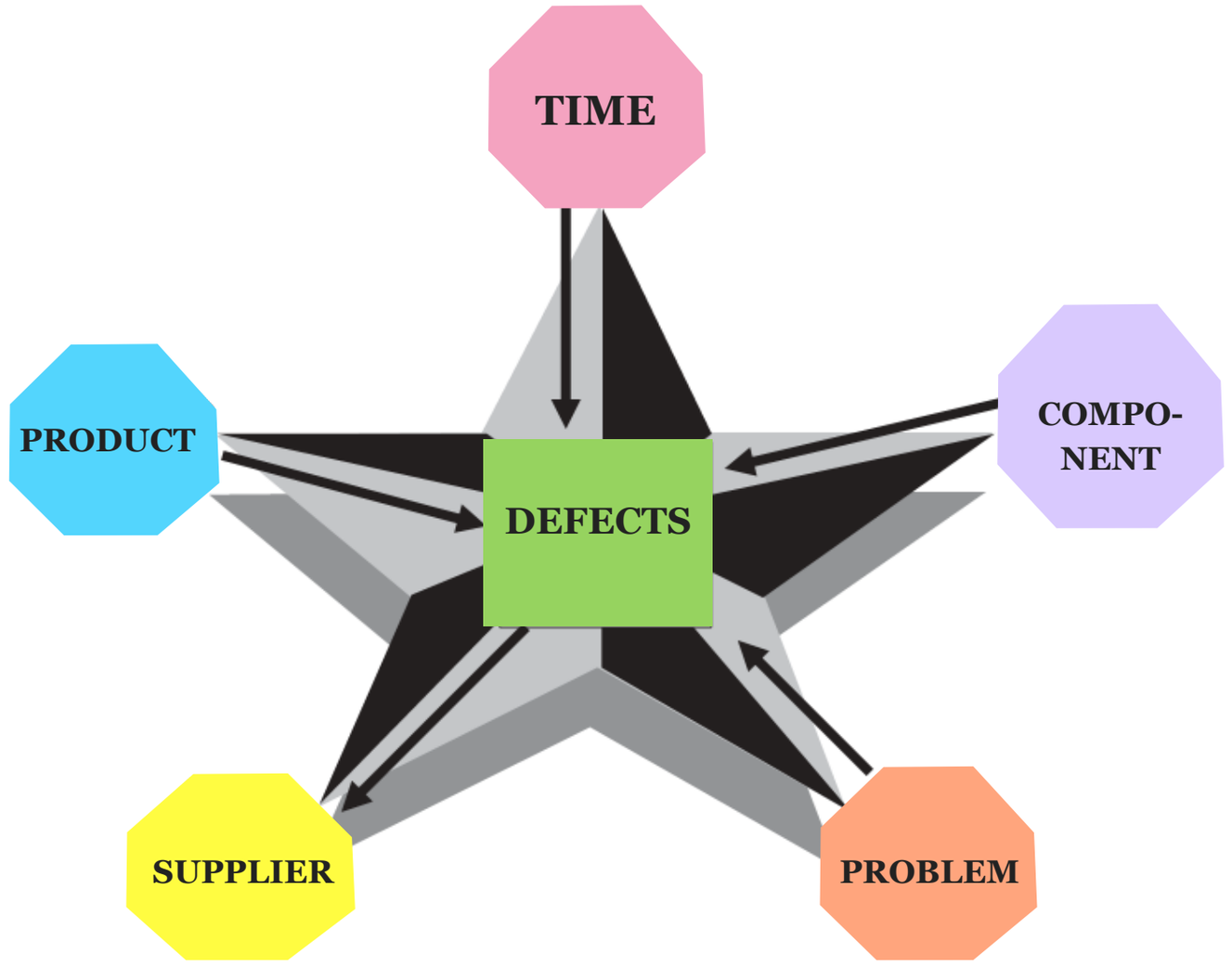
(2) component dimension ที่เก็บข้อมูลเกี่ยวกับส่วนต่างๆของรถ เช่น สีขาวมุก เป็นต้น

(3) problem dimension ที่เก็บข้อมูลเกี่ยวกับปัญหาชนิดต่าง ๆ ที่อาจะเกิดขึ้นจากการผลิต เช่น การพ่นสีรถยนต์ เป็นต้น

(4) product dimension ที่ประกอบไปด้วยข้อมูลรายละเอียดของรถยนต์แต่ละรุ่น

(5) supplier dimension ที่เก็บข้อมูลเกี่ยวกับผู้ผลิตชิ้นส่วนต่าง ๆ ของรถยนต์

ซึ่งจาก fact และ dimension table ของ star schema เราจะสามารถทราบถึงสาเหตุหรือต้นตอของการเกิดปัญหาต่าง ๆ ได้ โดยการดูจากลูกศรต่าง ๆ จาก dimension table ที่ชี้เข้าหา fact table ซึ่งจากลูกศรต่าง ๆ จะทำให้เรารู้เส้นทางของการสืบค้นข้อมูลจาก fact table โดยการแยก (1) รถ Corvette จาก product dimension (2) การพ่นสีออกจากปัญหา (3) อุปกรณ์ที่มีสีขาวมุกออกจาก component dimension และเดือนมกราคมปี 2012 ออกจาก time dimension ซึ่งจากข้อมูลที่ได้จาก fact table ที่ตรงตามเงื่อนไขต่าง ๆ ของทั้ง 4 dimension ข้างต้น เราจะทราบถึงต้นตอของปัญหาโดยการดูข้อมูลเฉพาะส่วนของ supplier dimension ที่เกี่ยวข้องกับแถวต่างๆของ fact table ที่ได้จากการสืบค้นก่อนหน้า ท้ายสุดเราจะได้รายชื่อของ supplier ที่ผลิตอุปกรณ์หรือชิ้นส่วนต่าง ๆ ที่มีปัญหาของรถ Corvette เพื่อแสดงผลให้ผู้ใช้สืบไป



รูปที่ 7-14 ตัวอย่างประสิทธิภาพที่เพิ่มขึ้นของการเข้าถึงข้อมูลใน star schema

เหมาะกับการประมวลผลคิวรี

เหมาะกับการประมวลผลคิวรี—ลองพิจารณา star schema ในรูปที่ 7-4 เพื่อทำความเข้าใจถึงประโยชน์ของ star schema ในการประมวลผลคิวรีจากผู้ใช้ ซึ่งจากรูปที่ 7-4 จะเป็น star schema สำหรับวิเคราะห์ข้อมูลการสั่งซื้อสินค้าจากลูกค้าที่ประกอบไปด้วย 1 fact table และ 3 dimension table ด้วยกัน ถ้าผู้ใช้ทำการสร้างคิวรีเพื่อถามถึงข้อมูลเกี่ยวกับ “ค่าใช้จ่าย (cost) ของสินค้า A ที่ขายให้กับลูกค้าในรัฐซานฟรานซิสโก ในช่วงเดือนมกราคมปี 2012 ที่ผ่านมา” ผลลัพธ์ที่ได้คือ ค่าใช้จ่ายที่มาจากผลรวมของข้อมูลแถวต่าง ๆ ใน fact table ที่เกี่ยวข้องกับเวลาในช่วงเดือน มกราคมปี 2012 รายการสินค้า A และลูกค้าที่อยู่ในรัฐซานฟรานซิสโก

ในรัฐซานฟรานซิสโก

จากผลลัพธ์ที่ได้จากการสืบค้นข้อมูลตามคิวรีที่ผู้ใช้กำหนด ลองพิจารณาแต่ละขั้นตอนของการสืบค้นข้อมูลดังต่อไปนี้

1. ทำการเลือก *(select)* แถวของข้อมูลจาก *customer dimension table* ที่พักอาศัยอยู่ในรัฐซานฟรานซิสโก

2. ทำการเลือกผลลัพธ์จาก *fact table* โดยทำการเลือกเฉพาะแถวของข้อมูลที่เกี่ยวข้องกับลูกค้าที่ได้จากการสืบค้นข้อมูลในขั้นตอนที่ *1*

3. ทำการเลือกแถวของข้อมูลจาก *time dimension table* ที่มีข้อมูลเดือนและปี เป็นเดือนมกราคมปี *2012* ตามลำดับ

4. ทำการเลือกผลลัพธ์จากผลลัพธ์ที่ได้จากการสืบค้นข้อมูลจาก *fact table* ในครั้งแรก *(*ผลลัพธ์จากขั้นตอนที่ *2)* ที่มีข้อมูลแกนเวลาเกี่ยวข้องกับผลลัพธ์ที่ได้จากการสืบค้นข้อมูลจาก *time dimension table (*ผลลัพธ์จากขั้นตอนที่ *3) (*ผลลัพธ์จากขั้นตอนนี้จะเป็นเซตของแถวของข้อมูลจาก *fact table*ที่มีจำนวนน้อยกว่าการสืบค้นข้อมูลจาก *fact table* ในครั้งแรกซึ่งเราอาจเรียกได้ว่าเป็นผลลัพธ์จำนวนแถวส่วนที่สองจาก *fact table)*

5. ทำการเลือกแถวของข้อมูลจาก *product dimension table* ที่เป็นข้อมูลรายการสินค้า *A*

6. ทำการเลือกผลลัพธ์จากผลลัพธ์จำนวนแถวส่วนที่สองจาก *fact table (*ผลลัพธ์จากขั้นตอนที่ *4)* ที่เกี่ยวข้องกับรายการสินค้า *A* เราจะได้ผลลัพธ์จำนวนแถวของข้อมูลส่วนที่สามจาก *fact table*

7. ทำการรวมค่าใช้จ่าย *(cost)* จากทุกแถวของข้อมูลในผลลัพธ์ส่วนที่สามจาก *fact table (*ผลลัพธ์จากขั้นตอนที่ *6)* เพื่อคืนค่าผลลัพธ์ให้กับผู้ใช้สืบไป

จากตัวอย่างข้างต้น ถ้าเราไม่คำนึงถึงจำนวนมิติของ star schema ที่เกี่ยวข้องกับคิวรีที่ผู้ใช้กำหนด และไม่คำนึงถึงประสิทธิภาพของการค้นคืนผลลัพธ์ให้กับคิวรีแล้ว เราจะสามารถเริ่มทำการเลือกแถวของข้อมูลจากมิติใดเป็นลำดับแรกก็ได้โดยใช้เงื่อนไขที่ผู้ใช้กำหนด การเลือกมิติใดก็ได้มาประมวลผลเป็นลำดับแรก และ ลำดับต่อ ๆ ไปจะช่วยให้ผู้ใช้สามารถใช้งานได้ง่าย และขั้นตอนการเชื่อมโยงระหว่างมิติต่าง ๆ กับ fact table ก็สามารถทำได้โดยง่าย

นอกเหนือจากประโยชน์และขั้นตอนการทำงานของ star schema ข้างต้น star schema ยังสามารถรองรับการเรียกดูหรือสืบค้นข้อมูลแบบขุดเจาะลงรายละเอียด (drill down) และแบบสรุปรายละเอียด (roll up) ได้อีกด้วย ลองพิจารณาการสืบค้นข้อมูลแบบขุดเจาะลงรายละเอียด ที่เริ่มจากการสอบถามข้อมูลค่าใช้จ่ายสำหรับลูกค้าในรัฐแคลิฟอร์เนีย ซึ่งผลลัพธ์จะได้มาจากแถวของข้อมูลใน fact table จากนั้นเราสามารถสืบค้นข้อมูลแบบขุดเจาะลงรายละเอียดจากการสืบค้นครั้งล่าสุดได้ โดยที่เราอาจจะต้องการข้อมูลค่าใช้จ่ายสำหรับลูกค้าที่พักอาศัยอยู่ในพื้นที่รหัสไปรษณีย์หนึ่ง ๆ ของรัฐแคลิฟอร์เนีย ซึ่งจากความต้องการดังกล่าวจะทำให้เราต้องทำการเลือกข้อมูลจากผลลัพธ์ที่ได้ในคิวรีแรกที่สอดคล้องกับรหัสไปรษณีย์ที่กำหนด เป็นต้น

ในการสร้างแบบจำลองมิติต่างๆสำหรับคลังข้อมูลนั้นไม่ได้มี star schema รูปแบบเดียว แต่ยังคงมีรูปแบบอื่น ๆ อีกหลายแบบ เช่น snowflaking และ family of stars (galaxy) schema เป็นต้น ลองพิจารณา schema ในรูปแบบอื่น ๆ ดังต่อไปนี้ที่อาจจะสามารถนำไปประยุกต์ใช้กับคลังข้อมูลเราได้

SECTION 4

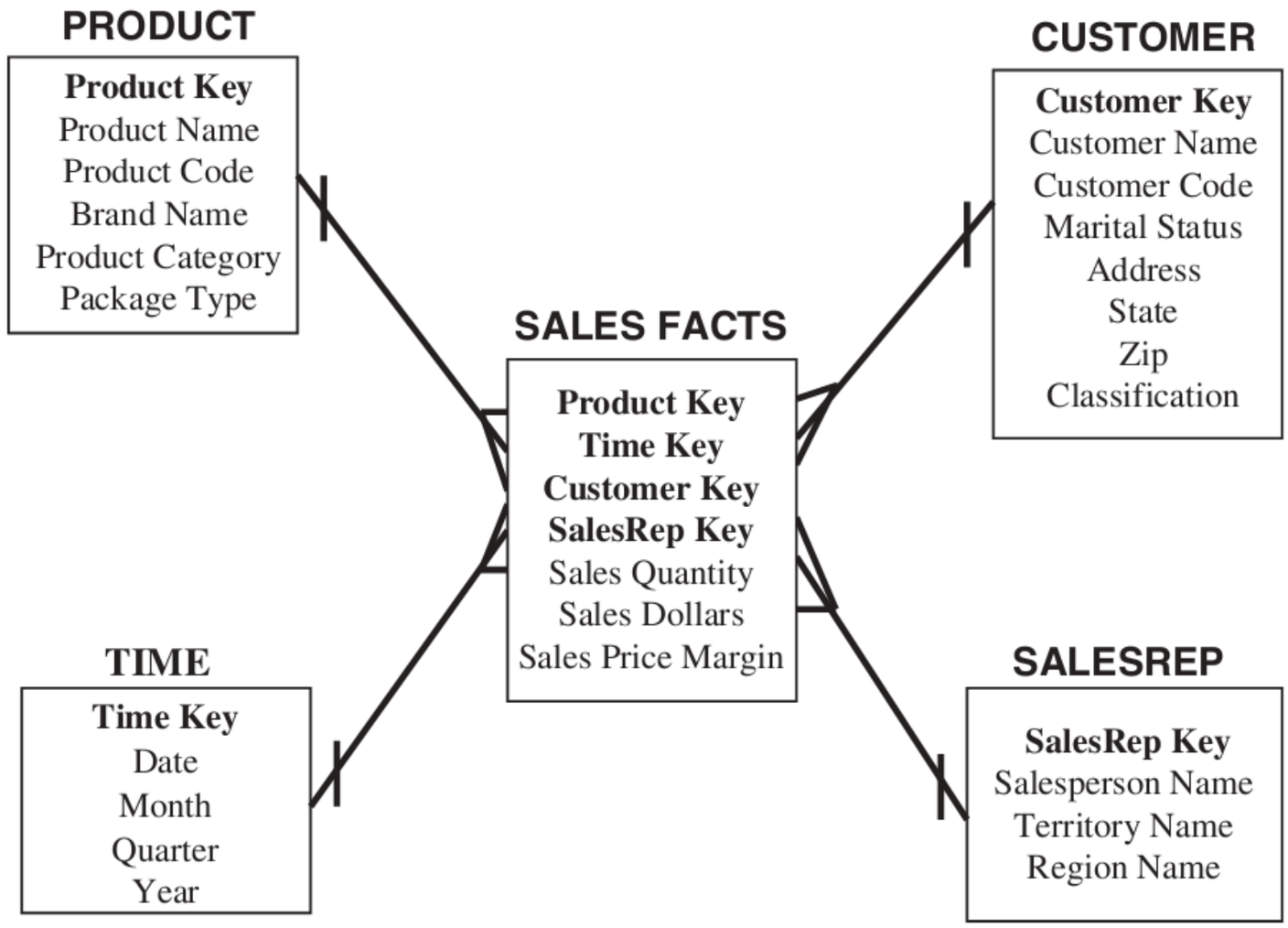
Snowflake Scehma

Snowflake schema

Snowflake schema เป็นแบบจำลองมิติต่าง ๆ รูปแบบหนึ่งที่พัฒนาต่อยอดจาก star schema โดยเพิ่มการทำนอร์มอลไลซ์กับข้อมูลในแต่ละ dimension table ที่เป็นส่วนประกอบของ star schema นั้น ๆ หลังจากการทำนอร์มอลไลซ์ทุก ๆ dimension แล้ว ผลที่ได้จะได้เป็น snowflake schema ที่มี fact table อยู่ตรงกลางรายล้อมไปด้วย dimension table ที่มีการทำนอร์มอลไลซ์แล้ว

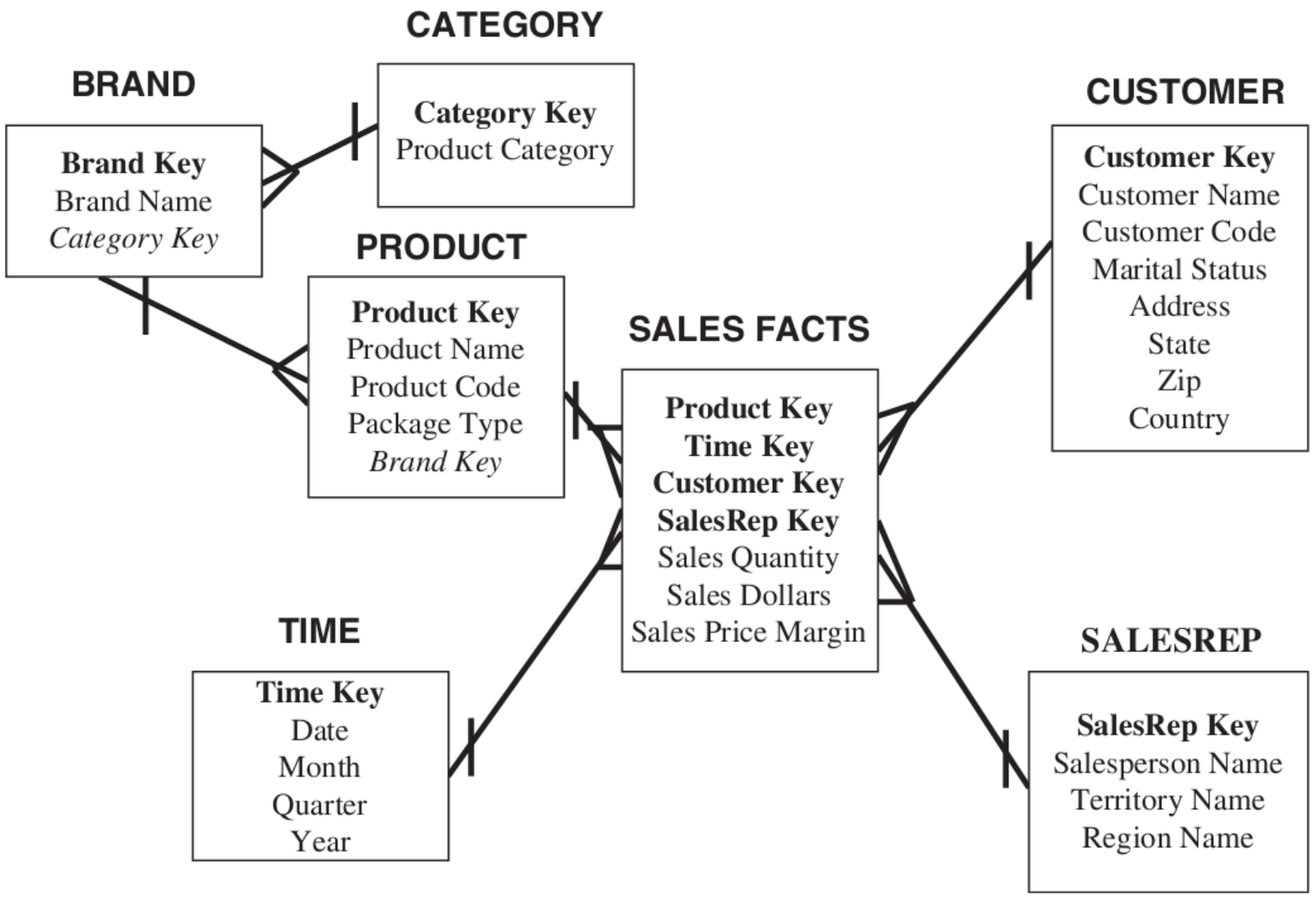
ลองพิจารณารูปที่ 7-15 ที่แสดงถึง star schema สำหรับการขายสินค้าจากบริษัทผู้ผลิตสินค้า ที่ประกอบไปด้วย 4 dimension table คือ รายการสินค้า (product dimension table) ลูกค้า (customer dimension table) พนักงานขาย (sales representative dimension table) และเวลา (time dimension table) และ fact table ที่ประกอบไปด้วยตัวชี้วัดต่าง ๆ เช่น จำนวนชิ้นสินค้าที่ขายได้จำนวนเงินที่ขายได้ และผลกำไรจากการขายสินค้าจากรูปจะเป็นตัวอย่าง star schema ที่ dimension table ยังไม่มีการทำนอร์มอลไลซ์ที่อยู่ในรูปแบบของ 3rd normal form เพื่อให้สามารถคิวรีข้อมูลได้อย่างรวดเร็ว

แต่อย่างไรก็ตาม ในบางสถานการณ์เราอาจจะต้องทำการนอร์มอลไลซ์ dimension ต่าง ๆ ของ schema เพื่อลดความซ้ำซ้อนของข้อมูล และเพื่อผลประโยชน์อื่น ๆ แต่ก่อนที่จะทำการสร้าง snowflake schema หรือการทำ normalized กับ dimension table ของ star schema เราจะต้องทำการพิจารณาทางเลือกต่าง ๆ รวมถึงข้อดี-ข้อเสียของการทำนอร์มอลไลซ์เสียก่อน แล้วจึงค่อยเริ่มดำเนินการ



รูปที่ 7-15 ตัวอย่าง star schema ของการขายสินค้า

สมมติว่าใน dimension table ของรายการสินค้ามีข้อมูลอยู่ 500,000 เรคคอร์ด โดยมียี่ห้อสินค้าทั้งหมด 500 ยี่ห้อและสามารถแบ่งหมวดหมู่ของสินค้าได้ทั้งหมด 10 ประเภทด้วยกัน ถ้าสมมติว่าข้อมูลรายการสินค้าไม่มีการทำนอร์มอลไลซ์ (ไม่มีแยกตารางสำหรับหมวดหมู่สินค้าใน product dimension table) การสืบค้นเพื่อคืนค่าผลลัพธ์ให้กับคิวรีที่เกี่ยวกับหมวดหมู่สินค้าจะต้องทำการอ่านข้อมูลทั้ง 500,000 เรคคอร์ด แต่ถ้ามีการทำนอร์มอลไลซ์ที่ product dimension table โดยทำการแยกข้อมูลเกี่ยวกับยี่ห้อสินค้าและประเภทสินค้ามาเก็บไว้ในแต่ละตาราง (ดังแสดงในรูปที่ 7-16) จะทำให้การค้นหาคำตอบในเบื้องต้นสำหรับคิวรีข้างต้นจะทำการอ่านข้อมูลเพียงแค่ 10 เรคคอร์ดจากตารางประเภทของสินค้าเท่านั้น

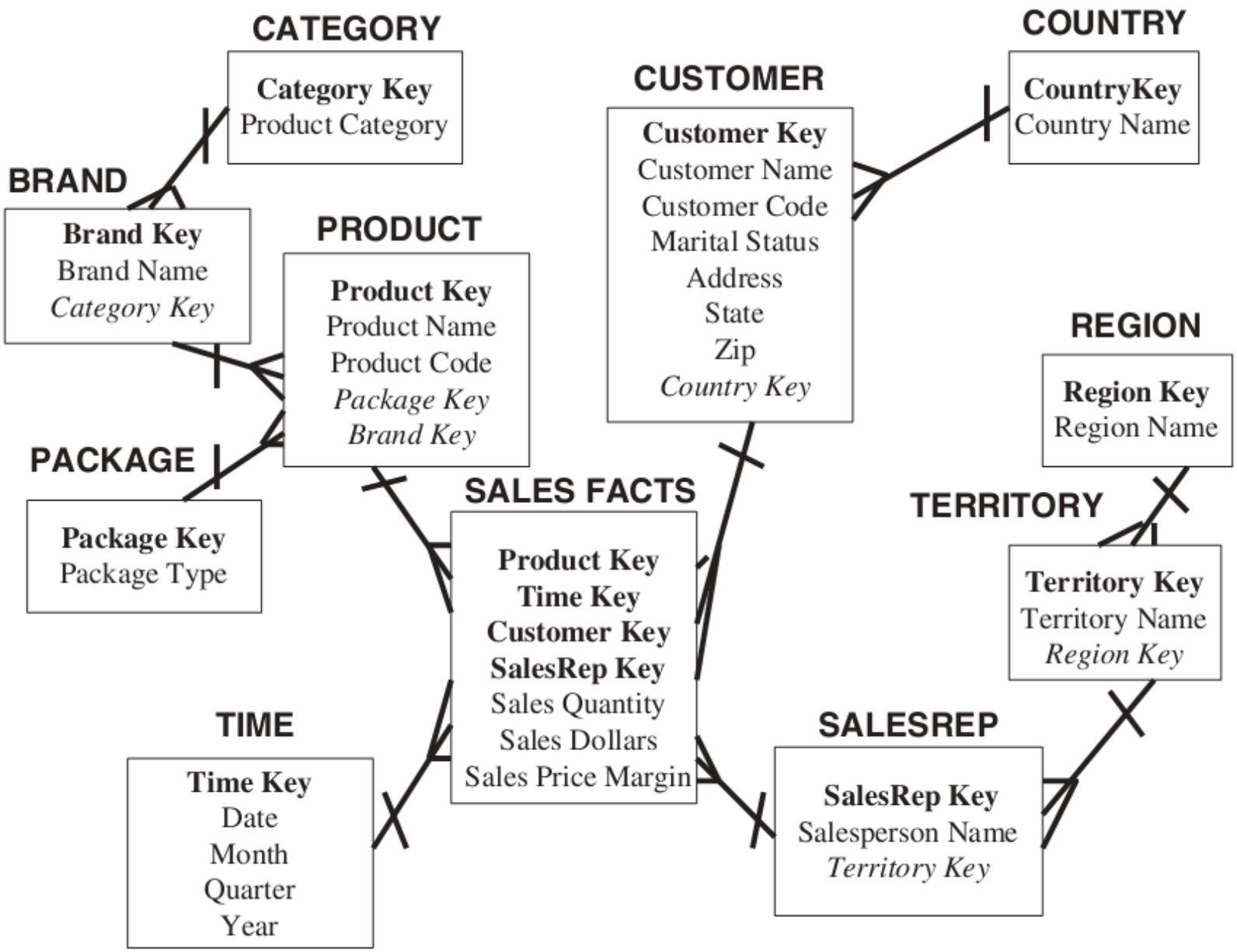


รูปที่ 7-16 การทำนอลมอลไลซ์เพียงบางส่วนกับ Product dimension table

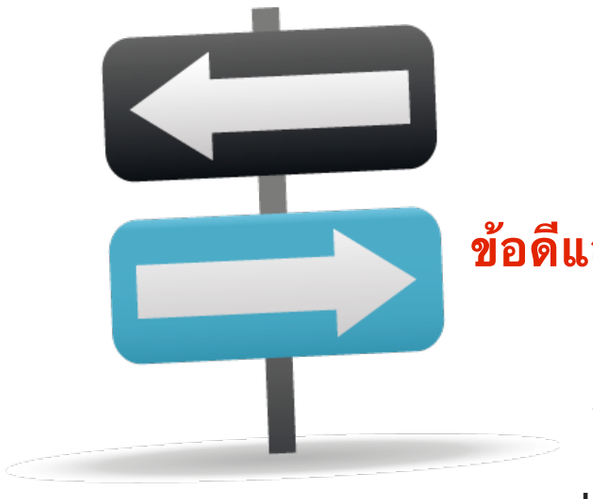
การทำนอร์มอลไลซ์กับตารางข้อมูลสินค้าดังแสดงในรูปที่ 7-16 นั้นยังไม่สมบูรณ์ เราสามารถทำการย้ายแอทริบิวต่าง ๆ ออกจากตารางข้อมูลสินค้า และทำการสร้างโครงสร้างสำหรับการนอร์มอลไลซ์ซึ่งในการทำนอร์มอลไลซ์ (snowflaking) กับ dimension table ต่าง ๆ เราจะสามารถทำได้หลายวิธี แต่ก่อนที่เราจะทำการนอร์มอลไลซ์ข้อมูล เราจะต้องพิจาณาถึงเนื้อหาของข้อมูลที่ถูกเก็บไว้ใน dimension table นั้น ๆ และ พิจารณาถึงพฤติกรรมการใช้งานข้อมูลเพื่อที่จะเลือกวิธีในการทำนอร์มอลไลซ์ดังนี้

- ทำการนอร์มอลไลซ์เพียงบางส่วนกับ dimension table ไม่กี่ตาราง แล้วไม่ทำการเปลี่ยนแปลงตารางอื่นๆ
- ทำการนอร์มอลไลซ์เพียงบางส่วนหรือนอร์มอลไลซ์ทั้งหมดกับ dimension table ไม่กี่ตาราง แล้วปล่อยให้ที่เหลือไม่เปลี่ยนแปลง
- ทำการนอร์มอลไลซ์เพียงบางส่วนกับทุก dimension table
- ทำการนอร์มอลไลซ์ทั้งหมดกับทุก dimension table

จากวิธีการทำนอร์มอลไลซ์ข้างต้น เราสามารถนำทั้ง 4 วิธีมาผสมกันได้ ดังแสดงในรูปที่ 7-17 ที่แสดงsnowflake schema ของการขายสินค้าที่ทุก dimension table จะถูกทำนอร์มอลไลซ์ โดยแต่ละตารางจะถูกนอร์มอลไลซ์เพียงบางส่วนหรือทั้งหมดอย่างใดอย่างหนึ่ง โดยที่ snowflake schema ในรูปที่ 7-17 จะมีตารางทั้งหมด 11 ตาราง ซึ่งเพิ่มจาก star schema เดิมที่มีเพียงแค่ 5 ตารางเท่านั้น โดยที่หลักในการพิจารณาของการเลือกแอทริบิวจาก dimension table ที่จะถูกย้ายไปยังตารางใหม่ควรจะเลือกแอทริบิวที่มีค่าที่แตกต่างกันน้อยๆ (low cardinality) และเมื่อทำการแยกตารางออกเป็นตารางใหม่แล้วเราจะต้องสร้างการเชื่อมโยงกลับไปยัง dimension table ผ่านคีย์ต่างๆ



รูปที่ 7-17 ตัวอย่างการทำ snowflaking กับการขายสินค้า



ข้อดีและข้อเสียของการทำ snowflaking

โดยปกติแล้วเราจะทำการนอร์มอลไลซ์เพื่อลดการจัดเก็บข้อมูลที่ซ้ำซ้อน ซึ่งจะช่วยให้ลดพื้นที่ใช้ในการเก็บข้อมูล แต่อย่างไรก็ดีการทำนอร์มอลไลซ์ก็มีข้อเสียเช่นกัน สมมติว่า dimension table ของรายการสินค้าประกอบด้วยข้อมูล 500,000 เรคคอร์ด เมื่อทำการ snowflaking โดยการสร้างตารางใหม่ให้กับแอทริบิวประเภทของสินค้า เราจะสามารถเคลื่อนย้ายข้อมูลเกี่ยวกับประเภทสินค้าจากแต่ละเรคคอร์ดได้ประมาณ 20 ไบต์ แต่เมื่อย้ายข้อมูลออกไปแล้ว อย่างที่เราทราบกันดีว่าเราต้องสร้างคีย์ไว้ใน dimension table เพื่อเชื่อมโยงข้อมูลกับตารางที่สร้างใหม่ โดยที่คีย์ที่สร้างขึ้นจะใช้พื้นที่ประมาณ 4 ไบต์ต่อ 1 เรคคอร์ด เมื่อคำนวณพื้นที่ที่สามารถลดได้จาก dimension table จะพบว่าจะสามารถลดการจัดเก็บข้อมูลได้ประมาณ 16 ไบต์สำหรับแต่ละเรคคอร์ด และประมาณ 8 เมกกะไบต์จาก 500,000 เรคคอร์ด เมื่อทำการเปรียบเทียบกับพื้นที่ที่ใช้ในการเก็บข้อมูล 500,000 เรคคอร์ดที่ใช้พื้นที่ประมาณ 200 เมกกะไบต์แล้ว การทำ snowflaking จะสามารถลดการใช้พื้นที่ได้เพียง 4% เท่านั้น ซึ่งเป็นจำนวนที่น้อยมากและไม่สามารถชดเชยได้กับข้อเสียของการทำ snowflaking เลยลองพิจารณาข้อดีและข้อจำกัดของการทำ snowflaking ดังต่อไปนี้

ข้อดีของการทำ snowflaking

- ลดพื้นที่จัดเก็บข้อมูลได้เล็กน้อย
- โครงสร้างข้อมูลที่มีการทำนอร์มอลไลซ์แล้วจะช่วยให้การอัพเดทและการดูแลรักษาข้อมูลทำได้โดยง่าย

ข้อเสียของการทำ snowflaking

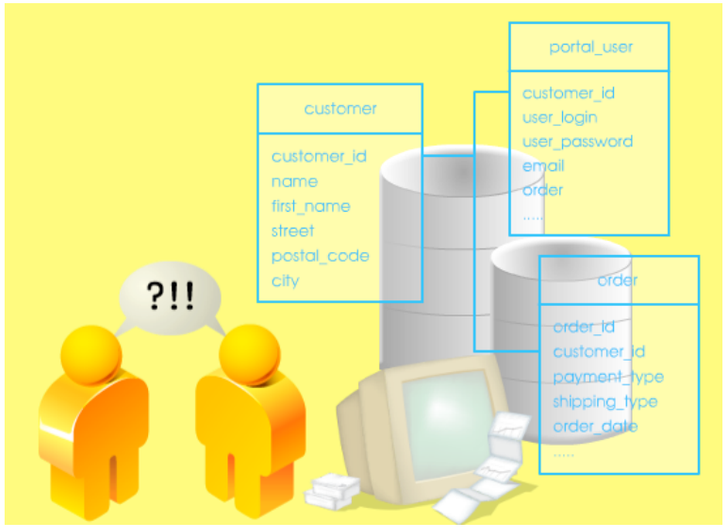
- ผู้ใช้อาจเกิดความสับสนหรือไม่เข้าใจกับโครงสร้างที่มีความซับซ้อนมากขึ้น
- ยากที่จะเรียกดูข้อมูลได้
- ลดทอนประสิทธิภาพของการทำคิวรีเนื่องจากต้องทำการ join คิวรีเพิ่มขึ้น

จากข้อดี-ข้อเสียข้างต้น เราจะทราบว่า snowflake schema นั้นอาจไม่เหมาะสำหรับการสร้างคลังข้อมูลแบบทั่วๆไป เนื่องจากการทำนอร์มอลไลซ์จะลดทอนประสิทธิภาพการทำคิวรี ซึ่งการทำคิวรีที่มีประสิทธิภาพเป็นสิ่งที่มีความสำคัญสูงสุดของการสร้างคลังข้อมูล

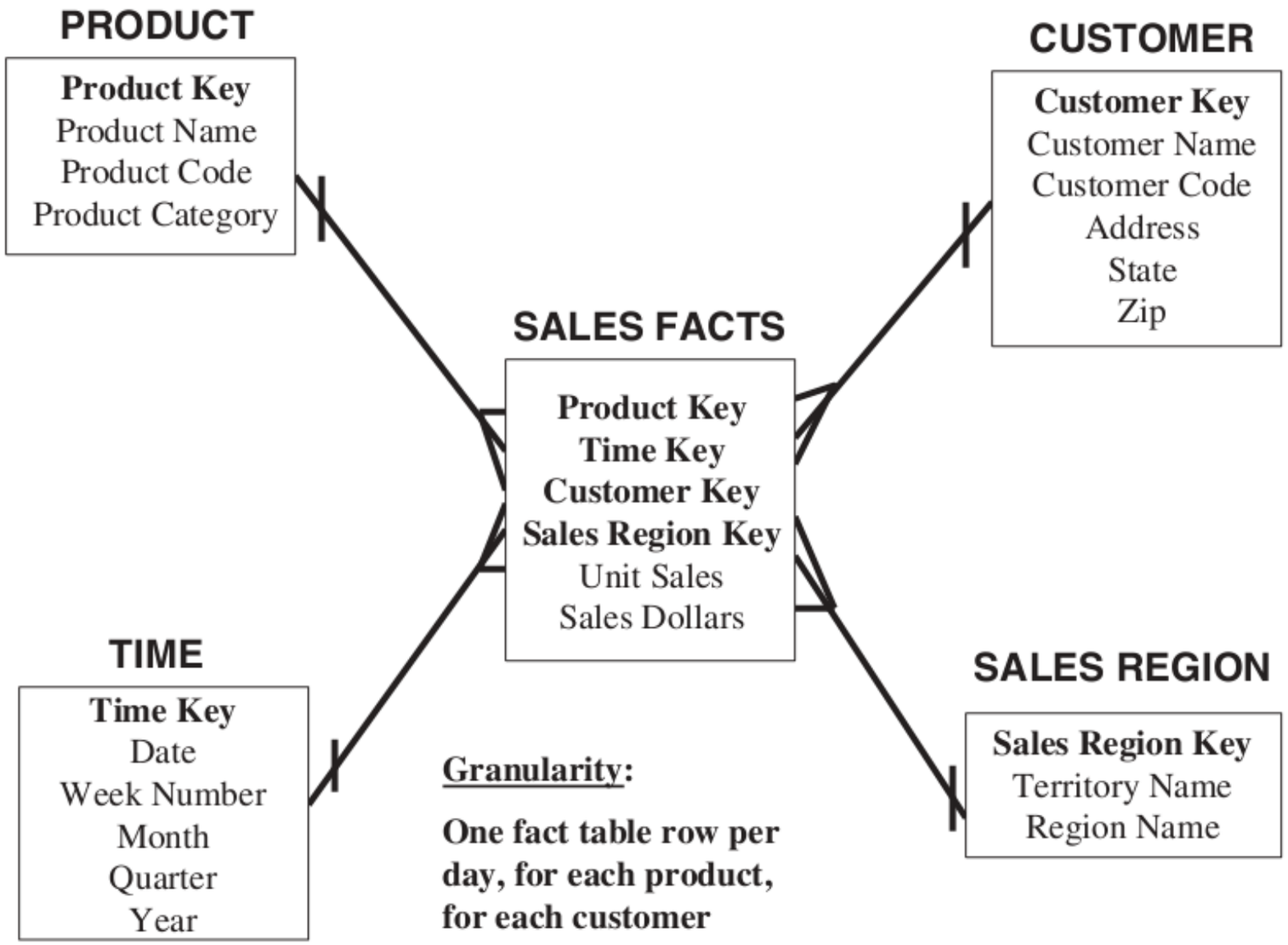
SECTION 5

การรวบยอดข้อมูลใน fact table

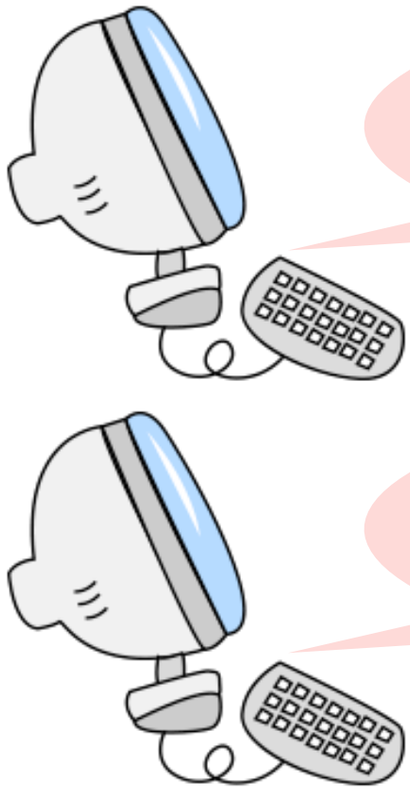
การรวบยอดข้อมูลใน fact table



นอกเหนือจากทำการนอร์มอลไลซ์ข้อมูลใน dimension table เพื่อลดการใช้พื้นที่สำหรับจัดเก็บข้อมูลแล้ว เรายังสามารถเพิ่มประสิทธิภาพให้กับคลังข้อมูลที่เรากำลังจะสร้างขึ้นด้วย**การรวบยอดข้อมูลใน*fact table (Aggregate fact tables)*** ที่เป็นการจัดเตรียมผลสรุปข้อมูลที่ได้จาก fact table ที่มีความละเอียดสูงที่สุด โดยทำการเก็บผลสรุปของข้อมูลจาก fact table ไปไว้ที่ fact table ใหม่ที่จะสามารถช่วยให้เราเข้าถึงข้อมูลที่มีความละเอียดน้อยกว่าของเดิมได้รวดเร็วยิ่งขึ้น ลองพิจารณา star schema ในรูปที่ 7-18 ซึ่งจะประกอบไปด้วย fact table เกี่ยวกับการขายสินค้าที่มีความละเอียดสูง (low level of granularity) และประกอบไปด้วย 4 dimension table ได้แก่ รายการสินค้า ข้อมูลลูกค้า เวลา และพื้นที่ของการขายสินค้า จาก fact table ที่มีข้อมูลที่มีความละเอียดเราสามารถจัดเตรียมผลสรุปของข้อมูลโดยใช้ฟังก์ชันพื้นฐานทางคณิตศาสตร์ เช่น การรวมกันของเรคคอร์ดใน fact table การหาค่าเฉลี่ยของตัวชี้วัดจาก fact table และอื่น ๆ แต่ก่อนที่จะเริ่มทำการจัดเตรียมผลสรุปของข้อมูลใน fact table เราจะต้องพิจารณาถึงคิวรีจากผู้ใช้งานว่ามีความต้องการข้อมูลในลักษณะใด โดยตัวอย่างของคิวรีในการสืบค้นข้อมูลจะแสดงดังนี้

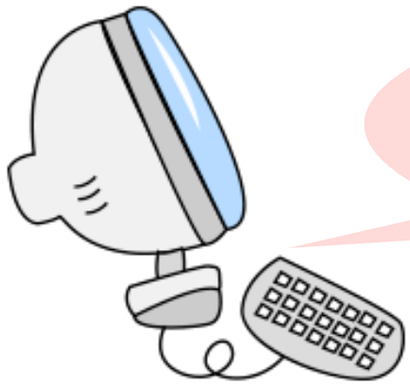


รูปที่ 7-18 ตัวอย่าง Star schema ของการขายสินค้าที่มีความละเอียดสูงสุด



ควิรีที่ 1 ต้องการสืบค้นข้อมูลยอดการซื้อสินค้าชนิด Widget-1 ของลูกค้าที่มีรหัส 12345678 ในสัปดาห์แรกของเดือนธันวาคมปี 2011

ควิรีที่ 2 ต้องการสืบค้นข้อมูลยอดการซื้อสินค้าชนิด Widget-1 ของลูกค้าที่มีรหัส 12345678 ในช่วง 3 เดือนแรกของปี 2012



ควิรีที่ 3 ต้องการสืบค้นข้อมูลยอดขายสินค้าประเภท Bigtools จากลูกค้าที่อยู่ในเขต south-central ใน 6 เดือนแรกของปี 2012

จากคิวรีทั้งสามข้างต้น เราจะต้องทำการหาผลสรุปของข้อมูลเพื่อคืนผลลัพธ์ให้แต่ละคิวรีดังนี้

การทำงานเพื่อตอบคิวรีที่ 1 ทุกเรคคอร์ดใน fact table ที่มี customer key = 12345678, product key = Widget-1 และวันที่อยู่ในช่วงวันที่ 1-7 เดือน = 12/December ปี = 2011 จะถูกอ่านเพื่อจัดเตรียมเป็นผลสรุปของข้อมูล สมมติว่าถ้าลูกค้ารหัส 12345678 มีการซื้อ Widget-1 ทุกวัน เราจะต้องทำการสร้างผลสรุปของข้อมูลจากข้อมูลทั้งสิ้น 7 เรคคอร์ดด้วยกัน

คิวรีที่ 1

การทำงานเพื่อตอบคิวรีที่ 2 ทุกเรคคอร์ดใน fact table ที่มี customer key = 12345678, product key = Widget-1 และเวลาอยู่ในช่วง 90 วันแรกของปี 2012 จะถูกอ่านเพื่อจัดเตรียมเป็นผลสรุปของข้อมูลสมมติว่าถ้าลูกค้ารหัส 12345678 มีการซื้อ Widget-1 ทุกวัน เราจะต้องทำการสร้างผลสรุปของข้อมูลจากข้อมูลทั้งสิ้น 90 เรคคอร์ดด้วยกัน

คิวรีที่ 2

การทำงานเพื่อตอบคิวรีที่ 3 ทุกเรคคอร์ดใน fact table ที่มี sale region = “Sound-central”, product category = “Bigtools”และ เวลาอยู่ในช่วง 180 วันแรกของปี 2012 จะถูกอ่านเพื่อจัดเตรียมเป็นผลสรุปของข้อมูล ในกรณีนี้จะต้องมีเรคคอร์ดเป็นจำนวนมากที่เกี่ยวข้องกับการทำผลสรุปของข้อมูล

คิวรีที่ 3

จากการทำงานทั้ง 3 การทำงานข้างต้น จะเห็นว่าการทำงานเพื่อตอบคิวรีที่ 3 จะ**ใช้เวลาค่อนข้างมาก** เนื่องจากต้องทำการรวมผลยอดขายจากเรคคอร์ดเป็นจำนวนมาก ดังนั้นเราควรจะทำการรวบยอดข้อมูลจาก fact table เพื่อช่วยลดเวลาในการสืบค้นข้อมูล แต่ก่อนที่จะทำการรวบยอดข้อมูลเราควรจะพิจารณาถึงจำนวนเรคคอร์ดทั้งหมดที่เก็บอยู่ใน fact table ก่อนเป็นลำดับแรก ว่ามีปริมาณมากน้อยเพียงใดเราสมควรจะทำการรวบยอดข้อมูลหรือไม่

ลองพิจารณา star schema ในรูปที่ 7-19 ที่แสดงยอดขายของซุปเปอร์มาร์เก็ต ที่ซึ่ง fact table มีข้อมูลที่มีรายละเอียดสูง จากข้อมูลใน *fact* และ *dimension tables* เราจะสามารถคำนวนจำนวนเรคคอร์ดสูงสุดของ *fact table* ได้ ถ้าเราทราบถึงจำนวนเรคคอร์ดในทุก ๆ *dimension table* โดยสามารถคำนวณได้ดังนี้

- Time dimension table มีการเก็บข้อมูล 5 ปี ปีละ 365 วัน จึงทำให้มีข้อมูลใน time dimension ทั้งหมด 1,825 เรคคอร์ด
- Store dimension table ประกอบไปด้วย 300 สาขาที่มีการขายสินค้าในแต่ละวัน
- Product dimension table จะมีรายการสินค้าทั้งหมด 40,000 รายการในแต่ละสาขา โดยที่จะมีการซื้อสินค้าในแต่ละวันและแต่ละสาขาอยู่ที่ประมาณ 4,000 รายการ
- Promotion dimension table การขายสินค้าจะมีการขายแบบมีโปรโมชั่นหรือไม่มีเท่านั้น

จากข้อมูลในทุก dimension จาก star schema เราจะสามารถคำนวณจำนวณเรคคอร์ดที่มากสุดของ fact table ได้เท่ากับ 1,825 × 300 × 4,000 × 1 ≅ 2 พันล้านเรคคอร์ด

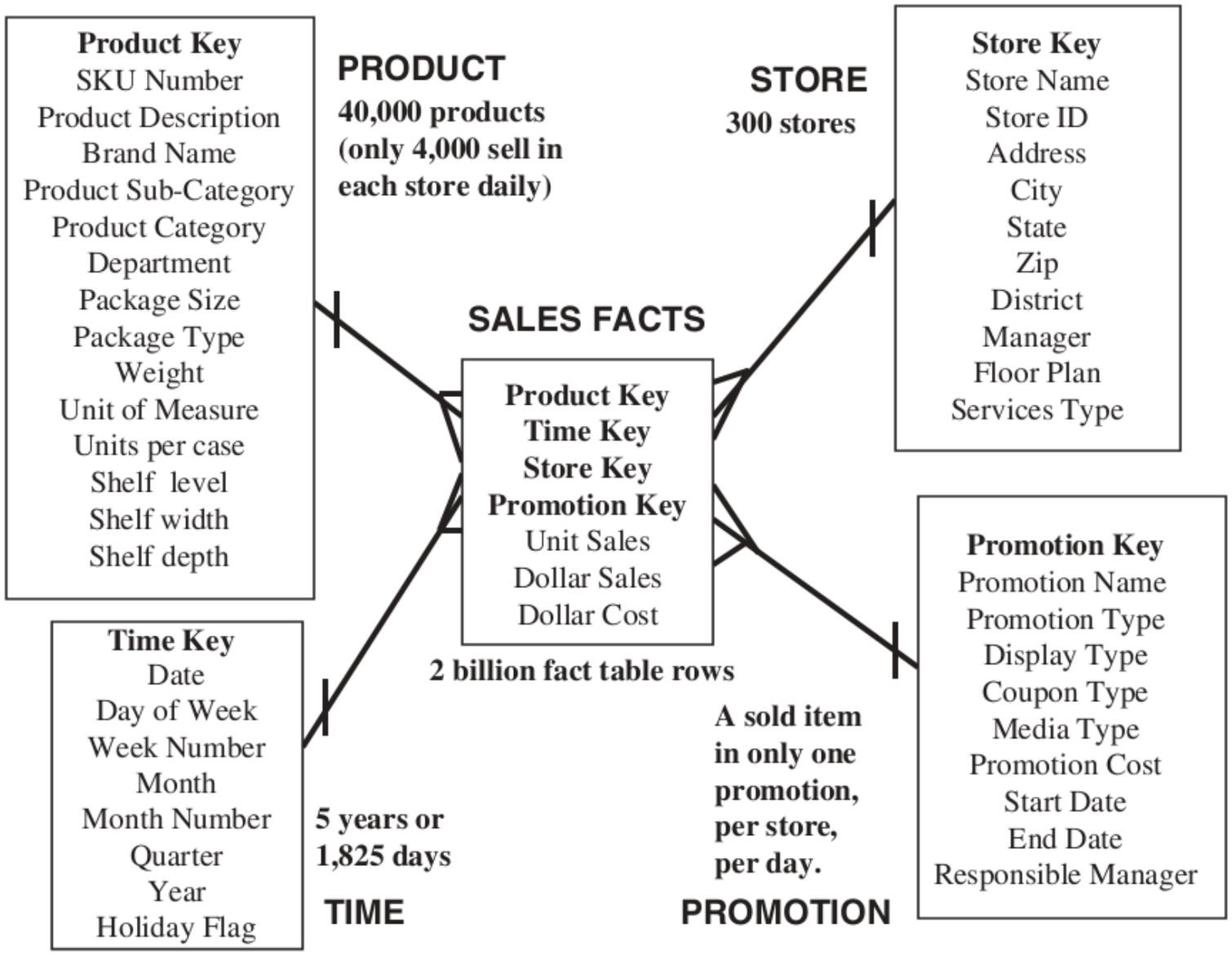
ในส่วนของกรณีอื่น ๆ เราจะสามารถคาดคะเนขนาดของ fact table ได้ดังนี้

การตรวจสอบโทรศัพท์

- Time dimension ทำการเก็บข้อมูลทั้งหมด 5 ปี ซึ่งเท่ากับ 1,825 วัน
- จำนวนครั้งของการโทรศัพท์ที่ตรวจสอบได้ในแต่ละวันอยู่ที่ประมาณ 150 ล้านครั้ง
- จำนวนเรคคอร์ดของ fact table ที่มากที่สุดสามารถคำนวณได้เป็น 274 พันล้านเรค คอร์ด

การตรวจสอบการใช้บัตรเครดิต

- Time dimension ทำการเก็บข้อมูลทั้งหมด 5 ปี ซึ่งเท่ากับ 1,825 วัน
- จำนวนบัญชีที่มีการใช้บัตรเครดิตเท่ากับ 150 ล้านบัญชี
- จำนวนการใช้บัตรเครดิตในรอบเดือนโดยเฉลี่ยของแต่ละบัญชีเท่ากับ 20
- จำนวนเรคคอร์ดของ fact table ที่มากที่สุดสามารถคำนวณได้เป็น 548 พันล้านเรค คอร์ด



รูปที่ 7-19 ตัวอย่าง star schema การขายสินค้าสำหรับธุรกิจค้าปลีก

จากตัวอย่างข้างต้น เราจะเห็นว่าเมื่อ fact table เก็บข้อมูลที่มีความละเอียดสูงจะทำให้มีจำนวนเรคคอร์ดค่อนข้างมาก ซึ่งโดยส่วนใหญ่ของการใช้งานคลังข้อมูลผู้ใช้จะไม่ทำคิวรีเพื่อเรียกดูข้อมูลเพียงเรคคอร์ดเดียวจาก fact table แต่อย่างไรก็ตาม เรายังคงต้องทำการเก็บข้อมูลที่มีความละเอียดสูงไว้ เนื่องจากเมื่อผู้ใช้ต้องการวิเคราะห์ข้อมูล ที่ซึ่งผลของการวิเคราะห์จะมาจากการรวมกันของข้อมูลในแต่ละแถวของ fact table ถ้าเราไม่เก็บข้อมูลที่มีความละเอียดสูงจะทำให้คลังข้อมูลไม่สามารถคืนผลลัพธ์ได้ตรงกับความต้องการนั้น ๆ ได้ เช่น ถ้าเราไม่เก็บรายละเอียดของแต่ละสาขาของซุปเปอร์ มาร์เก็ตจะทำให้เราไม่สามารถเรียกดูข้อมูลยอดขายของแต่ละรายการสินค้าที่ขายได้ในแต่ละสาขาได้ หรือถ้าเราไม่เก็บรายละเอียดของแต่ละรายการสินค้าจะทำให้เราไม่สามารถเรียกดูข้อมูลการขายของแต่ละสาขาว่ามีการขายสินค้าประเภทใดบ้าง

เมื่อเราทำการเก็บข้อมูลที่มีความละเอียดสูง เราจะสามารถทำการรวมผลลัพธ์จากข้อมูลหลายๆแถวใน fact table เพื่อคืนผลลัพธ์ให้กับคิวรีจากผู้ใช้ได้อย่างไร ลองพิจารณาคิวรีที่ต้องการการรวบยอดข้อมูลดังต่อไปนี้

- ยอดขาย 3 เดือนล่าสุดของสาขาใหม่ 3 สาขา ที่เพิ่งก่อตั้งขึ้นใน Wisconsin เปรียบเทียบกับยอดขายโดยเฉลี่ยของสาขาทั้งประเทศเป็นอย่างไร

- กลยุทธ์ทางการขายกับเนื้อสัตว์และสัตว์ปีกที่จัดทำขึ้นเมื่อวันหยุดที่ผ่านมามีผลต่อยอดขายอย่างไรบ้าง

- ยอดขายสินค้าแต่ละประเภทในวันหยุดที่ 4 กรกฎาคมที่ผ่านมา เปรียบเทียบกับยอดขายในวันเดียวกันของปีที่แล้วเป็นอย่างไร

จากตัวอย่างคิวรีทั้ง 3 ข้างต้น แต่ละคิวรีจะต้องทำการเลือกเรคคอร์ดที่เกี่ยวข้องกับคิวรีนั้น ๆ จาก fact table จากนั้นทำการรวบยอดข้อมูลจากเรคคอร์ดที่เลือกไว้เพื่อคืนค่าผลลัพธ์ให้กับแต่ละคิวรี ในการที่จะรวบยอดข้อมูลเราจะต้องทำการเก็บข้อมูลที่มีรายละเอียดสูงในหลาย ๆ dimension ยกตัวอย่างเช่น ในการตอบคิวรีที่ 3 เราต้องการข้อมูลที่มีความละเอียดในแกนเวลา (ความละเอียดที่ต้องทำการจัดเก็บคือ ยอดขายในแต่ละวัน) แต่ต้องการผลลรุปของยอดขายของสินค้าแต่ละประเภท ในกรณีนี้ ถ้าเราทำการหาผลสรุป/รวบยอด ข้อมูลไว้ก่อนหน้า (โดยทำการเก็บผลสรุปไว้ในอีก fact table หนึ่ง) จะช่วยให้สามารถตอบคิวรีได้เร็วขึ้น

ดังนั้นในการออกแบบ star schema เราจะต้องพิจารณาถึงการรวบยอดข้อมูลที่เหมาะสม เพื่อช่วยเพิ่มประสิทธิภาพของการตอบคิวรี ซึ่งเป็นหัวใจหลักของการทำงานของคลังข้อมูล

ความต้องการในการรวบยอดข้อมูลใน fact table

หลังจากที่เราทราบถึงประโยชน์ของการรวบยอดข้อมูลใน fact table แล้ว เราลองพิจารณาว่าเมื่อไหร่ที่เราควรจะทำการรวบยอดข้อมูล หรือคิวรีลักษณะใดที่ต้องการรวบยอดข้อมูลบ้าง เพื่อให้เข้าใจหรือเห็นภาพมากขึ้น ลองพิจารณารูปที่ 7-19 ที่จะทำให้ทราบถึงความต้องการในการรวบยอดข้อมูล โดยในรูปจะแสดงถึง star schema สำหรับธุรกิจค้าปลีกที่มีอยู่ 300 สาขา ซึ่งมีรายการสินค้าทั้งหมด 40,000 รายการสินค้าจากผู้ผลิต 500 ราย (500 brands) สมมติว่าสินค้าแต่ละรายการสินค้าจะถูกซื้ออย่างน้อย 1 ครั้งในแต่ละสาขา ซึ่งจากข้อมูลข้างต้นเราสามารถคำนวนจำนวนเรคคอร์ดของ fact table ที่เราต้องทำการเข้าถึงและทำการรวบยอดเพื่อคืนผลลัพธ์ให้กับคิวรีได้ ดังนี้

- คิวรีที่เกี่ยวข้องกับสินค้า 1 รายการที่ถูกขายในสาขาหนึ่งๆ ใน 1 สัปดาห์จะต้องทำการเข้าถึง/รวบยอดข้อมูลเพียง 1 เรคคอร์ดเท่านั้น
- คิวรีที่เกี่ยวข้องกับสินค้า 1 รายการที่ถูกขายในทุกสาขาใน 1 สัปดาห์ จะต้องทำการเข้าถึง/รวบยอดข้อมูล 300 เรคคอร์ด
- คิวรีที่เกี่ยวข้องกับการขายสินค้า 1 ยี่ห้อสินค้า ใน 1 สาขา ใน 1 สัปดาห์ จะต้องทำการเข้าถึง/รวบยอดข้อมูล 500 เรคคอร์ด
- คิวรีที่เกี่ยวข้องกับการขายสินค้า 1 ยี่ห้อสินค้า ในทุกสาขา ใน 1 ปี จะต้องทำการเข้าถึง/รวบยอดข้อมูลทั้งสิ้น 7,800,000 เรคคอร์ดด้วยกัน (= 500 products per band × 300 stores × 52 weeks)

สมมติว่าเราทำการคำนวณและสร้างตารางที่เก็บข้อมูลที่มีการรวบยอดไว้โดยที่แต่ละเรคคอร์ดของตารางจะเป็นผลสรุปของยอดขายยี่ห้อหนึ่งๆต่อ 1 สาขา ในรอบสัปดาห์หนึ่งๆ จะทำให้การตอบคิวรีที่ 3 จะทำการเข้าถึง/รวบยอดข้อมูลเพียง 1 เรคคอร์ดเท่านั้น และในการตอบคิวรีที่ 4 จะทำการรวบยอดข้อมูลเพียง 15,600 เรคคอร์ดเท่านั้น (300 storees× 52 weeks) แต่ถ้าเราทำการรวมยอดข้อมูลเพิ่มขึ้นไปอีก โดยทำผลสรุปของยอดขายของทุกๆยี่ห้อ ต่อ 1 สาขาในรอบปี จะทำให้ลดการจำนวนเรคคอร์ดที่ต้องทำการรวบยอดเหลือเพียง 300 เรคคอร์ดเท่านั้น

การสร้างตารางสำหรับข้อมูลที่ถูกรวบยอดมาจาก fact table หนึ่งๆจะเป็นตารางที่สร้างใหม่ที่จะใช้พื้นที่สำหรับเก็บข้อมูลน้อยกว่า fact table นั้นๆ เมื่อคิวรีจากผู้ใช้โดยส่วนใหญ่เกี่ยวข้องกับการถามถึงข้อมูลที่ถูกรวบยอดจะทำให้ประสิทธิการทำงานของคลังข้อมูลเพิ่มขึ้นเป็นอย่างมาก

วิธีในการรวบยอดข้อมูลใน fact table

ในการรวบยอดข้อมูลจาก fact table จะเป็นเพียงการสรุป/รวบรวมข้อมูลที่มีความละเอียดสูงไปยังระดับที่สูงขึ้นตามลำดับชั้นของ dimension ต่างๆ ในแบบจำลองมิติต่างๆ ดังแสดงตัวอย่างดังรูปที่ 7-20 ที่เป็นลำดับชั้นของข้อมูลจากระดับต่ำไปยังระดับสูง (จากความละเอียดมากไปยังความละเอียดน้อย) ของ product, store และ time dimension เมื่อเราทำการพิจารณาแต่ละ dimension โดยเริ่มจาก time dimension เราจะสามารถสรุป/รวบรวมข้อมูลจากแต่ละวันที่อยู่ในระดับล่างสุดไปเป็นระดับสูงสุดก็คือ 1 ปีได้ ในส่วนของ product dimension ที่มีข้อมูลระดับล่างสุดคือรายการสินค้าหนึ่งๆ ดังนั้นเราสามารถรวบยอดข้อมูลขึ้นไปยังระดับที่สูงขึ้นเป็นประเภทสินค้า/แผนกของสินค้าได้ และท้ายสุดคือ store dimension ที่ทำการเก็บข้อมูลชื่อสาขาไว้ในระดับล่างสุดและเก็บข้อมูลภูมิภาคไว้เป็นข้อมูลระดับสูงสุด เป็นต้น

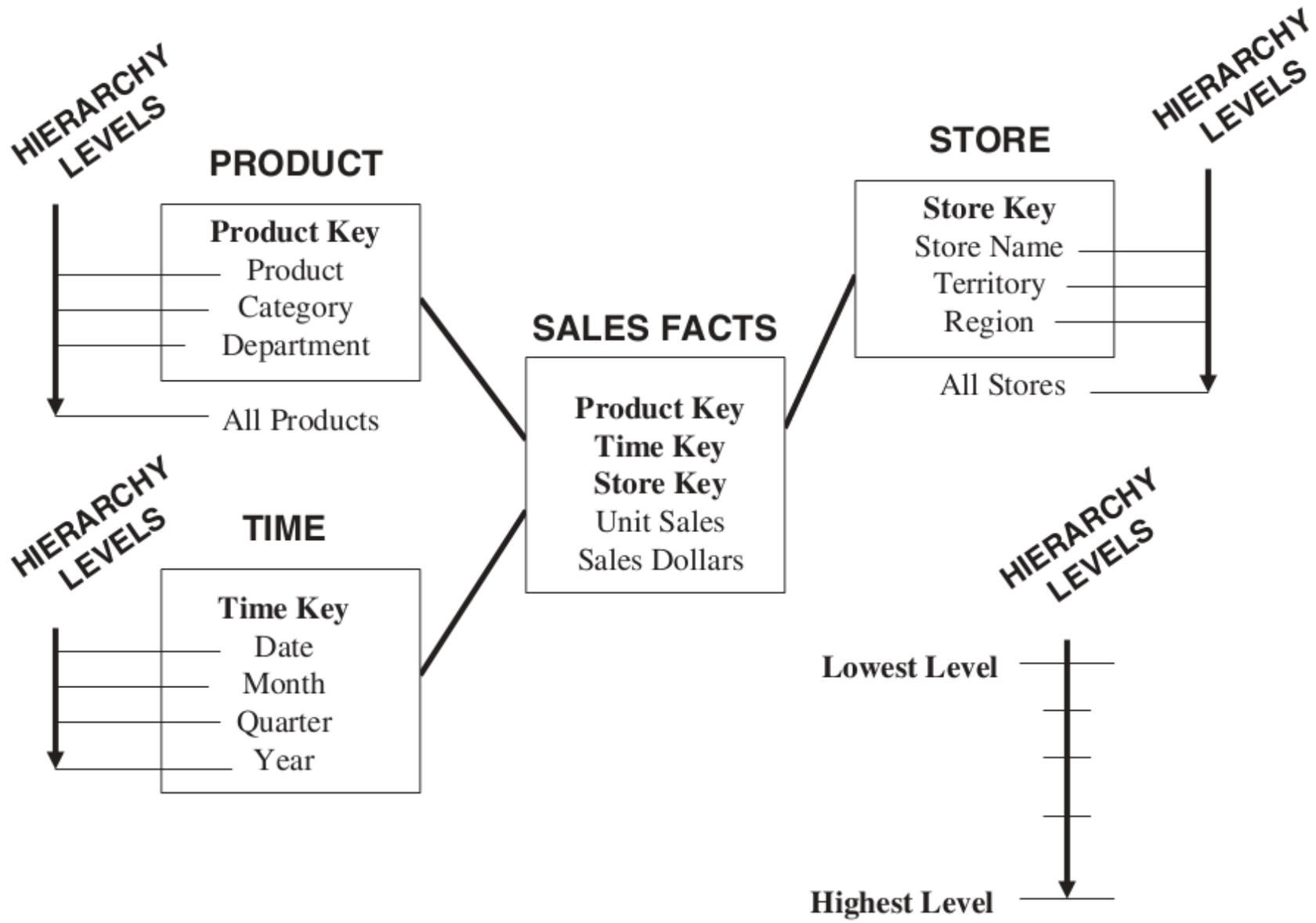
ข้อมูลแต่ละเรคคอร์ดจาก fact table (ในรูปที่ 7-20) จะประกอบด้วยข้อมูลระดับล่างสุดของแต่ละ dimension ตัวอย่างเช่น จำนวนชิ้นสินค้าและจำนวนเงินที่ขายได้ในหนึ่งวันต่อหนึ่งสาขาและต่อหนึ่งรายการสินค้า เมื่อเราทำการเลื่อนระดับของข้อมูลใน dimension หนึ่งๆ ให้สูงขึ้น เราจะสามารถสร้างตารางรวบยอดข้อมูลได้เป็นจำนวนมาก ลองพิจารณาความเป็นไปได้ในการรวบยอดข้อมูลดังแสดงในรูปที่ 7-21 ดังนี้

One-way aggregates

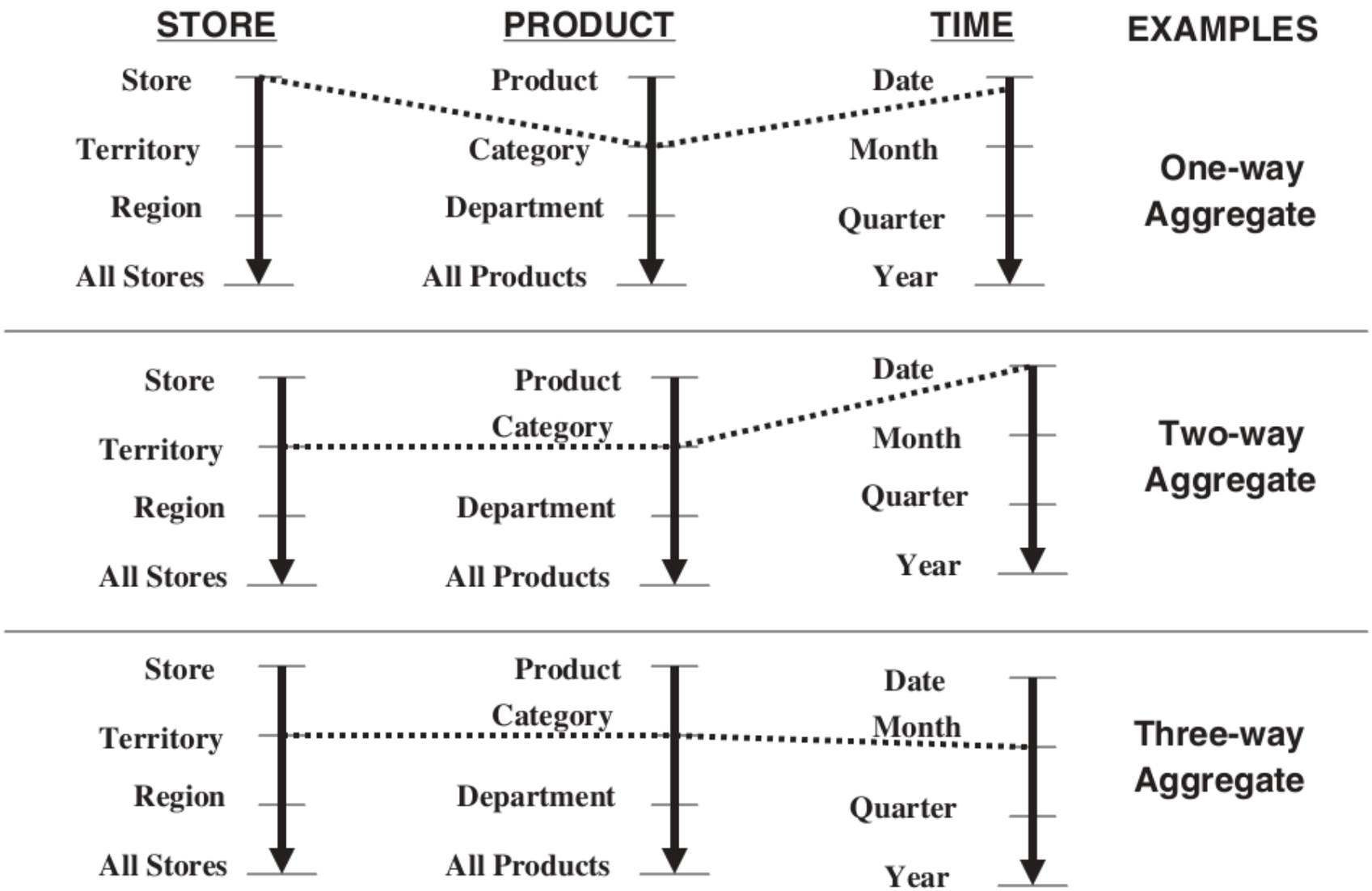
จะเป็นการปรับความละเอียดของข้อมูลใน dimension ใด dimension หนึ่งให้มีระดับสูงขึ้น (การปรับให้ข้อมูลมีความละเอียดลดลง) และปล่อยให้ dimension อื่นๆมีข้อมูลในระดับล่างสุดของลำดับชั้นความละเอียด จะทำให้เราจะสามารถสร้างตารางสำหรับการรวบยอดข้อมูลทางเดียว (One-way aggregate tables) ได้ ตัวอย่างเช่น

การปรับระดับข้อมูลใน product dimension ให้สูงขึ้น

- หมวดหมู่สินค้าต่อสาขาต่อวัน (Product category by store by date)
- แผนกของสินค้าต่อสาขาต่อวัน (Product department by store by date)
- ทุกรายการสินค้าต่อสาขาต่อวัน (All products by store by date)



รูปที่ 7-20 ลำดับชั้นของข้อมูลในแต่ละ dimension table



รูปที่ 7-21 วิธีการรวบยอดข้อมูลใน fact tables

การปรับระดับข้อมูลใน store dimension ให้สูงขึ้น

- อาณาเขตต่อรายการสินค้าต่อวัน (Territory by product by date)
- ภูมิภาคต่อรายการสินค้าต่อวัน (Region by product by date)
- ทุกสาขาต่อรายการสินค้าต่อวัน (All stores by product by date)

การปรับระดับข้อมูลใน time dimension ให้สูงขึ้น

- เดือนต่อสาขาต่อรายการสินค้า (Month by store by product)
- ไตรมาสต่อสาขาต่อรายการสินค้า (Quarter by store by product)
- ปีต่อสาขาต่อรายการสินค้า (Year by store by product)

Two-Way Aggregates

จะเป็นการปรับความละเอียดข้อมูลของ 2 dimensions ให้มีระดับสูงขึ้น (มีรายละเอียดลดลง) และปล่อยให้ dimension อื่นๆมีข้อมูลในระดับล่างสุดของลำดับชั้นความละเอียด จะทำให้เราจะสามารถสร้างตารางสำหรับการรวบยอดข้อมูลสองทาง (Two-way aggregate tables) ได้ ตัวอย่างเช่น

การปรับระดับข้อมูลใน product และ store dimensionให้สูงขึ้น

- หมวดหมู่สินค้าต่ออาณาเขตต่อวัน (Product category by territory by date)
- หมวดหมู่สินค้าต่อภูมิภาคต่อวัน (Product category by region by date)
- หมวดหมู่สินค้าต่อทุกสาขาต่อวัน (Product category by all stores by date)
- แผนกของสินค้าต่ออาณาเขตต่อวัน (Product department by territory by date)
- แผนกของสินค้าต่อภูมิภาคต่อวัน (Product department by region by date)
- แผนกของสินค้าต่อทุกสาขาต่อวัน (Product department by all stores by date)
- ทุกรายการสินค้าต่ออาณาเขตต่อวัน (All products by territory by date)
- ทุกรายการสินค้าต่อภูมิภาคต่อวัน (All products by region by date)
- ทุกรายการสินค้าต่อทุกสาขาต่อวัน (All products by all stores by date)

การปรับระดับข้อมูลใน product และ time dimensionให้สูงขึ้น

- หมวดหมู่สินค้าต่อเดือนต่อสาขา (Product category by month by store)
- หมวดหมู่สินค้าต่อไตรมาสต่อสาขา (Product category by quarter by store)
- หมวดหมู่สินค้าต่อปีต่อสาขา (Product category by year by store)
- แผนกของสินค้าต่อเดือนต่อสาขา (Product department by month by store)
- แผนกของสินค้าต่อไตรมาสต่อสาขา (Product department by quarter by store)
- แผนกของสินค้าต่อปีต่อสาขา (Product department by year by store)
- ทุกรายการสินค้าต่อเดือนต่อสาขา (All products by month by store)
- ทุกรายการสินค้าต่อต่อไตรมาสต่อสาขา (All products by quarter by store)
- ทุกรายการสินค้าต่อปีต่อสาขา (All products by year by store)

การปรับระดับข้อมูลใน storeและ time dimensionให้สูงขึ้น

- อาณาเขตต่อเดือนต่อรายการสินค้า (Territory by month byproduct)
- อาณาเขตต่อไตรมาสต่อรายการสินค้า (Territory by quarter byproduct)
- อาณาเขตต่อปีต่อรายการสินค้า (Territory by year byproduct)
- ภูมิภาคต่อเดือนต่อรายการสินค้า (Region by month byproduct)
- ภูมิภาคต่อไตรมาสต่อรายการสินค้า (Region by quarter byproduct)
- ภูมิภาคต่อปีต่อรายการสินค้า (Region by year byproduct)
- ทุกสาขาต่อเดือนต่อรายการสินค้า (All stores by month byproduct)
- ทุกสาขาต่อไตรมาสต่อรายการสินค้า (All stores by quarter byproduct)
- ทุกสาขาต่อปีต่อรายการสินค้า (All stores by year byproduct)

Three-Way aggregates

จะเป็นการปรับความละเอียดของข้อมูลจากทั้งหมด 3 dimensions ให้มีระดับสูงขึ้น (มีรายละเอียดลดลง) เราจะสามารถสร้างตารางสำหรับการรวบยอดข้อมูลสามทาง (Three-way aggregate tables) ได้ ตัวอย่างเช่น

- หมวดหมู่สินค้าต่ออาณาเขตต่อเดือน (Product category by territory by month)
- แผนกสินค้าต่ออาณาเขตต่อเดือน (Product department by territory by month)
- ทุกรายการสินค้าต่ออาณาเขตต่อเดือน (All products by territory by month)
- หมวดหมู่สินค้าต่อภูมิภาคต่อเดือน (Product category by region by month)
- แผนกสินค้าต่อภูมิภาคต่อเดือน (Product department by region by month)
- ทุกรายการสินค้าต่อภูมิภาคต่อเดือน (All products by region by month)
- หมวดหมู่สินค้าต่อทุกสาขาต่อเดือน (Product category by all stores by month)
- แผนกสินค้าต่อทุกสาขาต่อเดือน (Product department by all stores by month)
- หมวดหมู่สินค้าต่ออาณาเขตต่อไตรมาส (Product category by territory by quarter)
- แผนกสินค้าต่ออาณาเขตต่อไตรมาส (Product department by territory by quarter)

- ทุกรายการสินค้าต่ออาณาเขตต่อไตรมาส (All products by territory by quarter)
- หมวดหมู่สินค้าต่อภูมิภาคต่อไตรมาส (Product category by region by quarter)
- แผนกสินค้าต่อภูมิภาคต่อไตรมาส (Product department by region by quarter)
- ทุกรายการสินค้าต่อภูมิภาคต่อไตรมาส (All products by region by quarter)
- หมวดหมู่สินค้าต่อทุกสาขาต่อไตรมาส (Product category by all stores by quarter)
- แผนกสินค้าต่อทุกสาขาต่อไตรมาส (Product department by all stores by quarter)
- หมวดหมู่สินค้าต่ออาณาเขตต่อปี (Product category by territory by year)
- แผนกสินค้าต่ออาณาเขตต่อปี (Product department by territory by year)
- ทุกรายการสินค้าต่ออาณาเขตต่อปี (All products by territory by year)
- หมวดหมู่สินค้าต่อภูมิภาคต่อปี (Product category by region by year)
- แผนกสินค้าต่อภูมิภาคต่อปี (Product department by region by year)
- ทุกรายการสินค้าต่อภูมิภาคต่อปี (All products by region by year)
- หมวดหมู่สินค้าต่อทุกสาขาต่อปี (Product category by all stores by year)
- แผนกสินค้าต่อทุกสาขาต่อปี (Product department by all stores by year)
- ทุกรายการสินค้าต่อทุกสาขาต่อปี (All products by all stores by year)

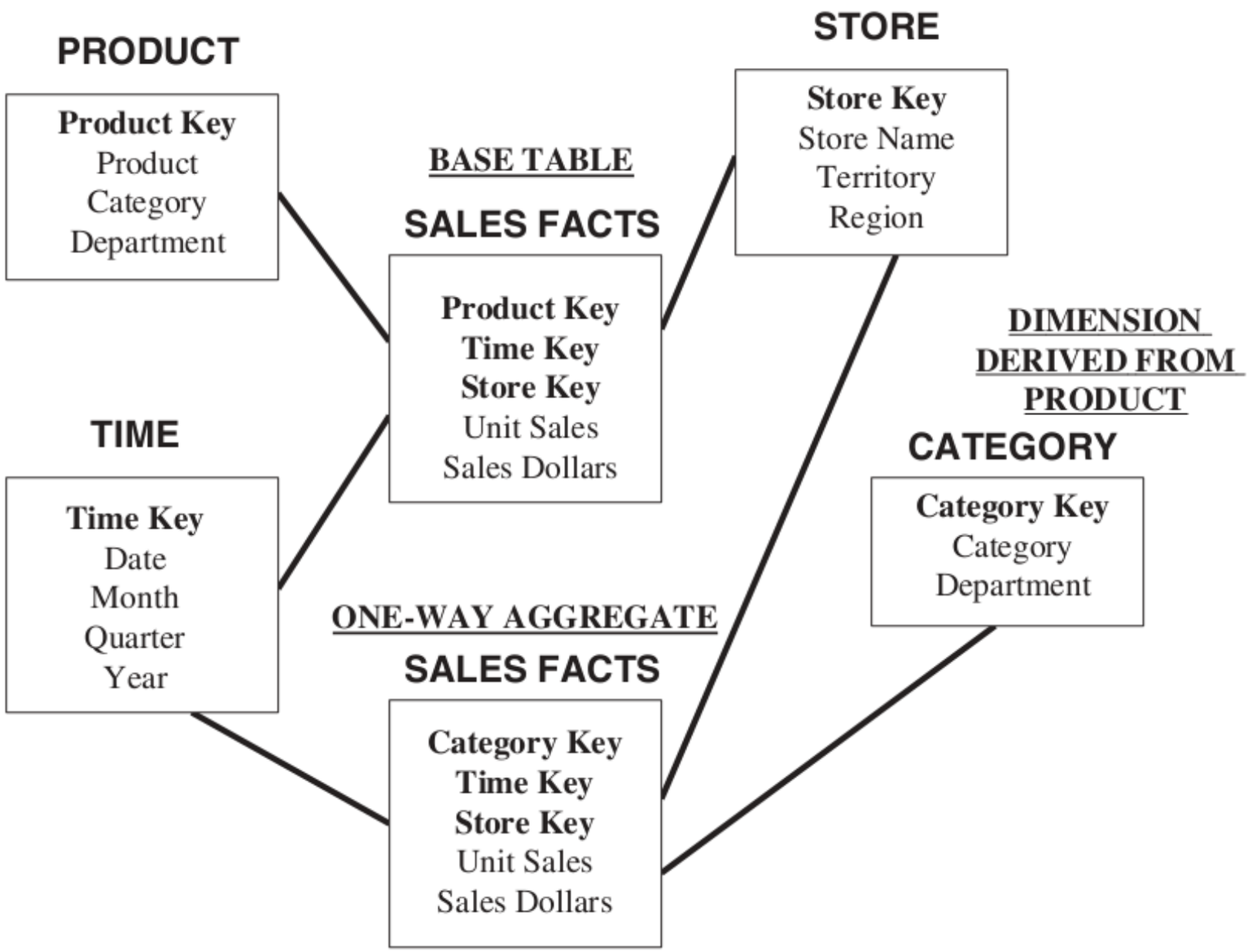
แต่ละตาราง fact table สำหรับการรวบยอดข้อมูลจะได้มากจาก fact table เดียว ซึ่งจะเกิดการรวมกันของ dimension table ที่มีข้อมูลลำดับชั้นสูงๆ เข้าด้วยกัน รูปที่ 7-22 แสดงถึงตาราง fact table สำหรับการรวบยอดข้อมูลที่มีการสร้างใหม่ โดยที่ตารางที่สร้างใหม่จะเกิดจากการทำ one-way aggregate กับ dimension รายการสินค้าที่มีการรวบยอดจากแต่ละรายการสินค้าเป็นประเภทสินค้า เป็นต้น ซึ่งจากรูปเราจะเห็นว่าเราต้องทำการสร้างตารางใหม่ถึง 2 ตารางด้วยกัน นั่นคือ 1) ตาราง category ที่ทำการรวบยอดข้อมูลจากรายการสินค้าไปเป็นประเภทสินค้า และ 2) ตาราง fact table ที่มีการรวบยอดข้อมูลยอดขายที่มีการเชื่อมต่อกับ dimension time store และ category ที่สร้างขึ้นใหม่

ลองพิจารณา กรณีการปรับความละเอียดของข้อมูลใน fact table ของคลังข้อมูลของธุรกิจค้าปลีกที่มี 300 สาขา โดยที่แต่ละสาขาจะมี 40,000 รายการสินค้าซึ่งสามารถจำแนกรายการสินค้าได้เป็น 500 รายการสินค้าต่อหนึ่งยี่ห้อสินค้า เมื่อทั้ง 300 สาขาเปิดทำการ ลองสามารถวิเคราะห์ได้ว่าจะมีเพียง 4,000 รายการสินค้าที่ขายได้ในแต่ละสาขาต่อวัน ถ้าเราทำการเก็บข้อมูลการขายไว้เป็นเวลา 5 ปี หรือ 1,825 วัน เราสามารถคำนวณเรคคอร์ดสูงสุดได้ 2,190,000,000 เรคคอร์ด (= 40,000 products × 300 stores × 1,825 days) แต่ด้วยเนื่องจากมีเพียง 4,000 รายการสินค้าที่ขายได้ในแต่ละวันทำให้จำนวนเรคคอร์ดใน fact table จะมีเพียง 10% จาก 2,190,000,000 เรคคอร์ดเท่านั้น เมื่อเราทำการสร้างตารางสำหรับรวบยอดข้อมูล โดยใช้ one-way aggregate กับ product dimension จะทำให้เราสามารถทราบถึงยอดขายต่อยี่ห้อหนึ่งๆ (เกิดจากการรวมกันของยอดขายสินค้า 500 รายการภายใต้ยี่ห้อนั้นๆ) ต่อสาขาต่อวัน

และเราจะสามารถคำนวณจำนวนเรคคอร์ดสูงสุดที่จะถูกเก็บไว้ใน fact table ที่จะทำการสร้างใหม่ได้เป็น 43,800,000 เรค คอร์ด (= 80brands *(= 40,000/500)*× 300 stores × 1,825 days) เมื่อเราทำการสร้าง one-way aggregate เราจะสังเกตุเห็นความเบาบาง (sparsity) ของข้อมูลที่ทำการรวบยอดว่าไม่ได้เป็นแค่ 10% เหมือนกับตาราง fact table

เนื่องจากเมื่อเราทำการรวบยอดรายการสินค้าเป็นยี่ห้อสินค้าจะมี brand code ที่เพิ่มขึ้นจากรหัสสินค้า (กล่าวคือสินค้าที่ถูกขาย 4,000 รายการอาจเป็นสินค้าที่มาจากหลากหลายนี่ห้อจึงทำให้มี brand code มากขึ้นซึ่งจะมากกว่า 10% ของ 80 ยี่ห้อสินค้าก็เป็นได้)

ดังนั้นความเบาบางของข้อมูลตารางสำหรับรวบยอดข้อมูลที่มีการทำ one-way aggregate อาจจะมีค่าประมาณ 50% ซึ่งจะทำให้มีข้อมูลประมาณ 21,900,000 เรคคอร์ด (50% ของ 43,800,000 เรคคอร์ด)



รูปที่ 7- 22 ตัวอย่างการสร้าง Aggregate fact table และ dimension table ใหม่

เมื่อเราทำให้ข้อมูลมีความละเอียดน้อยลง เปอร์เซ็นต์ความเบาบางของข้อมูลจะเพิ่มขึ้นใกล้เคียงกับ 100% แต่อย่างไรก็ตามการรวบยอดข้อมูลอาจเกิดความล้มเหลวในเชิงของความเบาบางของข้อมูลที่อาจมีความเบาบางมาก ทำให้เราอาจจะต้องเจอกับคำถามที่ว่า “การรวบยอดข้อมูลนั้นจะช่วยเพิ่มประสิทธิภาพการทำงานได้หรือไม่” หรือ “การรวบยอดข้อมูลจะลดจำนวนเรคคอร์ดที่ต้องทำการสืบค้นได้มากหรือไม่” ในการที่จะตอบคำถามเหล่านี้ ผู้ที่มีประสบการณ์ในการสร้างคลังข้อมูลได้ให้คำแนะนำไว้ว่า “เมื่อเราทำการรวบยอดข้อมูล เราจะต้องทำให้แน่ใจว่าในแต่ละตารางสำหรับรวบยอดข้อมูลจะต้องทำการสรุป/รวบรวมข้อมูลได้อย่างน้อย 10 เรคคอร์ดจาก fact table หรือ dimension table ถึงจะคุ้ม”จากข้อความ ดังกล่าวเราสามารถกล่าวได้อีกนัยหนึ่งว่าตารางสำหรับรวบยอดข้อมูลควรจะมีข้อมูลเพียงแค่ 10% จาก fact table แต่ถ้าเราสามารถสรุป/รวบรวมข้อมูลได้ 20 เรคคอร์ดหรือมากกว่านั้นจะเป็นสิ่งที่ยอดเยี่ยมมาก

เมื่อมองย้อนกลับไปกับวิธีการรวบยอดข้อมูลแบบ one-way, two-way และ three way aggregate สำหรับ star schema ที่มี 3 dimension เราจะเห็นว่าเราสามารถสร้างตารางรวบยอดข้อมูลได้ประมาณ 50 ตาราง แต่ในแอพพลิเคชันจริงจำนวน dimension ไม่ได้มีเพียงแค่ 3 dimension แต่จะมี dimension เป็นจำนวนมาก ซึ่งจะทำให้เราสามารถสร้างตารางรวบยอดข้อมูลได้เป็นจำนวนมาก ซึ่งจากการพิจารณาถึงความเบาบางข้อมูลที่เกี่ยวเนื่องกับการรวบยอดข้อมูลเราจะต้องพิจารณาถึงเปอร์เซ็นต์การลดลงของจำนวนเรคคอร์ดของตารางรวบยอดข้อมูลเปรียบเทียบกับจำนวนเรคคอร์ดของ fact table โดยแท้จริงแล้วเปอร์เซ็นต์ความเบาบางจะต้องเพิ่มขึ้นเมื่อทำการรวบยอดในระดับที่สูงขึ้น ดังนั้นก่อนที่เราจะทำการรวบยอดข้อมูลเราจะต้องพิจารณาถึงคำตอบของคำถามต่อไปนี้ การรวบยอดข้อมูลนั้นได้ผลดีเพียงใด? เรามีทางเลือกในการรวบยอดข้อมูลอย่างไรบ้าง? เราจะเลือกตาราง fact table ใดมาทำการสรุป/รวบรวมข้อมูล? จากคำถามเหล่านี้เราควรจะตั้งเป้าหมายสำหรับการรวบยอดข้อมูลก่อนเป็นลำดับแรก

เป้าหมายของการรวบยอดข้อมูลใน fact table

นอกเหนือจากเป้าหมายหลัก ๆ ที่ต้องการเพิ่มประสิทธิภาพของการใช้งานคลังข้อมูลแล้ว ยังมีเป้าหมายอื่น ๆ อีกมากมายดังนี้

- พยายามที่จะไม่ทำการรวบยอดข้อมูลมากจนเกินไป เนื่องจากในการรวบยอดแต่ละครั้งเราจะต้องสร้าง dimension ขึ้นใหม่เพื่อสนับสนุนการรวบยอดข้อมูล
- พยายามปิดบังการรวบยอดข้อมูลไม่ให้ผู้ใช้เห็น แต่จะต้องทำให้การเรียกดูข้อมูลจากคิวรีของผู้ใช้ทราบถึงการรวบยอดข้อมูลเพื่อที่จะได้เข้าถึงตารางสำหรับการรวบยอดข้อมูลได้โดยง่าย
- พยายามที่จะไม่ใช้พื้นที่ในการจัดเก็บข้อมูลสำหรับตารางสำหรับรวบยอดข้อมูลมากจนเกินไป โดยจะต้องพินิจพิเคราะห์เกี่ยวกับตารางสำหรับรวบยอดข้อมูลที่มีขนาดใหญ่ที่มีเปอร์เซ็นความเบาบางน้อยๆ
- พยายามที่จะลดผลกระทบของการประมวลผลข้อมูลการรวบยอดข้อมูลที่ staging area ให้น้อยที่สุดเท่าที่จะเป็นไปได้

จากสิ่งที่เราพิจารณาข้างต้น เราควรที่จะต้องทำการพินิจพิเคราะห์ถึงปัจจัยต่างๆ ให้ครบถ้วนก่อนที่เราจะทำการคำนวณใดๆ ที่เกี่ยวข้องกับการรวบยอดข้อมูล โดยที่สิ่งสำคัญสิ่งหนึ่งที่เราควรจะพิจารณาก็คือการกำหนดชนิดของการรวบยอดข้อมูลสำหรับคลังข้อมูล โดยเลือกชนิดหรือวิธีการจากเวลาที่ใช้ในการคืนผลลัพธ์ให้กับคิวรีทั่วๆ ไป เช่น

ผู้ใช้ต้องการ report ประเภทใด?

Report ที่ต้องการจะเกี่ยวข้องกับข้อมูลในระดับใด?

ยอดขายต่อสาขาหนึ่งๆ?

ยอดขายต่อเดือน?

ยอดขายต่อหมวดหมู่สินค้า?

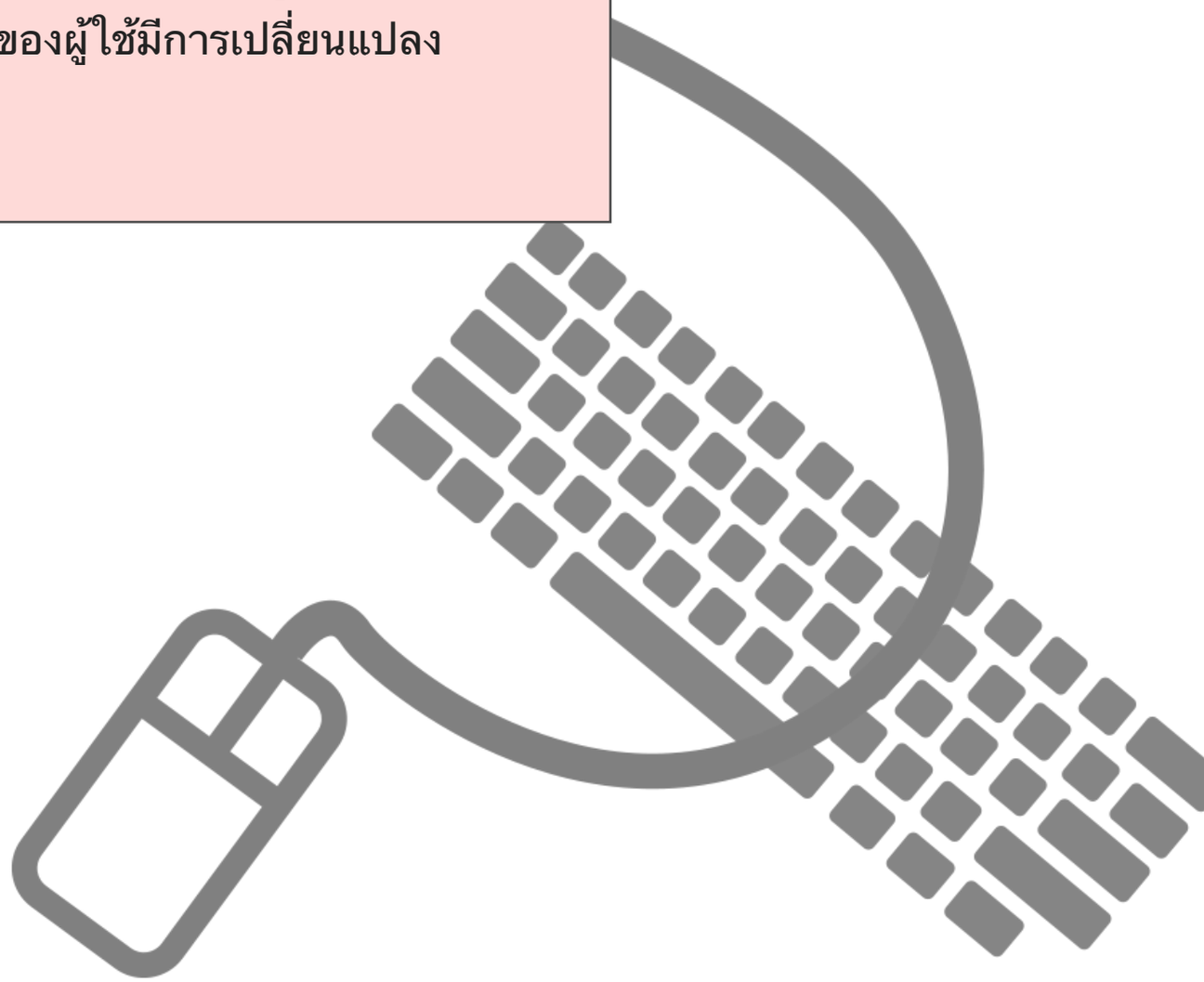
จากการพิจารณาดังกล่าวเราจะต้องทำการพิจารณาลำดับชั้นของข้อมูล/ความละเอียดของข้อมูลแต่ละ dimension โดยทำการตรวจสอบว่า dimension ที่พิจารณานั้นมีข้อมูลอยู่หลายลำดับชั้นหรือไม่ ถ้าข้อมูลมีหลายลำดับชั้น เราจะต้องหาว่าลำดับชั้นใดของข้อมูลที่มีความสำคัญ จากการตรวจสอบดังกล่าว เราจะได้แอทริบิวจำนวนหนึ่งที่ใช้สำหรับรวบยอดข้อมูลใน fact table ชั้นตอนต่อไปจะทำการเลือกแอทริบิวที่เหมาะสำหรับการรวบยอดข้อมูลต่อไป

ในการเลือกแอทริบิวที่เหมาะสมสำหรับการรวบยอดข้อมูลเราจะต้องพิจารณาที่จำนวนค่าที่เป็นไปได้ในแต่ละแอทริบิว ตัวอย่าง เช่น star schema ของธุรกิจโรงแรม จะมีข้อมูล “ชื่อโรงแรม” ประมาณ 25,000 โรงแรม (ข้อมูลระดับล่างสุด/ละเอียดมากสุด) และ “เมืองที่ตั้งโรงแรม” ประมาณ 15,000 เมือง (ข้อมูลระดับสูงขึ้น/ละเอียดน้อยกว่าชื่อโรงแรม)

จากตัวอย่างเราจะเห็นว่าจำนวนค่าที่เกิดขึ้นในสองแอทริบิวแตกต่างกันไม่มาก ดังนั้นการรวบยอดข้อมูลชื่อโรงแรมไปเป็นเมืองที่ตั้งโรงแรมอาจจะไม่ได้มีประโยชน์มากนัก แต่ถ้าโรงแรมทั้ง 25,000 โรงแรม ตั้งอยู่ในเมืองแค่ 500 เมือง จะทำให้ dimension ของโรงแรมมีความน่าสนใจในการรวบยอดข้อมูลเนื่องจากสามารถลดการจัดเก็บข้อมูลได้เป็นจำนวนมาก (โดยเฉลี่ยแล้ว 1 เมืองจะมี 500 โรงแรม) ดังนั้นในการเลือกแอทริบิวที่เหมาะสมเราจะต้องพิจารณาแอทริบิวทั้งหมดที่อยู่ในแกนลำดับชั้นเดียวกันของ dimension หนึ่งๆ แล้วทำการตรวจสอบจำนวนค่าที่เป็นไปได้ของแต่ละแอทริบิว รวมถึงเปรียบเทียบจำนวนค่าที่แตกต่างกัน แล้วทำการเลือกแอทริบิวมีความเหมาะสมสำหรับการรวบยอดข้อมูลสืบไป (ข้อสังเกตุ: แอทริบิวที่เหมาะสมสามารถมีได้มากกว่า 1 แอทริบิว)

ในการดำเนินการรวบยอดข้อมูลเราควรจะ<u>สร้างลิสต์ของแอทริบิวที่เหมาะสมสำหรับการรวบยอดข้อมูลไว้</u> จากนั้นดำเนินการรวบยอดข้อมูลจากแอทริบิวที่เลือกไว้โดยสร้างเป็น fact table ขึ้นใหม่ (ตารางรวบยอดข้อมูล) จากนั้นทำการพิจารณาถึง dimension table ที่จะต้องสร้างใหม่ที่สอดคล้องกับตารางรวบยอดข้อมูลที่จะทำการสร้างขึ้นใหม่ และทำการสร้างตารางสำหรับรวบยอดข้อมูลเป็นลำดับสุดท้าย

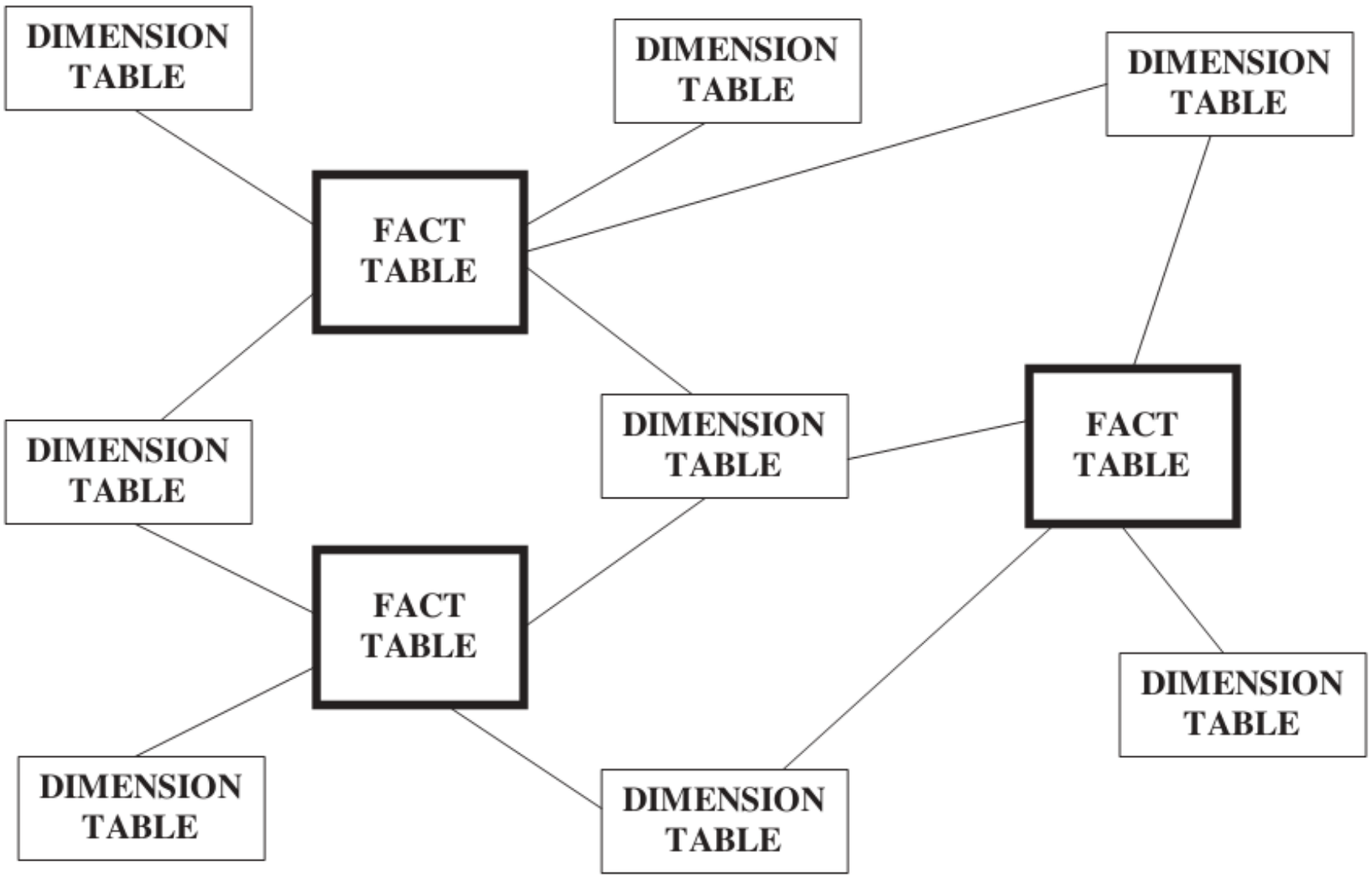
การรวบยอดข้อมูลจะเป็นการเพิ่มประสิทธิภาพการคืนผลลัพธ์ให้กับคิวรีจากผู้ใช้ ซึ่ง ในตอนเริ่มต้นของการรวบยอดข้อมูล เราอาจจะไม่ได้ทำการรวบยอดข้อมูลได้อย่างสมบูรณ์ตามพฤติกรรมของผู้ใช้ทุกส่วนงาน แต่เราก็ต้องทำการรวบยอดข้อมูลตามข้อมูลประเภท report ที่ผู้ใช้ต้องการที่ได้จาก business requirement ของขั้นตอนการสร้างคลังข้อมูลไปก่อน จากนั้นเราค่อยเฝ้าดูและปรับปรุงการรวบยอดข้อมูล เมื่อคิวรีของผู้ใช้มีการเปลี่ยนแปลง



SECTION 6

Family of stars

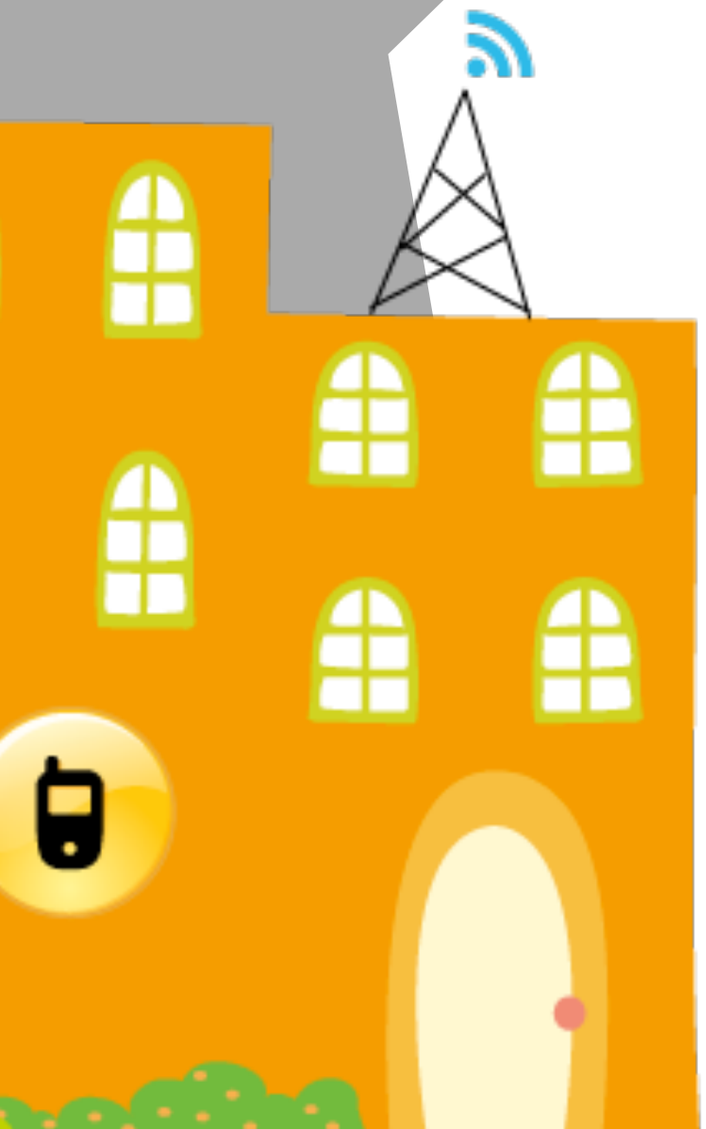
หลังจากได้เรียนรู้เกี่ยวกับแบบจำลองมิติต่าง ๆ ที่อยู่ในรูปแบบของ star และ snowflake schema รวมถึงการรวบยอดข้อมูลใน fact table แล้ว เมื่อเรามองที่ star schema เราจะพบว่า “star schema” หนึ่ง ๆ จะประกอบด้วย 1 fact table รายล้อมด้วย dimension table เป็นจำนวนมาก แต่ในปัจจุบันคลังข้อมูลจะประกอบไปด้วยหลาย star schema ด้วยกัน ซึ่งแต่ละ star schema จะถูกสร้างขึ้นเพื่อตอบสนองความต้องการที่จะติดตามการวัดผลของธุรกิจในแต่ละด้าน เมื่อเรามีหลายๆ star schema ที่รวมเข้าด้วยกันเราจะสามารถเรียกได้ว่า “a ***Family of STARS***” โดยที่ family of stars หนึ่ง ๆ จะเกิดจากการเพิ่มตารางรวบยอดข้อมูล และ dimension table ที่มีการปรับลดรายละเอียดเข้าไปใน star schema หนึ่งๆก็ได้ หรือจะสร้างตาราง fact table ขึ้นใหม่ที่มีข้อมูลที่เกี่ยวเนื่องกับความต้องการทางธุรกิจอื่น ๆ ก็เป็นได้ ตัวอย่างของ family of star จะแสดงดังรูปที่ 7-23 ซึ่งจากรูปเราจะเห็นว่า family of stars จะประกอบไปด้วยหลาย fact table ที่เกี่ยวกับ dimension table เดียวกัน ยกตัวอย่างเช่น time dimension จะเกี่ยวกับเนื่องกับทุก fact table ซึ่งอาจจะวาง time dimension ไว้ตรงกลางดังรูป



รูปที่ 7- 23 ตัวอย่าง Family of STARS

ถ้าเรากำลังออกแบบ star schema สำหรับธรุกิจธนาคารและผู้ให้บริการทางโทรศัพท์ เราจะต้องเก็บข้อมูลการทำรายการแต่ละรายการเพื่อป้องกันความผิดพลาด (capture individual transactions) และทำการสร้างผลสรุปรวมยอดการใช้บริการในช่วงเวลาที่กำหนด (snapshots at specific intervals) จากความต้องการดังกล่าว เราสามารถใช้ family of stars ในการเก็บข้อมูลแต่ละรายการและการสร้าง snapshots schema สำหรับธุรกิจที่ทำการผลิตสินค้าออกจำหน่าย เราจำเป็นต้องทำการเฝ้าดูตัวชี้วัดตามห่วงโซ่มูลค่า เป็นต้น ซึ่งจากธุรกิจต่าง ๆ ที่กล่าวมาข้างต้นการใช้ family of stars จะช่วยสนับการเรียกดูข้อมูลที่เป็นแบบ value chain และ value circle โดยที่เราจะสามารถออกแบบ family of stars ได้ดังนี้

Snapshot and Transaction Tables

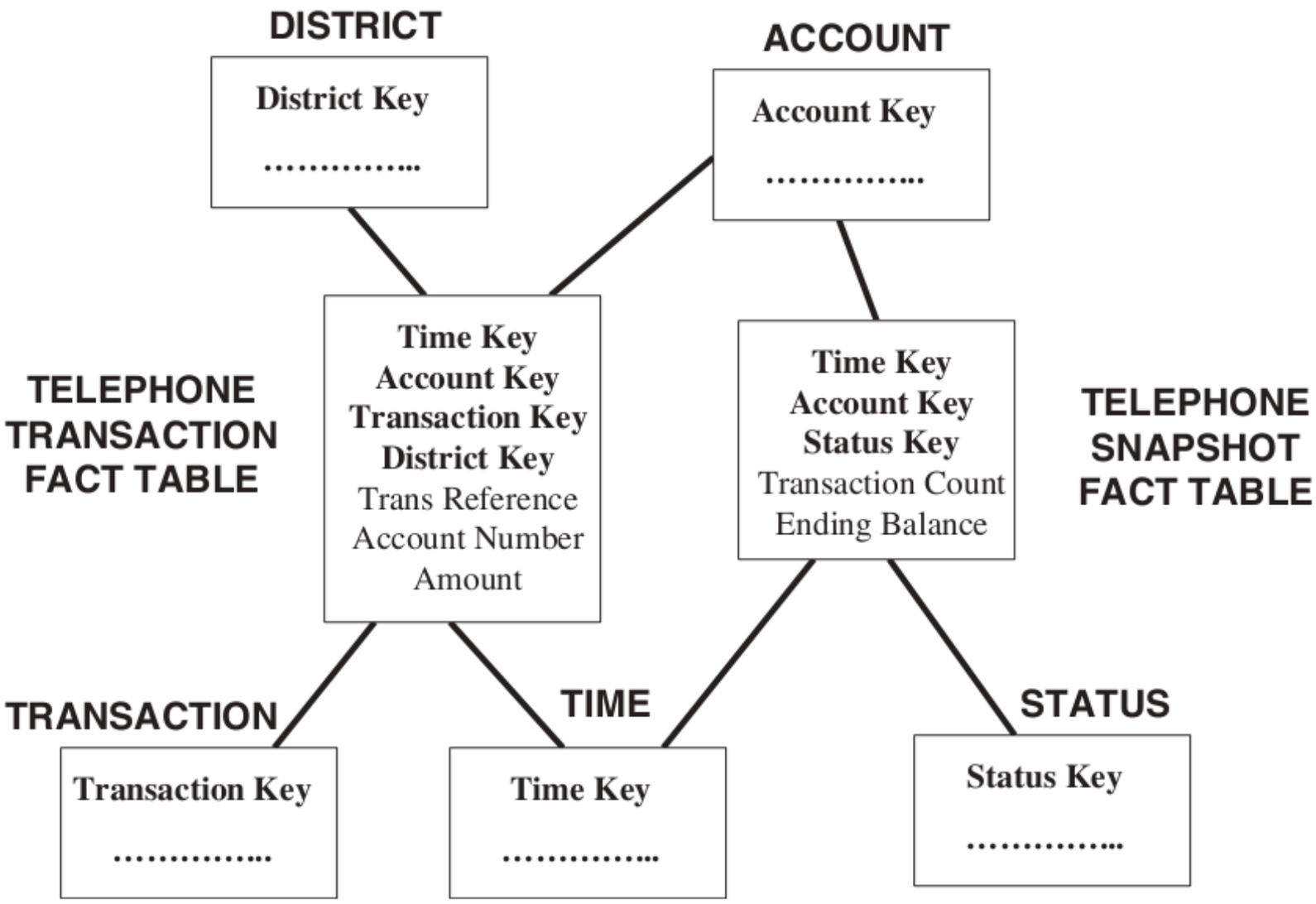


ลองพิจารณาความต้องการพื้นฐานของบริษัทผู้ให้บริการเครือข่ายโทรศัพท์เคลื่อนที่ ซึ่งแต่ละรายการข้อมูลที่เก็บจะเกี่ยวข้องกับการใช้โทรศัพท์ของลูกค้าแต่ละราย ซึ่งโดยส่วนใหญ่แล้ว รายการของการใช้โทรศัพท์จะอยู่ในช่วง 6.00 น. ถึง 22.00 น. และจะมีการใช้โทรศัพท์เพิ่มขึ้นในช่วงวันหยุด และช่วงสุดสัปดาห์ แต่สำหรับองค์กรต่างๆ จะมีการใช้โทรศัพท์ในช่วงวันธรรมดามากกว่าช่วงสุดสัปดาห์ จากพฤติกรรมการใช้ของลูกค้าหลายกลุ่มที่กล่าวข้างต้น บริษัทผู้ให้บริการเครือข่ายโทรศัพท์จะทำการรวบรวมข้อมูลการใช้โทรศัพท์ทั้งหมดของลูกค้าเพื่อมาวิเคราะห์ในแง่มุมต่างๆ

ดังนั้นบริษัทผู้ให้บริการเครือข่ายโทรศัพท์เคลื่อนที่จะต้องการ schema (transaction schema) สำหรับจัดเก็บข้อมูลการใช้โทรศัพท์แต่ละรายการเพื่อสนับสนุนการตัดสินใจในเชิงกลยุทธ์ เช่น การขยายกิจการหรือการขยายเครือข่ายสัญญาณตามที่ต่างๆ การปรับปรุงคุณภาพการให้บริการ และอื่นๆ โดยที่ schema ที่สร้างขึ้นจะช่วยในการตอบคำถามต่างๆ เช่น รายได้ของการใช้โทรศัพท์ในชั่วโมงเร่งด่วนในช่วงวันหยุดหรือช่วงสุปดาห์จะเป็นอย่างไร เมื่อเปรียบเทียบกับรายได้ของการใช้โทรศัพท์ในชั่วโมงเร่งด่วนของวันธรรมดาเป็นต้น

นอกจากคำถามข้างต้น บริษัทยังคงจำเป็นต้องตอบคำถามเกี่ยวกับยอดการใช้โทรศัพท์ให้กับลูกค้าอีกด้วย ในบางช่วงเวลา แผนกบัญชีอาจจะต้องการข้อมูล“การคาดการณ์การใช้โทรศัพท์ของลูกค้าในช่วงเวลากลางเดือนหน้า” หรือต้องการข้อมูลที่บอกเกี่ยวกับ “ลูกค้าใดมียอดการใช้โทรศัพท์สูงในรอบสิ้นเดือนบ้าง”

ซึ่งจากความต้องการดังกล่าวเราจะต้องทำการสร้าง schema (snapshot schema) เพื่อเก็บข้อมูลในช่วงเวลาหนึ่งๆ ดังแสดงในรูปที่ 7-24 ที่แสดงถึง snapshot และ transaction tables สำหรับบริษัทผู้ให้บริการเครือข่ายโทรศัพท์ โดยที่ transaction table จะเก็บข้อมูลการใช้โทรศัพท์แต่ละรายการ และในส่วนของ snapshot table จะเก็บข้อมูลการใช้โทรศัพท์ของแต่ละลูกค้าในช่วงเวลาที่กำหนด ซึ่งจาก fact table ทั้งสองลักษณะในรูปจะมีการใช้ dimension



รูปที่ 7- 24 ตัวอย่าง snapshot and transaction tables

ในส่วนของธุรกิจธนาคาร snapshot และ transaction tables สำหรับธุรกิจธนาคารจะค่อนข้างเหมือนกัน ตัวอย่างเช่น transaction table ของ ATM จะเก็บข้อมูลแต่ละครั้งของการทำธุรกรรมทางการเงินที่กระทำกับ ATM ซึ่ง fact table จะทำการเฝ้าดูและติดตามปริมาณการใช้ ATM ของลูกค้าแต่ละราย ในส่วนของ snapshot table จะเก็บข้อมูลยอดคงเหลือสำหรับแต่ละบัญชีในตอนท้ายของแต่ละวัน ตารางทั้งสองจะมีการทำงานที่แตกต่างกัน ซึ่งจาก transaction table เราสามารถทำการวิเคราะห์การใช้งาน ATM ได้หลายมุมมอง ในส่วนของ snapshot table จะให้ข้อมูลเกี่ยวกับจำนวนเงินรวมทั้งหมดในช่วงเวลาที่กำหนดซึ่งจะสามารถแสดงถึงการขยับและการเคลื่อนไหวของยอดเงินคงเหลือ

ในส่วนของคลังข้อมูลสำหรับบริษัททางการเงินจะต้องการทั้ง snapshot และ transaction tables เนื่องจาก transaction table จะช่วยในการเก็บข้อมูลธุรกรรมทางการเงินของลูกค้าในช่วงเวลาหนึ่ง และ snapshot table จะช่วยในการเก็บข้อมูลยอดคงเหลือในบัญชีของแต่ละบัญชีในช่วงเวลาที่กำหนดหรือเก็บข้อมูลยอดคงเหลือของกลุ่มของบัญชี ณ จุดสิ้นสุดของช่วงเวลาที่กำหนด

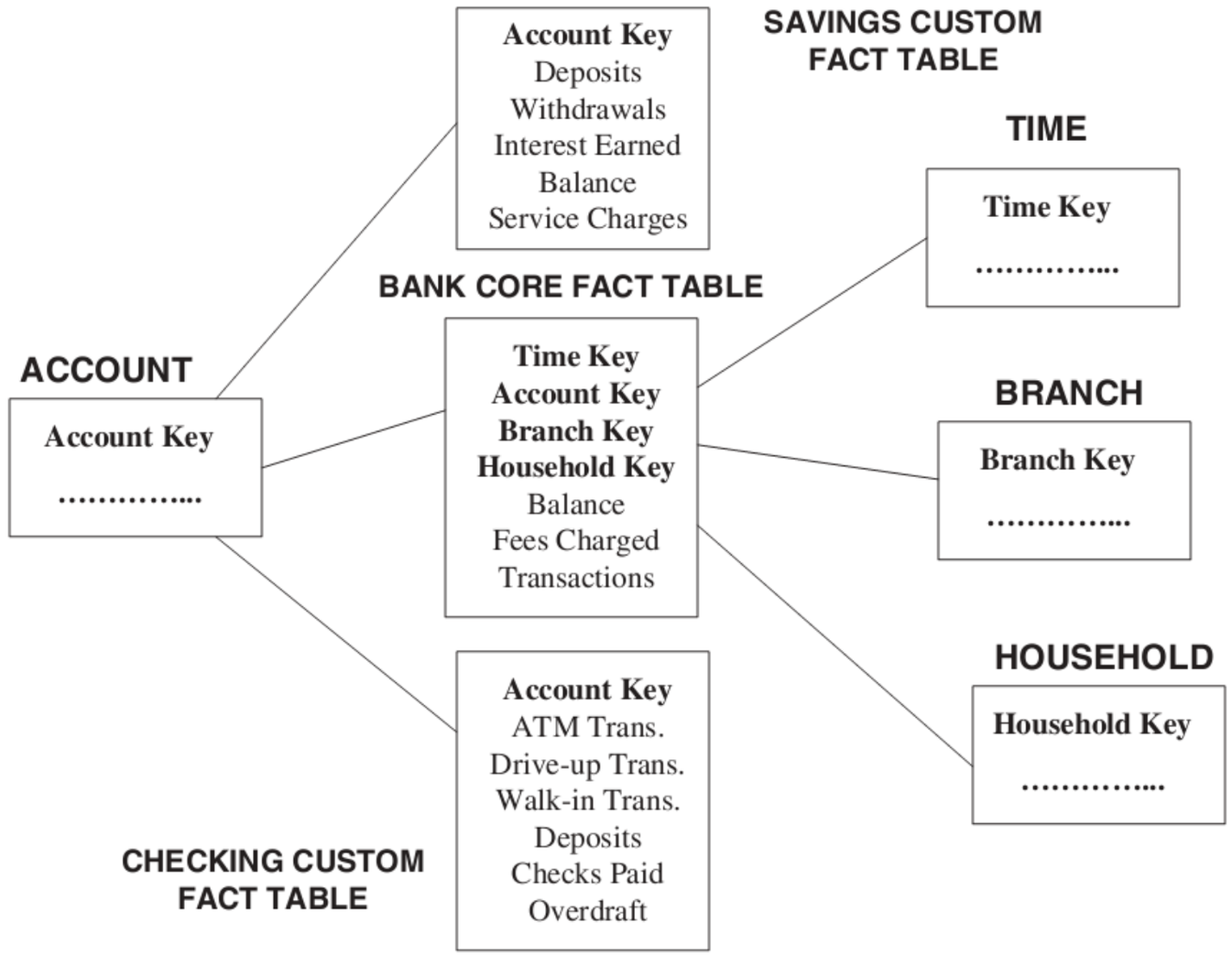
ตารางหลักและตารางที่กำหนดขึ้นเองใน family of stars

ลองพิจารณาธุรกิจสองชนิดที่มีความแตกต่างกันอย่างสิ้นเชิงดังต่อไปนี้

ธุรกิจที่ 1 คือ ธุรกิจธนาคารที่มีบริการที่หลากหลายมากและแต่ละบริการจะมีความแตกต่างกันไม่มากก็น้อย เช่น บริการตรวจสอบข้อมูลบัญชีและบริการข้อมูลบัญชีออมทรัพย์จะมีการทำงานที่เหมือนกัน แต่บริการข้อมูลบัญชีออมทรัพย์จะแตกต่างกับการให้บริการบัตรเครดิต ซึ่งจากการทำงานที่แตกต่างกัน เราจะสามารถแยกความแตกต่างของการให้บริการต่าง ๆ ได้อย่างไร และเราจะเก็บข้อมูลของการให้บริการที่มีความแตกต่างกันได้อย่างไร

ธุรกิจที่ 2 คือ บริษัทผลิตสินค้าที่มีการผลิตสินค้าที่แตกต่างกัน โดยจะมีเพียงบางส่วนของสินค้าเท่านั้นที่มีลักษณะเหมือนกัน หรือใช้วัสดุอุปกรณ์ที่เหมือนกัน เราจะสามารถเรียกดูข้อมูลเกี่ยวกับสินค้าที่แตกต่างกันได้อย่างไร

จากปัญหาของสองธุรกิจข้างต้นเราสามารถใช้ family of stars ในการเก็บข้อมูลที่มีความหลากหลายได้ โดยทำการเชื่อมโยงรายการสินค้าและบริการเข้ากับ fact table หลัก (core fact table) และแต่ละรายการสินค้าและบริการจะเกี่ยวข้องกับ fact table ย่อยอื่น ๆ (custom fact table) ซึ่งจาก family of stars ในลักษณะนี้จะมี fact table 2 ชนิดด้วยกัน ดังแสดงในรูปที่ 7-25 เราจะแสดง core และ custom fact tables ของธุรกิจธนาคาร ซึ่ง core fact table จะเก็บข้อมูลที่ใช้กับทุก ๆ การบริการนั้นก็คือข้อมูลเกี่ยวกับบัญชีของลูกค้า ในขณะที่แต่ละ custom fact table จะมีตัวชี้วัดสำหรับหัวข้อนั้น ๆ อยู่ และทั้งสอง fact table จะมีการใช้ dimension ร่วมกันอีกด้วย



รูปที่ 7- 25 ตัวอย่าง Core and custom tables

การทำให้มิติต่าง ๆ สอดคล้องกัน

เมื่อเรามองดูที่ family of stars หนึ่งๆ เราจะเห็นว่า fact table จะมีการใช้ dimension table ร่วมกัน ซึ่ง dimension table ที่ถูกใช้จะเป็นเหมือนกับตัวเชื่อมต่อระหว่าง fact table ต่างๆ เมื่อมีการใช้ dimension table ร่วมกันเราจะต้องทำให้แน่ใจว่า dimension table ที่ถูกใช้จะให้ความหมายที่เหมือนกันกับ fact table ที่เชื่อมต่อกับ dimension นั้นๆ ตัวอย่างเช่น dimension รายการสินค้าจะเกี่ยวข้องกับ fact table ของการขายสินค้า และ fact table สำหรับการส่งสินค้า ดังแสดงในรูปที่ 7-26 ซึ่งจะมีการใช้ dimension รายการสินค้า เวลา ลูกค้า และผู้ขายร่วมกัน ดังนั้นเราจะต้องพิจารณาแต่ละแอทริบิวใน dimension นั้น ๆ ว่าให้ข้อมูลที่ถูกต้องและสมบูรณ์กับทั้ง 2 fact tables หรือไม่ โดยเราจะต้องทำการตรวจสอบถึงชนิดของข้อมูล ความยาว และเงื่อนไขต่าง ๆ ของข้อมูลด้วย

การทำให้ dimension ต่างๆ มีความสอดคล้องกันนั้นเป็นสิ่งที่สำคัญมาก ดังนั้นเราควรจะใส่ใจกับความสอดคล้องของ dimension ที่ถูกใช้ร่วมกันจากหลายๆ fact table การทำให้ dimension เหมือนกันหรือสอดคล้องกันจะช่วยให้เราสามารถทำการสรุปข้อมูลจากหลาย ๆ data mart อีกด้วย

การสร้างมาตราฐานให้กับ fact table ต่าง ๆ ใน family of stars

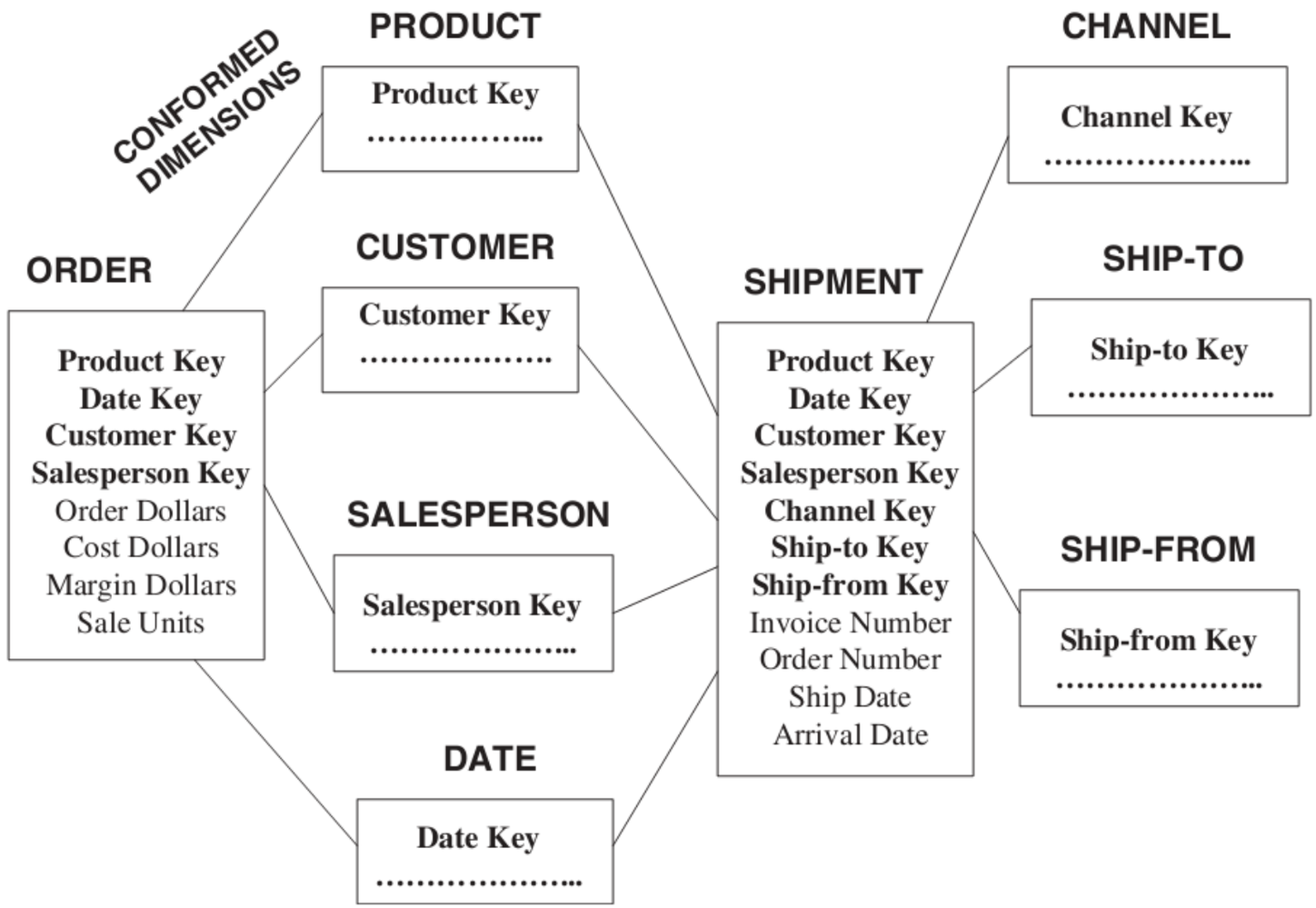
ในการทำให้ dimension ต่าง ๆ ที่ถูกใช้โดยหลาย fact table ให้มีความสอดคล้องกัน เราจะต้องทำให้ fact table นั้นมีมาตราฐานเดียวกันก่อน จึงจะสามารถทำให้ dimension เหล่านั้นมีความสอดคล้องกันได้ ในการทำให้ fact table มีรูปแบบที่เหมือนกันนั้นจะเกี่ยวกับ

(1) การกำหนดรูปแบบหรือนิยามของ fact table ที่อยู่ในแต่ละ data mart ให้เหมือนกัน

(2) แก้ปัญหาเกี่ยวกับคำที่พ้องรูปหรือพ้องเสียง

(3) ควรจะต้องใช้อัลกอริทึมเดียวกันในการเข้าถึงข้อมูลเข้าสู่แต่ละ fact table

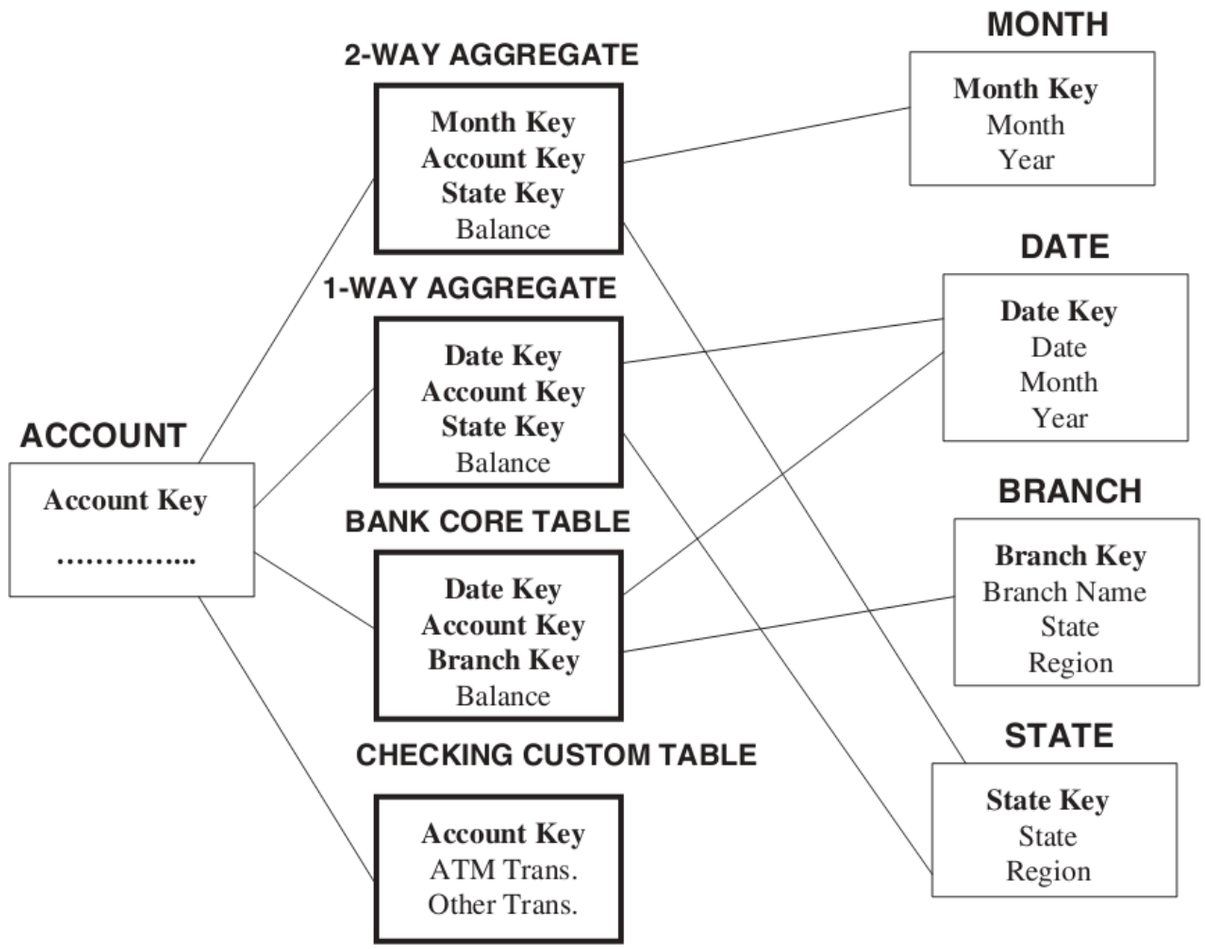
(4) ควรจะทำให้แน่ใจได้ว่าแต่ละ fact table จะใช้หน่วยของมาตรวัดเดียวกัน และอื่นๆ



รูปที่ 7-26 ตัวอย่างความสอดคล้องของ dimensions ต่าง ๆ ที่ถูกใช้โดยหลาย fact table

สรุปเกี่ยวกับ Family of Stars

ลองพิจารณารูปที่ 7-27 ที่จะแสดงตัวอย่าง family of stars ที่มีการใช้เทคนิคต่าง ๆ เช่น การทำให้ fact table มีมาตราฐานเดียวกัน การทำให้ dimension ต่าง ๆ เหมือนกันหรือสอดคล้องกัน การรวบยอดข้อมูลโดยใช้ one-way และ two-way aggregate ซึ่งจะทำให้เข้าใจการถึงการออกแบบ family of stars ได้มากขึ้น



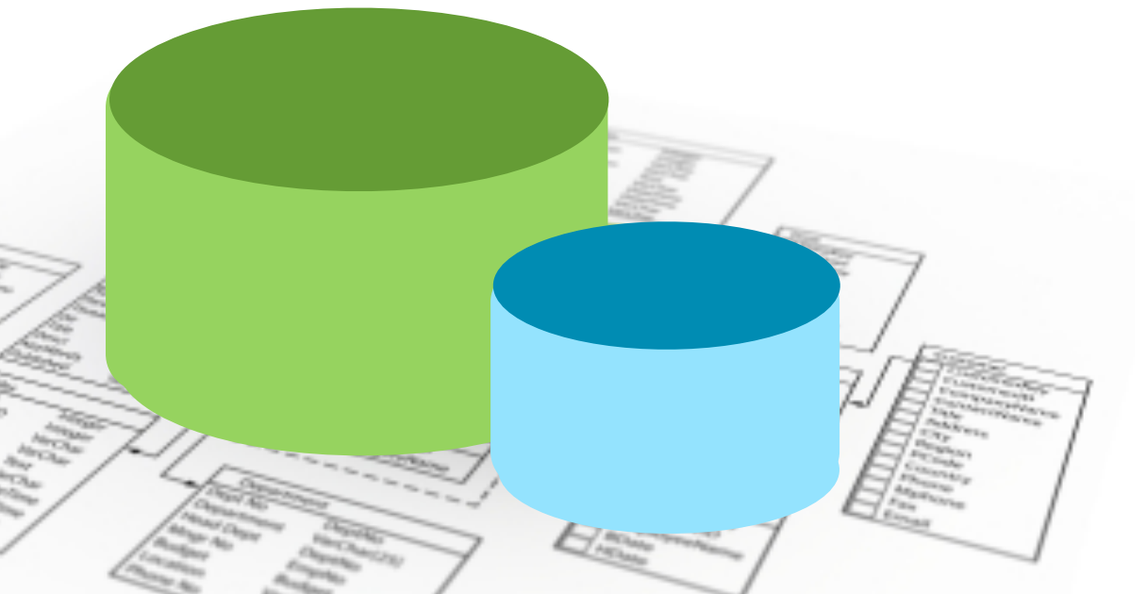
รูปที่ 7-27 ตัวอย่างสรุปเกี่ยวกับ family of stars

SECTION 7

ความเปลี่ยนแปลงที่เกิดขึ้นกับข้อมูลในคลังข้อมูล

ความเปลี่ยนแปลงที่เกิดขึ้นกับข้อมูลในคลังข้อมูล

จากส่วนก่อนหน้านี้เราจะทราบถึงโมเดลต้นแบบของการสร้างแบบจำลองมิติต่าง ๆ สำหรับการสืบค้นข้อมูลที่ประกอบไปด้วย star-schema snowflake-schema และ family of stars รวมถึงเทคนิคในการรวบยอดข้อมูลเพื่อเพิ่มประสิทธิภาพในการสืบค้นข้อมูล แต่อย่างไรก็ดีเมื่อเวลาเปลี่ยนแปลงไปจะทำให้ข้อมูลใน fact table เปลี่ยนแปลงไป โดยอาจมีแถว/เรคคอร์ดของข้อมูลเป็นจำนวนมากถูกเพิ่มเข้าไปใน fact table ซึ่งจะทำให้ fact table นั้นมีข้อมูลมากขึ้นเรื่อย ๆ จากการอัพเดทข้อมูลจากแหล่งข้อมูล แต่ในการอัพเดทข้อมูลส่วนใหญ่จะไม่ทำการอัพเดทข้อมูลใน fact table เท่าไรนักมักจะเกิดขึ้นในส่วนของ dimension table ที่อาจจะมีความเปลี่ยนแปลงเกิดขึ้น เช่น การเพิ่มจำนวนแถว/เรคคอร์ดของข้อมูล และการอัพเดทข้อมูลในแอทริบิวต่างๆ ในแต่ละแถวของข้อมูล ซึ่งในบางคลังข้อมูลอาจมีการเพิ่มรายการสินค้าให้กับ product dimension table หรือบางรายการสินค้าอาจมีการเปลี่ยนแปลงหมวดหมู่ของสินค้า (product category) เป็นต้น



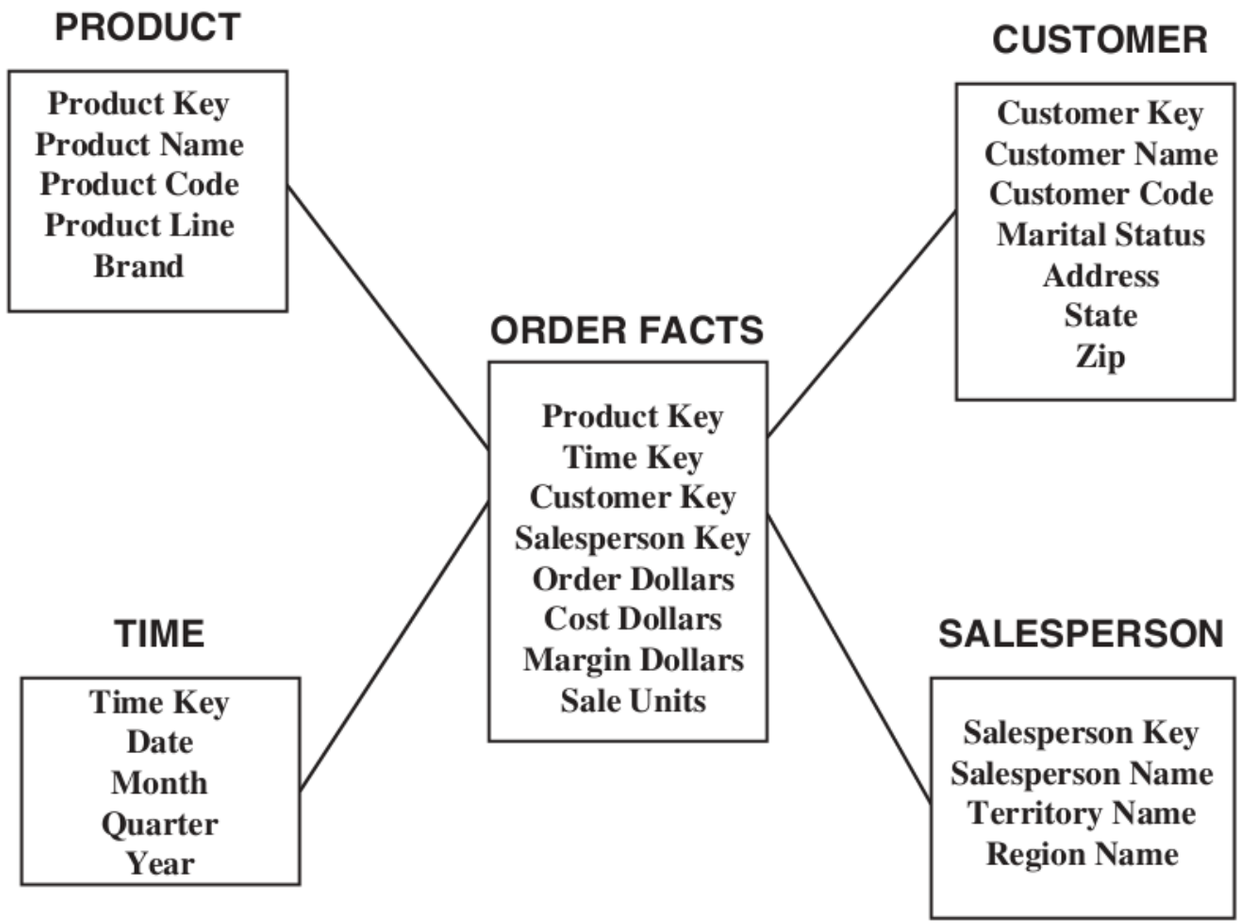
ความเปลี่ยนแปลงที่เกิดขึ้นกับ fact และ dimension table โดยส่วนใหญ่จะมีอยู่ด้วยกัน 2 ประเภทด้วยกันคือ ความเปลี่ยนแปลงที่เกิดขึ้นอย่างช้า ๆ (slowly changing) และความเปลี่ยนแปลงที่เกิดขึ้นอย่างรวดเร็ว (rapidly changing) โดยคลังข้อมูลที่เราสร้างขึ้นอาจมีความเปลี่ยนแปลงประเภทใดประเภทหนึ่งเกิดขึ้นหรืออาจมีทั้ง 2 ประเภทก็เป็นได้ ถ้าเราทราบถึงรายละเอียดของการเปลี่ยนแปลงของข้อมูลที่จะเกิดขึ้นจะช่วยให้เราออกแบบการจัดข้อมูลได้อย่างมีประสิทธิภาพมากขึ้น เพื่อให้เข้าใจถึงความเปลี่ยนแปลงที่อาจเกิดขึ้นกับคลังข้อมูล เราควรจะพิจารณารายละเอียดของความเปลี่ยนแปลงแต่ละประเภท ดังนี้

ความเปลี่ยนแปลงที่เกิดขึ้นอย่างช้าๆ

เป็นความเปลี่ยนแปลงที่มักจะเกิดขึ้นกับ dimension table โดยเกิดขึ้นอย่างช้าๆหรือเกิดขึ้นไม่บ่อยนัก ซึ่งจากความเปลี่ยนแปลงดังกล่าวจะทำให้เราได้หลักการพื้นฐานดังต่อไปนี้

- Dimension ส่วนใหญ่มักจะคงที่ตลอดเวลา
- จะมีหลายๆ dimension ที่ไม่คงที่ แต่ก็มีความเปลี่ยนแปลงเกิดขึ้นอย่างช้าๆ
- แอทริบิวต่างๆจะมีความเปลี่ยนแปลงเกิดขึ้นอย่างช้าๆ
- ในระบบการดำเนินงาน ค่าที่มีการเปลี่ยนแปลงจะเขียนทับลงค่าเก่าเสมอ
- การเขียนทับค่าที่มีการเปลี่ยนแปลงของแอทริบิวหนึ่งๆนั้นไม่ใช่ทางเลือกที่เหมาะสมสำหรับคลังข้อมูล
- วิถีของความเปลี่ยนแปลงของข้อมูลใน dimension table จะขึ้นอยู่กับชนิดของการเปลี่ยนแปลง และข้อมูลที่เราต้องการสงวนไว้ในคลังข้อมูล

จากหลักการพื้นฐานข้างต้นเราจะสามารถสรุปเกี่ยวกับประเภทของความเปลี่ยนแปลงที่เกิดขึ้นอย่างช้าๆ โดยสามารถแบ่งได้เป็น 3 ประเภทคือ ชนิดที่ 1 ชนิดที่ 2 และ ชนิดที่ 3 ตามลำดับ เพื่อให้เข้าใจถึงความเปลี่ยนแปลงทั้ง 3 ชนิด และวิธี/เทคนิคในการจัดการกับความเปลี่ยนแปลงทั้ง 3 ชนิด ลองพิจารณาตัวอย่าง star schema ของการสั่งซื้อสินค้าที่ประกอบไปด้วย 1 fact table และ 4 dimension table ดังแสดงในรูปที่ 7-28 เพื่อให้เห็นถึงลักษณะของความเปลี่ยนแปลงทั้ง 3 ชนิดดังนี้



รูปที่ 7-28 ตัวอย่าง star schema สำหรับการสั่งซื้อสินค้า

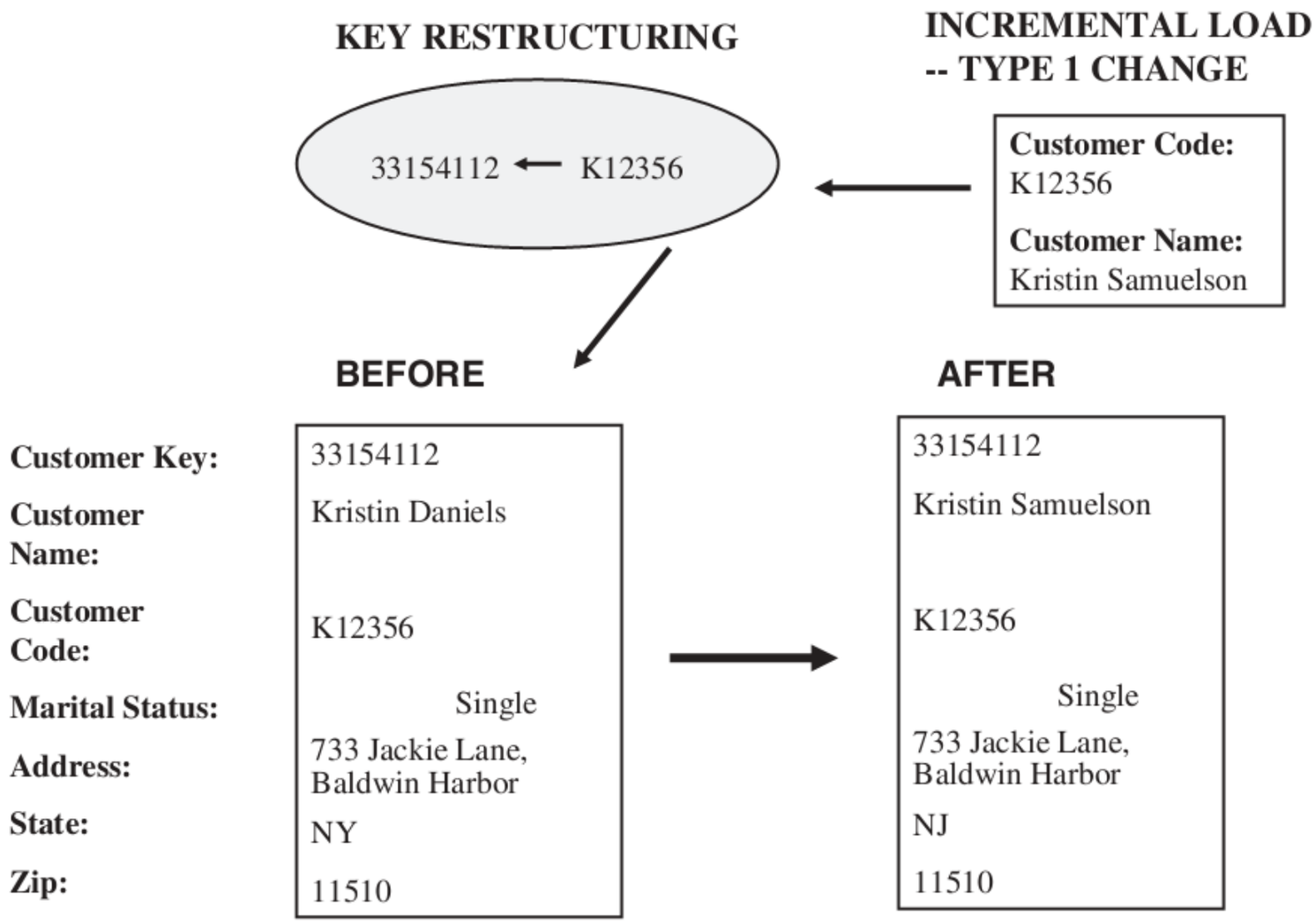
ความเปลี่ยนแปลงชนิดที่ 1

เป็นความเปลี่ยนแปลงที่เกิดจากการแก้ไขข้อมูลใน dimension table ที่มีความผิดพลาด ตัวอย่างเช่น การแก้ไขชื่อลูกค้าที่เกิดจากการสะกดผิด เช่น ชื่อจริงชื่อ Michael Romano แต่ถูกพิมพ์ผิดเป็น Michel Romano หรือเราจะต้องทำการเปลี่ยนชื่อลูกค้าจาก Kristin Daniels ไปเป็น Kristin Samuelson เมื่อลูกค้าแต่งงานและทำการเปลี่ยนนามสกุล เป็นต้น ซึ่งเมื่อพิจารณาการเปลี่ยนแปลงที่เกิดขึ้นกับชื่อของลูกทั้ง 2 กรณีเราจะไม่ต้องทำการเก็บค่าเก่าไว้ โดยที่เราสามารถเขียนข้อมูลที่ถูกต้องทับหรือแทนที่ข้อมูลที่มีความผิดพลาดได้โดยตรง จากความเปลี่ยนแปลงชนิดที่ 1 เราสามารถสรุปเป็นหลักการต่าง ๆ ได้ดังนี้

- ความเปลี่ยนแปลงชนิดที่ 1 จะเกี่ยวข้องกับการแก้ไขข้อผิดพลาดของข้อมูลที่เกิดขึ้น
- ในหลาย ๆ ครั้งที่เกิดความเปลี่ยนแปลงชนิดที่ 1 ในแหล่งข้อมูลมักจะเป็นความเปลี่ยนแปลงที่ไม่สำคัญ
- ค่าที่เกิดความผิดพลาดที่ถูกเก็บอยู่ในแหล่งข้อมูลจะถูกลบทิ้งหรือเขียนทับไป
- คลังข้อมูลไม่จำเป็นต้องเก็บข้อมูลความเปลี่ยนแปลงชนิดที่ 1 ไว้

จากหลักการข้างต้น เมื่อมีความเปลี่ยนแปลงชนิดที่ 1 เกิดขึ้นในระบบการดำเนินงานหรือแหล่งข้อมูล เราควรจะแก้ไขดังต่อไปนี้ (ดังแสดงในรูปที่ 7-29)

- ทำการเขียนข้อมูลแอทริบิวที่มีความเปลี่ยนแปลงทับข้อมูลเดิมใน dimension table
- ไม่ต้องทำการเก็บข้อมูลเดิมที่มีความผิดพลาดไว้ในคลังข้อมูล
- ไม่ต้องทำการเปลี่ยนแปลงข้อมูลแอทริบิวอื่นๆ ที่ไม่มีความเปลี่ยนแปลงเกิดขึ้น
- การอัพเดทข้อมูลจะไม่ส่งผลกระทบต่อคีย์หลักใน dimension table



รูปที่ 7-29 การแก้ปัญหาความเปลี่ยนแปลงชนิดที่ 1

ความเปลี่ยนแปลงชนิดที่ 2

จะเป็นความเปลี่ยนแปลงของข้อมูลทั่ว ๆ ไปเมื่อเวลาล่วงเลยไป ลองย้อนกลับไปที่ความเปลี่ยนแปลงของสถานะภาพทางสังคมของ Kristin Samuelson ที่เกิดขึ้นเมื่อวันที่ 1 ตุลาคม 2008 ซึ่งถ้าสมมติว่าความต้องการหนึ่งของการใช้งานคลังข้อมูลจะเกี่ยวกับการสั่งซื้อสินค้าโดยแยกตามสถานะภาพทางสังคม เราจะต้องทำการแยกข้อมูลการสั่งซื้อสินค้าของ Kristin Samuelson ก่อนวันที่ทำการแต่งงานไว้อยู่ในกลุ่มของลูกค้าสถานะภาพโสด และทำการแยกข้อมูลการสั่งซื้อสินค้าหลังวันแต่งงานไว้เป็นการสั่งซื้อสินค้าของกลุ่มลูกค้าที่แต่งงานแล้ว ต่อมา Kristin ได้ทำการย้ายที่อยู่หลังจากแต่งงานแล้ว โดยเธอได้ทำการย้ายบ้านจากรัฐนิวยอร์กไปยังรัฐแคลิฟอร์เนียตามสามีของเธอในวันที่ 1 พฤศจิกายน 2008 และเผอิญว่าผู้ใช้คลังข้อมูลต้องการเรียกดูข้อมูลยอดการสั่งซื้อสินค้าตามเขตที่อยู่อาศัยของลูกค้า ซึ่งจะทำให้เราต้องทำการแบ่งข้อมูลของ Kristin ออกไปอีก 1 เรคคอร์ด โดยทำการเปลี่ยนข้อมูลที่อยู่ของเธอ ดังแสดงในรูปที่ 7-30

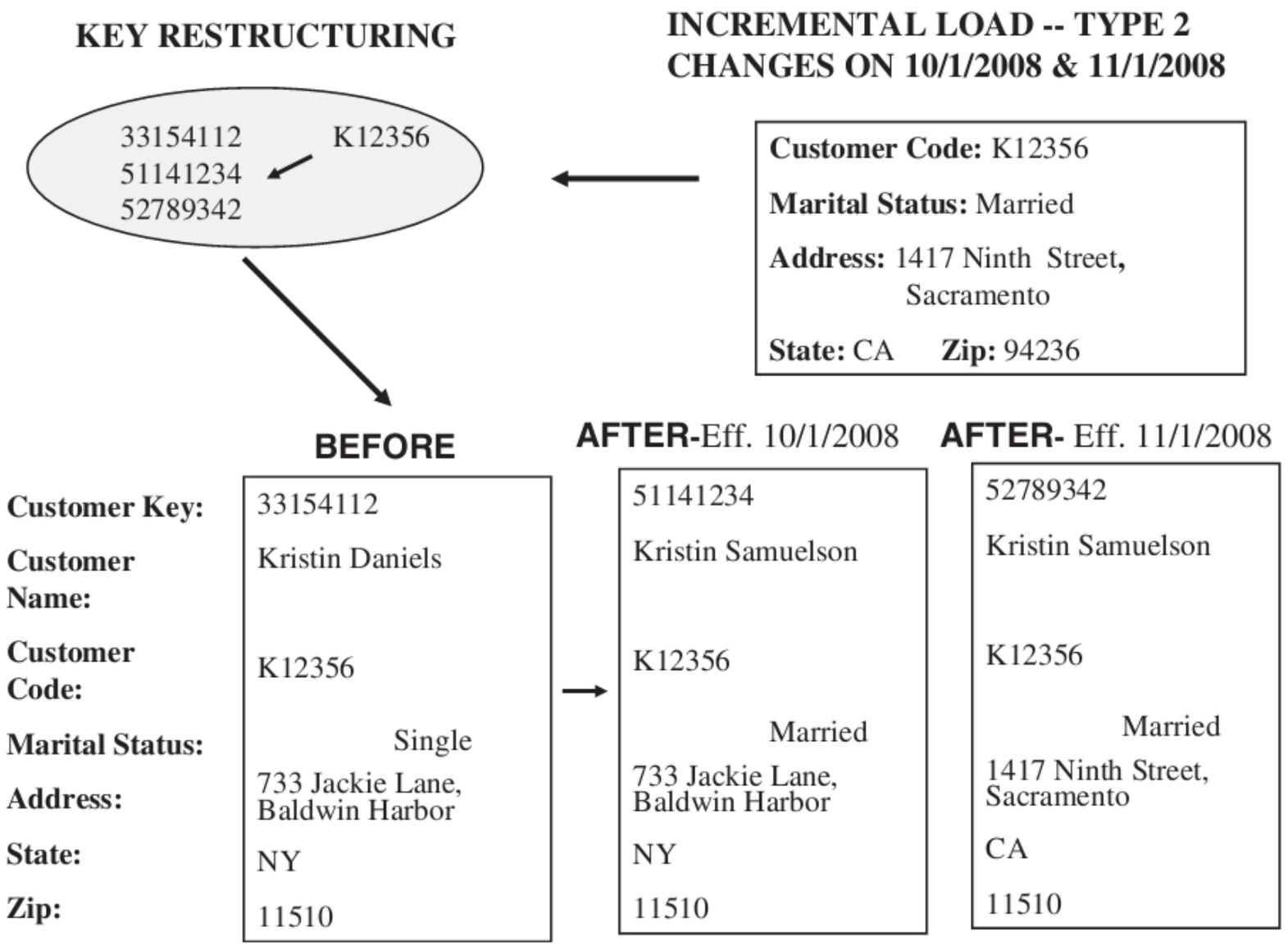
จากความเปลี่ยนแปลงที่ได้กล่าวข้างต้นเราสามารถสรุปเกี่ยวกับหลักการของการเกิดขึ้นของความเปลี่ยนแปลงชนิดที่ 2 และวิธีในการจัดการความเปลี่ยนแปลงดังกล่าวได้ดังนี้

หลักการ

- ความเปลี่ยนแปลงชนิดที่ 2 มักจะเกี่ยวข้องกับความเปลี่ยนแปลงที่แท้จริงที่เกิดขึ้นในแหล่งข้อมูล
- เราต้องทำการเก็บประวัติย้อนหลังของข้อมูล (ข้อมูลก่อนการเปลี่ยนแปลง) ไว้ในคลังข้อมูล
- ความเปลี่ยนแปลงที่เกิดขึ้นจะเปรียบเสมือนตัวแบ่งประวัติของข้อมูลในคลังข้อมูล
- เราจะต้องทำการเก็บข้อมูลที่เกี่ยวข้องกับทุกๆความเปลี่ยนแปลงที่เกิดขึ้นในแอทริบิวหนึ่ง

วิธีการจัดการ

- ทำการเพิ่มแถว/เรคคอร์ดใหม่ให้กับ dimension table โดยทำการเก็บค่าใหม่ที่มีการเปลี่ยนแปลงของแอทริบิวนั้นๆ
- เราอาจต้องทำการเก็บวันที่ของการเปลี่ยนแปลงลงไปด้วย
- จะไม่มีผลกระทบหรือความเปลี่ยนแปลงเกิดขึ้นกับข้อมูลก่อนหน้าที่ถูกเก็บอยู่ในคลังข้อมูล
- ทำการเพิ่มข้อมูลแถวใหม่ โดยการสร้างคีย์หลักใหม่ ซึ่งจะทำให้ไม่มีผลกระทบกับคีย์หลักของข้อมูลเก่าในคลังข้อมูล



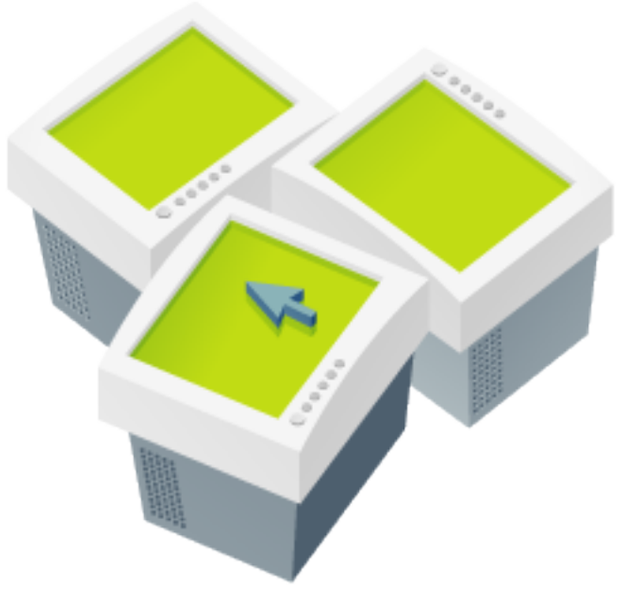
รูปที่ 7-30 แก้ปัญหาความเปลี่ยนแปลงชนิดที่ 2

ความเปลี่ยนแปลงชนิดที่ 3

โดยทั่วไปของความเปลี่ยนแปลงที่เกิดขึ้นใน dimension table จะเป็นความเปลี่ยนแปลงชนิดที่ 1 หรือ 2 แต่สำหรับความเปลี่ยนแปลงชนิดที่ 2 ที่เปรียบเสมือนตัวแบ่งข้อมูลที่ต้องมีการเพิ่มจำนวนแถวเข้าไปใน dimension table เช่น สถานะภาพทางสังคมของ Kristin ที่มีการเปลี่ยนแปลงวันที่ 1 ตุลาคมปี 2008 จะทำให้เราสามารถแบ่งข้อมูลการสั่งซื้อของ Kristin ออกเป็นยอดการสั่งซื้อก่อนแต่งงานและหลังแต่งงาน ตามลำดับ ซึ่งจะทำให้เราสามารถนับข้อมูลได้เฉพาะกลุ่มใดกลุ่มหนึ่งของสถานะภาพทางสังคมเท่านั้น แต่อย่างไรก็ดีถ้าเราต้องการที่จะดูข้อมูลการสั่งซื้อทั้งหมดโดยไม่แยกการสั่งซื้อก่อนหรือหลังแต่งงาน จากความต้องการดังกล่าว วิธีที่ใช้จัดการกับความเปลี่ยนแปลงชนิดที่ 2 ไม่สามารถนำมาประยุกต์ใช้งานได้ เราจำเป็นจะต้องหาวิธีจัดการกับความเปลี่ยนแปลงชนิดนี้ ที่เรียกได้ว่าเป็นความเปลี่ยนแปลงชนิดที่ 3

ความเปลี่ยนแปลงชนิดที่ 3 นั้นจะเป็นความเปลี่ยนแปลงแบบชั่วคราวหรือความเปลี่ยนแปลงแบบผิวเผิน ตัวอย่างเช่น พนักงานขายคนหนึ่งนั้นทำการย้ายเขตการขายสินค้าจากเดิมทำการขายสินค้าในเขต New England ไปเป็น Chicago ซึ่งเราอาจจะต้องการเรียกดูข้อมูลยอดขายทั้งหมดจากทั้ง 2 เขต เราจะต้องทำการหาวิธีการในการจัดการกับความเปลี่ยนแปลงดังกล่าว ซึ่งมีหลักการดังต่อไปนี้

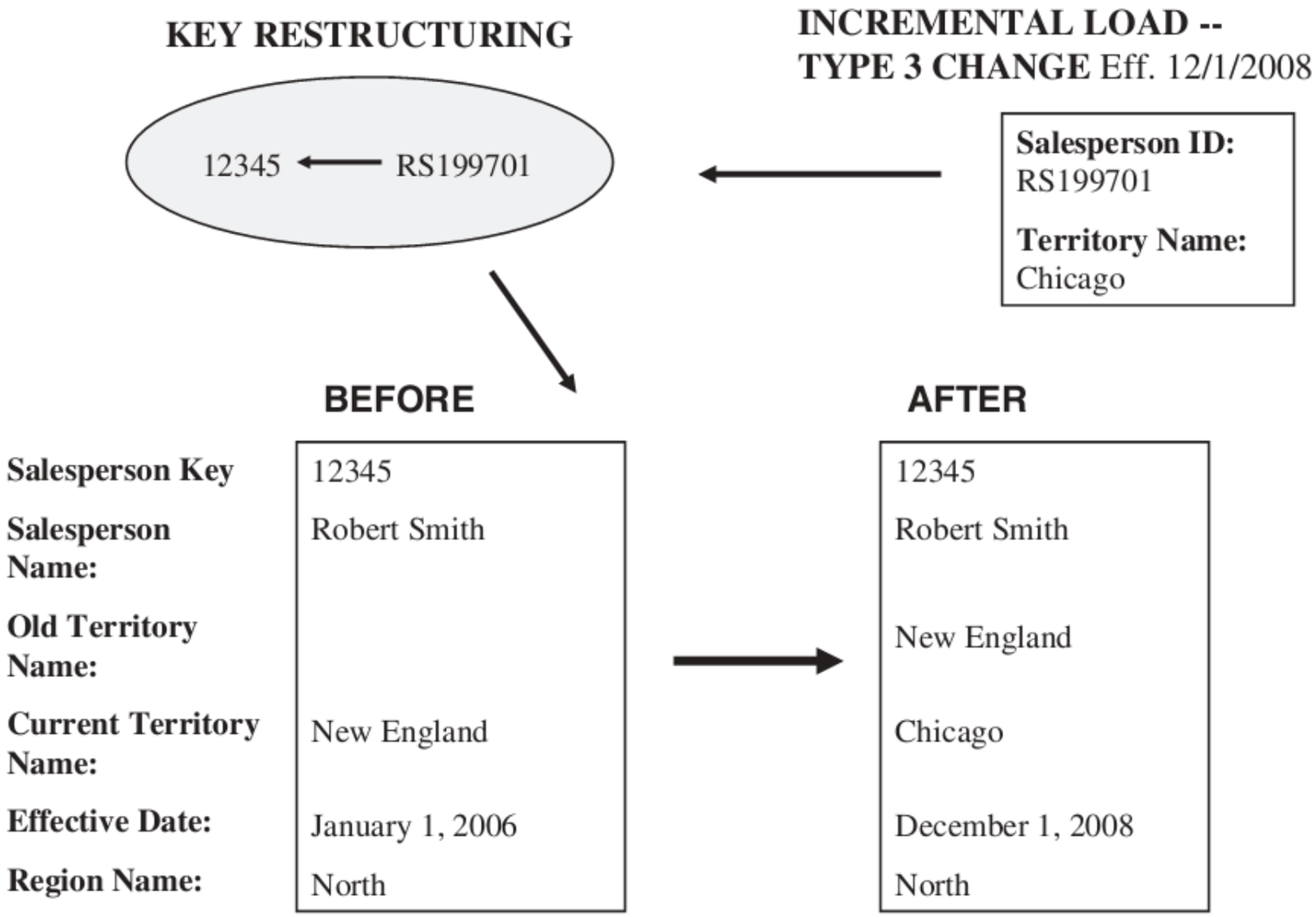
- ความเปลี่ยนแปลงชนิดที่ 3 จะเป็นความเปลี่ยนแปลงแบบชั่วคราวหรือความเปลี่ยนแปลงแบบผิวเผินที่เกิดขึ้นในแหล่งข้อมูล
- เราต้องการที่จะเก็บข้อมูลทั้งข้อมูลเก่าและข้อมูลใหม่ที่มีการเปลี่ยนแปลงไว้
- ข้อมูลทั้งเก่าและใหม่จะถูกใช้เพื่อเปรียบเทียบความเปลี่ยนแปลงที่เกิดขึ้น
- ความเปลี่ยนแปลงลักษณะนี้จะมีความสามารถในการเรียกดูข้อมูลทั้งก่อนหน้า และ ย้อนหลังได้



จากหลักการข้างต้น เราสามารถจัดการกับความเปลี่ยนแปลงชนิดที่ 3 ได้ดังรูปที่ 7-31 ซึ่งสามารถแสดงรายละเอียดได้ดังนี้

- ทำการเพิ่มฟิลด์ (filed) เพื่อทำการเก็บค่าของข้อมูลเก่าไว้ในเรคคอร์ดเดิม
- ทำการเก็บค่าของข้อมูลเก่าเข้าไปในฟิลด์ใหม่ที่ทำการสร้างขึ้น
- ทำการเก็บข้อมูลใหม่ไว้ในแอทริบิวเดิม
- เราอาจจะทำการเพิ่มวันที่ของการเปลี่ยนแปลงไว้ในแอทริบิวที่เกี่ยวข้อง
- คีย์หลักของข้อมูลแต่ละแถวจะไม่ได้รับผลกระทบใดๆ
- ไม่ต้องทำการเพิ่มข้อมูลใหม่ใน dimension table
- การสืบค้นข้อมูลสามารถสืบค้นข้อมูลที่เป็นค่าใหม่และค่าเก่า

จากความเปลี่ยนแปลงทั้ง 3 ชนิดของความเปลี่ยนแปลงที่เกิดขึ้นอย่างช้าๆจะทำให้เราทราบถึงความเปลี่ยนแปลงที่เกิดขึ้นในระบบการดำเนินงานและส่งผลถึงข้อมูลใน dimension table แต่อย่างไรก็ดียังมีความเปลี่ยนแปลงอีกชนิดหนึ่งที่เป็นความเปลี่ยนแปลงอย่างรวดเร็วที่อาจจะเกิดขึ้นกับ dimension table

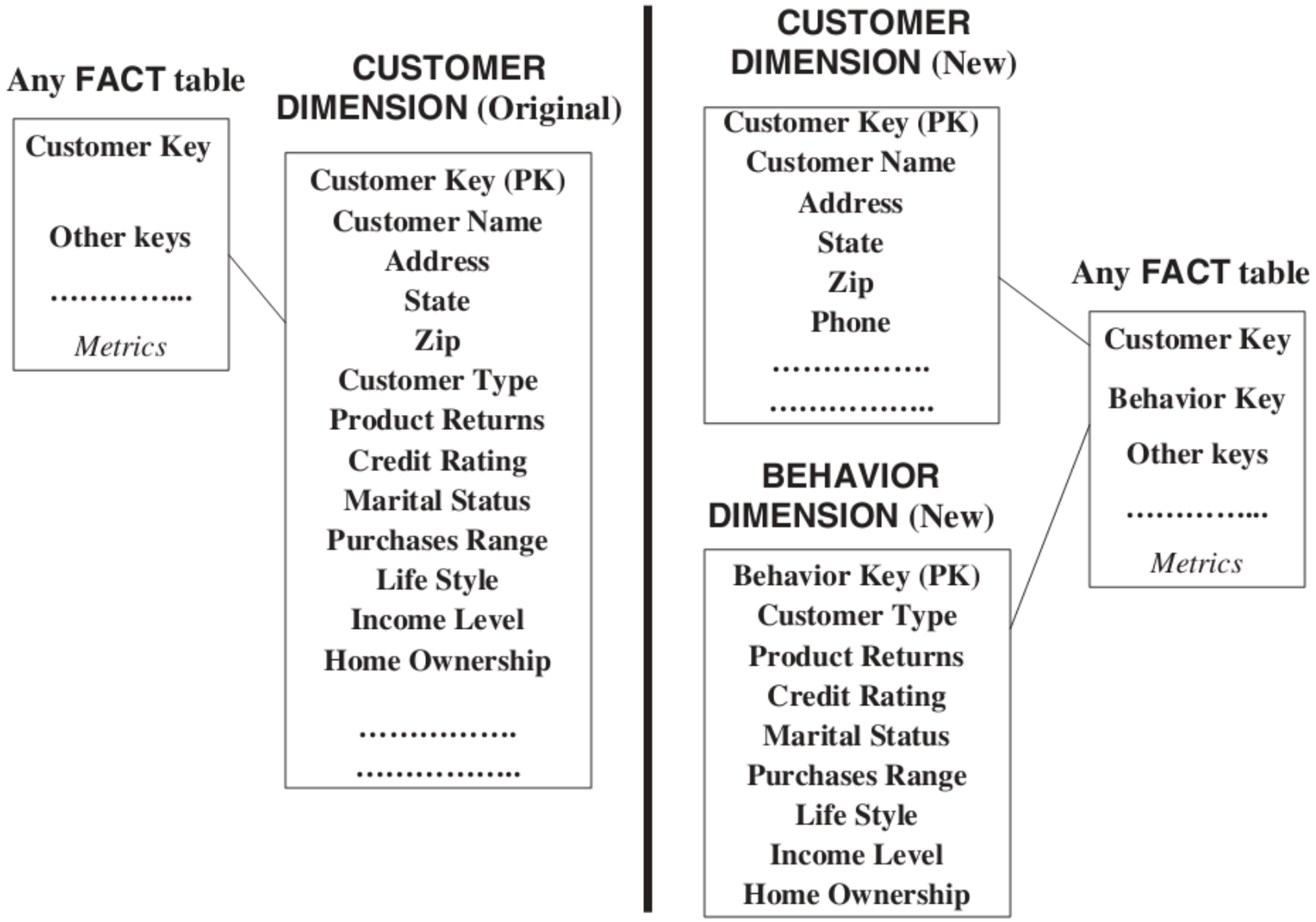


รูปที่ 7-31 แก้ปัญหาความเปลี่ยนแปลงชนิดที่ 3

ความเปลี่ยนแปลงที่เกิดขึ้นอย่างรวดเร็ว

จากส่วนที่แล้วเราจะทราบว่าความเปลี่ยนแปลงที่เกิดขึ้นอย่างช้าๆ จะเป็นความเปลี่ยนแปลงที่เกิดขึ้นไม่บ่อยและเกิดขึ้นกับ dimension table เป็นส่วนใหญ่ ตัวอย่างเช่น product dimension table อาจมีการเปลี่ยนแปลงของเรคคอร์ดหนึ่งๆเพียง 1 หรือ 2 ครั้งต่อปีเท่านั้น หรือใน customer dimension table จะมีเรคคอร์ดเป็นจำนวนมากที่มีการเปลี่ยนแปลง แต่เรคคอร์ดเหล่านั้นก็เปลี่ยนแปลงไม่บ่อย ซึ่งจากความเปลี่ยนแปลงที่ไม่บ่อย เราสามารถใช้วิธีแก้ปัญหาสำหรับความเปลี่ยนแปลงชนิดที่ 2 ด้วยการเก็บข้อมูลแยกระหว่างข้อมูลก่อนการเปลี่ยนแปลง และข้อมูลหลังการเปลี่ยนแปลงออกเป็น 2 เรคคอร์ดด้วยกัน แต่เมื่อข้อมูลมีการเปลี่ยนแปลงเกิดขึ้นอย่างรวดเร็วกับ dimension table ที่มีขนาดใหญ่ การใช้วิธีการแก้ปัญหาสำหรับความเปลี่ยนแปลงชนิดที่ 2 อาจไม่สามารถแก้ปัญหาได้ดีเนื่องจากจะยิ่งทำให้ dimension table มีขนาดใหญ่มากขึ้นซึ่งจะทำให้สิ้นเปลืองเวลาในการตอบคิวรีต่าง ๆ มากขึ้นตามไปด้วย

ทางเลือกหนึ่งที่น่าจะใช้การได้ดี คือ การแบ่ง dimension table ที่มีขนาดใหญ่ออกเป็นหลาย ๆ dimension table ที่มีขนาดเล็กลง ดังแสดงในรูปที่ 7-32 ซึ่งจะทำการแบ่งส่วนของแอทริบิวที่มีการเปลี่ยนแปลงอย่างรวดเร็วแยกไว้เป็นอีกตารางหนึ่งแล้วทำการเชื่อมต่อกับ fact table ตรงๆ และทำการปล่อยให้แอทริบิวที่มีการเปลี่ยนแปลงค่อนข้างช้าไว้ใน dimension table เดิม



รูปที่ 7- 32 การแบ่ง dimension table ที่มีขนาดใหญ่เพื่อแก้ปัญหาความเปลี่ยนแปลงที่เกิดขึ้นอย่างรวดเร็ว

SECTION 8

คำถามท้ายบท

1. Star schema คืออะไร ส่วนประกอบของ star schema มีอะไรบ้าง อย่างไร
2. จงอธิบายถึงความเบาบางของข้อมูลใน fact table
3. จงอธิบายถึงความแตกต่างระหว่างมาตรวัดที่เป็นแบบ full-additive และ semi-additive
4. จงอธิบายถึงคีย์ต่างๆ ใน dimension และ fact tables
5. จงอธิบายถึงความละเอียดของข้อมูลในคลังข้อมูล
6. จงอธิบายถึงประโยชน์ของ star schema มา 3 ข้อ และอธิบายถึงข้อเสียของ star schema
7. ลำดับชั้นและหมวดหมู่ของข้อมูลใน dimension table เป็นอย่างไร
8. Snowflake schema แตกต่างจาก Star schema อย่างไร จงยกตัวอย่างข้อดี-ข้อเสียของ Snowflake schema มาอย่างละ 2 ข้อ
9. ความสอดคล้องของ dimension table คืออะไร
10. Aggregate fact table คืออะไร ทำไมเราถึงต้องการ aggregate fact table จงยกตัวอย่าง
11. จงอธิบายถึง slowly changing dimensions และประเภทของ slowly chaning dimensions
12. จงเปรียบเทียบความแตกต่างระหว่างความเปลี่ยนแปลงชนิดที่ 2 และความเปลี่ยนแปลงชนิดที่ 3
13. จงอธิบายถึงความแตกต่างระหว่าง slowly และ rapidly changing dimensions

CHAPTER 8

การสกัด การเปลี่ยนแปลง และการถ่ายโอนข้อมูล

SECTION 1

แผนการสอนประจำบท

วัตถุประสงค์ของบทเรียน

- ศึกษาเกี่ยวกับการสกัด การเปลี่ยนแปลง และถ่ายโอนข้อมูล รวมถึงสถาปัตยกรรมของอีทีแอล
- ศึกษาเกี่ยวกับลักษณะของข้อมูลที่มักพบบ่อยในระบบการดำเนินงาน
- ศึกษาเกี่ยวกับการสกัดข้อมูลด้วยวิธีการต่าง ๆ
- ศึกษาเกี่ยวกับฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล
- ศึกษาเกี่ยวกับการถ่ายโอนข้อมูลเข้าสู่คลังข้อมูลในลักษณะต่าง ๆ

เนื้อหาของบทเรียน

เนื้อหาในบทนี้จะประกอบด้วย สถาปัตยกรรมของอีทีแอล ขั้นตอนการทำงานของอีทีแอล การสกัดข้อมูล วิธีการสกัดข้อมูล การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล ขั้นตอนการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล ประเภทการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล การถ่ายโอนข้อมูลไปยังคลังข้อมูล

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารประกอบการสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

กิจกรรมการเรียน-การสอน

- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารประกอบการสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

การวัดและประเมินผล

- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

SECTION 2

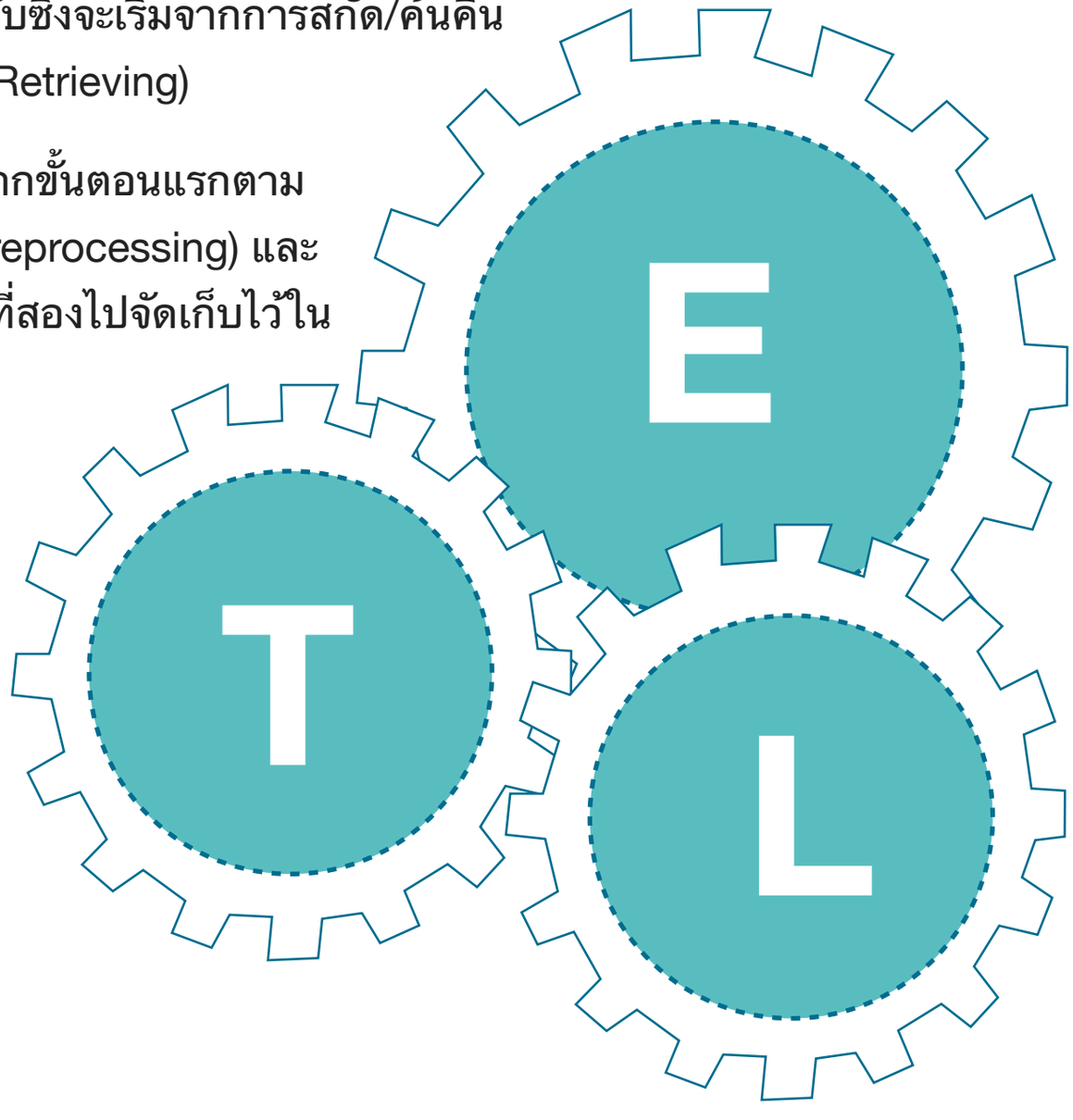
บทนำ

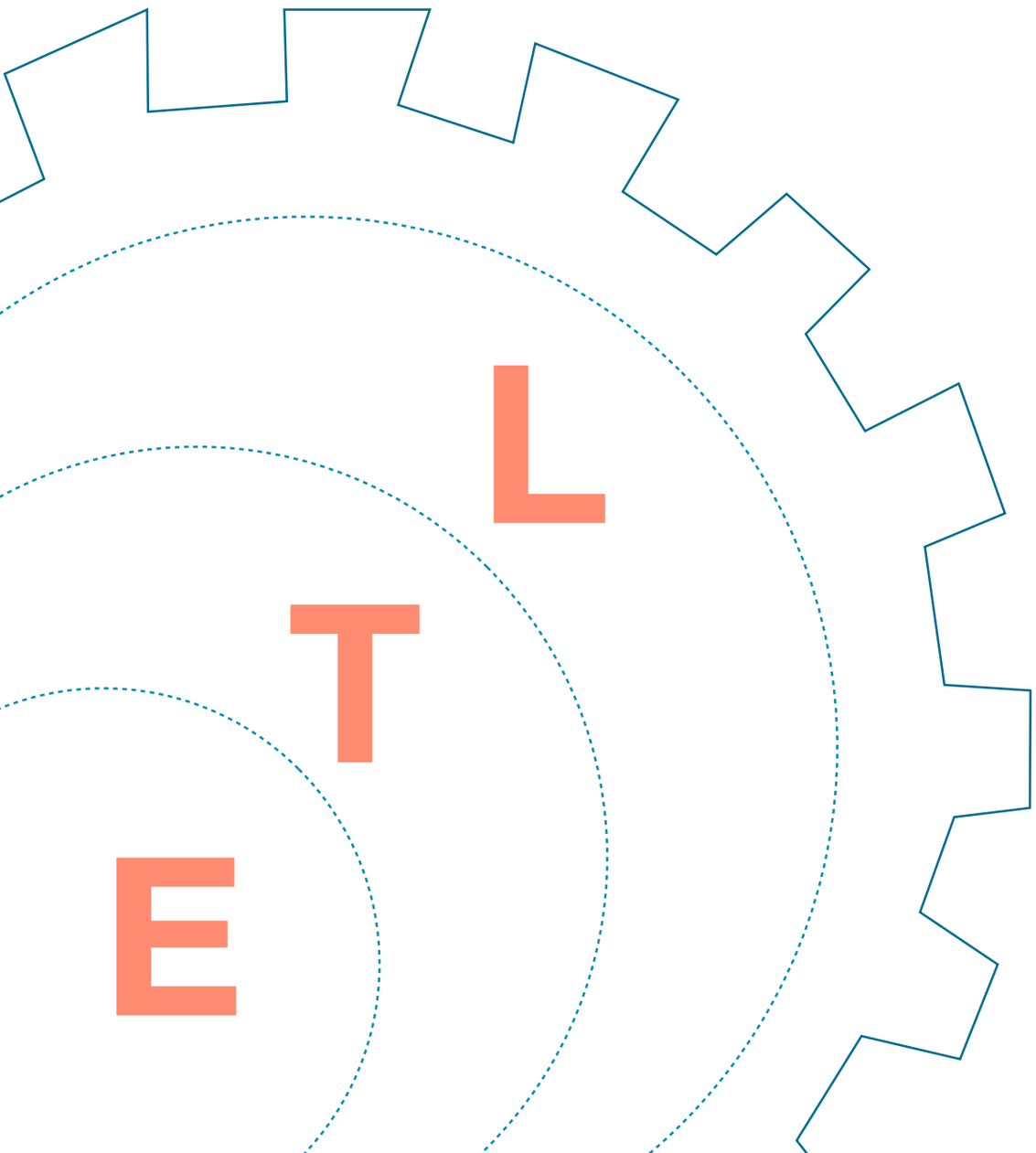
จากบทก่อน ๆ หน้า เราจะทราบว่าการทำงานหลักของคลังข้อมูลจะสามารถถูกแบ่งได้เป็น 3 ส่วนใหญ่ ๆ ด้วยกัน คือ

เมื่อเราทำการพิจารณาถึงการทำงานแต่ละส่วน เราจะทราบว่าการทำงานส่วนแรกและส่วนที่สอง ซึ่งก็คือ การได้มาซึ่งข้อมูลและการจัดเก็บข้อมูลจะเป็นการประมวลผลแบบเบื้องหลังที่ผู้ใช้จะไม่ทราบหรือรับรู้เกี่ยวกับการทำงานดังกล่าว โดยฟังก์ชันการทำงานทั้งสองจะประกอบไปด้วยฟังก์ชันการทำงานหลัก 3 ฟังก์ชันด้วยกัน นั่นคือ **“อีทีแอล *(ETL)*” ซึ่งย่อมาจาก “การสกัดข้อมูล การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล และการถ่ายโอนข้อมูล *(data Extraction, Transformation and Loading)”**

โดยการทำงานจะเป็นการทำงานทีละฟังก์ชันทำงานเรียงต่อกันเป็นลำดับซึ่งจะเริ่มจากการสกัด/ค้นคืนข้อมูลที่ต้องการจากแหล่งข้อมูลหรือระบบการดำเนินงาน (Extracting/Retrieving)

จากนั้นเป็นการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล (Transformation) ที่ได้จากขั้นตอนแรกตามกรรมวิธีต่าง ๆ หรือตามฟังก์ชันการประมวลผลข้อมูลเบื้องต้น (Data preprocessing) และท้ายสุดคือ การถ่ายโอน/เคลื่อนย้าย (Loading) ข้อมูลที่ได้จากขั้นตอนที่สองไปจัดเก็บไว้ในฐานข้อมูลของคลังข้อมูล ตามลำดับ





จากฟังก์ชันการทำงานทั้ง 3 ของอีทีแอล การสกัดข้อมูลจะเป็นขั้นตอนการทำงานแรกที่จะทำหน้าที่ในการสกัดข้อมูลจากแหล่งข้อมูลหรือระบบการดำเนินงานตามความต้องการของผู้ใช้ โดยก่อนที่เราจะทำการสกัดข้อมูลนั้นเราจะต้องทราบถึงความต้องการจากผู้ใช้ว่าต้องการสกัดข้อมูลอะไรบ้าง แล้วจึงค่อยลงมือทำการสกัดข้อมูล ซึ่งจากบทที่ 7 จะกล่าวถึงการสร้างแบบจำลองมิติต่างๆ (Dimensional model) จากความต้องการของผู้ใช้ที่อยู่ในรูปของแพคเกจของข้อมูล (Information package) จากเนื้อหาในบทดังกล่าวเราสามารถกล่าวได้ว่า “แบบจำลองมิติต่างๆจะเป็นแบบจำลองที่สะท้อนถึงข้อมูลจากความต้องการของผู้ใช้ ซึ่งข้อมูลดังกล่าวเป็นข้อมูลที่ควรจะจัดเก็บในคลังข้อมูล” โดยหลังจากที่เราทำการสร้างแบบจำลองมิติต่างๆเรียบร้อยแล้ว เราจะสามารถนำข้อมูลที่เก็บอยู่ในแต่ละ dimension tableและ fact table มาทำการค้นหา/เปรียบเทียบกับข้อมูลที่เก็บอยู่ในแหล่งข้อมูลเพื่อทำการสกัดข้อมูลต่อไป โดยในการสกัดข้อมูลนั้นเราสามารถเรียกได้อีกอย่างหนึ่งว่าเป็น “ขั้นตอนการเลือกข้อมูล” ที่จะทำการเลือกข้อมูลตามความต้องการของผู้ใช้เพื่อที่จะนำข้อมูลเหล่านั้นไปเก็บไว้ในคลังข้อมูลต่อไป

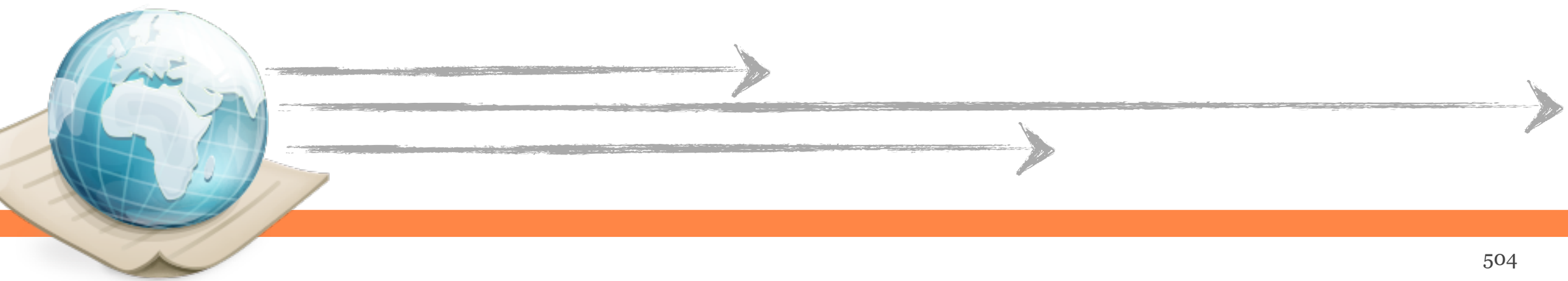
แต่ก่อนที่เราทำการสกัดข้อมูลเราควรที่จะต้องทำความเข้าใจเกี่ยวกับหลักการพื้นฐาน ข้อเท็จจริงหรือแนวปฏิบัติสำหรับการสกัดข้อมูลดังต่อไปนี้

- ข้อมูลจากแหล่งข้อมูลที่จะถูกสกัดหรือค้นคืนมีจำนวนมาก อาจมีขนาดตั้งแต่หลักร้อยเมกะไบท์จนกระทั่งหลายสิบกิกะไบท์ก็เป็นได้
- ระบบการดำเนินงานส่วนใหญ่จะถูกออกแบบสำหรับการสืบค้นข้อมูลจำนวนไม่มาก ซึ่งจะมีขนาดของข้อมูลไม่มากเท่ากับจำนวนข้อมูลที่ต้องทำการสกัดจากแหล่งข้อมูล ดังนั้นเราต้องให้ความระมัดระวังในการออกแบบการทำงานของการสกัดข้อมูลเพื่อไม่ทำให้ระบบการดำเนินงานทำงานช้าลง
- การสกัดข้อมูลควรที่จะต้องทำให้เร็วที่สุดเท่าที่จะเป็นไปได้
- ควรจะสกัดข้อมูลให้น้อยที่สุดเท่าที่จะเป็นไปได้
- ไม่ควรทำการสกัดข้อมูลจากแหล่งข้อมูลบ่อยๆ
- ควรจะทำการเปลี่ยนแปลงการทำงานของระบบดำเนินการให้น้อยที่สุดเท่าที่จะเป็นไปได้

จากข้อเท็จจริงและหลักการทั้งหมดข้างต้น สิ่งสำคัญที่สุดที่เราควรจะต้องคำนึงถึงในการสกัดข้อมูลก็คือ เราจะต้องออกแบบการสกัดข้อมูลให้มีการใช้งานหรือรบกวนการทำงานของแหล่งข้อมูล/ระบบการดำเนินงานให้น้อยที่สุดเท่าที่จะเป็นไปได้ ซึ่งจะช่วยให้การทำงานของระบบการดำเนินงานนั้นยังคงสามารถดำเนินไปได้อย่างปกติ

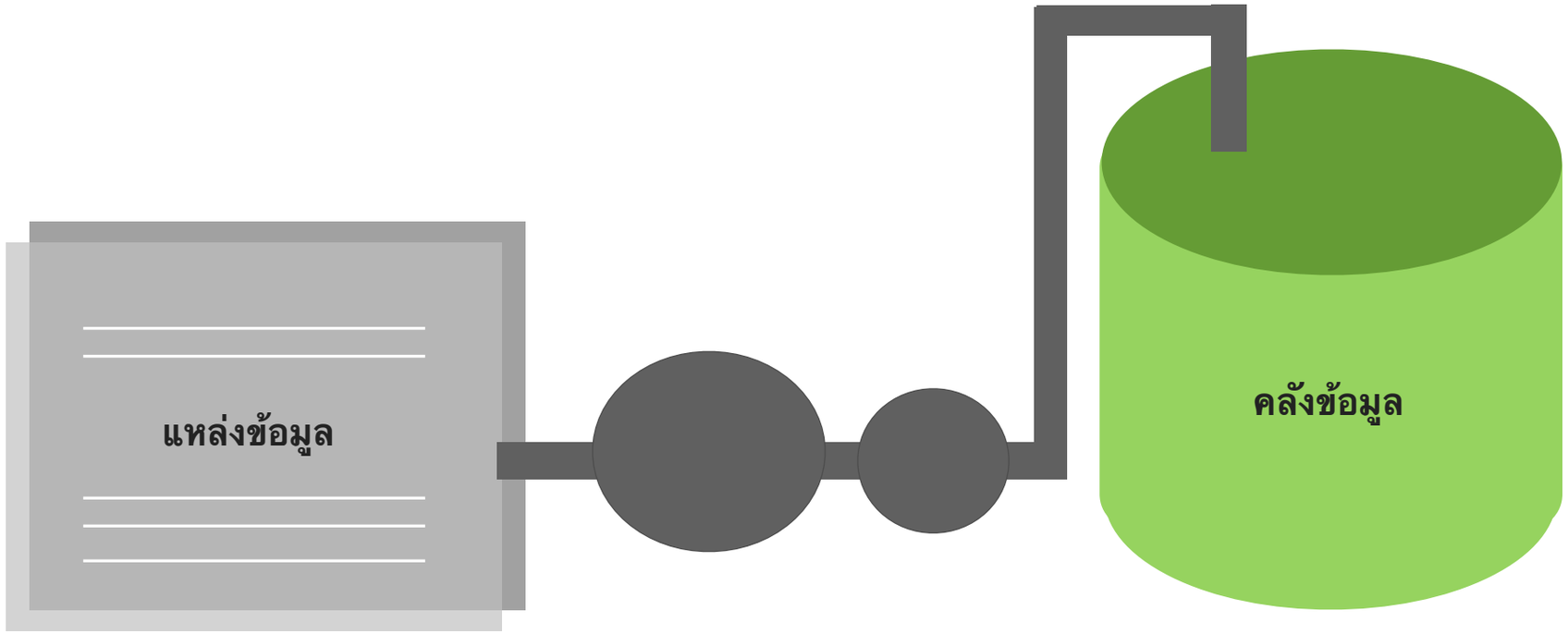
ในส่วนของฟังก์ชันที่สองซึ่งก็คือ การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลจะเริ่มการทำงานก็ต่อเมื่อข้อมูลที่ได้รับมาจากขั้นตอนการสกัดข้อมูลนั้นยังไม่มีคุณภาพหรือยังไม่เป็นมาตรฐานเพียงพอซึ่งข้อมูลที่สกัดได้ ซึ่งการด้อยคุณภาพของข้อมูลอาจมาจากข้อมูลที่มาจากหลายแหล่งข้อมูลที่มีความแตกต่างกันทั้งในแง่มุมของเทคโนโลยีหรือแพลตฟอร์มการคำนวณ ดังนั้น ฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลจะมีหน้าที่ในการทำความสะอาดข้อมูล รวมข้อมูลเข้าด้วยกัน ทำข้อมูลให้เป็นมาตรฐาน และอื่นๆ

ซึ่งจากฟังก์ชันการทำงานดังกล่าว เราสามารถกล่าวได้ว่าฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลเป็นฟังก์ชันที่ทำการประมวลผลข้อมูลเบื้องต้นก่อนที่จะนำข้อมูลเหล่านั้นไปเก็บไว้ในคลังข้อมูลต่อไป และหลังจากทำการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลแล้ว เราจะทำการถ่ายโอนข้อมูลเข้าสู่คลังข้อมูล ซึ่งฟังก์ชันการทำงานนี้เป็นอีกหนึ่งฟังก์ชันสุดท้ายของอีทีแอลที่ทำหน้าที่ในการถ่ายโอนข้อมูลที่ได้จากแหล่งข้อมูลและผ่านการประมวลผลแล้วไปจัดเก็บไว้ในคลังข้อมูลต่อไป ซึ่งในการถ่ายโอนข้อมูลเราจะต้องหาช่วงเวลาที่เหมาะสมโดยจะต้องไม่กระทบต่อการทำงานของระบบการดำเนินงานและระบบคลังข้อมูลมากนัก



จากที่กล่าวข้างต้น เราสามารถสรุปได้ว่า **“อีทีแอล”** นั้นเป็นขั้นตอนการโหลด/ถ่ายโอนข้อมูลจากระบบการดำเนินงานเข้าสู่คลังข้อมูล โดยการทำงานจะเริ่มจากการสกัดข้อมูลที่ผู้ใช้ต้องการจากแหล่งข้อมูลจากนั้นทำการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลให้มีคุณภาพหรือเป็นมาตราฐาน แล้วจึงทำการถ่ายโอนข้อมูลเข้าสู่คลังข้อมูล ซึ่งจากการทำงานทั้งหมดของคลังข้อมูล ฟังก์ชัน “อีทีแอล” จะเป็นขั้นตอนที่สำคัญมากในการสร้างคลังข้อมูลและเป็นขั้นตอนที่ใช้เวลามากที่สุดในการสร้างคลังข้อมูล

ดังนั้นในการสร้างคลังข้อมูล เราจำเป็นที่จะต้องออกแบบฟังก์ชันอีทีแอลให้มีประสิทธิภาพมากที่สุด ซึ่งจะทำให้คลังข้อมูลที่เราสร้างขึ้นมีประสิทธิภาพที่ดีในการทำงานตามไปด้วย



SECTION 3

สถาปัตยกรรมของอีทีแอล

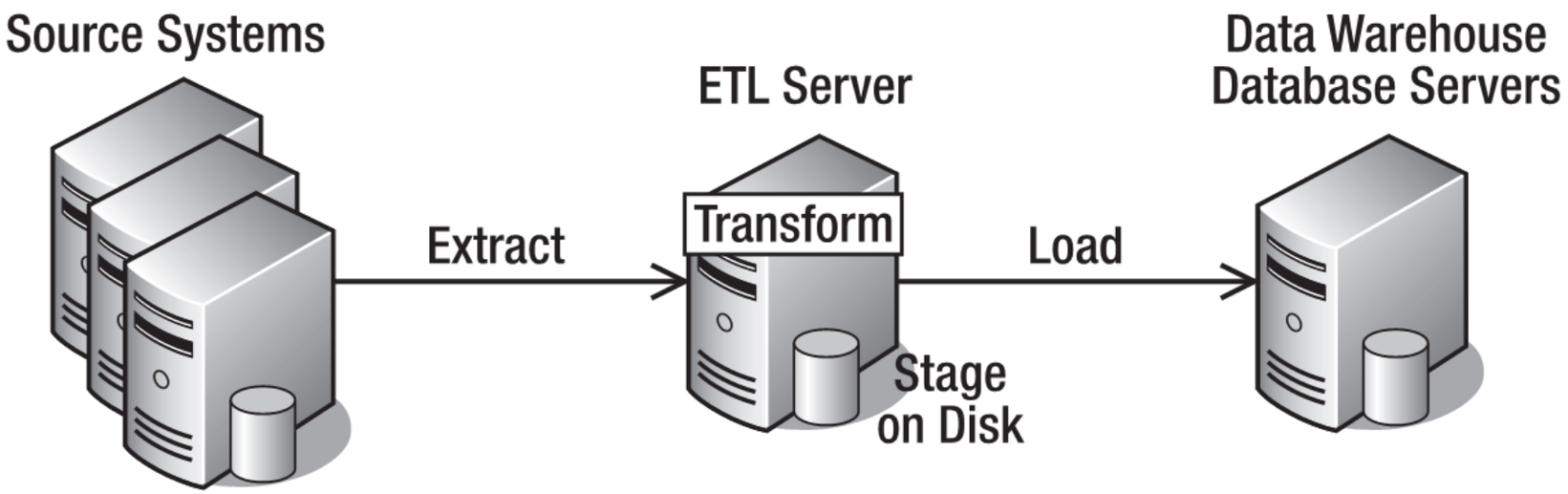
สถาปัตยกรรมของอีทีแอล

ขั้นตอนพื้นฐานของฟังก์ชัน “อีทีแอล” จะเริ่มจากการสกัดข้อมูลที่ต้องการเพียงบางส่วนจากแหล่งข้อมูลต่าง ๆ แล้วจึงทำการประมวลผลกับข้อมูลนั้น ๆ เช่น การรวบรวมข้อมูลเข้าด้วยกัน การแยกข้อมูลออกจากกัน การทำความสะอาดข้อมูล การปรับรูปแบบของข้อมูลและอื่นๆ โดยหลังจากทำการประมวลผลเบื้องต้นข้อมูลแล้วจะทำการถ่ายโอนข้อมูลที่ถูกประมวลผลเหล่านั้นเข้าสู่คลังข้อมูล ซึ่งจากการทำงานดังกล่าว เราควรที่จะต้องศึกษาเกี่ยวกับสถาปัตยกรรมที่เกี่ยวข้องกับฟังก์ชัน “อีทีแอล” ซึ่งจะมีอยู่ 2 ประเภทคือ

1. การใช้ staging area ในการสกัดข้อมูล
2. การสกัดข้อมูลโดยใช้หน่วยความจำ

1 การใช้ staging area ในการสกัดข้อมูล

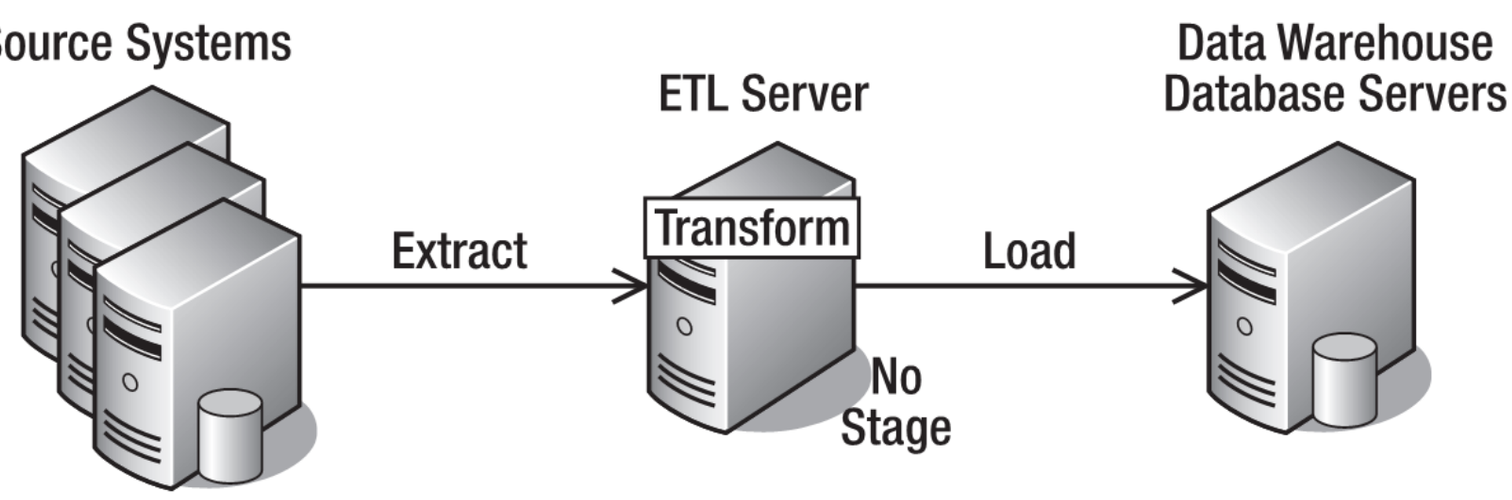
การใช้ staging area ในการสกัดข้อมูล กล่าวคือ หลังจากทำการสกัด/เลือกข้อมูลที่ต้องการเพียงบางส่วนจากแหล่งข้อมูล/ระบบการดำเนินงานแล้ว จะทำการถ่ายโอนข้อมูลเหล่านั้นไปยัง “staging area” (ดังแสดงในรูปที่ 8-1) โดยที่ staging area หรือที่เรียกอีกอย่างหนึ่งว่า “data staging” นั้นเปรียบได้กับพื้นที่ที่ใช้สำหรับพักข้อมูล ที่จะใช้แฟ้มข้อมูลหรือฐานข้อมูลเพื่อจัดเก็บข้อมูลที่ถูกสกัดมาจากแหล่งข้อมูล จากนั้น ณ ที่ staging area ข้อมูลที่อยู่ในแต่ละแฟ้มข้อมูลหรือฐานข้อมูลจะถูกประมวลผลหรือทำการเปลี่ยนแปลง/เปลี่ยนรูป เพื่อทำให้ข้อมูลมีความถูกต้อง ครบถ้วน สมบูรณ์และเป็นมาตราฐาน แล้วจึงถ่ายโอนข้อมูลเหล่านั้นไปยังคลังข้อมูลต่อไป



รูปที่ 8-1 การสกัดข้อมูลโดยการใช้ staging area

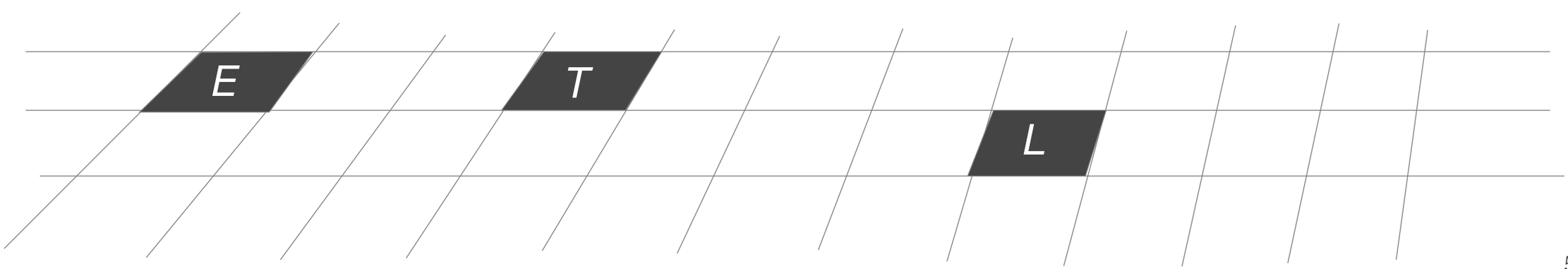
2 การสกัดข้อมูลโดยใช้หน่วยความจำ

การสกัดข้อมูลโดยใช้หน่วยความจำ กล่าวคือ การสกัดข้อมูลที่ต้องการเพียงบางส่วนจากแหล่งข้อมูล แล้วทำการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลในหน่วยความจำ (ดังแสดงในรูปที่ 8-2) แล้วจึงทำการถ่ายโอนข้อมูลไปยังคลังข้อมูลต่อไป



รูปที่ 8-2 การสกัดข้อมูลโดยการใช้หน่วยความจำ

จากสถาปัตยกรรมทั้งสองข้างต้น การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลในหน่วยความจำจะสามารถทำงานได้เร็วกว่าการเปลี่ยนแปลงข้อมูลโดยใช้ staging area ถ้าข้อมูลมีขนาดเล็กเราสามารถทำการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลในหน่วยความจำได้เลย ซึ่งการประมวลผลข้อมูลในหน่วยความจำจะสามารถลดการอ่าน/เขียนข้อมูลลงในแฟ้มข้อมูลหรือลงในฐานข้อมูลที่เก็บอยู่ใน staging area ได้ แต่ถ้าข้อมูลมีขนาดใหญ่มากเราจะต้องทำการเขียนข้อมูลลงใน staging area ก่อน เนื่องจากขนาดของหน่วยความจำมีขนาดค่อนข้างเล็กซึ่งอาจไม่สามารถรองรับข้อมูลปริมาณมากได้ ดังนั้นก่อนที่จะทำการเลือกใช้สถาปัตยกรรมสำหรับฟังก์ชัน “อีทีแอล” เราจะต้องคำนึงถึงปริมาณข้อมูลที่ต้องทำการสกัดจากแหล่งข้อมูลเป็นลำดับแรก



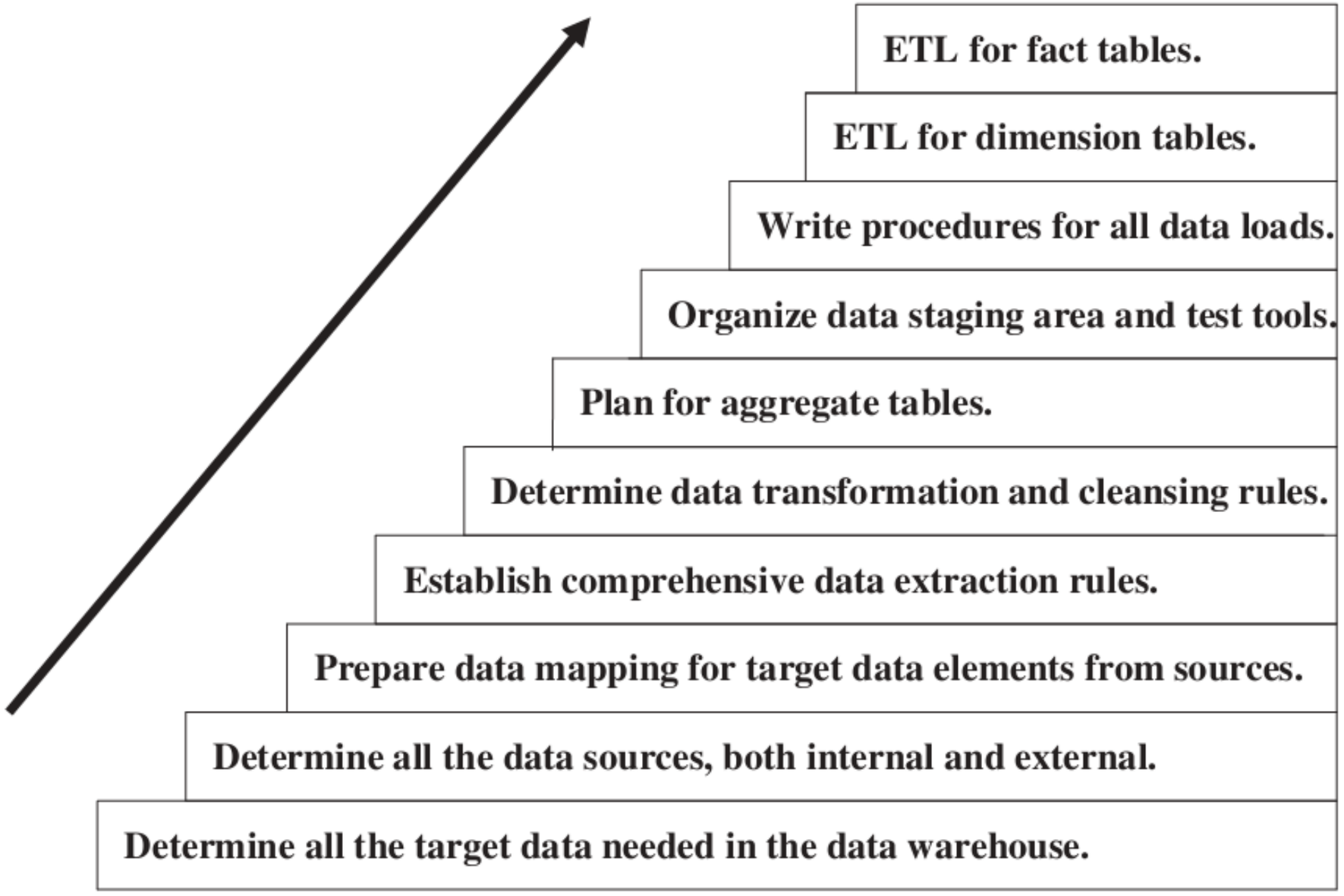
SECTION 4

ขั้นตอนการทำงานของอีทีแอล

อย่างที่เราทราบเบื้องต้นว่าขั้นตอนการทำการของฟังก์ชัน “อีทีแอล”จะประกอบไปด้วย 3 ฟังก์ชันหลัก คือ การสกัดข้อมูล การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล และการถ่ายโอนข้อมูล แต่โดยแท้จริงแล้วรายละเอียดของขั้นตอนดังกล่าวยังมีอยู่อีกมากโดยสามารถแจกแจงรายละเอียดขั้นตอนการทำงานทั้งหมดได้ดังรูปที่ 8-3 ซึ่งจากรูปจะแสดงขั้นตอนหลักในการสร้างฟังก์ชัน“อีทีแอล” ที่มีการเพิ่มรายละเอียดต่างๆ และแสดงถึงกิจกรรมต่าง ๆ ดังต่อไปนี้

- ☑ การวางแผนสำหรับการรวมรวมข้อมูลเข้าสู่ fact table
- ☑ การกำหนดกฎสำหรับการเปลี่ยนแปลง/เปลี่ยนรูปและการทำความสะอาด
- ☑ การสร้างกฎในการสกัดข้อมูล
- ☑ การเตรียมการเชื่อมโยงข้อมูลจากแหล่งข้อมูลเข้าสู่ dimension หรือ fact table
- ☑ การรวบรวมข้อมูลจากแหล่งข้อมูลต่าง ๆ ทั้งแหล่งข้อมูลภายในและภายนอก
- ☑ การกำหนดข้อมูลทั้งหมดที่ต้องการเก็บไว้ในคลังข้อมูล
- ☑ การรวมโครงสร้างข้อมูลจากหลายแหล่งข้อมูลไปเป็นข้อมูลเพียง row เดียวใน dimension หรือ fact table ที่ถูกเก็บไว้ในคลังข้อมูล

- ☑ การแยกโครงสร้างข้อมูลหนึ่ง ๆ ไปเป็นข้อมูลที่มีหลายโครงสร้างเพื่อสร้างเป็นข้อมูลหลายๆ row ที่ถูกเก็บไว้ในคลังข้อมูล
- ☑ การอ่านข้อมูลจากพจนานุกรมข้อมูลที่ถูกเก็บไว้ในแหล่งข้อมูล
- ☑ การอ่านข้อมูลจากหลาย ๆ แหล่งข้อมูล อาทิเช่น flat file, indexed file และระบบฐานข้อมูล
- ☑ การโหลดรายละเอียดสำหรับสร้าง fact table
- ☑ การรวบรวมข้อมูลหรือทำผลสรุปข้อมูลให้กับ fact tables
- ☑ การแปลงข้อมูลจากรูปแบบหนึ่งของข้อมูลไปเป็นอีกรูปแบบหนึ่งของ dimension หรือ fact table
- ☑ การรับเอาข้อมูลที่เป็นเป้าหมายจาก field ต่างๆของแหล่งข้อมูล ตัวอย่างเช่น อายุจากวันเกิด
- ☑ การเปลี่ยนข้อมูลที่มีความกำกวมให้มีความหมายมากขึ้น ตัวอย่างเช่น 1 และ 2 หมายถึงเพศชายและหญิง ตามลำดับ



รูปที่ 8-3 ขั้นตอนการทำงานหลักของ ETL

SECTION 5

การสกัดข้อมูล

การสกัดข้อมูล

ในการสกัดข้อมูลจากระบบการดำเนินงาน เราจะต้องให้ความใส่ใจกับระบบการดำเนินงานของธุรกิจที่ซึ่งในปัจจุบันระบบเหล่านี้กระจายตัวอยู่ตามที่ต่าง ๆ ทั่วโลกและข้อมูลในระบบการดำเนินงานอาจเกิดความเปลี่ยนแปลงเกิดขึ้นได้ตลอด เช่น การเพิ่ม ลบหรืออัพเดทข้อมูล เป็นต้น ดังนั้นเมื่อเราทำการออกแบบการสกัดข้อมูลเราจะต้องให้ความสนใจกับสถานะของข้อมูลทั้งในระบบการดำเนินงาน และสถานะของคลังข้อมูลด้วย ซึ่งในตอนเริ่มต้นหลังจากทำการสร้างฐานข้อมูลและตารางต่าง ๆ ตามแบบจำลองมิติต่าง ๆ แล้ว คลังข้อมูลจะยังไม่มีข้อมูลอยู่เลย โดยเราจะต้องทำการสกัดข้อมูลที่ต้องการทั้งหมดจากระบบการดำเนินงานเพื่อทำการถ่ายโอนข้อมูลเหล่านั้นไปยังคลังข้อมูล ซึ่งวิธีการถ่ายโอนข้อมูลจากการสกัดข้อมูลเข้าสู่คลังข้อมูลครั้งแรกจะเรียกว่า “Full load” หรือ “Initial load”

ต่อมาเมื่อระบบการดำเนินงานมีข้อมูลเพิ่มขึ้นหรือมีการเปลี่ยนแปลง/อัพเดทข้อมูล เราจะต้องทำการสกัดข้อมูลที่ถูกเพิ่มขึ้นหรืออัพเดทเหล่านั้น เข้าสู่คลังข้อมูลในครั้งต่อๆ ไป ซึ่งวิธีการนี้เราจะเรียกว่า “(Ongoing) Incremental load” ซึ่งจากวิธีการดังกล่าว เราจะต้องทำการตั้งเวลาหรือกำหนดช่วงเวลาสำหรับการสกัดข้อมูลและทำการถ่ายโอนข้อมูล เช่น การสกัดข้อมูลสัปดาห์ละครั้ง เป็นต้น ซึ่งจากวิธีการดังกล่าว เราจะต้องทำการเสาะหาข้อมูลที่ถูกเพิ่มขึ้นหรือมีการเปลี่ยนแปลงในช่วงหนึ่งสัปดาห์ แล้วทำการสกัดข้อมูลที่ต้องการจากข้อมูลเหล่านั้น แล้วจึงถ่ายโอนไปยังคลังข้อมูลต่อไป

จากการถ่ายโอนข้อมูลทั้งสองวิธีข้างต้น จะทำให้การสกัดข้อมูลมีขั้นตอนการสกัดข้อมูลที่มีความยุ่งยากและซับซ้อนเพิ่มขึ้น ดังนั้นในการออกแบบฟังก์ชันการทำงานของคลังข้อมูล เราจะต้องพิจารณาถึงขั้นตอนการทำงานของวิธีการถ่ายโอนข้อมูลทั้งสอง และจำเป็นต้องพิจารณาขั้นตอนและปัจจัยต่าง ๆ ดังต่อไปนี้

Source Identification—การระบุแหล่งข้อมูลและโครงสร้างของแหล่งข้อมูล ว่ามีข้อมูลที่เราต้องการหรือสนใจอยู่ที่ใดบ้าง เช่น อยู่ที่เซิร์ฟเวอร์ใด ฐานข้อมูลใด ตารางใด แอทริบิวใด ตามลำดับ

Method for extraction—การกำหนดวิธีที่จะใช้ในการสกัดข้อมูลสำหรับแต่ละแหล่งข้อมูล ว่าจะใช้วิธีการใดระหว่างสร้างฟังก์ชันการสกัดข้อมูลเองหรือ การใช้เครื่องมือในการสกัดข้อมูล

Extraction frequency—การกำหนดความถี่หรือความบ่อยในการสกัดข้อมูลสำหรับแต่ละแหล่งข้อมูล เช่น ทำการสกัดข้อมูลวันละ 1 ครั้ง อาทิตย์ละ 1 ครั้ง หรือ ไตรมาสละ 1 ครั้ง เป็นต้น

Time window—การกำหนดช่วงเวลาที่เหมาะสมที่จะทำการสกัดข้อมูลสำหรับแต่ละแหล่งข้อมูล เช่น 1.00 – 3.00 นาฬิกา ซึ่งเป็นช่วงเวลาที่มีคนใช้ระบบการดำเนินงานค่อนข้างน้อย เป็นต้น

Job sequencing—การกำหนดลำดับการทำงานว่างานไหนในการสกัดข้อมูลควรทำเป็นลำดับแรกจากการทำงานทั้งหมด ซึ่งในขณะที่ขั้นตอนแรกทำงานขั้นตอนอื่น ๆ จะต้องรอเป็นลำดับ

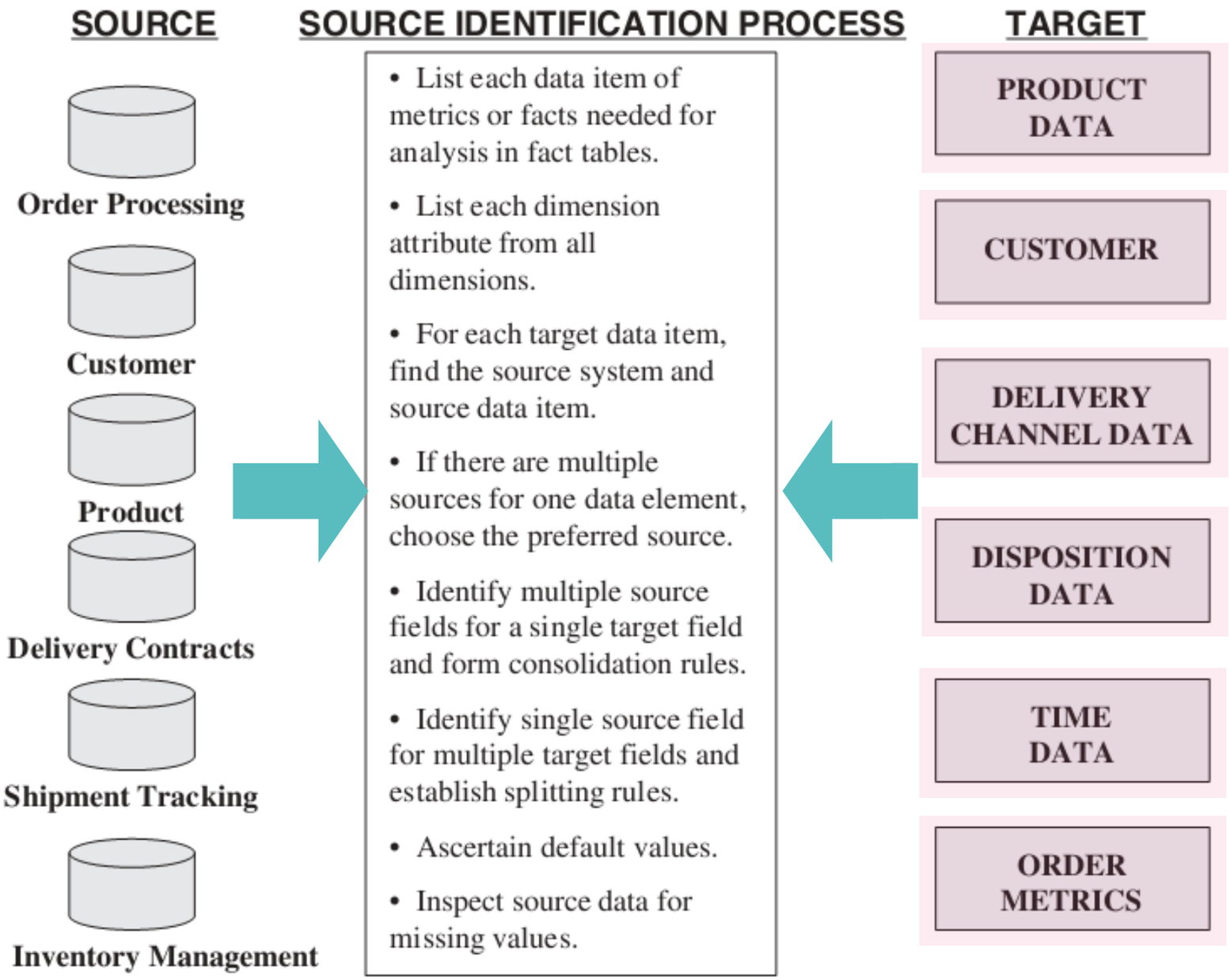
Exception handling—กำหนดวิธีในการจัดการกับเรคคอร์ดที่ไม่สามารถสกัดได้ หรือเหตุการณ์ผิดปกติที่อาจเกิดขึ้น

การระบุแหล่งข้อมูลที่สนใจ
(Source Identification)

การระบุแหล่งข้อมูลจะเป็นการระบุถึงแหล่งข้อมูลอินพุตที่ต้องการว่าข้อมูลเหล่านั้นปรากฏหรือถูกจัดเก็บอยู่ในแหล่งข้อมูลใดบ้าง รวมถึงการทดสอบและการทวนสอบแหล่งข้อมูลที่ต้องทำการติดต่อว่ามีข้อมูลที่จำเป็นต่อการสร้างคลังข้อมูลหรือไม่ ซึ่งการระบุแหล่งข้อมูลนั้นมักจะนิยมใช้ในระบบที่มีแบ่งส่วนการเก็บข้อมูลออกเป็นส่วน ๆ ตามดาต้ามาร์ทต่าง ๆ (ดังแสดงในรูปที่ 8-4) ซึ่งจะประกอบไปด้วยขั้นตอนต่าง ๆ ดังนี้

- ทำการกำหนดข้อมูลที่เป็นตัวชี้วัดหรือข้อเท็จจริงที่ต้องการสำหรับวิเคราะห์ข้อมูลใน fact table
- ทำการกำหนดข้อมูลแต่ละแอทริบิวที่เกี่ยวกับข้องทุก ๆ dimension table
- ทำการหาแหล่งข้อมูลกับข้อมูลที่จะเก็บใน dimension และ fact table
- ในกรณีที่ข้อมูลหนึ่งๆ ถูกเก็บอยู่ในหลายแหล่งข้อมูล ต้องทำการเลือกแหล่งข้อมูลหนึ่งแหล่งที่จะทำการสกัดข้อมูล
- ทำการระบุว่าข้อมูลฟิลด์หนึ่งๆ ที่ต้องการเก็บไว้ในคลังข้อมูลนั้น มาจากหลายฟิลด์ของแหล่งข้อมูลหรือไม่ จากนั้นทำการสร้างกฎการรวมข้อมูล (Consolidation rules)

- ทำการระบุว่าข้อมูลหลาย ๆ ฟิลด์ที่ต้องการเก็บไว้ในคลังข้อมูลนั้นมาจากข้อมูลเพียงฟิลด์เดียวของแหล่งข้อมูลหรือไม่ จากนั้นทำการสร้างกฎการแยกข้อมูล (Splitting rules)
- ทำการกำหนดค่า default value สำหรับข้อมูลแต่ละ field ที่ต้องการเก็บไว้ในคลังข้อมูล
- ทำการตรวจสอบการขาดขายไปของข้อมูล



รูปที่ 8-4 ขั้นตอนการระบุแหล่งข้อมูลโดยละเอียด

เทคนิคในการสกัดข้อมูล

ก่อนที่จะทำการสกัดข้อมูลเราต้องทำความเข้าใจเกี่ยวกับธรรมชาติของข้อมูลที่ถูกเก็บอยู่ในระบบดำเนินการรวมถึงโครงสร้างการจัดเก็บข้อมูลด้วยเพื่อที่จะสามารถสกัดข้อมูลได้อย่างมีประสิทธิภาพ ซึ่งโดยปกติแล้ว ข้อมูลที่ถูกเก็บอยู่ในระบบทั่ว ๆ ไปจะมีการเปลี่ยนแปลงไปตามกาลเวลา

ตัวอย่างเช่น
ลูกค้าของบริษัทแห่งหนึ่งต้องย้ายที่อยู่จากรัฐนิวยอร์กไปยังแคลิฟอร์เนีย ระบบสำหรับจัดเก็บข้อมูลลูกค้าจะต้องทำการอัพเดทข้อมูลที่อยู่ของลูกค้า ถ้าโครงสร้างข้อมูลของระบบไม่มีการเก็บประวัติก่อนหน้า ระบบจะลบข้อมูลที่อยู่ในรัฐนิวยอร์กออกแล้วแทนที่ด้วยข้อมูลที่อยู่ในรัฐแคลิฟอร์เนีย ซึ่งการเก็บข้อมูลลักษณะนี้จะส่งผลกระทบโดยตรงต่อข้อมูลในคลังข้อมูล เช่น ถ้าเราต้องการวิเคราะห์ยอดขายในรัฐหนึ่ง ๆ ข้อมูลการซื้อของลูกค้า ณ ตอนที่อยู่ที่รัฐนิวยอร์กควรจะถูกรวมเป็นยอดขายสำหรับรัฐนิวยอร์กและข้อมูลการซื้อของลูกค้า ณ ปัจจุบันควรจะถูกรวมเป็นยอดขายของรัฐแคลิฟอร์เนียแทน

จากตัวอย่าง เราจะเห็นได้ว่าการเก็บประวัติย้อนหลังของข้อมูลนั้นเป็นสิ่งที่ไม่อาจละเลยได้ในคลังข้อมูล นี้จึงเป็นคำถามที่ว่าเราจะสามารถเก็บประวัติย้อนหลังของข้อมูลจากแหล่งข้อมูลได้อย่างไร?—เพื่อที่จะตอบคำถามเราจำเป็นต้องเข้าใจก่อนว่าข้อมูลนั้นถูกเก็บอยู่ในแหล่งข้อมูลอย่างไรและข้อมูลจากแหล่งข้อมูลนั้นมีลักษณะอย่างไร?

ประเภทของข้อมูลในระบบการดำเนินงาน

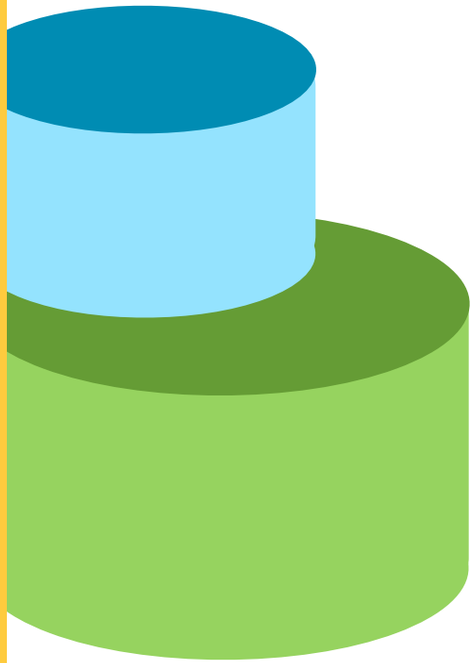
โดยส่วนใหญ่แล้วข้อมูลที่ถูกเก็บในระบบดำเนินการสามารถแบ่งออกเป็น 2 ประเภท ดังแสดงตัวอย่างในรูปที่ 8-5 ที่ประกอบไปด้วย (1) ข้อมูล ณ ปัจจุบัน (Current value) และ (2) ข้อมูลที่มีการเปลี่ยนแปลงไปตามการเวลา (Periodic status) ซึ่งจากชนิดของข้อมูลที่ปรากฏในระบบการดำเนินงาน เราควรที่จะต้องออกแบบ/กำหนดเทคนิคที่จะใช้ในการสกัดข้อมูลให้สอดคล้องกับชนิดของข้อมูลที่ถูกเก็บอยู่ในระบบการดำเนินงานด้วย ลองพิจารณารายละเอียดของข้อมูลแต่ละประเภทที่เกิดขึ้นในระบบการดำเนินงานดังต่อไปนี้

ข้อมูล ณ ปัจจุบัน (Current Value)

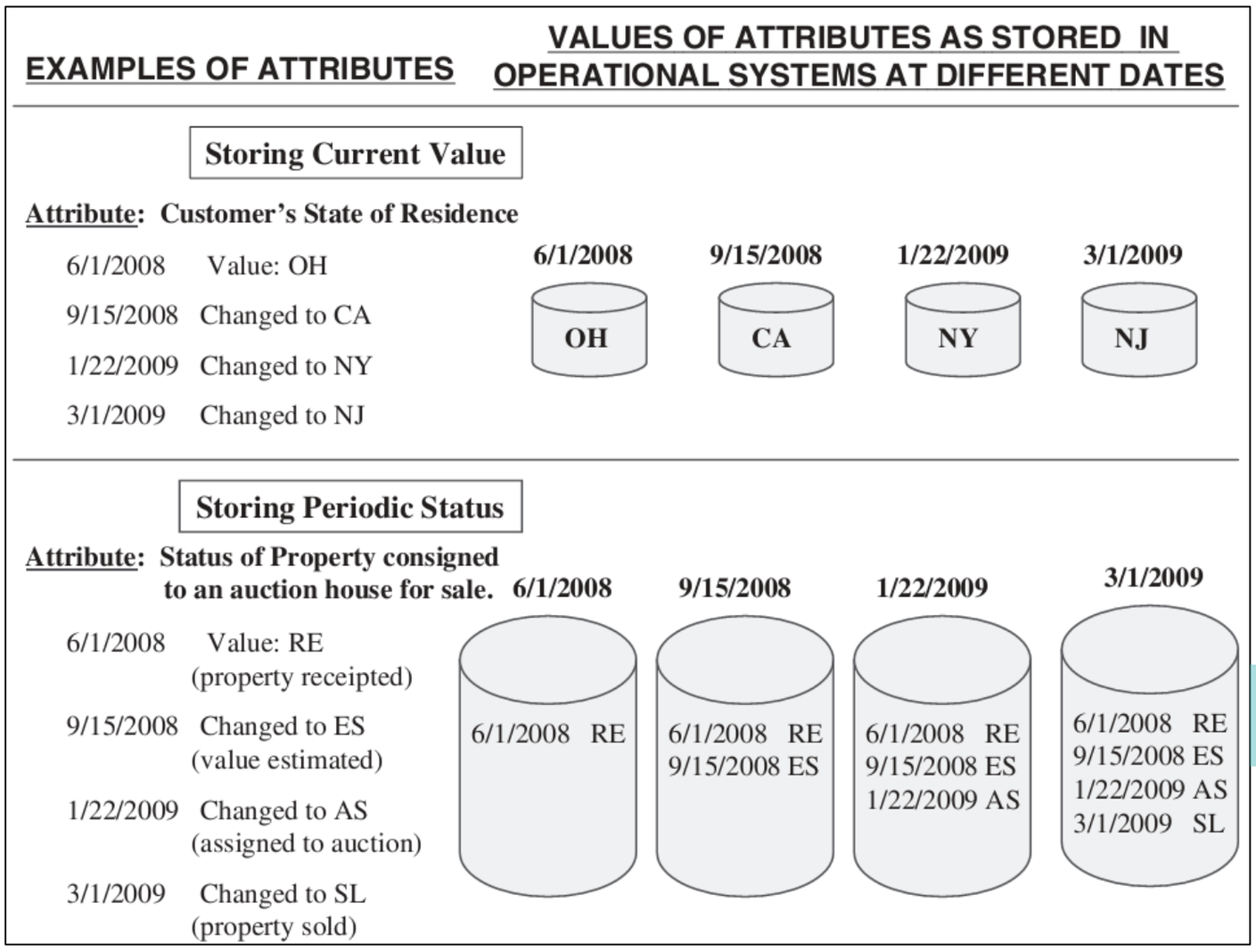
ในระบบการดำเนินงานทั่วๆ ไปแอทริบิวส่วนใหญ่จะมีข้อมูลที่เป็นปัจจุบันที่แสดงถึงข้อมูล ณ เวลานั้นๆ และเป็นข้อมูลแบบชั่วคราวไม่ยั่งยืนสามารถเปลี่ยนแปลงได้ตลอดซึ่งเราไม่สามารถคาดเดาได้ว่าค่าของข้อมูลที่เก็บนั้นจะคงอยู่นานเท่าไหร่ ? หรือจะมีการเปลี่ยนแปลงเมื่อไร ? ข้อมูลเหล่านี้จะยังคงไม่มีการเปลี่ยนแปลงจนกระทั่งมีการเกิดขึ้นของธุรกรรมทางธุรกิจที่ทำการเปลี่ยนแปลงข้อมูลนั้น ๆ ตัวอย่างของข้อมูลที่เป็นปัจจุบันที่เราสามารถพบได้บ่อย คือ ชื่อและที่อยู่ของลูกค้า ยอดเงินในบัญชี และอื่นๆ

ข้อมูลที่มีการเปลี่ยนแปลงไปตามกาลเวลา (Periodic Status)

ข้อมูลลักษณะนี้จะมีการเก็บสถานะของข้อมูลเมื่อมีการเปลี่ยนแปลงเกิดขึ้นและรวมถึงการเพิ่มข้อมูลเข้าสู่ฐานข้อมูล ในแต่ละครั้ง โดยที่ในแต่ละครั้งที่มีการเปลี่ยนแปลงเกิดขึ้นเราจะต้องทำการเก็บเวลาที่ใช้ในการอ้างอิงถึงการเปลี่ยนแปลงหรือการเพิ่มข้อมูลด้วย ซึ่งเวลาที่ถูกจัดเก็บไว้จะสามารถบอกได้ถึงลำดับการเกิดขึ้นข้อมูลในฐานข้อมูลและสถานะปัจจุบันของข้อมูลได้อีกด้วย โดยในการเก็บข้อมูลที่มีการเปลี่ยนแปลงไปตามการเวลานั้นจะเป็นการเก็บข้อมูลเหตุการณ์ที่เกิดขึ้นโดยจะมีเวลาที่เกี่ยวข้องเกิดขึ้นด้วย โดยที่การเก็บสถานะต่าง ๆ ของข้อมูลนั้นจะเก็บแยกไว้ในอีกฟิลด์หรือแอทริบิวหนึ่ง ๆ โดยการเก็บข้อมูลลักษณะนี้จะช่วยให้การสกัดข้อมูลที่ต้องการการวิเคราะห์หาสถิติย้อนหลังของข้อมูลสามารถทำงานได้ง่ายขึ้น



จากตัวอย่างชนิดของข้อมูลที่มักปรากฏในระบบการดำเนินงาน ดังแสดงในรูปที่ 8-5 เราจะสามารถเข้าใจถึงลักษณะของข้อมูลที่ถูกเก็บอยู่ในแหล่งข้อมูล ซึ่งจากวิธีในการถ่ายโอนเข้าสู่คลังข้อมูลที่อธิบายข้างต้น (Full load และ incremental load) เราจะต้องดำเนินการกับข้อมูลที่เป็นปัจจุบันและข้อมูลที่มีการเปลี่ยนแปลงตามลำดับ โดยในการถ่ายโอนข้อมูลนั้น เราจะสามารถทำการสกัดข้อมูลจากระบบการดำเนินงานได้ 2 รูปแบบหลัก ๆ คือ **1) “As is” และ 2) “Data of revision”**



รูปที่ 8-5 ชนิดของข้อมูลในระบบการดำเนินงาน

“As is”

“As is” หรือ **static data** คือ การดักจับ/เข้าถึงข้อมูล ณ เวลาช่วงเวลาหนึ่งๆที่กำหนด ซึ่งการทำงานของ “as is” จะคล้ายกับการทำ snapshot กับข้อมูลที่เกี่ยวข้อง ณ ช่วงเวลาหนึ่ง ๆ โดยข้อมูลที่เราจะทำการดักจับนั้นจะเป็นข้อมูล ณ ปัจจุบัน หรือเป็นข้อมูลชั่วคราวที่สามารถเปลี่ยนแปลงได้ นอกจากนั้นยังทำการดักจับข้อมูลที่มีการเปลี่ยนแปลงไปตามกาลเวลาที่จะดักจับข้อมูลที่ถูกเพิ่มเข้าสู่ระบบและข้อมูลที่ถูกเปลี่ยนแปลง ในช่วงเวลาที่กำหนด วิธีการสกัดข้อมูลแบบ “as is” มักถูกใช้ในการถ่ายโอนข้อมูลครั้งแรก (initial load) หรือการถ่ายโอนข้อมูลทั้งหมดจากระบบการดำเนินงานเข้าสู่คลังข้อมูล (full refresh)

ตัวอย่างเช่น เมื่อเวลาผ่านไป มีการเปลี่ยนแปลงชื่อสินค้าในระบบการดำเนินงาน ดังนั้นถ้าเราทำการถ่ายโอนข้อมูลชื่อสินค้าเข้าสู่คลังข้อมูลทั้งหมดอีกรอบหนึ่ง โดยไม่เลือกว่าข้อมูลใดเคยถูกถ่ายโอนข้อมูลเข้าสู่คลังข้อมูลแล้ว จะช่วยให้การทำงานและการจัดการกับการทำงานสามารถดำเนินการได้โดยง่าย ดังนั้น เราจะสามารถใช้ “as is” ในการสกัดข้อมูลได้ก็ต่อเมื่อเราต้องการถ่ายโอนข้อมูลทั้งหมดเข้าสู่คลังข้อมูลอีกครั้งหนึ่ง

“Data of revisions”

“Data of revisions” เป็นการสกัดข้อมูลแบบ incremental ซึ่งเป็นการสกัดข้อมูลที่มีการแก้ไขนับตั้งแต่การสกัดข้อมูลครั้งล่าสุด ถ้าข้อมูลจากแหล่งข้อมูลเป็นแบบข้อมูลชั่วคราว (เป็นข้อมูลไม่มีการเก็บสถานะของการเปลี่ยนแปลงของข้อมูล) การสกัดข้อมูลจะสามารถทำได้ยาก แต่ในทางกลับกัน ถ้าข้อมูลเป็นข้อมูลที่มีการเปลี่ยนแปลงไปตามกาลเวลา (เก็บสถานะของการเปลี่ยนแปลงด้วย) การสกัดข้อมูลจะสามารถทำได้ง่าย โดยเราสามารถทราบถึงความเปลี่ยนแปลงของข้อมูลได้จากสถานะหรือเวลาที่ถูกเก็บไว้ในระบบ เป็นต้น

การสกัดข้อมูลหรือดักจับแบบ incremental นั้นอาจจะสามารถทำได้แบบทันที (Immediate data extraction) หรือแบบรอเวลา (Deferred data extraction) ก็ได้ ซึ่งการทำงานแบบแรกจะมี 3 ทางเลือกที่แตกต่างกัน ในขณะที่การทำงานแบบที่ 2 จะการทำงาน 2 แบบที่แตกต่างกัน ดังต่อไปนี้

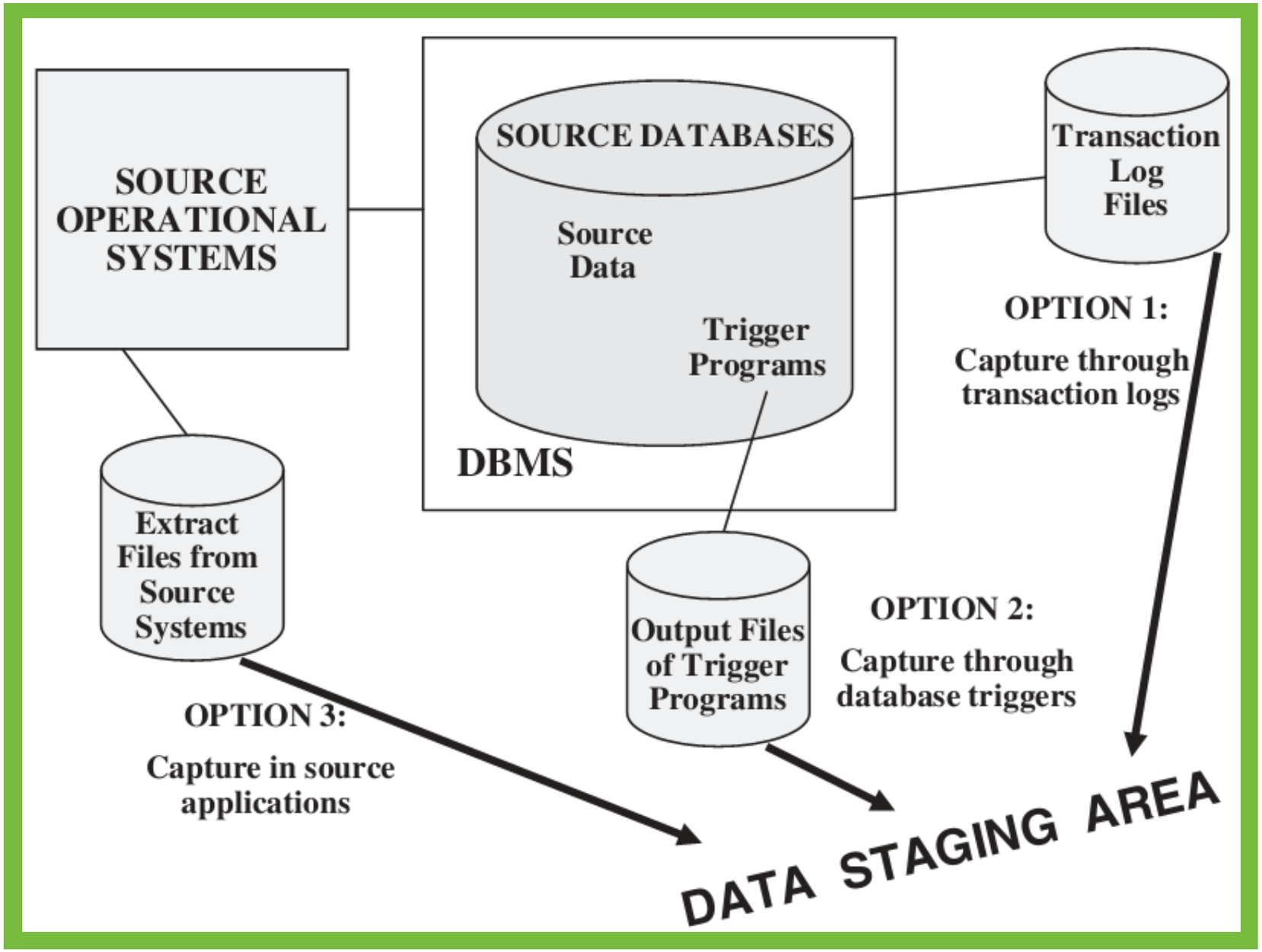
การสกัดข้อมูลแบบทันที
(Immediate Data Extraction)

คือ การสกัดข้อมูลแบบทันท่วงทีที่จะเกิดขึ้นเมื่อแหล่งข้อมูลมีการเพิ่มหรือทำการเปลี่ยนแปลงข้อมูล โดยการสกัดข้อมูลแบบทันท่วงทีจะมีวิธีการทำงาน 3 แบบ ดังนี้ (แสดงดังรูปที่ 8-6)

1. การสกัดข้อมูลโดยใช้ล็อกไฟล์ของฐานข้อมูล
(Capture through Transaction Logs)

2. การสกัดข้อมูลโดยใช้ดาต้าเบสทริกเกอร์
(Capture through Database Triggers)

3. การสกัดข้อมูลโดยทำการสร้างแอพพลิเคชันไว้ที่แหล่งข้อมูล
(Capture in Source Applications)



รูปที่ 8-6 ทางเลือกในการสกัดข้อมูลแบบทันที

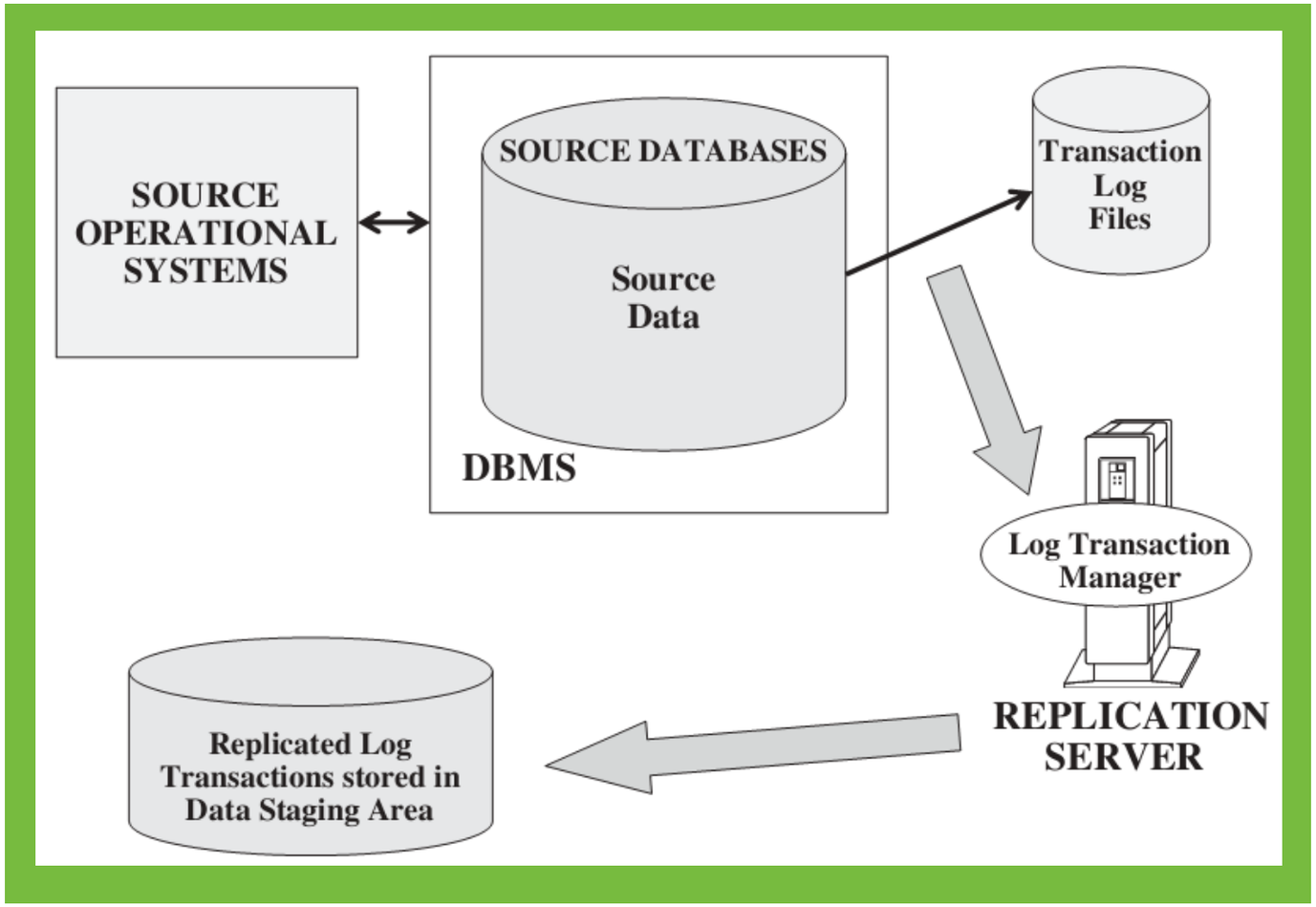
การสกัดข้อมูลแบบทันที (Immediate Data Extraction)

1. การสกัดข้อมูลโดยใช้ล็อกไฟล์ของฐานข้อมูล
(Capture through Transaction Logs)

วิธีการนี้จะเป็นการสกัดข้อมูลโดยใช้ล็อกไฟล์ที่ถูกจัดเก็บไว้ในระบบจัดการฐานข้อมูล (DBMS) ซึ่งโดยปกติของระบบจัดการฐานข้อมูลจะมีการเขียนข้อมูลลงล็อกไฟล์อยู่แล้วเมื่อมีการเพิ่ม ลบหรืออัพเดทข้อมูลเรคคอร์ดหนึ่ง ๆ ในฐานข้อมูล ซึ่งการจัดเก็บข้อมูลลงในล็อกไฟล์ของฐานข้อมูลจะทำเพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นกับการทำงานของระบบจัดการฐานข้อมูล ดังนั้น เมื่อระบบจัดการฐานข้อมูลมีการจัดเก็บข้อมูลลงในล็อกไฟล์จะทำให้เราสามารถทำการสกัดข้อมูลโดยทำการอ่านล็อกไฟล์เพื่อที่จะทราบถึงแต่ละรายการที่มีการกระทำกับฐานข้อมูล โดยการอ่านล็อกไฟล์จากระบบจัดการฐานข้อมูลนั้นจะไม่ทำให้การทำงานของระบบการดำเนินงานเพิ่มขึ้นเลย เนื่องจากล็อกไฟล์นั้นเป็นส่วนหนึ่งที่ต้องมีการจัดเก็บไว้ในฐานข้อมูลอยู่แล้ว แต่ถ้าเราใช้การจัดเก็บข้อมูลแบบอื่น ๆ เช่น การทำดัชนี (index) หรือการใช้แฟ้มข้อมูลจะเป็นการเพิ่มการทำงานให้กับระบบการดำเนินงานเนื่องจากต้องการสร้างดัชนีและทำการเขียนข้อมูลลงแฟ้มข้อมูลเอง

ในการสกัดข้อมูลโดยทำการอ่านล็อกไฟล์สามารถดำเนินการได้กับหลายแหล่งข้อมูลที่อาจเป็นระบบเป็นแบบกระจาย (Distributed system) ซึ่งโดยธรรมชาติของระบบแบบกระจายจะมีการทำสำเนาข้อมูล (Data replication) เพื่อให้แต่ละส่วนของระบบสามารถใช้ข้อมูลได้ทั้งหมด

ดังนั้นในการสกัดข้อมูล เราอาจใช้เทคโนโลยีการทำสำเนาข้อมูลเพื่อช่วยในการหาความเปลี่ยนแปลงของข้อมูลที่ถูกจัดเก็บอยู่ในแหล่งข้อมูลแบบกระจาย ดังแสดงในรูปที่ 8-7



รูปที่ 8-7 การสกัดข้อมูลโดยใช้การทำเสานำข้อมูลร่วมกับการใช้ล็อกไฟล์

จากรูปที่ 8-7 เป็นการทำงานของการสกัดข้อมูลจากระบบกระจายโดยใช้การทำสำเนาข้อมูลต่าง ๆ ระบบและฐานข้อมูลอาจมีด้วยกันหลายแห่ง ซึ่งอาจทำให้มีล็อกไฟล์หลายไฟล์ตามไปด้วย ดังนั้นในการสกัดข้อมูลโดยใช้การทำสำเนาข้อมูลเป็นส่วนหนึ่งของขั้นตอนการทำงานจะมีขั้นตอนการทำงานดังต่อไปนี้

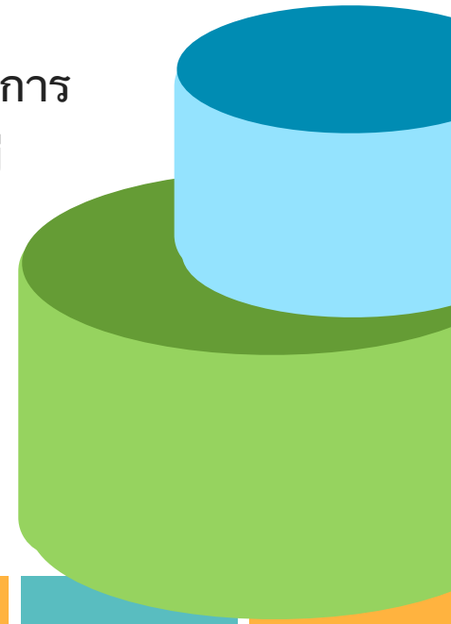
1. ระบุตารางจากแหล่งข้อมูลที่ใช้ที่เก็บข้อมูลที่เราต้องการสกัด (Identify the source system database table)
2. ระบุและกำหนดไฟล์เป้าหมายของ staging area (Identify and define target files in the staging area)
3. สร้างการเชื่อมโยงระหว่างตารางข้อมูลและไฟล์เป้าหมาย (Create mapping between the source table and target files)
4. ระบุโหมดการทำซ้ำ (Define the replication mode)
5. กำหนดตารางเวลาสำหรับกระบวนการทำซ้ำ (Schedule the replication process)
6. ทำการหาความแตกต่างของข้อมูลที่ต้องการจากล็อกไฟล์ (Capture the change from the transaction logs)
7. ถ่ายโอนข้อมูลที่สามารถสกัดได้จากล็อกไฟล์ไปยังแฟ้มเป้าหมายใน staging area (Transfer captured data from logs to target files)
8. ตรวจสอบความถูกต้องของการถ่ายโอนข้อมูล (Verify transfer of data changes)
9. ทำการเก็บผลของการทำสำเนาข้อมูลไว้เป็นเมตาดาต้า (In metadata, document the outcome of replication)
10. ดูแลรักษาคำจำกัดความของแหล่งข้อมูล ไฟล์เป้าหมาย และการเชื่อมโยง (Maintain definitions of sources, targets and mappings)

การสกัดข้อมูลแบบทันที (Immediate Data Extraction)

2. การสกัดข้อมูลโดยใช้ดาต้าเบสทริกเกอร์
(Capture through Database Triggers)

วิธีการนี้เป็นการสกัดข้อมูลจากระบบการดำเนินงานที่มีการใช้ฐานข้อมูลโดยใช้ทริกเกอร์ (Trigger) โดยที่ทริกเกอร์ คือ วิธีพิเศษในการจัดเก็บข้อมูล (หรือเราจะเรียกว่าโปรแกรมก็ได้) ซึ่งโดยปกติแล้ว โปรแกรมทริกเกอร์จะถูกเก็บอยู่ในฐานข้อมูลและจะถูกเรียกใช้งานก็ต่อเมื่อมีเหตุการณ์ที่เรากำหนดไว้ล่วงหน้าแล้วเกิดขึ้น ในการสร้างโปรแกรมทริกเกอร์นั้น เราสามารถสร้างโปรแกรมทริกเกอร์สำหรับทุก ๆ เหตุการณ์ที่เกี่ยวข้องกับข้อมูลที่เราต้องการสกัดได้ โดยผลลัพธ์ที่ได้จากโปรแกรมทริกเกอร์จะถูกเขียนอยู่ในไฟล์ที่ถูกแยกไว้เพื่อนำไปใช้ในการสกัดข้อมูลเข้าสู่คลังข้อมูล

ตัวอย่างเช่น ถ้าเราต้องการสกัดข้อมูลที่เป็นความเปลี่ยนแปลงของข้อมูลในตารางลูกค้า เราสามารถเขียนโปรแกรมทริกเกอร์เพื่อตรวจจับการเปลี่ยนแปลงทุกๆครั้งที่เกิดขึ้นกับข้อมูลลูกค้า อาทิเช่น การเพิ่ม ลบ และอัพเดทข้อมูลต่าง ๆ เป็นต้น โดยข้อมูลที่สกัดได้จากการใช้โปรแกรมทริกเกอร์จะมีความน่าเชื่อถือ ซึ่งเราสามารถทราบถึงข้อมูลก่อนและหลังการเปลี่ยนแปลง แต่อย่างไรก็ดี การสร้างและการดูแลโปรแกรมทริกเกอร์จะทำให้แหล่งข้อมูลหรือระบบดำเนินการมีการทำงานเพิ่มขึ้นจากเดิม ดังนั้น ในพิจารณาการใช้การสกัดข้อมูลโดยใช้โปรแกรมทริกเกอร์เราจะต้องพิจาณาถึงการทำงานที่เพิ่มขึ้นมาด้วย



การสกัดข้อมูลแบบทันที (Immediate Data Extraction)

3. การสกัดข้อมูลโดยทำการสร้างแอพพลิเคชันไว้ที่แหล่งข้อมูล
(Capture in Source Applications)

วิธีนี้จะทำการสร้างโปรแกรมที่เกี่ยวข้องกับการสกัดข้อมูลที่เป็นโปรแกรมสำหรับเขียนข้อมูลลงในไฟล์และฐานข้อมูล โดยจะทำการเขียนข้อมูลลงในไฟล์และฐานข้อมูลทุก ๆ ครั้งที่มีการเพิ่ม ลบ และอัพเดทข้อมูล ตามลำดับ วิธีการนี้จะต่างกับสองวิธีข้างต้นเล็กน้อย ตรงที่วิธีนี้สามารถประยุกต์ใช้กับระบบที่มีการเก็บข้อมูลหลากหลายไม่ว่าจะเป็น ฐานข้อมูล การทำดัชนี หรือแฟ้มข้อมูล เป็นต้น แต่อย่างไรก็ตามวิธีการนี้อาจจะให้ประสิทธิภาพของระบบดำเนินการหรือระบบของแหล่งข้อมูลลดลงเนื่องจากต้องมีขั้นตอนการเขียนข้อมูลลงไฟล์เพิ่มเข้ามาเพื่อช่วยในการหาความเปลี่ยนแปลงที่เกิดขึ้นกับข้อมูล

การสกัดข้อมูลแบบรอเวลา

(Deferred Data Extraction)

วิธีการนี้จะไม่ได้เป็นการสกัดข้อมูลแบบเรียลไทม์ (Real time) เหมือนกับการสกัดข้อมูลแบบทันที แต่จะเป็นการสกัดข้อมูลในภายหลังจากที่มีการเพิ่ม ลบ หรืออัพเดทข้อมูลในฐานข้อมูลของระบบการดำเนินงาน แต่จะกระทำก็ต่อเมื่อถึงเวลาที่เรากำหนด โดยการสกัดข้อมูลแบบรอเวลาจะมีวิธีการทำงาน 2 วิธี ดังแสดงในรูปที่ 8-8 ซึ่งสามารถอธิบายรายละเอียดได้ดังนี้

1. การสกัดข้อมูลโดยใช้ข้อมูลวันและเวลา
(Capture Based on Date and Time Stamp)

2. การสกัดข้อมูลโดยการเปรียบเทียบไฟล์
(Capture by Comparing Files)

การสกัดข้อมูลแบบรอเวลา (Deferred Data Extraction)

1. การสกัดข้อมูลโดยใช้ข้อมูลวันและเวลา
(Capture Based on Date and Time Stamp)

วิธีการนี้จะตั้งสมมติฐานที่ว่าข้อมูลแต่ละเรคคอร์ดในฐานข้อมูลจะมีข้อมูลเวลา (time stamp) แนบอยู่ด้วย ซึ่งข้อมูลเวลาจะเป็นเวลาในการเพิ่มหรืออัพเดทข้อมูลลงในฐานข้อมูล (ลบไม่มีเพราะเวลาจะถูกเก็บที่เรคคอร์ดในฐานข้อมูล เมื่อเรคคอร์ดถูกลบไปเราจะไม่สามารถเรียกดูข้อมูลเวลาได้) ในการเก็บข้อมูลเวลาจะช่วยให้เราสามารถเลือกเรคคอร์ดที่ต้องทำการสกัดข้อมูลได้ ซึ่งการสกัดข้อมูลโดยใช้ข้อมูลเวลานั้น เราจะต้องกำหนดช่วงเวลาที่แน่นอนในการสกัดข้อมูล เช่น การสกัดข้อมูลแต่ละครั้งจะเริ่มทำงานตอนเที่ยงคืนของทุกวัน โดยในแต่ละครั้งของการสกัดข้อมูล ข้อมูลที่จะถูกสกัดจะต้องเกิดขึ้นตั้งแต่ 00.00 ของเมื่อวานจนถึง 23.59 ของเมื่อวานเช่นกัน โดยในการสกัดข้อมูลจะเรียกใช้ข้อมูลเวลาของแต่ละเรคคอร์ดที่อยู่ในช่วงเวลาที่กำหนดและเป็นข้อมูลเวลาล่าสุดเท่านั้น ดังนั้น ถ้าในหนึ่งวันข้อมูลเรคคอร์ดหนึ่ง ๆ มีการเปลี่ยนแปลงมากกว่า 1 ครั้งจะทำให้ข้อมูลที่มีการเปลี่ยนแปลงที่ไม่ใช่ข้อมูลล่าสุดหายไป

การสกัดข้อมูลโดยใช้ข้อมูลเวลาอาจมีปัญหาเกิดขึ้นเมื่อมีการลบข้อมูลเกิดขึ้นกับระบบการดำเนินงาน ซึ่งจะทำให้เราไม่สามารถมองเห็นข้อมูลเวลาของข้อมูลนั้น ๆ ได้ นี้จึงเป็นเหตุให้การสกัดข้อมูลไม่สามารถทราบถึงการลบข้อมูลที่เกิดขึ้นเหล่านั้นได้

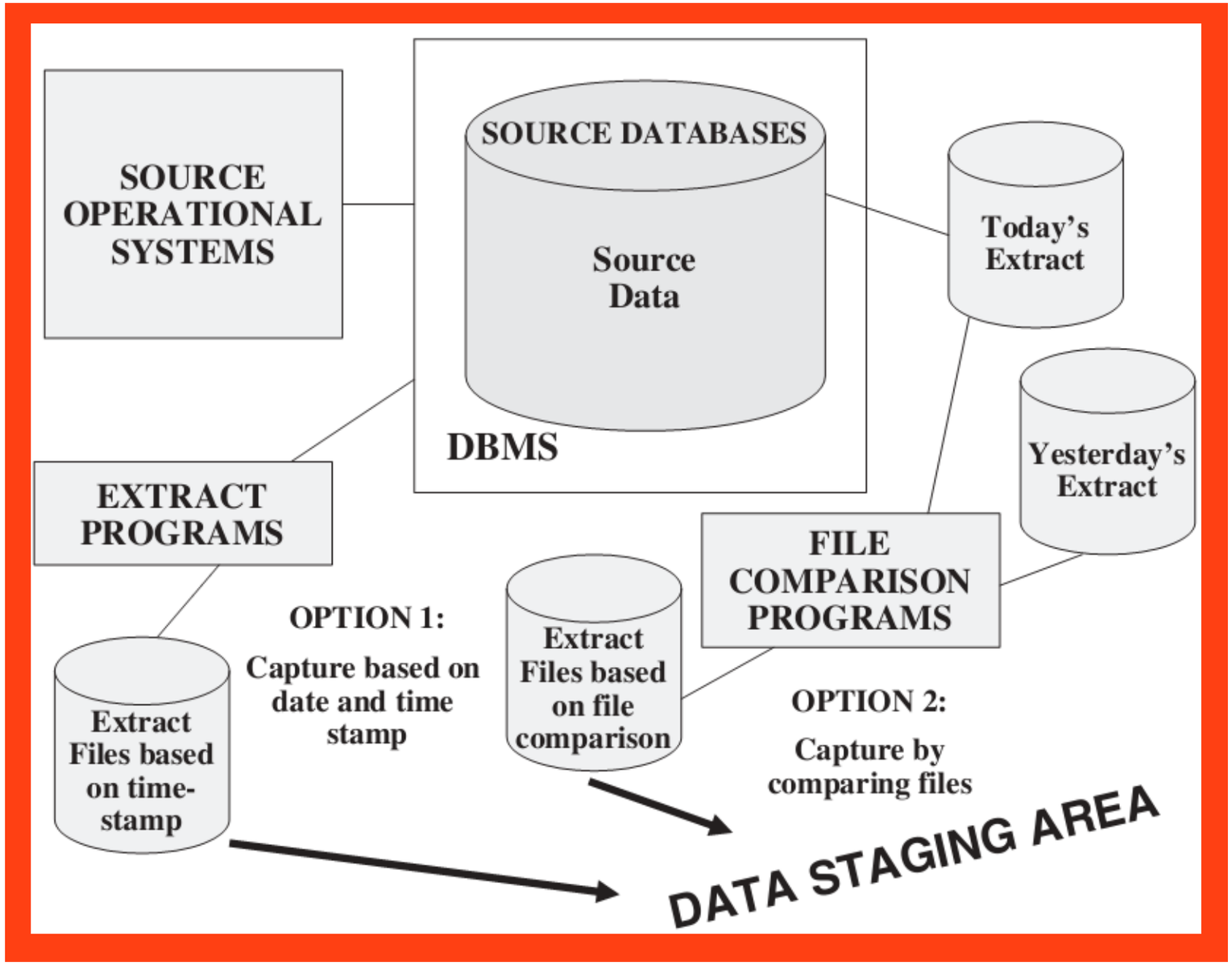
จากปัญหาการลบข้อมูลเราสามารถดำเนินการแก้ไขปัญหาได้โดยทำการย้ายข้อมูลที่จะถูกลบไปยังตารางสำหรับข้อมูลที่จะถูกลบแทนที่จะทำการลบข้อมูลจริงๆ จากนั้นเมื่อถึงเวลาสำหรับการสกัดข้อมูล ข้อมูลในตารางสำหรับที่จะถูกลบจะถูกอ่านขึ้นมาเพื่อประมวลผลแล้วจึงค่อยทำการลบข้อมูลเหล่านั้นออกจากระบบการดำเนินงานจริงๆ ซึ่งวิธีดังกล่าวจะช่วยให้ข้อมูลในคลังข้อมูลมีความถูกต้องและตรงกับข้อมูลในแหล่งข้อมูล แต่ระบบการดำเนินงานจะต้องมีการทำงานที่เพิ่มขึ้น

การสกัดข้อมูลแบบรอเวลา (Deferred Data Extraction)

2. การสกัดข้อมูลโดยการเปรียบเทียบไฟล์
(Capture by Comparing Files)

วิธีการนี้สามารถเรียกได้อีกอย่างหนึ่งว่า “snapshot differential technique” ซึ่งจะเป็นการเปรียบเทียบข้อมูลที่ได้จากการทำสกัดข้อมูล 2 ครั้งล่าสุด ถ้าระบบคลังข้อมูลทำการสกัดข้อมูลวันละหนึ่งครั้ง ขั้นตอนการทำงานจะเริ่มจากการสกัดข้อมูลในวันปัจจุบัน จากนั้นทำสำเนาเก็บไว้ แล้วนำสำเนาที่ทำไว้ไปเปรียบเทียบกับสำเนาที่ทำไว้สำหรับการสกัดข้อมูลของเมื่อวาน เมื่อการเปรียบเทียบเสร็จสิ้น เราจะทราบถึงเรคคอร์ดที่มีการเพิ่ม ลบ หรืออัพเดทในระหว่าง 2 วัน ซึ่งผลลัพธ์จากการเปรียบเทียบจะช่วยให้เราสามารถหาความแตกต่างของข้อมูลระหว่าง 2 วันได้

โดยในการสกัดข้อมูลด้วยวิธีการนี้จะต้องทำการเก็บข้อมูลที่มีการเปลี่ยนแปลงจาก 2 ครั้งล่าสุดที่ทำการสกัดข้อมูล จึงทำให้ต้องเสียพื้นที่ในการจัดเก็บข้อมูลเพิ่มขึ้น รวมถึงต้องเสียเวลาในการเปรียบเทียบข้อมูลด้วย ถ้าข้อมูลที่ต้องทำการเปรียบเทียบมีจำนวนมาก ระบบอาจไม่สามารถใช้วิธีนี้ในการสกัดข้อมูลได้ วิธีการนี้จะไม่เหมาะกับระบบทั่ว ๆ ไป แต่จะเหมาะกับระบบที่ไม่มีล็อกไฟล์และข้อมูลเวลาเท่านั้น



รูปที่ 8-8 วิธีสำหรับสกัดข้อมูลแบบรอเวลา

ประสิทธิภาพของการสกัดข้อมูลวิธีต่างๆ (Evaluation of the Techniques)

จากที่กล่าวมาข้างต้น เราได้ทราบถึงวิธีในการสกัดข้อมูลหลายวิธีด้วยกัน เช่น

การสกัดข้อมูลที่เป็น “static data” (Capture of static data)

การสกัดข้อมูลโดยใช้ล็อกไฟล์ของ DBMS (Capture through transaction logs)

การสกัดข้อมูลโดยใช้โปรแกรมทริกเกอร์ (Capture through database triggers)

การสกัดข้อมูลโดยการเขียนโปรแกรมเพื่อจัดการกับข้อมูลในแหล่งข้อมูล (Capture in source application)

การสกัดข้อมูลโดยใช้ข้อมูลวันและเวลา (Capture based on date and time stamp)

การสกัดข้อมูลโดยการเปรียบเทียบไฟล์ (Capture by comparing files)

จากวิธีทั้งหมดข้างต้น แต่ละวิธีในการสกัดข้อมูลก็มีข้อดี-ข้อเสียที่ต่างกัน (ดังแสดงในรูปที่ 8-9) ดังนั้นเมื่อเราทำการออกแบบฟังก์ชันการสกัดข้อมูลจากระบบการดำเนินงาน เราจะต้องเลือกวิธีการสกัดข้อมูลให้เหมาะกับสภาพแวดล้องของระบบการดำเงินงานหรือแหล่งข้อมูลที่เราจะต้องจัดการ โดยจะต้องคำนึงถึงการใช้ทรัพยากรหรือการทำงานจากแหล่งข้อมูลให้น้อยที่สุด รวมถึงความยากง่ายและค่าใช้จ่ายในการสร้างฟังก์ชันการสกัดข้อมูลด้วย

Capture of static data

Good flexibility for capture specifications.
Performance of source systems not affected.
No revisions to existing applications.
Can be used on legacy systems.
Can be used on file-oriented systems.
Vendor products are used. No internal costs.

Capture in source applications

Good flexibility for capture specifications.
Performance of source systems affected a bit.
Major revisions to existing applications.
Can be used on most legacy systems.
Can be used on file-oriented systems.
High internal costs because of in-house work.

Capture through transaction logs

Not much flexibility for capture specifications.
Performance of source systems not affected.
No revisions to existing applications.
Can be used on most legacy systems.
Cannot be used on file-oriented systems.
Vendor products are used. No internal costs.

Capture based on date and time stamp

Good flexibility for capture specifications.
Performance of source systems not affected.
Major revisions to existing applications likely.
Cannot be used on most legacy systems.
Can be used on file-oriented systems.
Vendor products may be used.

Capture through database triggers

Not much flexibility for capture specifications.
Performance of source systems affected a bit.
No revisions to existing applications.
Cannot be used on most legacy systems.
Cannot be used on file-oriented systems.
Vendor products are used. No internal costs.

Capture by comparing files

Good flexibility for capture specifications.
Performance of source systems not affected.
No revisions to existing applications.
May be used on legacy systems.
May be used on file-oriented systems.
Vendor products are used. No internal costs.

รูปที่ 8-9 ข้อดีและข้อเสียของเทคนิคการสกัดข้อมูล

SECTION 6

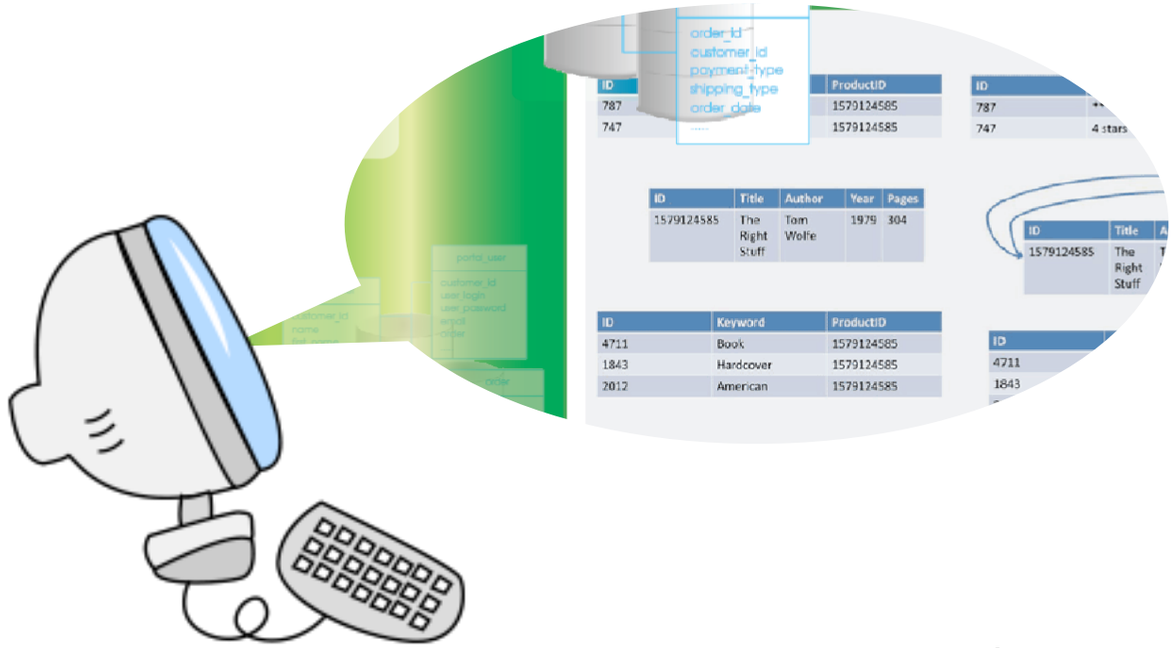
การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล

การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล

ในการสกัดข้อมูลจากแหล่งข้อมูล ข้อมูลที่สกัดได้จะเป็นข้อมูลดิบ ซึ่งข้อมูลเหล่านี้อาจจะยังไม่สามารถนำไปประยุกต์ใช้กับคลังข้อมูลได้โดยตรง เนื่องจากคุณภาพของข้อมูลอาจยังไม่ดีพอต่อการตัดสินใจเชิงกลยุทธ์

ดังนั้นเราจึงต้องทำการปรับปรุงคุณภาพของข้อมูลให้ดีขึ้นเสียก่อนที่จะนำไปใช้ในคลังข้อมูลซึ่งก็คือ “การเปลี่ยนแปลงหรือเปลี่ยนรูปข้อมูล *(Data transformation)*” ที่ได้รับมาจากขั้นตอนการสกัดข้อมูล ในการเปลี่ยนแปลงข้อมูลนั้น ข้อมูลจะถูกทำการเปลี่ยนแปลงให้เป็นมาตรฐานมากขึ้น

เนื่องจากข้อมูลที่สกัดได้อาจมาจากหลายแหล่งข้อมูล และแต่ละแหล่งข้อมูลอาจมีความโครงสร้างข้อมูลที่แตกต่างกัน การเปลี่ยนแปลงข้อมูลจะช่วยทำให้แน่ใจได้ว่า เมื่อทำการรวมข้อมูลเข้าด้วยกันแล้วข้อมูลที่ได้จะสามารถตอบสนองความต้องการทางธุรกิจได้



วัตถุประสงค์หลักของการเปลี่ยนแปลงข้อมูลจะเน้นที่การปรับปรุงคุณภาพของข้อมูลให้ดีขึ้น เมื่อทำการเปลี่ยนแปลงข้อมูลอาจทำให้ฟิลด์ (field) ต่าง ๆ ของข้อมูลเปลี่ยนแปลงไป หรืออาจทำให้โครงสร้างของข้อมูลมีการเปลี่ยนแปลง รวมทั้งยังส่งผลต่อประสิทธิภาพของการทำงานของคลังข้อมูลอีกด้วย

ดังนั้นก่อนที่จะทำการเปลี่ยนแปลงข้อมูลเราควรจะต้องเข้าใจถึงข้อมูลที่มีอยู่ในแหล่งข้อมูล และเทคนิคต่าง ๆ ที่จะใช้ในการเปลี่ยนแปลงข้อมูลเสียก่อน

ขั้นตอนการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล

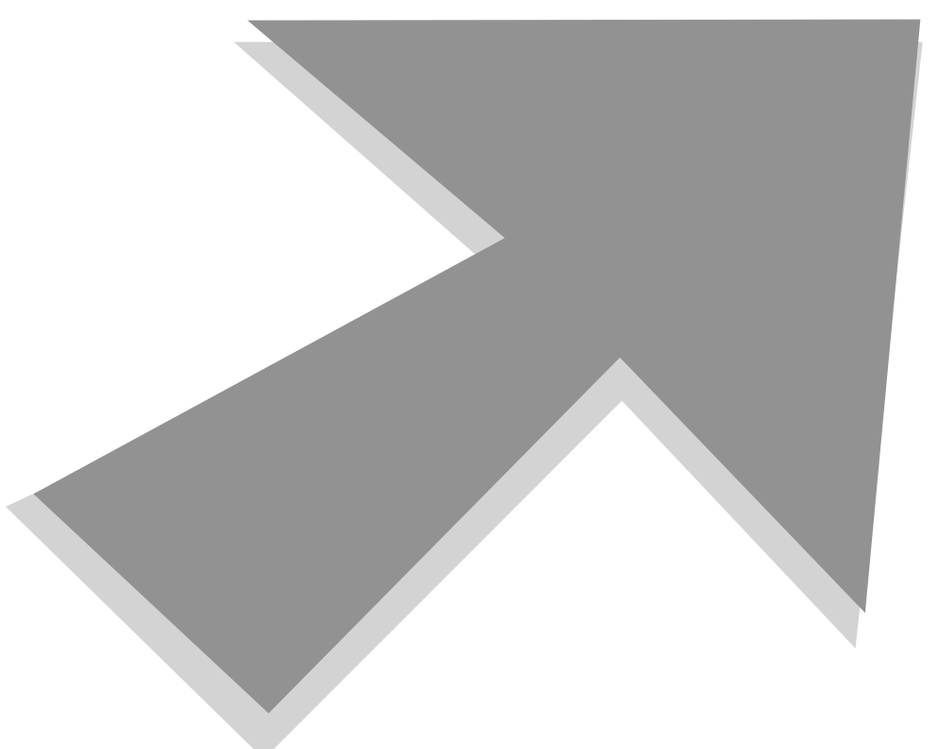
การเปลี่ยนแปลงข้อมูลสามารถแบ่งเป็นขั้นตอนย่อยได้หลายขั้นตอน ดังนี้

1 Selection

คือ การเลือกเรคคอร์ดทั้งหมดหรือกลุ่มของเรคคอร์ดเพียงบางกลุ่มจากแหล่งข้อมูล ซึ่งการทำ Selection จะเป็นขั้นตอนหนึ่งในการสกัดข้อมูล แต่ในบางกรณี การจัดวางองค์ประกอบของโครงสร้างข้อมูลจากแหล่งข้อมูลอาจไม่ตอบสนองต่อการเลือกข้อมูลที่สำคัญในขั้นตอนการสกัดข้อมูล ในกรณีเหล่านี้ เราควรจะทำการสกัดข้อมูลอีกครั้งหนึ่งโดยทำการสกัดเรคคอร์ดทั้งหมดก่อน จากนั้นค่อยทำการเลือกข้อมูลที่จะทำการเปลี่ยนแปลง

2 Splitting/Joining

คือ การแบ่งส่วนของข้อมูลที่ถูกเลือกไว้แล้วเพื่อที่จะทำการเปลี่ยนแปลงข้อมูล และการรวมข้อมูลที่ถูกเลือกไว้แล้วบางส่วนเข้าด้วยกัน โดยข้อมูลที่นำมารวมกันอาจมาจากหลายแหล่งข้อมูล



3 Conversion

คือ การทำให้ข้อมูลที่ถูกสกัดออกมาจากแหล่งข้อมูลเป็นมาตราฐานเดียวกัน และ การทำให้ฟิลด์ต่าง ๆ สามารถใช้งานได้ รวมถึงการทำให้ผู้ใช้เข้าใจในฟิลด์นั้นๆ

4 Summarization

คือ การสรุปข้อมูล กล่าวคือ ในบางสถานการณ์เราอาจพบว่าเป็นไปไม่ได้เลยที่จะเก็บข้อมูลที่มีรายละเอียดสูงมาก ๆ ในคลังข้อมูลที่สร้างขึ้น ซึ่งเหตุการณ์เหล่านี้อาจเกิดจากผู้ใช้ไม่ได้ต้องการข้อมูลที่มีรายละเอียดสูงสำหรับการวิเคราะห์ ตัวอย่างเช่น ซุปเปอร์มาร์เก็ตที่ซึ่งข้อมูลการขายที่มีความละเอียดสูงที่สุดจะเกิดขึ้นที่ขั้นตอนจ่ายเงิน ข้อมูลที่ละเอียดที่สุดที่สามารถเก็บได้ดังเช่น ยาสระผม ยี่ห้อแฟซ่า สูตรลดรังแค เป็นต้น ซึ่งข้อมูลที่ละเอียดมาก ๆ เหล่านี้อาจไม่ใช่สิ่งที่ต้องการก็เป็นได้ ผู้จัดการอาจต้องการแค่ยอดขายของสินค้านั้น ๆ ยอดขายในแต่ละสาขาของซุปเปอร์มาเก็ต (ในกรณีที่มีหลายสาขา) ยอดขายในแต่ละวัน ดังนั้น เมื่อมีกรณีแบบนี้เกิดขึ้น ขั้นตอนการเปลี่ยนแปลงข้อมูลจะต้องทำการสร้างผลสรุปของข้อมูลเพื่อให้สอดคล้องกับสิ่งที่ผู้จัดการต้องการ

5 Enrichment

คือ การจัดแจงฟิลด์ต่าง ๆ ที่มีอยู่ใหม่ รวมถึงการทำให้ฟิลด์นั้น ๆ เข้าใจได้ง่าย เพื่อทำให้ฟิลด์เหล่านั้นมีประโยชน์มากขึ้น เราอาจใช้ 1 ฟิลด์ (หรือมากกว่านั้น) จากเรคคอร์ดที่เป็นอินพุตเดียวกัน เพื่อสร้างมุมมองของข้อมูลที่ขึ้นสำหรับคลังข้อมูล

6 Format Revisions

คือ การเปลี่ยนชนิดของข้อมูลและความยาวของข้อมูลในบางฟิลด์ ตัวอย่างเช่น ชนิดของแพคเกจสินค้าอาจแสดงได้ด้วยโค้ดและชื่อ ซึ่งจะถูกเก็บอยู่ในฟิลด์ที่เป็นตัวเลขและข้อความ ในอีกทำนองหนึ่ง ความยาวของชนิดของแพคเกจสินค้าอาจแตกต่างกันระหว่างแหล่งข้อมูลที่แตกต่างกัน เราควรจะทำการสร้างมาตรฐานและทำการเปลี่ยนแปลงชนิดของข้อมูลไปเป็นข้อความเพื่อให้ข้อมูลดังกล่าวมีความหมายเพื่อที่ผู้ใช้จะสามารถเข้าใจได้ง่าย

7 Decoding of Fields

คือ การถอดรหัสลับที่อาจเกิดขึ้นในแหล่งข้อมูล เมื่อเราทำการสร้างคลังข้อมูลจากแหล่งข้อมูลหลายแหล่ง เราอาจจะพบเจอข้อมูลที่เป็นรหัสซึ่งอาจจะทำให้ผู้ใช้ไม่เข้าใจในรหัสเหล่านั้นได้ ตัวอย่างเช่น ข้อมูลเกี่ยวกับเพศ บางแหล่งข้อมูลจะเก็บข้อมูลเกี่ยวกับเพศ เป็น 1 และ 2 สำหรับเพศชายและหญิงตามลำดับ ในขณะที่บางแหล่งข้อมูลจะใช้การกำหนดค่าเกี่ยวกับเพศเป็น M และ F (บางครั้งอาจใช้ W) จากการสังเกตุระบบในปัจจุบันจะพบว่า มีหลายระบบที่ใช้รหัสลับแทนการอ้างถึงข้อมูลต่าง ๆ ทางธุรกิจ อาทิเช่น AC, IN, RE และ SU เมื่อเราพบเจอโค้ดในลักษณะแบบนี้เราจำเป็นต้องถอดรหัสลับและเปลี่ยนข้อมูลเหล่านั้นให้เป็นข้อมูลที่ผู้ใช้สามารถเข้าใจได้ จากตัวอย่างข้างต้น Active = AC, Inactive = IN, Regular = Re และ Suspended = SU เป็นต้น

8 Calculated and Derived Values

คือ การคำนวณหรือหาผลสรุปของข้อมูลและการนำข้อมูลเหล่านั้นไปใช้ ตัวอย่างเช่น ข้อมูลที่ถูกสกัดจากระบบการขายประกอบด้วยจำนวนยอดขาย และประมาณการค่าใช้จ่ายในการดำเนินงานจำแนกตามผลิตภัณฑ์ เราจะต้องทำการคำนวณต้นทุนทั้งหมดและอัตรากำไรก่อนที่จะเก็บข้อมูลลงในคลังข้อมูล เป็นต้น

9 Splitting of Single Fields

คือ การแยกฟิลด์หนึ่ง ๆ ออกเป็นหลายฟิลด์ ตัวอย่างเช่น ในระบบเดิมที่ทำการเก็บข้อมูลชื่อและที่อยู่ของลูกค้าและพนักงานในแบบที่เป็นข้อความ โดยที่ ชื่อแรก ชื่อกลาง และนามสกุล จะถูกเก็บอยู่ในฟิลด์เพียงฟิลด์เดียว หรือในบางระบบจะเก็บ เขต จังหวัด และ รหัสไปรษณีย์ไว้ในฟิลด์เดียว เป็นต้น จากระบบเดิมที่กล่าวมาข้างต้น เราจำเป็นต้องเก็บองค์ประกอบของ ชื่อ และที่อยู่ ในคลังข้อมูลให้แยกออกจากกัน เพื่อ (1) ช่วยในเรื่องประสิทธิภาพในการดำเนินงานโดยใช้การสร้างดัชนี (indexing) กับองค์ประกอบนั้นๆ และ (2) ผู้ใช้คลังข้อมูลอาจต้องการวิเคราะห์ข้อมูลโดยใช้ข้อมูลที่ถูกแยกจากกระบวนการข้างต้น

10 Merging Information

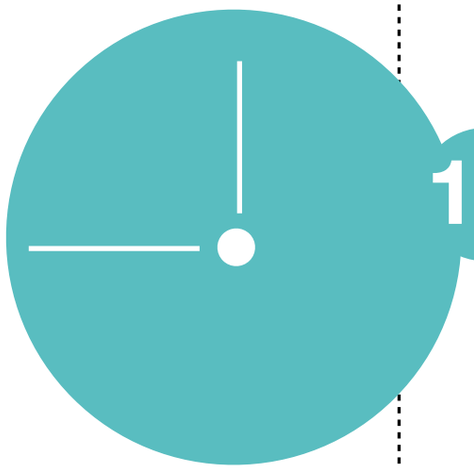
คือ การผสานข้อมูลที่มีรายละเอียดแต่ละส่วนกระจายอยู่ที่หลายแหล่งข้อมูลเข้าด้วยกัน ตัวอย่างเช่น ข้อมูลเกี่ยวกับสินค้าที่อาจมาจากหลายแหล่งข้อมูล โค้ดและคำอธิบายสินค้าอาจมาจากแหล่งข้อมูลเพียงแหล่งเดียว ข้อมูลเกี่ยวกับประเภทของแพคเกจอาจจะพบในอีกแหล่งข้อมูลหนึ่ง ข้อมูลต้นทุนของสินค้าอาจจะได้มาจากอีกแหล่งข้อมูลหนึ่ง เราต้องทำการผสานข้อมูลของโค้ดของสินค้า คำอธิบายสินค้า ชนิดของแพคเกจของสินค้า และต้นทุนของสินค้าเข้าด้วยกันและเก็บไว้ในเอนทิตี (Entity) เดียวกัน

11 Character set conversion

คือ การแปลงเซตของตัวอักษรให้เป็นมาตราฐาน เพื่อใช้ในการจัดเก็บข้อมูลในคลังข้อมูล เช่น ถ้าเรามีเมนเฟรม (Mainframe) ที่เก็บระบบเดิม (แหล่งข้อมูล) โดยข้อมูลที่ได้จากระบบนี้จะอยู่ภายใต้อักขระแบบ EBCDIC จากระบบข้างต้น ถ้าเราต้องการที่จะสร้างคลังข้อมูลโดยใช้คอมพิวเตอร์ส่วนบุคคล (Personal Computer, PC) เราต้องทำการแปลงเซตของอักขระจาก EBCDIC ให้อยู่ในรูปของ ASCII ถึงจะเก็บข้อมูลเข้าสู่คลังข้อมูลได้

12 Conversion of Units of Measurements

คือ การแปลงหน่วยของมาตรวัด ในปัจจุบันหลายๆบริษัทได้กระจายการทำธุรกิจอยู่ในหลาย ๆ ประเทศ ถ้าทำธุรกิจกับประเทศในทวีปยุโรปหน่วยที่ใช้วัดจะเป็นเมตร แต่ในบางประเทศอาจใช้หน่วยวัดเป็นเซนติเมตร ดังนั้นถ้าเราทำธุรกิจกับหลาย ๆ ประเทศเราต้องทำการแปลงมาตรวัดให้อยู่ในมาตราฐานเดียวกัน



13 Data/Time Conversion

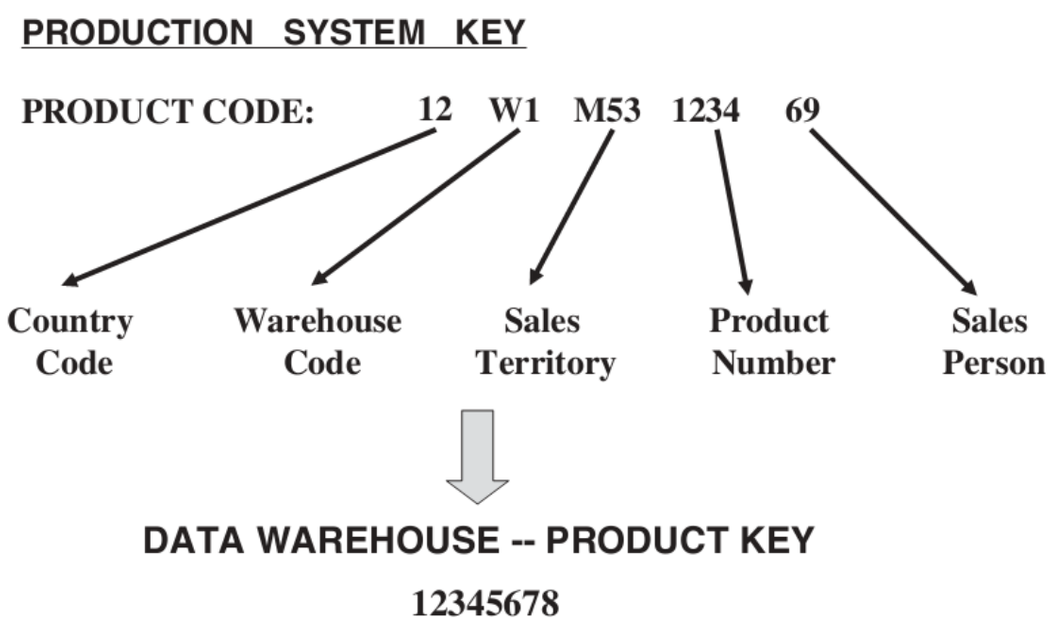
คือ การแปลงหน่วยของวันและเวลาให้อยู่ในมาตราฐานเดียวกัน รูปแบบของวันที่ที่ใช้ในประเทศสหรัฐอเมริกาและอังกฤษอาจเป็นรูปแบบมาตราฐานสากล แต่รูปแบบที่ใช้ในประเทศทั้งสองก็มีความแตกต่างกัน เช่น วันที่ October 11, 2012 รูปแบบที่ใช้ในประเทศสหรัฐอเมริกาจะสามารถเขียนได้เป็น 10/11/2012 แต่รูปแบบที่ใช้ในประเทศอังกฤษจะเขียนเป็น 11/10/2012 ดังนั้น เพื่อทำให้วันและเวลาในคลังข้อมูลเป็นมาตราฐานเดียวกันเราอาจเก็บรูปแบบวันได้เป็น 11 OCT 2012 เป็นต้น

14 Summarization

คือ การสรุปข้อมูลเพื่อโอนถ่ายเข้าไปยังคลังข้อมูล แทนที่จะโอนถ่ายข้อมูลที่มีความละเอียดค่อนข้างมากเข้าไปโดยตรง ตัวอย่างเช่น ในการวิเคราะห์รูปแบบการขายของบริษัทบัตรเครดิต เราอาจจะไม่จำเป็นต้องเก็บข้อมูลแต่ละการทำธุรกรรมของบัตรใบหนึ่ง ๆ เราอาจจะต้องทำการหาผลสรุปของการธุรกรรมในหนึ่งวันต่อบัตรหนึ่งใบแล้วทำการเก็บข้อมูลเหล่านั้นไว้ในคลังข้อมูล

15 Key Restructuring

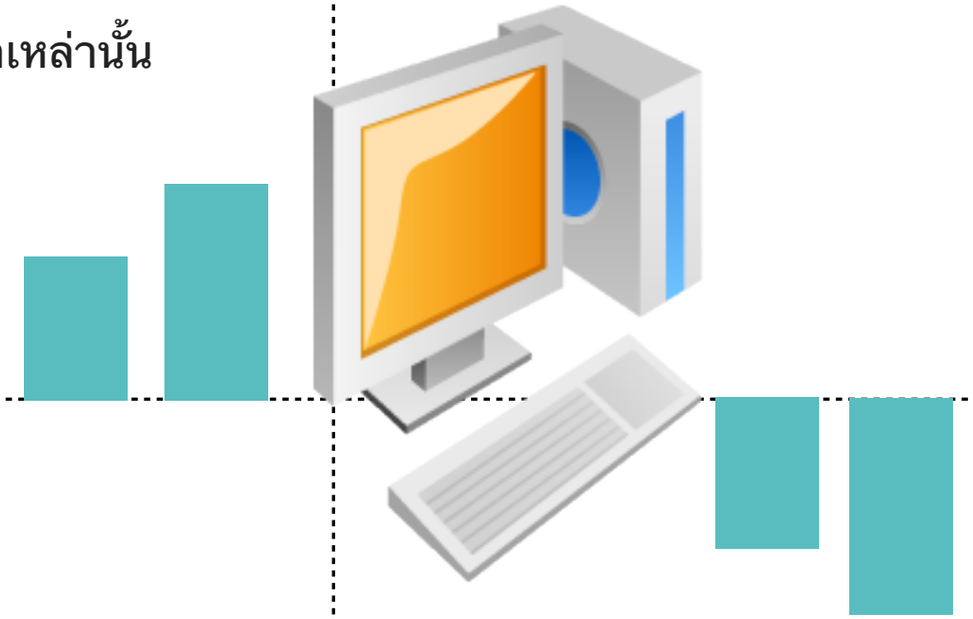
คือ การสร้างคีย์ใหม่แทนที่ของเดิม ตัวอย่างเช่น จากรูปที่ 8-10 ที่แสดงโค้ดของสินค้าซึ่งถูกกำหนดโดยสามารถบอกถึงแหล่งที่มาได้และมีความซับซ้อน ซึ่งถ้าเราใช้โค้ดของสินค้าลักษณะนี้เป็นคีย์หลัก (Primary key) อาจทำให้เกิดปัญหาในกรณีที่โค้ดของสินค้านี้ถูกเคลื่อนย้ายไปยังคลังข้อมูลอื่น ก็จะทำให้ส่วนของโค้ดของสินค้าในส่วนของโค้ดของคลังข้อมูลต้องเปลี่ยนไปซึ่งการเปลี่ยนแปลงนี้จะทำให้เกิดปัญหากับระบบดั้งเดิมที่จะต้องแก้ตามด้วย



รูปที่ 8-10
การสร้างคีย์ของข้อมูลขึ้นใหม่
(Key restructuring)

16 Deduplication

คือ การขจัดความซ้ำซ้อนของข้อมูล ตัวอย่างเช่น ในหลาย ๆ บริษัท การเก็บข้อมูลของลูกค้าอาจจะทำการเก็บข้อมูลอยู่ในหลายเรคคอร์ด แต่สำหรับคลังข้อมูลแล้ว เราอาจต้องการที่จะเก็บข้อมูลเพียงเรคคอร์ดเดียว สำหรับลูกค้าหนึ่งคน เราจึงต้องทำการเชื่อมข้อมูลหลายเรคคอร์ดเหล่านั้นเข้าด้วยกันแล้วเก็บไว้ให้อยู่ในเรคคอร์ดเดียว



หลักการสร้างฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล

การทำการเปลี่ยนแปลงข้อมูลจะมีอยู่ 2 วิธีหลัก ในการดำเนินการ คือ

(1) การใช้เครื่องมือ

(2) การสร้างฟังก์ชันการเปลี่ยนแปลงข้อมูลขึ้นเอง

ในการจะเลือกว่าจะใช้วิธีการใด ในการสร้างฟังก์ชันการเปลี่ยนแปลงข้อมูลนั้น ผู้สร้างจะต้องพิจารณาปัจจัยต่าง ๆ ที่มีผลกระทบต่อการดำเนินการ ถ้าเราต้องการใช้เครื่องมือในการสร้าง เราอาจจะต้องใช้เวลาในการดำเนินงานต่าง ๆ เช่น ศึกษาเกี่ยวกับเครื่องมือนั้น ๆ ปรับแต่งเครื่องมือเพื่อให้เหมาะสมกับคลังข้อมูลที่สร้างขึ้น ติดตั้งเครื่องมือที่จำเป็นต้องใช้ ฝึกอบรมทีมงานเกี่ยวกับเครื่องมือ และรวมเครื่องมือที่จะใช้เข้ากับคลังข้อมูล เครื่องมือสำหรับการเปลี่ยนแปลงข้อมูลอาจมีราคาแพง ถ้าของเขตของคลังข้อมูลที่เราจะสร้างขึ้นมีขนาดไม่ใหญ่มาก เราอาจจะไม่มีงบประมาณสำหรับเครื่องมือที่จะใช้ก็เป็นได้

การใช้เครื่องมือสำเร็จรูปในการสร้างฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล

ในหลาย ๆ ปีที่ผ่านมา มีเครื่องมือสำหรับการเปลี่ยนแปลงข้อมูลถูกพัฒนาอย่างแพร่หลายเพื่อรองรับหลาย ๆ การทำงานและมีความยืดหยุ่นมากขึ้น การใช้เครื่องมือในการเปลี่ยนแปลงข้อมูลจะสามารถช่วยเพิ่มประสิทธิภาพในการทำงานรวมถึงการเพิ่มความถูกต้องของข้อมูลด้วย ข้อดีอย่างหนึ่งของการใช้เครื่องมือในการแปลงข้อมูล คือ เครื่องมือจะทำการเก็บเมตาดาต้า (Metadata) ให้เองอัตโนมัติ

Metadata

เมื่อเราทำการกำหนดค่าพารามิเตอร์ และกฎต่าง ๆ เครื่องมือจะทำการเก็บข้อมูลเหล่านั้นไว้เป็นเมตาดาต้าด้วย เมื่อเราทำการเปลี่ยนฟังก์ชันการเปลี่ยนแปลงข้อมูล ซึ่งอาจเกิดจากการเปลี่ยนแปลงของการดำเนินการทางธุรกิจ หรือ นิยามของข้อมูล เราเพียงเลือกฟังก์ชันที่ต้องการเปลี่ยนจากฟังก์ชันเดิม เครื่องมือจะทำการปรับแก้เมตาดาต้าให้เองโดยอัตโนมัติ

การสร้างฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลขึ้นเอง

การสร้างฟังก์ชันการเปลี่ยนแปลงข้อมูลเอง ได้รับความนิยมเป็นอย่างมากในช่วงแรกของการพัฒนาคลังข้อมูล แต่เมื่อมีเครื่องมือวางขายอยู่ในตลาดไอที ความนิยมก็ลดลง แต่อย่างไรก็ดีการสร้างฟังก์ชันการเปลี่ยนแปลงข้อมูลเองก็ยังได้รับความนิยมสำหรับคลังข้อมูลที่มีขนาดเล็ก ๆ ในการสร้างฟังก์ชันต่าง ๆ เองจะต้องมีนักวิเคราะห์และโปรแกรมเมอร์ที่มีความรู้อยู่แล้วและเชี่ยวชาญที่จะสามารถผลิตโปรแกรมและสคริปต์ได้ ข้อเสียของการสร้างฟังก์ชันการเปลี่ยนแปลงข้อมูลขึ้นเองจะเกี่ยวกับเมตาดาต้า นั่นคือผู้พัฒนาฟังก์ชันการเปลี่ยนแปลงข้อมูลจะต้องออกแบบว่าควรเก็บข้อมูลใดบ้างไว้ในเมตาดาต้า และต้องคิดเกี่ยวกับการดูแลรักษาเมตาดาต้าด้วย

SECTION 7

การถ่ายโอนข้อมูลไปยังคลังข้อมูล

การถ่ายโอนข้อมูลไปยังคลังข้อมูล

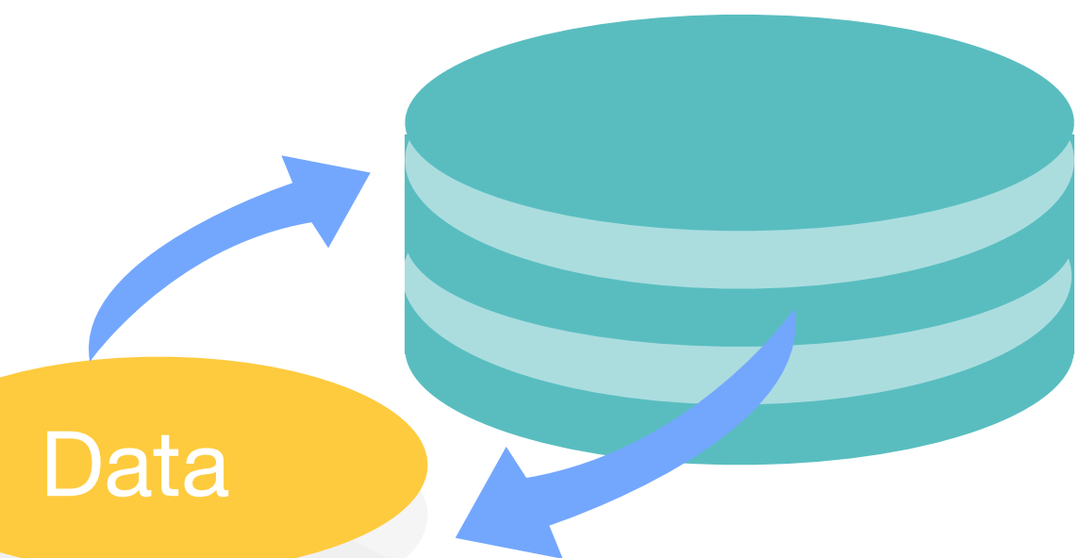
การถ่ายโอนข้อมูลจะเป็นการรับเอาข้อมูลที่สกัดได้จากแหล่งข้อมูลและทำการเปลี่ยนแปลงข้อมูลแล้วไปเก็บไว้ในคลังข้อมูล การถ่ายโอนข้อมูลนั้นสามารถทำได้หลายวิธีซึ่งมักจะเรียกการถ่ายโอนข้อมูลว่า การประยุต์ใช้ข้อมูล (applying the data) การโหลดข้อมูล (loading the data) และการทำให้ข้อมูลมีความสดใหม่ (refreshing the data) ซึ่งสามารถอธิบายคร่าว ๆ ได้ดังนี้

Initial load

คือ การถ่ายโอนข้อมูลจากแหล่งข้อมูลไปยังคลังข้อมูลครั้งแรก

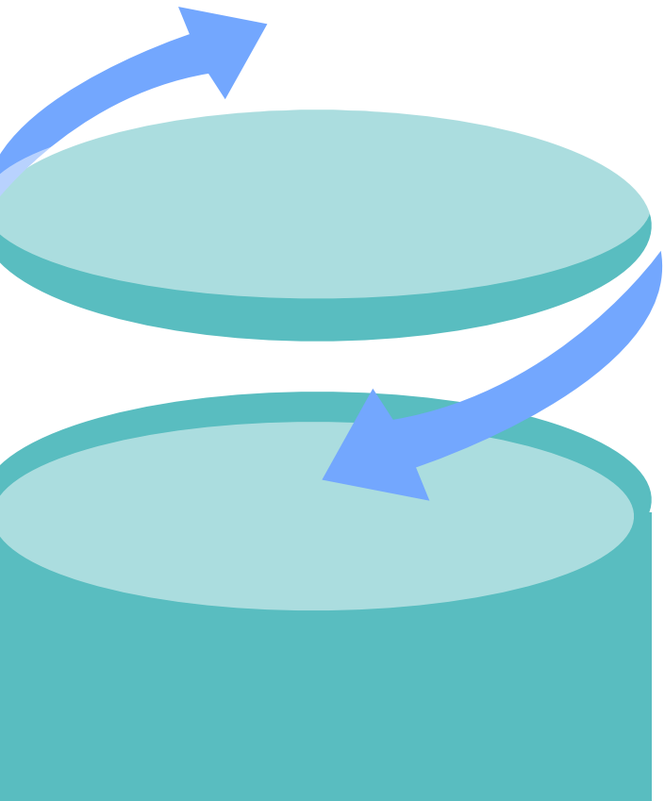
Incremental load

คือ การเพิ่มข้อมูลที่มีการเปลี่ยนแปลงเข้าสู่คลังข้อมูลอย่างต่อเนื่องตามความจำเป็นในลักษณะที่เป็นระยะๆ



Full refresh

คือ การลบข้อมูลทั้งหมดออกจากคลังข้อมูลแล้วทำการถ่ายโอนข้อมูลที่มีความสดใหม่ลงไปในคลังข้อมูล ซึ่งการถ่ายโอนข้อมูลใหม่จะเหมือนกับการทำ initial load อีกครั้งหนึ่ง



ในการโหลดข้อมูลเข้าสู่คลังข้อมูล ในแต่ละครั้งอาจมีข้อมูลที่ต้องทำการถ่ายโอนเป็นจำนวนมาก ซึ่งเป็นเหตุให้ใช้เวลาในการถ่ายโอนข้อมูลค่อนข้างมาก ดังนั้นเพื่อให้ไม่เกิดข้อผิดพลาดของการใช้งานคลังข้อมูลผู้ดูแลระบบควรทำการปิดคลังข้อมูลระหว่างการถ่ายโอนข้อมูล ซึ่งในการโหลดข้อมูลนั้นเราควรหาช่วงเวลาที่เหมาะสมซึ่งไม่ส่งผลกระบบต่อผู้ใช้คลังข้อมูล ในการโหลดข้อมูลถ้าขนาดของข้อมูลมีจำนวนมากเราอาจจะพิจารณาที่จะแบ่งส่วนของการถ่ายโอนข้อมูลออกเป็นส่วนที่เล็กลงหลาย ๆ ส่วน เพื่อที่จะได้ใช้เวลาในถ่ายโอนในแต่ละครั้งน้อยลง การแบ่งส่วนของการถ่ายโอนข้อมูลมีข้อดีอยู่สองข้อ คือ

(1) เราอาจจะสามารถทำการถ่ายโอนข้อมูลแบบขนานได้ (loads in parallel)

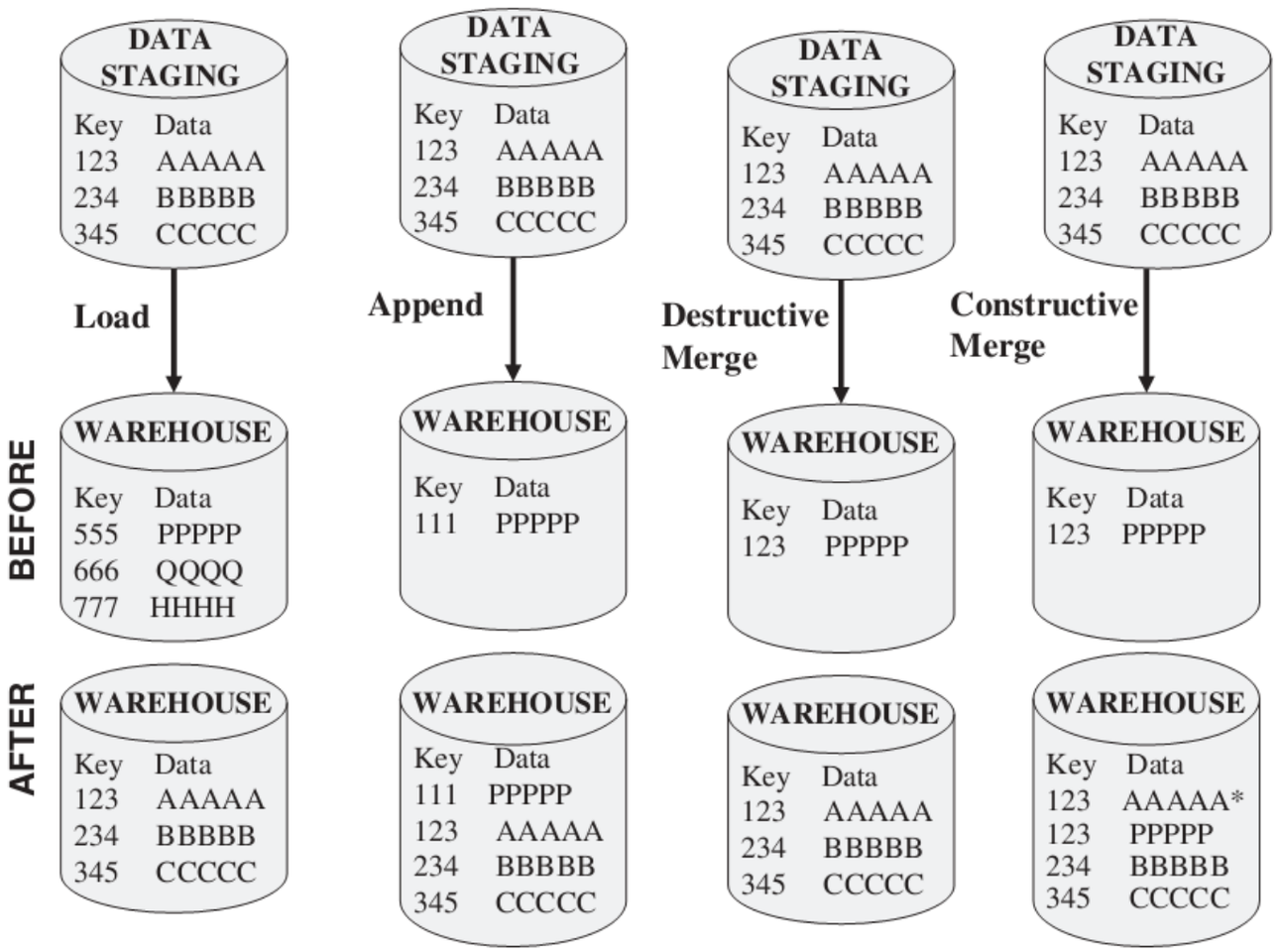
(2) เราอาจจะสามารถทำการสำรองข้อมูลในส่วนที่ไม่ได้ใช้จากคลังข้อมูลได้ (back up data in data warehouse) แล้วทำการถ่ายโอนข้อมูลส่วนอื่น ๆ เข้าสู่คลังข้อมูลได้ ซึ่งในการโหลดข้อมูลในแต่ละครั้งเราไม่สามารถคาดคะเนเวลาที่ใช้ในการถ่ายโอนข้อมูลได้ โดยเฉพาะอย่างยิ่งการทำ initial load และ full refresh

ในการถ่ายโอนข้อมูล เราอาจจะไม่ได้ทำการถ่ายโอนข้อมูลสำเร็จทุกครั้งไป การถ่ายโอนเรคคอร์ดหนึ่ง ๆ เข้าสู่ fact table ของคลังข้อมูลอาจมีปัญหาเกิดขึ้น เนื่องจาก concatenated key อาจผิด และไม่สอดคล้องกับ dimension tables เมื่อเกิดข้อผิดพลาดในการถ่ายโอนข้อมูลเกิดขึ้น เราจำเป็นต้องเตรียมกระบวนการสำหรับจัดการกับเรคคอร์ดที่ไม่ถูกโหลดเข้าสู่คลังข้อมูล และควรมีแผนสำหรับการประกันคุณภาพของการถ่ายโอนข้อมูล นอกจากความถูกต้องและความสมบูรณ์ของการโหลดข้อมูลแล้ว

ยังมีปัจจัยหนึ่งที่สำคัญมาก คือ การเลือกวิธีในการถ่ายโอนข้อมูลที่ซึ่งจะต้องสอดคล้องกับสถาปัตยกรรมของคลังข้อมูลว่าเป็นแบบใด เช่น กรณีที่ staging area และฐานข้อมูลของคลังข้อมูลอยู่ในเซิร์ฟเวอร์เดียวกัน เราจะสามารถถ่ายโอนข้อมูลได้โดยตรงโดยไม่เสียเวลาในการถ่ายโอนข้อมูลมาก แต่สำหรับกรณีอื่น ๆ เราจะต้องทำการเลือกวิธีในการถ่ายโอนข้อมูลว่าจะส่งข้อมูลทางใดระหว่าง web, FTP, และ database link ซึ่งแต่ละวิธีใช้ bandwidth ไม่เท่ากัน ถ้าการถ่ายโอนข้อมูลมีการใช้ bandwidth ค่อนข้างมากเราอาจต้องประยุกต์ใช้การบีบอัดข้อมูล (data compression) เข้าช่วยด้วย (ในกรณีที่ staging area และฐานข้อมูลไม่ได้อยู่ในเซิร์ฟเวอร์เดียวกัน การใช้ database link จะมีประโยชน์มาก)

เทคนิคในการประยุกต์ใช้ข้อมูล

ดังที่กล่าวมาแล้วข้างต้น การถ่ายโอนข้อมูลที่นิยมใช้กันในปัจจุบันจะมีอยู่ด้วยกัน 3 วิธี ได้แต่ 1) Initial Load, 2) Incremental Load และ 3) Full Refresh ซึ่งแต่ละวิธีจะถูกเรียกใช้ในเหตุการณ์หรือสภาวะแวดล้อมที่แตกต่างกัน ดังนั้นเพื่อให้การถ่ายโอนข้อมูลเป็นไปอย่างมีประสิทธิภาพ มีความถูกต้อง และลดการซ้ำซ้อนของข้อมูล วิธีการถ่ายโอนข้อมูลควรจะต้องประยุกต์ใช้เทคนิคดังต่อไปนี้ (แสดงดังรูปที่ 8-11)



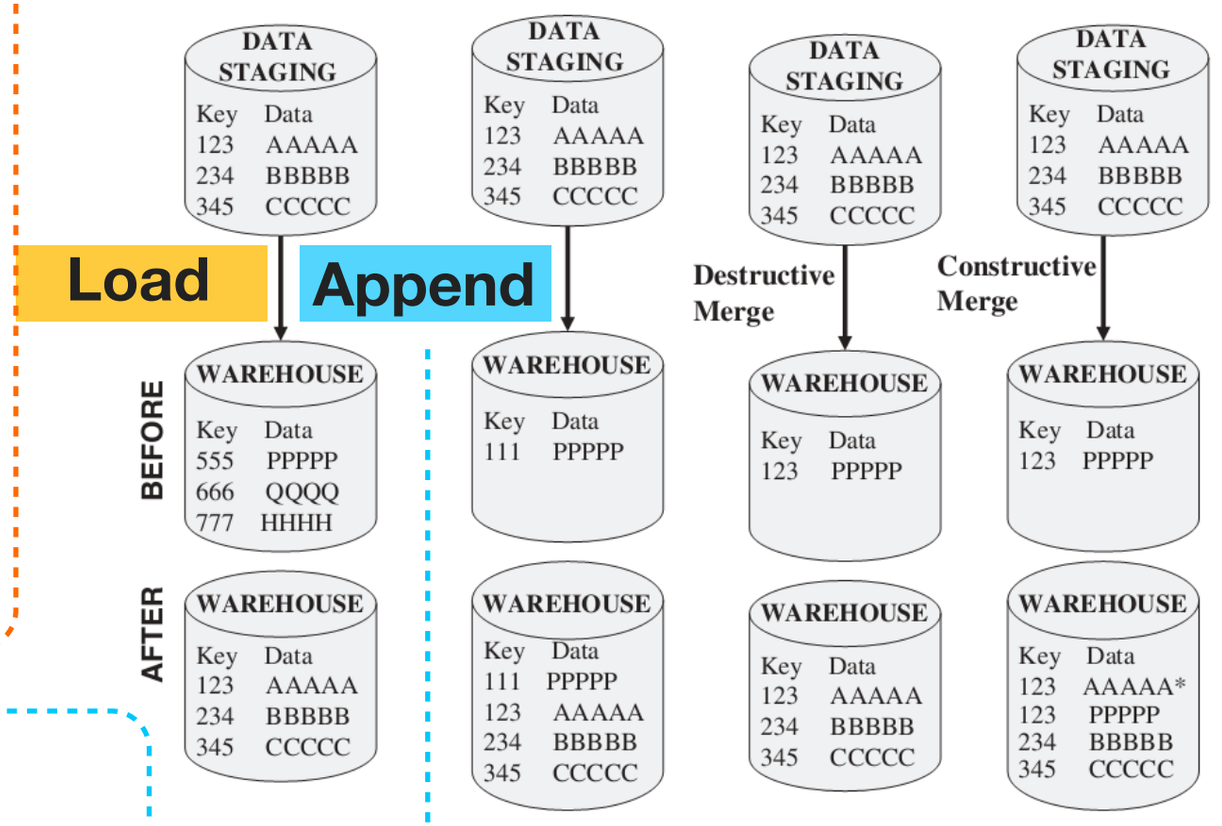
รูปที่ 8-11
เทคนิคการประยุกต์ใช้ข้อมูล

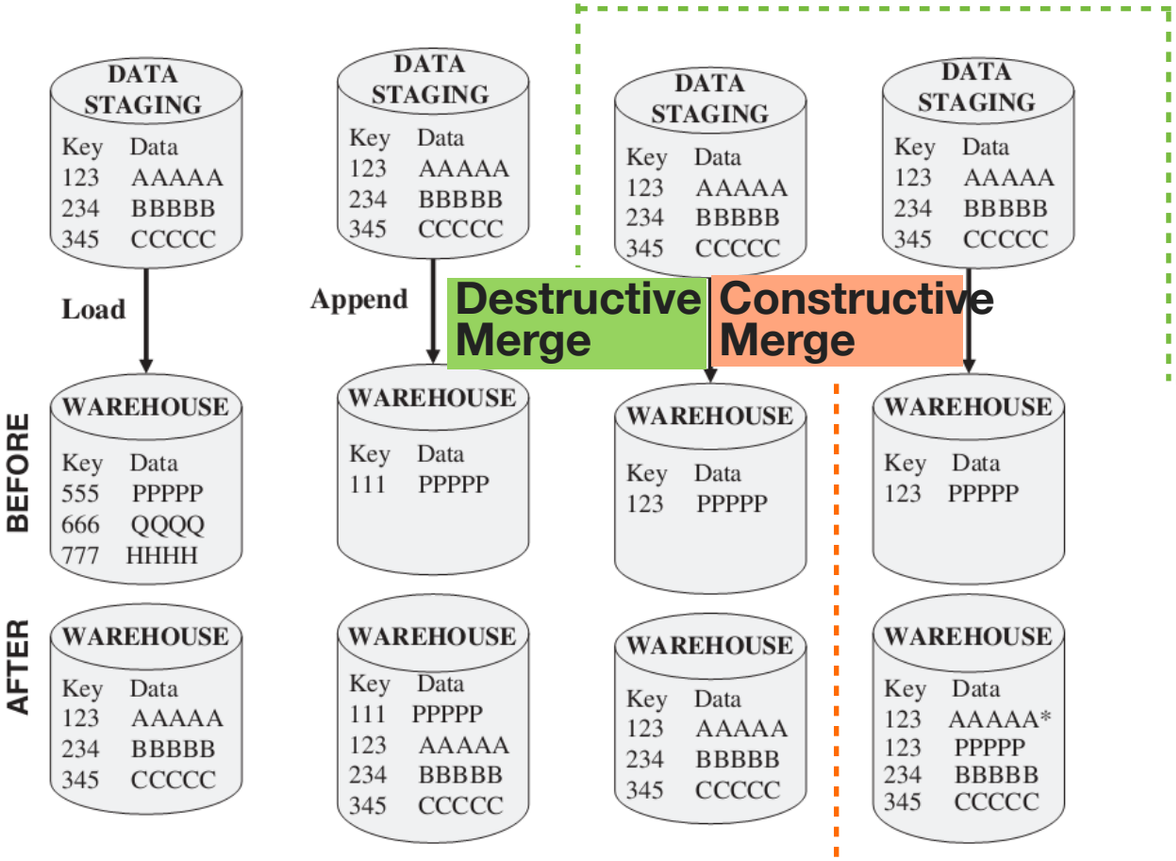
Load

ถ้าคลังข้อมูลมีตารางเป้าหมาย (target table) ที่จะทำการถ่ายโอนข้อมูลอยู่แล้วและในตารางนั้นมีข้อมูลอยู่ การทำงานของวิธี Load จะทำการล้างข้อมูลที่มีอยู่และถ่ายโอนข้อมูลใหม่เข้าไปในตาราง แต่ในกรณีที่ตารางข้อมูลที่มีอยู่แล้วไม่มีข้อมูลอยู่ก่อนหน้าจะทำการโหลดข้อมูลใหม่เข้าไปทันที

Append

ถ้าคลังข้อมูลมีตารางเป้าหมายที่จะทำการถ่ายโอนข้อมูลอยู่แล้วและในตารางมีข้อมูลอยู่ก่อนแล้ว การ Append จะทำการถ่ายโอนข้อมูลเข้าสู่ตารางนั้นโดยไม่มีเงื่อนไข แต่ถ้ากรณีที่ตารางมีข้อมูลที่ต้องการถ่ายโอนอยู่แล้ว (มีข้อมูลซ้ำ) เราสามารถเลือกว่าจะปฏิบัติอย่างไร ระหว่างเพิ่มข้อมูลสู่ตารางเป้าหมายโดยยอมให้มีการซ้ำกันของข้อมูลหรือไม่ต้องทำการเพิ่มข้อมูลใหม่สู่ตารางเนื่องจากมีข้อมูลอยู่แล้วและหลีกเลี่ยงความซ้ำซ้อน





Destructive Merge

ถ้าคีย์หลัก (Primary key) ของเรคคอร์ดใหม่ที่จะทำการถ่ายโอนข้อมูลตรงกับคีย์หลักของข้อมูลที่มีอยู่แล้วในตารางเป้าหมาย Destructive Merge จะทำการอัพเดทข้อมูลเรคคอร์ดที่มีคีย์หลักซ้ำกับเรคคอร์ดใหม่ที่จะทำการถ่ายโอน แต่ถ้าเรคคอร์ดใหม่มีคีย์หลักไม่ซ้ำกับเรคคอร์ดใด ๆ ที่อยู่ในตารางเป้าหมายจะทำการถ่ายโอนข้อมูลเข้าสู่ตาราง

Constructive Merge

ถ้าคีย์หลัก (Primary key) ของเรคคอร์ดใหม่ที่จะทำการถ่ายโอนข้อมูลตรงกับคีย์หลักของข้อมูลที่มีอยู่แล้วในตารางเป้าหมาย Constructive Merge จะทำการเก็บข้อมูลเก่าในตารางไว้ แล้วทำการถ่ายโอนข้อมูลใหม่เข้าสู่ตาราง และทำเครื่องหมายให้กับข้อมูลใหม่ที่ทำการถ่ายโอนว่าเป็นข้อมูลที่มาแทนข้อมูลเก่า

จากข้างต้น เราได้ทราบถึงวิธีในการประยุกต์ใช้ข้อมูลกับคลังข้อมูล จากนี้ไปเราจะมาพิจารณาว่าเราจะสามารถนำวิธีต่าง ๆ มาใช้กับแต่ละชนิดของการถ่ายโอนข้อมูลได้อย่างไร

Initial Load

ในการทำ Initial load ถ้าเราสามารถทำการถ่ายโอนข้อมูลได้ภายในครั้งเดียว เราจะสามารถใช้วิธีการ load ในการถ่ายโอนข้อมูลได้โดยตรง แต่ถ้าเราทำการแบ่ง Initial load ออกเป็นการทำงานหลาย ๆ ครั้งเราจะต้องใช้วิธีการ load ในการถ่ายโอนข้อมูลครั้งแรกแล้วใช้วิธีการ append ในการถ่ายโอนข้อมูลครั้งถัด ๆ ไป จากการที่เราทำการแบ่ง Initial load ออกเป็นการทำงานหลาย ๆ ครั้ง แสดงให้เห็นว่าข้อมูลที่ต้องทำการถ่ายโอนข้อมูลนั้นมีจำนวนมาก ดังนั้น การสร้างดัชนี (indexes) ของ Initial load จะใช้เวลาค่อนข้างมาก เราควรจะมองข้ามการสร้างดัชนีก่อนการถ่ายโอนข้อมูล เพื่อให้การโหลดนั้นสามารถทำงานได้อย่างรวดเร็ว จากนั้นค่อยทำการดัชนีขึ้นมาใหม่เมื่อการถ่ายโอนข้อมูลเสร็จสิ้น

Incremental Loads

ในหลาย ๆ แอพพลิเคชันอาจมีความเปลี่ยนแปลงเกิดขึ้นในแหล่งข้อมูล ณ ช่วงเวลาหนึ่ง ๆ ซึ่งเราอาจจำเป็นต้องเก็บช่วงเวลาของการเปลี่ยนแปลงนั้น ๆ ไว้ในคลังข้อมูล ถ้าช่วงเวลาเป็นส่วนหนึ่งของคีย์หลัก หรือถ้าช่วงเวลาถูกรวมอยู่ในการเปรียบเทียบระหว่างข้อมูลที่จะเข้ามาใหม่และข้อมูลที่มีอยู่เดิมในคลังข้อมูล Constructive Merge อาจถูกใช้ในการถ่ายโอนข้อมูล ซึ่งวิธีการนี้จะสามารถช่วยในการเก็บรักษาช่วงเวลาของการเปลี่ยนแปลงข้อมูล แต่ในกรณีที่ข้อมูลใน dimension table มีการเปลี่ยนแปลงอย่างช้าๆเรคคอร์ดที่ถูกเก็บอยู่ในคลังข้อมูลควรจะถูกแทนที่ด้วยเรคคอร์ดที่จะเข้ามาใหม่ที่มีคีย์หลักตรงกัน ดังนั้นเราควรใช้ Destructive Merge ในการถ่ายโอนข้อมูล โดยที่การถ่ายโอนข้อมูลโดย Destructive Merge นั้นจะใช้งานได้กับทุกตารางที่เป็นเป้าหมายได้เมื่อข้อมูลเก่า ๆ ไม่ได้มีความสำคัญในแอพพลิเคชัน

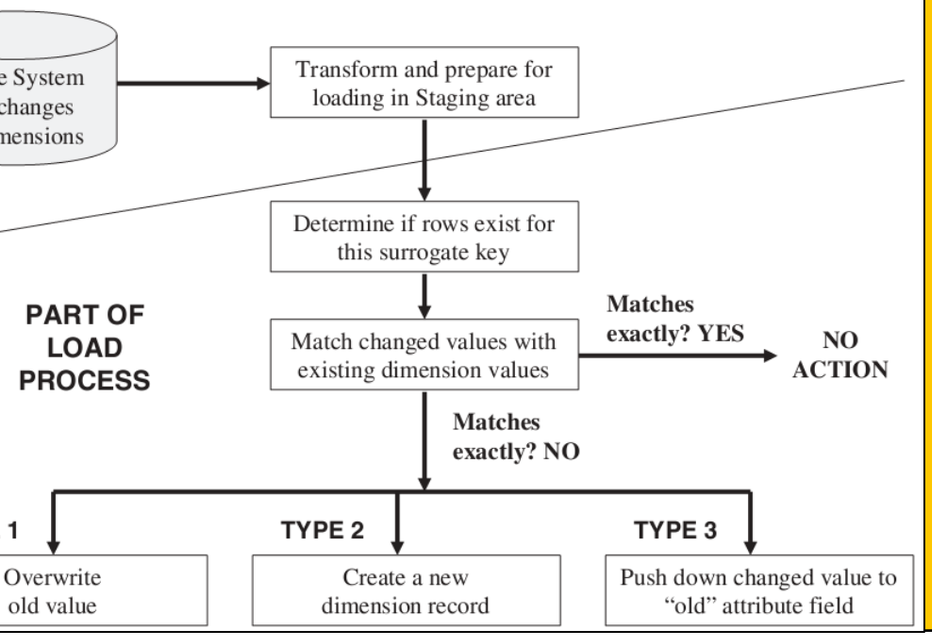
Full Refresh

วิธีในการถ่ายโอนข้อมูลของการทำ Full Refresh ก็จะเหมือนกับวิธีที่ใช้ใน Initial load ซึ่งก็คือจะใช้วิธี Load และ Append ในการถ่ายโอนข้อมูล การทำงานจะเริ่มจากการลบข้อมูลออกจากตารางในคลังข้อมูลก่อน จากนั้นใช้การ Load เพื่อทำการถ่ายโอนข้อมูลเข้าสู่ข้อมูลโดยตรง แต่ถ้าการถ่ายโอนข้อมูลมีการแบ่งส่วนการทำงานออกเป็นหลาย ๆ ครั้ง เราจะทำการใช้วิธีการ Load เพื่อทำการถ่ายโอนข้อมูลในครั้งแรก จากนั้นในครั้งต่อ ๆ ไปก็จะใช้วิธีการ Append ในการโหลดข้อมูลเข้าสู่คลังข้อมูล

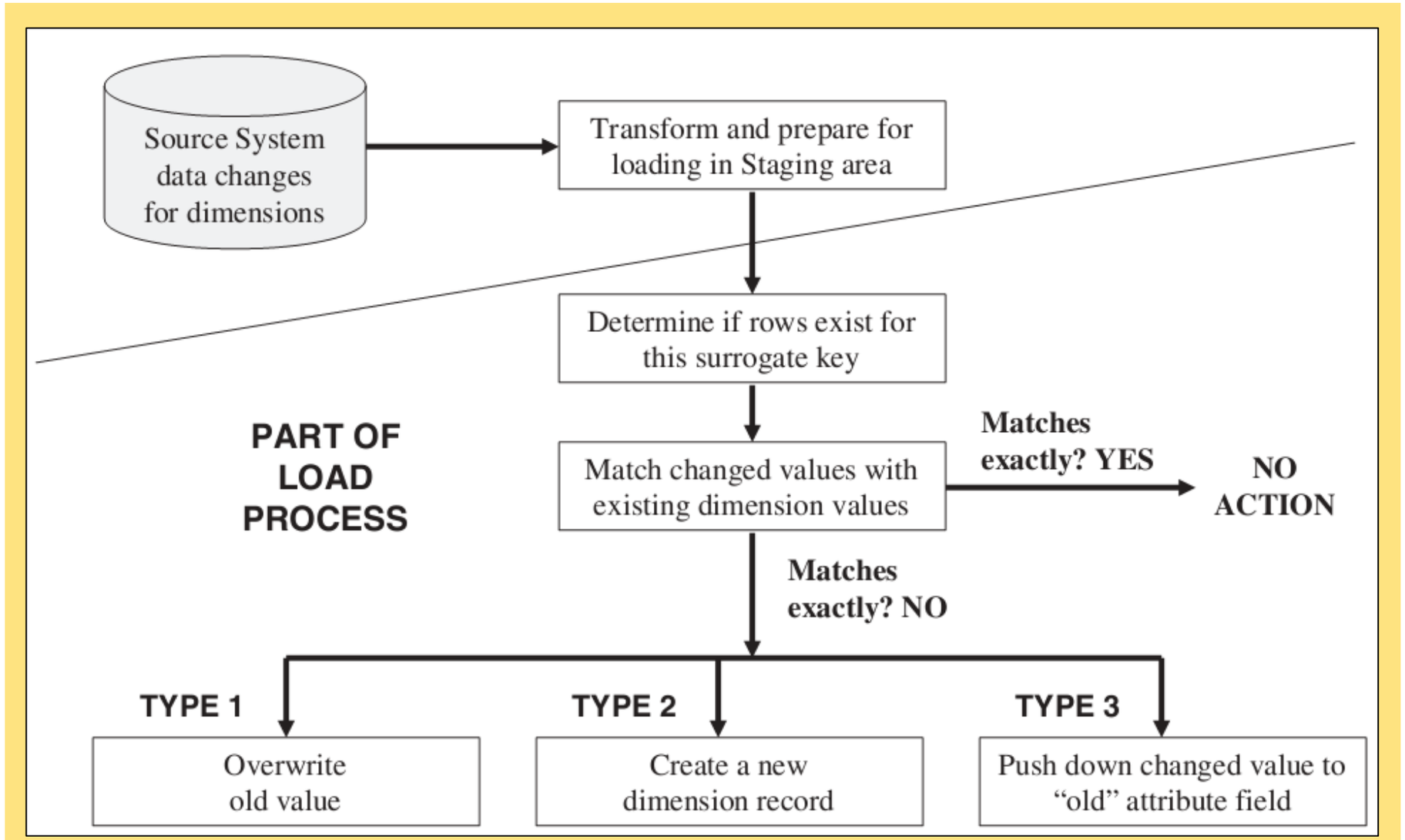
การถ่ายโอนข้อมูลเข้าสู่ Dimension table

Dimension table ที่ถูกเก็บอยู่ในคลังข้อมูล เช่น customer dimension, product dimension และ time dimension จะมีแอทริบิวที่เป็นมาตราวัดอยู่ (measurement) เช่น sales และ costs ข้อมูลเหล่านี้เป็นข้อมูลที่สอดคล้องกับข้อมูลที่ถูกเก็บไว้ในแหล่งข้อมูล ในการสร้างคลังข้อมูล เราจะต้องทำการถ่ายโอนข้อมูลจากแหล่งข้อมูลเข้าสู่ dimension table ซึ่งมีวิธีการ 2 วิธีด้วยกันคือ 1) การทำ Initial loading ของตารางต่าง ๆ และ 2) การประยุกต์ใช้ข้อมูลที่มีการเปลี่ยนแปลง (applying change)

จากสองวิธีที่ได้กล่าวข้างต้นยังมีอีกหนึ่งปัจจัยสำคัญที่เราจำเป็นต้องพิจารณาคือความสอดคล้องกันระหว่างคีย์ของเรค คอร์ดจากแหล่งข้อมูลและคีย์ของเรคคอร์ดในคลังข้อมูล อย่างที่เราทราบกันดีว่าเราไม่ได้ใช้คีย์ของเรคคอร์ดจากแหล่งข้อมูลไปเป็นคีย์ของข้อมูลในคลังข้อมูล เนื่องจากคีย์ของเรคคอร์ดในแหล่งข้อมูลอาจไม่มีความสมบูรณ์เพียงพอ ดังนั้นเมื่อเราทำการสกัดข้อมูลจากแหล่งข้อมูลแล้ว เราจะต้องสร้างระบบสำหรับการสร้างคีย์ขึ้นมาใหม่สำหรับใช้ในคลังข้อมูลด้วย ในการทำ Initial load และการโหลดครั้งต่อ ๆ ไป คีย์ของเรคคอร์ดจากแหล่งข้อมูลจะต้องถูกเปลี่ยนโดยระบบการสร้างคีย์สำหรับคลังข้อมูลก่อน ซึ่งการเปลี่ยนคีย์อาจอยู่ในขั้นตอนการเปลี่ยนแปลงข้อมูลหรือเป็นระบบที่สร้างขึ้นมาโดยเฉพาะก็เป็นได้



รูปที่ 8-12 เป็นการแสดงถึงขั้นตอนการถ่ายโอนข้อมูลที่มีการเปลี่ยนแปลงเข้าสู่ dimension table ซึ่งการทำงานจะแบ่งการทำงานออกเป็น 3 ชนิด คือ Type1, Type2 และ Type3 ซึ่งขั้นตอนการดำเนินงานทั้ง 3 จะถูกกระทำก็ต่อเมื่อข้อมูลจากแหล่งข้อมูลมีการเปลี่ยนแปลงไม่ตรงกับข้อมูลใน dimension table



รูปที่ 8- 12 การถ่ายโอนข้อมูลเมื่อมีความเปลี่ยนแปลงเกิดขึ้นกับ dimension tables

การถ่ายโอนข้อมูลเข้าสู่ Fact Tables

ในการถ่ายโอนข้อมูลเข้าสู่ fact table มีสิ่งหนึ่งที่จะต้องพิจารณาคือ คีย์หลักของ fact table (Primary key of fact table) จะเกิดจากการเรียงต่อกันของคีย์หลักของ dimension tables ด้วยเหตุนี้เราจึงต้องทำการถ่ายโอนข้อมูลเข้าสู่ dimension table ก่อน แล้วจึงค่อยทำการถ่ายโอนข้อมูลเข้าสู่แต่ละ fact table โดยก่อนที่จะทำการถ่ายโอนข้อมูลแต่ละเรคคอร์ดจากแหล่งข้อมูลเข้าสู่ fact table เราจะต้องทำการสร้างคีย์หลักให้กับเรคคอร์ดนั้น ๆ ก่อนการโหลดข้อมูลเข้าสู่ fact table จะมีเคล็ดลับมากมายดังนี้

- ระบุข้อมูลทางประวัติศาสตร์ที่เป็นประโยชน์และน่าสนใจสำหรับคลังข้อมูล (Identify historical data useful and interesting for the data warehouse)
- กำหนดและปรับแต่งกฎเกณฑ์ทางธุรกิจที่ถูกสกัดออกมา (Define and refine extract business rules)
- เก็บข้อมูลการตรวจสอบทางสถิติเพื่อทำการสร้างการเชื่อมโยงกลับไปยังระบบการดำเนินงาน (Capture audit statistics to tie back to the operational systems)
- ค้นหา surrogate key ของ fact table (Perform fact table surrogate key look-up)
- ปรับปรุงเนื้อหาใน fact table (Improve fact table content)
- เปลี่ยนแปลงโครงสร้างข้อมูล (Restructure the data)

ข้อสังเกต ในการถ่ายโอนข้อมูลเข้าสู่ fact table แบบ incremental loads มีดังต่อไปนี้

การ**สกัด**ข้อมูลแบบ incremental สำหรับ fact table

- ประกอบไปด้วย transaction ใหม่ๆ
  (Consist of new transactions)
- ประกอบไปด้วย transaction ที่มีการอัพเดทข้อมูล
  (Consist of updated transactions)
- ใช้ database transaction log สำหรับการเก็บข้อมูล
  (Use database transaction logs for data capture)

การ**ถ่ายโอน**ข้อมูลแบบ incremental สำหรับ fact table

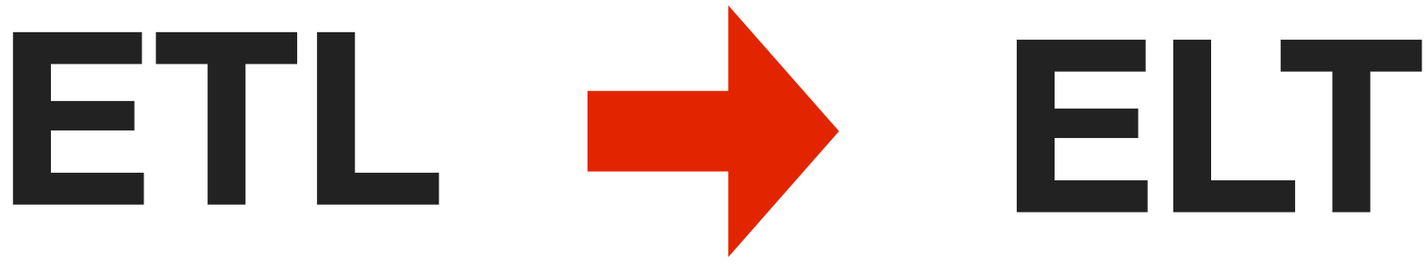
- ทำการโหลดข้อมูลบ่อยที่สุดเท่าที่เป็นไปได้
  (Load as frequently as possible)
- ใช้การแบ่งส่วนและการทำดัชนีเข้าช่วย
  (Use partitioned files and indexes)
- ประยุกต์ใช้การทำงานแบบขนานในการทำงาน
  (Apply parallel processing techniques)

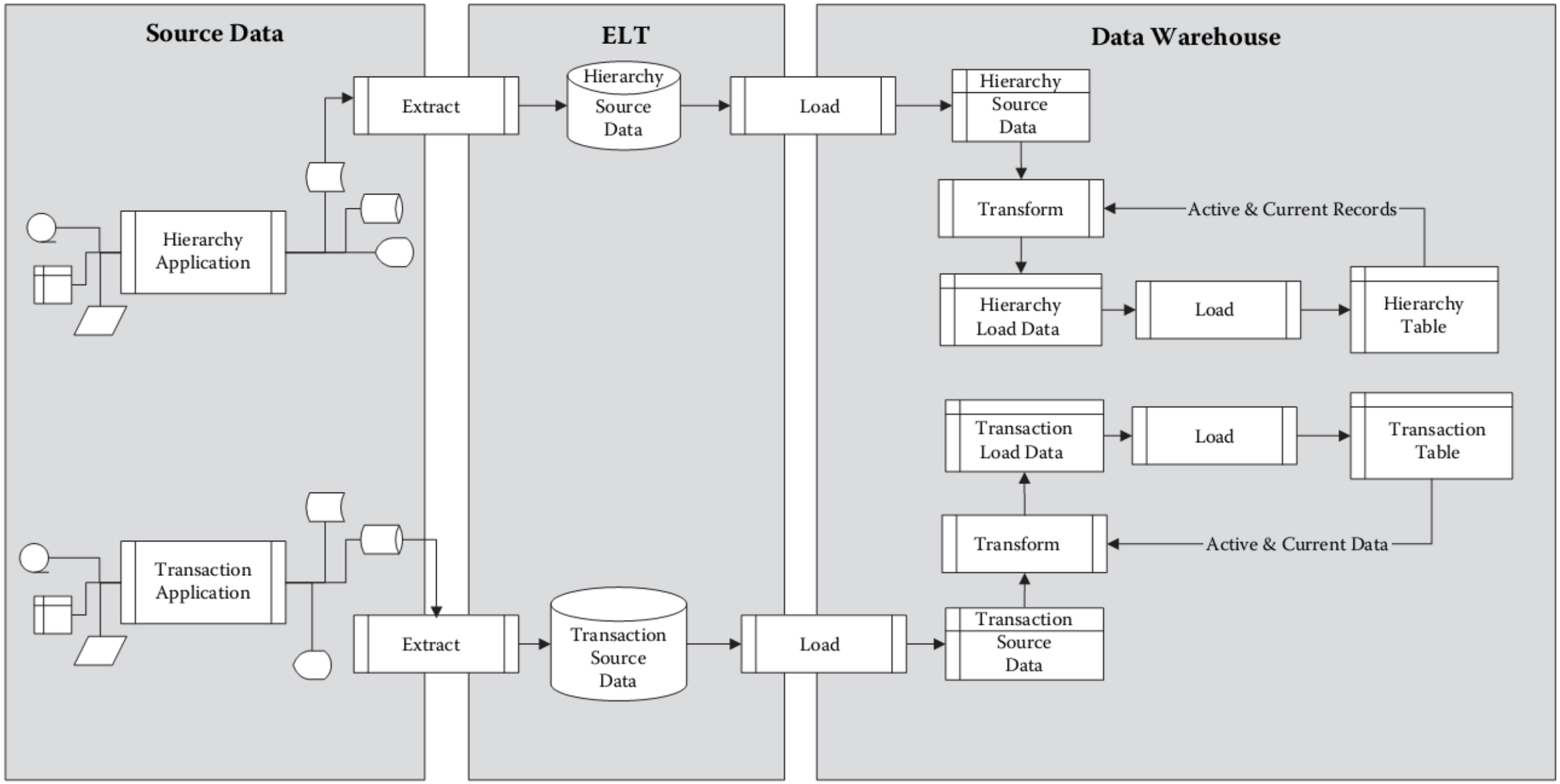
SECTION 8

การสลับตำแหน่งการทำงานจาก “อีทีแอล” เป็น “อีแอลที”

การสลับตำแหน่งการทำงานจาก “อีทีแอล” เป็น “อีแอลที”

“อีแอลที” เป็นกระบวนการสกัดข้อมูลจากแหล่งข้อมูลแล้วทำการถ่ายโอนข้อมูลเข้าสู่คลังข้อมูลเช่นเดียวกับ “อีทีแอล” แต่แตกต่างกันที่ลำดับของกระบวนการในการทำงาน การทำงานของอีแอลทีจะเริ่มจากการสกัดข้อมูลจากแหล่งข้อมูล แล้วถ่ายโอนข้อมูลเข้าสู่คลังข้อมูลโดยตรง โดยทำการเก็บข้อมูลไว้ที่ RDBMS (Relational DataBase Management Systems) ของคลังข้อมูล จากนั้นจะทำการเปลี่ยนแปลงข้อมูลที่ถูกเก็บไว้ใน RDBMS ท้ายสุดเราจะทำการถ่ายโอนข้อมูลภายในคลังข้อมูลเพื่อใช้งาน รายละเอียดของการทำงานของอีแอลทีจะสามารถแสดงได้ดังรูปที่ 8-13





รูปที่ 8-13 ตัวอย่างการทำงานของ ELT (Extract-Load-Transform)

SECTION 9

คำถามท้ายบท

1. จงอธิบายเหตุผลว่าเพราะเหตุใดฟังก์ชันอีทีแอลถึงมีความสำคัญต่อการทำงานของคลังข้อมูล
2. จงอธิบายเหตุผลว่าเพราะเหตุใดฟังก์ชันอีทีแอลถึงใช้เวลาในการทำงานค่อนข้างมาก
3. แนวปฏิบัติในการสกัดข้อมูลเป็นอย่างไร
4. จงอธิบายถึงความแตกต่างระหว่างการใช้และไม่ใช้ staging area สำหรับการสกัด เปลี่ยนแปลงและถ่ายโอนข้อมูลเข้าสู่คลังข้อมูล
5. จงอธิบายเกี่ยวกับลักษณะของข้อมูลที่มักพบในระบบการดำเนินงาน
6. จงยกตัวอย่างวิธีในการสกัดข้อมูล อย่างน้อย 2 วิธี
7. จงอธิบายเหตุผลการของสร้างคีย์ของผลลัพธ์ขึ้นใหม่
8. จงระบุถึงปัญหาที่มักพบในการรวมข้อมูลเข้าด้วยกัน รวมถึงวิธีการแก้ปัญหา
9. จงยกตัวอย่างฟังก์ชันการเปลี่ยนแปลงเปลี่ยนรูปข้อมูล อย่างน้อย 5 ฟังก์ชัน
10. จงอธิบายการทำงานและความแตกต่างระหว่าง initial load, incremental load และ full refresh
11. จงอธิบายเกี่ยวกับวิธีในการประยุกต์ใช้ข้อมูลในการจัดเก็บข้อมูลที่มีด้วยกัน 4 วิธี

CHAPTER 9

บทบาทสำคัญของเมตาดาต้า

SECTION 1

แผนการสอนประจำบท

วัตถุประสงค์ของบทเรียน

- ศึกษาว่าเพราะเหตุใดเมตาดาต้าถึงมีความสำคัญกับคลังข้อมูล
- ทำความเข้าใจเกี่ยวกับความต้องการของเมตาดาต้า
- ศึกษาเกี่ยวกับเมตาดาต้าที่สอดคล้องกับ 3 ฟังก์ชันการทำงานหลักของคลังข้อมูล
- ศึกษาเกี่ยวกับชนิดของเมตาดาต้า
- ศึกษาเกี่ยวกับทางเลือกในการจัดการเมตาดาต้า

เนื้อหาของบทเรียน

เนื้อหาในบทนี้จะประกอบด้วย นิยามเบื้องต้นของเมตาดาต้า ความต้องการเมตาดาต้าสำหรับการสร้างและการใช้งานคลังข้อมูล การแบ่งชนิดของเมตาดาต้าตามฟังก์ชันการทำงานหลัก เมตาดาต้าเชิงธุรกิจ เมตาดาต้าเชิงเทคนิค อุปสรรคและความท้าทายในการจัดเก็บเมตาดาต้า ขั้นตอนการจัดเก็บเมตาดาต้า

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารประกอบการสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

กิจกรรมการเรียน-การสอน

- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารประกอบการสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

การวัดและประเมินผล

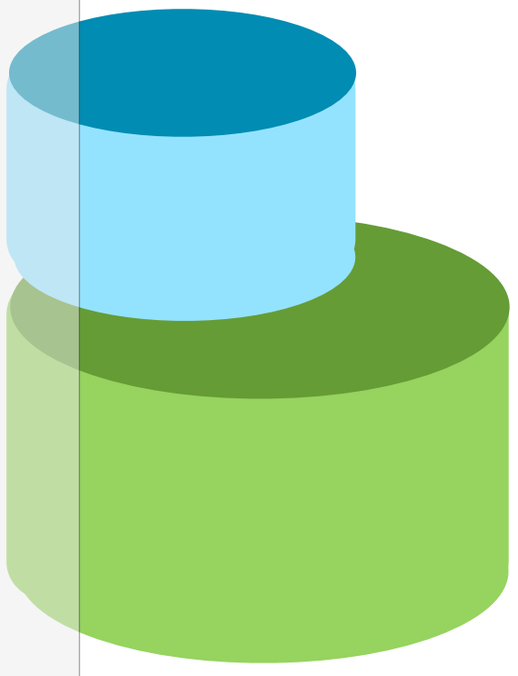
- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

SECTION 2

บทนำ

เมตาดาต้าในคลังข้อมูลอาจหมายถึงคำตอบสำหรับคำถามต่าง ๆ ที่เกี่ยวกับข้อมูลที่ถูกเก็บอยู่ในคลังข้อมูลว่ามีข้อมูลอะไรบ้าง มีลักษณะเป็นอย่างไร และอื่น ๆ เมตาดาต้าส่วนใหญ่จะถูกเก็บไว้ที่พื้นที่สำหรับจัดเก็บเมตาดาต้า (Metadata repository) โดยเมตาดาต้าสามารถนิยามได้หลายแบบขึ้นอยู่กับผู้สร้าง ผู้ใช้งาน หรือผู้ดูแลระบบจะทำการนิยาม โดยในบางแง่มุมเมตาดาต้า คือ

- ข้อมูลเกี่ยวกับข้อมูล (Data about data)
- สารบัญสำหรับข้อมูล (Table of contents for the data)
- แคตตาล็อกสำหรับข้อมูล (Catalog for the data)
- สมุดแผนที่ของคลังข้อมูล (Data warehouse atlas)
- แผนงานของคลังข้อมูล (Data warehouse roadmap)
- พจนานุกรมของคลังข้อมูล (Data warehouse dictionary)
- ไดเรกทอรีของคลังข้อมูล (Data warehouse directory)



จากนิยามข้างต้นเราจะเห็นว่าเมตาดาต้าสามารถนิยามได้หลายแบบ แต่โดยแท้จริงแล้วเมตาดาต้าหมายถึงอะไร? นิยามใดจากข้างต้นที่มีความหมายใกล้เคียงกับเมตาดาต้ามากที่สุด? เพื่อที่จะตอบคำถามเหล่านี้ ลองพิจารณาตัวอย่างดังต่อไปนี้

สมมติว่าผู้ใช้ต้องการทราบถึงตารางหรือเอนทิตีของข้อมูลลูกค้าที่ถูกเก็บอยู่ในคลังข้อมูลเพื่อที่จะสร้างคิวรีสำหรับเรียกดูข้อมูลลูกค้า จากความต้องการข้างต้น ผู้ใช้จะต้องการข้อมูลที่เกี่ยวข้องกับเอนทิตีหรือตารางข้อมูลลูกค้าที่เก็บไว้ แล้วตกลงเมตาดาต้าคืออะไร?

เพื่อที่จะตอบคำถาม ลองพิจารณารูปที่ 9-1 ที่แสดงถึงเมตาดาต้าของเอนทิตีของลูกค้า ซึ่งจะอธิบายถึงเนื้อหาข้อมูลที่เป็นเมตาดาต้าของข้อมูลลูกค้าอย่างละเอียดในแง่มุมต่างๆ

Entity Name:	Customer
Alias Names:	Account, Client

Definition:	A person or an organization that purchases goods or services from the company.
Remarks:	Customer entity includes regular, current, and past customers.
Source Systems:	Finished Goods Orders, Maintenance Contracts, Online Sales.

Create Date:	January 15, 2006
Last Update Date:	January 21, 2008
Update Cycle:	Weekly
Last Full Refresh Date:	December 29, 2007
Full Refresh Cycle:	Every six months
Data Quality Reviewed:	January 25, 2008
Last Deduplication:	January 10, 2008
Planned Archival:	Every six months

รูปที่ 9-1 ตัวอย่างเมตาดาต้าที่เกี่ยวข้องกับข้อมูลลูกค้า

SECTION 3

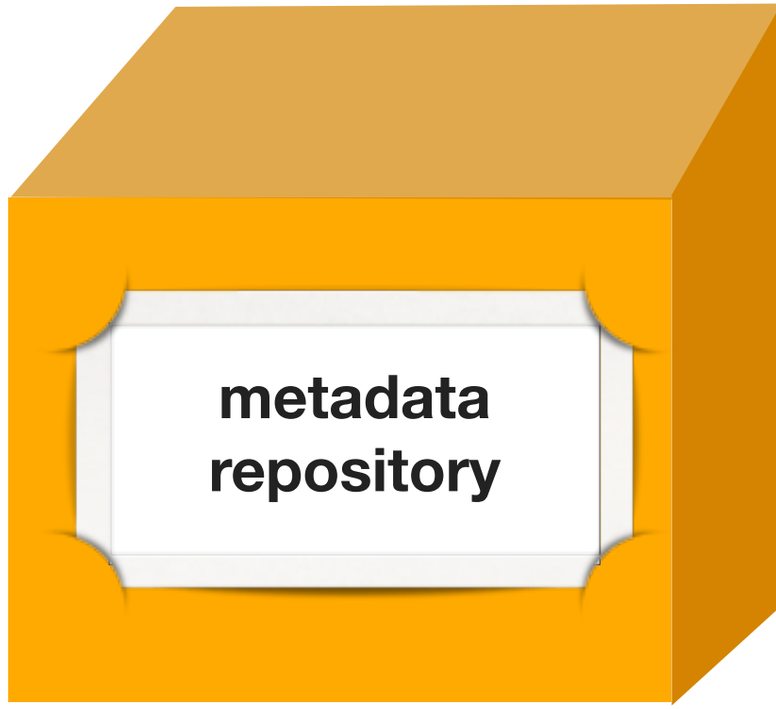
ความต้องการเมตาดาต้าสำหรับการสร้างและการใช้งานคลังข้อมูล

ความต้องการเมตาดาต้าสำหรับการสร้างและการใช้งานคลังข้อมูล

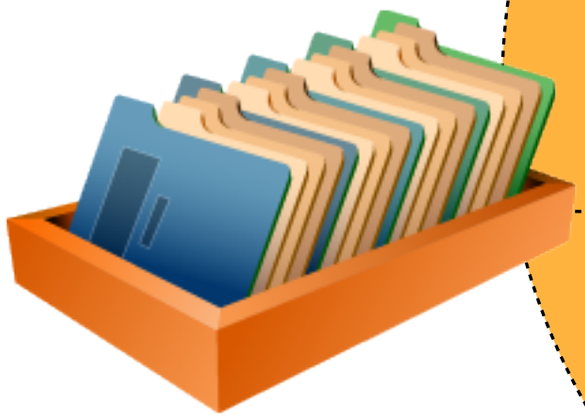
เมตาดาต้าที่ดีและเหมาะสมจะเป็นสิ่งจำเป็นสำหรับการใช้คลังข้อมูล การสร้างหรือต่อเติมคลังข้อมูล รวมถึงการดูแลรักษาคลังข้อมูลด้วย ซึ่งแต่ละการทำงานจะต้องการเมตาดาต้าที่แตกต่างกันดังนี้

เมตาดาต้าสำหรับการใช้คลังข้อมูล

ระบบการดำเนินงานและระบบคลังข้อมูลจะมีความแตกต่างที่สำคัญอย่างหนึ่งคือการใช้งานหรือการเข้าถึงข้อมูล โดยทั่วไปของระบบดำเนินการเราจะทราบถึงเนื้อหาหรือรายละเอียดของข้อมูลในฐานข้อมูลได้ค่อนข้างยากและการเข้าถึงเนื้อหาของข้อมูลต่าง ๆ จะไม่ค่อยยืดหยุ่นเท่าที่ควร ผู้ใช้จะสามารถเข้าถึงข้อมูลผ่านทางหน้าจอที่เรียกว่า GUI (Graphic User Interface) และผ่านทางรูปแบบรายงานที่มีการจัดเตรียมไว้แล้วเท่านั้น ผู้ใช้ไม่สามารถทำการเปลี่ยนแปลงการกระทำการกับข้อมูลได้มากนัก และผู้ใช้จะไม่ทราบถึงรายละเอียดทั้งหมดของข้อมูลที่ถูกเก็บไว้ในฐานข้อมูล ถึงแม้ว่าในระบบการดำเนินงานจะมีการจัดเก็บดาต้าดิกชันนารีในขั้นตอนสร้างระบบ



แต่อย่างไรก็ดี ดาต้าดิกชันนารีจะเป็นข้อมูลที่เตรียมไว้สำหรับทีมพัฒนาระบบโดยเฉพาะ ผู้ใช้ทั่ว ๆ ไปไม่สามารถเข้าถึงข้อมูลเหล่านั้นได้ แต่สำหรับคลังข้อมูลจะแตกต่างจากระบบดำเนินการโดยสิ้นเชิง กล่าวคือ ผู้ใช้คลังข้อมูลต้องการที่จะได้รับประโยชน์สูงสุดจากคลังข้อมูล ซึ่งผู้ใช้ต้องการวิธีที่จะเรียกดูและตรวจสอบเนื้อหาของข้อมูลในคลังข้อมูลที่อาจมีความซับซ้อน



ในบางครั้งผู้ใช้จะต้องการที่จะทราบถึงความหมายของข้อมูลแต่ละรายการ ก่อนที่จะทำการประมวลผลคิวรีเพื่อทราบถึงข้อมูลสารสนเทศที่จะใช้ช่วยในการตัดสินใจ จากความต้องการดังกล่าว เราควรทำการเก็บเมตาดาต้าไว้ที่ **“metadata repository”** เพื่อที่จะอนุญาตให้ผู้ใช้สามารถเรียกดูรายละเอียดของข้อมูลที่ถูกเก็บไว้ในคลังข้อมูลได้ ซึ่งจะช่วยให้ผู้ใช้มีความเข้าใจกับโครงสร้างของข้อมูลในคลังข้อมูล และจะช่วยป้องกันผู้ใช้จากความเข้าใจผิดเกี่ยวกับความหมายของข้อมูลในคลังข้อมูลได้

อย่างที่เราทราบดีว่าในยุคปัจจุบันคลังข้อมูลจะมีขอบเขตที่ใหญ่และกว้างไกลมาก ถ้าเราไม่มีการเก็บเมตาดาต้าที่เพียงพอ เราสามารถเปรียบผู้ใช้ได้กับเป็นคนพิการที่ไม่สามารถช่วยเหลือตัวเองในการใช้คลังข้อมูลเองได้

เมตาดาต้าสำหรับการสร้างคลังข้อมูล

ในการสร้างคลังข้อมูลเราจะต้องทราบถึงแหล่งข้อมูลและโครงสร้างของข้อมูลที่ถูกเก็บอยู่ในแหล่งข้อมูลนั้นๆ และจำเป็นต้องทราบถึงโครงสร้างและรายละเอียดของข้อมูลในคลังข้อมูลเพื่อที่จะสามารถสร้างฟังก์ชันการสกัดข้อมูลและฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลได้ ในการสร้างฟังก์ชันการทำงานดังกล่าว เราจำเป็นต้องใช้เมตาดาต้าที่เกี่ยวข้องกับแหล่งข้อมูล เมตาดาต้าสำหรับการเชื่อมโยงระหว่างแหล่งข้อมูลกับเป้าหมายในคลังข้อมูล และเมตาดาต้าสำหรับกฎการเปลี่ยนแปลงข้อมูล ในส่วนของการถ่ายโอนข้อมูลเราจำเป็นต้องทราบถึงเมตาดาต้าที่เกี่ยวกับวงจรหรือช่วงเวลาสำหรับการอัพเดทและการถ่ายโอนข้อมูล เป็นต้น

ถ้าเราพิจารณาในทุก ๆ ขั้นตอนของการสร้างคลังข้อมูลโดยละเอียด เราจะทราบว่าเมตาดาต้าเป็นสิ่งจำเป็นสำหรับการสร้างคลังข้อมูล และเราควรที่จะให้ความสำคัญกับเมตาดาต้ามากขึ้น

เมตาดาต้าสำหรับการดูแลและจัดการสิ่งต่าง ๆ ในคลังข้อมูล

ในปัจจุบันคลังข้อมูลจะมีความซับซ้อนและมีข้อมูลเป็นจำนวนมาก ดังนั้นเราจะไม่สามารถดูแลรักษาคลังข้อมูลได้ ถ้าเราไม่ทำการเก็บเมตาดาต้าไว้ เนื่องจากผู้ดูแลคลังข้อมูลไม่สามารถจำรายละเอียดของข้อมูลทั้งหมดที่เก็บอยู่ในคลังข้อมูลได้ จากที่กล่าวข้างต้นเมตาดาต้าจะเป็นคำตอบของคำถามที่เกี่ยวข้องกับข้อมูลในคลังข้อมูล

รูปที่ 9-2 แสดงถึงตัวอย่างของคำถามพื้นฐานที่เกี่ยวข้องกับการดูแลรักษาคลังข้อมูล ซึ่งจากคำถามดังกล่าว เราจะไม่สามารถดูแลระบบได้อย่างสมบูรณ์ถ้าเราไม่สามารถตอบคำถามในรูปได้ทั้งหมด และในรูปที่ 9-3 จะแสดงถึงผู้ใช้ที่ต้องการเรียกใช้เมตาดาต้าว่าประกอบไปด้วยผู้ใช้ฝ่ายใดบ้าง

Data Extraction/Transformation/Loading

How to handle data changes?
How to include new sources?
Where to cleanse the data? How to change the data cleansing methods?
How to cleanse data after populating the warehouse?
How to switch to new data transformation techniques?
How to audit the application of ongoing changes?

Data from External Sources

How to add new external data sources?
How to drop some external data sources?
When mergers and acquisitions happen, how to bring in new data to the warehouse?
How to verify all external data on ongoing basis?

Data Warehouse

How to add new summary tables?
How to control runaway queries?
How to expand storage?
When to schedule platform upgrades?
How to add new information delivery tools for the users?
How to continue ongoing training?
How to maintain and enhance user support function?
How to monitor and improve ad hoc query performance?
When to schedule backups?
How to perform disaster recovery drills?
How to keep data definitions up-to-date?
How to maintain the security system?
How to monitor system load distribution?

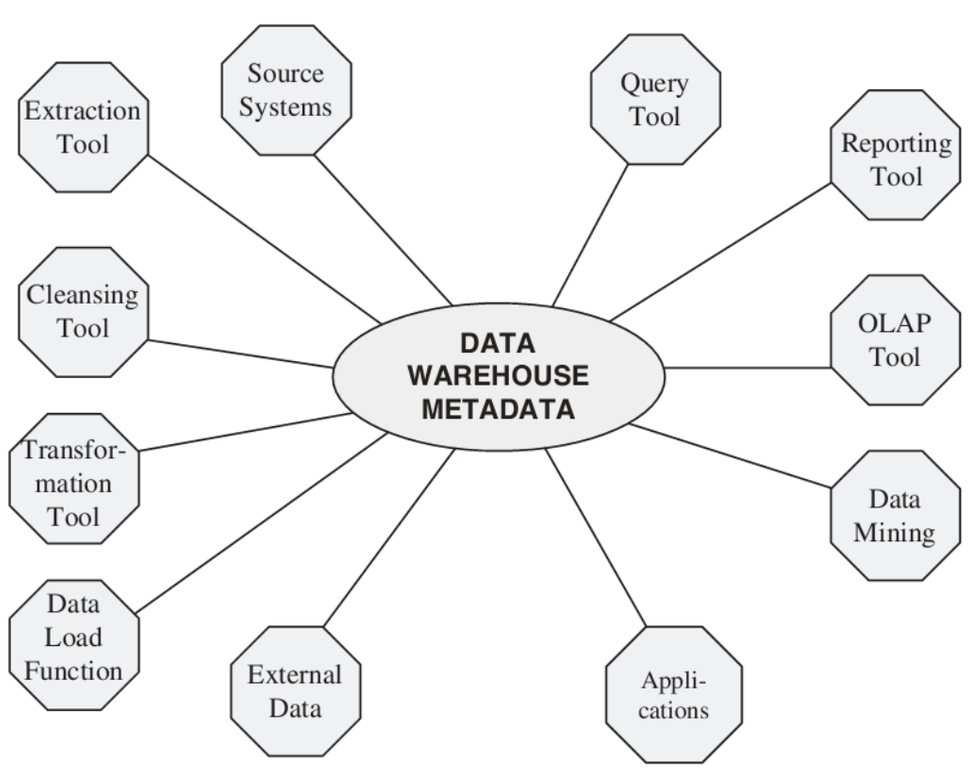
รูปที่ 9-2 คำถามที่มักจะพบเจอในการบำรุงและดูแลรักษาคลังข้อมูล

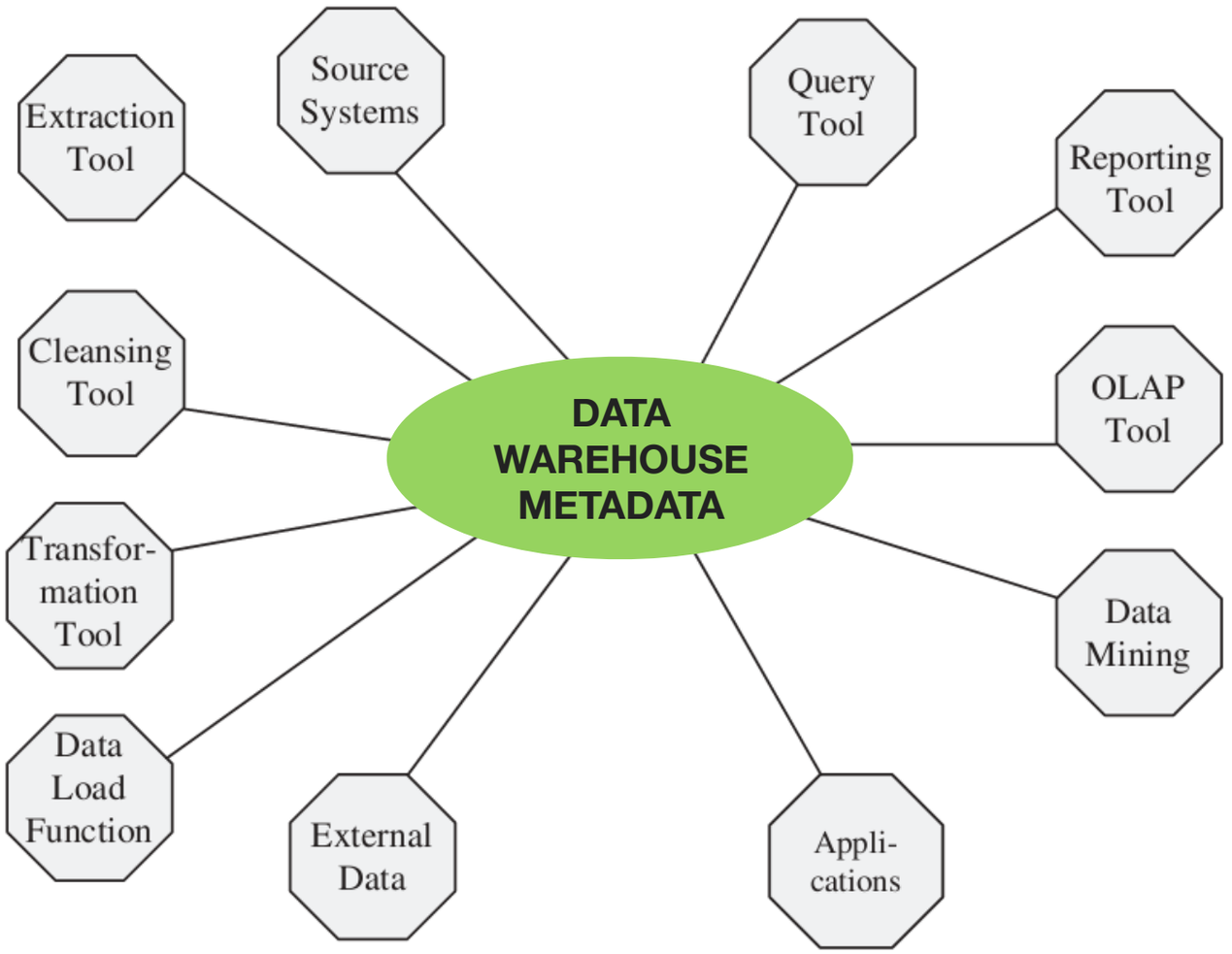
	IT Professionals	Power Users	Casual Users
			
Information Discovery	Databases, Tables, Columns, Server Platforms	Databases, Tables, Columns	List of Predefined Queries and Reports, Business Views
Meaning of Data	Data Structures, Data Definitions, Data Mapping, Cleansing Functions, Transformation Rules	Business Terms, Data Definitions, Data Mapping, Cleansing Functions, Transformation Rules	Business Terms, Data Definitions, Filters, Data Sources, Conversion , Data Owners
Information Access	Program Code in SQL, 3GL, 4GL, Front-end Applications, Security	Query Toolsets, Database Access for Complex Analysis	Authorization Requests, Information Retrieval into Desktop Applications such as Spreadsheets

รูปที่ 9-3 ตัวอย่างบุคคลที่ต้องการเมตาดาต้า

การเปรียบเทียบเมตาด้ากับศูนย์กลางประสาท

ในหลาย ๆ ขั้นตอนของการสร้างและการดูแลคลังข้อมูลจะมีการสร้างหรือการผลิตเมตาดาต้าเก็บไว้ในคลังข้อมูล ซึ่งเมตาดาต้าที่สร้างขึ้นจากขั้นตอนหนึ่ง ๆ อาจจะถูกใช้โดยขั้นตอนอื่น ๆ ด้วย เมตาดาต้าจะทำหน้าที่เหมือนกับศูนย์กลางประสาทของคลังข้อมูล ซึ่งจะถูกใช้ในการติดต่อสื่อสารระหว่างขั้นตอนการทำงานต่าง ๆ ของคลังข้อมูลด้วย





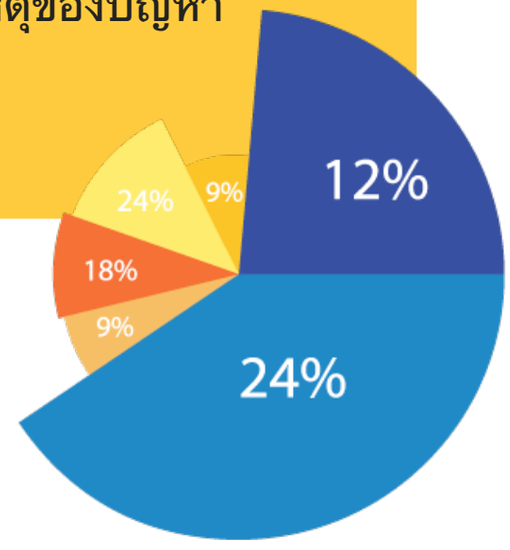
รูปที่ 9-4 แสดงถึงตำแหน่งที่ตั้งของเมตาดาต้าในคลังข้อมูล ซึ่งจากรูปเราจะสามารถกำหนดส่วนประกอบของเมตาดาต้าที่จะประยุกต์ใช้ในคลังข้อมูล และสามารถรับรู้ถึงชิ้นส่วนของเมตาดาต้าที่เป็นที่ต้องการของผู้ใช้ และผู้พัฒนาและผู้ดูแลคลังข้อมูลได้อีกด้วย

รูปที่ 9-4 ส่วนประกอบต่าง ๆ ของคลังข้อมูลที่มีเมตาดาต้า

เหตุผลที่เมตาดาต้ามีความสำคัญกับผู้ใช้งานคลังข้อมูล

ลองพิจารณาการใช้งานทั่ว ๆ ไปของผู้ใช้ที่กระทำกับคลังข้อมูล จะทำให้เราทราบถึงความสำคัญของเมตาดาต้าต่อผู้ใช้งาน

ตัวอย่างเช่น ผู้จัดการฝ่ายการตลาดทำการถามนักวิเคราะห์เชิงธุรกิจให้ทำการวิเคราะห์ปัญหาที่เกิดขึ้นกับการขายสินค้า ซึ่งผู้จัดการฝ่ายการตลาดต้องการจะทราบปัญหาเกี่ยวกับการเปิดสาขาใหม่ในเขตมิดเวสต์และในเขตตะวันออกเฉียงใต้ เขตละ 5 สาขา ซึ่งในช่วง 2 เดือนแรกของการเปิดให้บริการยอดขายมีการเพิ่มขึ้นอย่างมาก แต่หลังจากนั้นยอดขายก็ลดลงอย่างน่าใจหาย ซึ่งปัญหาดังกล่าวเป็นปัญหาที่สำคัญมากในการดำเนินธุรกิจและถ้าผู้จัดการทราบถึงสาเหตุของปัญหาจะสามารถเลือกวิธีการแก้ปัญหาที่ถูกต้องได้



จากความต้องการดังกล่าว ผู้วิเคราะห์เชิงธุรกิจจะต้องทำการสืบค้นข้อมูลจากคลังข้อมูล แต่อย่างไรก็ดีผู้วิเคราะห์ไม่ทราบถึงรายละเอียดของข้อมูลในคลังข้อมูล โดยเฉพาะอย่างยิ่งเขาไม่ทราบคำตอบของคำถามดังต่อไปนี้

ยอดขาย (จำนวนชิ้นสินค้าและจำนวนเงิน) ถูกเก็บในลักษณะใด ระหว่างการเก็บแบบแยกตามรายการที่ถูกซื้อหรือเก็บผลรวมของแต่ละรายการสินค้าสำหรับแต่ละวันในแต่ละสาขาของร้าน?

เราสามารถทำการวิเคราะห์ข้อมูลยอดขายของแต่ละสินค้า โปรโมชั่น ร้านค้า/สาขา และแต่ละเดือนได้หรือไม่?

?!$#@!?

เราสามารถทำการเปรียบเทียบยอดขายของเดือนนี้ปีนี้กับยอดขายของเดือนนี้เมื่อปีที่แล้วได้หรือไม่?

เราจะสามารถคำนวณผลกำไรได้อย่างไร? กฎในการทำธุรกิจคืออะไร?

ยอดขายนั้นถูกเก็บอยู่ที่ใด? จากแหล่งข้อมูลใด?

จากคำถามข้างต้น ถ้าไม่มีการเตรียมเมตาดาต้าไว้สำหรับผู้ใช้จะทำให้ผู้วิเคราะห์จะไม่เข้าใจเกี่ยวกับธรรมชาติของข้อมูล ซึ่งอาจทำให้เกิดความผิดพลาดในการแปลความหมายของผลลัพธ์ ถ้าเรามีการจัดเตรียมเมตาดาต้าอย่างเพียงพอจะทำให้ผู้วิเคราะห์เข้าใจถึงรายละเอียดของข้อมูล และเขาสามารถใช้งานคลังข้อมูลได้จากการเรียกดูเมตาดาต้า แต่อย่างไรก็ดี ทางผู้พัฒนาคลังข้อมูลจะต้องทำให้การเข้าถึงเมตาดาต้าสามารถทำได้โดยง่ายซึ่งเป็นสิ่งจำเป็นสำหรับผู้ใช้งาน

END-USERS

METADATA VITAL FOR END-USERS	
Data content Summary data Business dimensions Business metrics Navigation paths Source systems External data Data transformation rules Last update dates Data load/update cycles Query templates Report formats Predefined queries/reports OLAP data	**METADATA ESSENTIAL FOR IT**

รูปที่ 9-5 จะเป็นการสรุปความต้องการเมตาดาต้าจากผู้ใช้คลังข้อมูล ซึ่งจากรูปจะแสดงชนิดของเมตาดาต้าที่มีการเตรียมไว้ให้ผู้ใช้ได้ใช้งานอีกด้วย

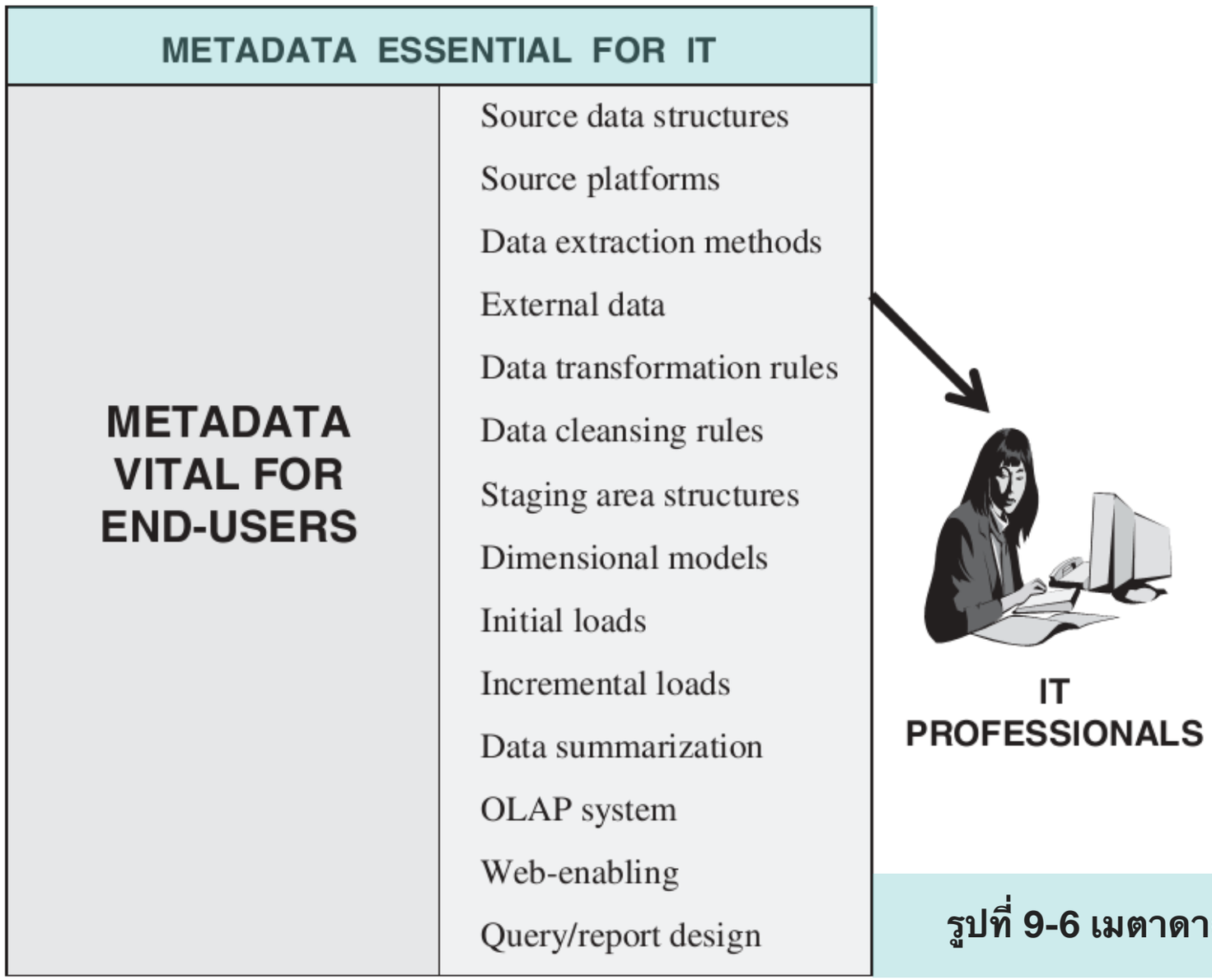
รูปที่ 9-5 เมตาดาต้าที่สำคัญสำหรับผู้ใช้

เหตุผลที่เมตาดาต้ามีความสำคัญกับผู้สร้างและผู้ดูแลคลังข้อมูล

เมตาดาต้ามีส่วนสำคัญอย่างมากในการสร้างคลังข้อมูล ซึ่งถ้าเราสามารถเรียกดูเมตาดาต้าที่เหมาะสมและมีคุณภาพได้ เราจะสามารถทำการออกแบบและทำการดูแลคลังข้อมูลได้เป็นอย่างดี ซึ่งเมตาดาต้าจะมีส่วนช่วยในขั้นตอนการสร้างคลังข้อมูลดังนี้

- **การสกัดข้อมูลจากแหล่งข้อมูล**
  (Data extraction from sources)
- **การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล**
  (Data transformation)
- **การทำความสะอาดข้อมูล**
  (Data scrubbing)
- **การรวมและสรุปข้อมูล**
  (Data aggregation and summarization)
- **การเก็บข้อมูลไว้ใน staging area**
  (Data staging)
- **การทำให้ข้อมูลมีความทันสมัย**
  (Data refreshment)
- **การออกแบบฐานข้อมูล**
  (Database design)
- **การออกแบบ query และรายงาน**
  (Query and report design)

รูปที่ 9-6 จะเป็นการสรุปการต้องการใช้เมตาดาต้าของผู้พัฒนาคลังข้อมูล ซึ่งในรูปจะแสดงถึงชนิดของเมตาดาต้าที่จัดเตรียมไว้สำหรับผู้พัฒนาคลังข้อมูล



รูปที่ 9-6 เมตาดาต้าที่สำคัญกับผู้ดูแลและผู้สร้างคลังข้อมูล

การทำงานของคลังข้อมูลที่เป็นไปอย่างอัตโนมัติ

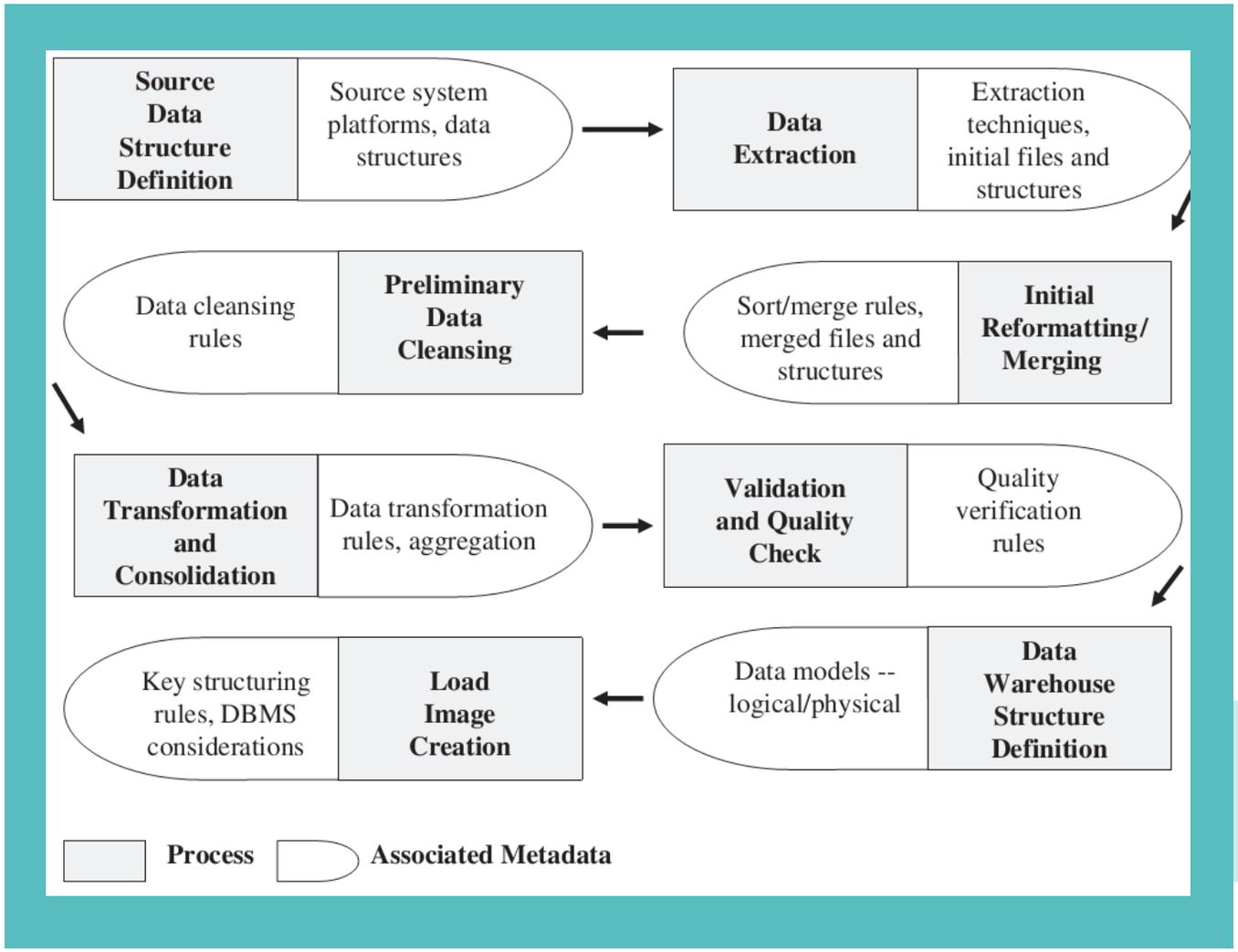
ในยุคแรกของการสร้างคลังข้อมูล เมตาดาต้าจะถูกเก็บอยู่ในรูปแบบของเอกสาร แต่ในยุคปัจจุบันได้มีการเปลี่ยนแปลงการเก็บเมตาดาต้าซึ่งเก็บไว้ที่ metadata repository ซึ่งในปัจจุบันเครื่องมือต่าง ๆ ไม่ว่าจะเป็นเครื่องมือเกี่ยวกับการสกัดข้อมูล เครื่องมือสำหรับทำการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล เครื่องมือสำหรับการตรวจสอบคุณภาพของข้อมูล และอื่น ๆ จะทำการเก็บข้อมูลเมตาดาต้าที่จำเป็นต่อกระบวนการทำงานของเครื่องมือนั้น ๆ ไว้

นอกจากนี้ เมื่อเราใช้เครื่องมือหลาย ๆ ชิ้นประกอบกันในการทำงาน เราจะสามารถใช้เครื่องมือหนึ่ง ๆ จากการเรียกดูข้อมูลเมตาดาต้าที่ถูกเก็บโดยอีกข้อมูลหนึ่งได้ ซึ่งจากกระบวนการทำงานดังกล่าวเมตาดาต้าจะเป็นข้อมูลที่สำคัญและจำเป็นต่อทุกกระบวนการทำงาน และเป็นส่วนช่วยให้การทำงานเป็นไปอย่างอัตโนมัติมากขึ้น โดยที่ถ้าเครื่องมือรู้ที่อยู่ของเมตาดาต้าก็จะมาสามารถอ่านข้อมูลเมตาดาต้าแล้วทำการประมวลผลได้อย่างอัตโนมัติ

ในการสร้างคลังข้อมูลจะใช้เครื่องมือต่าง ๆ ดังนี้

- การนิยามเกี่ยวกับโครงสร้างของข้อมูลที่อยู่ในแหล่งข้อมูล (Source data structure definition)
- การสกัดข้อมูล (Data extraction)
- การเปลี่ยนรูปแบบของข้อมูล รวมถึงการรวมข้อมูลเข้าด้วยกัน (Initial reformatting/merging)
- การทำความสะอาดข้อมูลเบื้องต้น (Preliminary data cleansing)
- การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล และการรวมข้อมูลให้เป็นหนึ่งเดียว (Data transformation and consolidation)
- การตรวจสอบความถูกต้อง และการตรวจสอบคุณภาพของข้อมูล (Validation and quality check)
- การนิยามเกี่ยวกับโครงสร้างของคลังข้อมูล (Data warehouse structure definition)
- การสร้างข้อมูลสำหรับการถ่ายโอน (Load image creation)

ขั้นตอนการทำงานจะมีเมตาดาต้าที่เกี่ยวข้องดังแสดงในรูปที่ 9-7 จากรูปแสดงถึงแต่ละขั้นตอนใน 8 ขั้นตอนที่กล่าวข้างต้น และเมตาดาต้าที่จะถูกเก็บไว้ในแต่ละขั้นตอนการทำงานด้วย ซึ่งเมตาดาต้านั้นเป็นสิ่งสำคัญสำหรับคลังข้อมูล เนื่องจากเมตาดาต้าจะทำให้ขั้นตอนการทำงานสามารถทำงานได้โดยอาศัยการอ่านข้อมูลเมตาดาต้า แต่อย่างไรก็ตาม เราจะต้องตระหนักถึงเครื่องมือที่อาจจะทำการเก็บเมตาดาต้าไว้ในรูปแบบเฉพาะสำหรับเครื่องมือนั้น ๆ



รูปที่ 9-7 การใช้เมตาดาต้าในการขับเคลื่อนขั้นตอนการทำงานต่าง ๆ ของคลังข้อมูล

การสร้างบริบทของข้อมูล

ลองจินตนาการเมื่อผู้ใช้คลังข้อมูลต้องการทำการประมวลผลคิวรีเพื่อค้นคืนข้อมูลการขายสินค้าจากสินค้า 3 ชนิด ในช่วงเวลา 7 วันแรกของเดือนเมษายนซึ่งยอดขายดังกล่าวเป็นยอดขายของเมืองทางใต้ทั้งหมด ดังนั้น คิวรีจะประกอบไปด้วยข้อมูลต่าง ๆ ดังนี้

Product = Widget-1 or Widget-2 or Widget-3, Region= ‘SOUTH’

Period = 04-01-2012 to 04-07-2012

ผลลัพธ์ที่ได้จะเป็นดังนี้

	Sale Units	Amount
Widget-1	25,335	253,550
Widget-2	16,978	254,670
Widget-3	7,994	271,796

ลองพิจารณาที่คิวรีและผลลัพธ์ที่ได้ ในการกำหนดพื้นที่ของการขายสินค้ามีเขตใดบ้างที่ถูกนับว่าเป็นเขตทางใต้ ? เขตที่ถูกนับเหล่านั้นเป็นเขตที่ผู้ใช้สนใจใช่หรือไม่ ? ความหมายของข้อมูลที่เป็น “SOUTH” ในคลังข้อมูลคืออะไร ? ข้อมูลวันที่ที่เป็น “04-01-2012” จะหมายถึง วันที่ 1 เมษายน 2012 หรือวันที่ 4 มกราคม 2012 ? ในคลังข้อมูลมีการจัดเก็บข้อมูลแบบใด ลองเปลี่ยนมุมมองมาเป็นการพิจารณาที่ยอดขายที่เป็นจำนวนชิ้นบ้าง ยอดขายที่ถูกแสดงเป็นผลลัพธ์จะเป็นยอดขายมีหน่วยเป็นอย่างไร จำนวนชิ้น แพ็ค หีบห่อ หรือเป็นไปตามมาตรวัดตาชั่ง เช่น ปอนด์ หรือ กิโลกรัม? ยอดขายที่แสดงถึงจำนวนเงินใช้สกุลใดระหว่าง $, £ หรือ € ?

จากคำถามข้างต้นผู้ใช้งานคลังข้อมูลจะสามารถทราบถึงความหมายที่แท้จริงของผลลัพธ์ที่ได้จากคลังข้อมูลได้อย่างไร? คำตอบคือ **“เมตาดาต้า”** ซึ่งเป็นสิ่งที่บ่งบอกความหมายของแต่ละข้อมูลว่ามีลักษณะเป็นอย่างไร หรือเราอาจจะมองได้ว่าเมตาดาต้านั้นสร้างเนื้อหาหรือคำอธิบายเกี่ยวกับข้อมูลแต่ละตัว ดังนั้น ผู้ใช้ ผู้พัฒนา และผู้ดูแลคลังข้อมูลจะต้องทำการแปลข้อมูลแต่ละรายการข้อมูลให้เป็นเมตาดาต้าด้วย

SECTION 4

การแบ่งชนิดของเมตาดาต้าตามฟังก์ชันการทำงานหลัก

การแบ่งชนิดของเมตาดาต้าตามฟังก์ชันการทำงานหลัก

ชนิดของเมตาดาต้าสามารถแบ่งได้หลายชนิด ผู้เขียนหนังสือหรือผู้สร้างคลังข้อมูลบางคนจะแบ่งชนิดของเมตาดาต้าตามการใช้งาน บางคนก็แบ่งชนิดของเมตาดาต้าตามผู้ใช้ว่าใครคือผู้ใช้งานเมตาดาต้านั้น ๆ ซึ่งในการแบ่งชนิดของเมตาดาต้าสามารถแบ่งได้ตามหัวข้อดังต่อไปนี้

แบ่งตามผู้ดูแลคลังข้อมูล ผู้ใช้คลังข้อมูล หรือผู้พัฒนาคลังข้อมูล (Administrator/end-user/optimization)

แบ่งตามการพัฒนาและการใช้งาน (Development/usage)

แบ่งตามที่อยู่ของเมตาดาต้า ว่าอยู่ในดาต้ามาร์ทหรืออยู่ใน workstation (In the data mart/at the workstation)

แบ่งตามการสร้าง การดูแล การจัดการ และการใช้งาน (Building/maintaining/managing/using)

แบ่งตามเชิงเทคนิคในการสร้างคลังข้อมูล หรือเชิงธุรกิจ (Technical/business)

แบ่งตามการทำงานที่เป็นการทำงานที่ติดต่อกับผู้ใช้หรือการทำงานเบื้องหลัง (Back room/front room)

แบ่งตามแหล่งข้อมูลว่ามาจากภายในหรือภายนอก (Internal/external)

จากที่ได้กล่าวมาแล้ว ในบทก่อนหน้า ฟังก์ชันการทำงานของคลังข้อมูลจะประกอบไปด้วย 3 ส่วนด้วยกันคือ

1. การได้มาซึ่งข้อมูล
2. การจัดเก็บข้อมูล
3. การเข้าถึง/ส่งผ่านข้อมูลไปยังผู้ใช้

ซึ่งทุก ๆ ขั้นตอนการทำงานของคลังข้อมูลจะต้องเกิดขึ้นใน 3 ส่วนนี้ แต่ละขั้นตอนการทำงานที่เกิดขึ้น 3 ส่วนของคลังข้อมูลจะทำการสร้างและเก็บเมตาดาต้าไว้และอาจใช้เมตาดาต้าเพื่อส่งต่อหรือส่งผลต่อการทำงานในขั้นตอนอื่น ๆ ด้วย

ดังนั้น จากที่กล่าวข้างต้น เราจะทำการแบ่งเมตาดาต้าออกเป็น 3 ชนิดตามส่วนประกอบของการทำงานของคลังข้อมูล เนื่องจากทุกขั้นตอนการทำงานของคลังข้อมูลจะต้องเกิดขึ้นในส่วนใดส่วนหนึ่งขององค์ประกอบนี้ เมื่อทำการแบ่งชนิดของเมตาดาต้าออกตามฟังก์ชันการทำงานจะทำให้เราจะทราบถึงการทำงานทั้งหมดของคลังข้อมูลรวมถึงข้อมูลที่เป็นเมตาดาทั้งหมดด้วย

ซึ่งการแบ่งข้อมูลเมตาดาต้าจะแบ่งได้ดังนี้

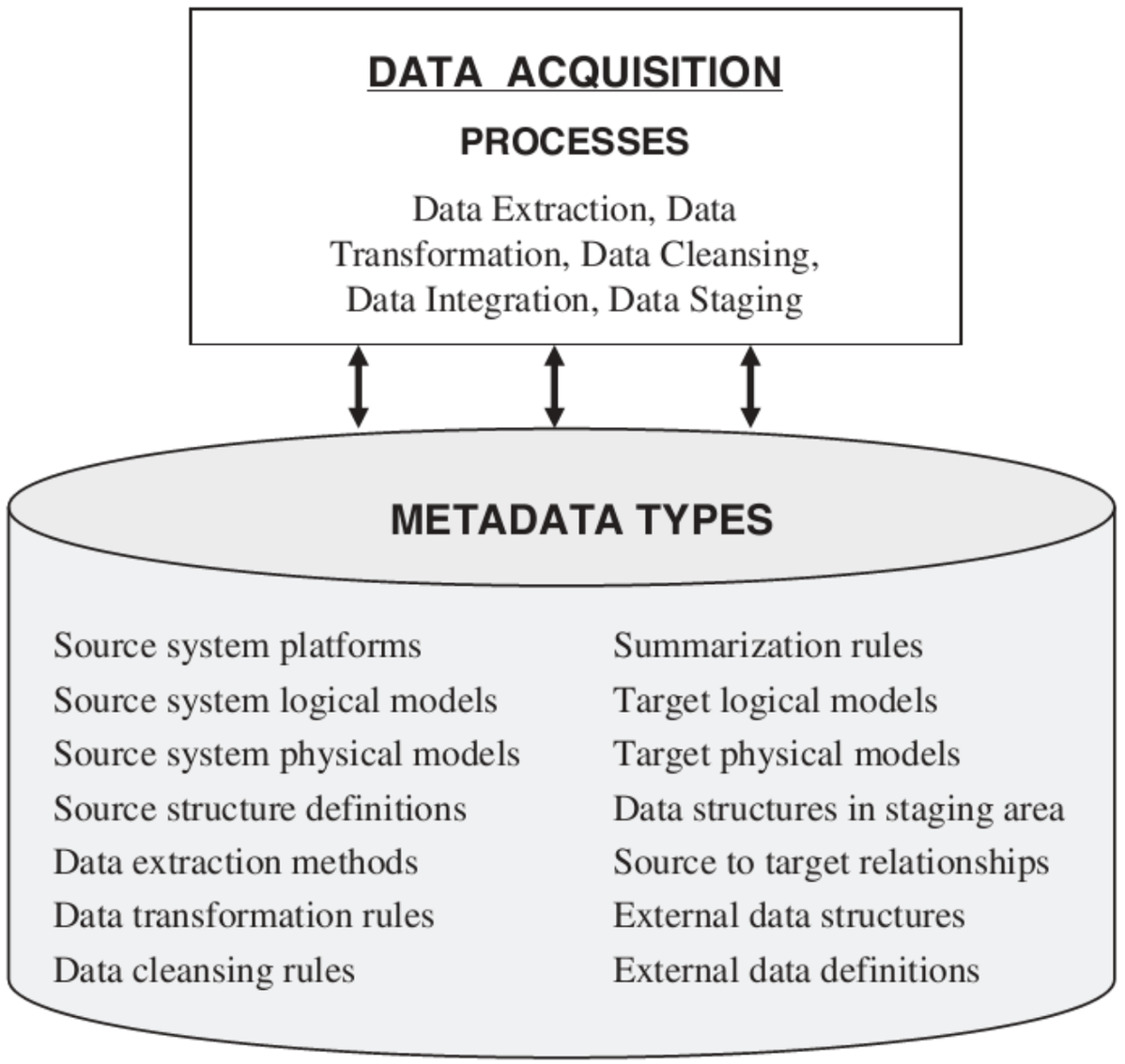
เมตาดาต้าสำหรับฟังก์ชันการได้มาซึ่งข้อมูล	เมตาดาต้าสำหรับฟังก์ชันการจัดเก็บข้อมูล	เมตาดาต้าสำหรับฟังก์ชันการส่งผ่านข้อมูล

เมตาดาต้าสำหรับฟังก์ชันการได้มาซึ่งข้อมูล

ในการเก็บหรือรวบรวมข้อมูลจะมีขั้นตอนที่เกี่ยวกับฟังก์ชันการทำงานดังต่อไปนี้

- การสกัดข้อมูล (Data extraction)
- การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล (Data transformation)
- การทำความสะอาดข้อมูล (Data cleansing)
- การรวมข้อมูล (Data integration)
- การเก็บข้อมูลไว้ใน staging area (Data staging)

ขั้นตอนการได้มาซึ่งข้อมูลจะทำการเก็บข้อมูลที่เกี่ยวข้องกับการสร้างและการดูแลรักษาคลังข้อมูล โดยจะเป็นเหมือนกับข้อมูลการเฝ้าดูคลังข้อมูลหลังจากเริ่มใช้งานไปแล้ว เราอาจใช้เมตาดาต้าในการเฝ้าดูการทำงานของขั้นตอนการสกัด และการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล เมตาดาต้าที่เก็บในขั้นตอนการได้มาซึ่งข้อมูลจะแสดงในรูปที่ 9-8



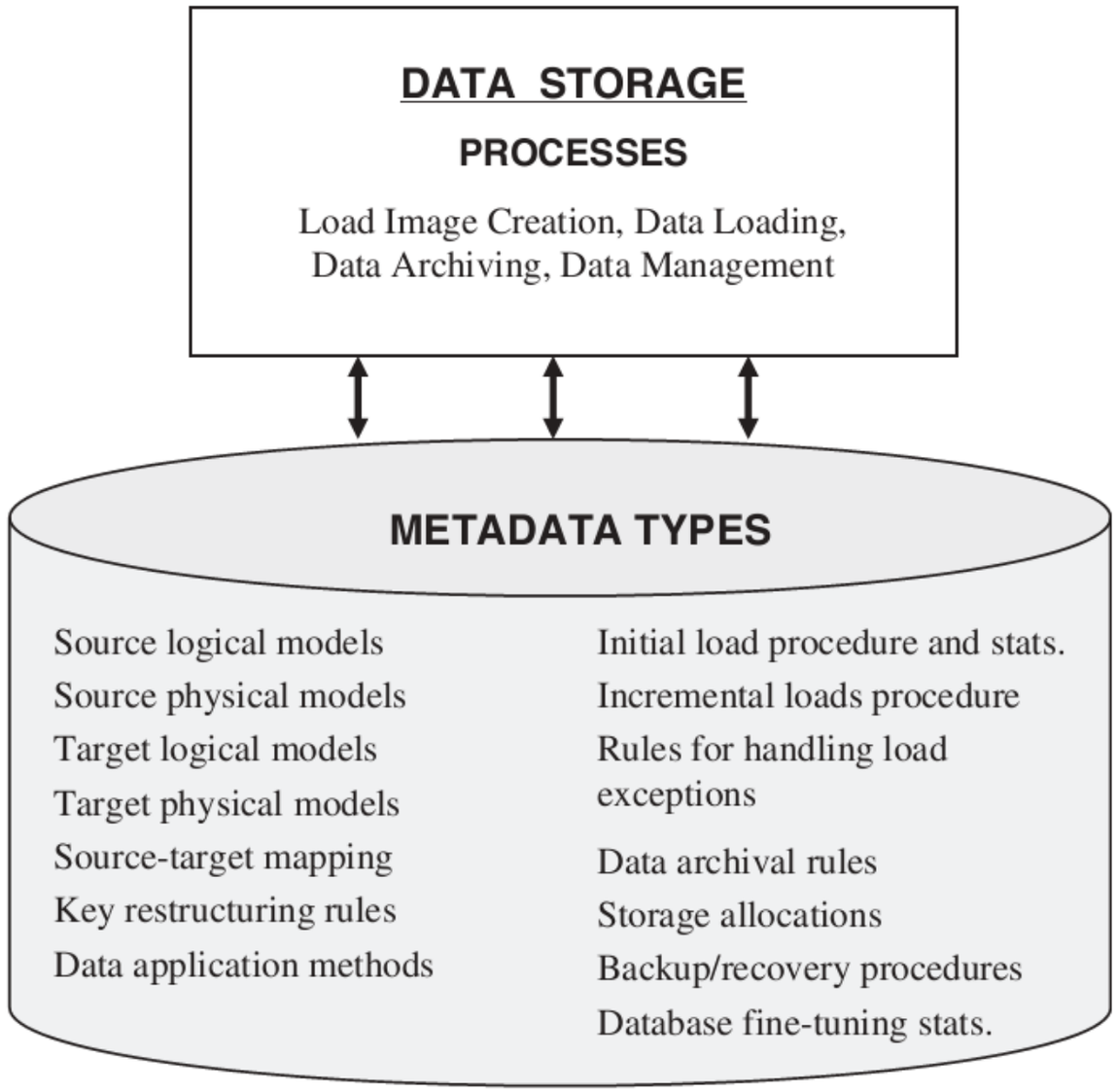
รูปที่ 9-8 เมตาดาต้าสำหรับฟังก์ชันการได้มาซึ่งข้อมูล

เมตาดาต้าสำหรับฟังก์ชันการจัดเก็บข้อมูล

ในการจัดเก็บข้อมูลเข้าสู่คลังข้อมูลจะมีฟังก์ชันการทำงานดังต่อไปนี้

- การถ่ายโอนข้อมูล (Data loading)
- การจัดเก็บข้อมูล (Data archiving)
- การจัดการข้อมูล (Data management)

เมตาดาต้าที่เก็บในขั้นตอนการจัดเก็บข้อมูลโดยส่วนใหญ่แล้วจะเป็นข้อมูลที่เกี่ยวข้องกับผู้สร้าง และ ผู้ดูแลคลังข้อมูล โดยที่เราอาจใช้เมตาดาต้าในการออกแบบเกี่ยวกับการทำ full refresh หรือการทำ incremental data load ซึ่งผู้ดูแลฐานข้อมูลจะใช้เมตาดาต้าทำการ backup recovery และปรับจูนฐานข้อมูล รายละเอียดของเมตาดาต้าที่เก็บในขั้นตอนการจัดเก็บข้อมูลจะแสดงดังรูปที่ 9-9



รูปที่ 9-9 เมตาดาต้าสำหรับฟังก์ชันการจัดเก็บข้อมูล

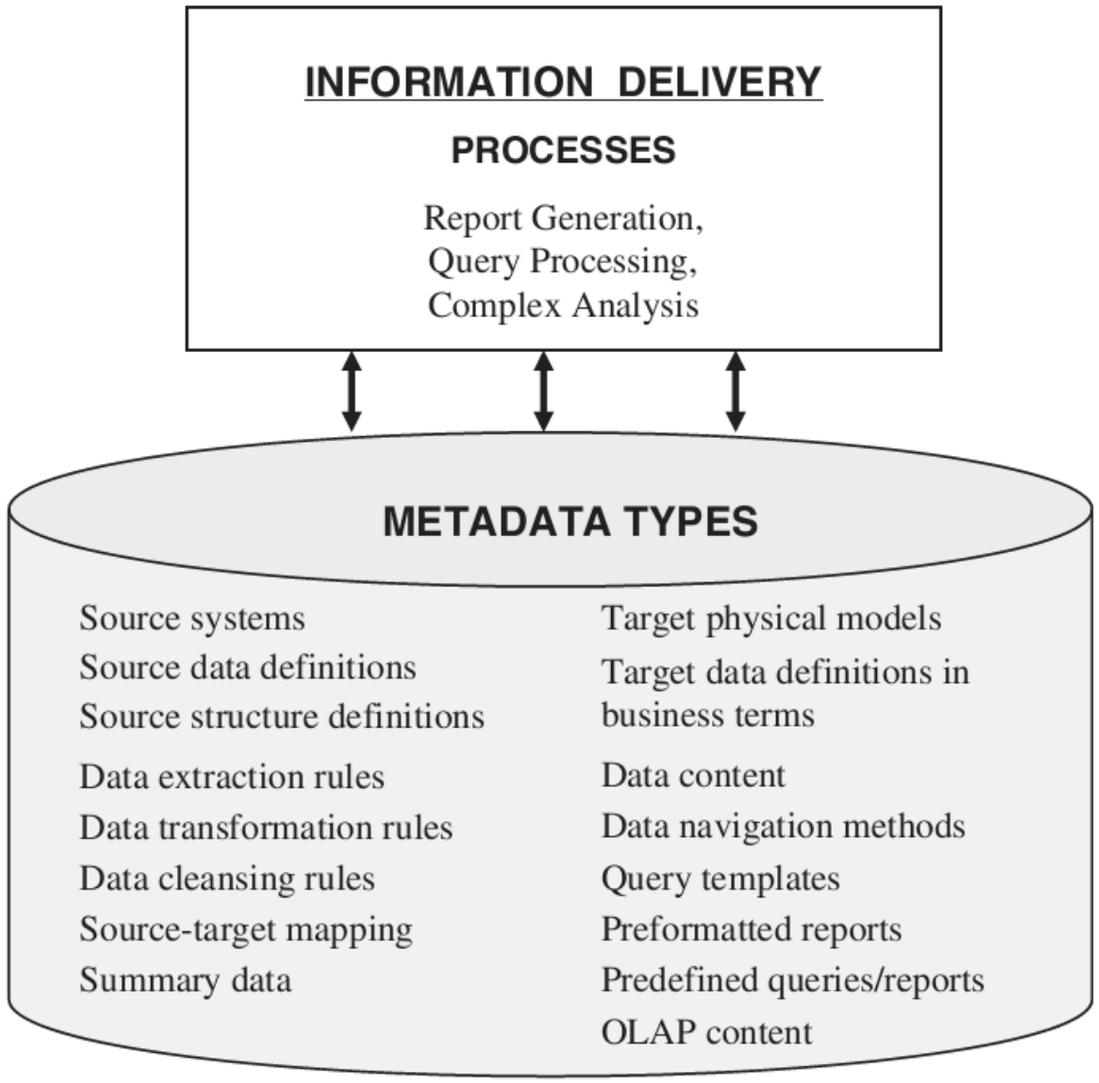
เมตาดาต้าสำหรับฟังก์ชันการส่งผ่านข้อมูล

ในการส่งข้อมูลจากคลังข้อมูลให้กับผู้ใช้จะมีฟังก์ชันการทำงานดังต่อไปนี้

- การสร้างรายงาน (Report generation)
- การประมวลผลคิวรี (Query processing)
- การวิเคราะห์ที่มีความซ้ำซ้อน (Complex analysis)

ขั้นตอนส่วนใหญ่ของการส่งผ่านข้อมูลจะเกี่ยวข้องกับผู้ใช้ โดยที่เราจะใช้เมตาดาต้าสำหรับสำหรับการได้มาซึ่งข้อมูล และการจัดเก็บข้อมูลเข้ามาช่วยในการทำงาน เมื่อผู้ใช้ทำการสร้างคิวรี พวกเขาจะสามารถอ้างย้อนไปถึงเมตาดาต้าที่เก็บไว้จากขั้นตอนการได้มาซึ่งข้อมูล และขั้นตอนการจัดเก็บข้อมูล เพื่อจะได้เรียกดูข้อมูลเกี่ยวกับ การเปลี่ยนแปลงของแหล่งข้อมูล โครงสร้างของข้อมูล และการเปลี่ยนแปลงข้อมูลจากเมตาดาต้าที่เก็บไว้ในขั้นตอนการได้มาซึ่งข้อมูล สำหรับเมตาดาต้าที่เก็บไว้ในขั้นตอนการส่งข้อมูลให้ผู้ใช้จะทำการเก็บข้อมูลเกี่ยวกับการอัพเดทข้อมูลล่าสุดจากตารางต่าง ๆ ในคลังข้อมูล เพื่อให้ผู้ใช้สามารถเรียกดูข้อมูลเหล่านี้ได้

โดยทั่วไปแล้วเมตาดาต้าที่เก็บในขั้นตอนการส่งข้อมูลจะเกี่ยวข้องกับคิวรีและรายงานที่มีการกำหนดไว้แล้ว พารามิเตอร์ต่าง ๆ ที่เกี่ยวข้องกับคิวรีหรือรายงานนั้น ๆ รายละเอียดจะแสดงดังรูปที่ 9-10



รูปที่ 9-10 เมตาดาต้าสำหรับการเข้าถึง/ส่งผ่านข้อมูลไปยังผู้ใช้

SECTION 5

เมตาดาต้าเชิงธุรกิจ

เมตาดาต้าเชิงธุรกิจ

เมตาดาต้าเชิงธุรกิจจะสามารถเชื่อมโยงผู้ใช้ทางธุรกิจเข้ากับคลังข้อมูลได้ โดยผู้ใช้ต้องการที่จะทราบว่าในคลังข้อมูลมีอะไรให้ใช้บ้าง ในมุมที่แตกต่างจากผู้สร้างคลังข้อมูล เมตาดาต้าจะเป็นเหมือนแผนที่ที่แสดงรายละเอียดต่าง ๆ ของคลังข้อมูลซึ่งจะช่วยให้ผู้จัดการและผู้วิเคราะห์เชิงธุรกิจสามารถใช้งานคลังข้อมูลได้

เมตาดาต้าเชิงธุรกิจจะเป็นสิ่งที่ใช้บอกถึงรายละเอียดของข้อมูล เช่น ชื่อตาราง ซึ่งการบอกถึงรายละเอียดจะต้องไม่มีความกำกวม เช่น cal_pr_sle ซึ่งทำให้ผู้ใช้งานไม่สามารถเข้าใจได้ เราจึงจำเป็นต้องทำการเปลี่ยนชื่อตารางใหม่เป็น calculated-prior-month-sale เป็นต้น

เมตาดาต้าเชิงธุรกิจจะไม่มีโครงสร้างที่ซับซ้อนเหมือนกับเมตาดาต้าเชิงเทคนิค ซึ่งจะหาได้จากเอกสารทั่ว ๆ ไป เช่น spreadsheets และอื่น ๆ ที่ซึ่งจะบอกถึงข้อมูลต่างๆ เช่น การเป็นเจ้าของของข้อมูล กฎที่ใช้สำหรับธุรกิจ ขั้นตอนหรือกระบวนการทางธุรกิจ และอื่น ๆ

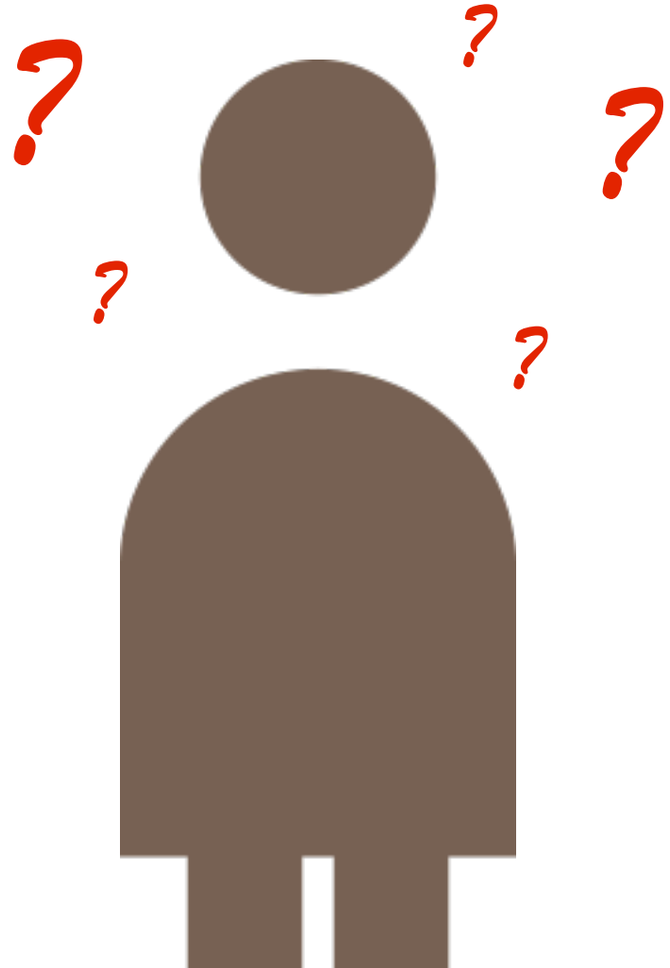
ตัวอย่างของเมตาดาต้าเชิงธุรกิจ

- ขั้นตอนการเชื่อมต่อ (Connectivity procedures)
- สิทธิ์ในการรักษาความปลอดภัยและการเข้าถึง (Security and access privileges)
- โครงสร้างโดยรวมของข้อมูลเชิงธุรกิจ (The overall structure of data in business term)
- ระบบที่เป็นแหล่งข้อมูล (Source systems)
- การเชื่อมโยงระหว่างแหล่งข้อมูลและเป้าหมาย (Source-to-target mapping)
- การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลที่สอดคล้องกับกฎทางธุรกิจ (Data transformation business rules)
- การทำสรุปและได้รับข้อมูล (Summarization and derivations)
- ชื่อตารางและความหมายทางธุรกิจ (Table names and business definition)
- ชื่อคุณสมบัติและความหมายทางธุรกิจ (Attribute names and business definition)
- ความเป็นเจ้าของข้อมูล (Data ownership)
- เครื่องมือในการสืบค้นข้อมูลและการทำรายงาน (Query and reporting tools)

ตัวอย่างของเมตาดาต้าเชิงธุรกิจ (ต่อ)

- การกำหนดรายงานไว้ล่วงหน้า (Predefined reports)
- ข้อมูลการกระจายรายงาน (Report distribution information)
- เส้นทางการเข้าถึงข้อมูลโดยทั่วไป (Common information access routes)
- กฎระเบียบสำหรับการวิเคราะห์ข้อมูลโดยใช้ OLAP (Rules for analysis using OLAP)
- การหมุนเวียนของข้อมูลจาก OLAP (Currency of OLAP data)
- ตารางการอัพเดตคลังข้อมูล (Data warehouse refresh schedule)

จากตัวอย่างของเมตาดาต้าเชิงธุรกิจ
มีคำถามชนิดใดบ้างที่เมตาดาต้าเชิงธุรกิจสามารถตอบได้?
หรือ
ข้อมูลชนิดใดที่ผู้ใช้จะได้รับจากเมตาดาต้าเชิงธุรกิจ?



ดังนั้นเพื่อที่จะตอบคำถามข้างต้นเรา
จะแสดงตัวอย่างของคำถามที่เมตา
ดาต้าเชิงธุรกิจสามารถตอบได้
ดังต่อไปนี้

ฉันจะสามารถลงชื่อเข้าใช้และเชื่อมต่อกับคลังข้อมูลได้อย่างไร?

ส่วนใดของคลังข้อมูลที่ฉันจะสามารถเข้าถึงได้บ้าง?

ฉันสามารถเรียกดูแอทริบิวทั้งหมดจากตารางใดตารางหนึ่งที่ต้องการได้หรือไม่?

นิยามของแอทริบิวใดบ้างที่ฉันต้องการสำหรับการประมวลผลคิวรี?

สำหรับข้อมูลที่ฉันต้องการ มีการกำหนดคิวรีหรือรายงานไว้ก่อนหน้าหรือไม่?

ข้อมูลที่ฉันต้องการมาจากแหล่งข้อมูลใด?

อะไรคือค่าโดยปริยาย (default values) ที่จะถูกใช้เมื่อมีการสืบค้นข้อมูล?

การรวมข้อมูลประเภทใดบ้างที่สามารถใช้ได้?

ค่าของข้อมูลที่ต้องการซึ่งมาจากข้อมูลอื่นๆจะเป็นอย่างไร?

การอัพเดทของข้อมูลที่ฉันต้องการครั้งสุดท้ายทำเมื่อใด?

ข้อมูลใดบ้างที่ฉันสามารถใช้ drill-down ในการวิเคราะห์ข้อมูลได้?

ข้อมูลสำหรับ OLAP มีอายุเท่าไร? ฉันควรรอการอัพเดทข้อมูลครั้งถัดไปไหม?

เมตาดาต้าเชิงธุรกิจจะมีประโยชน์ต่อผู้ใช้อย่างแน่นอน แต่ยังมีบุคคลอื่น ๆ ที่ยังได้รับประโยชน์จากการเก็บข้อมูลเมตาดาต้าอีกเป็นจำนวนมากดังนี้

นักวิเคราะห์ทางธุรกิจ
(Business analysis)

ผู้ใช้ที่มีอำนาจ
(Power users)

ผู้ใช้ทั่ว ๆ ไป
(Regular users)

ผู้จัดการ
(Managers)

ผู้ใช้ที่ใช้ระบบเพียงบางครั้งบางคราว
(Casual users)

ผู้จัดการอาวุโส/ผู้บริหารที่มีอายุน้อย
(Senior managers/junior executives)

SECTION 6

เมตาดาต้าเชิงเทคนิค

เมตาดาต้าเชิงเทคนิค

เมตาดาต้าเชิงเทคนิคเปรียบเสมือนคู่มือในการสร้าง จัดการและดูแลรักษาคลังข้อมูล ซึ่งจะเป็นข้อมูลเกี่ยวกับรายละเอียดของข้อมูลในบางส่วน และรวมถึงการทำงานในขั้นตอนต่าง ๆ ของคลังข้อมูล ว่ามีการทำงานอย่างไร กฎในการทำงานเป็นอย่างไร ใช้เครื่องมืออะไรในการทำงานแต่ละขั้นตอน ซึ่งข้อมูลเหล่านี้เป็นสิ่งจำเป็นมากในการบริหารจัดการคลังข้อมูล ซึ่งถ้าเราไม่มีการเก็บเมตาดาต้าเชิงเทคนิคจะทำให้เราดูแลรักษาให้ข้อมูลในคลังข้อมูลมีคุณภาพและอัพเดทเป็นไปได้ยาก

ผู้พัฒนาคลังข้อมูลจะต้องการเมตาดาต้าเชิงเทคนิคเพื่อช่วยเหลือในการทำความเข้าใจกับการได้มาซึ่งข้อมูล การสกัดข้อมูล การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล และขั้นตอนการทำความสะอาดข้อมูล

ดังนั้นเราจะต้องเข้าใจถึงกฎของการทำงานต่าง ๆ รวมถึงเครื่องมือที่ใช้ด้วย ว่าใช้เครื่องมืออะไรมีข้อจำกัดอย่างไร

เมตาดาต้าเชิงเทคนิคจะถูกใช้ในการทำงาน 3 ประเภทคือ

1

สำหรับสร้างฟังก์ชันการทำงานต่าง ๆ ของคลังข้อมูล เช่น ถ้าเราต้องทำการสร้างฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล เราจะต้องใช้เมตาดาต้าสำหรับการสกัดข้อมูล เพื่อให้ทราบถึงรายละเอียดและปริมาณของข้อมูลที่ทำการสกัดจากแหล่งข้อมูล

2

สำหรับทำการอัพเดทข้อมูล หรือ ดูแลจัดการสิ่งต่าง ๆ ถ้าเราต้องทำการหาความแตกต่างของข้อมูล เราจะต้องการเมตาดาต้าสำหรับเรียกดูรายละเอียดของข้อมูลว่ามีความเปลี่ยนแปลงหรือไม่

3

สำหรับการดูและรักษาคลังข้อมูล เราจะต้องเฝ้าดูการสกัดข้อมูล เราต้องให้แน่ใจว่าการทำ incremental load นั้นมีความถูกต้องสมบูรณ์ ซึ่งเรารวมถึงอาจต้องทำการสำรองข้อมูลเก่าๆอีกด้วย ซึ่งเราจะไม่สามารถทำการดูแลและรักษาคลังข้อมูลได้ถ้าเราไม่มีการจัดเก็บเมตาดาต้าที่ดี

ตัวอย่างของเมตาดาต้าเชิงเทคนิค

เมตาดาต้าเชิงเทคนิคจะถูกใช้ในการสร้าง และดูแลรักษาคลังข้อมูล โดยเมตาดาต้าเชิงเทคนิคจะมีโครงสร้างที่ซับซ้อนมากกว่ามาตาดาต้าเชิงธุรกิจ เมตาดาต้าจะมีลักษณะดังต่อไปนี้

- โมเดลข้อมูลสำหรับแหล่งข้อมูล
(Data models of source systems)
- ข้อมูล Layout ของแหล่งข้อมูลภายนอก
(Record layouts of outside sources)
- การเชื่อมโยงระหว่างแหล่งข้อมูลกับ staging area
(Source to staing area mapping)
- กฎและกำหนดการสำหรับการสกัดข้อมูล
(Data extraction rules and schedules)
- กฎสำหรับการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล
(Data transformation rules)
- กฎการรวมข้อมูล
(Data aggregation rules)
- กฎการทำความสะอาดข้อมูล
(Data cleansing rules)
- การสรุปและการนำข้อมูลไปใช้
(Summarization and derivations)
- การถ่ายโอนข้อมูล และกำหนดการสำหรับอัพเดทข้อมูล
(Data loading and refresh schedules)
- การขึ้นแก่กันของการทำงาน
(Job dependencies)

ชื่อโปรแกรมและคำอธิบายต่าง ๆ

- โมเดลข้อมูลสำหรับคลังข้อมูล (Data warehouse data model)
- ชื่อฐานข้อมูล (Database names)
- ชื่อตาราง (Table names)
- ชื่อคอลัมน์ และคำอธิบายคอลัมน์ (Column names and descriptions)
- แอทริบิวที่เป็นคีย์ (Key attributes)
- กฎทางธุรกิจสำหรับเอนทิตีและความสัมพันธ์ระหว่างเอนติที (Business rules for entities and relationships)
- การเชื่อมโยงระหว่างโมเดลทางตรรกะและโมเดลทางกายภาพ (Mapping between logical and physical models)

ชื่อโปรแกรมและคำอธิบายต่าง ๆ (ต่อ)

- ข้อมูลเกี่ยวกับเครือข่ายและเซริฟเวอร์ (Network/server information)
- ข้อมูลที่มีการเชื่อมโยง (Connectivity data)
- การควบคุมการตรวจสอบการเคลื่อนที่ของข้อมูล (Data movement audit controls)
- สิทธิ์ในการเข้าถึงข้อมูล (Authority/access privileges)
- การใช้ข้อมูลและเวลาที่ใช้ข้อมูล (Data usage/timinig)
- รูปแบบของคิวรีและรายงาน (Query and report access patterns)
- เครื่องมือสำหรับคิวรีและรายงาน (Query and report tools)

รายการข้างล่างนี้จะสามารถให้แนวคิดเกี่ยวกับข้อมูลที่เป็นเมตาดาต้าเชิงเทคนิคที่คลังข้อมูลควรจะเก็บได้ ซึ่งข้อมูลที่เป็นเมตาดาต้าเชิงเทคนิคจะใช้ในการตอบคำถามสำหรับผู้พัฒนาและผู้ดูแลคลังข้อมูล ซึ่งมีลักษณะคำถามดังต่อไปนี้

- ฐานข้อมูลและตารางที่มีอยู่เป็นอย่างไร?
- คอลัมน์ของแต่ละตารางเป็นอย่างไร?
- คีย์และดัชนีเป็นอย่างไร?
- คำอธิบายทางธุรกิจสอดคล้องกับเทคนิคการทำงานอย่างใดอย่างหนึ่งหรือไม่?
- การอัพเดทข้อมูลครั้งสุดท้ายทำเมื่อใด?
- แหล่งข้อมูลและโครงสร้างของแหล่งข้อมูลเป็นอย่างไร?
- กฎในการสกัดข้อมูลจากแต่ละแหล่งข้อมูลเป็นอย่างไร?
- การเชื่อมโยงระหว่างข้อมูลในแหล่งข้อมูลไปยังข้อมูลเป้าหมายในคลังข้อมูลเป็นอย่างไร?
- กฎสำหรับการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลเป็นอย่างไร?
- ค่า default ที่ใช้ในการเติมข้อมูลหนึ่งๆ ที่มีการขาดหายไปของข้อมูลเป็นอย่างไร?
- ประเภทของการรวมข้อมูลเป็นอย่างไร?
- ตารางเวลาสำหรับการถ่ายโอนและการอัพเดทข้อมูลเป็นอย่างไร?
- ตารางเวลาสำหรับการสร้างข้อมูลสำหรับ OLAP เป็นอย่างไร?
- เครื่องมือสำหรับคิวรีและรายงานเป็นอย่างไร

**คนที่จะได้รับประโยชน์จาก
เมตาดาต้าเชิงเทคนิคมีดังต่อไปนี้**

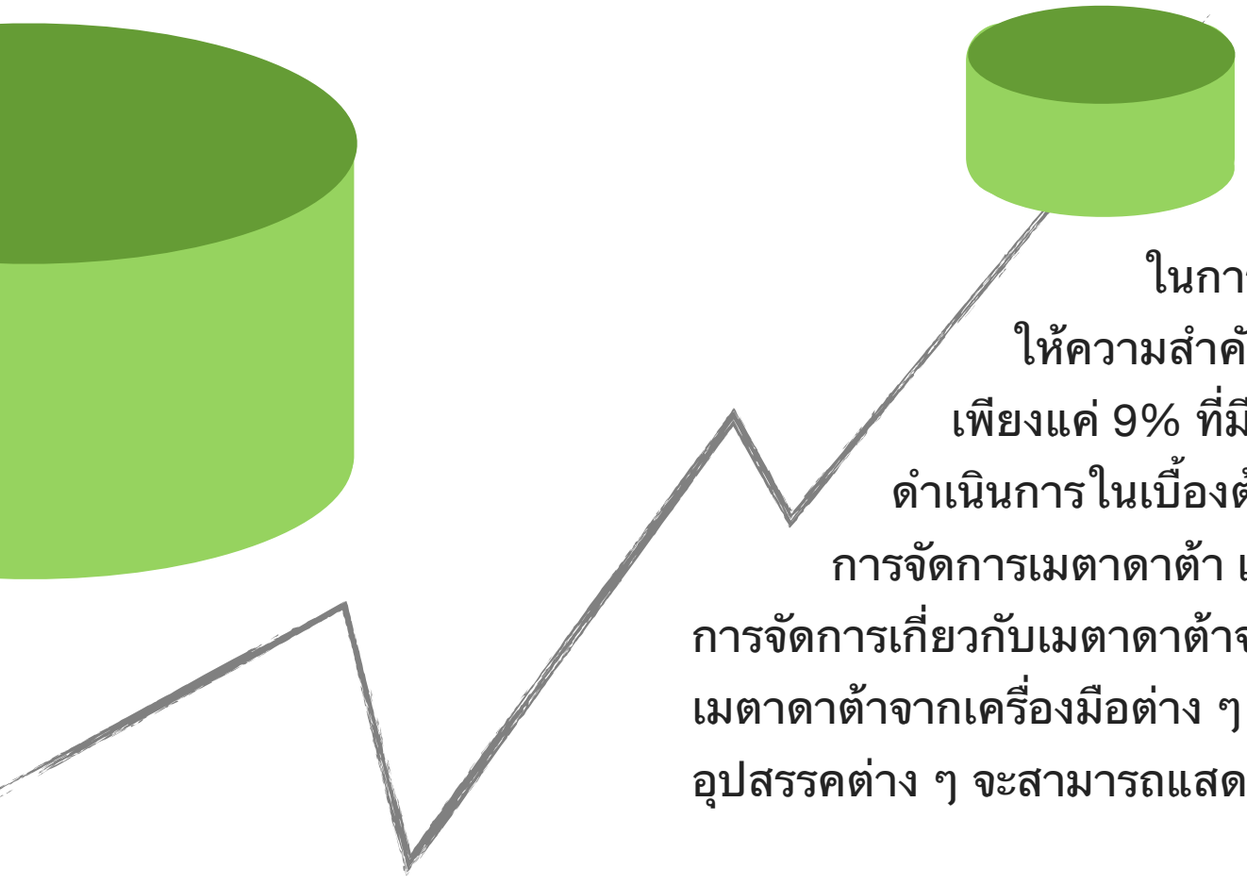
- ผู้จัดการโครงการ (Project manager)
- ผู้ดูคลังข้อมูล (Data warehouse administrator)
- ผู้ดูแลฐานข้อมูล (Database administrator)
- ผู้จัดการเกี่ยวกับเมตาดาต้า (Metadata manager)
- ผู้สร้างคลังข้อมูล (Data warehouse architect)
- ผู้พัฒนาการเก็บรวบรวมข้อมูล (Data acquisition developer)
- ผู้วิเคราะห์คุณภาพของข้อมูล (Data quality analyst)
- ผู้วิเคราะห์เชิงธุรกิจ (Business analyst)
- ผู้ดูแลระบบ (System administrator)
- ผู้เชี่ยวชาญด้านโครงสร้างพื้นฐาน (Infrastructure specialist)
- ผู้สร้างโมเดลข้อมูล (Data modeler)
- ผู้สร้างระบบรักษาความปลอดภัย (Security architect)

SECTION 7

ขั้นตอนการจัดเก็บเมตาดาต้า

ขั้นตอนการจัดเก็บเมตาดาต้า

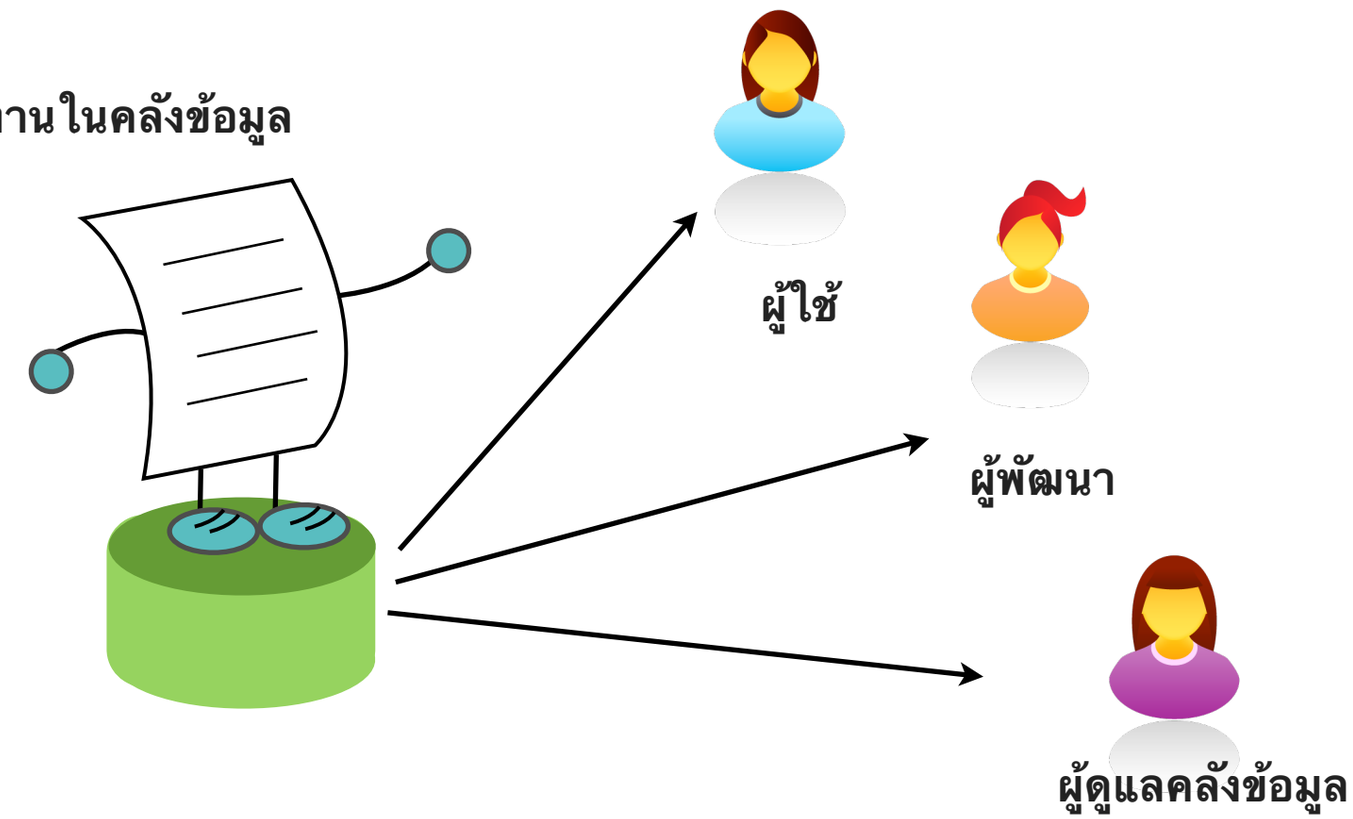
อย่างที่เราทราบกันดีว่า เมตาดาต้าจำเป็นต้องถูกเก็บรวบรวมและถูกเก็บไว้ โดยที่เมตาดาต้าจะสามารถอธิบายถึงคลังข้อมูลในมุมมองต่าง ๆ ซึ่งเราสามารถค้นหาแหล่งข้อมูลผ่านเมตาดาต้า สามารถเข้าใจในเครื่องมือที่ใช้สำหรับการสกัดข้อมูล และเครื่องมือสำหรับการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล สามารถกำหนดวิธีการในค้นหาข้อมูลได้



ในการศึกษาเกี่ยวกับการสร้างคลังข้อมูลในหลาย ๆ ปีก่อน มี 86% ที่ให้ความสำคัญกับการมีกลยุทธ์ที่จะจัดการกับเมตาดาต้า แต่อย่างไรก็ตามมีเพียงแค่ 9% ที่มีการสร้างการจัดการเมตาดาต้า ที่เหลืออีก 16% นั้นมีแค่แผนและดำเนินการในเบื้องต้นเท่านั้น แต่ในบริษัทส่วนใหญ่ก็มีการตระหนักถึงความสำคัญของการจัดการเมตาดาต้า แต่ก็มีเพียงส่วนน้อยเท่านั้นที่ดำเนินการกับการจัดการนั้น ๆ ซึ่งในการจัดการเกี่ยวกับเมตาดาต้าจะมีอุปสรรคอยู่พอสมควร ซึ่งอุปสรรคจะไม่ได้เกิดจากการจัดเก็บเมตาดาต้าจากเครื่องมือต่าง ๆ แต่อุปสรรคจะอยู่ที่การรวบรวมเมตาดาต้าด้วยเครื่องมือต่าง ๆ ซึ่งอุปสรรคต่าง ๆ จะสามารถแสดงได้ดังต่อไปนี้

ความต้องการเมตาดาต้า

เมตาดาต้าจะต้องทำหน้าที่เป็นแผนงานในคลังข้อมูลสำหรับผู้ใช้ ผู้พัฒนา และผู้ดูแลคลังข้อมูล ซึ่งในการจัดเก็บเมตาดาต้าจะต้องมีคุณสมบัติหรือตอบสนองความต้องการดังต่อไปนี้



การจัดเก็บเมตาดาต้าจะต้องมีคุณสมบัติ
หรือตอบสนองความต้องการดังต่อไปนี้

- ***Capturing and Storing data***
- ***Variety of Metadata Sources***
- ***Metadata Integration***
- ***Metadata Standardization***
- ***Rippling Through of Revisions***
- ***Keeping Metadata Synchronized***
- ***Metadata Exchange***
- ***Support for End-Users***

Capturing and Storing data

โดยทั่วไปในระบบการดำเนินงานจะต้องมีการเก็บพจนานุกรมของข้อมูล (Data dictionary) ซึ่งจะทำการเก็บโครงสร้างข้อมูลและกฎทางธุรกิจ ณ ช่วงเวลาปัจจุบัน โดยที่ไม่สนใจเกี่ยวกับการเก็บประวัติของพจนานุกรมข้อมูล แต่อย่างไรก็ตามประวัติของข้อมูลมีความสำคัญต่อคลังข้อมูล ซึ่งโดยทั่วไปแล้วจะเก็บประวัติย้อนหลังของข้อมูลอยู่ในช่วงระหว่าง 5 ถึง 15 ปี ระหว่างช่วงเวลาที่กำหนดอาจมีการเปลี่ยนแปลงเกิดขึ้นจาก ข้อมูลในแหล่งข้อมูล วิธีในการสกัดข้อมูล อัลกอริทึมสำหรับแปลงข้อมูล และตัวโครงสร้าง และเนื้อหาของคลังข้อมูล เมตาดาต้าในคลังข้อมูลจะต้องติดตามการเปลี่ยนแปลงของเมตาดาต้านั้น ดังนั้นการจัดการเมตาดาต้าจะต้องอำนวยความสะดวกในการสกัดและจัดเก็บการเปลี่ยนแปลงที่ตอบสนองความเปลี่ยนแปลงต่อการเวลา

Variety of Metadata Sources

เมตาดาต้าสำหรับคลังข้อมูลจะมาจากหลายแหล่งข้อมูล หลายระบบดำเนินการ เครื่องมือต่าง ๆ สำหรับการสกัดข้อมูลและการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล นิยามต่าง ๆ สำหรับดาต้าดิกชันนารีและแหล่งข้อมูลอื่น ๆ ที่มีข้อมูลสำหรับเมตาดาต้าสำหรับคลังข้อมูล ดังนั้นในการจัดการเกี่ยวกับเมตาดาต้าเราควรจะต้องเปิดกว้างเพื่อรองรับการได้มาซึ่งเมตาดาต้าจากหลาย ๆ แหล่งข้อมูล

Metadata Integration

เมื่อเราทำการเก็บเมตาดาต้าเชิงธุรกิจและเชิงเทคนิค การจัดเก็บอาจจะต้องทำการผสมผสานและรวมเมตาดาต้าเข้าด้วยกัน โดยที่การรวบรวมควรจะทำให้มีความหมายต่อผู้ใช้งานคลังข้อมูล และเมตาดาต้าจากแหล่งข้อมูลจะต้องสามารถผสมผสานเข้ากับเมตาดาต้าที่ถูกเก็บในคลังข้อมูลได้อีกด้วย

Metadata Standardization

ถ้าเครื่องมือสำหรับสกัดและเปลี่ยนแปลงข้อมูลอธิบายถึงโครงสร้างของข้อมูลแล้ว ทั้งสองเครื่องมือจะต้องเก็บเมตาดาต้าที่เกี่ยวข้องกับโครงสร้างของข้อมูลให้เป็นมาตราฐานเดียวกัน ซึ่งเมตาดาต้าที่เหมือนกันที่มาจากแหล่งข้อมูลที่แตกต่างกันซึ่งมาจากเครื่องมือต่างๆ จะต้องถูกอธิบายไปในทิศทางเดียวกันด้วย

Rippling Through of Revisions

การแก้ไขหรือปรับปรุงอาจเกิดขึ้นสำหรับเมตาดาต้า ซึ่งอาจเกิดจากข้อมูลหรือกฎทางธุรกิจมีการเปลี่ยนแปลง ซึ่งการแก้ไขเมตาดาต้าจะถูกเฝ้าติดตามในกระบวนการหนึ่งของคลังข้อมูล การแก้ไขหรือปรับปรุงจะมีการทำงานเป็นระลอก ๆ ไปทั่วคลังข้อมูลซึ่งจะส่งผลถึงขั้นตอนอื่น ๆ ด้วย

Keeping Metadata Synchronized

เมตาดาต้าที่เกี่ยวกับโครงสร้างข้อมูล องค์ประกอบของข้อมูล เหตุการณ์ กฎ และอื่น ๆ จะต้องถูกทำให้สอดคล้องกันตลอดทั้งคลังข้อมูล

Metadata Exchange

เมื่อผู้ใช้มีการใช้งานเครื่องมือสำหรับการเข้าถึงข้อมูล เครื่องมือเหล่านั้นจะต้องสามารถเรียกดูเมตาดาต้าที่ถูกเก็บไว้โดยเครื่องมืออื่น ๆ (เช่น เครื่องมือสำหรับการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล) ซึ่งเราควรจะทำให้การแลกเปลี่ยนเมตาดาต้าระหว่างเครื่องมือหนึ่ง ๆ กับเครื่องมืออื่น ๆ สามารถทำได้โดยง่าย และไม่เปลืองประสิทธิภาพมากนัก

Support for End-Users

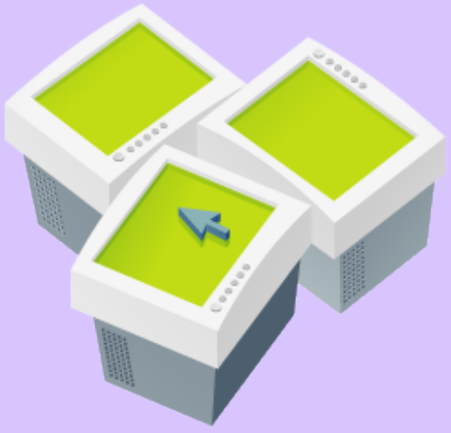
การจัดการเมตาดาต้าจะทำต้องจัดเตรียมกราฟฟิกและตารางอย่างง่าย ๆ ให้ผู้ใช้งาน และจะต้องทำให้การเรียกดูและการทำความเข้าใจข้อมูลในคลังข้อมูลจากมุมมองทางธุรกิจสามารถทำได้โดยง่าย

เมื่อเราใช้เครื่องมือในขั้นตอนการทำงานต่าง ๆ ของคลังข้อมูล ตัวอย่างเช่น แหล่งข้อมูล เครื่องมือสำหรับสกัดข้อมูล เครื่องมือสำหรับเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลและทำความสะอาดข้อมูล เครื่องมือสำหรับถ่ายโอนข้อมูล เครื่องมือสำหรับเก็บข้อมูล และเครื่องมือสำหรับส่งข้อมูล เป็นต้น ซึ่งเครื่องมือแต่ละชนิดจะทำการเก็บเมตาดาต้าไว้ โดยจะทำการเก็บข้อมูลที่เกี่ยวข้องกับการทำงานนั้น ๆ ดังนี้

แหล่งข้อมูล

- คู่มือหรือเครื่องมือสำหรับโมเดลของข้อมูลสำหรับระบบการดำเนินงาน
  (Data models of operational systems: manual or with Case tools)
- คำนิยามขององค์ประกอบของข้อมูลจากเอกสารของระบบ
  (Definitions of data element from system documentation)
- ข้อกำหนดสำหรับการทำซ้ำและการควบคุมการทำงานต่าง ๆ
  (COBOL copybooks and control block specification)
- รูปแบบไฟล์ทางกายภาพและคำจำกัดความของเขตข้อมูล
  (Physical file layouts and field definitions)
- รายละเอียดของโปรแกรม
  (Program specifications)
- เค้าโครงแฟ้มข้อมูลและคำจำกัดความสำหรับเขตข้อมูลจากแหล่งข้อมูลภายนอก
  (File layouts and field definitions for data from outside sources)
- แหล่งข้อมูลอื่น ๆ เช่น เอกสารสเปรดชีท และคู่มือต่าง ๆ
  (Other sources such as spreadsheets and manual lists)

การสกัดข้อมูล

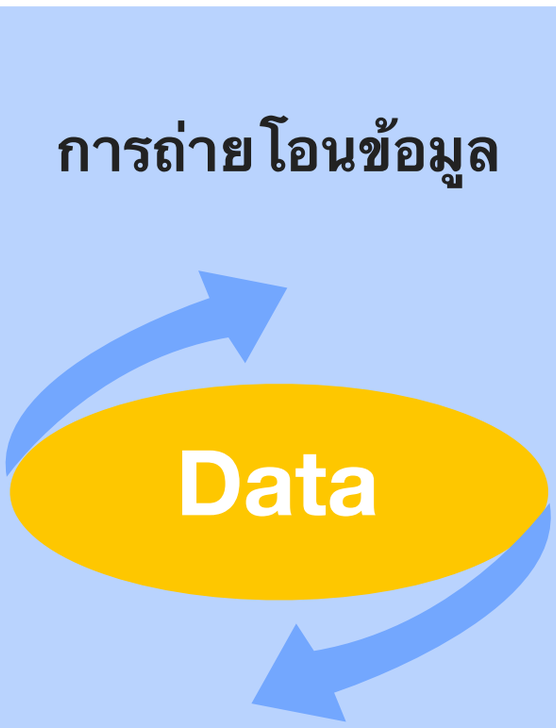


- ข้อมูลจากแพลตฟอร์มต่าง ๆ จากแหล่งข้อมูลและการเชื่อมต่อ
  (Data on source platforms and connectivity)
- เค้าโครงและคำจำกัดความของแหล่งข้อมูล
  (Layouts and definitions of selected data sources)
- ความหมายของฟิลด์ต่าง ๆ ที่ถูกเลือกเพื่อใช้ในการสกัดข้อมูล
  (Definitions of fields selected for extraction)
- เกณฑ์สำหรับการรวมข้อมูลเพื่อให้ได้เป็นไฟล์เริ่มต้นสำหรับแต่ละแพลตฟอร์มของแหล่งข้อมูล (Criteria for merging into initial extract files on each platform)
- หลักเกณฑ์สำหรับการทำฟิลด์ต่าง ๆ ให้เป็นมาตราฐาน
  (Rules for standardizing filed type)
- ตารางเวลาสำหรับการสกัดข้อมูล
  (Data extraction schedules)
- วิธีการสกัดข้อมูลที่มีการเปลี่ยนแปลง
  (Extraction methods for incremental changes)
- ปริมาณงานที่ต้องทำ ในการสกัดข้อมูล
  (Data extraction job streams)

การเปลี่ยนแปลง/เปลี่ยนรูปและการทำความสะอาดข้อมูล

- รายละเอียดของการเชื่อมโยงระหว่างข้อมูลที่ทำการสกัดกับที่อยู่ของข้อมูลที่อยู่ใน staging area (Specifications for mapping extracted files to data staging files)
- กฎการแปลงสำหรับแต่ละข้อมูล
  (Conversion rules for individual files)
- ค่าโดยปริยายสำหรับฟิลด์ที่มีข้อมูลบางส่วนขาดหายไป
  (Default values for fields with missing values)
- กฎเกณฑ์ทางธุรกิจสำหรับการตรวจสอบความถูกต้อง
  (Business rules for validity checking)
- การเรียงและการจัดลำดับข้อมูล
  (Sorting and resequencing arrangements)
- การตรวจสอบเส้นทางการเคลื่อนย้ายข้อมูลที่สกัดได้เข้าสู่ staging area
  (Audit trail for the movement from data extraction to data staging)

การถ่ายโอนข้อมูล



- ข้อกำหนดสำหรับการเชื่อมโยงข้อมูลจาก staging area เข้าสู่คลังข้อมูล
  (Specification for mapping data staging files to load images)
- หลักเกณฑ์ในการกำหนดคีย์สำหรับแต่ละไฟล์
  (Rules for assigning keys for each files)
- การตรวจสอบเส้นทางการเคลื่อนย้ายข้อมูลจาก staging area ไปยังคลังข้อมูล
  (Audit trail for the movement from data staging to load images)
- ตารางเวลาสำหรับการทำ full refreshes
  (Schedules for full refreshes)
- ตารางเวลาสำหรับการทำ incremental loads
  (Schedules for incremental loads)
- ปริมาณงานที่ต้องทำการถ่ายโอนข้อมูล
  (Data loading job stream)

การจัดเก็บข้อมูล

- แบบจำลองข้อมูลสำหรับคลังข้อมูลส่วนกลางและสำหรับคลังข้อมูลที่มีการแบ่งเป็นส่วนย่อย ๆ (Data models for centralized data warehouse and dependent data marts)
- การจัดกลุ่มของตารางต่างๆที่เกี่ยวกับหัวข้อต่าง ๆ (Subject area groupings of tables)
- แบบจำลองข้อมูลสำหรับส่วนของข้อมูลที่มีความสอดคล้องกัน (Data models for conformed data marts)
- ไฟล์ทางกายภาพ (Physical files)
- คำจำกัดความของตารางและคอลัมน์ (Table and column definitions)
- กฎเกณฑ์ทางธุรกิจสำหรับการตรวจสอบความถูกต้อง (Business rules for validity checking)

การเข้าถึง/
ส่งผ่านข้อมูล

- ทำการลิสต์เครื่องมือสำหรับการประมวลผลคิวรีและรายงาน
  (List of query and report tools)
- ทำการลิสต์คิวรีและรายงานต่างๆที่ทำการกำหนดไว้ก่อนหน้าแล้ว
  (List of predefined queries and reports)
- แบบจำลองข้อมูลสำหรับฐานข้อมูลในระบบ OLAP
  (Data model for special databases for OLAP)
- กำหนดตารางเวลาสำหรับการค้นคืนข้อมูลจาก OLAP
  (Schedules for retrieving data for OLAP)

อุปสรรคและความท้าทายในการจัดการเมตา

เมตาดาต้านั้นมีความสำคัญมากกับคลังข้อมูล แต่อย่างไรก็ตาม การผสมผสานส่วนของเมตาดาต้าจากเครื่องมือหรือขั้นตอนการทำงานต่าง ๆ เป็นเรื่องที่ทำได้ค่อนข้างยาก เมตาดาต้าที่สร้างด้วยกระบวนการหนึ่งอาจไม่สามารถเรียกใช้ได้ในอีกเครื่องมือหนึ่งหรืออีกขั้นตอนหนึ่งได้ ดังนั้นเรื่องการจัดการเกี่ยวกับเมตาดาต้าจะเป็นเรื่องที่มีความท้าทายและอาจเป็นอุปสรรคต่อการทำงานของคลังข้อมูลได้ ซึ่งอุปสรรคต่าง ๆ สามารถแสดงได้ดังนี้

- แต่ละเครื่องมือจะมีการเก็บเมตาดาต้าในตัวเอง ถ้าเราใช้เครื่องมือหลายชิ้น เราจะทำให้เมตาดาต้าที่ถูกเก็บโดยหลาย ๆ เครื่องมือมีรูปแบบที่เหมือนกันได้อย่างไร?
- รูปแบบของเมตาดาต้ายังไม่มีมาตราฐานสากล
- มีความขัดแย้งในการอ้างถึงประโยชน์จากการจัดเก็บเมตาดาต้าไว้แบบศูนย์กลางกับเก็บเมตาดาต้าแบบแยกส่วน
- ยังไม่มีวิธีที่ง่ายและเป็นที่ยอมรับสำหรับการส่งต่อเมตาดาต้าระหว่างกระบวนการทำงาน เช่น การส่งต่อเมตาดาต้าจากแหล่งข้อมูลไปยังที่พักข้อมูล แล้วส่งต่อไปยังที่สำหรับจัดเก็บข้อมูลในคลังข้อมูล
- การรักษาเวอร์ชั่นของเมตาดาต้าอย่างสม่ำเสมอนั้นเป็นเรื่องที่ยาก
- ในระบบคลังข้อมูลที่มีขนาดใหญ่ซึ่งได้รับข้อมูลมาจากหลายแหล่งข้อมูล การทำให้เมตาดาต้าเป็นหน่วยเดียวกันเป็นเรื่องยาก เนื่องจากเราอาจต้องประสบกับปัญหาความไม่สอดคล้องกันระหว่างมาตราฐาน รูปแบบ การตั้งชื่อข้อมูล นิยามของข้อมูล แอทริบิว ค่าต่าง ๆ กฎทางธุรกิจ และหน่วยของการวัด

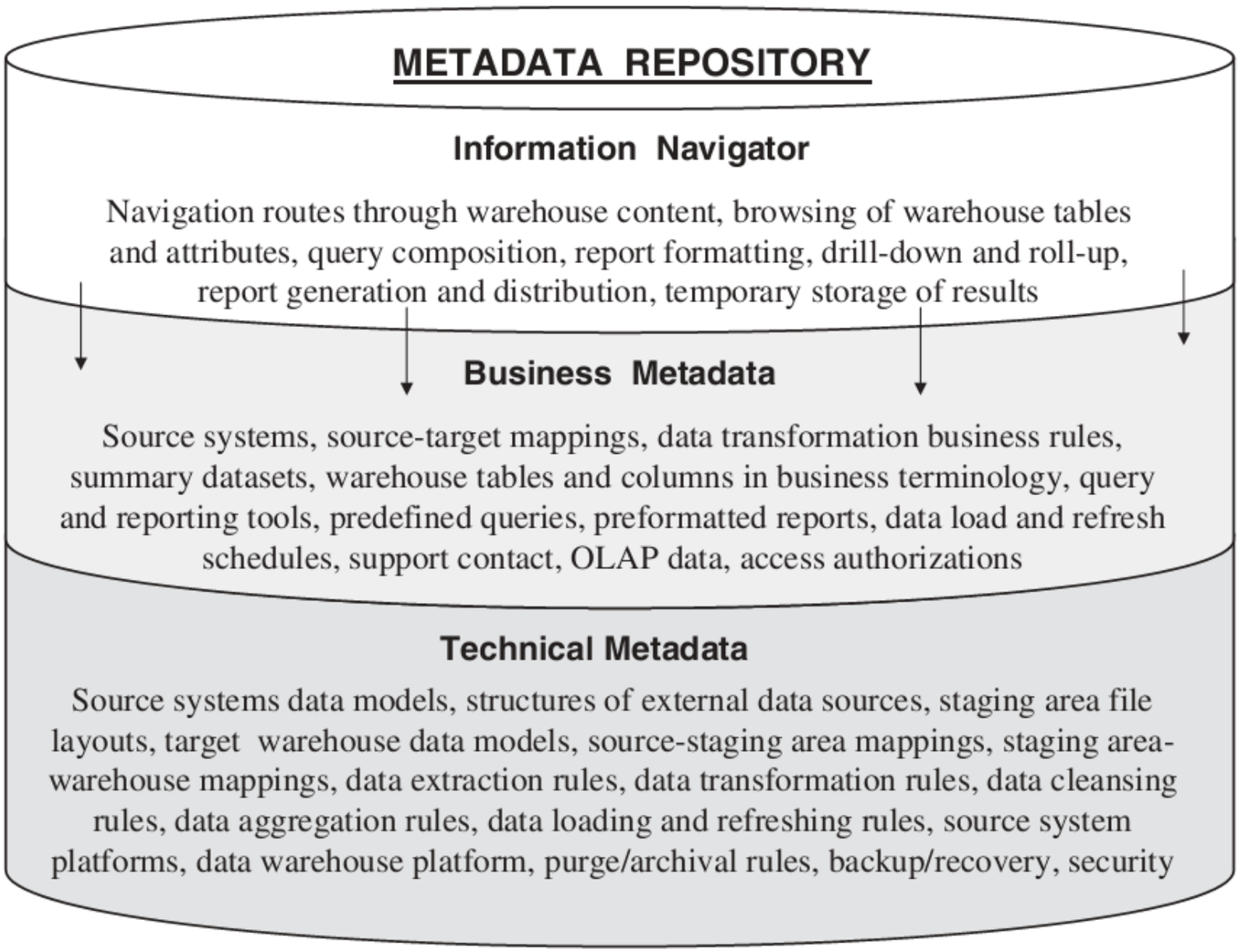
พื้นที่ที่ใช้สำหรับจัดเก็บเมตาดาต้า

พื้นที่สำหรับเก็บเมตาดาต้า (Metadata repository) เปรียบเสมือนไดเรกทอรีหรือแคตตาล็อกที่ช่วยในการแบ่งแยกข้อมูล จัดเก็บข้อมูล และจัดการเรื่องต่าง ๆ เกี่ยวกับข้อมูล อย่างที่เราทราบดีว่าเมตาดาต้าจะมีด้วยกัน 2 ประเภทคือ เมตาดาต้าเชิงธุรกิจ และเมตาดาต้าเชิงเทคนิค ดังนั้นพื้นที่สำหรับเก็บเมตาดาต้าอาจแบ่งออกเป็น 2 ไดเรกทอรี ไดเรกทอรีหนึ่งจะใช้เก็บเมตาดาต้าเชิงธุรกิจ และอีกไดเรกทอรีหนึ่งจะใช้เก็บเมตาดาต้าเชิงเทคนิค รูปที่ 9-11 แสดงพื้นที่สำหรับเก็บเมตาดาต้า ซึ่งจะแบ่งส่วนออกเป็นเมตาดาต้าเชิงธุรกิจและเมตาดาต้าเชิงเทคนิค นอกจากนี้ยังมีส่วนของฟังก์ชันการนำทางของข้อมูล (Information navigator) ซึ่งจะประกอบไปด้วยสิ่งต่าง ๆ ดังนี้

***Interface from Query Tools*—**จะเป็นส่วนที่เชื่อมคลังข้อมูลกับเครื่องมืออื่นๆเพื่อที่เครื่องมือเหล่านี้จะสามารถเรียกดูนิยามของเมตาดาต้าได้

***Drill Down for Details*—**ผู้ใช้เมตาดาต้าจะสามารถดูรายละเอียดแบบเจาะลึกได้ โดยทำการดูข้อมูลที่มีความละเอียดน้อยไปยังข้อมูลที่มีความละเอียดมาก เช่น เริ่มแรกผู้ใช้อาจทำการเรียกดูนิยามของตาราง จากนั้นเจาะลึกลงไปที่นิยามของทุก ๆ attribute และสามารถเจาะลึกแยกลงไปถึงแต่ละแอทริบิวได้

***Review Predefined Queries and Reports*—**ผู้ใช้สามารถที่จะทบทวนเกี่ยวกับคิวรีและรายงานต่างๆที่มีการกำหนดไว้แล้ว และยังสามารถเลือกอันใด ๆ ขึ้นมาใช้งานได้



รูปที่ 9-11 การจัดเก็บเมตาดาต้าลงใน metadata repository

พื้นที่สำหรับเก็บเมตาดาต้าในอุดมคติจะเป็นแบบศูนย์กลางซึ่งจะเก็บข้อมูลทุกส่วนไว้รวมกัน ไม่ว่าจะเป็นเมตาดาต้าสำหรับผู้ใช้ ผู้พัฒนา และผู้ดูแลคลังข้อมูล แต่อย่างไรก็ดี ในการที่จะสร้างพื้นที่สำหรับจัดเก็บเมตาดาต้าในอุดมคตินั้นจะคุณสมบัติต่าง ๆ ดังนี้

***Integrated*—** ทำการเก็บเมตาดาต้าทั้งในเชิงธุรกิจและเชิงเทคนิคในรูปแบบที่มีความหมายเพื่อที่ผู้ใช้ทุกประเภทจะสามารถใช้งานได้

***Good Compartmentalization*—** สามารถที่จะแยกและจัดเก็บโมเดลของฐานข้อมูลทั้งแบบ logical และ physical

***Flexible organization*—** อนุญาตให้ผู้ดูแลข้อมูลทำการแบ่งและจัดการเมตาดาต้าให้เป็นกลุ่มต่าง ๆ และยังสามารถทำการกำหนดส่วนประกอบต่างๆของเมตาดาต้าที่ใช้การการแบ่งกลุ่มอีกด้วย

***Analysis and look-up Capabilities*—** ความสามารถในการเรียกดูทุกส่วนของเมตาดาต้าและยังนำทางผ่านทางความสัมพันธ์

***Historica*—** ทำการแบ่งเมตาดาต้าออกเป็นเวอร์ชั่นต่าง ๆ เพื่อที่จะเก็บประวัติของเมตาดาต้าไว้

Customizable — สามารถที่จะกำหนดมุมมองของเมตาดาต้าสำหรับกลุ่มของผู้ใช้และให้รวมเมตาดาต้าใหม่เข้ากับเมตาดาต้าเก่าตามความจำเป็น

Standardize Naming Conventions — มีความยืดหยุ่นในการที่ตั้งชื่อให้กับเมตาดาต้าและสร้างมาตราฐานให้กับข้อมูลทั้งหมดในพื้นที่สำหรับเก็บเมตาดาต้า

Synchronization — ควรทำให้เมตาดาต้ามีความสอดคล้องกันในทุก ๆ ส่วนของคลังข้อมูล รวมถึงสอดคล้องกับระบบภายนอกด้วย

Maintain Descriptions and Definitions — สามารถเรียกดูข้อมูลเมตาดาต้าทั้งในเชิงธุรกิจและเชิงเทคนิคได้

Open — ควรสนับสนุนการแลกเปลี่ยนเมตาดาต้าระหว่างขั้นตอนการทำงานต่างๆ ผ่านทางอินเทอร์เฟซ และทำให้เมตาดาต้าที่สร้างขึ้นจากเครื่องมือต่างๆมีความสอดคล้องกัน

SECTION 8

คำถามท้ายบท

1. เพราะเหตุใดเมตาดาต้าจึงมีความสำคัญกับคลังข้อมูล จงอธิบาย
2. จงอธิบายเหตุผลที่เมตาดาต้ามีความสำคัญกับการสร้างและการดูแลคลังข้อมูล
3. จงแจกแจงชนิดของเมตาดาต้า ว่าแต่ละชนิดมีลักษณะเป็นอย่างไร
4. จงยกตัวอย่างขั้นตอนการทำงานที่เมตาดาต้ามีช่วยช่วยในกระบวนการทำงาน
5. จงยกตัวอย่างเมตาดาต้าเชิงธุรกิจ 5 ตัวอย่าง
6. จงยกตัวอย่างเมตาดาต้าเชิงเทคนิค 5 ตัวอย่าง
7. จงแจกแจงผู้ที่ได้รับประโยชน์จากการจัดเก็บเมตาดาต้าเชิงธุรกิจและเชิงเทคนิค
8. จงยกตัวอย่างอุปสรรคและความท้าทายในการจัดเก็บเมตาดาต้า 5 ตัวอย่าง
9. จงอธิบายถึงการจัดเก็บเมตาดาต้าใน metadata repository
10. จงยกตัวอย่างเมตาดาต้าในอุดมคติ 5 ตัวอย่าง

CHAPTER 10

คุณภาพของข้อมูลในคลังข้อมูล

SECTION 1

แผนการสอนประจำบท

วัตถุประสงค์ของบทเรียน

- ทำความเข้าใจเหตุผลที่คุณภาพของข้อมูลจะเป็นปัจจัยที่สำคัญของการสร้างคลังข้อมูล
- ศึกษาเกี่ยวกับประโยชน์ของการมีข้อมูลที่มีคุณภาพ
- ศึกษาเกี่ยวกับปัญหาที่จะเกิดขึ้นเมื่อข้อมูลไม่มีคุณภาพ
- ศึกษาเกี่ยวกับเครื่องมือในการปรับปรุงคุณภาพของข้อมูล

เนื้อหาของบทเรียน

เนื้อหาในบทนี้จะประกอบด้วย สาเหตุที่คุณภาพข้อมูลเป็นสิ่งสำคัญ อุปสรรคและความท้าทายของการทำให้ข้อมูลมีคุณภาพ เครื่องมือสำหรับการปรับปรุงคุณภาพของข้อมูล การปรับปรุงคุณภาพของข้อมูล

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารประกอบการสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

กิจกรรมการเรียน-การสอน

- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารประกอบการสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

การวัดและประเมินผล

- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

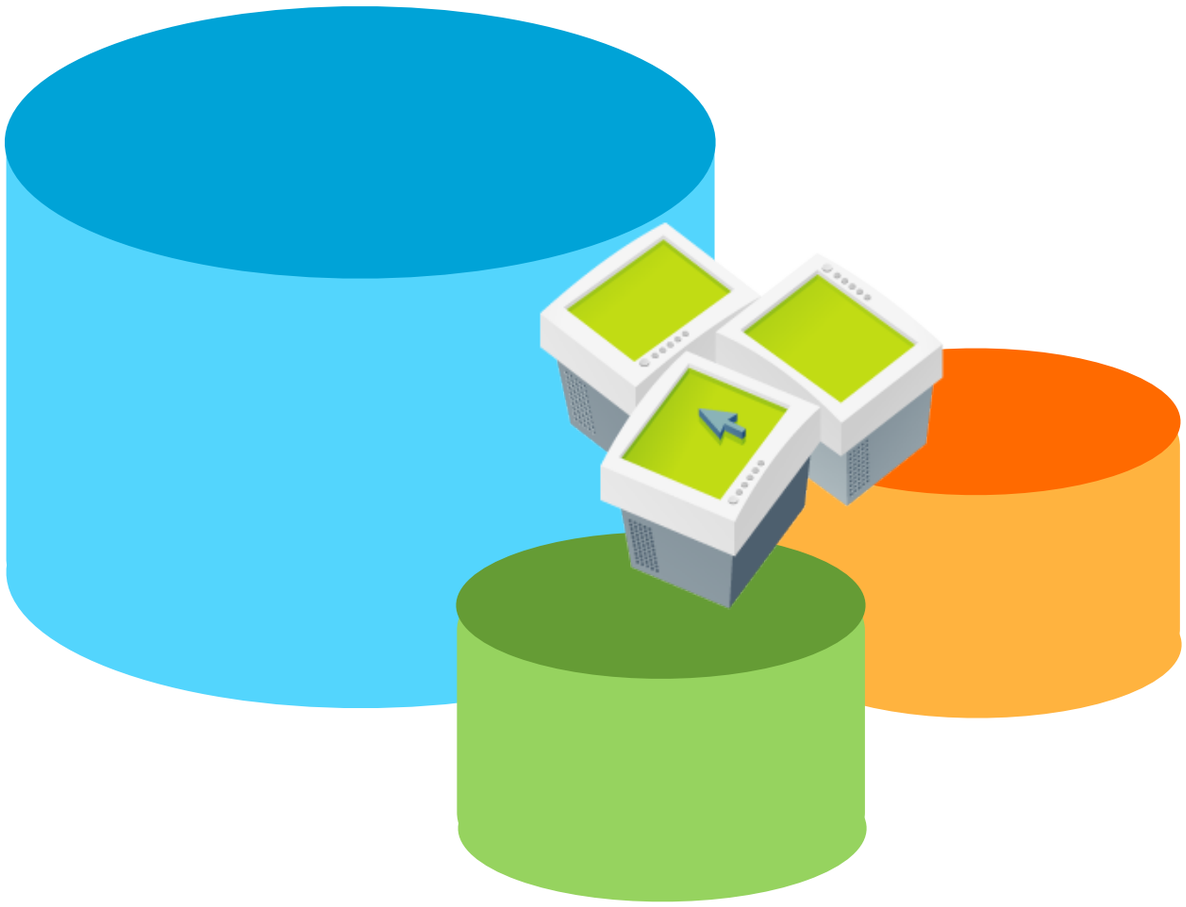
SECTION 2

บทนำ

บทนำ

โดยปกติแล้วบริษัทส่วนใหญ่ที่ใช้ระบบคอมพิวเตอร์ในการดำเนินธุรกิจมักจะมีความเชื่อมั่นในคุณภาพของข้อมูลที่ระบบทำการประมวลผลหรือทำการเก็บข้อมูลไว้ โดยจะมีบริษัทจำนวนน้อยมากที่มีขั้นตอนหรือระบบที่จะตรวจสอบคุณภาพของข้อมูล แต่อย่างไรก็ตาม ในหลาย ๆ ระบบก็ยังคงมีข้อมูลที่ไม่มีคุณภาพ หรือข้อมูลที่ผิดพลาดอยู่ได้ (ในบางครั้งเราอาจเรียกข้อมูลที่มีความผิดพลาดว่า**ข้อมูลที่ไม่สะอาด**) ข้อมูลที่ไม่สะอาดหรือไม่มีคุณภาพเป็นเหตุผลหลักของความล้มเหลวในการสร้างคลังข้อมูล เนื่องจาก**เมื่อข้อมูลไม่มีคุณภาพเป็นที่ยอมรับได้ ผู้ใช้ก็จะหมดความเชื่อมั่นต่อคลังข้อมูล และอาจจะไม่ใช้งานคลังข้อมูลอีก** เมื่อผู้ใช้หมดความเชื่อมั่นไปแล้วจะเป็นการยากมากที่จะเรียกความมั่นใจเหล่านั้นกลับมา

ดังนั้นเมื่อเราทราบว่าในระบบมีข้อมูลที่ไม่สะอาดอยู่ เราจะต้องหาระดับของความไม่สะอาดของข้อมูลว่าเป็นเช่นไร จากนั้นพยายามหาวิธีการหรือขั้นตอนวิธีที่จะทำความสะอาดข้อมูล

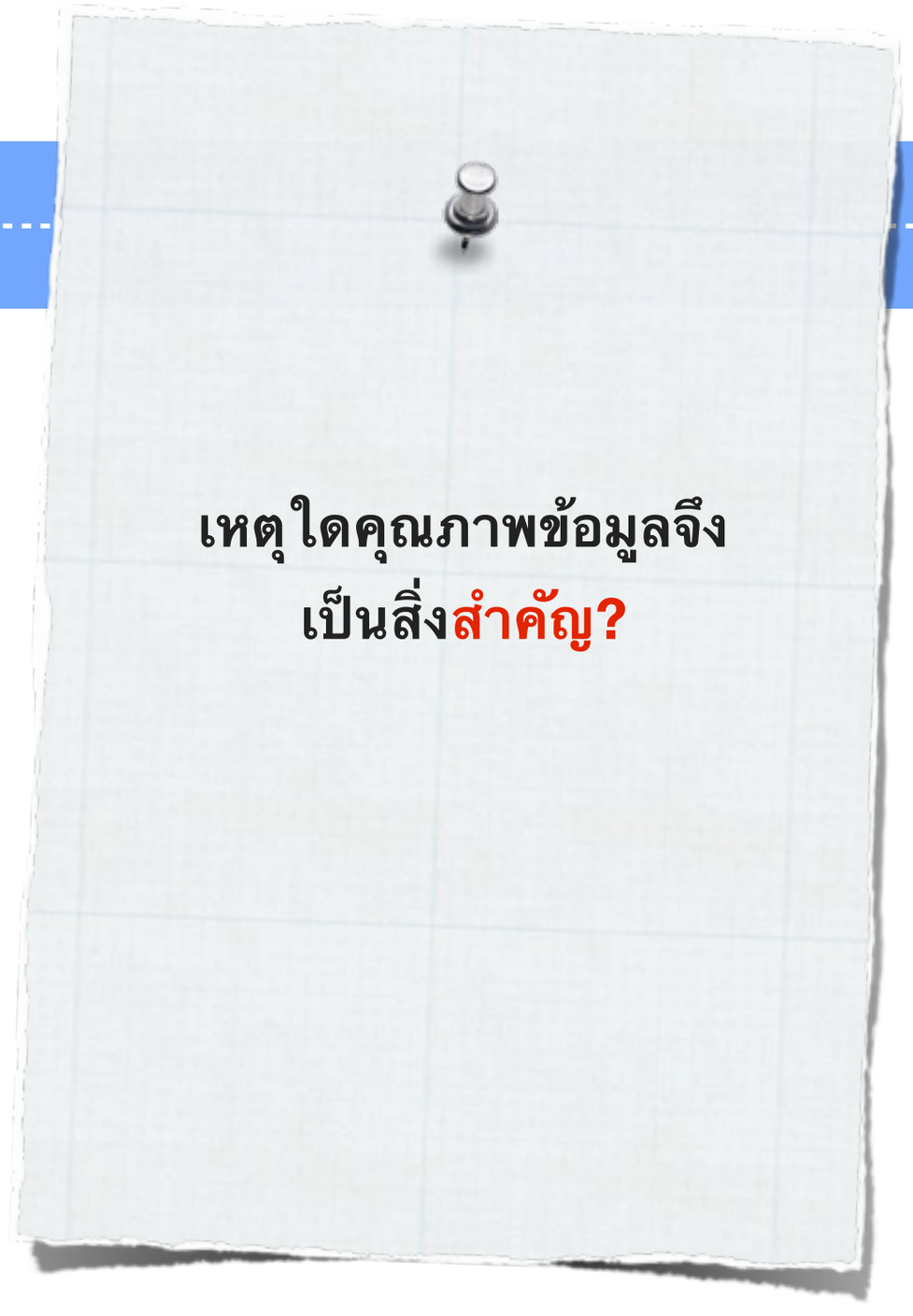


ในการสร้างคลังข้อมูลจากระบบการดำเนินงาน หรือแหล่งข้อมูลที่มีการกระจายตัวอยู่หลายแหล่ง และแต่ละแหล่งข้อมูลอาจมีความแตกต่างกันในโครงสร้างการจัดเก็บข้อมูลรวมถึงสถาปัตยกรรมของระบบ เราจะต้องเริ่มด้วยสมมติฐานที่ว่าข้อมูลนั้นมีความผิดพลาดอยู่

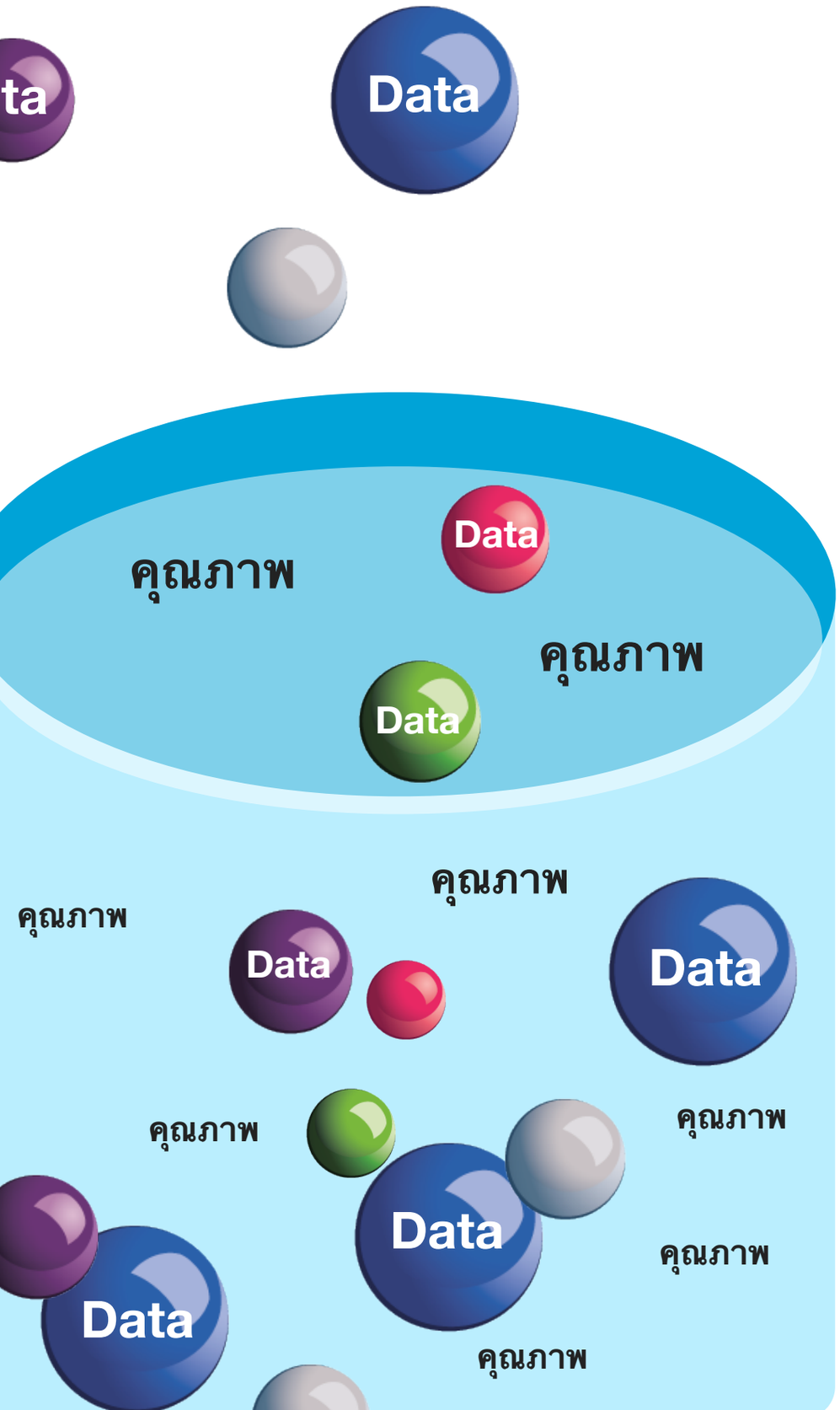
ดังนั้นเราจะต้องค้นหาระบบที่มีความผิดพลาดของข้อมูล แล้วทำการปรับปรุงข้อมูลให้มีคุณภาพมากขึ้น ข้อมูลที่ไม่มีคุณภาพจากระบบการดำเนินงานหรือแหล่งข้อมูลอาจเป็นสาเหตุให้ทำให้เกิดการตัดสินใจที่ผิดพลาด เมื่อผู้ใช้ทำการตัดสินใจจากข้อมูลที่ได้จากคลังข้อมูล เราควรจะให้ความสำคัญกับคุณภาพของข้อมูลที่ใช้ในการสร้างคลังข้อมูล

SECTION 3

เหตุใดคุณภาพข้อมูลจึงเป็นสิ่งสำคัญ?



เหตุใดคุณภาพข้อมูลจึง
เป็นสิ่งสำคัญ?



คุณภาพของข้อมูลในคลังข้อมูลมีความสำคัญมากและมีความสำคัญมากกว่าระบบการดำเนินงานเสียอีก เหตุที่คุณภาพของข้อมูลในคลังข้อมูลมีความสำคัญมาก

เนื่องจากผู้บริหารจะนำข้อมูลเหล่านี้มาใช้ในการตัดสินใจเชิงกลยุทธ์ซึ่งเป็นการตัดสินใจที่ส่งผลในวงกว้างต่อการดำเนินธุรกิจ ดังนั้นเราควรจะทำการปรับปรุงคุณภาพของข้อมูลในคลังข้อมูล เมื่อเราทำการปรับปรุงคุณภาพของข้อมูลแล้วอาจทำให้เกิดเหตุการณ์ดังต่อไปนี้

- เพิ่มความมั่นใจให้กับกระบวนการตัดสินใจ
- สามารถเพิ่มประสิทธิภาพในการบริการลูกค้า
- เพิ่มโอกาสในการเพิ่มมูลค่าในการบริการ
- ลดความเสี่ยงในการตัดสินใจผิดพลาด
- ลดต้นทุน
- เพิ่มประสิทธิภาพในการตัดสินใจเชิงกลยุทธ์
- เพิ่มผลผลิตด้วยขั้นตอนวิธีการที่มีประสิทธิภาพ
- หลีกเลี่ยงผลกระทบที่เพิ่มขึ้นจากข้อมูลที่ไม่มีคุณภาพ

คุณภาพของข้อมูล คืออะไร?

คุณภาพของข้อมูลในคลังข้อมูลไม่ได้หมายถึงคุณภาพของข้อมูลใดข้อมูลหนึ่งเท่านั้น แต่หมายถึงคุณภาพของข้อมูลทั้งหมดในคลังข้อมูล ถ้าข้อมูลเป็นไปตามวัตถุประสงค์ที่ตั้งไว้และถูกใช้โดยถูกวิธี เราจะสามารถเรียกได้ว่าข้อมูลนั้นมีคุณภาพ คุณภาพของข้อมูลจะเกี่ยวข้องกับความถูกต้องของข้อมูลที่สามารถบ่งบอกได้ถึงข้อมูลนั้น ๆ แต่อย่างไรก็ดีคุณภาพของข้อมูลยังมีปัจจัยอื่นๆที่เกี่ยวข้องที่มากกว่าความถูกต้องของข้อมูลเพียงอย่างเดียว ดังนั้นก่อนที่จะเข้าใจถึงปัจจัยต่างๆของคุณภาพของข้อมูล เราจำเป็นต้องเข้าใจถึงความแตกต่างระหว่างคุณภาพของข้อมูลและความถูกต้องของข้อมูลเสียก่อน

DATA INTEGRITY	DATA QUALITY
Specific instance of an entity accurately represents that occurrence of the entity.	The data item is exactly fit for the purpose for which the business users have defined it.
Data element defined in terms of database technology.	Wider concept grounded in the specific business of the company.
Data element conforms to validation constraints.	Relates not just to single data elements but to the system as a whole.
Individual data items have the correct data types.	Form and content of data elements consistent across the whole system.
Traditionally relates to operational systems.	Essentially needed in a corporate-wide data warehouse for business users.

รูปที่ 10-1 การเปรียบเทียบระหว่างความถูกต้องและคุณภาพของข้อมูล

DATA INTEGRITY	DATA QUALITY
Specific instance of an entity accurately represents that occurrence of the entity.	The data item is exactly fit for the purpose for which the business users have defined it.
Data element defined in terms of database technology.	Wider concept grounded in the specific business of the company.
Data element conforms to validation constraints.	Relates not just to single data elements but to the system as a whole.
Individual data items have the correct data types.	Form and content of data elements consistent across the whole system.
Traditionally relates to operational systems.	Essentially needed in a corporate-wide data warehouse for business users.

รูปที่ 10-1 การเปรียบเทียบระหว่างความถูกต้องและคุณภาพของข้อมูล

รูปที่ 10-1 แสดงถึงความแตกต่างระหว่างความถูกต้องของข้อมูลและคุณภาพของข้อมูล แต่อย่างไรก็ดี เราจะสามารถกำหนดหรือระบุถึงคุณภาพของข้อมูลได้อย่างไร? จะสามารถพิจารณาได้อย่างไรว่าข้อมูลใดมีคุณภาพสูงหรือไม่? ถ้าเราสามารถพิจารณาได้ว่าข้อมูลมีคุณภาพสูง คำถามที่ตามมาก็คือ เราจะพิจารณาว่าข้อมูลใดมีคุณภาพสูงได้อย่างไรโดยใช้วิธีใด? จากคำถามที่ได้กล่าวมาข้างต้น จะมีวิธีการที่เป็นรูปธรรมสำหรับจำแนกคุณภาพของข้อมูลในคลังข้อมูลที่จะบอกถึงคุณลักษณะของข้อมูลที่มีคุณภาพสูง ดังนั้น ในการสร้างคลังข้อมูลเราอาจใช้วิธีเหล่านี้เพื่อเป็นการวัดคุณภาพของข้อมูลจากแหล่งข้อมูลได้ โดยที่วิธีการเหล่านี้จะพิจารณาถึงหลายๆ องค์ประกอบดังนี้

Accuracy

หมายถึง ค่าของข้อมูลที่ถูกเก็บไว้ในระบบนั้นเป็นค่าที่ถูกต้องสำหรับข้อมูลนั้น ๆ ตัวอย่างเช่น ถ้าเรามีข้อมูล ชื่อและที่อยู่ของลูกค้าเก็บอยู่ในเรคคอร์ดหนึ่งๆ ข้อมูลอยู่ที่เก็บไว้จะต้องเป็นที่อยู่ที่ถูกต้องของลูกค้าชื่อนั้น ๆ และถ้าเราทำการหายอดสั่งซื้อสินค้าเป็นจำนวน 1,000 ชิ้น ในเรคคอร์ดของการสั่งซื้อเลขที่ 12345678 จำนวนสินค้าที่ถูกต้องจะต้องสอดคล้องกับเลขที่ใบสั่งซื้อด้วย

Domain Integrity

หมายถึง ค่าของข้อมูล ในแอทริบิวหนึ่งๆจะต้องอยู่ในช่วงที่เรากำหนดไว้ ตัวอย่างเช่น ในการเก็บข้อมูลเพศของลูกค้า ซึ่งมีค่าที่เรายอมรับคือ “male” และ “female” เท่านั้น

Data Type

หมายถึง ค่าของข้อมูล ในแอทริบิวหนึ่งๆ จะต้องถูกเก็บให้ตรงกับชนิดของข้อมูลที่กำหนดไว้ ตัวอย่างเช่น ถ้าเราทำการกำหนดชนิดของข้อมูลที่จะทำการจัดเก็บข้อมูลชื่อลูกค้าไว้เป็นข้อความ “text” ดังนั้นทุกข้อมูลที่ถูกเก็บไว้ในแอทริบิวที่ใช้เก็บชื่อลูกค้าจะต้องประกอบไปด้วยข้อมูลที่เป็นข้อความเท่านั้น ไม่สามารถเป็นชุดของตัวเลขได้

Consistency

หมายถึง รูปแบบและเนื้อหาของข้อมูลในฟิลด์หนึ่ง ๆ จะต้องเหมือนกันในทุกแหล่งข้อมูล ถ้าเราทำการเก็บรหัสสินค้า ABC ในแหล่งข้อมูลหนึ่ง ๆ เป็น 1234 ดังนั้นรหัสของสินค้า ABC ในแหล่งข้อมูลอื่น ๆ จะต้องใช้รหัส 1234 ด้วย

Completeness

หมายถึง การไม่มีการขาดหายไปของข้อมูล (Missing value) เกิดขึ้นในแอทริบิว หนึ่ง ๆ ตัวอย่างเช่น ข้อมูลการสั่งสินค้า ทุกรายละเอียดของการสั่งซื้อจะต้องถูกกรอกทั้งหมด

Redundancy

หมายถึง ข้อมูลที่เหมือนกันจะไม่สามารถเก็บได้มากกว่าหนึ่งที่ในระบบหนึ่งๆ แต่อย่างไรก็ตาม เราสามารถยอมให้เกิดการซ้ำซ้อนของข้อมูลได้ ซึ่งการซ้ำซ้อนของข้อมูลอาจเกิดจากความต้องการที่จะเพิ่มประสิทธิภาพในการเข้าถึงข้อมูล

Duplication

หมายถึง เรคคอร์ดที่ซ้ำกันในระบบจะต้องถูกแยกแยะ เช่น ถ้าข้อมูลสินค้ามีการซ้ำกัน เราจะต้องทำการระบุถึงทุกเรคคอร์ดที่ซ้ำกันของแต่ละสินค้าและต้องทำการอ้างอิงเพิ่ม

Conformance to Business Rules

หมายถึง ข้อมูลสำหรับแต่ละรายการข้อมูลจะต้องสอดคล้องกับกฎทางธุรกิจ ตัวอย่างเช่น ระบบการขายด้วยการประมูล ราคาที่ขายสินค้าจะต้องไม่ต่ำกว่าราคาที่ตั้งไว้ หรือ ในระบบการกู้ยืมเงินจากธนาคาร ยอดเงินการกู้ยืมจะต้องเป็นเลขบวกหรือเป็นศูนย์เท่านั้น

Structural Definiteness

หมายถึง รายการข้อมูลหนึ่ง ๆ จะต้องมีโครงสร้างที่ชัดเจน และเราจะต้องเก็บข้อมูลตามโครงสร้างนั้น ๆ ตัวอย่างเช่น ชื่อของลูกค้าจะประกอบไปด้วย ชื่อต้น ชื่อกลาง และ นามสกุล ดังนั้นในการเก็บข้อมูลเราควรจะเก็บตามโครงสร้างของข้อมูลนั้น คือ เราควรจะเก็บข้อมูลทั้ง ชื่อต้น ชื่อกลาง และนามสกุล ซึ่งการเก็บข้อมูลตามโครงสร้างของข้อมูลจะช่วยให้ข้อมูลมีมาตราฐานและลดการขาดหายไปของข้อมูลได้

Data Anomaly

หมายถึง ข้อมูลฟิลด์หนึ่ง ๆ จะต้องถูกจัดเก็บตามวัตถุประสงค์ที่ตั้งไว้ เช่น ข้อมูล Address-3 ถูกกำหนดไว้เพื่อเก็บข้อมูลที่อยู่เป็นบรรทัดที่ 3 ซึ่งฟิลด์นี้จะถูกใช้ก็ต่อเมื่อข้อมูลที่อยู่ยาวมาก ดังนั้นเราจะต้องใช้ข้อมูล filed Address-3 สำหรับเก็บข้อมูลในลำดับที่ 3 เท่านั้น ห้ามใช้ในการเก็บข้อมูลอื่น ๆ เช่น เบอร์โทรศัพท์ หรือเบอร์แฟกซ์ เป็นต้น

Clarity

หมายถึง องค์ประกอบของข้อมูลอาจมีคุณลักษณะที่จะเป็นข้อมูลที่มีคุณภาพแต่ถ้าผู้ใช้ไม่เข้าใจความหมายของข้อมูลอย่างชัดเจน ข้อมูลนั้นก็จะไม่ได้มีคุณค่าสำหรับผู้ใช้ ดังนั้น การกำหนดข้อตกลงในการตั้งชื่อหรือการเขียนคำอธิบายข้อมูลจะสามารถช่วยให้ผู้ใช้มีความเข้าในองค์ประกอบของข้อมูลได้ง่ายขึ้น

Timely

หมายถึง การที่ผู้ใช้สามารถกำหนดช่วงเวลาที่เหมาะสมให้กับข้อมูลได้ (อายุของข้อมูล) เช่น ถ้าผู้ใช้คาดหวังว่าข้อมูลลูกค้าที่เก็บอยู่ใน Customer dimension ไม่ควรจะมีอายุเกิน 1 หนึ่งวัน ดังนั้นความเปลี่ยนแปลงของข้อมูลลูกค้าจากแหล่งข้อมูลจะต้องถูกถ่ายโอนเข้าสู่คลังข้อมูลทุก ๆ วัน

Usefulness

หมายถึง ข้อมูลทุกข้อมูลในคลังข้อมูลจะต้องตอบสนองความต้องการของผู้ใช้ ในบางกรณีข้อมูลจากคลังข้อมูลที่มีความถูกต้องและมีคุณภาพอาจจะไม่ได้มีคุณค่าต่อผู้ใช้ก็เป็นได้ ซึ่งข้อมูลเหล่านี้ไม่ควรจะถูกเก็บอยู่ในคลังข้อมูล

Adherence to Data Integrity Rules

หมายถึง ข้อมูลที่ถูกเก็บอยู่ในฐานข้อมูลของแหล่งข้อมูลจะต้องยึดมั่นกับกฎ entity integrity (ข้อมูลที่มีเอกลักษณ์โดยสิ้นเชิง) และ กฎ referential integrity (ข้อมูลที่สามารถอ้างอิงได้) โดยที่ตารางใดก็ตามที่อนุญาตให้สามารถมีค่า NULL ได้ในคีย์หลัก (Primary key) ตารางนั้นจะไม่เป็นไปตามกฎ entity integrity ในส่วนของกฎ referential integrity จะเป็นการบังคับเกี่ยวกับความสัมพันธ์แบบ parent-child ตัวอย่างเช่น ถ้าในการสั่งซื้อสินค้าของลูกค้าเป็นไปตามกฎ referential integrity เราสามารถมั่นใจได้ว่าในการสั่งซื้อสินค้าใด ๆ ก็ตามจะต้องมีข้อมูลเกี่ยวกับลูกค้าแนบอยู่ด้วยเสมอ

ประโยชน์ของการปรับปรุงคุณภาพข้อมูล

ประโยชน์ของการปรับปรุงคุณภาพข้อมูล

อย่างที่เราทราบกันดีว่า “ข้อมูลที่ไม่ดีจะทำให้เกิดการตัดสินใจที่ไม่ดี” ดังนั้นเราจะทำการปรับปรุงคุณภาพของข้อมูลให้ดีขึ้นเพื่อช่วยเหลือในการตัดสินใจเชิงกลยุทธ์ ซึ่งในการพิจาณาถึงคุณภาพของข้อมูลหรือการปรับปรุงข้อมูลให้ดีขึ้นจะมีประโยชน์ดังต่อไปนี้

Analysis with Timely Information

สมมติว่าร้านค้าปลีกขนาดใหญ่ทำโปรโมชั่นกับสินค้ามากมายแบบวันต่อวัน และโปรโมชั่นนี้ได้ถูกใช้กับร้านค้ากว่า 200 สาขาทั่วประเทศ ข้อมูลการทำโปรโมชั่นจะถูกเก็บไว้เป็นหนึ่ง dimension ของคลังข้อมูล ถ้านักการตลาดต้องการที่จะวิเคราะห์ข้อมูลโดยใช้ข้อมูลโปรโมชั่นเพื่อที่จะเฝ้าดูและปรับโปรโมชั่นตามยอดการขาย โดยการวิเคราะห์ข้อมูลจะทำทุกวันเนื่องจากจะมีการออกโปรโมชั่นใหม่ๆทุกวัน สมมติว่าข้อมูลโปรโมชั่นจะถูกทำการถ่ายโอนเข้าสู่คลังข้อมูลอาทิตย์ละครั้ง คุณคิดว่าข้อมูลโปรโมชั่นจะมีอายุที่เหมาะสมสำหรับนักการตลาดหรือไม่ ข้อมูลโปรโมชั่นในคลังข้อมูลจะเป็นข้อมูลที่มีคุณภาพและเป็นที่ต้องการของผู้ใช้หรือไม่

คำตอบของทั้งสองคำถามคือ **“ไม่”**
ดังนั้น จากเนื้อหาก่อนหน้าจะทำให้เราทราบว่าข้อมูลที่มีคุณภาพจะทำให้เราสามารถวิเคราะห์ข้อมูลได้ตามช่วงเวลาที่เหมาะสม นี้จึงเป็นข้อดีอย่างหนึ่งของข้อมูลที่มีคุณภาพที่เหมาะกับความต้องการทางด้านเวลาของผู้ใช้

Better Customer Service

ประโยชน์ของข้อมูลที่มีความถูกต้องและสมบูรณ์สำหรับการบริการลูกค้านั้นไม่สามารถเน้นย้ำจนเกินไปได้ พิจารณาเกี่ยวกับตัวแทนฝ่ายบริการลูกค้าในธนาคารขนาดใหญ่ที่ทำการรับโทรศัพท์ลูกค้าที่ต้องการพูดคุยเกี่ยวกับค่าบริการในการตรวจสอบยอดเงินในบัญชี

จากการตรวจสอบข้อมูลตัวแทนฝ่ายบริการลูกค้าเห็นว่าเงินในบัญชีมีอยู่เพียง €27.38 ซึ่งจากจำนวนเงินดังกล่าวจะทำให้เข้าใจได้ว่าทำไมลูกค้าถึงกังวลเกี่ยวกับค่าบริการ และอาจมองว่าลูกค้าไม่มีเงิน แต่ถ้าตัวแทนฝ่ายบริการลูกค้าลองไปตรวจสอบดูในบัญชีอื่น ๆ ลูกค้าและพบว่าลูกค้ามีเงินเก็บอยู่ €35,000 และมีบัตรเงินฝาก (Certificate of deposit: CD) ซึ่งมีมูลค่ามากกว่า €120,000 เมื่อตัวแทนฝ่ายบริการลูกค้าสามารถเห็นข้อมูลทั้งหมดแล้วคุณคิดว่าเขาจะตอบคำถามในโทรศัพท์อย่างไร?

แน่นอนว่าต้องตอบด้วยความสุภาพมาก เนื่องจากมีความเกรงใจในลูกค้าที่มีเงินค่อนข้างมาก จากตัวอย่างข้างต้นจะทำให้เห็นว่าข้อมูลที่มีความถูกต้องและสมบูรณ์สามารถที่จะช่วยปรับปรุงการให้บริการได้อย่างมาก

Newer Opportunities

คุณภาพของข้อมูลในคลังข้อมูลจะเป็นสิ่งที่เป็นประโยชน์อย่างมาก สำหรับนักการตลาด ซึ่งเมื่อเรามีข้อมูลที่มีคุณภาพจะทำให้เรา สามารถเพิ่มโอกาสเป็นอย่างมากในการขายสินค้าได้อย่างทั่วถึง ผู้ใช้คลังข้อมูลสามารถเลือกผู้ซื้อที่ซื้อสินค้าชนิดหนึ่งแล้วทำการหาว่าผู้ซื้อคนนั้นมีแนวโน้มที่จะซื้อสินค้าอื่น ๆ ด้วย

นอกจากนี้นักการตลาดยังสามารถสร้างเคมเปญให้แก่กลุ่มลูกค้าเป้าหมายได้อีกด้วย อย่างไรก็ดีถ้าข้อมูลไม่มีคุณภาพ เคมเปญที่สร้างขึ้นก็จะล้มเหลว

Reduced Costs and Risks

การมีข้อมูลที่มีคุณภาพไม่ดีจะมีความเสี่ยงอย่างไร? ความเสี่ยงที่เห็นได้ชัดเจนอย่างหนึ่งคือการตัดสินใจเชิงกลยุทธ์ที่อาจทำให้เกิดความเสียหายต่อการดำเนินธุรกิจได้ นอกจากนี้ยังมีความเสี่ยงอื่น ๆ อีก เช่น การเสียเวลา การไม่สามารถทำงานได้อย่างปกติของขั้นตอนการทำงานและระบบ และอื่น ๆ แต่อย่างไรก็ตาม ข้อมูลที่มีคุณภาพจะสามารถลดต้นทุนในการดำเนินงานได้ เช่น สามารถลดต้นทุนในการส่งแคมเปญทางการตลาดให้แก่ลูกค้าทางไปรษณีย์ ถ้าข้อมูลที่อยู่ของลูกค้าไม่สมบูรณ์ ไม่ถูกต้อง หรือซ้ำซ้อนกัน แคมเปญที่จัดทำขึ้นนั้นจะสูญเปล่า

Improved Productivity

ผู้ใช้สามารถดูข้อมูลได้แบบทั้งองค์กรซึ่งทำให้สามารถปรับปรุงประสิทธิภาพของกระบวนการทำงานได้ เพราะเราจะเห็นการเพิ่มผลผลิตได้ ตัวอย่างเช่น การดูข้อมูล purchasing patterns ของลูกค้าในห้างสรรพสินค้าขนาดใหญ่ จะทำให้เกิดการพัฒนาขั้นตอนการซื้อสินค้าของลูกค้าและกลยุทธ์ต่าง ๆ

ปัญหาที่เกี่ยวกับการมี
คุณภาพที่ไม่ดี

ปัญหาที่เกี่ยวกับการมีคุณภาพที่ไม่ดี

อย่างที่เราทราบกันดีว่า “คุณภาพของข้อมูลจะส่งผลถึงความถูกต้องในการตัดสินใจเชิงกลยุทธ์” ดังนั้นในการสร้างคลังข้อมูลเราจะต้องพยายามทำให้ข้อมูลมีความถูกต้องและมีคุณภาพ ไม่อย่างนั้นอาจทำให้เกิดความผิดพลาดในกระบวนการดำเนินธุรกิจได้ ตัวอย่างเช่น ถ้ามูลค่าของการขายสินค้ามีความผิดพลาดเกิดขึ้น 4% จากมูลค่าการขายสินค้าทั้งหมด $2,000,000,000 นั่นหมายความว่า เงินรายได้อาจสูญหายไป $80,000,000 เมื่อมีรายได้ต่ำกว่าข้อมูลที่เก็บอยู่ในระบบการดำเนินงานและคลังข้อมูลจะส่งผลในแนวกว้างต่อการดำเนินธุรกิจ

ในแง่ของการลงทุน การเสียภาษี และอื่น ๆ อีกตัวอย่างหนึ่งที่เห็นได้ชัดคือ ในการส่งแค็ตตาล็อกรายการสินค้าให้กับลูกค้าทางไปรษณีย์โดยใช้ข้อมูลที่อยู่ลูกค้าที่เก็บไว้ในระบบการดำเนินงาน แต่ถ้าข้อมูลที่อยู่ลูกค้าที่เก็บไว้มีความซ้ำซ้อนของข้อมูลเกิดขึ้น จะทำให้บริษัททำการส่งแค็ตตาล็อกรายสินค้าให้กับลูกค้าซ้ำซ้อน ซึ่งอาจจะสร้างความไม่พอใจให้แก่ลูกค้าได้ รวมถึงจะต้องสูญเสียงบประมาณในการทำรายการสินค้าและค่าส่งไปรษณีย์โดยไม่จำเป็น จากตัวอย่างข้างต้นเราจะเห็นว่าคุณภาพของข้อมูลนอกจากจะส่งผลต่อการตัดสินใจแล้ว ยังส่งผลถึงการดำเนินการทางธุรกิจอีกด้วย ดังนั้นเราจะต้องใส่ใจกับคุณภาพของข้อมูลและพยายามทำให้ข้อมูลมีคุณภาพมากที่สุด เพื่อที่จะไม่ให้เกิดปัญหาเกี่ยวกับคุณภาพของข้อมูลดังต่อไปนี้

Dummy Values in Fields

คุณมีความตระหนักถึงการกรอกข้อมูลที่ถูกต้องหรือไม่ หลาย ๆ ครั้งคุณอาจจะไม่ได้สนใจการกรอกข้อมูลที่ถูกต้อง หลาย ๆ ครั้งที่คุณอาจกรอกข้อมูล 88888 ในฟิลด์รหัสไปรษณีย์สำหรับลูกค้าในแถบเอเชีย และกรอกข้อมูล 77777 สำหรับลูกค้าฝั่งยุโรป ซึ่งเป็นข้อมูลที่ไม่ถูกต้อง

Absence of Data Values

การขาดหายไปของข้อมูลจะพบบ่อยในการเก็บข้อมูลลูกค้า ในระบบการดำเนินงาน ผู้ใช้จะสนใจเพียงข้อมูลที่อยู่ลูกค้าที่จะใช้ในการส่งใบแจ้งหนี้เรียกเก็บเงิน การส่งจดหมายติดตาม หรือการโทรศัพท์ไปหาลูกค้าเกี่ยวกับการจ่ายเงินเมื่อเกินกำหนดเวลาการจ่าย โดยส่วนใหญ่แล้วจะไม่สนใจข้อมูลในเชิงสถิติจำนวนประชากรที่ไม่ได้ใช้งานในระบบการดำเนินงาน ซึ่งจะทำให้เกิดการขาดหายไปของข้อมูลจำนวนประชากรโดยไม่รู้ตัว โดยที่ข้อมูลสถิติจำนวนประชากรจะมีประโยชน์อย่างมากในการวิเคราะห์ข้อมูลจากคลังขอมูล

Unofficial Use of Fields

มีกี่ครั้งที่คุณถามลูกค้าหรือผู้ใช้ของระบบให้ทำการกรอกข้อมูลความคิดเห็นในฟิลด์หนึ่งๆของข้อมูลที่ใช้ในการติดต่อลูกค้า โดยส่วนใหญ่แล้วระบบการดำเนินงานจะไม่ทำการเก็บข้อมูลความคิดเห็นลงในข้อมูลลูกค้า ซึ่งการนำข้อมูลความคิดเห็นของลูกค้าไปใช้จะเป็นการใช้ข้อมูลจากฟิลด์ที่ไม่เป็นทางการ

Absence of Data Values

เป็นปัญหาเกี่ยวกับข้อมูลที่มีความกำกวมหรือข้อมูลที่มีความลับซึ่งเราจะพบบ่อยในระบบการดำเนินงานที่มีการเข้ารหัสข้อมูลที่การจัดเก็บข้อมูลนั้นไม่ได้ออกแบบที่ตรงใจผู้ใช้งาน ตัวอย่างเช่น การตั้งรหัสข้อมูลโดยใช้ตัวอักษรใหญ่ตัวแรกของโค้ดที่ต้องการเก็บ ในตอนเริ่มต้นโค้ดของสถานะของลูกค้าอาจใช้เป็น R=Regular และ N=New ต่อมาได้ทำการเพิ่มข้อมูล D=Decreased และ A=Archive และทำการลบข้อมูล R และ N ออก เมื่อทำการใช้ระบบต่อไปเรื่อย ๆ อาจทำการเพิ่มข้อมูล R=Remove เข้าไปใหม่ ซึ่งการใช้ตัวอักษรตัวเดียวกันแทนโค้ดสถานะของข้อมูลที่แตกต่างกันจะก่อให้เกิดความกำกวมและสับสนกับข้อมูลได้

Contradicting Values

มีข้อมูลอยู่จำนวนมากในแหล่งข้อมูลที่เกี่ยวข้องกัน ตัวอย่างเช่น ข้อมูลชื่อรัฐและรหัสไปรษณีย์จะเกี่ยวข้องกัน ซึ่งในการเกี่ยวข้องกันข้อมูลนั้นอาจเกิดความผิดพลาดได้ เช่น ระบบดำเนินการทำการเก็บข้อมูลรัฐแคลิฟอร์เนีย และรหัสไปรษณีย์เป็น 08817 (ซึ่งเป็นรหัสไปรษณีย์ของรัฐนิว เจอซี่ย์) ไว้ในเรคคอร์ดเดียวกัน ซึ่งเป็นข้อมูลที่มีความผิดพลาด

Reused Primary Keys

สมมติว่าระบบการดำเนินงานกำหนดคีย์หลักของข้อมูลลูกค้าเป็นตัวเลขจำนวน 5 หลัก ซึ่งเป็นตัวเลขที่รองรับจำนวนลูกค้าที่น้อยกว่า 100,000 ราย แต่เมื่อวันเวลาผ่านไปลูกค้ามีจำนวนเพิ่มขึ้นมากเกินกว่า 100,000 ราย โดยที่ลูกค้าบางรายก็ไม่ทำการซื้อ-ขายสินค้าจากบริษัทนั้น ๆ อีก บางบริษัททำการแก้ไขปัญหานี้โดยทำการเก็บข้อมูลลูกค้าเก่าไว้และทำการกำหนดค่าคีย์หลักให้กับลูกค้าใหม่โดยใช้คีย์หลักเริ่มต้นที่เลข 1 การแก้ปัญหาในลักษณะนี้จะไม่เป็นปัญหากับระบบดำเนินการ แต่จะทำให้เกิดปัญหาในคลังข้อมูลในการเข้าถึงข้อมูลลูกค้าใหม่ (หลังการเปลี่ยนแปลงค่าเริ่มต้นของคีย์หลักแล้ว) และข้อมูลลูกค้าเก่าที่ทำการเก็บไว้ ซึ่งจากการเข้าถึงข้อมูลทั้งสองชนิดจะทำให้เกิดปัญหาการซ้ำซ้อนของการใช้คีย์หลักซ้ำ

Violation of Business Rules

ในระบบพนักงานและระบบบัญชีเงินเดือน เราจะเห็นกฎทางธุรกิจที่ว่าจำนวนวันที่ทำงาน รวมกับ วันลากิจ วันหยุดพักผ่อน และ วันลาป่วย จะต้องมีค่าไม่เกิน 365 หรือ 366 วัน ดังนั้น ถ้าจำนวนวันรวมทั้งหมดของพนักงานมีค่าเกิดกว่าจำนวนวันในหนึ่งปี การมีข้อมูลแบบนี้จะทำให้เกิดการฝ่าฝืนกฎทางธุรกิจ

Nonunique Identifiers

สมมติว่าระบบบัญชีมีการกำหนดโค้ดของสินค้าเองซึ่งแตกต่างจากโค้ดสินค้าในระบบการขายและระบบคลังสินค้า ตัวอย่างเช่น โค้ด 355 จะแสดงถึงรายการสินค้าหนึ่ง ๆ ในระบบการขาย ซึ่งจะเป็นสินค้าเดียวกับโค้ด A226 ในระบบบัญชี การทำงานในลักษณะนี้จะทำให้เกิดปัญหาการอ้างถึงตัวสินค้าที่ว่า โค้ดของสินค้าหนึ่ง ๆ ไม่สามารถระบุถึงสินค้าได้ในสองระบบที่มีการทำงานเชื่อมต่อกัน

Inconsistent Values

เป็นปัญหาการไม่สอดคล้องกันของข้อมูล เช่น โค้ดของชนิดของนโยบายในบริษัทประกันอาจจะใช้ไม่เหมือนกันในแต่ละสาขา เช่น สาขาหนึ่งอาจใช้ A=Auto, H=Home, F=Flood, W=Workers Comp แต่ในอีกสาขาหนึ่งอาจใช้ 1=Auto, 2=Home, 3=Flood และ 4=Workers Comp ตามลำดับ

Incorrect Values

สมมติว่าเราทำการเก็บข้อมูลรหัสสินค้าเป็น 146 ชื่อสินค้าเป็น แจกันคริสตัล และ ความสูงของสินค้าเป็น 486 นิ้ว ซึ่งข้อมูลที่เก็บทั้งหมดจะถูกเก็บอยู่ในเรคคอร์ดเดียวกัน จากตัวอย่างข้างต้นชี้ให้เห็นว่าข้อมูลที่ทำการเก็บนั้นไม่ถูกต้อง เนื่องจากข้อมูลชื่อสินค้าและความสูงของสินค้าไม่สอดคล้องกัน (เนื่องจากคงไม่มีใครสร้างแจกันสูง 486 นิ้วแน่ๆ) และโดยส่วนใหญ่แล้วเมื่อเกิดความไม่สอดคล้องกันของข้อมูลเกิดขึ้น รหัสสินค้าที่ทำการเก็บไว้จะเกิดความผิดพลาดด้วย

Multipurpose Fields

ข้อมูลที่เหมือนกันในฟิลด์หนึ่ง ๆ แต่ถูกเพิ่มข้อมูลจากแผนกที่ต่างกันอาจหมายถึงสิ่งที่แตกต่างกันได้ฟิลด์หนึ่ง ๆ ควรจะถูกเก็บไว้เป็นแบบรหัสของส่วนที่เก็บข้อมูล (storage area code) เพื่อที่จะสามารถอ้างอิงได้ถึงส่วนที่ทำการเก็บข้อมูลหรือควรจะทำการแยกฟิลด์โดยให้ความหมายของแต่ละฟิลด์นั้น ๆ

Erroneous Integration

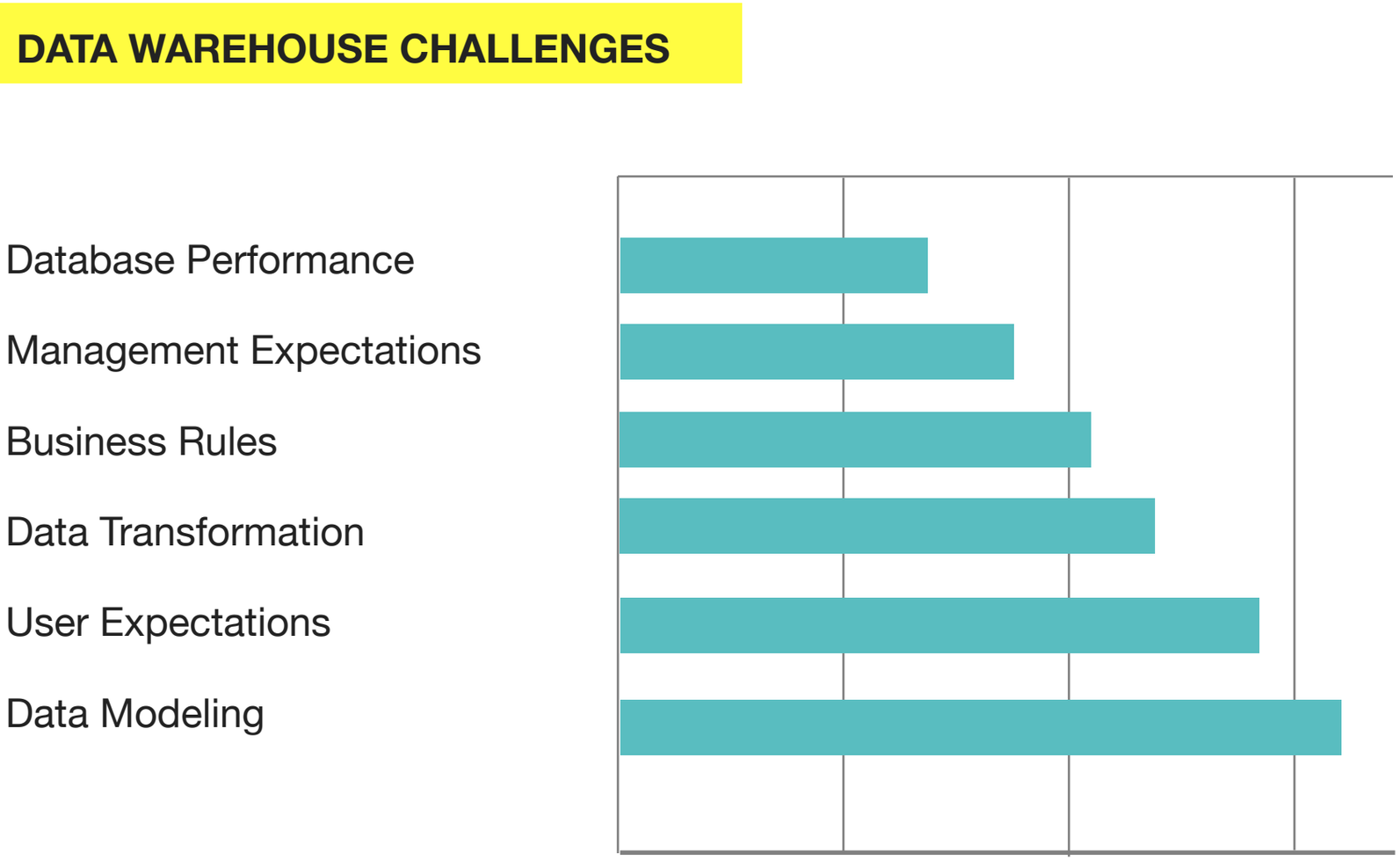
ในบริษัทการประมูล ผู้ซื้อ คือ ลูกค้าที่ทำการประมูลสินค้าชนะ กล่าวคือให้ราคาสินค้าสูงที่สุดจากนั้นทำการซื้อสินค้าหลังจากการประมูลเสร็จสิ้น ผู้ขาย คือ ลูกค้าที่ทำการขายสินค้าผ่านระบบการประมูล ลูกค้าของบริษัทการประมูลอาจเป็นทั้งผู้ซื้อและผู้ขายก็ได้ ถ้าระบบที่เก็บข้อมูลผู้ซื้อและระบบการออกใบเสร็จสร้างไม่พร้อมกันและมีการแยกการทำข้อมูล จะทำให้ลูกค้าที่มีรหัสเป็น 12345 จากระบบการประมูลสินค้า แต่จะมีรหัสเป็น 34567 ในระบบการออกใบเสร็จ เมื่อทำการรวมระบบเข้าด้วยกันแล้วจะทำให้เกิดความแตกต่างของข้อมูลลูกค้าเกิดขึ้นเช่น ลูกค้าที่รหัส 55555 ในระบบการประมูลสินค้า จะเป็นคนละคนกับลูกค้าที่รหัส 55555 ในระบบการออกใบเสร็จ ซึ่งจะทำให้เกิดปัญหาความไม่สอดคล้องของข้อมูลซึ่งอาจเกิดขึ้นในการสร้างระบบที่แยกจากกัน โดยทำการสร้างข้อมูลไม่พร้อมกัน

SECTION 4

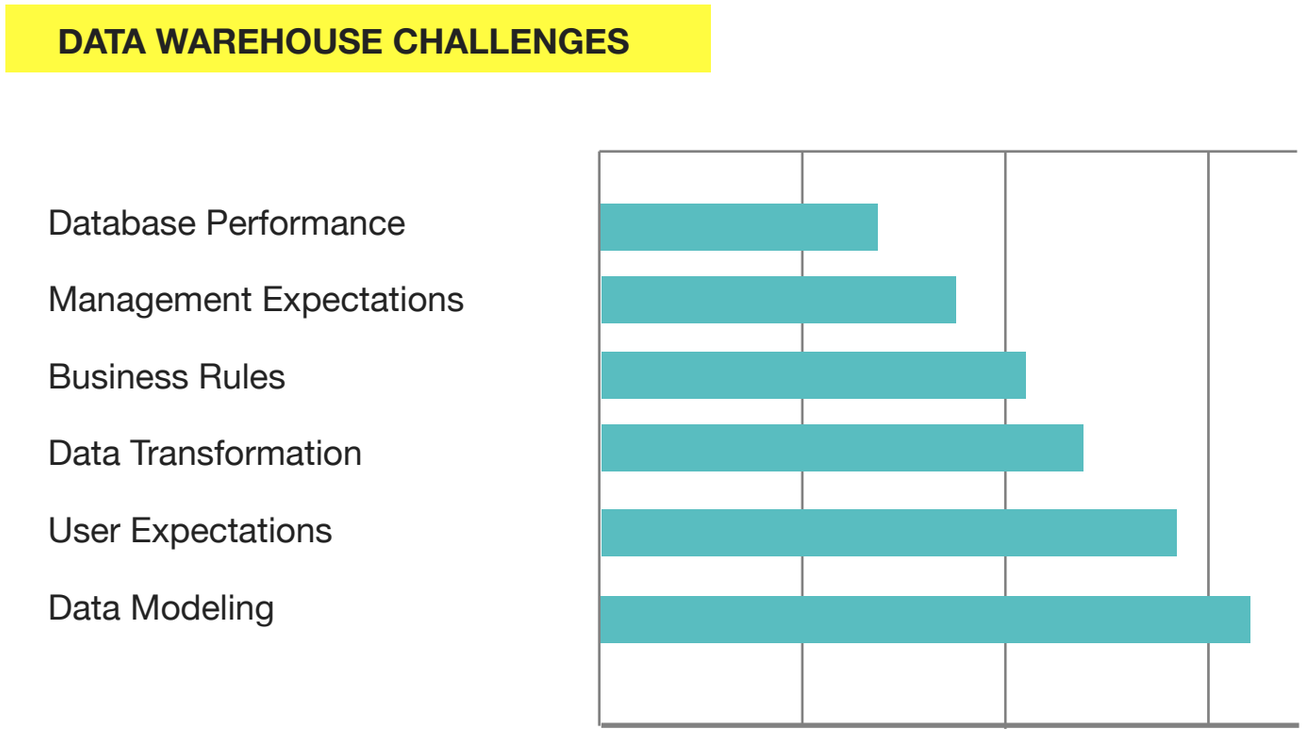
อุปสรรคและความท้าทายของการทำให้ข้อมูลมีคุณภาพ

อุปสรรคและความท้าทายของการทำให้ข้อมูลมีคุณภาพ

ในการสำรวจล่าสุดในหัวข้อเรื่องการทำธุรกิจกับคลังข้อมูล ได้มีการซักถามคำถามที่ว่า “อะไรคือสิ่งที่ท้าทายหรือเป็นอุปสรรคต่อการสร้างและการใช้คลังข้อมูล” คำตอบที่ได้จะแสดงดังรูปที่ 10-2 ซึ่งจะเป็นการจัดลำดับความท้าทายหรืออุปสรรคจากน้อยไปมาก



รูปที่ 10-2 ผลสำรวจอุปสรรคที่เกิดขึ้นกับการดำเนินการต่าง ๆ ระหว่างการสร้างคลังข้อมูล



รูปที่ 10-2 ผลสำรวจอุปสรรคที่เกิดขึ้นกับการดำเนินการต่าง ๆ ระหว่างการสร้างคลังข้อมูล

จากผลสำรวจจะเห็นได้ว่าคุณภาพของข้อมูลนั้นเป็นอุปสรรคชิ้นโตในการสร้างและใช้คลังข้อมูล ดังนั้นเราควรให้ความใส่ใจกับการปรับปรุงคุณภาพของข้อมูล เช่น การทำความสะอาดข้อมูล (Data cleansing) และวิธีการอื่น ๆ ในบางคลังข้อมูลอาจมีข้อมูลที่ความสกปรก (Dirty data) เก็บไว้ โดยที่ความสกปรกของข้อมูลหรือความไม่สมบูรณ์ของข้อมูลอาจไม่ได้เกิดจากคลังข้อมูล แต่เกิดจากความผิดพลาดของแหล่งข้อมูล ดังนั้นเราอาจต้องทำการกำจัดความสกปรกของข้อมูลจากแหล่งข้อมูล หรือจากระบบการดำเนินงานโดยที่ไม่ได้รับความช่วยเหลือใด ๆ จากระบบเก่าก็เป็นได้

หลักแหล่งที่มักจะมีข้อมูลที่ไม่มีคุณภาพ

ในการที่จะเลือกวิธีการหรือกลยุทธ์ที่ดีในการทำความสะอาดข้อมูล ถ้าเราทราบถึงประเภทหรือชนิดของลักษณะของระบบที่เกิดข้อมูลที่มีความสกปรกก่อน อาจช่วยให้เราสามารถเลือกวิธีที่เหมาะสมกับระบบที่เรากำลังสร้างขึ้นได้

System Conversions

ระบบสำหรับการสั่งซื้อเริ่มมาจากการใช้แฟ้มข้อมูล ซึ่งเริ่มขึ้นในปี 1970 ซึ่งข้อมูลการซื้อจะถูกเขียนลง flat files หรือ indexed file และไม่มีการตรวจสอบคลังสินค้าหรือการตรวจสอบเครดิตของลูกค้ามากนัก ต่อมาระบบได้ถูกเปลี่ยนให้เป็นระบบออนไลน์มากขึ้นซึ่งเราต้องเปลี่ยนจากระบบเดิม หรือการเก็บข้อมูลแบบเดิมมาเป็นการเก็บข้อมูลด้วย RDBMS เมื่อทำการเปลี่ยนอะไรจะเกิดขึ้นกับข้อมูลการสั่งซื้อของลูกค้า การเปลี่ยนแปลงระบบและการย้ายการใช้งานไปสู่ระบบ

Data aging

ข้อมูลก็มีความคล้ายกับมนุษย์ที่มีอายุขัย และสามารถเสื่อมสลายลงตามอายุ ซึ่งข้อมูลที่มีความเก่ามากก็จะสูญเสียความสำคัญและอาจไม่มีความหมายกับผู้ใช้ได้ ซึ่งถ้าข้อมูลมาจากระบบที่มีความเก่ามาก ๆ เราต้องให้ความใส่ใจกับอายุขัยของข้อมูลด้วย

Heterogeneous System Integration

ยิ่งข้อมูลที่นำเข้าสู่คลังข้อมูลมาจากหลายแหล่งข้อมูลหรือ แหล่งข้อมูลที่ต่างชนิดกันมากเท่าไหร่ ก็จะทำให้ยิ่งมีโอกาสในการที่จะมีข้อผิดพลาดหรือข้อมูลไม่มีคุณภาพมากขึ้นเท่านั้น ปัญหาความสอดคล้องของข้อมูลจะเป็นปัญหาที่มาจากระบบในลักษณะนี้ ดังนั้นเราควรจะต้องระมัดระวังให้มากกับข้อมูลที่นำมาใส่ใน dimension หรือ fact table ว่ามาจากระบบที่มีความแตกต่างกันมากน้อยเพียงใดหรือมาจากหลายแหล่งข้อมูล

Poor Database Design

การออกแบบฐานข้อมูลที่ดีจะสามารถลดความผิดพลาดเกิดขึ้นได้ ถ้าเราใช้ DBMS เราจะสามรถแก้ไขข้อมูลในฟิลด์ต่าง ๆ ได้ แต่ถ้าเราใช้ RDBMS เราจะสามารถตรวจสอบความสอดคล้องของข้อมูลกับกฎทางธุรกิจได้ผ่าน trigger ซึ่งจะสามารถประยุกต์ใช้กฎ “adhering to entity integrity” และ “referential integrity” ที่จะสามารถช่วยลดความผิดพลาดหรือความสกปรกของข้อมูลได้

Incomplete Information at Data Entry

ในการกรอกข้อมูลถ้าผู้กรอกข้อมูลทำการกรอกข้อมูลไม่ครบจะทำให้เกิดความผิดพลาดขึ้น 2 ชนิดด้วยกันคือ (1) อาจเกิดการขาดหายไปของข้อมูลเนื่องจากผู้ใช้ไม่กรอกข้อมูลบางฟิลด์ และ (2) ถ้าข้อมูลที่ไม่สามารถหาได้จำเป็นต่อการกรอกข้อมูล ผู้ใช้จะพยายามที่จะใช้ค่าทั่วไป (generic value) กับข้อมูลที่ต้องทำการกรอก เช่น การกรอกข้อมูล N/A สำหรับชื่อเมืองจะทำให้เกิดความผิดพลาดกับข้อมูลได้

Input Errors

การเกิดข้อผิดพลาดกับการกรอกข้อมูลในระบบข้อมูลแบบเก่า เมื่อเสมียนทำการกรอกข้อมูลเข้าสู่ระบบจะมีขั้นตอนการตรวจสอบความถูกต้องของข้อมูลโดยการให้เสมียนอีกคนทำการตรวจสอบข้อมูล แต่ในปัจจุบัน เมื่อพนักงานที่เกี่ยวข้องกับการทำธุรกรรมทางธุรกิจทำการกรอกข้อมูล การตรวจสอบข้อมูลจะทำการตรวจสอบทางสายตาเท่านั้น ซึ่งอาจทำให้เกิดความผิดพลาดได้

Internationalization/Localization

เนื่องจากการดำเนินธุรกิจมีการเปลี่ยนแปลงไป บางบริษัทได้ก้าวเข้าสู่การทำธุรกิจระหว่างประเทศซึ่งมีความหลากหลายทั้งทางภูมิศาสตร์และวัฒนธรรม ดังนั้นเมื่อบริษัทก้าวเข้าสู่การทำธุรกิจระหว่างประเทศจะทำให้ข้อมูลที่ทำการเก็บไว้อาจมีการเปลี่ยนแปลง โดยอาจมีค่าใหม่ ๆ เพิ่มขึ้น เช่น บริษัท NIKE ถ้าดำเนินธุรกิจในประเทศอเมริกาเพียงอย่างเดียวจะต้องทำการเก็บขนาดของรองเท้าตามหน่วยวัดของประเทศอเมริกา แต่เมื่อ NIKE ได้ดำเนินธุรกิจในประเทศญี่ปุ่นซึ่งใช้หน่วยเซนติเมตรในการวัดขนาดของรองเท้า จึงเป็นเหตุให้ระบบจะต้องทำการเก็บค่าใหม่ ๆ ตามมาตรวัดที่เปลี่ยนแปลงไป ในทำนองเดียวกันถ้าบริษัทดำเนินธุรกิจเฉพาะบางภูมิภาคของประเทศจะทำให้บางค่าของข้อมูลไม่ถูกใช้ได้ ซึ่งการเปลี่ยนแปลงโครงสร้างการทำธุรกิจของบริษัทจะส่งผลต่อการแก้ไขระบบดำเนินการซึ่งอาจก่อให้เกิดความผิดพลาดกับข้อมูลได้

Fraud

ไม่ต้องตกใจว่ามีความพยายามที่จะกรอกข้อมูลที่ไม่ถูกต้องเพราะมันเป็นเรื่องธรรมดา ซึ่งการกรอกข้อมูลที่ผิดจะเป็นการปลอมแปลงหรือบิดเบือนข้อมูล ตัวอย่างเช่น การกรอกข้อมูลรายได้ของลูกค้า ลูกค้าบางคนไม่ต้องการเปิดเผยถึงจำนวนที่แท้จริง จึงอาจทำให้มีการกรอกข้อมูลที่ไม่ถูกต้องได้ ดังนั้นเราอาจจะต้องมีมาตราการในการป้องกันในการกรอกข้อมูลที่ไม่ถูกต้องได้ เช่น กำหนดช่วงของเงินของรายได้ซึ่งจะทำให้ลูกค้ารู้สึกได้ว่าไม่ได้เปิดเผยถึงรายได้ที่แท้จริง ซึ่งการกรอกข้อมูลที่ไม่ถูกต้องจะส่งผลถึงคุณภาพของข้อมูลโดยตรง

Lack of Policies

การป้องกันความผิดพลาดของข้อมูลและการรักษาไว้ซึ่งคุณภาพของข้อมูลที่ถูกเก็บอยู่ในแหล่งข้อมูลเป็นเรื่องที่ต้องทำการใส่ใจและพิจารณาให้รอบคอบ ถ้าองค์กรใดไม่ได้ให้ความสนใจกับคุณภาพของข้อมูล องค์กรนั้นจะไม่สามารถมีข้อมูลที่มีคุณภาพในระดับที่ต้องการได้

การตรวจสอบข้อมูลชื่อและที่อยู่

บริษัทโดยส่วนใหญ่จะเผชิญความยุ่งยากกับความซ้ำซ้อนของข้อมูลชื่อและที่อยู่ สำหรับบุคคลหนึ่ง ๆ อาจมีข้อมูลถูกเก็บไว้หลายเรคคอร์ด เมื่อระบบทำการจัดเก็บข้อมูลไว้ในหลายแหล่งข้อมูลและไม่เว้นแม้แต่การเก็บข้อมูลไว้ในแหล่งข้อมูลเดียว แต่สำหรับคลังข้อมูลแล้วเราจะต้องทำการรวมข้อมูลชื่อและที่อยู่ของบุคคลเข้าไว้ด้วยกัน ซึ่งจากเดิมอาจมีการเก็บข้อมูลที่มีการซ้ำซ้อนอยู่ในหลายแหล่งข้อมูล ซึ่งการรวมข้อมูลเข้าด้วยกันอาจทำให้เกิดปัญหาขึ้นได้เมื่อเราต้องทำการจัดการกับข้อมูลที่มีความแตกต่างกัน เช่น ข้อมูลของลูกค้า พนักงาน ผู้ผลิตสินค้า และผู้ขายวัตถุดิบ

ตัวอย่างของปัญหาที่อาจเกิดขึ้นในการเก็บชื่อและที่อยู่ของบุคคล คือ บริษัทที่ทำการประมูลสินค้าที่มีลูกค้าหลายชนิดทำในกิจกรรมที่ต่าง ๆ กัน เช่น ลูกค้าบางรายอาจต้องทำการซื้อสินค้าจากการประมูล ลูกค้าบางรายอาจต้องการนำสินค้ามาทำการประมูล ลูกค้าบางรายอาจต้องการที่จะขอแค๊ตตาล็อกรายการสินค้าเพื่อทำการตัดสินใจก่อนการประมูล ลูกค้าบางรายอาจนำสินค้ามาประเมินราคา เนื่องจากบริษัทมีผู้เชี่ยวชาญในการประเมินราคาสินค้าและอื่น ๆ จากที่กล่าวมาข้างต้นระบบการดำเนินงานอาจต้องทำการเก็บข้อมูลของลูกค้าแต่ละประเภท โดยที่ลูกค้าบางคนอาจทำกิจกรรมหลาย ๆ อย่างทำให้ระบบอาจทำการเก็บข้อมูลชื่อและที่อยู่ของลูกค้าอย่างซ้ำซ้อนได้ ในกรณีที่บริษัทประมูลได้ขยายกิจการไปสู่การประมูลระดับนานาชาติ การกรอกชื่อและที่อยู่อาจะเกิดขึ้นในแต่ละประเทศที่ทำการประมูล ซึ่งเป็นการทำให้ข้อมูลเกิดความซ้ำซ้อนอย่างมาก และอาจทำให้คุณภาพของข้อมูลต่ำ

การเก็บชื่อและที่อยู่สามารถทำได้ 2 วิธีดังแสดงในรูปที่ 10-3 ถ้าชื่อและที่อยู่มีการเก็บข้อมูลโดยใช้หลาย ๆ ฟิลด์ จะทำให้ง่ายในการตรวจสอบความซ้ำซ้อนตอนกรอกข้อมูล แต่อย่างไรก็ดีการกรอกข้อมูลชื่อและที่อยู่ก็ยังคงสร้างปัญหาให้กับระบบได้ดังนี้

SINGLE FIELD FORMAT

Name & Address:	Dr. Jay A. Harreld, P.O. Box 999, 100 Main Street, Anytown, NX 12345, U.S.A.

MULTIPLE FIELD FORMAT

Title:	Dr.
First Name:	Jay
Middle Initial:	A.
Last Name:	Harreld
Street Address-1:	P.O. Box 999
Street Address-2:	100 Main Street
City:	Anytown
State:	NX
Zip:	12345
Country Code:	U.S.A.

รูปที่ 10-3 รูปแบบการกำหนดชื่อและที่อยู่ของพนักงานหรือลูกค้า

- ไม่มีคีย์ที่ไม่ซ้ำกัน
  (No unique key)
- มีหลายชื่อในบรรทัดเดียวกัน
  (Many names on one line)
- มีชื่อเดียวในสองบรรทัด
  (One name in two lines)
- ชื่อและที่อยู่ถูกกรอกในบรรทัดเดียวกัน
  (Name and the address in a single line)
- มีการกรอกข้อมูลผสมกันระหว่างชื่อและชื่อบริษัท
  (Personal and company names mixed)
- คนหนึ่งคนมีหลายที่อยู่ซึ่งแตกต่างกัน
  (Different addresses for the same person)
- ลูกค้าคนหนึ่งๆ มีหลายชื่อ และอาจมีการสะกดที่แตกต่างกัน
  (Different names and spelling for the same customer)

ในการที่จะทำการลดความซ้ำซ้อนของการเก็บข้อมูล ชื่อและที่อยู่ของบุคคลในระบบ เราอาจต้องทำการจัดเก็บข้อมูลโดยใช้การเก็บแบบหลายฟิลด์ (จะต้องทำในกรณีที่ข้อมูลถูกเก็บไว้ในฟิลด์เดียว)

จากนั้นทำการออกแบบอัลกอริทึมสำหรับตรวจสอบความเหมือนกันของข้อมูลและทำการหาแหล่งของข้อมูลที่ซ้ำกัน เพื่อลบข้อมูลที่ซ้ำกันออก

ค่าใช้จ่ายในการมีข้อมูลที่ไม่มีคุณภาพ

ในการที่จะทำความสะอาดข้อมูลและการปรับปรุงข้อมูลเราจะต้องเสียค่าใช้จ่ายและความพยายาม ถึงแม้ว่าคุณภาพของข้อมูลจะเป็นสิ่งสำคัญ แต่ก่อนที่จะเริ่มทำความสะอาดข้อมูลเราจะต้องทำการประเมินค่าใช้จ่ายที่ต้องใช้รวมถึงพยายามทราบถึงข้อมูลใดที่จำเป็นต้องมีความถูกต้องหรือมีคุณภาพเป็นอย่างมาก ในการประเมินเราอาจขอความช่วยเหลือจากผู้ใช้ระบบซึ่งจะทราบถึงการสูญเสียโอกาสและความน่าจะเป็นที่จะตัดสินใจ ซึ่งในการประเมินถึงความจำเป็นในการทำความสะอาดข้อมูลจะต้องพิจารณาสิ่งเหล่านี้

- การตัดสินใจที่ไม่ดีจากการวิเคราะห์งานประจำ (Bad decisions based on routine analysis)
- การเสียโอกาสทางธุรกิจเนื่องจากข้อมูลไม่พร้อมใช้งานหรือข้อมูลสกปรก (Lost business opportunities because of unavailable or “Dirty” data)
- ค่าปรับจากทางรัฐบาลหรือหน่วยงานราชการสำหรับการไม่ยินยอมทำงานหรือฝ่าฝืนกฎข้อบังคับ (Fines from governmental agencies for noncompliance or violation of regulatios)
- ความละเอียดของการตรวจสอบปัญหา (Resolution of audit problems)
- ความซ้ำซ้อนของข้อมูลที่ไม่จำเป็นต่อการทำงาน (Redundant data unnecessarily for business routine)
- รายงานที่ไม่สอดคล้องกัน (Inconsistent reports)
- เวลาและความพยายามในการแก้ไขข้อมูลให้ถูกต้องในทุก ๆ ครั้งที่มีการพบข้อมูลที่ไม่มีความผิดพลาด (Time and effort for correcting data every time data corruption is discovered)

SECTION 5

เครื่องมือสำหรับการปรับปรุงคุณภาพของข้อมูล

เครื่องมือสำหรับการปรับปรุงคุณภาพของข้อมูล

ในปัจจุบันได้มีเครื่องมือสำหรับทำความสะอาดข้อมูลวางจำหน่าย เราสามารถประยุกต์ใช้เครื่องมือนั้น ๆ ทำความสะอาดข้อมูลที่ถูกเก็บอยู่ในแหล่งข้อมูลได้ รวมถึงสามารถนำไปประยุกต์ใช้ใน staging area กล่าวคือ หลังจากทำการสกัดข้อมูลแล้วเราจะประยุกต์ใช้เครื่องมือ ในการทำความสะอาดข้อมูลก่อนที่จะทำการถ่ายโอนข้อมูลเข้าสู่คลังข้อมูล โดยที่ในปัจจุบันผู้พัฒนาเครื่องมือในการทำความสะอาดข้อมูลได้ผลิตชุดเครื่องมือที่สามารถรวมเข้ากับเครื่องมือสำหรับการทำอีทีแอลแล้ว

เครื่องมือสำหรับการทำความสะอาดข้อมูล

การใช้เครื่องมือช่วยในการทำความสะอาดข้อมูลโดยส่วนใหญ่แล้วจะมีการทำงานหลัก ๆ 2 ฟังก์ชันด้วยกันคือ

(1) ช่วยในการระบุถึงข้อมูลที่ไม่ถูกต้องและข้อมูลที่ไม่สอดคล้องกัน (Data error discovery tools)

(2) ช่วยทำการแก้ไขข้อมูลที่มีความผิดพลาดได้อีกด้วย (Data correction tools)

ซึ่งเครื่องมือดังกล่าวจะใช้อัลกอริทึม การแจงส่วน (parse) การเปลี่ยนรูป (transform) การเปรียบเทียบ (match) การรวมเข้าด้วยกัน (Consolidate) และการแก้ไขข้อมูลให้ถูกต้อง (Correct the data)

Error Discovery Features

เป็นครื่องมือที่ใช้ในการค้นหาความผิดพลาดของข้อมูลจะประกอบไปด้วยฟังก์ชันการทำงานดังต่อไปนี้

- ระบุเรคคอร์ดที่ซ้ำกันได้อย่างรวดเร็วและง่ายดาย
  (Quickly and easily identify duplicate records)
- ระบุถึงข้อมูลที่มีค่าที่อยู่นอกช่วงของค่าโดเมนที่กำหนด
  (Identify data items whose values are outside the range of legal domain values)
- ค้นหาข้อมูลที่ไม่สอดคล้องกัน
  (Find inconsistent data)
- ตรวจสอบขอบเขตของข้อมูล
  (Check for range of allowable values)
- ตรวจจับข้อมูลที่ไม่สอดคล้องกันจากแหล่งข้อมูลที่แตกต่างกัน
  (Detect inconsistencies among data items from different sources)
- อนุญาตให้ผู้ใช้ระบุและบอกปริมาณปัญหาของคุณภาพของข้อมูล
  (Allow users to identify and quantify data quality problems)

Error Discovery Features (ต่อ)

- ตรวจสอบแนวโน้มคุณภาพของข้อมูลในช่วงเวลาต่างๆ
  (Monitor trends in data quality over time)
- รายงานคุณภาพของข้อมูลที่ใช้ในการวิเคราะห์ไปยังผู้ใช้
  (Report to users on the quality of data used for analysis)
- ตรวจสอบปัญหาเกี่ยวกับ RDBMS referential integrity
  (Reconcile problems of RDBMS referential integrity)

Data Correction Features

เป็นเครื่องมือที่ใช้ในการแก้ไขข้อมูลให้ถูกต้องจะต้องประกอบไปด้วยฟังก์ชันการทำงานดังต่อไปนี้

- ทำให้ข้อมูลที่ไม่สอดคล้องกันกลายเป็นปกติ
  (Normalize inconsistent data)
- ปรับปรุงการรวมข้อมูลจากแหล่งข้อมูลที่ไม่เหมือนกัน
  (Improve merging of data from dissimilar data sources)
- ทำการรวมกลุ่มและสร้างความสัมพันธ์ให้กับข้อมูลลูกค้าที่อยู่ในครอบครัวเดียวกัน
  (Group and relate customer records belonging to the same household)
- จัดเตรียมการวัด/ประเมินคุณภาพของข้อมูล
  (Provide measurements of data quality)
- ทำข้อมูลให้อยู่ในรูปแบบที่เป็นมาตราฐาน
  (Standardize data elements to common formats)
- ตรวจสอบค่าที่อนุญาต
  (Validate for allowable values)

The DBMS for Quality Control

เราสามารถใช้ระบบการจัดการฐานข้อมูลเพื่อควบคุมคุณภาพของข้อมูลได้ โดยที่ RDBMS จะมีหลายคุณลักษณะพิเศษที่สามารถช่วยป้องกันการเกิดข้อผิดพลาดกับข้อมูลที่ค่อยเกิดขึ้นกับคลังข้อมูลดังนี้

Domain Integrity — อนุญาตให้มีการแก้ไขขอบเขตของข้อมูล และทำการป้องกันการกรอกข้อมูลออกนอกขอบเขตที่กำหนดไว้ โดยการใช้ดาต้าดิกชันนารี

Update Security — เป็นการป้องกันการทำงานที่ไม่มีอำนาจมาอัพเดทข้อมูลในฐานข้อมูล

Entity Integrity Checking — เป็นการทำให้แน่ใจว่าข้อมูลเรคคอร์ดที่ซ้ำกัน และมีคีย์หลักเหมือนกันจะไม่ถูกเพิ่มเข้ามาในฐานข้อมูล

Minimize missing values — เป็นการทำให้แน่ใจว่าจะไม่อนุญาตให้มี NULL ในฟิลด์ที่จำเป็น

Referential Integrity Checking — เป็นการทำให้แน่ใจว่าความสัมพันธ์ของ foreign key นั้นยังคงมีอยู่ และทำการป้องกันการลบของแถวแม่ที่เกี่ยวข้อง

Conformance to Business Rules — ใช้โปรแกรม trigger และขั้นตอนการจัดเก็บข้อมูล (Stored procedures) ในการบังคับให้ข้อมูลเป็นไปตามกฎทางธุรกิจ ซึ่ง trigger จะถูกเก็บอยู่ในฐานข้อมูลอยู่แล้ว โปรแกรม trigger จะทำงานอัตโนมัติเมื่อข้อมูลที่กำหนดมีการอัพเดทหรือถูกลบ ในขณะที่ขั้นตอนในการจัดเก็บข้อมูลอาจถูกออกแบบสำหรับการบังคับทำให้ข้อมูลที่ถูกกรอกเข้ามามีความสอดคล้องกับกฎทางธุรกิจ ซึ่งขั้นตอนในการจัดเก็บข้อมูลอาจถูกเรียกว่า application program

SECTION 6

การปรับปรุงคุณภาพของข้อมูล

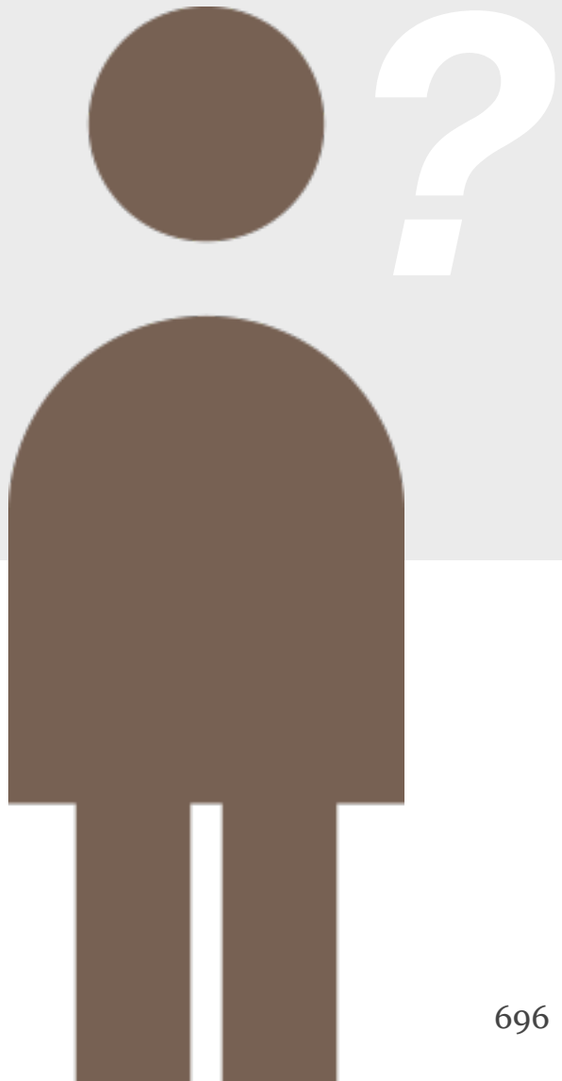
การปรับปรุงคุณภาพของข้อมูล

แม้ว่าคุณภาพของข้อมูลจะมีความสำคัญมาก แต่หลายๆ บริษัทยังคงมีคำถามว่าควรจะเอาใจใส่กับคุณภาพของข้อมูลหรือการทำความสะอาดข้อมูลหรือไม่ ในหลายๆกรณีข้อมูลที่ขาดหายไปในบางแอทริบิวก็ไม่สามารถสร้างขึ้นใหม่ได้หรือบางข้อมูลก็มีความซับซ้อนมากเกินกว่าที่จะทำความสะอาดได้ จากเหตุการณ์ที่เกิดขึ้นจึงมีคำถามตามมามากมาย เช่น

เราควรทำความสะอาดข้อมูลหรือไม่?

เราต้องทำความสะอาดข้อมูลเป็นจำนวนเท่าใด?

ส่วนใดของข้อมูลที่จะมีความสำคัญสูงที่จะทำความสะอาดข้อมูล?



จากคำถามที่เกิดขึ้นอาจทำให้บริษัทต่าง ๆ ไม่สนใจหรือต่อต้านที่จะทำความสะอาดข้อมูล ซึ่งการละเลยในการทำความสะอาดข้อมูลอาจเกิดจากปัจจัยดังต่อไปนี้

- การทำความสะอาดข้อมูลเป็นเรื่องที่น่าเบื่อหน่ายและเสียเวลา
- เมตาดาต้าจากหลาย ๆ แหล่งข้อมูลอาจสูญหายไปหรือไม่ได้มีการเก็บข้อมูลไว้
- ผู้ใช้ที่ต้องการข้อมูลที่มีคุณภาพมีหน้าที่อื่น ๆ ที่ต้องรับผิดชอบมากมาย ซึ่งทำให้การทำให้คุณภาพอยู่ในความสนใจลำดับท้าย ๆ
- ในบางครั้งการทำความสะอาดข้อมูลนั้นเป็นกิจกรรมที่ยิ่งใหญ่มาก มีการทำงานที่ซ้ำซ้อน ซึ่งเป็นเหตุให้บริษัทต่าง ๆ เกิดความหวาดกลัวที่จะเริ่มต้นการทำความสะอาดข้อมูล

ในการสร้างการทำความสะอาดข้อมูล คุณอาจเลือกกลยุทธ์ใดกลยุทธ์หนึ่งจาก 2 กลยุทธ์ดังต่อไปนี้

1

คุณอาจเลือกที่จะความสะอาดเฉพาะข้อมูลที่จะทำการถ่ายโอนเข้าสู่คลังข้อมูล ซึ่งจะทำให้ข้อมูลที่มีคุณภาพ 100% เท่านั้นที่จะถูกถ่ายโอนเข้าสู่คลังข้อมูล ข้อมูลที่มีความผิดพลาดจะต้องถูกแก้ไขก่อนที่จะทำการโหลด ซึ่งการทำงานจะเริ่มจากการตรวจจับความผิดพลาดของข้อมูล แล้วทำการแก้ไขข้อมูล ซึ่งการทำงานในลักษณะนี้จะทำให้ใช้เวลานานในการทำความสะอาดข้อมูลก่อนที่จะถ่ายโอนข้อมูลเข้าสู่คลังข้อมูล

2

“Clean as you go” จะทำการโหลดข้อมูลที่เป็น “as is” (อ่านบท อีทีแอล) เข้าสู่คลังข้อมูล จากนั้นทำความสะอาดข้อมูลในหลังข้อมูลในภายหลัง ข้อมูลที่ถูกโหลดเข้าสู่คลังข้อมูลแล้วแต่ยังไม่ได้ทำความสะอาดจะเป็นข้อมูลที่ยังไม่มีความน่าเชื่อถือ เราจะต้องรอให้ข้อมูลนั้น ๆ ถูกทำความสะอาดเสียก่อน วิธีการนี้จะทำให้คุณภาพของข้อมูลนะช่วงเวลาหนึ่งๆ ขาดความน่าเชื่อถือ ซึ่งความน่าเชื่อถือนั้นถือว่าเป็นสิ่งสำคัญมาก ๆ สำหรับการใช้คลังข้อมูล

กระบวนการตัดสินใจ

ในการดำเนินการทำความสะอาดข้อมูล

กระบวนการตัดสินใจในการดำเนินการทำความสะอาดข้อมูล

ก่อนที่เริ่มทำความสะอาดข้อมูล ผู้พัฒนาคลังข้อมูลและผู้ใช้คลังข้อมูลควรจะต้องรู้ถึงการตัดสินเบื้องต้นก่อนว่ามีอะไรบ้าง? จากนั้นเราต้องทำการค้นหาว่าข้อมูลใดส่งผลต่อการตัดสินใจบ้าง? อย่างที่เราทราบกันดีว่าเราไม่สามารถทำความสะอาดข้อมูลทั้งหมดได้

เนื่องจากข้อจำกัดทางด้านเวลาและงบประมาณ ดังนั้น เราจะทำความสะอาดข้อมูลที่จำเป็นบางส่วนเท่านั้น ตัวอย่างเช่น ถ้าข้อมูลจากคลังข้อมูลจะต้องสร้างรายงานเกี่ยวกับลูกค้า 25 คนที่มีการซื้อสินค้าสูงที่สุด ข้อมูลการขายจะต้องมีคุณภาพสูงหรือถ้าข้อมูลสถิติประชาการจะถูกใช้สำหรับการคิดค้นแคมเปญใหม่ ๆ คุณภาพของข้อมูลสถิติประชากรอาจจำเป็นต้องมีรายละเอียดมาก

ข้อมูลใดที่ควรจะถูกทำความสะอาด?

ในการเลือกว่าข้อมูลใดที่ควรจะต้องทำความสะอาดนั้น เราจะต้องรู้ว่ารายงานหรือคำถามที่ต้องการถามจากคลังข้อมูลมีอะไรบ้าง จากนั้นการหาแหล่งข้อมูลเพื่อตอบคำถามเหล่านั้น นอกจากนี้เราต้องทำการประเมินถึงประโยชน์ที่คาดว่าจะได้รับจากการทำความสะอาดข้อมูลแต่ละชิ้น และประเมินว่าถ้าเราทำความสะอาดข้อมูลหรือปล่อยให้ข้อมูลมีความสกปรกอยู่ จะมีผลกระทบต่อการวิเคราะห์ข้อมูลจากคลังข้อมูลหรือไม่ ในการประเมินต่างๆ เราจะให้ผู้ใช้เป็นคนตัดสินใจ โดยที่ผู้สร้างคลังข้อมูลจะช่วยเหลือเล็กน้อยเท่านั้น

เราควรทำความสะอาดข้อมูลที่ใด ?

ข้อมูลในคลังข้อมูลจะมาจากระบบการดำเนินงานซึ่ง
ข้อมูลอาจจะมีความผิดพลาดอยู่บ้าง การเก็บข้อมูลเข้าสู่คลังข้อมูล
จะเริ่มจากการสกัดข้อมูลแล้วทำการเก็บไว้ใน staging area
จากนั้นทำการถ่ายโอนข้อมูลจาก staging area เข้าสู่คลังข้อมูล
จากขั้นตอนการดำเนินงานที่กล่าวมาข้างต้นจะมีอยู่ 3 พื้นที่ที่จะมีการ
จัดเก็บข้อมูล คือ **แหล่งข้อมูล staging area และคลังข้อมูล**
เราสามารถเลือกทำความสะอาดที่ใดที่หนึ่งก็ได้ หรือจะทำความสะอาด 2 ใน 3
หรือทำความสะอาดในทุกที่ที่มีข้อมูลเลยก็ได้

การทำความสะอาดข้อมูลในคลังข้อมูลอาจไม่มีประสิทธิภาพมากนักเมื่อเทียบกับการทำความสะอาดในที่อื่น ๆ เนื่องจากอาจประสบปัญหาการมีฟังก์ชันการทำงานที่มากเกินไปสำหรับการเคลื่อนย้ายข้อมูล และการถ่ายโอนข้อมูลเข้าสู่คลังข้อมูล ดังนั้นในทางปฏิบัติเราจะทำความสะอาดข้อมูลก่อนที่จะทำการโอนถ่ายข้อมูลเข้าสู่คลังข้อมูล การทำความสะอาดข้อมูลอาจทำจาก staging area หรือจากแหล่งข้อมูลจะเป็นสองทางเลือกที่จะดีกว่าการทำความสะอาดข้อมูลในคลังข้อมูล

การทำความสะอาดข้อมูลใน staging area น่าเป็นสิ่งที่จะดีที่สุด เนื่องจากเป็นการทำงานหลังจากทำการสกัดข้อมูลแล้ว ซึ่งจะมีข้อมูลที่เหมาะกับคลังข้อมูลเท่านั้น แต่อย่างไรก็ตามการทำความสะอาดข้อมูลใน staging area ยังมีข้อด้อยอยู่ที่ข้อมูลในแหล่งข้อมูลจะยังคงมีความผิดพลาดอยู่ และยังไม่ได้รับการแก้ไข นอกจากนี้จะมีความแตกต่างของรายงานที่สร้างขึ้นจากข้อมูลเหมือนกันซึ่งถูกเก็บไว้ในแหล่งข้อมูลและคลังข้อมูล จากความแตกต่างที่เกิดขึ้นจะทำให้ระบบขาดความน่าเชื่อถือได้

การทำความสะอาดข้อมูลในแหล่งข้อมูล อาจเป็นการขจัดปัญหาต่าง ๆ ที่ได้กล่าวมาข้างต้น แต่อย่างไรก็ดีเราอาจจะต้องพบกับความยุ่งยากและอุปสรรคมากมายในการทำงาน เมื่อบางแหล่งข้อมูลหรือบางระบบไม่มีเอกสารคู่มือการทำงานที่ดีพอ รวมถึงในบางระบบอาจไม่มี source code ให้เลยก็เป็นได้

เราจะทำความสะอาดข้อมูลได้อย่างไร ?

นี่คือคำถามเมื่อคุณตัดสินใจที่จะใช้เครื่องมือในการทำความสะอาดข้อมูล ซึ่งจะมีคำถามอื่น ๆ ตามมาอีกมากมาย เช่น คุณใช้เครื่องมือในการทำความสะอาดข้อมูลเพียงอย่างเดียวใช่หรือไม่ ? คุณทำการรวมชุดเครื่องมือสำหรับทำความสะอาดข้อมูลเข้ากับเครื่องมือสำหรับอีทีแอลหรือไม่ ? ถ้าไม่ คุณจะต้องทำการสร้างโปรแกรมเองมากน้อยแค่ไหน จากคำถามที่กล่าวข้างต้น ถ้าเราเลือกใช้เครื่องมือในการทำความสะอาดข้อมูล เราต้องตรวจสอบว่าระบบที่เราสร้างขึ้นนั้นสามารถรองรับชุดเครื่องมือเหล่านี้ได้หรือไม่ ถ้าระบบของเราไม่สามารถรองรับข้อมูลเหล่านี้ได้เราจำเป็นต้องสร้างขึ้นเอง

เราจะสามารถค้นพบขอบเขตของความผิดพลาดของข้อมูลได้อย่างไร ?

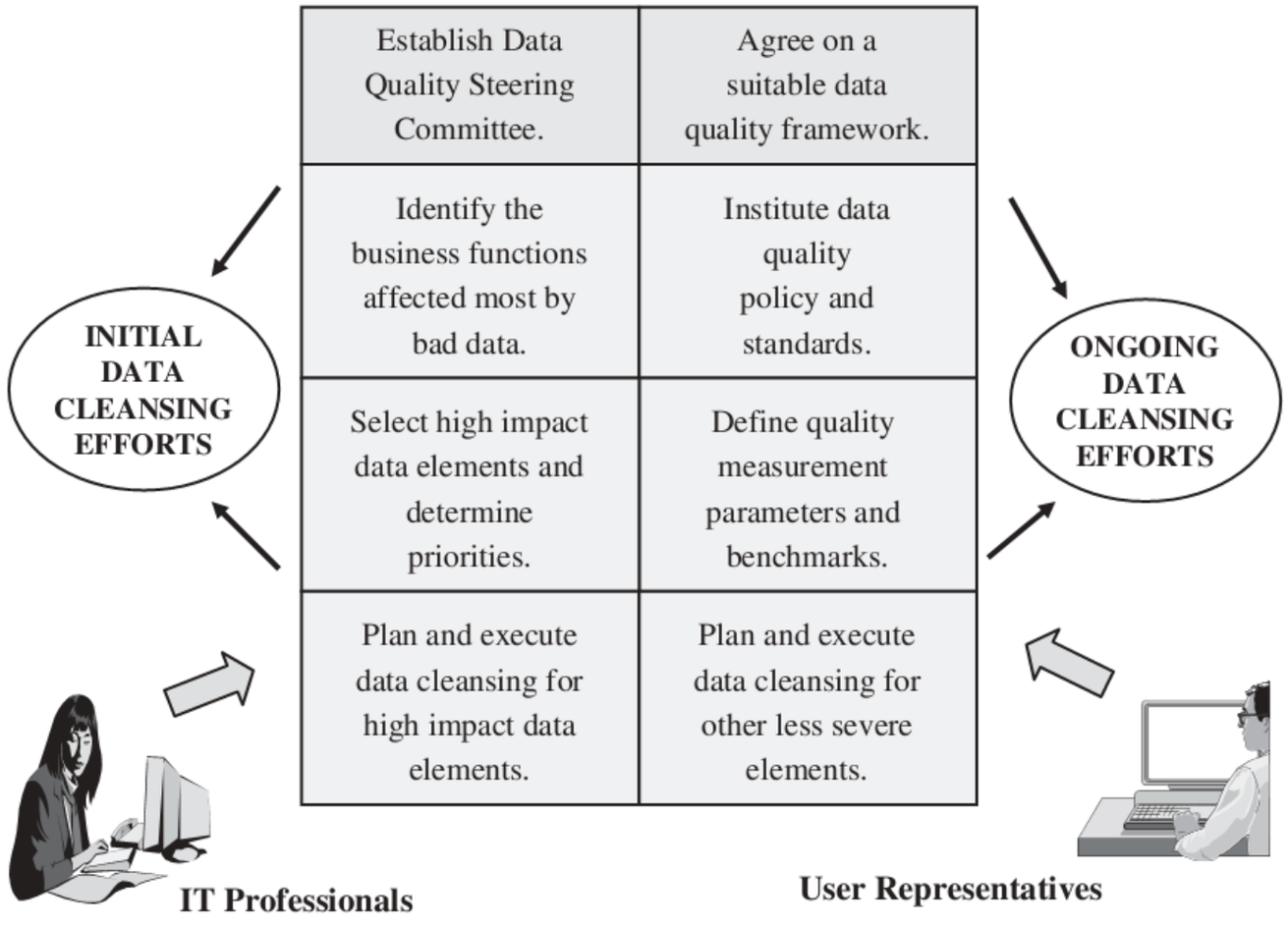
ก่อนที่จะใช้เทคนิคต่าง ๆ ในการทำความสะอาดข้อมูล เราจะต้องประเมินขอบเขตของความผิดพลาดหรือคุณภาพของข้อมูลเสียก่อน โดยทำลิสต์ที่สะท้อนถึงความผิดพลาดของระบบการดำเนินงาน จากนั้นทำการประเมินขอบเขตของความผิดพลาด โดยพิจารณาถึงความผิดพลาดของระบบที่ได้ลิสต์ไว้ รูปที่ 10-4 แสดงถึงวิธีการในการตรวจสอบขอบเขตของความผิดพลาด เราสามารถใช้วิธีเหล่านี้เข้ากับคลังข้อมูลที่เราสร้างขึ้น

- Operational systems converted from older versions are prone to the perpetuation of errors.
- Operational systems brought in house from outsourcing companies converted from their proprietary software may have missing data.
- Data from outside sources that is not verified and audited may have potential problems.
- When applications are consolidated because of corporate mergers and acquisitions, these may be error-prone because of time pressures.
- When reports from old legacy systems are no longer used, that could be because of erroneous data reported.
- If users do not trust certain reports fully, there may be room for suspicion because of bad data.
- Whenever certain data elements or definitions are confusing to the users, these may be suspect.
- If each department has its own copies of standard data such as Customer or Product, it is likely corrupt data exists in these files.
- If reports containing the same data reformatted differently do not match, data quality is suspect.
- Wherever users perform too much manual reconciliation, it may be because of poor data quality.
- If production programs frequently fail on data exceptions, large parts of the data in those systems are likely to be corrupt.
- Wherever users are not able to get consolidated reports, it is possible that data is not integrated.

รูปที่ 10-4 ขอบเขตของการผิดพลาดของข้อมูล

การสร้างเฟรมเวิร์คสำหรับการปรับปรุงคุณภาพของข้อมูล

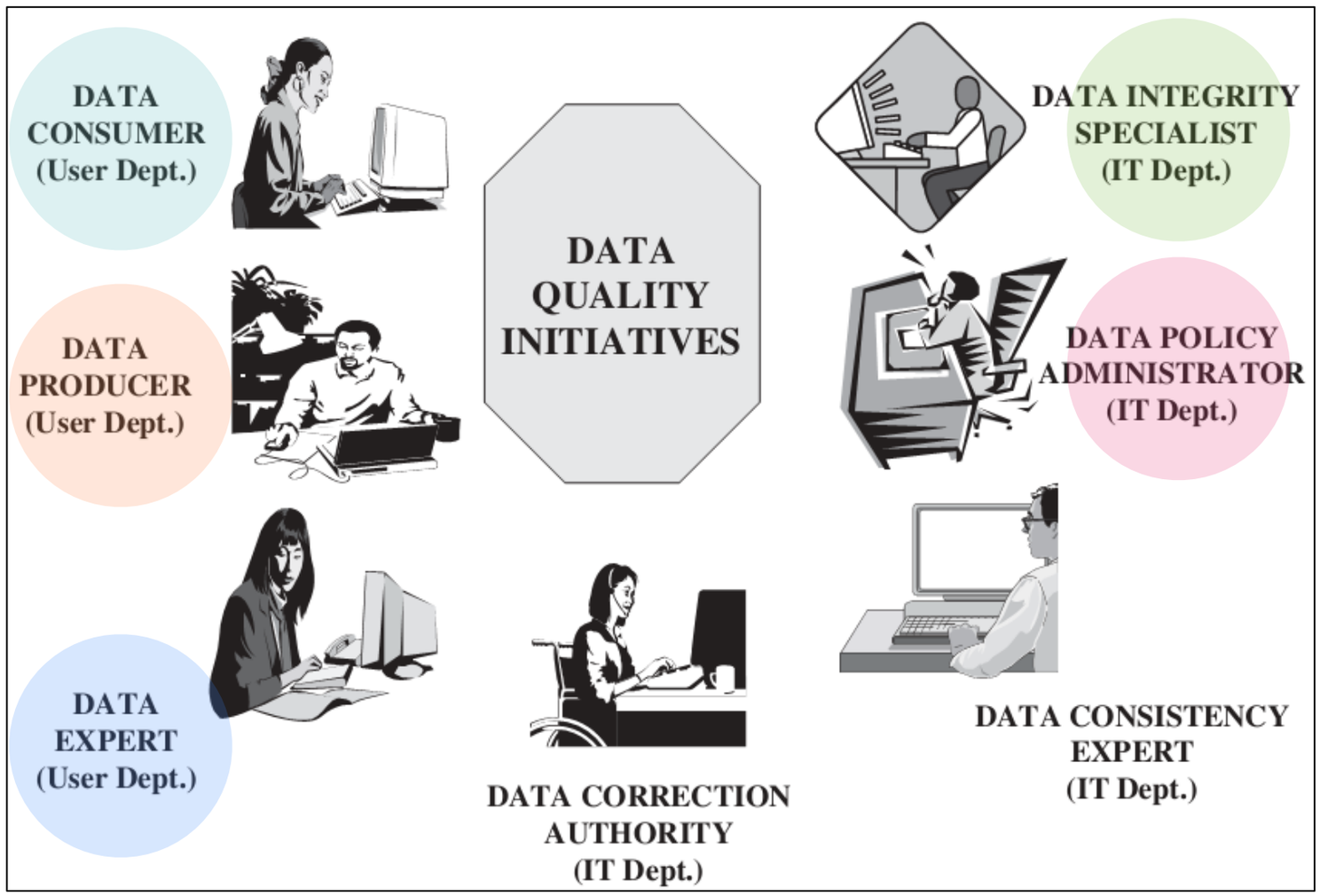
ขั้นตอนการสร้างเฟรมเวิร์คสำหรับการปรับปรุงคุณภาพของข้อมูลจะแสดงดังรูปที่ 10-5



รูปที่ 10-5 โครงสร้างของการปรับปรุงคุณภาพข้อมูล

ผู้ที่มีหน้าที่รับผิดชอบในการปรับปรุงคุณภาพของข้อมูลคือใคร ?

รูปที่ 10-6 แสดงถึงผู้ที่มีส่วนร่วมกับการปรับปรุงคุณภาพของข้อมูล
ซึ่งมีหน้าที่ที่แตกต่างกันดังนี้



รูปที่ 10-6 ผู้ที่มีส่วนเกี่ยวข้องกับการปรับปรุงคุณภาพข้อมูล

Data Consumer

เป็นคนที่ใช้คลังข้อมูลสำหรับการประมวลผลคิวรี การทำรายงาน และการวิเคราะห์ข้อมูล ซึ่งจะเป็นคนกำหนดระดับของคุณภาพของข้อมูลที่ยอมรับได้

Data Producer

รับหน้าที่การพิจารณาคุณภาพของข้อมูลที่ได้จากแหล่งข้อมูล

Data Expert

เป็นผู้เชี่ยวชาญในเกี่ยวกับข้อมูลในแหล่งข้อมูล รับหน้าที่ในการระบุความผิดพลาดของแหล่งข้อมูล

Data Policy Administrator

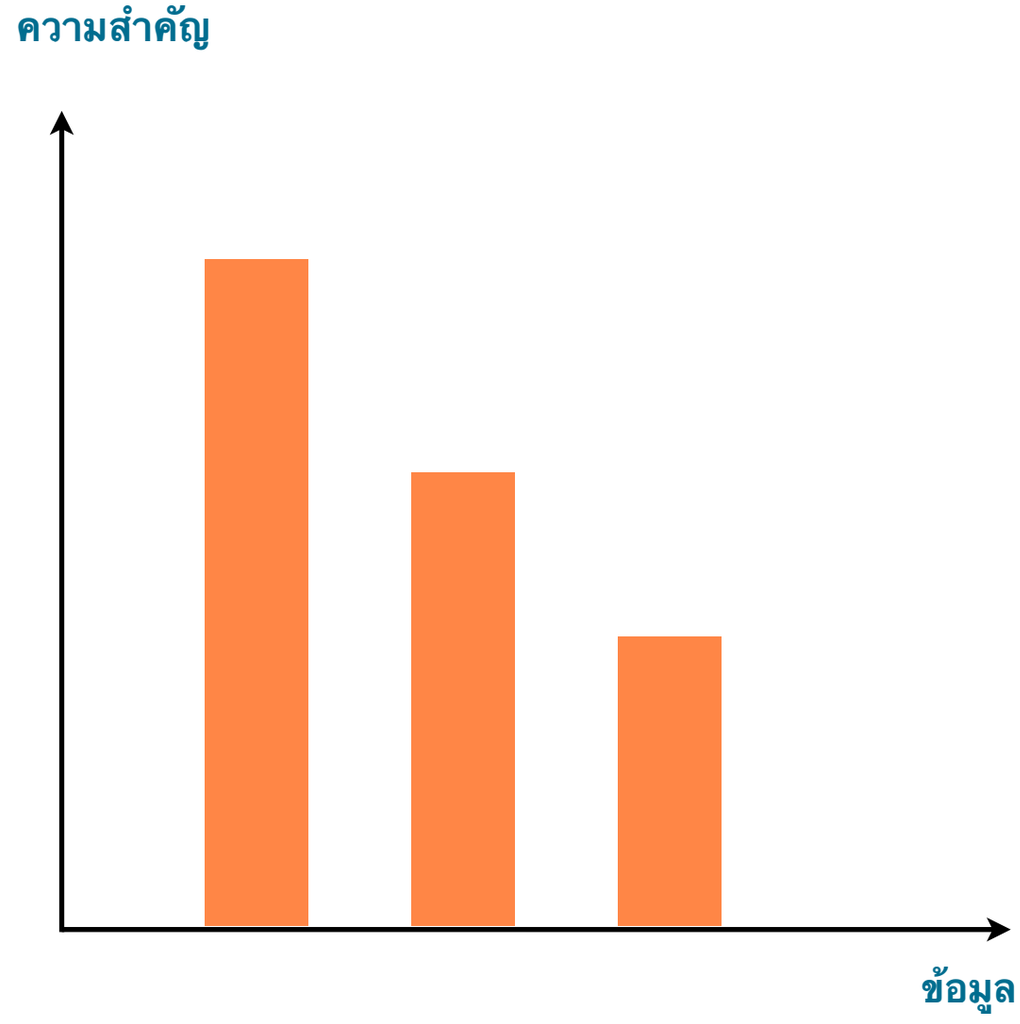
รับผิดชอบเกี่ยวกับการแก้ไขความผิดพลาดของข้อมูล การเปลี่ยนรูปข้อมูล และการถ่ายโอนข้อมูลเข้าสู่คลังข้อมูล

Data Integrity Specialist

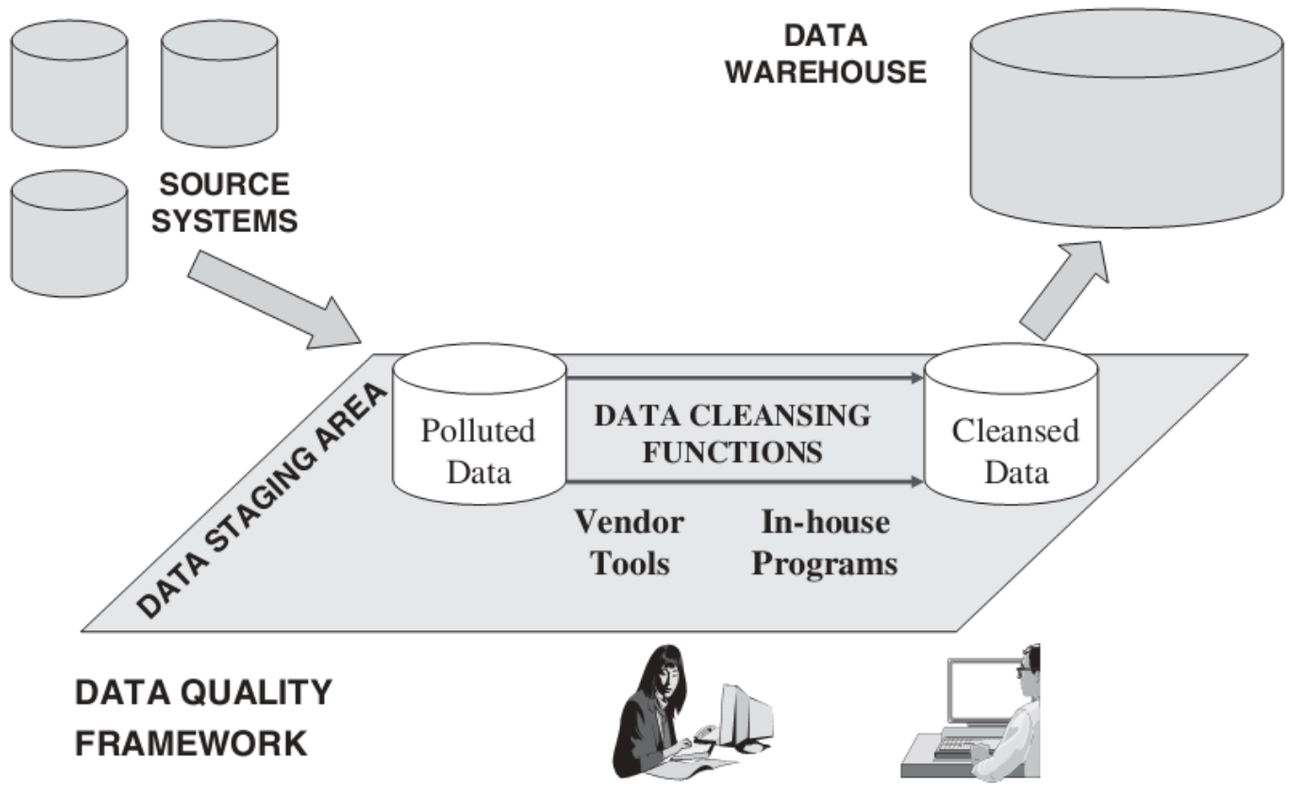
ทำหน้าที่เกี่ยวกับการตรวจสอบความสอดคล้องของข้อมูลจากแหล่งข้อมูลกับกฎทางธุรกิจ

กระบวนการปรับปรุงคุณภาพของข้อมูล

คุณจะสามารถทำการปรับปรุงคุณภาพของข้อมูลได้อย่างไร? ในการปรับปรุงคุณภาพของข้อมูล เราควรจัดลำดับความสำคัญของข้อมูล โดยอาจแบ่งเป็น 3 ระดับ คือ สูง กลาง และต่ำ ในกลุ่มของความสำคัญระดับสูง ข้อมูลอาจต้องมีคุณภาพถึง 100% แต่ในระดับกลางอาจต้องการข้อมูลที่มีคุณภาพมากที่สุดเท่าที่จะเป็นไปได้ และระดับต่ำข้อมูลอาจถูกทำความสะอาด ถ้าเรามีเวลาและทรัพยากรเพียงพอ จากการแบ่งลำดับความสำคัญของข้อมูลทำให้เราสามารถเริ่มการทำความสะอาดข้อมูลที่มีความสำคัญระดับสูง จากนั้นค่อยเปลี่ยนมาทำความสะอาดข้อมูลที่มีความสำคัญระดับกลางต่อไป



ปัญหาที่มักจะพบบ่อยเกี่ยวกับคุณภาพของข้อมูล คือ การซ้ำกันของข้อมูล ซึ่งมักจะเกิดกับข้อมูลของบุคคล เช่น ลูกค้า พนักงาน และผู้ร่วมลงทุน จากปัญหาที่เกิดขึ้นเราต้องตรวจสอบให้แน่ใจว่าเทคนิคต่าง ๆ จะสามารถระบุถึงเรคคอร์ดที่มีการซ้ำกัน และทำการเปลี่ยนการเชื่อมโยงกับเรคคอร์ดอื่น ๆ ที่เกี่ยวข้องกัน รูปที่ 10-7 แสดงถึงขั้นตอนในการปรับปรุงคุณภาพของข้อมูล ซึ่งประกอบไปด้วยขั้นตอนย่อยดังต่อไปนี้



รูปที่ 10-7 ภาพรวมของการปรับปรุงคุณภาพข้อมูล

- กำหนดความสำคัญของข้อมูลที่มีคุณภาพ
  (Establish the importance of data quality)
- กำหนดคณะกรรมการดูแลคุณภาพของข้อมูล
  (Form a data quality steering committee)
- กำหนดขอบข่ายงานของการทำให้ข้อมูลมีคุณภาพ
  (Institute a data quality framework)
- กำหนดหน้าที่และความรับผิดชอบ
  (Assign roles and responsibilities)
- เลือกเครื่องมือสำหรับช่วยในการปรับปรุงคุณภาพของข้อมูล
  (Select tools to assist in the data purification process)
- จัดเตรียมการโปรแกรมภายในองค์กรเมื่อจำเป็น
  (Prepare in-house programs as needed)
- อบรมพนักงานเกี่ยวกับเทคนิคในการทำความสะอาดข้อมูล
  (Train the participants in data cleansing techniques)

- ทบทวนและยืนยันมาตรฐานข้อมูล
  (Review and confirm data standards)
- จัดลำดับความสำคัญของข้อมูลเป็น ระดับสูง ระดับกลาง และระดับต่ำ
  (Prioritize data into high, medium, and low categories)
- เตรียมตารางเวลาสำหรับการเริ่มการปรับปรุงคุณภาพของข้อมูลที่มีลำดับความสำคัญสูง
  (Prepare a schedule for data purification beginning with the high priority data)
- ตรวจสอบให้แน่ใจว่าเทคนิคสำหรับการลดความซ้ำซ้อนของเรคคอร์ด และการตรวจสอบข้อมูลจากภายนอกสามารถทำงานได้อย่างถูกต้อง
  (Ensure that techniques are available to correct duplicate records and to audit external data)
- ดำเนินการปรับปรุงข้อมูลตามตารางเวลาที่กำหนดไว้
  (Proceed with the purification process according to the defined schedule)

ข้อแนะนำในการปรับปรุงคุณภาพของข้อมูล

- ทำการเชื่อมโยงคุณภาพของข้อมูลกับความต้องการทางธุรกิจ (Link data quality with specific business objectives)
- ระบุข้อมูลที่มีความผิดพลาดซึ่งมีผลกระทบต่อการทำงานค่อนข้างมาก และเริ่มการปรับปรุงคุณภาพของข้อมูลจากข้อมูลเหล่านั้น (Identify high-impact pollution sources and begin your purification process with these)

- อย่าพยายามที่จะทำทุกอย่างที่โดยการเขียนโปรแกรมเอง (Do not try to do everything with in-house programs)
- เครื่องมือเป็นสิ่งที่ดีและมีประโยชน์ ทำการเลือกเครื่องมือที่เหมาะสม (Tools are good and useful. Select the proper tools)
- เห็นด้วยกับมาตราฐาน (Agree on standards)

SECTION 7

คำถามท้ายบท

1. จงอธิบายเหตุผลที่คุณภาพข้อมูลมีความสำคัญกับการสร้างคลังข้อมูล โดยยกตัวอย่าง 5 เหตุผล
2. จงอธิบายเหตุที่คุณภาพของข้อมูลจะมีความสำคัญมากกว่าความถูกต้องของข้อมูล
3. จงยกตัวอย่างประโยชน์ที่คาดว่าจะได้รับเมื่อคลังข้อมูลมีข้อมูลที่มีคุณภาพ (อย่างน้อย 3 ข้อ)
4. จงยกตัวอย่างปัญหาการเกิดขึ้นของข้อมูลที่ไม่มีคุณภาพ (4 ตัวอย่าง)
5. จงอธิบายเกี่ยวกับฟังก์ชันการทำงานของ data correction ในเครื่องมือสำหรับทำความสะอาดข้อมูล
6. จงแจกแจงสาเหตุที่ทำให้เกิดข้อผิดพลาดกับข้อมูล (3 ข้อ)
7. วิธีการทำความสะอาดแบบ “Clean as you go” จะมีรายละเอียดการทำงานอย่างไร
8. จงแจกแจงผู้ที่มีส่วนร่วมกับการปรับปรุงคุณภาพของข้อมูล (3 ข้อ)
9. ข้อแนะนำในการปรับปรุงคุณภาพข้อมูลมีอะไรบ้าง

CHAPTER 11

การประมวลผลการวิเคราะห์ข้อมูลแบบออนไลน์

SECTION 1

แผนการสอนประจำบท

วัตถุประสงค์ของบทเรียน

- ทำความเข้าใจเกี่ยวกับความต้องการในการประมวลผลการวิเคราะห์ข้อมูลแบบออนไลน์
- ศึกษาเกี่ยวกับนิยามและกฎต่าง ๆ ของการประมวลผลการวิเคราะห์ข้อมูลแบบออนไลน์
- ศึกษาเกี่ยวกับคุณลักษณะและฟังก์ชันการทำงานของการประมวลผลการวิเคราะห์ข้อมูลออนไลน์
- ศึกษาเกี่ยวกับการวิเคราะห์ข้อมูลที่กระทำจากการประมวลผลการวิเคราะห์ข้อมูลแบบออนไลน์
- ศึกษาเกี่ยวกับโมเดลต่าง ๆ และสถาปัตยกรรมของโมเดลสำหรับสร้างระบบการประมวลผลการวิเคราะห์ข้อมูลแบบออนไลน์

เนื้อหาของบทเรียน

เนื้อหาในบทนี้จะประกอบด้วย ความต้องการในการประมวลผลการวิเคราะห์ข้อมูลแบบออนไลน์ นิยามและกฎต่างๆของ OLAP ฟังก์ชันและคุณลักษณะหลักของ OLAP โมเดลต่างๆของ OLAP ปัจจัยที่ต้องพิจารณาในการสร้างระบบ OLAP

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารประกอบการสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

กิจกรรมการเรียน-การสอน

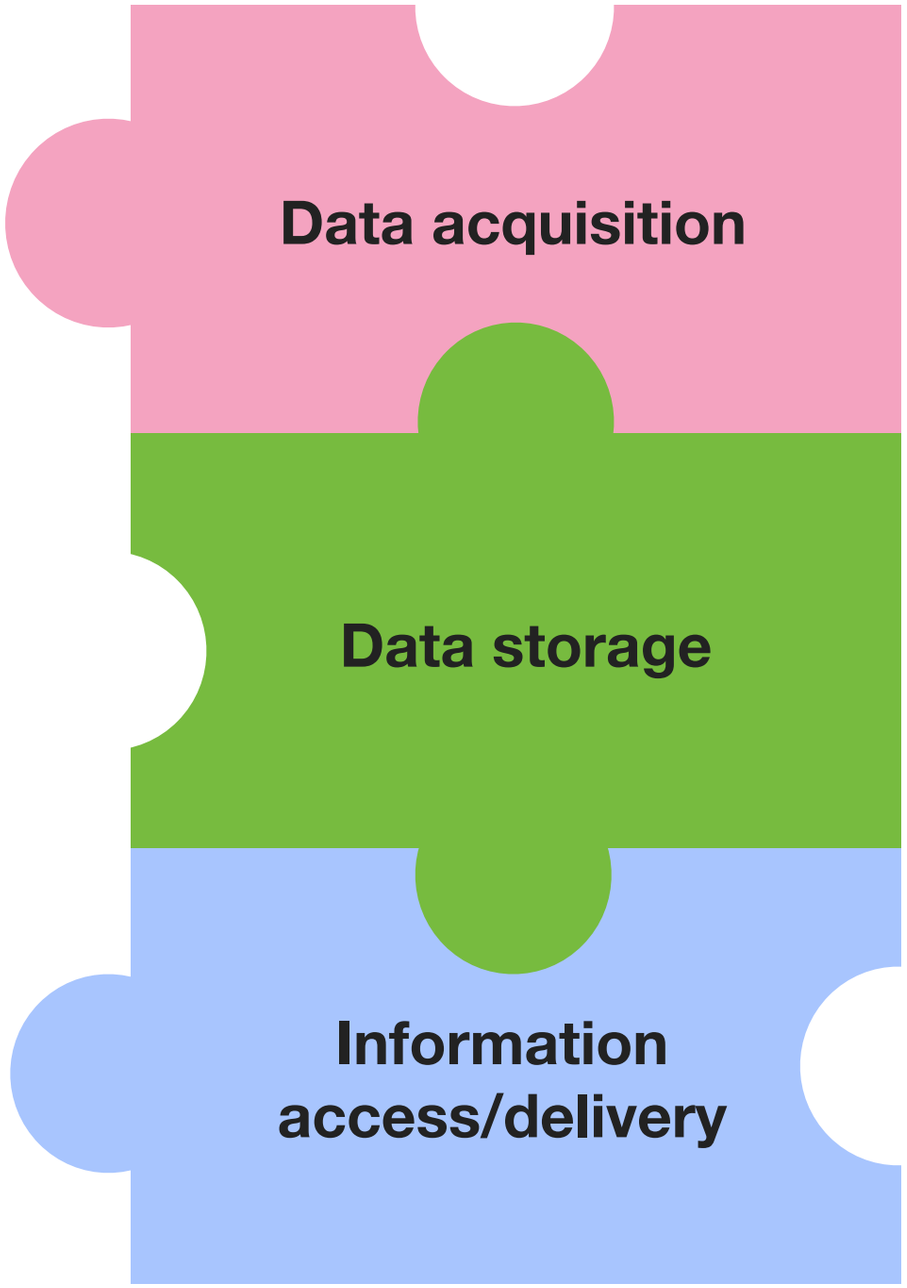
- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารประกอบการสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

การวัดและประเมินผล

- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

SECTION 2

บทนำ



บทนำ

จากบทก่อน ๆ หน้า เราจะทราบถึงฟังก์ชันการทำงานหลักของคลังข้อมูลจะประกอบด้วย 3 ฟังก์ชันหลัก ๆ คือ

(1) การได้มาซึ่งข้อมูล (Data acquisition)

(2) การจัดเก็บข้อมูล (Data storage)

(3) การเข้าถึงหรือการส่งผ่านข้อมูล (Information access/delivery)

โดยในบทนี้จะกล่าวถึงการใช้การประมวลผลการวิเคราะห์ข้อมูลแบบออนไลน์ (OnLine Analytical Processing, OLAP) ที่เปรียบเสมือนยานพาหนะที่ใช้ในการเข้าถึง/ส่งผ่านข้อมูลไปยังผู้ใช้งาน และจะกล่าวถึงคุณลักษณะและฟังก์ชันการทำงานต่าง ๆ รูปแบบการแสดงผล และโมเดลต่าง ๆ รวมถึงสถาปัตยกรรมของ OLAP ที่จะทำให้เราทราบว่าเราควรจะใช้โมเดลใดให้เหมาะสมกับสภาวะแวดล้อมที่เราเป็นอยู่มากที่สุด แต่ก่อนที่เราจะทำการศึกษารายละเอียดต่าง ๆ ของระบบ OLAP เราควรที่จะเข้าใจถึงความต้องการในการวิเคราะห์แบบออนไลน์และความสามารถของ OLAP ในการตอบสนองความต้องการเหล่านั้นเป็นอันดับแรก ซึ่งจะสามารถอธิบายได้ดังนี้

SECTION 3

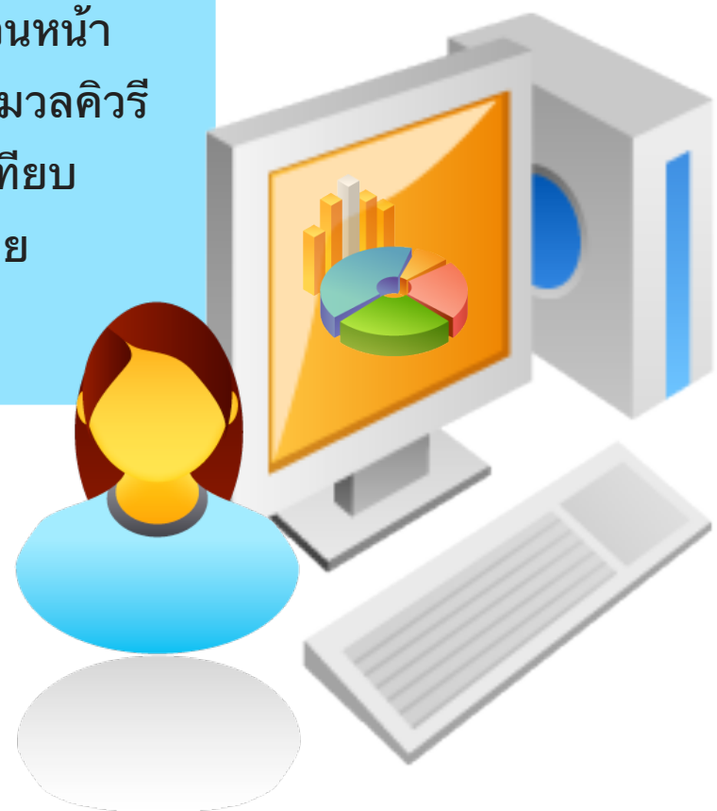
ความต้องการในการประมวลผลการวิเคราะห์ข้อมูลแบบออนไลน์

ความต้องการในการประมวลผลการวิเคราะห์ข้อมูลแบบออนไลน์

ในการสร้างคลังข้อมูลที่สนับสนุนการดำเนินธุรกิจในแต่ละวัน คลังข้อมูลที่เราสร้างขึ้นควรที่จะต้องมีความสามารถในการวิเคราะห์ข้อมูลที่มีความซับซ้อนได้ในเวลาอันรวดเร็ว โดยในปัจจุบันการวิเคราะห์ข้อมูลจะมีความซับซ้อนมากขึ้นและการวิเคราะห์แบบทั่วไปไม่สามารถตอบสนองต่อความต้องการของผู้ใช้ที่มากขึ้นได้อย่างครบถ้วน ลองพิจารณาตัวอย่างความต้องการของผู้ใช้ที่เกี่ยวเนื่องกับธุรกิจค้าปลีกขนาดใหญ่ที่ทำการพิจารณาข้อมูลการขายสินค้า ซึ่งโดยปกติของข้อมูลการขายสินค้าจะเกี่ยวเนื่องกับมิติทางธุรกิจหลายมิติด้วยกัน

เช่น วันที่มีการขายสินค้า รายการสินค้าที่ถูกขาย ช่องทางการขายสินค้า สาขา เขตหรือภูมิภาคที่มีการขายสินค้า โปรโมชั่นต่าง ๆ และอื่น ๆ ซึ่งจากมิติที่มีความหลากหลายทำให้ผู้ที่ต้องทำการวิเคราะห์ข้อมูลมักจะไม่ทำการวิเคราะห์ข้อมูลเพียงแค่มิติเดียวแต่จะทำการวิเคราะห์ข้อมูลในหลาย ๆ มิติ

เช่น จำนวนชิ้นสินค้า A ที่ถูกขายในร้านต่าง ๆ ในเมืองเอดิสัน รัฐนิวเจอซี่ย์เป็นอย่างไรบ้าง หรือรายได้/ผลกำไรที่ได้จากการขายสินค้า X ใน 3 เดือนหลังสุด โดยแบ่งเป็นผลกำไรในแต่ละเดือน ซึ่งรายได้นั้นจะเป็นรายได้จากการขายสินค้าในเขต south central โดยทำการแบ่งผลกำไรตามโปรโมชั่นต่าง ๆ และท้ายสุดจะต้องการข้อมูลในเชิงเปรียบเทียบผลกำไรกับรายการสินค้าเดียวกันในรุ่น (เวอร์ชั่น) ก่อนหน้า จากคิวรีดังกล่าวเราจะเห็นว่าการวิเคราะห์ข้อมูลนั้นไม่ได้หยุดเพียงแค่การประมวลคิวรีที่เกี่ยวข้องกับหลายมิติเพียงคิวรีเดียวเท่านั้น แต่ผู้ใช้ยังคงถามถึงการเปรียบเทียบยอดขายของสินค้าเวอร์ชั่นก่อนหน้าและรายการสินค้า X เทียบกับมิติอื่น ๆ ด้วย

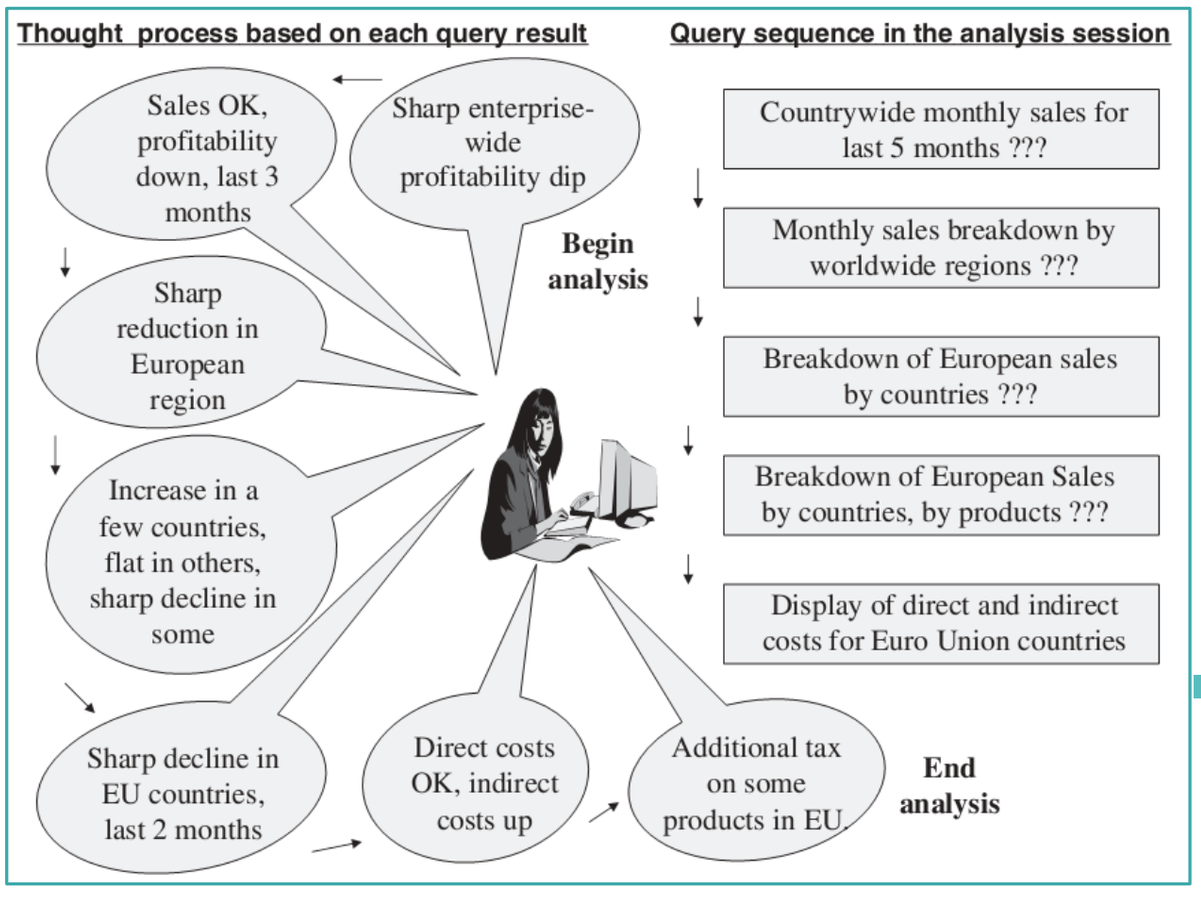


ซึ่งจากความต้องการดังกล่าว เราจำเป็นที่จะต้องทำการหมุนหรือปรับเปลี่ยนข้อมูลในแกนคอลัมน์และแถวของข้อมูล เพื่อทำการแสดงผลลัพธ์ตามที่ผู้ใช้ต้องการ

จากตัวอย่างข้างต้น เราจะเห็นว่าข้อมูลที่ต้องทำการวิเคราะห์นั้นค่อนข้างมีความซับซ้อน ดังนั้น ในการที่จะวิเคราะห์ข้อมูลได้อย่างมีประสิทธิภาพ ผู้ใช้มักต้องการวิธีการที่ง่ายสำหรับดำเนินการวิเคราะห์ข้อมูลที่มีความซับซ้อนที่เกี่ยวข้องกับข้อมูลทางธุรกิจหลายมิติด้วยกัน ผู้ใช้จะต้องการเครื่องมือที่จะสามารถปรับเปลี่ยนมุมมองของข้อมูลได้หลายมุมมอง และมีความยืดหยุ่นในการเข้าถึงข้อมูลอีกด้วย โดยในการใช้งานคลังข้อมูลของนักวิเคราะห์มักจะต้องการวิเคราะห์ข้อมูลหลาย ๆ มิติ และไล่ไปตามลำดับชั้นต่าง ๆ ของแอทริบิวในแต่ละมิติ รวมถึงต้องการที่จะได้รับผลรับในหลาย ๆ รูปแบบอีกด้วย เช่น ตาราง กราฟและชาร์ต ต่าง ๆ

ดังนั้นถ้าระบบคลังข้อมูลไม่มีความสามารถในการวิเคราะห์ข้อมูลหลาย ๆ มิติหรือหลาย ๆ มุมมอง เราจะสามารถกล่าวได้ว่าคลังข้อมูลที่สร้างขึ้นนั้นยังไม่สมบูรณ์ ซึ่งเราจะต้องทำการพัฒนาคลังข้อมูลต่อไป

ในการวิเคราะห์ข้อมูลครั้งหนึ่ง ๆ จะสามารถทำการวิเคราะห์โดยการกำหนดคิวรีที่ต้องการ จากนั้นสั่งให้ระบบคลังข้อมูลทำการประมวลผล แล้วทำการรอผลลัพธ์ที่จะแสดงทางหน้าจอ ซึ่งในบางคิวรีอาจได้ผลลัพธ์แบบทันที เนื่องจากทำการย่อยข้อมูลหรือรวมข้อมูลจากผลลัพธ์ของคิวรีก่อนหน้า การทำงานในลักษณะนี้จะช่วยให้เกิดการคำนวณที่รวดเร็ว ลองพิจารณาตัวอย่างของการวิเคราะห์ข้อมูลครั้งหนึ่ง ๆ ดังแสดงในรูปที่ 11-1 ที่จะเป็นการประมวลผลหลาย ๆ คิวรีเรียงตามลำดับ โดยคิวรีเหล่านั้นจะมีความเกี่ยวเนื่องของข้อมูลกันอยู่ ซึ่งในการประมวลผลคิวรีเหล่านี้จะต้องการระบบที่ทำการประมวลผลที่รวดเร็ว ยืดหยุ่นและสนับสนุนการทำงานที่ซับซ้อนและการคำนวณที่มีประสิทธิภาพ



รูปที่ 11-1 การประมวลผลคิวรีครั้งหนึ่ง ๆ ของผู้ใช้งาน

โดยในการคำนวณหรือประมวลผลครั้งหนึ่งจะประกอบไปด้วย

- การสร้างผลสรุปของข้อมูลและรวบยอดข้อมูลเหล่านั้นตามลำดับชั้นของข้อมูลในมิติต่าง ๆ
- การเจาะลึกไปตามลำดับชั้นของข้อมูลในแต่ละมิติ
- การคำนวณพื้นฐาน เช่น การคำนวณผลกำไรโดยการนำยอดขายลบด้วยต้นทุน เป็นต้น
- การใช้สมการทางคณิศาสตร์ดำเนินการกับตัวชี้วัดต่าง ๆ
- การหาค่าเฉลี่ยและเปอร์เซ็นต์ของการเติบโตในแง่มุมต่าง ๆ
- การวิเคราะห์แนวโน้มต่าง ๆ โดยใช้หลักทางสถิติ

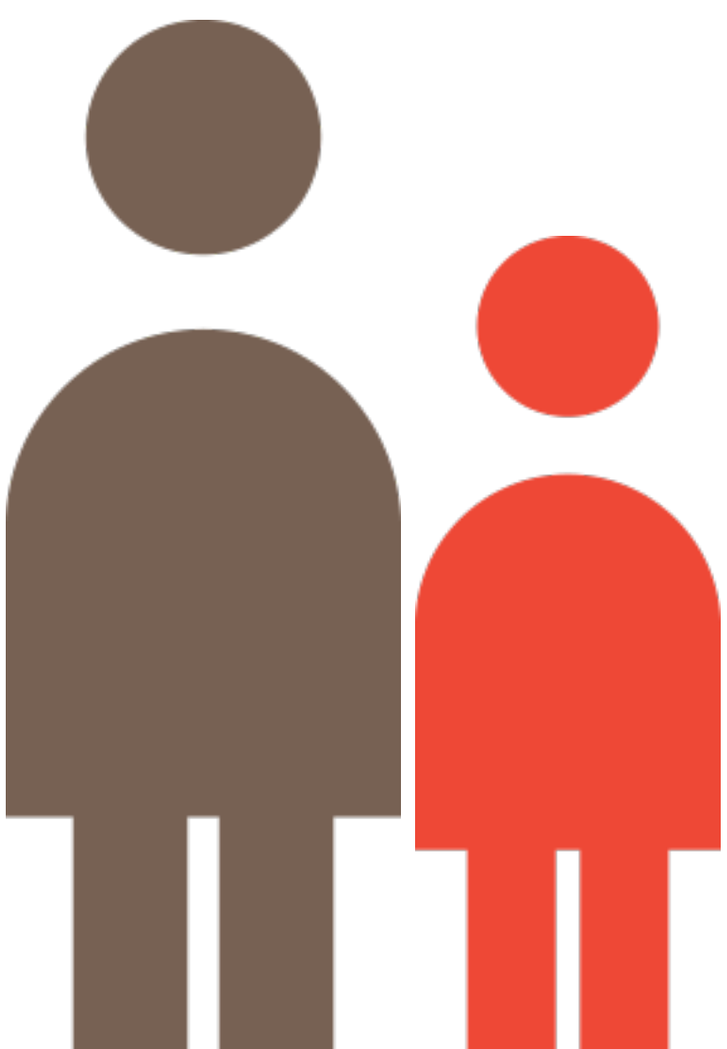
จากที่กล่าวมาข้างต้นเราจะทราบถึงความต้องการของผู้ใช้ในการประมวลคิวรีต่าง ๆ ที่ซึ่งสามารถแบ่งเป็น 3 ข้อหลัก ๆ ด้วยกันคือ

1) ระบบที่ใช้ส่งผ่านข้อมูลจะต้องมีความสามารถในการแสดงผลลัพธ์ได้หลาย ๆ มิติในหลาย ๆ มุมมอง

2) ระบบที่ใช้ส่งผ่านข้อมูลจะต้องสามารถรองรับวิธีการวิเคราะห์ข้อมูลหลาย ๆ มิติและลำดับชั้นของข้อมูลต่าง ๆ ได้หลายวิธี

3) ระบบที่ใช้ส่งผ่านข้อมูลจะต้องสามารถประมวลผลคิวรีที่ซับซ้อนได้อย่างรวดเร็ว

จากความต้องการทั้ง 3 ข้อ ถ้าเราใช้เทคโนโลยีที่มีในปัจจุบัน เช่น การใช้ SQL query ในการเรียกใช้ข้อมูลจากฐานข้อมูล วิธีการนี้จะต้องทำการอ่านข้อมูลค่อนข้างมากเช่นการอ่านข้อมูลทั้งตาราง การทำ multiple join การ grouping และการเรียงลำดับข้อมูล ซึ่งจากการทำงานดังกล่าวจะใช้เวลาค่อนข้างนานมาก



นอกจากการใช้ภาษา SQL แล้ว เรายังสามารถแสดงผลของการวิเคราะห์หนึ่ง ๆ ในรูปแบบสเปรดชีทได้ แต่อย่างไรก็ตาม สเปรดชีทสามารถทำการแสดงผลได้เพียง 2 มิติเท่านั้น คือ ในแกนคอลัมน์ และแถวของข้อมูล ซึ่งจากข้อจำกัดดังกล่าว จึงได้มีผู้คิดค้นวิธีการแก้ปัญหาการแสดงผลข้อมูลให้อยู่ในรูปแบบของ 3 มิติ นั่นคือ แกนคอลัมน์ แถว และเพจของข้อมูล

ตัวอย่างเช่น ข้อมูลแต่ละแถวจะแสดงถึงแต่ละรายการสินค้า ข้อมูลแต่ละคอลัมน์จะแสดงถึงร้านค้าหรือสาขาหนึ่ง ๆ และเพจหนึ่ง ๆ จะแสดงข้อมูลแกนเวลา โดยเราสามารถเลือกได้ว่าข้อมูลหนึ่งเพจหมายถึงข้อมูลในเดือนหนึ่ง ๆ เป็นต้น

โดยในปัจจุบัน สเปรดชีทใหม่ ๆ ได้มีการพัฒนาให้สามารถแสดงผลได้หลายมิติมากขึ้น โดยถึงแม้ว่าสเปรดชีทจะมีการพัฒนาอย่างต่อเนื่องแต่สเปรดชีทก็ยังคงยุ่งยาก ซึ่งถ้าเราต้องการที่จะทำการวิเคราะห์ข้อมูล 4 มิติ โดยที่แต่ละมิติจะมีข้อมูล 5 ระดับในแต่ละลำดับชั้น

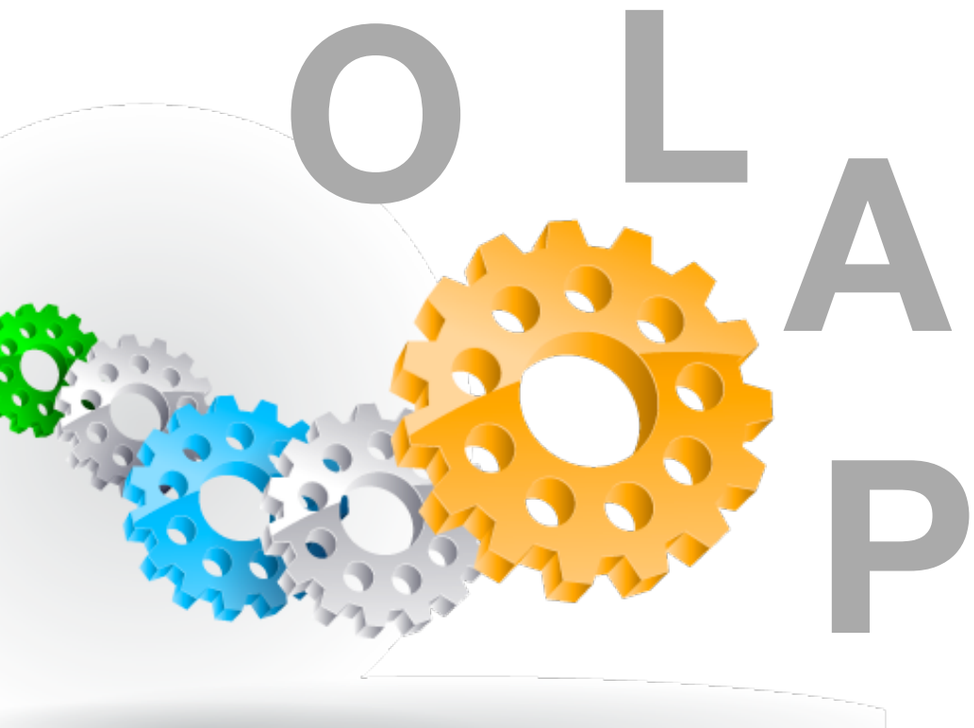
จากนั้นลองทำการสร้างการวิเคราะห์ข้อมูลที่อยู่ในรูปแบบของสเปรดชีทและทำการแสดงผลในหลาย ๆ มุมมอง เราจะเห็นว่าการวิเคราะห์แบบเจาะลึกและแบบผลสรุปของข้อมูลสามารถทำได้ค่อนข้างยากอีกด้วย

การใช้ OLAP ในการวิเคราะห์ข้อมูล

จากข้างต้นที่เราจะทราบว่าเครื่องมือทั่ว ๆ ไป เช่น SQL และสเปรดชีทไม่สามารถตอบสนองความต้องการของผู้ใช้ได้อย่างครบถ้วน ดังนั้น ในบทนี้เราจะทำการศึกษาเกี่ยวกับการใช้ระบบ OLAP ในการวิเคราะห์ข้อมูลหลายๆมิติ โดยเมื่อระบบคลังข้อมูลมีการใช้ระบบ OLAP ในการวิเคราะห์ข้อมูล จะทำให้ระบบคลังข้อมูลมีความสามารถดังต่อไปนี้

- สามารถมีการจัดระเบียบตัวชี้วัดต่างๆให้สอดคล้องกับมิติต่าง ๆ ได้
- สนับสนุนการวิเคราะห์ข้อมูลในหลาย ๆ แง่มุม
- มีความสามารถในการวิเคราะห์ข้อมูลแบบเจาะลึกและแบบผลสรุปในแต่ละมิติทางธุรกิจ
- สามารถนำสูตรคำนวณทางคณิตศาสตร์มาประยุกต์ใช้ในการคำนวณต่าง ๆ ได้
- สามารถทำงานได้รวดเร็ว
- เป็นส่วนเติมเต็มของระบบการวิเคราะห์ข้อมูลอื่น ๆ เช่น การทำเหมืองข้อมูล (Data mining)
- สามารถเพิ่มความเข้าใจเกี่ยวกับผลลัพธ์โดยการแสดงผลโดยใช้กราฟและชาร์ตต่าง ๆ
- สามารถทำงานบนเว็บได้
- สามารถทำงานแบบโต้ตอบทันทีได้

จากประโยชน์ของการประยุกต์ใช้ระบบ OLAP ข้างต้น ระบบ OLAP จะมีจุดแข็งอย่างหนึ่งในการทำงานดังแสดงในรูปที่ 11-2 ที่จะแสดงถึงการวิเคราะห์ข้อมูลแบบต่อเนื่อง โดยการวิเคราห์ข้อมูลจะเริ่มจากการวิเคราะห์ข้อมูลที่ค่อนข้างจะเป็นผลสรุป นั่นคือ ยอดขายทั้งหมดของแต่ละรายการสินค้า จากนั้นผู้ใช้อาจทำการเจาะลึกลงไปตามรายละเอียดตามปี โดยการแยกยอดขายสินค้าออกตามปีหนึ่ง ๆ หรือเราสามารถปรับเปลี่ยนมุมมองของการวิเคราะห์ได้ โดยทำการเปลี่ยนปีมาเป็นแถว และรายการสินค้ามาเป็นคอลัมน์เพื่อทำการเรียกดูผลสรุปของข้อมูลที่ง่ายขึ้นได้ ซึ่งจากความสามารถในการทำงานดังกล่าวจะช่วยให้ระบบคลังข้อมูลสามารถวิเคราะห์ข้อมูลตามที่ผู้ใช้ต้องการภายในเวลาอันรวดเร็วได้



LINE	TOTAL SALES
Clothing	$12,836,450
Electronics	$16,068,300
Video	$21,262,190
Kitchen	$17,704,400
Appliances	$19,600,800
Total	$87,472,140

1 High level summary by product line

2 Drill down by year

LINE	1998	1999	2000	TOTAL
Clothing	$3,457,000	$3,590,050	$5,789,400	$12,836,450
Electronics	$5,894,800	$4,078,900	$6,094,600	$16,068,300
Video	$7,198,700	$6,057,890	$8,005,600	$21,262,190
Kitchen	$4,875,400	$5,894,500	$6,934,500	$17,704,400
Appliances	$5,947,300	$6,104,500	$7,549,000	$19,600,800
Total	$27,373,200	$25,725,840	$34,373,100	$87,472,140

3 Rotate columns to rows

YEAR	Clothing	Electronics	Video	Kitchen	Appliances	TOTAL
1998	$3,457,000	$5,894,800	$7,198,700	$4,875,400	$5,947,300	$27,373,200
1999	$3,590,050	$4,078,900	$6,057,890	$5,894,500	$6,104,500	$25,725,840
2000	$5,789,400	$6,094,600	$8,005,600	$6,934,500	$7,549,000	$34,373,100
Total	$12,836,450	$16,068,300	$21,262,190	$17,704,400	$19,600,800	$87,472,140

รูปที่ 11-2 ตัวอย่างการใช้งาน OLAP ครั้งหนึ่ง ๆ

SECTION 4

นิยามและกฎต่าง ๆ ของ OLAP

O
L
A
P

นิยามและกฎต่าง ๆ ของ OLAP

ระบบ OLAP ตั้งอยู่บนพื้นฐานของการวิเคราะห์ข้อมูลหลายมิติได้ถูกคิดค้นขึ้นโดย Dr. E.F. Codd ในปี 1993 ซึ่ง ณ เวลานั้น Dr. Codd ได้ทำการนิยามว่า

"ระบบ *OLAP (OnLine Analytical Processing)* นั้นเป็นซอร์ฟแวร์ชนิดหนึ่งที่ทำให้นักวิเคราะห์ ผู้จัดการ และผู้บริหารสามารถเข้าถึงข้อมูลได้อย่างรวดเร็ว โดยในการเข้าถึงข้อมูลจะสามารถเข้าถึงได้หลายแง่มุม โดยข้อมูลที่ถูกแสดงผลให้กับผู้ใช้จะเกิดจากแปลงข้อมูลดิบไปเป็นข้อมูลในมิติต่าง ๆ ที่อยู่ในรูปแบบที่ผู้ใช้สามารถเข้าใจได้"

โดยนอกเหนือจากการนิยามข้างต้นแล้ว Dr. Codd ยังได้กำหนดกฎต่าง ๆ เกี่ยวกับระบบ OLAP ทั้งสิ้น 12 ข้อ โดยกฎเหล่านั้นเปรียบเสมือนกับคู่มือในการสร้างและใช้งานระบบ OLAP โดยกฎทั้ง 12 ข้อจะมีรายละเอียดคร่าว ๆ ดังนี้

1

Multidimensional conceptual view:

ระบบ OLAP จะต้องมีความสามารถในการสร้างโมเดลข้อมูลในหลาย ๆ มิติที่มีความสามารถในการวิเคราะห์และใช้งานง่าย โดยโมเดลข้อมูลหลายมิติที่สร้างขึ้นจะต้องสอดคล้องกับสิ่งที่ผู้ใช้อยากรู้ในการแก้ปัญหาทางธุรกิจ

2

Transparency:

ในการสร้างระบบ OLAP เราจะต้องทำให้ผู้ใช้สามารถเห็นส่วนประกอบต่างๆของระบบได้ เช่น เทคโนโลยีต่าง ๆ พื้นที่สำหรับจัดเก็บข้อมูล สถาปัตยกรรมการคำนวณ และแหล่งข้อมูล เป็นต้น การเปิดเผยข้อมูลเหล่านี้จะช่วยเพิ่มขีดความสามารถให้แก่ผู้ใช้ให้สามารถใช้ระบบคลังข้อมูลได้ดีขึ้น

3

Acessibility:

ระบบ OLAP จะต้องทำให้ผู้ใช้สามารถเข้าถึงข้อมูลที่ต้องการได้เท่านั้น

4

Consistent reporting performance:

ระบบ OLAP ควรที่จะต้องมีการแจ้งให้ผู้ใช้ได้รับรู้เกี่ยวกับเวลาในการประมวลผล หรือการใช้ทรัพยากรของเครื่องมือทำการประมวลผลคิวรีต่าง ๆ

5

Client/server architecture:

ระบบ OLAP ควรที่จะตั้งอยู่บนพื้นฐานของสถาปัตยกรรม client-server ที่จะทำให้ระบบมีประสิทธิภาพการทำงานที่ดี มีความยืดหยุ่น มีความสามารถในการปรับตัว และสามารถทำงานร่วมกับระบบอื่น ๆ ได้

6

Generic dimensionality:

เราต้องทำให้แน่ใจว่าทุกๆมิติของข้อมูลจะมีความสามารถในการดำเนินการต่าง ๆ รวมถึงโครงสร้างพื้นฐานและเทคนิคในการเข้าถึงข้อมูลที่เหมือนกัน

7

Dynamic sparse matrix handling:

ควรจะมีการนำ physical schema มาใช้ทำการสร้างโมเดลในการวิเคราะห์ข้อมูล โดย schema ที่นำมาใช้จะสามารถใช้หน่วยความจำและการเข้าถึงข้อมูลได้อย่างมีประสิทธิภาพ เช่น การคำนวณต่างๆกับข้อมูลโดยตรง การใช้ B-tree ในการจัดเก็บข้อมูล การใช้แฮช และการรวมเทคนิคต่าง ๆ เข้าด้วยกัน

8

Multiuser support:

ระบบ OLAP จะต้องสามารถให้บริการแก่ผู้ใช้หลาย ๆ คนได้พร้อม ๆ กัน โดยผู้ใช้หลาย ๆ รายอาจจะเรียกใช้ข้อมูลที่เหมือนกันพร้อม ๆ กันก็เป็นได้

9

Unrestricted cross-dimensional operations:

ระบบ OLAP จะต้องมีความสามารถที่จะทำการรู้จำลำดับชั้นของข้อมูลในมิติทางธุรกิจต่าง ๆ และสามารถทำการเรียกดูข้อมูลแบบเจาะลึกและแบบผลสรุปทั้งในมิติหนึ่ง ๆ หรือหลาย ๆ มิติได้อย่างอัตโนมัติ

10

Intuitive data manipulation:

ระบบ OLAP จะต้องมีความสามารถที่จะดำเนินการวิเคราะห์ข้อมูลในแง่มุมต่าง ๆ ได้ เช่น การวิเคราะห์ข้อมูลแบบเจาะลึกไปตามคอลัมน์หรือแถวของข้อมูล สามารถทำการซูมเข้า-ออกได้ โดยการใช้ point-and-click และ drag-and-drop ได้ในเซลล์ (Cell) ต่าง ๆ ของโมเดล

11

Flexible reporting:

ระบบ OLAP จะต้องมีความสามารถที่จะทำให้ผู้ใช้สามารถปรับเปลี่ยนข้อมูลในคอลัมน์และแถวที่จะง่ายต่อการวิเคราะห์ต่าง ๆ

12

Unlimited dimensions and aggregation levels:

ระบบ OLAP จะต้องไม่จำกัดจำนวนมิติและลำดับชั้นในการวิเคราะห์ข้อมูล รวมถึงการแสดงผลด้วยเช่นกัน

จากกฎทั้งหมดข้างต้น เราจะทราบถึงความต้องการในการวิเคราะห์ข้อมูลในหลาย ๆ มิติ และความสามารถของ OLAP ในการตอบสนองความต้องการเหล่านั้น ซึ่งกฎที่กล่าวมาทั้งหมดข้างต้นจะช่วยบ่งบอกถึงลักษณะพื้นฐานของ OLAP ได้ กล่าวคือ ระบบ OLAP จะทำให้ผู้ใช้สามารถทำการวิเคราะห์หรือเรียกดูข้อมูลได้หลายมิติ และสามารถประมวลผลคิวรีที่มีความซับซ้อนแบบเร่งด่วนได้ นอกจากนั้นระบบ OLAP ยังควรที่จะมีความสามารถในการวิเคราะห์ข้อมูลแบบเจาะลึกเพื่อให้ได้รายละเอียดของข้อมูลมากขึ้น รวมถึงการวิเคราะห์ข้อมูลแบบผลสรุปที่เป็นการรวบยอดข้อมูลตัวชี้วัดต่าง ๆ ที่เกี่ยวเนื่องกับมิติทางธุรกิจหนึ่ง ๆ หรือหลายมิติได้ ท้ายสุด คือ ระบบ OLAP จะมีความสามารถในการคำนวณและเปรียบเทียบข้อมูลที่ซับซ้อนและแสดงผลลัพธ์เหล่านั้นในรูปแบบของชาร์ตและกราฟต่าง ๆ ได้

SECTION 5

ฟังก์ชันและคุณลักษณะหลักของ OLAP

ฟังก์ชันและคุณลักษณะหลักของ **OLAP**

OLAP

ในการประยุกต์ใช้ OLAP ที่เป็นระบบหรือเครื่องมือสำหรับการเข้าถึง/ส่งผ่านข้อมูลนั้นจะช่วยเพิ่มขีดความสามารถในการเข้าถึงข้อมูล และจะทำให้ผู้ใช้สามารถเข้าถึงข้อมูลได้ง่ายขึ้น โดยระบบ OLAP จะมีคุณลักษณะต่างๆมากมายดังแสดงในดังรูปที่ 11-3 ซึ่งจากรูปจะแสดงเกี่ยวกับคุณลักษณะของ OLAP ที่จะประกอบไปด้วย

(1) คุณลักษณะพื้นฐานที่ประกอบไปด้วยการวิเคราะห์มิติต่าง ๆ การวิเคราะห์ข้อมูลแบบเจาะลึก (drill down) และการวิเคราะห์ข้อมูลแบบสรุป (roll up) รวมถึงการวิเคราะห์ข้อมูลเพียงบางส่วนและการปรับเปลี่ยนมุมมองของผลลัพธ์ (slicing and dicing)

(2) คุณลักษณะพิเศษที่จะเป็นคุณลักษณะที่แอบแฝงอยู่ภายใต้ฟังก์ชันการทำงานต่าง ๆ

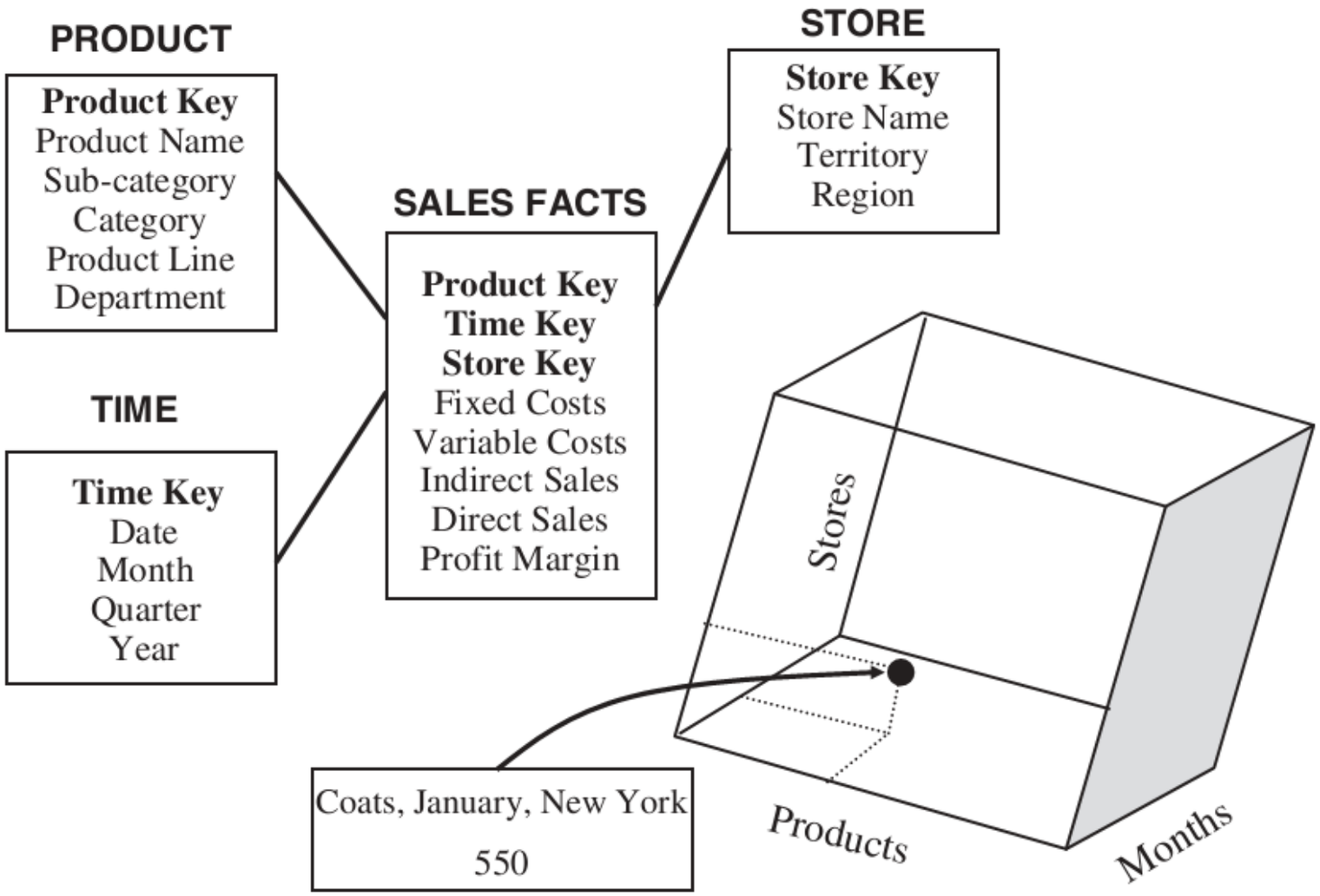
BASIC FEATURES	Multidimensional analysis	Consistent performance	Fast response times for interactive queries
	Drill-down and roll-up	Navigation in and out of details	Slice-and-dice or rotation
	Multiple view modes	Easy scalability	Time intelligence (year-to-date, fiscal period)
ADVANCED FEATURES	Powerful calculations	Cross-dimensional calculations	Pre-calculation or pre-consolidation
	Drill-through across dimensions or details	Sophisticated presentation & displays	Collaborative decision making
	Derived data values through formulas	Application of alert technology	Report generation with agent technology

รูปที่ 11-3 คุณลักษณะของ OLAP

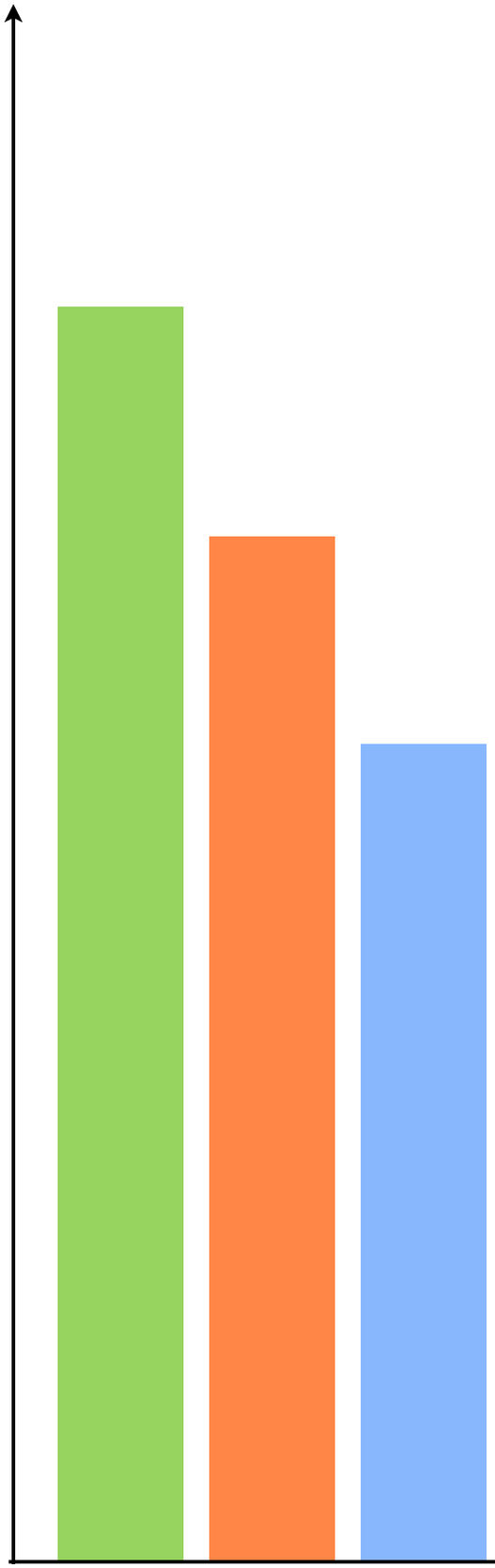
การวิเคราะห์ข้อมูลมิติต่าง ๆ

การวิเคราะห์ข้อมูลหลาย ๆ มิติจะเป็นการแสดงถึงขีดความสามารถของ OLAP ที่จะทำให้เราสามารถวิเคราะห์ข้อมูลได้อย่างมีประสิทธิภาพ เพื่อให้เห็นภาพของการประยุกต์ใช้ OLAP กับการวิเคราะห์มิติต่าง ๆ ลองพิจารณา star schema ในรูปที่ 11-4 ที่เป็นข้อมูลการขายหรือยอดขายสินค้าที่ประกอบไปด้วยมิติทางธุรกิจ 3 มิติซึ่งก็คือ มิติรายการสินค้า มิติแกนเวลา และมิติสาขา/ห้างร้าน ของบริษัทที่มีการประกอบการ

จากทั้ง 3 มิติเราสามารถแสดงการเชื่อมโยงและความสัมพันธ์ระหว่างมิติโดยใช้ลูกบาศก์ (cube) ดังรูป โดยจากรูป มิติรายการสินค้าจะอยู่ที่แกน X มิติแกนเวลาจะอยู่ที่แกน Y และมิติสาขา/ห้างร้านจะอยู่ที่แกน Z โดยค่าที่แสดงในแต่ละแกนก็จะเป็นค่าในที่เป็นตัวชี้วัดที่สัมพันธ์กับแต่ละแอทริบิวของแต่ละมิติ เช่น ข้อมูลเดือนจะเป็นแอทริบิวหนึ่งในมิติแกนเวลา ซึ่งเราสามารถเรียกดูยอดขายโดยตรงของแต่ละเดือนได้ แต่เมื่อไรก็ตามที่เราเลือกรายการสินค้ารายการหนึ่ง ๆ จากมิติรายการสินค้า เดือนหนึ่ง ๆ จากมิติแกนเวลา และเขตพื้นที่หนึ่งๆจากมิติสาขา/ห้างร้าน ตัวอย่างเช่น เลือกรายการสินค้าเสื้อคลุม เดือนมกราคม พื้นที่รัฐนิวยอร์ก จะทำให้ผู้ใช้สามารถเรียกดูข้อมูลยอดขายเสื้อคลุมของเดือนมกราคมซึ่งถูกขายในรัฐนิวยอร์ก โดยผลลัพธ์ที่ได้เกิดจากการอินเทอร์เซกชันกันระหว่างเส้นเชื่อมของแต่ละมิติ เป็นต้น



รูปที่ 11-4 ตัวอย่าง star schema และ cube



จากการเลือกข้อมูลในแต่ละมิติข้างต้น ถ้าเราต้องการแสดงผลลัพธ์ยอดขายสินค้าจากข้อมูลที่ทำการเลือกจากทั้ง 3 มิติข้างต้น (เสื้อคลุม เดือนมกราคม และรัฐนิวยอร์ก) ในรูปแบบสเปรดชีต เราจะสามารถนำข้อมูลรายการสินค้าไปแสดงผลในแต่ละคอลัมน์ ข้อมูลแต่ละเดือนไปแสดงผลในแต่ละแถว และข้อมูลแต่ละสาขา/ห้างร้านไปแสดงผลในแต่ละหน้า (page) ของสเปรดชีต เป็นต้น

ซึ่งตัวอย่างของข้อมูลหน้าหนึ่ง ๆ ที่ประกอบไปด้วยยอดขายสินค้าทุกรายการ ที่ถูกขายในแต่ละเดือน และถูกขายในรัฐนิวยอร์ก จะสามารถแสดงผลได้ดังรูปที่ 11-5 โดยจากรูปจะเป็นเหมือนกับการตัดส่วนของข้อมูลออกจากลูกบาศก์ (slice of the cube) โดยจะทำการตัดส่วนของข้อมูลในส่วนของรัฐนิวยอร์กออกจากมิติสาขา/ห้างร้าน (ตัวส่วนของแกน Z) โดยข้อมูลในแกน X และ Y ยังอยู่ครบถ้วน

ซึ่งจากการตัดส่วนของข้อมูลจากลูกบาศก์จะทำให้เราสามารถเรียกดูข้อมูลเฉพาะส่วนได้หลายส่วนด้วยกัน โดยในการเรียกดูข้อมูลในแต่ละส่วนจะสอดคล้องกับคิวรีที่ถูกกำหนดจากผู้ใช้ ซึ่งผลลัพธ์ที่ได้จะถูกแสดงผลบนหน้าจอที่ประกอบไปด้วย 3 มิตินั่นคือ แถวของข้อมูล คอลัมน์ของข้อมูล และเพจ (page) ของข้อมูลตามลำดับ

Store: New York

PAGES: STORE dimension

Products

COLUMNS: PRODUCT dimension

ROWS: TIME dimension

Months

	Hats	Coats	Jackets	Dresses	Shirts	Slacks
Jan	200	550	350	500	520	490
Feb	210	480	390	510	530	500
Mar	190	480	380	480	500	470
Apr	190	430	350	490	510	480
May	160	530	320	530	550	520
Jun	150	450	310	540	560	330
Jul	130	480	270	550	570	250
Aug	140	570	250	650	670	230
Sep	160	470	240	630	650	210
Oct	170	480	260	610	630	250
Nov	180	520	280	680	700	260
Dec	200	560	320	750	770	310

รูปที่ 11-5 การแสดงผลในหน้าหนึ่งในรูปแบบสเปรดชีตของข้อมูล 3 มิติ

ลองพิจารณาตัวอย่างคิวรีดังต่อไปนี้ ซึ่งเป็นตัวอย่างของการแบ่งส่วนของข้อมูลเพื่อเรียกดูข้อมูลในแง่มุมต่าง ๆ ดังนี้

คิวรี: จงแสดงยอดขายทั้งหมดของทุกรายการสินค้าในช่วง 5 ปีที่ผ่านมาที่ขายได้ในทุกสาขาของบริษัท (ณ ปัจจุบัน ปี 2012)

ผลลัพธ์ที่ได้:

แถวของข้อมูลที่เกิดขึ้นในปี 2007, 2008, 2009, 2010, 2011

คอลัมน์ของข้อมูลที่ประกอบไปด้วยยอดขายทั้งหมดของทุกรายการสินค้า

เพจหนึ่งสาขาต่อหนึ่งหน้า

คิวรี: จงแสดงผลลัพธ์การเปรียบเทียบยอดขายของทุกสาขาโดยทำการเปรียบเทียบยอดขายของแต่ละรายการสินค้าที่ถูกขายในระหว่างปี 2010 และ 2011

ผลลัพธ์ที่ได้:

แถวของข้อมูลในช่วงปี 2010 และ 2011 ที่เป็นเปอร์เซ็นต์การเพิ่มขึ้นหรือลดลงของยอดขายสินค้า

คอลัมน์หนึ่งคอลัมน์จะแสดงถึงแต่ละรายการสินค้า (จำนวนคอลัมน์ที่ถูกแสดงเป็นผลลัพธ์จะเท่ากับจำนวนรายการสินค้าทั้งหมดของบริษัท)

เพจหนึ่งหน้าที่แสดงทุกสาขา

คิวรี: จงแสดงผลลัพธ์การเปรียบเทียบยอดขายของทุกสาขาโดยทำการเปรียบเทียบยอดขายของแต่ละรายการสินค้าที่ถูกขายแบบลดราคาในระหว่างปี 2010 และ 2011

ผลลัพธ์ที่ได้:

แถวของข้อมูลในช่วงปี 2010 และ 2011 ที่เป็นเปอร์เซ็นต์การเพิ่มขึ้นหรือลดลงของยอดขายสินค้า

คอลัมน์หนึ่งคอลัมน์จะแสดงถึงแต่ละรายการสินค้า (จะแสดงเฉพาะรายการสินค้าที่ถูกขายแบบลดราคา)

เพจหนึ่งหน้าที่แสดงทุกสาขา

คิวรี: จงแสดงผลลัพธ์การเปรียบเทียบยอดขายของแต่ละสาขาโดยทำการเปรียบเทียบยอดขายของแต่ละรายการสินค้าที่ถูกขายแบบลดราคาในระหว่างปี 2010 และ 2011

ผลลัพธ์ที่ได้:

แถวของข้อมูลในช่วงปี 2010 และ 2011 ที่เป็นเปอร์เซ็นต์การเพิ่มขึ้นหรือลดลงของยอดขายสินค้า

คอลัมน์หนึ่งคอลัมน์จะแสดงถึงแต่ละรายการสินค้า (จะแสดงเฉพาะรายการสินค้าที่ถูกขายแบบลดราคา)

เพจหนึ่งสาขาต่อหนึ่งหน้า

คิวรี: จงแสดงผลลัพธ์ที่เป็นผลลัพธ์ของคิวรีก่อนหน้า โดยทำการปรับเปลี่ยนการแสดงผลระหว่างแถวและคอลัมน์ (เปลี่ยนจากแถวเป็นคอลัมน์และเปลี่ยนจากคอลัมน์เป็นแถว)

ผลลัพธ์ที่ได้:

แถวของข้อมูลซึ่งแต่ละแถวจะแสดงถึงแต่ละรายการสินค้าที่มีการขายลดราคา

คอลัมน์จะเป็นปี 2010 และ 2011 ที่เป็นเปอร์เซ็นต์การเพิ่มขึ้นหรือลดลงของยอดขายสินค้า

เพจหนึ่งสาขาต่อหนึ่งหน้า

คิวรี: จงแสดงผลลัพธ์ที่เป็นผลลัพธ์ของคิวรีก่อนหน้า โดยทำการปรับเปลี่ยนการแสดงผลระหว่างแถวและหน้า (เปลี่ยนจากแถวเป็นหน้าและเปลี่ยนจากหน้าเป็นแถว)

ผลลัพธ์ที่ได้:

แถวของข้อมูลโดยที่หนึ่งแถวจะแสดงข้อมูลหนึ่งสาขา

คอลัมน์จะเป็นปี 2010 และ 2011 ที่เป็นเปอร์เซ็นต์การเพิ่มขึ้นหรือลดลงของยอดขายสินค้า

เพจหนึ่งรายการสินค้าต่อหนึ่งหน้า ซึ่งจะเป็นหน้าของรายการสินค้าที่มีการขายลดราคา

จากตัวอย่างคิวรีข้างต้นที่ทำการประมวลผลกับ star schema ที่ประกอบไปด้วยข้อมูลทางธุรกิจ 3 มิติจะสามารถวิเคราะห์เกี่ยวกับการขายสินค้าที่เกี่ยวข้องกับมิติแกนเวลา มิติรายการสินค้า และมิติสาขาที่มีการขายสินค้า โดยความสัมพันธ์ระหว่างมิติต่าง ๆ จะสามารถแสดงได้ผ่านทางเส้นเชื่อมของมิติต่าง ๆ ซึ่งจากความสัมพันธ์ดังกล่าว เมื่อทำการประมวลผลคิวรีจากผู้ใช้จะทำให้ผลลัพธ์ที่ได้จะถูกแสดงผลผ่านคอลัมน์ แถว และเพจของลูกบาศก์ (cube) ที่ประกอบไปด้วยมิติทางธุรกิจ 3 มิติด้วยกัน

แต่เมื่อไรก็ตามที่ star schema มีมิติเพิ่มขึ้นจากเดิม เช่น เพิ่มขึ้นจาก 3 มิติเป็น 4 มิติ จะทำให้เราไม่สามารถแสดงผลลัพธ์ของการเชื่อมโยงข้อมูลผ่านลูกบาศก์ 3 มิติได้ ดังนั้นเมื่อ star schema มีมิติทางธุรกิจมากกว่า 3 มิติ เราควรจะใช้ **"Hypercubes"** ในการแสดงความเชื่อมโยงความสัมพันธ์ระหว่างมิติ ซึ่งจะสามารถอธิบายได้ดังนี้

“Hypercubes”

คืออะไร?

เพื่อที่จะอธิบายเกี่ยวกับ “hypercubes” เราจะทำการอธิบายโดยใช้ตัวอย่าง เพื่อให้เข้าใจได้โดยง่าย ถ้าเราทำการพิจารณา 2 มิติทางธุรกิจ คือ มิติรายการสินค้าและมิติแกนเวลาที่เกี่ยวข้องกับตัวชี้วัดต่าง ๆ ซึ่งจากข้อมูลทั้งหมด ค่าของข้อมูลในแต่ละส่วนจะถูกแสดงทางด้านขวาของรูปที่ 11-6 โดยข้อมูลในแกนเวลาจะประกอบไปด้วยรายชื่อเดือนเริ่มตั้งแต่เดือนมกราคมไปจนถึงเดือนธันวาคม ข้อมูลรายการสินค้าจะประกอบไปด้วย “Hats”, “Coats”, “Jackets”, “Dresses”, “Shirt” และ “Slacks” ตามลำดับ

ในส่วนของข้อมูลที่เป็นตัวชี้วัดจะประกอบไปด้วย เช่น
(1) ต้นทุนคงที่ (Fixed cost)
(2) ต้นทุนผันแปร (Variable cost)
(3) ยอดขายทางอ้อม (indirect sale)
(4) ยอดขายโดยตรง (direct sale)
(5) อัตรากำไร (Profit margin) ตามลำดับ

โดยจากรูปเราจะเห็นว่าข้อมูลจะถูกเก็บตามลำดับชั้นของข้อมูลในแต่ละมิติและถูกเรียงอยู่ในเส้นตรงตามมิติแกนเวลา มิติรายการสินค้า และตัวชี้วัด โดยในแต่ละเส้นเราสามารถเลื่อนขึ้น-ลงภายในเส้นได้และสามารถทำได้อย่างเป็นอิสระต่อกันด้วย ซึ่งการเลื่อนขึ้น-ลงเปรียบเสมือนกับการเรียกดูข้อมูลแต่ละส่วนที่เราต้องการ โดยการแสดงผลในลักษณะเส้นตรงดังกล่าวเราจะเรียกว่า *“Multidimensional domain structure (MDS)”* นอกจากนั้นในรูปที่ 11-6 ยังแสดงถึงการแสดงผลข้อมูลในรูปแบบตาราง โดยจากรูปจะเป็นการแสดงข้อมูลใน 1 เพจ

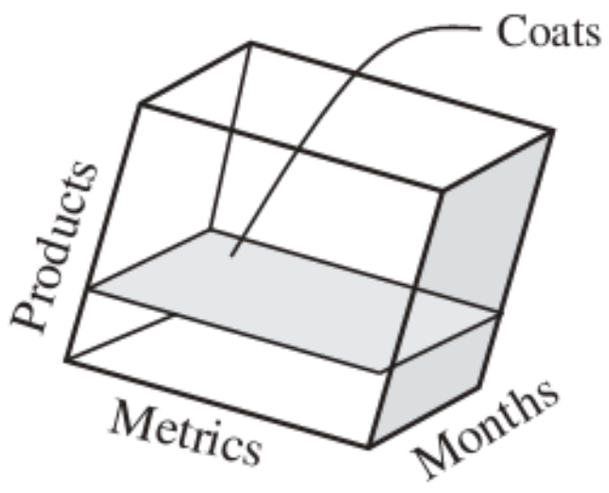
ซึ่งเป็นเพจของสินค้า “coats” ที่ถูกขายในเดือนต่าง ๆ (แกน X) โดยเทียบกับตัวชี้วัดต่าง ๆ (แกน Y) ถ้าเราต้องการที่จะเรียกดูข้อมูลของรายการสินค้าอื่น ๆ เราจะต้องทำการสร้างเพจใหม่สำหรับรายการสินค้านั้น ๆ นอกจากการแสดงผลในรูปตารางแล้วยังแสดงให้เห็นถึงการใช้ลูกบาศก์ในการแสดงผลข้อมูลทั้ง 3 เส้น ซึ่งจากรูป แกน Y คือการสินค้า แกน X คือ ตัวชี้วัด และแกน Z คือ ข้อมูลแต่ละเดือน ซึ่งถ้าเราต้องการที่จะเรียกดูข้อมูลของรายการสินค้า “coat” ในทุก ๆ เดือนกับทุก ๆ มาตรวัด เราจะต้องทำการหาจุดของข้อมูลในลูกบาศก์ที่เกี่ยวข้องกับรายการสินค้า “coat” เป็นต้น ซึ่งจากรูปจะแสดงถึงข้อมูลเพจหนึ่ง ๆ ที่เกิดจากการตัดทอนข้อมูลรายการสินค้าทั้งหมดให้เหลือเพียงหนึ่งรายการสินค้าเท่านั้น

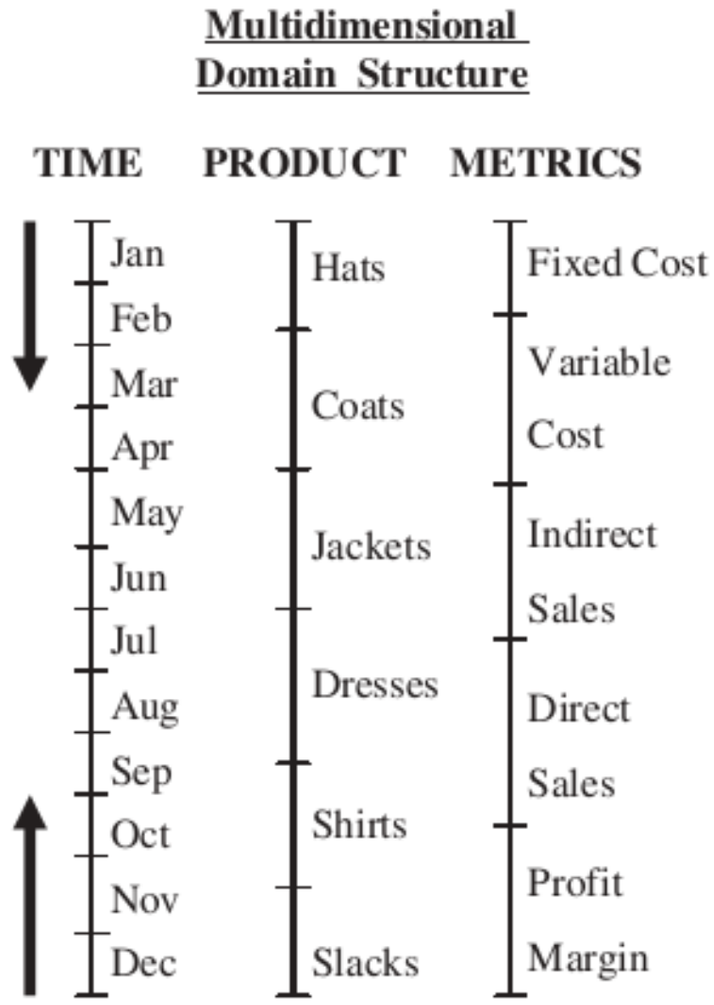
PRODUCT: Coats

PAGES: PRODUCT dimension COLUMNS: Metrics

ROWS: TIME dimension

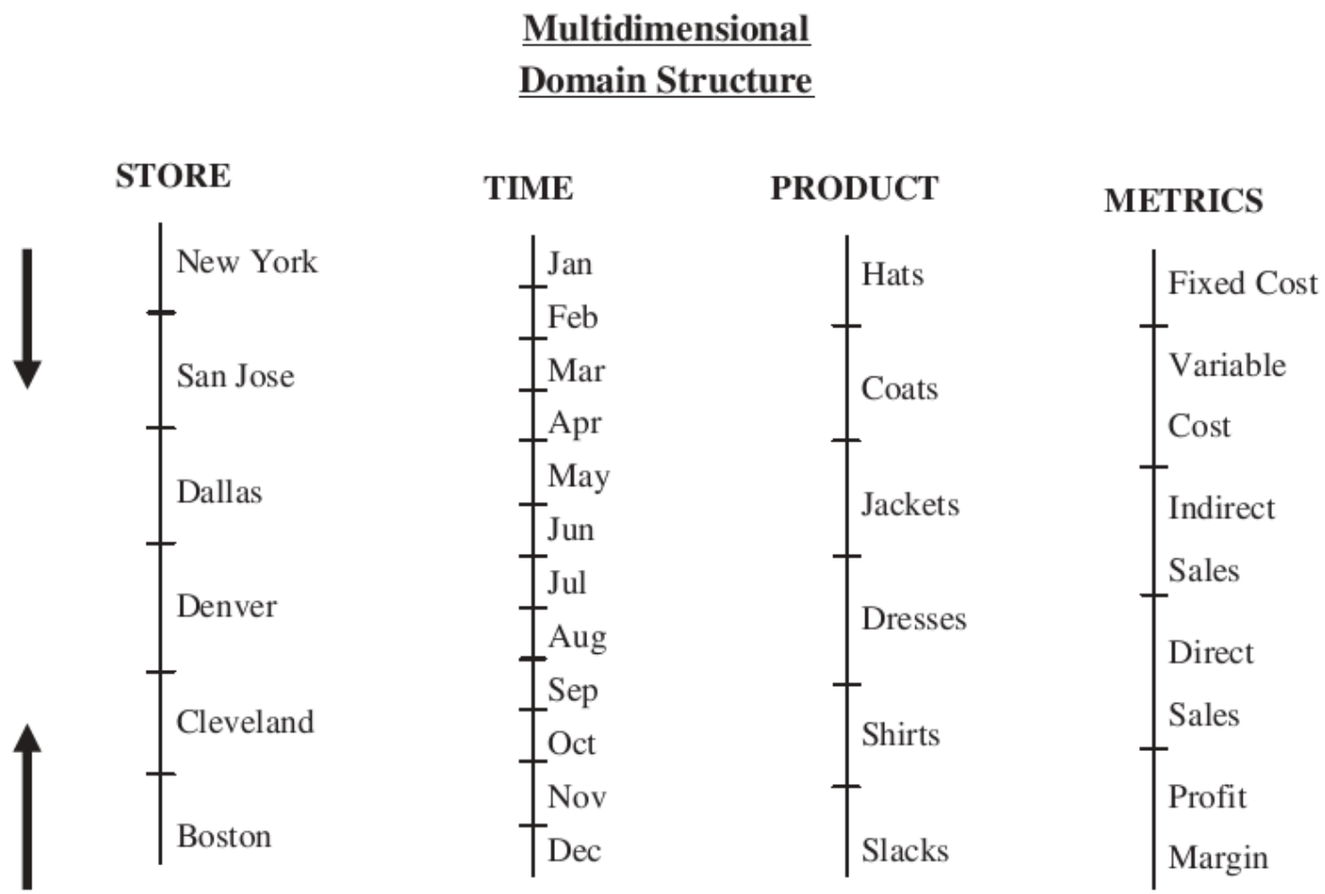
	Fixed Cost	Variable Cost	Indirect Sales	Direct Sales	Profit Margin
Jan	340	110	230	320	100
Feb	270	90	200	260	100
Mar	310	100	210	270	70
Apr	340	110	210	320	80
May	330	110	230	300	90
Jun	260	90	150	300	100
Jul	310	100	180	300	70
Aug	380	130	210	360	60
Sep	300	100	180	290	70
Oct	310	100	170	310	70
Nov	330	110	210	310	80
Dec	350	120	200	360	90





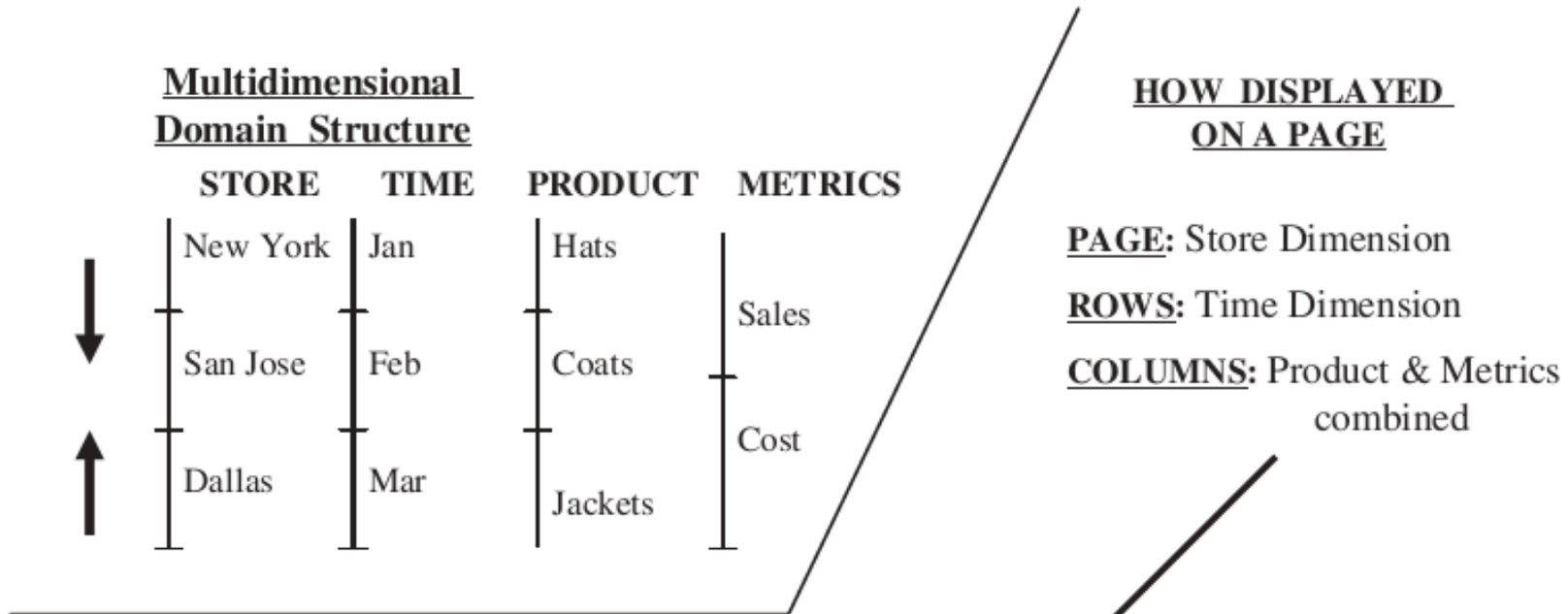
รูปที่ 11-6 การแสดงผลในรูปแบบของแถว คอลัมน์และเพจ

จากข้อมูล 2 มิติทางธุรกิจและ 1 กลุ่มตัวชี้วัด เราสามารถเปรียบได้เป็นการมีข้อมูล 3 มิติที่ซึ่งสามารถแสดงผลให้อยู่ในรูปของลูกบาศก์ได้ แต่เมื่อไหร่ก็ตามที่จำนวนมิติเพิ่มขึ้น เช่น ทำการเพิ่มมิติสาขาห้างร้านซึ่งเป็นมิติทางธุรกิจ จะทำให้เรามีข้อมูลทั้งหมด 4 มิติที่ประกอบไปด้วย 3 มิติทางธุรกิจ และ 1 กลุ่มตัวชี้วัด ซึ่งเราจะไม่สามารถใช้ลูกบาศก์ในการแสดงผลได้ ดังนั้นในการแสดงผลเราควรที่จะใช้ MDS diagram โดยการวาดเส้นตรงทั้งหมด 4 เส้น (ดังแสดงในรูปที่ 11-7)



รูปที่ 11-7 MDS สำหรับการวิเคราะห์ข้อมูล 4 มิติ

ซึ่งจากรูปเราจะสมมติว่าเส้นตรงเหล่านั้นเป็นลูกบาศก์ของข้อมูลหลายมิติ (Hypercubes) หลังจากที่เราทำการสร้าง hypercubes ที่อยู่ในรูปเส้นตรงแล้ว เราจะสามารถทำการแสดงผลบนหน้าจอของผู้ใช้ได้อย่างไร ? ซึ่งการแสดงผลในหน้าจอของผู้ใช้เราจะสามารถแสดงผลได้ในรูปแบบของแถว คอลัมน์ และเพจของข้อมูลเท่านั้น ลองพิจารณารูปที่ 11-8 ที่ประกอบไปด้วยข้อมูล 3 มิติทางธุรกิจและ 1 กลุ่มตัวชี้วัด ซึ่งในการแสดงผลเราจะกำหนดให้แกนแถวของข้อมูลแสดงข้อมูลมิติแกนเวลา คอลัมน์จะแสดงข้อมูลที่เกิดจากการรวมระหว่างมิติรายการสินค้าและตัวชี้วัดต่างๆ และท้ายสุดคือเพจจะแสดงข้อมูลที่ตั้งของสาขาต่าง ๆ ที่อยู่ในมิติสาขาห้างร้าน ซึ่งจากตัวอย่างมิติสาขาจะประกอบไปด้วยสาขาที่ตั้งอยู่ในรัฐนิวยอร์ก ซานโจเซ่ และดัลลัส ตามลำดับ โดยในการแสดงผลจะแสดงผลเพียงแค่เพจของสาขาที่อยู่ในรัฐนิวยอร์กเท่านั้น

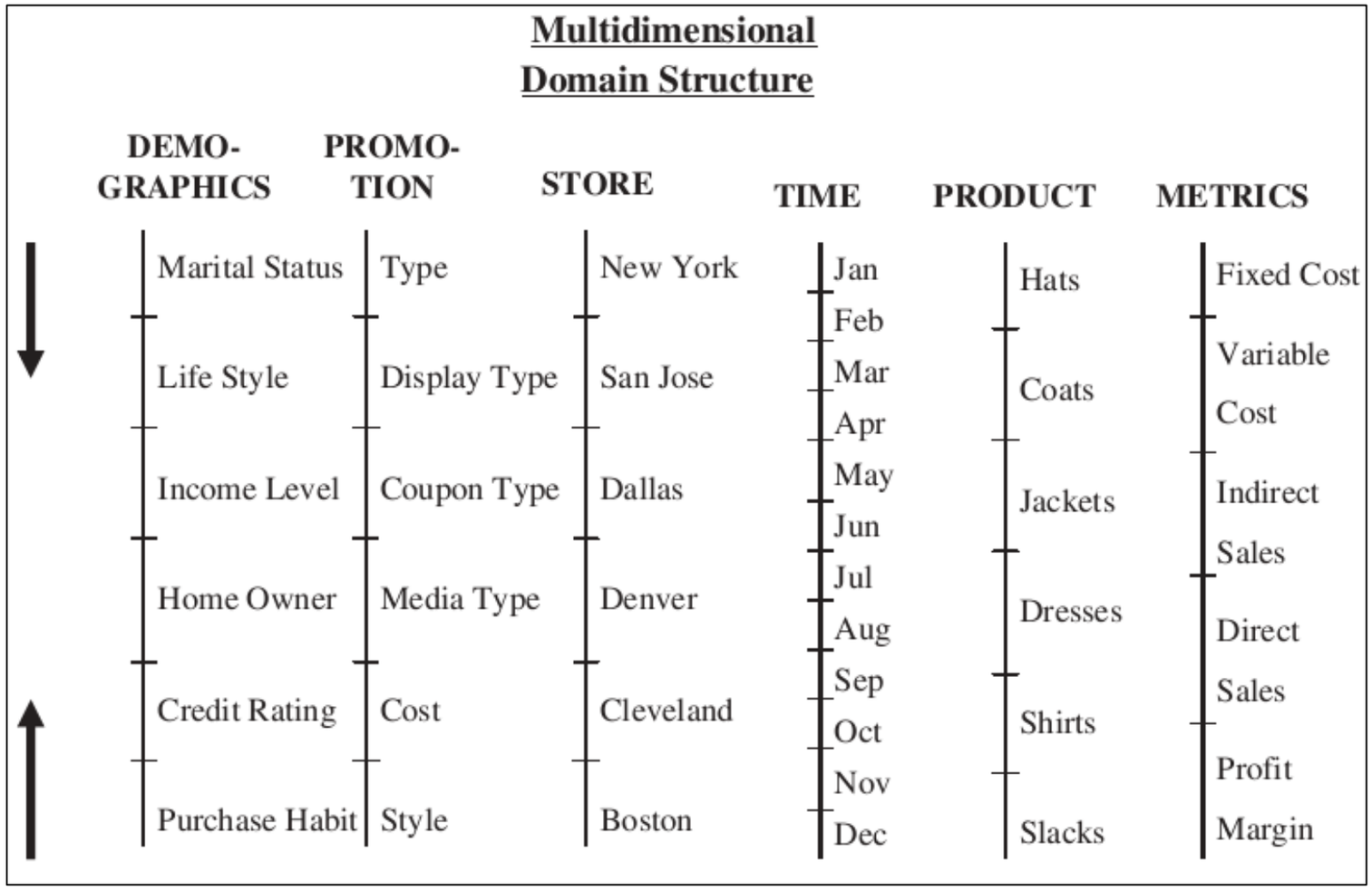


New York Store

	Hats:Sales	Hats:Cost	Coats:Sales	Costs:Cost	Jackets:Sales	Jackets:Cost
Jan	450	350	550	450	500	400
Feb	380	280	460	360	400	320
Mar	400	310	480	410	450	400

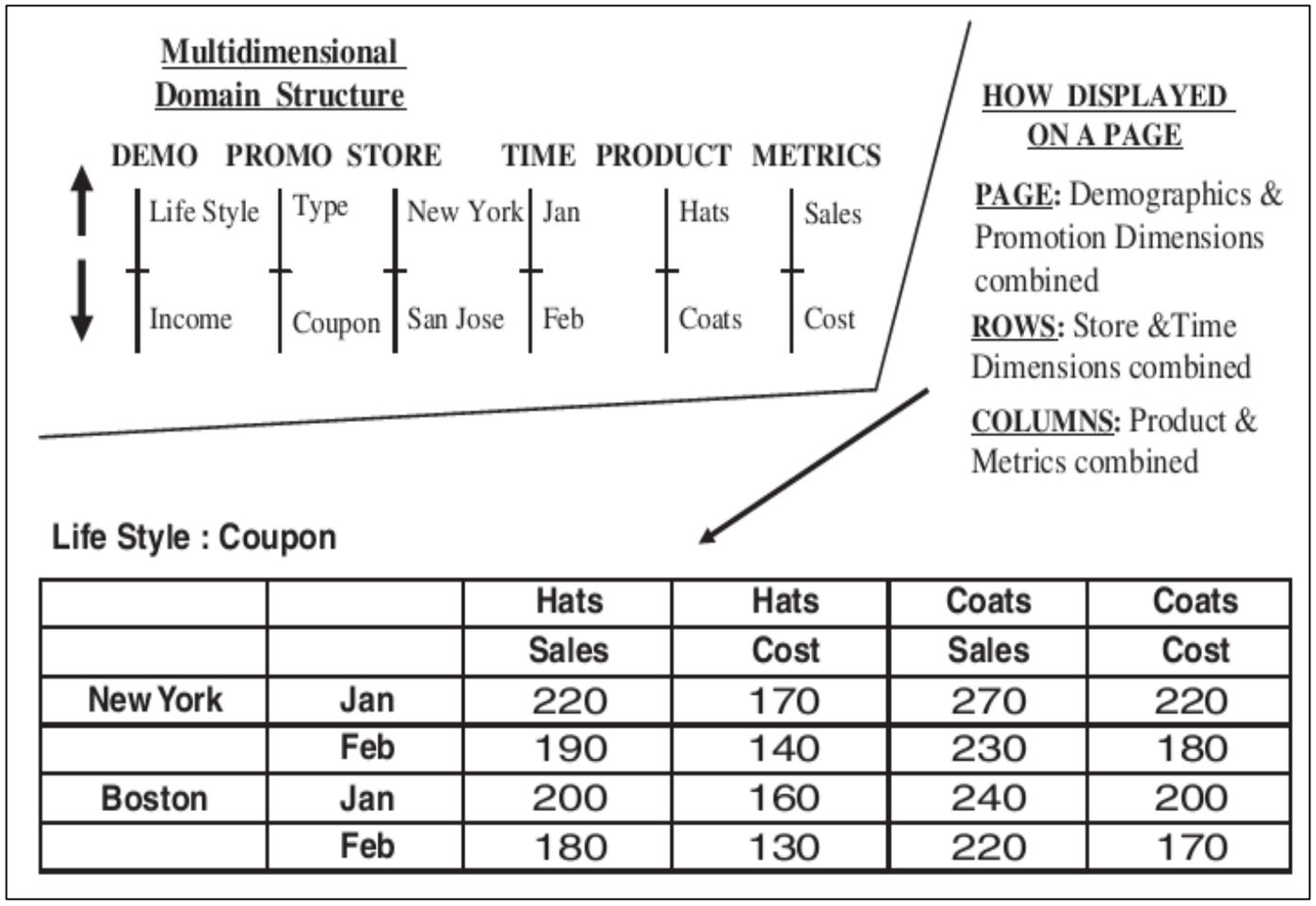
รูปที่ 11-8 ตัวอย่าง MDS สำหรับข้อมูล 4 มิติ

นอกเหนือจากข้อมูล 4 มิติแล้ว MDS ยังสามารถแสดงผลข้อมูลที่มีมากกว่า 4 มิติได้ ลองพิจารณาอีกตัวอย่างหนึ่งที่ประกอบไปด้วยข้อมูลทั้งสิ้น 6 มิติ ดังแสดงในรูปที่ 11-9 ที่ประกอบไปด้วย 5 มิติทางธุรกิจ ได้แก่ มิติข้อมูลลูกค้า (Demographics) มิติข้อมูลโปรโมชั่น มิติที่อยู่ของสาขาต่าง ๆ มิติแกนเวลา และมิติรายการสินค้า และ 1 กลุ่มตัวชี้วัด



รูปที่ 11-9 ตัวอย่าง MDS สำหรับข้อมูล 6 มิติ

โดยจากข้อมูลดังกล่าว เราจะแสดงผลของการวิเคราะห์ข้อมูลในรูปแบบของตารางข้อมูลได้ โดยนำข้อมูลรายการสินค้าและตัวชี้วัดรวมกันเป็นข้อมูลในคอลัมน์หนึ่งๆ นำข้อมูลสาขาต่าง ๆ รวมกับข้อมูลแกนเวลาเป็นแถวหนึ่ง ๆ และนำข้อมูลลูกค้ารวมกับข้อมูลโปรโมชั่นเป็นเพจหนึ่ง ๆ โดยการแสดงผลในรูปแบบตารางจะแสดงดังรูปที่ 11-10



Life Style : Coupon

		Hats	Hats	Coats	Coats
		Sales	Cost	Sales	Cost
New York	Jan	220	170	270	220
	Feb	190	140	230	180
Boston	Jan	200	160	240	200
	Feb	180	130	220	170

รูปที่ 11-10 ตัวอย่างข้อมูลเพจหนึ่งที่แสดงข้อมูลทั้ง 6 มิติ

จากที่กล่าวมาทั้งหมดข้างต้น เราจะสามารถสรุปเกี่ยวกับ **“Hypercubes”** ได้ 2 แง่มุมคือ ในการอธิบายเกี่ยวกับโมเดลของข้อมูลในระบบ OLAP ที่มีมากว่า 3 มิติ เราจะสามารถใช้ Multidimensional domain structure (MDS) ในการแสดงโครงสร้างต่าง ๆ ของข้อมูล และการแสดงผลข้อมูลบนหน้าจอของผู้ใช้ ซึ่งโดยส่วนใหญ่จะทำการแสดงผลข้อมูลในรูปแบบของ แถว คอลัมน์ และเพจของข้อมูล ซึ่งจะเป็นการแสดงผลข้อมูลที่อยู่ในรูปแบบของตาราง โดยเราสามารถแสดงผลข้อมูลที่มีมากกว่าหรือเท่ากับ 3 มิติให้อยู่ในรูปแบบของตารางได้

โดยหลังจากที่เราเข้าใจถึงโมเดลข้อมูล โครงสร้างข้อมูล การแสดงผลข้อมูลแล้ว เราควรที่จะต้องศึกษาถึงการดำเนินการวิเคราะห์ข้อมูลในระบบ OLAP ที่จะมีตัวดำเนินการหลายชนิดด้วยกัน เช่น การวิเคราะห์แบบเจาะลึก การวิเคราะห์ข้อมูลที่เป็นผลสรุป การวิเคราะห์ข้อมูลเพียงบางส่วน และการปรับเปลี่ยนมุมมองของข้อมูลในการวิเคราะห์ โดยรายละเอียดของแต่ละวิธีการวิเคราะห์ข้อมูลจะสามารถอธิบายได้ดังนี้

การวิเคราะห์แบบเจาะลึก
และการสร้างผลสรุปของข้อมูล
(Drill down and roll up)

การวิเคราะห์แบบเจาะลึกและการสร้างผลสรุปของข้อมูล (Drill down and roll up)

ในการที่จะทำความเข้าใจเกี่ยวกับการวิเคราะห์แบบเจาะลึกและการสร้างผลสรุปของข้อมูล ลองพิจารณาตัวอย่างในรูปที่ 11-4 ที่แสดงถึง STAR schema ที่ประกอบไปด้วย 3 มิติทางธุรกิจ นั่นคือ

1. มิติรายการสินค้าที่ประกอบไปด้วย ชื่อรายการสินค้า ชนิดสินค้า หมวดหมู่สินค้า สายการผลิตสินค้า และแผนกของสินค้า ตามลำดับ
2. มิติแกนเวลา
3. มิติสาขาต่างๆ

ซึ่งจากข้อมูลทั้งสามมิติ เราจะเห็นว่าข้อมูลแอทริบิวต่าง ๆ ในแต่ละมิติจะถูกจัดเก็บอยู่ในรูปแบบของลำดับชั้น (Hierachical sequence) ที่มีความเกี่ยวเนื่องกัน ซึ่งจากการจัดเก็บข้อมูลดังกล่าวจะทำให้ระบบ OLAP สามารถทำการวิเคราะห์ข้อมูลแบบเจาะลึก และสามารถทำการสร้างผลสรุปของข้อมูลได้ โดยในการที่จะทำความเข้าใจเกี่ยวกับการวิเคราะห์ข้อมูลทั้งสองแบบ ลองพิจารณารูปที่ 11-11 ที่จะแสดงเกี่ยวกับความสามารถในการเรียกดูข้อมูลที่มีความละเอียดต่างกันไล่ไปตามลำดับชั้น

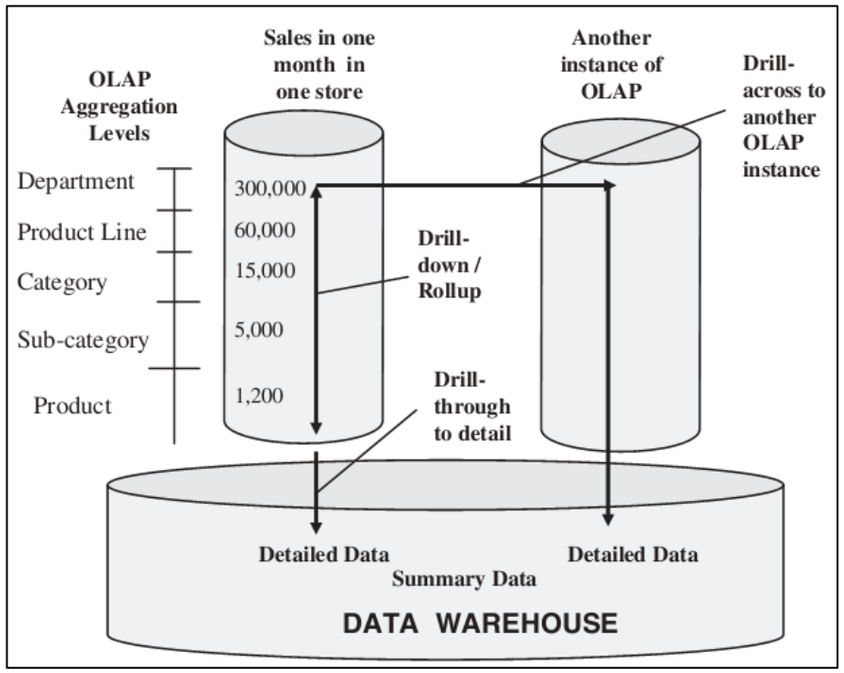
ซึ่งจากรูปจะเป็นข้อมูลยอดขายในหนึ่งเดือนที่เกิดขึ้นในสาขาหนึ่ง ๆ โดยทางซ้ายมือสุดจะเป็นลำดับชั้นความละเอียดของข้อมูลมิติรายการสินค้าที่จะไล่จากข้อมูลที่มีความละเอียดน้อยไปจนถึงข้อมูลที่มีความละเอียดสูงสุด โดยแต่ข้อมูลในแต่ละระดับจะมียอดขายแนบอยู่ด้วย เช่น

ยอดขายสินค้าในแผนกหนึ่งเป็นจำนวน 300,000$ ยอดขายสินค้าในหมวดหมู่สินค้าหนึ่ง ๆ เป็นจำนวน 15,000$ และ ยอดขายสินค้ารายการหนึ่ง ๆ เป็นจำนวน 1,200$ เป็นต้น

ซึ่งจากข้อมูลดังกล่าว จะทำให้ผู้ใช้สามารถเลือกดูข้อมูลตามแต่ละระดับได้ โดยถ้าผู้ใช้ทำการเลือกดูข้อมูลยอดขายในหมวดหมู่สินค้าหนึ่ง แล้วทำการเปลี่ยนไปเลือกดูข้อมูลที่มีความละเอียดมากขึ้นนั่นคือ ข้อมูลยอดขายในประเภทสินค้าหนึ่ง เราจะเรียกว่าเป็นการวิเคราะห์ข้อมูลแบบเจาะลึก แต่ถ้าผู้ใช้ทำการเลือกดูข้อมูลยอดขายในหมวดหมู่สินค้าหนึ่ง แล้วขยับขึ้นไปดูข้อมูลที่มี ความละเอียดน้อยลง ซึ่งก็คือ ยอดขายของสายการผลิตสินค้าหนึ่ง ๆ เราจะเรียกว่าเป็นการวิเคราะห์ข้อมูลแบบผลสรุป

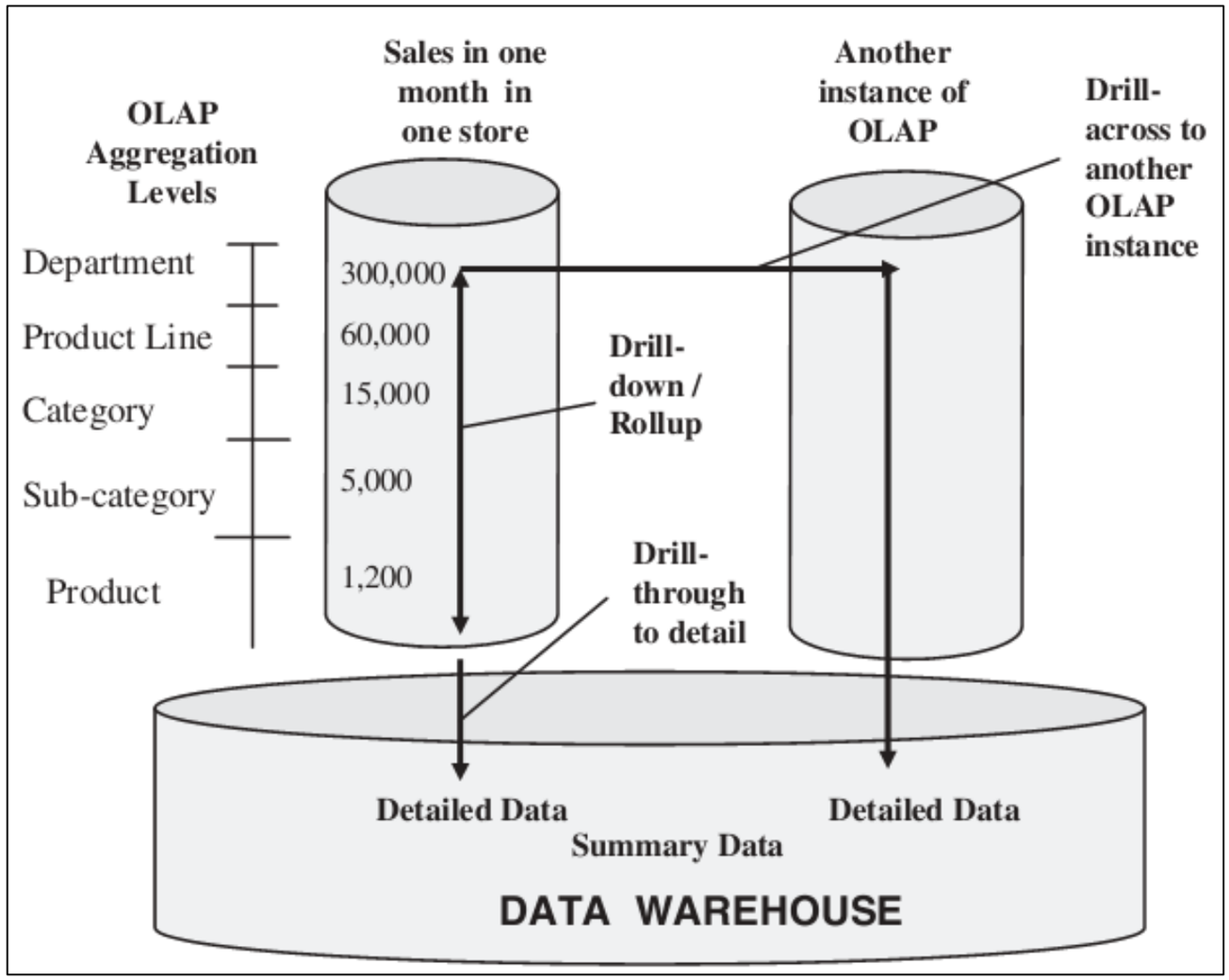
นอกจากการวิเคราะห์ทั้งสองแบบแล้ว ในรูปยังแสดงถึงการวิเคราะห์ข้อมูลแบบเจาะลึกและผลสรุปข้ามมิติอีกด้วย ซึ่งจะเป็นการวิเคราะห์แบบเจาะลึกหลาย ๆ มิติพร้อม ๆ กัน หรือไม่พร้อมกันก็ได้ เช่น ผู้ใช้อาจทำการเลือกดูข้อมูลยอดขายสินค้าในหมวดหมู่หนึ่ง ๆ ที่เกิดขึ้นในเดือนหนึ่ง ๆ ต่อมาอาจจะเปลี่ยนเป็นยอดขายสินค้าแต่ละรายการสินค้าในแต่ละไตรมาสก็เป็นได้ ซึ่งการเรียกดูนี้จะเป็นการเจาะลึกลงไปรายละเอียดของมิติรายการสินค้า และเป็นการเรียกดูผลสรุปในมิติแกนเวลา ซึ่งจากที่กล่าวมาทั้งหมดระบบ OLAP จะสามารถวิเคราะห์ข้อมูลได้หลายรูปแบบทั้งแบบข้ามมิติ และไม่ข้ามมิติ

แต่อย่างไรก็ดีจากการวิเคราะห์ข้อมูลทั้งแบบเจาะลึกและผลสรุป เราจะทำการแสดงผลได้อย่างไร ลองพิจารณารูปที่ 11-6 อีกครั้งหนึ่งที่จะแสดงถึงการแสดงผลข้อมูลในรูปแบบของตาราง ซึ่งจากรูปจะแสดงข้อมูลรายการสินค้าในแต่ละคอลัมน์ แสดงข้อมูลเดือนต่าง ๆ ในแต่ละแถวข้อมูล และแสดงสาขาหนึ่ง ๆ ในแต่ละเพจ แต่ถ้าจากรูปเราทำการวิเคราะห์แบบผลสรุปโดยทำการเปลี่ยนการเรียกดูข้อมูลจากแต่ละรายการสินค้าไปเป็นหมวดหมู่สินค้า เราจะสามารถแสดงได้ดังรูปที่ 11-12 ซึ่งจะทำการเปลี่ยนข้อมูลรายการสินค้าที่แสดงในแต่ละคอลัมน์ไปเป็นหมวดหมู่ของสินค้าแทน โดยข้อมูลที่แสดงในแต่ละแถวและแต่ละเพจยังคงเดิม แต่เมื่อไหร่ก็ตามที่เราต้องการเปลี่ยนการเรียกดูข้อมูลที่แกนของสาขาด้วย โดยปรับระดับให้มีความละเอียดมากขึ้นจากเดิมเป็นแต่ละรัฐหรือเป็นแต่ละเมือง เราจะสามารถแสดงผลข้อมูลเดือนในแต่ละแถวข้อมูลได้เหมือนเดิม แสดงข้อมูลแต่ละหมวดหมู่สินค้าในแต่ละคอลัมน์ และทำการปรับเปลี่ยนเพจจากแต่ละรัฐเป็นแต่ละเมืองแทน ตามลำดับ



Store: New York
PAGES: STORE dimension
Sub-categories
COLUMNS: PRODUCT dimension

ROWS: TIME dimension (Months)	Outer	Dress	Casual
Jan	1,100	1,020	490
Feb	1,080	1,040	500
Mar	1,050	980	470
Apr	970	1,000	480
May	1,010	1,080	520
Jun	910	1,100	330
Jul	880	1,120	250
Aug	960	1,320	230
Sep	870	1,280	210
Oct	910	1,240	250
Nov	980	1,380	260
Dec	1,080	1,520	310



รูปที่ 11-11 การวิเคราะห์แบบเจาะลึกและการสร้างผลสรุปของข้อมูล

Store: New York

Sub-categories

PAGES: STORE dimension

COLUMNS: PRODUCT dimension

ROWS: TIME dimension	Outer	Dress	Casual
Jan	1,100	1,020	490
Feb	1,080	1,040	500
Mar	1,050	980	470
Apr	970	1,000	480
May	1,010	1,080	520
Jun	910	1,100	330
Jul	880	1,120	250
Aug	960	1,320	230
Sep	870	1,280	210
Oct	910	1,240	250
Nov	980	1,380	260
Dec	1,080	1,520	310

Months

รูปที่ 11-12 การแสดงผลสรุปของข้อมูลใน 3 มิติ

การวิเคราะห์ข้อมูลเพียงบางส่วนและการปรับเปลี่ยนมุมมองของข้อมูล (Slice and dice)

การวิเคราะห์ข้อมูลเพียงบางส่วนและการปรับเปลี่ยนมุมมองของข้อมูล (Slice and dice)

ในการที่จะทำความเข้าใจเกี่ยวกับการวิเคราะห์ข้อมูลเพียงบางส่วนและการปรับเปลี่ยนมุมมองของข้อมูล ลองพิจารณาตัวอย่างในรูปที่ 11-5 ที่เป็นการแสดงข้อมูลเดือนต่าง ๆ ในแถวต่าง ๆ รายการสินค้าในคอลัมน์ต่าง ๆ และสาขาในเพจต่าง ๆ โดยแต่ละเพจจะแสดงยอดขายทุกเดือนและรายการสินค้าในสาขาหนึ่ง ๆ

ซึ่งจากรูปจะเป็นยอดขายสินค้าของสาขาในรัฐนิวยอร์ก ซึ่งเพจหนึ่ง ๆ จะเป็นการแสดงผลข้อมูลเพียงบางส่วน (Slice) ที่แสดงข้อมูลเพียง 2 มิติเท่านั้น

Store: New York

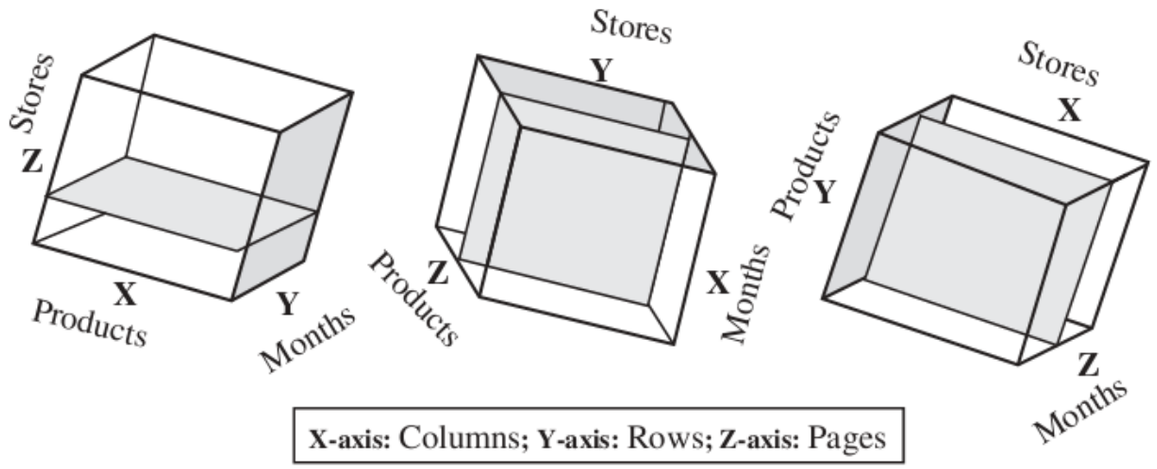
PAGES: STORE dimension

Products

COLUMNS: PRODUCT dimension

ROWS: TIME dimension / Months	Hats	Coats	Jackets	Dresses	Shirts	Slacks
Jan	200	550	350	500	520	490
Feb	210	480	390	510	530	500
Mar	190	480	380	480	500	470
Apr	190	430	350	490	510	480
May	160	530	320	530	550	520
Jun	150	450	310	540	560	330
Jul	130	480	270	550	570	250
Aug	140	570	250	650	670	230
Sep	160	470	240	630	650	210
Oct	170	480	260	610	630	250
Nov	180	520	280	680	700	260
Dec	200	560	320	750	770	310

รูปที่ 11-5 การแสดงผลในหน้าหนึ่งในรูปแบบสเปรดชีตของข้อมูล 3 มิติ



Store: New York

	Hats	Coats	Jackets
Jan	200	550	350
Feb	210	480	390
Mar	190	480	380

Product: Hats

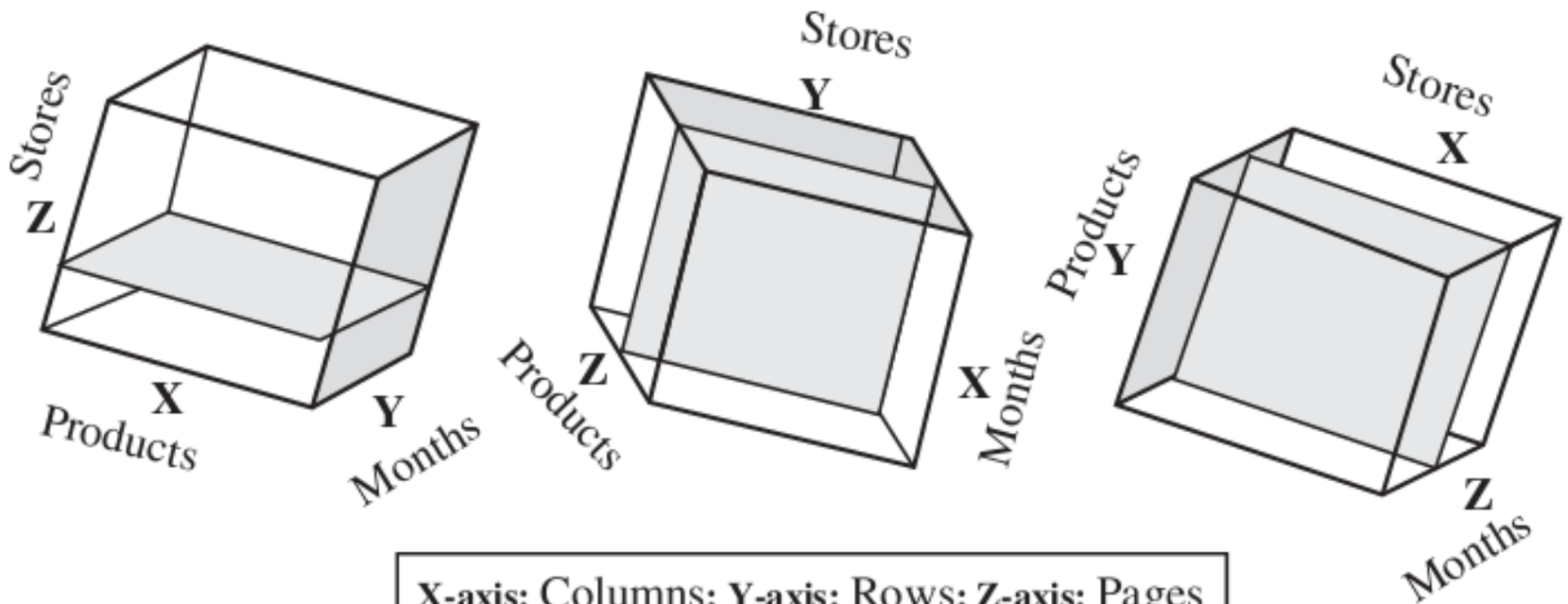
	Jan	Feb	Mar
New York	200	210	190
Boston	210	250	240
San Jose	130	90	70

Month: January

	New York	Boston	San Jose
Hats	200	210	130
Coats	550	500	200
Jackets	350	400	100

รูปที่ 11-13 การวิเคราะห์ข้อมูลเพียงบางส่วน และการปรับเปลี่ยนมุมมองข้อมูล

ซึ่งจากการเลือกดูข้อมูลเพียงบางส่วน ลองพิจารณาถึงการปรับเปลี่ยนมุมมองของข้อมูลดังแสดงในรูปที่ 11-13 ที่ซึ่งรูปทางซ้ายมือจะแสดงข้อมูลดังแสดงในรูปที่ 11-5 รูปตรงกลางจะเปลี่ยนการหมุนแกนของข้อมูล โดยจะเป็นการเปลี่ยนรายการสินค้าจากการแสดงผลในคอลัมน์เป็นการแสดงผลในเพจ การแสดงผลข้อมูลสาขาจากเพจไปเป็นแถว และข้อมูลเดือนจากแถวไปเป็นคอลัมน์ โดยจากรูปจะเป็นการแสดงผลยอดขายของรายการสินค้า “Hats” ในเดือนต่าง ๆ ที่ขายได้ในสาขาต่าง ๆ และท้ายสุดรูปทางขวามือจะเป็นการหมุนแกนของข้อมูลโดยทำการแสดงผลข้อมูลเดือนต่าง ๆ ในแต่ละเพจ แต่ละรายการสินค้าในแถวต่าง ๆ และสาขาต่าง ๆ ในคอลัมน์ โดยจะเป็นการแสดงผลข้อมูลยอดขายรายการสินค้าต่างๆที่ขายได้ในสาขาต่าง ๆ ในเดือนมกราคม ซึ่งจากความสามารถในการเลือกข้อมูลเพียงบางส่วนและการปรับเปลี่ยนมุมมองของข้อมูลจะทำให้ผู้ใช้สามารถมองข้อมูลได้หลายๆมุมและสามารถทำความเข้าใจกับข้อมูลเหล่านั้นได้ง่ายขึ้น



X-axis: Columns; **Y-axis:** Rows; **Z-axis:** Pages

Store: New York

	Hats	Coats	Jackets
Jan	200	550	350
Feb	210	480	390
Mar	190	480	380

Product: Hats

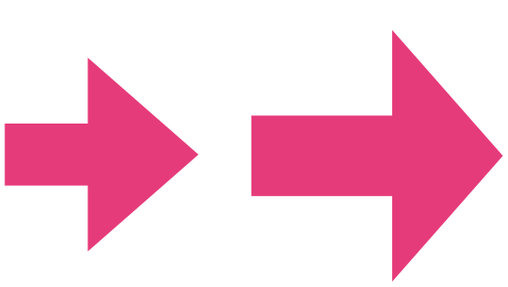
	Jan	Feb	Mar
New York	200	210	190
Boston	210	250	240
San Jose	130	90	70

Month: January

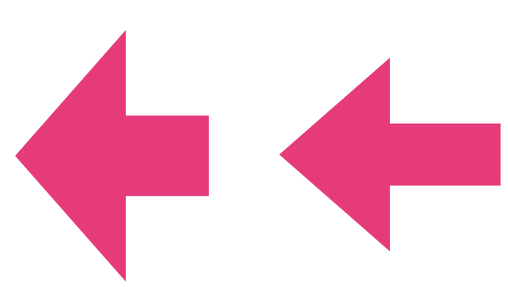
	New York	Boston	San Jose
Hats	200	210	130
Coats	550	500	200
Jackets	350	400	100

รูปที่ 11-13 การวิเคราะห์ข้อมูลเพียงบางส่วนและการปรับเปลี่ยนมุมมองข้อมูล

การใช้งานและประโยชน์ของ



OLAP



การใช้งานและประโยชน์ของ OLAP

หลังจากทำการพิจารณาเกี่ยวกับคุณลักษณะต่าง ๆ ของ OLAP แล้ว เราจะทราบถึงประโยชน์ของ OLAP ที่สามารถทำการวิเคราะห์ข้อมูลในมิติต่าง ๆ ที่มีความซับซ้อนได้ โดยเราสามารถสรุปประโยชน์ของการใช้ระบบ OLAP ได้ดังนี้

- การเพิ่มผลผลิต/ประสิทธิภาพในการทำงานของผู้จัดการ และนักวิเคราะห์เชิงธุรกิจ
- ระบบ OLAP มีความยืดหยุ่นที่จะทำให้ผู้ใช้สามารถทำการสั่งประมวลผลคิวรีต่าง ๆ ได้ด้วยตนเอง โดยไม่ต้องการความช่วยเหลือจากฝ่ายไอทีแต่อย่างใด
- เมื่อทำการรวมระบบ OLAP ไว้เป็นส่วนหนึ่งของคลังข้อมูลจะทำให้การสร้างระบบเป็นไปอย่างรวดเร็ว
- ระบบคลังข้อมูลสามารถทำงานได้อย่างรวดเร็วโดยไม่มีงานคั่งค้าง
- สามารถลดเวลาที่ใช้ในการประมวลผลคิวรีต่าง ๆ และลดการใช้แบนด์วิดท์ของเครือข่าย

SECTION 6

โมเดลต่าง ๆ ของ OLAP

โมเดลต่าง ๆ ของ OLAP

ในการสร้าง OLAP เราจะสามารถเลือกวิธีในการจัดเก็บข้อมูลได้หลายวิธี เช่น

ROLAP

(Relational OnLine Analytical Processing) ที่เป็นระบบ OLAP ที่สร้างขึ้นโดยใช้ relational database

MOLAP

(Multidimensional OnLine Analytical Processing) ที่เป็นระบบ OLAP ที่สร้างขึ้นโดยใช้ multidimensional database

HOLAP

(Hybrid OnLine Analytical Processing) จะเป็นระบบที่นำข้อดีและคุณลักษณะเด่นต่างๆของ ROLAP และ MOLAP มารวมกัน

DOLAP

(Desktop OnLine Analytical Processing) จะเป็นระบบที่พัฒนามาจาก ROLAP ที่มีความสามารถที่จะทำให้ผู้ใช้สามารถพกพาระบบ OLAP ไปใช้งานในที่ต่าง ๆ ได้ โดย DOLAP จะทำการสร้างข้อมูลที่มีหลายมิติแล้วทำการส่งข้อมูลเหล่านั้นไปยังเครื่องของผู้ใช้ (desktop) ที่มีการติดตั้งซอร์ฟแวร์ DOLAP สำหรับใช้งานไว้แล้ว

Database OLAP

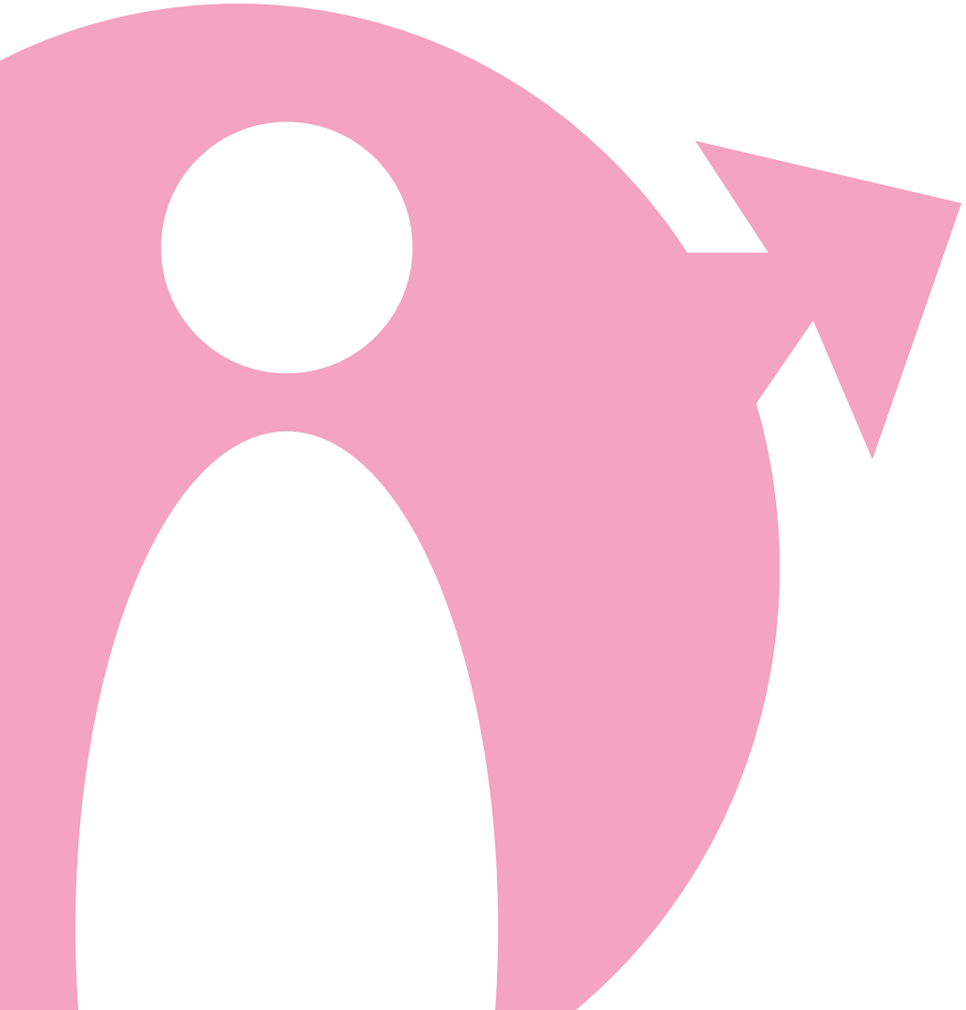
จะเป็นระบบจัดการฐานข้อมูลแบบ relational database management system (RDBMS) ที่ถูกสร้างหรือกำหนดให้สนับสนุนการทำงานของ OLAP และเอื้อต่อการคำนวณต่าง ๆ ของ OLAP ด้วย

Web OLAP

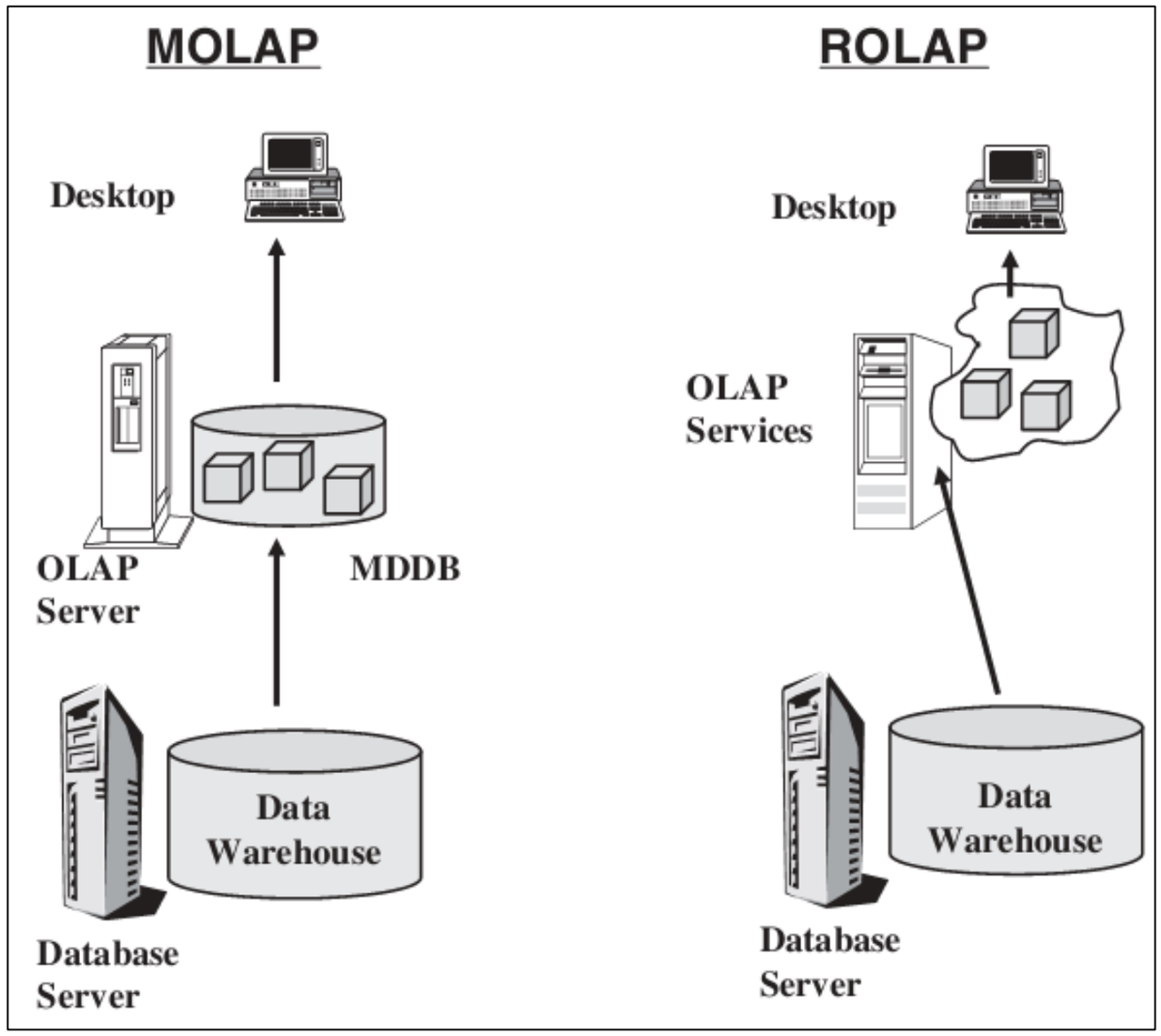
จะเป็นระบบ OLAP ที่สามารถใช้งานผ่านเว็บเบราเซอร์ได้

จากวิธีการจัดเก็บข้อมูลทั้งหมดข้างต้น ROLAP และ MOLAP จะเป็นโมเดลพื้นฐานของการสร้างระบบ OLAP

โดยที่ระบบ MOLAP จะเป็นระบบที่เหมาะกับการนำไปใช้มากที่สุดเนื่องจากมีการจัดเก็บข้อมูลในหลายมิติ ซึ่งจะช่วยให้ผู้ใช้สามารถมองหรือเรียกใช้ข้อมูลได้หลายมุมมอง แต่ในขณะที่ ROLAP จะเป็นระบบที่ใช้ RDBMS ในการจัดเก็บข้อมูล ซึ่งจะให้ OLAP ที่สร้างขึ้นนั้นมีคุณสมบัติตามคุณสมบัติพื้นฐานของ RDBMS เท่านั้น



เพื่อให้เข้าใจถึงความแตกต่างระหว่าง ROLAP และ MOLAP มากขึ้น ลองพิจารณารูปที่ 11-14 ที่จะแสดงการเปรียบเทียบกันระหว่างสถาปัตยกรรมของ MOLAP และ ROLAP ที่มีความแตกต่างกัน



รูปที่ 11-14 ตัวอย่างสถาปัตยกรรมของ MOLAP และ ROLAP

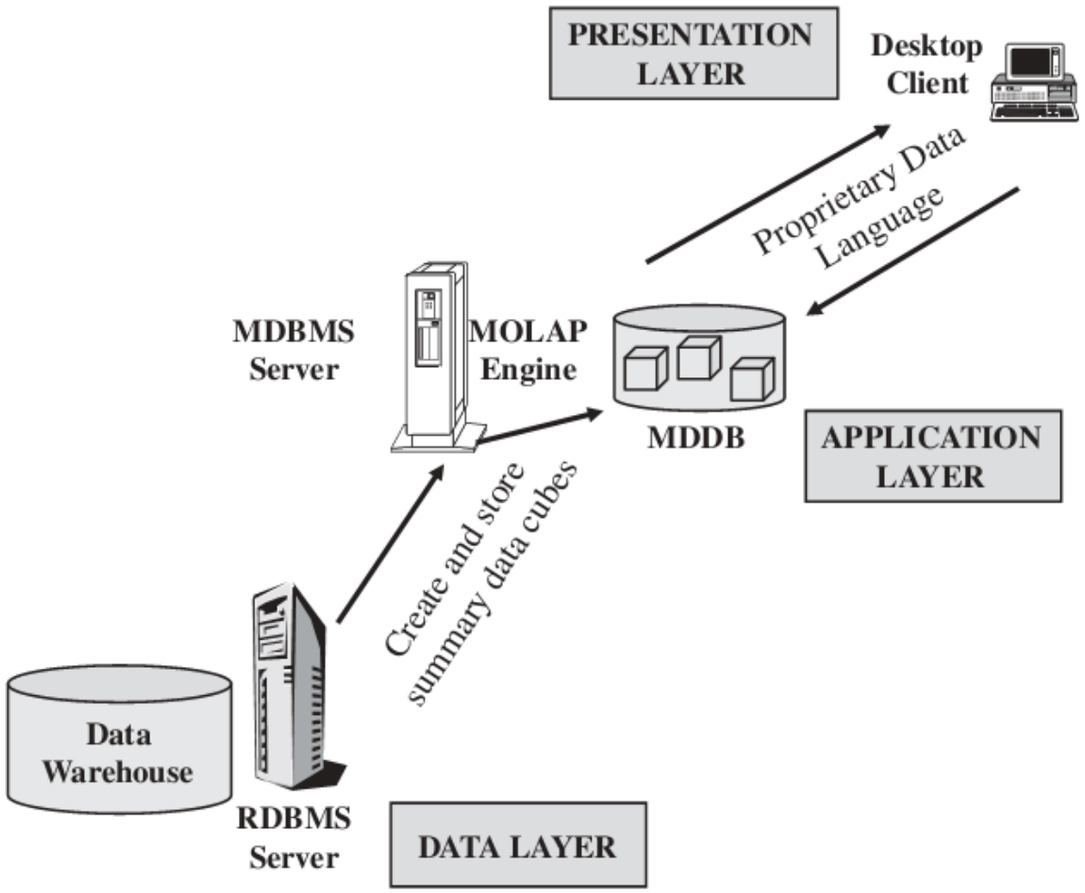
คือ ในระบบ MOLAP จะมีการใช้เซิร์ฟเวอร์ต่างหากสำหรับติดตั้งระบบ OLAP ที่มีการจัดเก็บข้อมูลโดยใช้ MDBMS ที่จะเก็บข้อมูลอยู่ในรูปแบบของลูกบาศก์หลายมิติ (Multidimensional cubes) ซึ่งจากสถาปัตยกรรมดังกล่าว จะทำให้เราต้องทำการสร้างกระบวนการในการสกัดและรวบรวมข้อมูลจากฐานข้อมูลของคลังข้อมูลที่ใช้ RDBMS จากนั้นทำการสร้างลูกบาศก์หลายมิติแล้วทำการเก็บไว้ใน MDBMS ที่อยู่ในเซิร์ฟเวอร์ที่เราแยกการทำงานไว้แล้ว แต่ในส่วนของระบบ ROLAP จะทำการติดตั้งระบบ OLAP ไว้ในแต่ละเครื่องของผู้ใช้และจะไม่มีการสร้างลูกบาศก์หลายมิติ ซึ่งในการแสดงผลจะทำการเรียกข้อมูลจาก RDBMS แล้วทำการแสดงผลให้มีความคล้ายคลึงกับลูกบาศก์หลายมิติ โดยในการแสดงผลในลักษณะดังกล่าวจะต้องใช้การประมวลผลเพิ่มเติม

โมเดล MOLAP

M OLAP

จะเป็น OLAP โมเดลที่ทำการเก็บข้อมูลไว้ใน “multidimensional database” ซึ่งโครงสร้างการจัดเก็บข้อมูลเป็นแบบอะเรย์หลายมิติ (multidimensional array)

ตัวอย่าง เช่น ในการจัดเก็บข้อมูลยอดขายสินค้า A เป็นจำนวน 500 ชิ้น ที่ถูกขายในเดือนมกราคมปี 2012 (2012/01) และขายได้ในสาขาที่ชื่อว่า Store-1 ซึ่งข้อมูลรายการขายทั้ง 500 ชิ้นนี้จะถูกเก็บอยู่ในอะเรย์ที่มีค่าเป็น (Product A, 2012/01/, Store-S1) เมื่อเราทำการจัดเก็บข้อมูลค่า ๆ ต่างไว้ในอะเรย์ เราจะสามารถนำข้อมูลเหล่านี้มารวมกัน เพื่อระบุถึงข้อมูลที่เป็นตัวชี้วัดได้ (จำนวนชิ้นสินค้าที่ขายได้) แต่อย่างไรก็ดี ไม่ใช่ว่าทุกค่าที่นำมารวมกันของแอทริบิวต่าง ๆ จะมีข้อมูลที่เป็นตัวชี้วัดเสมอไป ในบางข้อมูลอาจจะไม่มีตัวชี้วัดเลยก็เป็นได้ ตัวอย่างเช่น ถ้าห้างร้านค้าไม่เปิดให้บริการวันอาทิตย์ จะทำให้ข้อมูลยอดขายสินค้าทุกชนิด ทุกสาขา ในทุก ๆ วันอาทิตย์จะมีค่าเป็น 0 หรือ NULL เป็นต้น



รูปที่ 11-15 สถาปัตยกรรมของโมเดล MOLAP

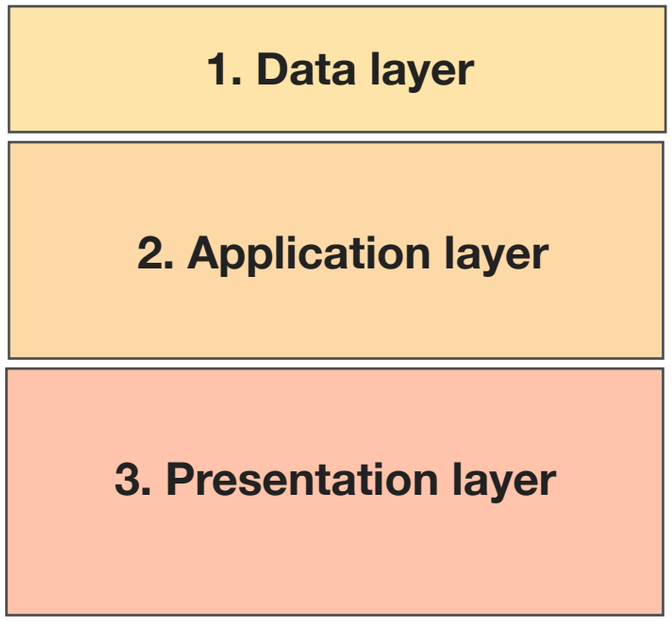
การจัดเก็บข้อมูลลงใน multidimensional database ของ OLAP จะเป็นอีกฐานข้อมูลหนึ่งที่แยกออกมาจากฐานข้อมูลของคลังข้อมูล เมื่อเราทำการพิจารณาสถาปัตยกรรมดังแสดงในรูปที่ 11-15 จะทำให้เห็นภาพคร่าว ๆ ของการทำงานของ MOLAP ที่สามารถแบ่งสถาปัตยกรรมของ MOLAP ได้เป็น 3 ลำดับชั้น คือ (1) Data layer—ฐานข้อมูลของคลังข้อมูล (2) application layer—ส่วนที่ใช้จัดเก็บข้อมูล ในรูปแบบหลาย ๆ มิติ ซึ่งในส่วนนี้จะประกอบไปด้วยลูกบาศก์ต่าง ๆ ที่มีการคำนวณและสร้างไว้แล้ว โดยลูกบาศก์เหล่านี้จะถูกเก็บไว้ใน multidimensional database และ (3) Presentation layer—จะเป็นส่วนที่ใช้ในการเชื่อมต่อกับผู้ใช้งาน ตามลำดับ

โมเดล ROLAP
(ROLAP model)

ROLAP

ในโมเดล ROLAP จะทำการเก็บข้อมูลในรูปแบบของแถวและคอลัมน์เหมือนกับ relational data model แต่เมื่อไรก็ตามที่ต้องทำการแสดงผลให้กับผู้ใช้ จะทำการแสดงผลข้อมูลในลักษณะของมิติเชิงธุรกิจต่าง ๆ หลายมิติ ซึ่งจะต้องทำการแปลงข้อมูลที่อยู่ในรูปแบบของแถวและคอลัมน์ก่อน โดยในโมเดล ROLAP จะมีการจัดเก็บข้อมูลที่เป็นเมตาดาต้าสำหรับใช้ในการเชื่อมโยง (แปลง) ข้อมูลที่อยู่ในรูปแบบของแถวและคอลัมน์ไปยังมิติเชิงธุรกิจต่าง ๆ เมตาดาต้าเหล่านี้ยังมีส่วนช่วยในการทำการรวบรวม (aggregations) และการทำผลสรุปของข้อมูล (summrizations) ด้วย โดยที่เมตาดาต้าเหล่านี้อาจถูกเก็บไว้ในฐานข้อมูลแบบ relational database หรือแบบอื่น ๆ ก็ได้

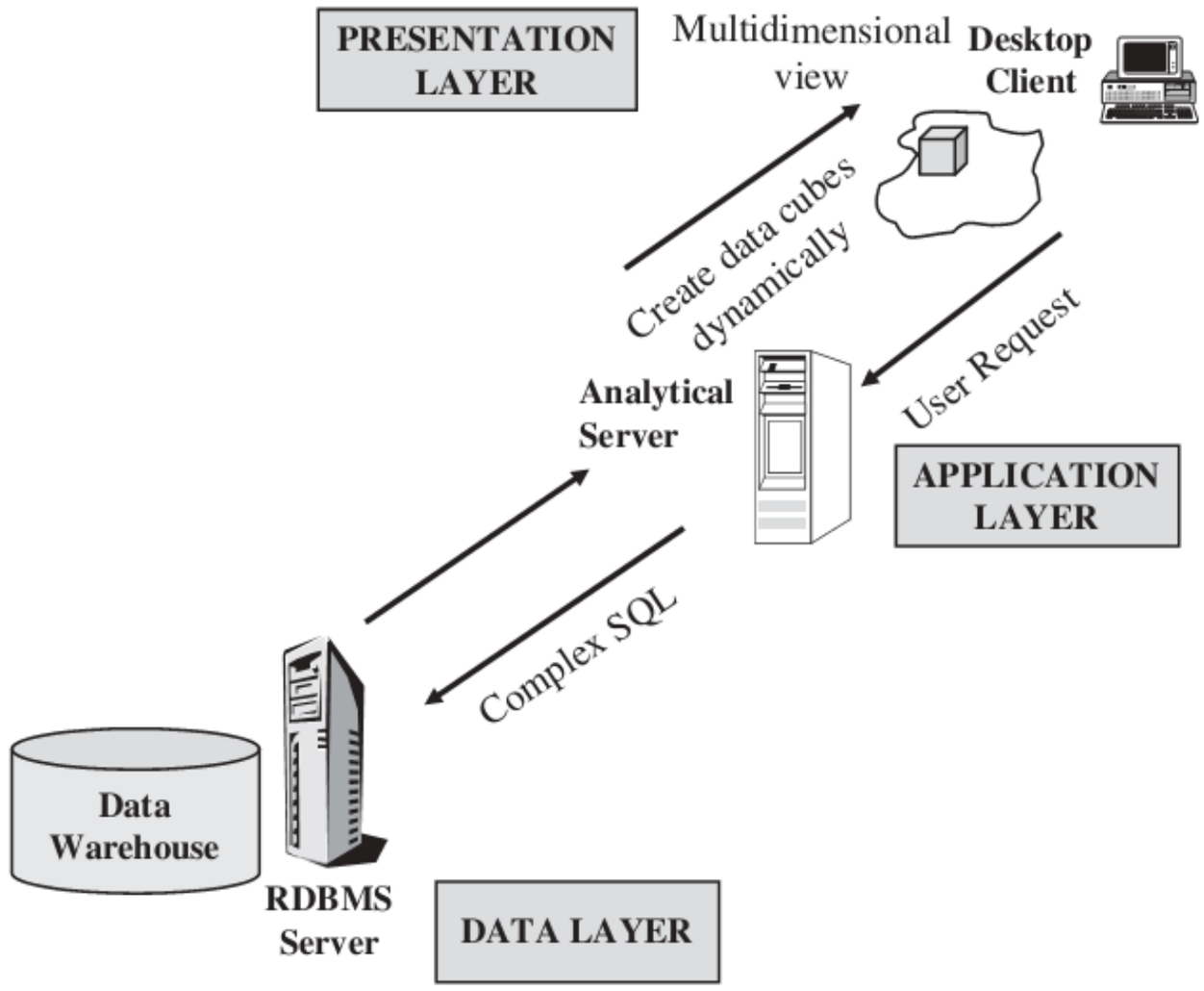
ลองพิจารณารูปที่ 11-16 เพื่อศึกษาถึงสถาปัตยกรรมของโมเดล ROLAP ที่ประกอบไปด้วย 3 layer คือ



—ฐานข้อมูลของคลังข้อมูล

—ลำดับตรงกลางที่ทำการเชื่อมต่อระหว่างฐานข้อมูลกับผู้ใช้งาน และจะเป็นลำดับชั้นที่ทำการแปลงข้อมูลจากแถวและคอลัมน์ให้เป็นข้อมูลในมิติต่าง ๆ แบบทันท่วงที

—ที่เป็นส่วนที่แสดงผลลัพธ์ให้กับผู้ใช้ โดยจะทำการแสดงผลในรูปแบบของมิติต่างๆทางธุรกิจ ซึ่งเมื่อไรก็ตามที่ผู้ใช้ต้องการวิเคราะห์ข้อมูลที่มีความซับซ้อนที่สอดคล้องกับมิติเชิงธุรกิจ คิวรีเหล่านั้นจะถูกเปลี่ยนไปเป็น SQL คิวรีที่มีความซับซ้อน เพื่อใช้ในการค้นหาข้อมูลในฐานข้อมูลโดยตรง



รูปที่ 11-16 สถาปัตยกรรมของโมเดล ROLAP

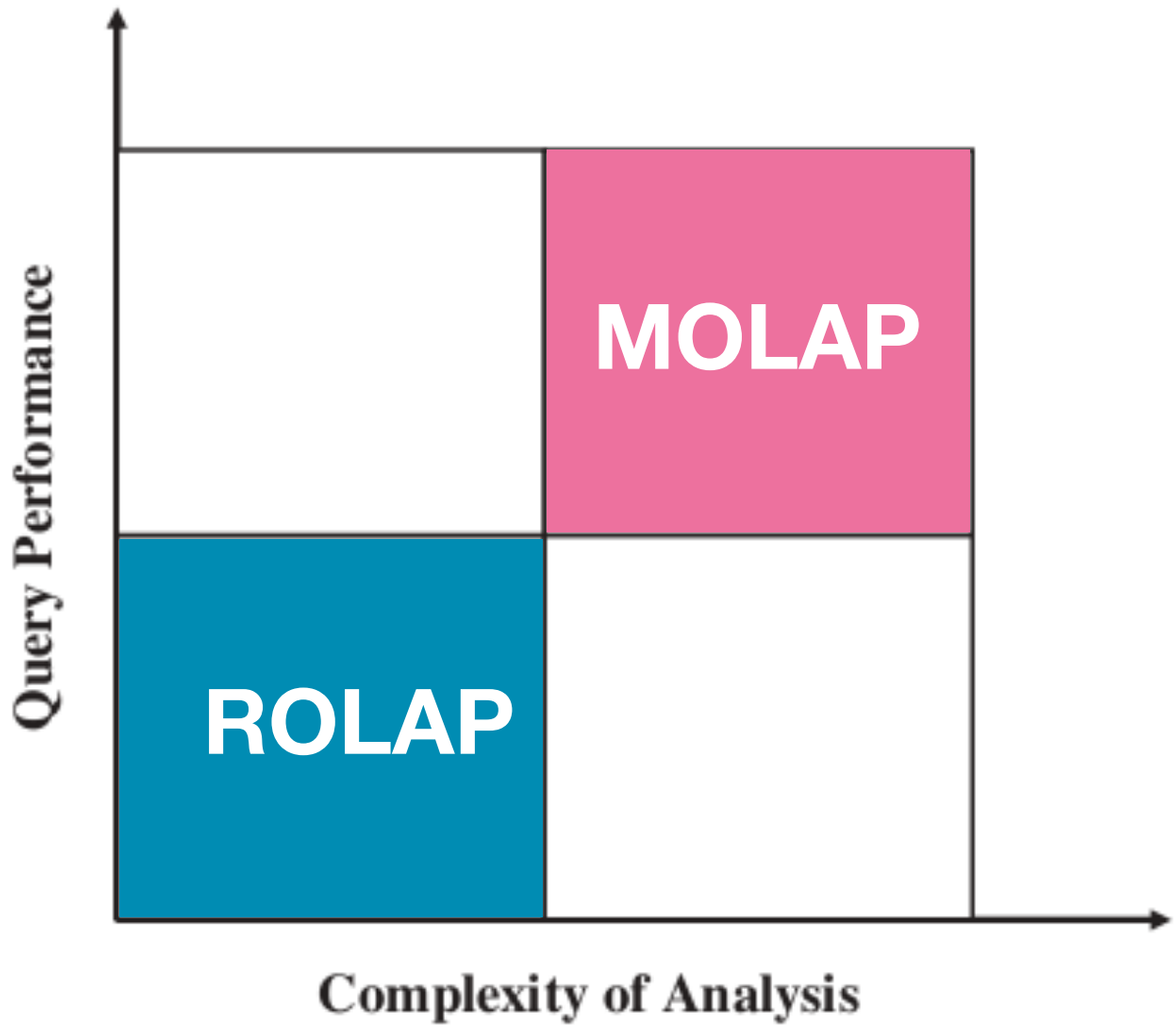
จากข้างต้นเราจะทราบเกี่ยวกับความแตกต่างระหว่างขั้นตอนการทำงานและสถาปัตยกรรมระหว่าง **MOLAP** และ **ROLAP** ซึ่งจากทั้งสองโมเดลข้างต้น**เราควรที่จะเลือกโมเดลใด?**

เพื่อช่วยในการประมวลผลการวิเคราะห์ข้อมูลต่าง ๆ แบบออนไลน์ ในการเลือกโมเดลสำหรับการประมวลผลนั้นจะขึ้นอยู่กับ 2 ปัจจัยหลักด้วยกันคือ

1. ความซับซ้อนของคิวรีที่ผู้ใช้เป็นผู้กำหนด และ
2. ความสำคัญของประสิทธิภาพของการประมวลผลคิวรีว่าสำคัญต่อผู้ใช้มากน้อยเพียงใด

โดยประสิทธิภาพของการประมวลคิวรีของทั้งสองโมเดลจะแตกต่างกันค่อนข้างมากดังแสดงในรูปที่ 11-17 ซึ่งจากรูปเราจะเห็นว่า MOLAP สามารถประมวลผลคิวรีได้ที่มีความซับซ้อนมากในเวลาที่น้อยกว่า ROLAP ค่อนข้างมาก ดังนั้น เพื่อให้เราสามารถเลือกโมเดลสำหรับการประมวลผลได้อย่างถูกต้อง ลองพิจารณาความแตกต่างระหว่างโมเดล MOLAP และ ROLAP ดังแสดงในรูปที่ 11-18 ที่จะแสดงถึงความแตกต่าง 3 แง่มุมด้วยกันคือ

1. การจัดเก็บข้อมูล
2. เทคนิคหรือเทคโนโลยีที่ซ่อนอยู่ในแต่ละโมเดล
3. ฟังก์ชันและคุณลักษณะต่างๆ



รูปที่ 11-17 การเปรียบเทียบประสิทธิภาพการประมวลผลคิวรี และหว่างโมเดล MOLAP และ ROLAP

	Data Storage	Underlying Technologies	Functions and Features
ROLAP	Data stored as relational tables in the warehouse. Detailed and light summary data available. Very large data volumes. All data access from the warehouse storage.	Use of complex SQL to fetch data from warehouse. ROLAP engine in analytical server creates data cubes on the fly. Multidimensional views by presentation layer.	Known environment and availability of many tools. Limitations on complex analysis functions. Drill-through to lowest level easier. Drill-across not always easy.
MOLAP	Data stored as relational tables in the warehouse. Various summary data kept in proprietary databases (MDDBs) Moderate data volumes. Summary data access from MDDB, detailed data access from warehouse.	Creation of pre-fabricated data cubes by MOLAP engine. Propriety technology to store multidimensional views in arrays, not tables. High speed matrix data retrieval. Sparse matrix technology to manage data sparsity in summaries.	Faster access. Large library of functions for complex calculations. Easy analysis irrespective of the number of dimensions. Extensive drill-down and slice-and-dice capabilities.

รูปที่ 11-18 การเปรียบเทียบความแตกต่างระหว่างโมเดล MOLAP และ ROLAP ในแง่มุมต่างๆ

SECTION 7

ปัจจัยที่ต้องพิจารณาในการสร้างระบบ OLAP

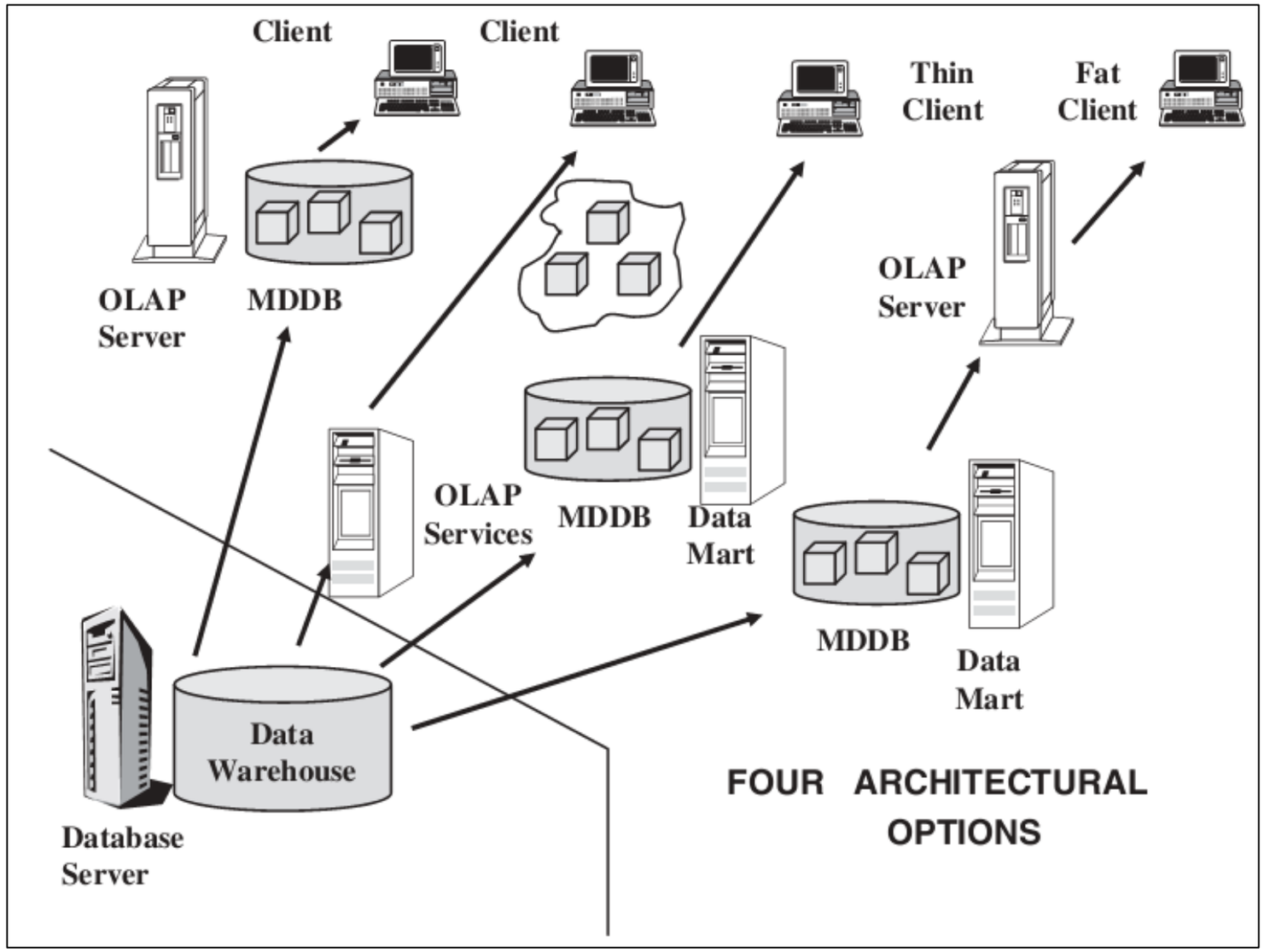
ก่อนที่เราจะทำการสร้างระบบ OLAP เราจะต้องพิจารณาถึงปัจจัยต่างๆของระบบที่เราจะทำการสร้างขึ้น ถ้าเราทำการสร้างระบบ OLAP โดยใช้โมเดล MOLAP ที่ใช้ MDDBMS ในการจัดเก็บข้อมูล เราจะต้องพิจารณา 2 ปัจจัยหลัก ๆ ด้วยกัน คือ

(1)
โมเดล MOLAP นั้นยังไม่มีมาตราฐานที่แน่นอน ซึ่งแต่ละผู้ขายจะทำการสร้างเครื่องมือสำหรับ MOLAP โดยทำการออกแบบอินเทอร์เฟซตามแต่ที่พวกเข้าต้องการและอยากให้เป็น

(2)
ความสามารถในการย่อหรือขยายได้ (Scalability) ซึ่งโดยส่วนใหญ่แล้วระบบ OLAP มักจะทำงานได้ดีกับการเรียกดูหรือวิเคราะห์ข้อมูลที่เป็นผลสรุป แต่เมื่อไรก็ตามที่ต้องทำการวิเคราะห์ที่มีรายละเอียดสูง ซึ่งมีปริมาณข้อมูลค่อนข้างมาก จะทำให้ประสิทธิภาพของระบบ OLAP ลดลง

ในการประมวลผลข้อมูลที่ถูกนอร์มอลไลซ์จะทำให้มีการประมวลผลเพิ่มขึ้น เมื่อเราทำการประมวลผลคิวรีที่มีความซับซ้อนค่อนข้างมาก ดังนั้น เพื่อที่จะหลีกเลี่ยงหรือลดการประมวลผลเราควรจะใช้ STAR schema ในการจัดเก็บข้อมูลมิติต่าง ๆ ซึ่งในบางเครื่องมือสำหรับ ROLAP จะเก็บข้อมูลมิติต่าง ๆ ไว้ใน STAR schema ที่ถูกประมวลไว้ก่อนหน้าแล้ว

จากปัจจัยข้างต้น เราสามารถดัดแปลงสถาปัตยกรรมพื้นฐานของโมเดล MOLAP และ ROLAP เพื่อเพิ่มประสิทธิภาพในการประมวลผล โดยเราจะมีทางเลือกของสถาปัตยกรรม 4 ทางเลือกด้วยกัน ดังแสดงในรูปที่ 11-19 ซึ่งจากรูปทางฝั่งซ้ายสุด 2 สถาปัตยกรรมแรกจะเป็นสถาปัตยกรรมพื้นฐานของโมเดล MOLAP และ ROLAP แต่ในส่วนสถาปัตยกรรมที่ 3 จะมีการนำดาต้ามาร์ทเข้ามาช่วยในการวิเคราะห์ข้อมูลที่มีรายละเอียดสูง และในสถาปัตยกรรมที่ 4 จะค่อนข้างเหมือนกับสถาปัตยกรรมที่ 3 แต่จะมีการเพิ่ม OLAP server เพื่อช่วยในการประมวลผลหรือดำเนินการในการวิเคราะห์ข้อมูลที่มีความแตกต่างกัน



เมื่อเราทราบถึงทางเลือกของสถาปัตยกรรมที่สนับสนุนการทำงานที่มีประสิทธิภาพ ขั้นตอนต่อไปเราควรที่จะต้องศึกษาเกี่ยวกับทางเลือกในการสร้างฟังก์ชันต่าง ๆ ของ OLAP ซึ่งเราจะต้องทำการศึกษาเกี่ยวกับองค์ประกอบต่าง ๆ ดังนี้

รูปที่ 11-19 ทางเลือกของสถาปัตยกรรม OLAP

การออกแบบและเตรียมข้อมูล

ข้อมูลในระบบ OLAP จะมาจากข้อมูลในคลังข้อมูลที่เราสร้างขึ้น ถ้าเราต้องการที่จะสร้างระบบ OLAP โดยใช้โมเดล MOLAP เราจะต้องทำการสร้าง multidimensional database แยกจากฐานข้อมูลของคลังข้อมูลเพื่อใช้ในการจัดเก็บข้อมูลในรูปแบบของลูกบาศก์มิติต่างๆ แต่ถ้าเราเลือกที่จะทำการสร้างระบบ OLAP โดยใช้โมเดล ROLAP เราจะไม่ต้องทำการสร้างฐานข้อมูลใหม่แต่อย่างใด แต่ข้อมูลจากฐานข้อมูลของคลังข้อมูลจะถูกส่งมายังระบบ OLAP เพื่อทำการสร้างลูกบาศก์เพื่อการวิเคราะห์ต่าง ๆ ทันที ซึ่งจากการทำงานทั้งสองระบบเราจะเห็นว่าระบบ OLAP จะมีการประมวลผลเพื่อสร้างเป็นลูกบาศก์มิติต่างที่ใช้สำหรับการวิเคราะห์ในแง่มุมต่าง ๆ

ดังนั้นก่อนที่เราจะทำการสร้างลูกบาศก์มิติต่าง ๆ เราควรที่จะต้องทำการศึกษาคุณลักษณะของข้อมูลในระบบ OLAP ซึ่งมีคุณลักษณะดังต่อไปนี้

- ระบบ OLAP จะทำการใช้และจัดเก็บข้อมูลน้อยกว่าระบบคลังข้อมูล
- ข้อมูลในระบบ OLAP มักจะเป็นข้อมูลที่เป็นผลสรุป จะมีข้อมูลที่มีรายละเอียดสูงอยู่ไม่มากนัก
- ข้อมูลในระบบ OLAP จะมีความยืดหยุ่นในการประมวลผลและวิเคราะห์ต่าง ๆ เนื่องจากข้อมูลเหล่านี้มีปริมาณน้อย
- ข้อมูลในระบบ OLAP มีแนวโน้มที่จะถูกแบ่งตามการดำเนินงานของแต่ละแผนก

ในการจัดเตรียมข้อมูลระบบ OLAP จะเป็นการจัดเตรียมข้อมูลของผู้ใช้เฉพาะกลุ่มหรือ เฉพาะแผนก เช่น แผนกการตลาด เป็นต้น โดยที่เมื่อเราได้กลุ่มของผู้ใช้แล้วเราจะทำการออกแบบข้อมูลให้ตรงกับการดำเนินงานของแต่ละกลุ่มดังนี้

Define subset	Summarize	Denormalize	Calculate and derive	Index
จะเป็นการเลือกข้อมูลเพียงส่วนที่แผนกการตลาดสนใจ	จะเป็นการสรุปและรวมข้อมูลที่ตรงกับความต้องการของแผนกการตลาด ตัวอย่างเช่น ทำการสรุปรายการสินค้าพร้อมกับหมวดหมู่ของสินค้าตามที่แผนกการตลาดได้กำหนดไว้ ซึ่งในบางครั้งแผนกการตลาดและแผนกบัญชีจะมีการแบ่งหมวดหมู่สินค้าโดยวิธีที่แตกต่างกัน	จะเป็นการดีนอร์มอลไลซ์ข้อมูลที่ผู้ใช้ต้องการ โดยทำการรวมข้อมูลจาก relational tables ต่าง ๆ เข้าด้วยกัน เช่น ถ้าแผนกการตลาดต้องการรวมตาราง A และ B เข้าด้วยกัน แต่แผนกการเงินต้องการรวมตาราง B และ C เข้าด้วยกัน เราจะทำการรวมตาราง A และ B เข้าด้วยกันสำหรับ OLAP สำหรับแผนกการตลาด	จะเป็นการคำนวณค่าต่าง ๆ ของตัวชี้วัดให้ตรงกับความต้องการของกลุ่มผู้ใช้เป้าหมายให้มากที่สุด	จะเป็นการเลือกแอทริบิวที่เหมาะสมสำหรับการสร้างดัชนีสำหรับแผนกการตลาด

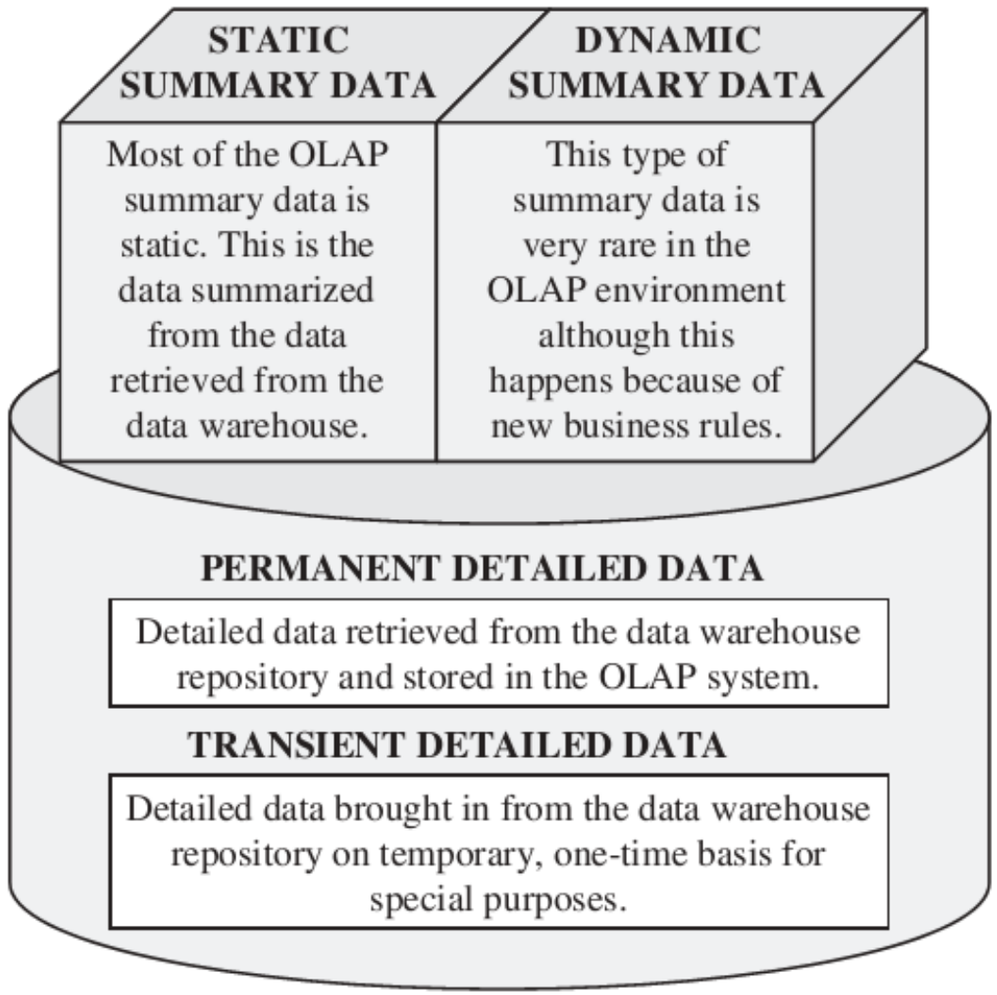
จากขั้นตอนการเตรียมข้อมูลข้างต้นจะทำให้ระบบ OLAP นั้นมีข้อมูลที่เป็นผลสรุปหลายระดับ และมีข้อมูลที่เป็นรายละเอียดเล็กน้อย โดยข้อมูลที่ถูกจัดเก็บอยู่ใน MOLAP หรือข้อมูลที่เป็นผลลัพธ์จากการประมวลผลของระบบ ROLAP จะมีลักษณะต่าง ๆ ดังรูปที่ 11-20 ที่จะประกอบไปด้วยข้อมูล 4 ลักษณะด้วยกันคือ

(1) Static summary data

(2) Dynamic summary data

(3) Permanent detailed data

(4) Transient detailed data



รูปที่ 11-20 ลักษณะของข้อมูลใน OLAP

การดูแลและจัดการสิ่งต่าง ๆ ใน OLAP

ด้วยเหตุที่ระบบ OLAP นั้นเป็นส่วนประกอบหนึ่งของระบบคลังข้อมูลที่ใช้สำหรับการส่งผ่านข้อมูลไปยังผู้ใช้ ดังนั้น การจัดการต่าง ๆ ของระบบ OLAP จะเป็นส่วนหนึ่งของการจัดการของระบบคลังข้อมูลด้วยเช่นกัน โดยในการจัดการสิ่งต่าง ๆ ของระบบ OLAP เราจะต้องพิจารณาถึงปัจจัยต่าง ๆ ดังต่อไปนี้

- ความคาดหวังกับข้อมูลที่สามารถเรียกใช้งานได้และวิธีการในการเรียกใช้งานข้อมูลเหล่านั้น
- การเลือกมิติทางธุรกิจที่ถูกต้องและเหมาะสม
- วิธีการสำหรับเคลื่อนย้าย/ถ่ายโอนข้อมูลจากคลังข้อมูลเข้าสู่ระบบ OLAP (ในกรณีของ MOLAP)
- การเลือกวิธีการในการรวบรวมข้อมูล การสร้างผลสรุปของข้อมูล และการคำนวณข้อมูลต่าง ๆ ไว้ล่วงหน้า
- การพัฒนาระบบ OLAP โดยใช้ซอร์ฟแวร์ต่าง ๆ ที่วางอยู่ตามท้องตลาด
- ขนาดของ multidimensional database
- การจัดการกับความเบาบางของข้อมูลที่อาจเกิดขึ้นกับโครงสร้างที่เป็นแบบ multidimension
- การเรียกดูข้อมูลแบบเจาะลึกไปยังข้อมูลที่มีรายละเอียดสูงสุด
- การเรียกดูข้อมูลแบบเจาะลึกที่เป็นข้อมูลที่ต้องเป็นการติดต่อข้ามแผนก
- มาตราการในการเข้าถึงข้อมูลและรักษาความปลอดภัย
- การสำรองและกู้คืนข้อมูล

ประสิทธิภาพการทำงานของระบบ OLAP

ก่อนที่จะพิจารณาถึงประสิทธิภาพของการประมวลผลในระบบ OLAP เราจะต้องเข้าใจก่อนว่า ระบบคลังข้อมูลที่มีระบบ OLAP เป็นส่วนประกอบจะมีการเคลื่อนย้ายการประมวลผลคิวรีมายังระบบ OLAP ซึ่งจะทำให้ระบบคลังข้อมูลหลักนั้นมีการทำงานที่ลดลง โดยคิวรีที่มักจะถูกประมวลผลที่ระบบ OLAP มักจะเป็นคิวรีที่มีความซับซ้อนและเต็มไปด้วยการคำนวณต่าง ๆ ซึ่งจากการประมวลผลคิวรีที่มีความซับซ้อนจะทำให้เวลาที่ผู้ใช้เข้าใช้งานแต่ละครั้งจะค่อนข้างยาวนานด้วยเช่นกัน

ระบบ OLAP มักถูกสร้างขึ้นเพื่อทำการประมวลผลคิวรีที่ความซับซ้อน ซึ่งในการคำนวณและประมวลผลคิวรีต่าง ๆ ในระบบ OLAP จะสามารถทำงานได้เร็วกว่าเนื่องจากระบบ OLAP มีการรวบยอดข้อมูล และทำการคำนวณข้อมูลในแง่มุมต่าง ๆ ไว้ล่วงหน้าแล้ว ทำการเก็บไว้ใน multidimensional database ในรูปแบบของลูกบาศก์หลายมิติ (Hypercubes) โดยการเก็บข้อมูลที่เป็นผลสรุปดังกล่าวจะช่วยลดการใช้พื้นที่ในการจัดเก็บข้อมูลใน multidimensional database ได้ ในการประมวลผลคิวรีจะทำการประมวลผลคิวรีที่เหมาะสมหรือเกี่ยวข้องกับลูกบาศก์หลายมิติที่เราทำการสร้างไว้

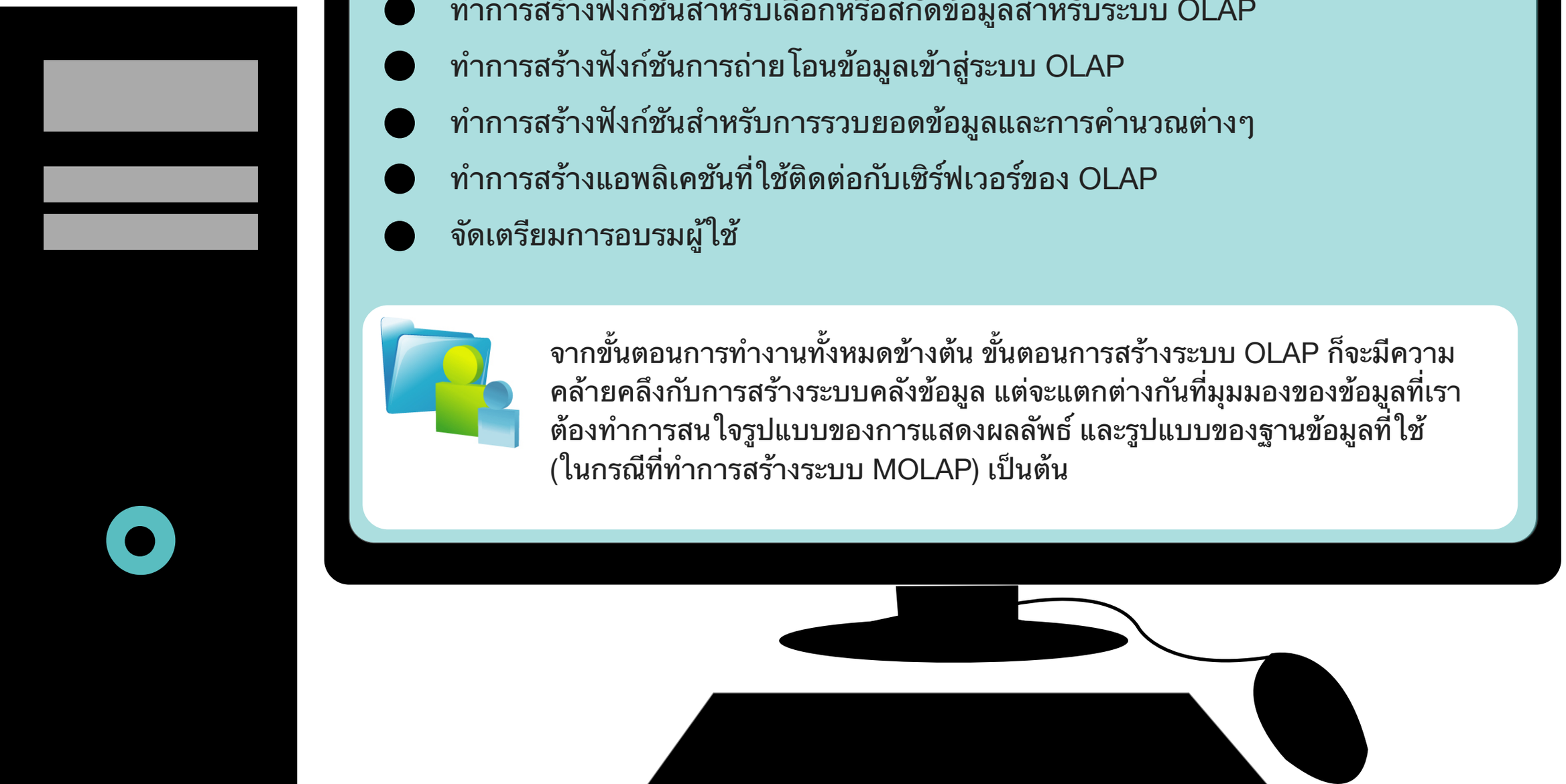
เพื่อให้เข้าใจถึงประสิทธิภาพของการทำงานของระบบ OLAP ลองพิจารณาตัวอย่างดังต่อไปนี้ที่ระบบคลังข้อมูลจะประกอบไปด้วยมิติทางธุรกิจ 3 มิติด้วยกัน ซึ่งจากมิติต่างๆ ระบบ OLAP จะสามารถทำการคำนวณและรวบยอดข้อมูลได้ดังนี้

- ข้อมูลที่มีความละเอียดสูงทั้ง 3 มิติจะถูกเก็บไว้ในอะเรย์ 3 มิติ
- ข้อมูลที่เกี่ยวข้องกับมิติที่ 1 และมิติที่ 2 จะถูกเก็บไว้ในอะเรย์ 2 มิติ
- ข้อมูลที่เกี่ยวข้องกับมิติที่ 2 และมิติที่ 3 จะถูกเก็บไว้ในอะเรย์ 2 มิติ
- ข้อมูลที่เป็นผลสรุปของมิติที่ 1 จะถูกเก็บไว้ในอะเรย์ 1 มิติ
- ข้อมูลที่เป็นผลสรุปของมิติที่ 2 จะถูกเก็บไว้ในอะเรย์ 1 มิติ
- ข้อมูลที่เป็นผลสรุปของมิติที่ 3 จะถูกเก็บไว้ในอะเรย์ 1 มิติ

จากการคำนวณและรวบยอดข้อมูลไว้ก่อนหน้าจะช่วยให้ระบบ OLAP สามารถประมวลคิวรีที่เป็นการวิเคราะห์ผลสรุปของข้อมูลได้อย่างรวดเร็ว แต่อย่างไรก็ดี การใช้ระบบ OLAP ก็นำมาซึ่งค่าใช้จ่ายและยังมีข้อเสียตรงที่การอัพเดทข้อมูลในระบบ OLAP จะไม่สามารถทำได้ทันทีหรือทำได้ในแต่ละวันเนื่องจากการถ่ายโอนข้อมูลสำหรับทำการคำนวณล่วงหน้าและการถ่ายโอนข้อมูลที่เป็นลูกบาศก์หลายมิติใช้เวลาค่อนข้างมาก ซึ่งระบบ OLAP ส่วนใหญ่จะทำการอัพเดทข้อมูลเพียงเดือนละครั้งเท่านั้น

ขั้นตอนการสร้าง OLAP

ในการสร้างระบบ OLAP เราจะต้องทราบถึงคุณลักษณะและฟังก์ชันการทำงานของ OLAP เราจะต้องทราบถึงความสำคัญ และปัจจัยต่าง ๆ ที่เกี่ยวข้องกับ OLAP เมื่อเราทราบเกี่ยวกับสิ่งเหล่านี้ทั้งหมดแล้ว เราจึงทำการลงมือสร้างระบบ OLAP ซึ่งจะประกอบไปด้วยขั้นตอนคร่าว ๆ ดังต่อไปนี้ โดยที่แต่ละขั้นตอนจะมีขั้นตอนย่อยๆอีกเป็นจำนวนมาก แต่เราจะทำการพิจารณาถึงขั้นตอนการทำงานหลักๆ ดังนี้



- ทำการสร้างแบบจำลองมิติต่างๆ (Dimensional modeling)
- ทำการออกแบบและสร้าง MDDB
- ทำการเลือกข้อมูลที่จะทำการเคลื่อนย้ายเข้าไปยังระบบ OLAP
- ทำการสร้างฟังก์ชันสำหรับเลือกหรือสกัดข้อมูลสำหรับระบบ OLAP
- ทำการสร้างฟังก์ชันการถ่ายโอนข้อมูลเข้าสู่ระบบ OLAP
- ทำการสร้างฟังก์ชันสำหรับการรวบยอดข้อมูลและการคำนวณต่างๆ
- ทำการสร้างแอพลิเคชันที่ใช้ติดต่อกับเซิร์ฟเวอร์ของ OLAP
- จัดเตรียมการอบรมผู้ใช้

จากขั้นตอนการทำงานทั้งหมดข้างต้น ขั้นตอนการสร้างระบบ OLAP ก็จะมีความคล้ายคลึงกับการสร้างระบบคลังข้อมูล แต่จะแตกต่างกันที่มุมมองของข้อมูลที่เราต้องทำการสนใจรูปแบบของการแสดงผลลัพธ์ และรูปแบบของฐานข้อมูลที่ใช้ (ในกรณีที่ทำการสร้างระบบ MOLAP) เป็นต้น

SECTION 8

คำถามท้ายบท

1. จงอธิบายเกี่ยวกับการวิเคราะห์ข้อมูลหลายมิติ
2. จงอธิบายเกี่ยวกับนิยามของระบบ OLAP ที่ถูกนิยามโดย Dr. Codd
3. Hypercubes และ MDS คืออะไร ใช้ทำอะไร มีประโยชน์อย่างไร
4. การดำเนินการ (operation) ของระบบ OLAP มีอะไรบ้าง แล้วแต่ละการดำเนินการมีการทำงานอย่างไร
5. จงอธิบายเกี่ยวกับการออกแบบสถาปัตยกรรมของระบบ OLAP ที่มี 4 ทางเลือก
6. การวิเคราะห์แบบเจาะลึก แบบผลสรุป แบบบางส่วน และการปรับเปลี่ยนมุมมองของข้อมูลเป็นอย่างไร จงอธิบายและยกตัวอย่างประกอบ
7. ข้อแตกต่างระหว่าง MOLAP และ ROLAP มีอะไรบ้าง จงอธิบาย
8. ปัจจัยที่ต้องพิจารณาในการสร้างระบบ OLAP มีอะไรบ้าง
9. การสร้างระบบ OLAP ประกอบไปด้วยขั้นตอนอะไรบ้าง

CHAPTER 12

การวางแผนและการจัดการสร้างคลังข้อมูล

SECTION 1

แผนการสอนประจำบท

วัตถุประสงค์ของบทเรียน

- ศึกษาเกี่ยวกับการออกแบบแบบจำลองทางกายภาพ
- ทำความเข้าใจเกี่ยวกับปัจจัยในการสร้างแบบจำลองทางกายภาพ
- ทำความเข้าใจบทบาทของการจัดเก็บข้อมูลที่มีผลต่อการทำงานของคลังข้อมูล
- พิจารณาเทคนิคการสร้างดัชนีสำหรับคลังข้อมูล
- ทบทวนและสรุปเกี่ยวกับทางเลือกในการเพิ่มประสิทธิภาพให้กับการทำงานของคลังข้อมูล

เนื้อหาของบทเรียน

เนื้อหาในบทนี้จะประกอบด้วย ขั้นตอนการออกแบบการสร้างแบบจำลองทางกายภาพ ปัจจัยที่ต้องพิจารณาในการออกแบบโมเดลทางกายภาพ การจัดเก็บข้อมูลทางกายภาพ การสร้างดัชนีในคลังข้อมูล เทคนิคการเพิ่มประสิทธิภาพในการทำงานอื่น ๆ

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารประกอบการสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

กิจกรรมการเรียน-การสอน

- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารประกอบการสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

การวัดและประเมินผล

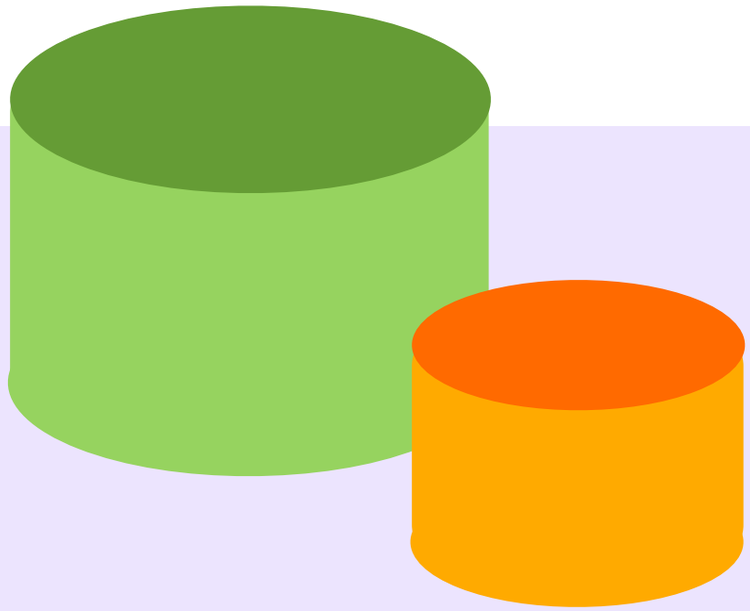
- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

SECTION 2

บทนำ

ในการสร้างระบบการดำเนินงานทั่วไป เราจะคุ้นเคยกับแบบจำลองเชิงตรรกะ (Logical model) และแบบจำลองเชิงกายภาพ (Physical model) เป็นอย่างดี ถ้าการสร้างระบบเริ่มต้นด้วยการออกแบบแบบจำลองเชิงตรรกะ (ซึ่งเกี่ยวข้องกับการเชื่อมโยงความสัมพันธ์ของ *object* ต่าง ๆ เช่น ความสัมพันธ์ของตาราง คอลัมน์ แถวของข้อมูล และอื่น ๆ) เราอาจต้องทำการเปลี่ยนจากแบบจำลองเชิงตรรกะให้เป็นแบบจำลองเชิงกายภาพ (ซึ่งเกี่ยวข้องกับวิธีการจัดเก็บและค้นคืนข้อมูลที่มีประสิทธิภาพ รวมถึงการถ่ายโอนข้อมูล การสำรองข้อมูล และการกู้คืนข้อมูล) โดยในการออกแบบแบบจำลองเชิงกายภาพเราจะต้องพิจารณาเกี่ยวกับแพลตฟอร์มของระบบ ซอร์ฟแวร์ที่ใช้สำหรับฐานข้อมูล อุปกรณ์ฮาร์ดแวร์ และเครื่องมือต่าง ๆ ที่ใช้ในการสร้างแบบจำลองเชิงกายภาพ

นการสร้างแบบจำลองเชิงกายภาพของระบบการดำเนินงานเราจะต้องยุ่งเกี่ยวกับงานในหลาย ๆ งาน ด้วยกัน เช่น การกำหนดที่ตั้งของฐานข้อมูล การพิจารณาการเลือกเก็บคุณลักษณะ (Feature) ต่าง ๆ ไว้ในฐานข้อมูล และอื่น ๆ ซึ่งสิ่งเหล่านี้จะเป็นตัวกำหนดพารามิเตอร์ต่าง ๆ สำหรับการจัดเก็บข้อมูล นอกจากนี้เรายังต้องคิดออกแบบและวางแผนเกี่ยวกับการเพิ่มประสิทธิภาพให้กับการงาน เช่น การสร้างดัชนี (index) ให้กับข้อมูล ซึ่งจะต้องศึกษารายละเอียดของข้อมูลว่าคอลัมน์ใดในแต่ละตารางที่ควรทำการสร้างดัชนีบ้าง หรือเราอาจต้องมองหาวิธีการอื่นเพื่อเพิ่มประสิทธิภาพในการทำงาน

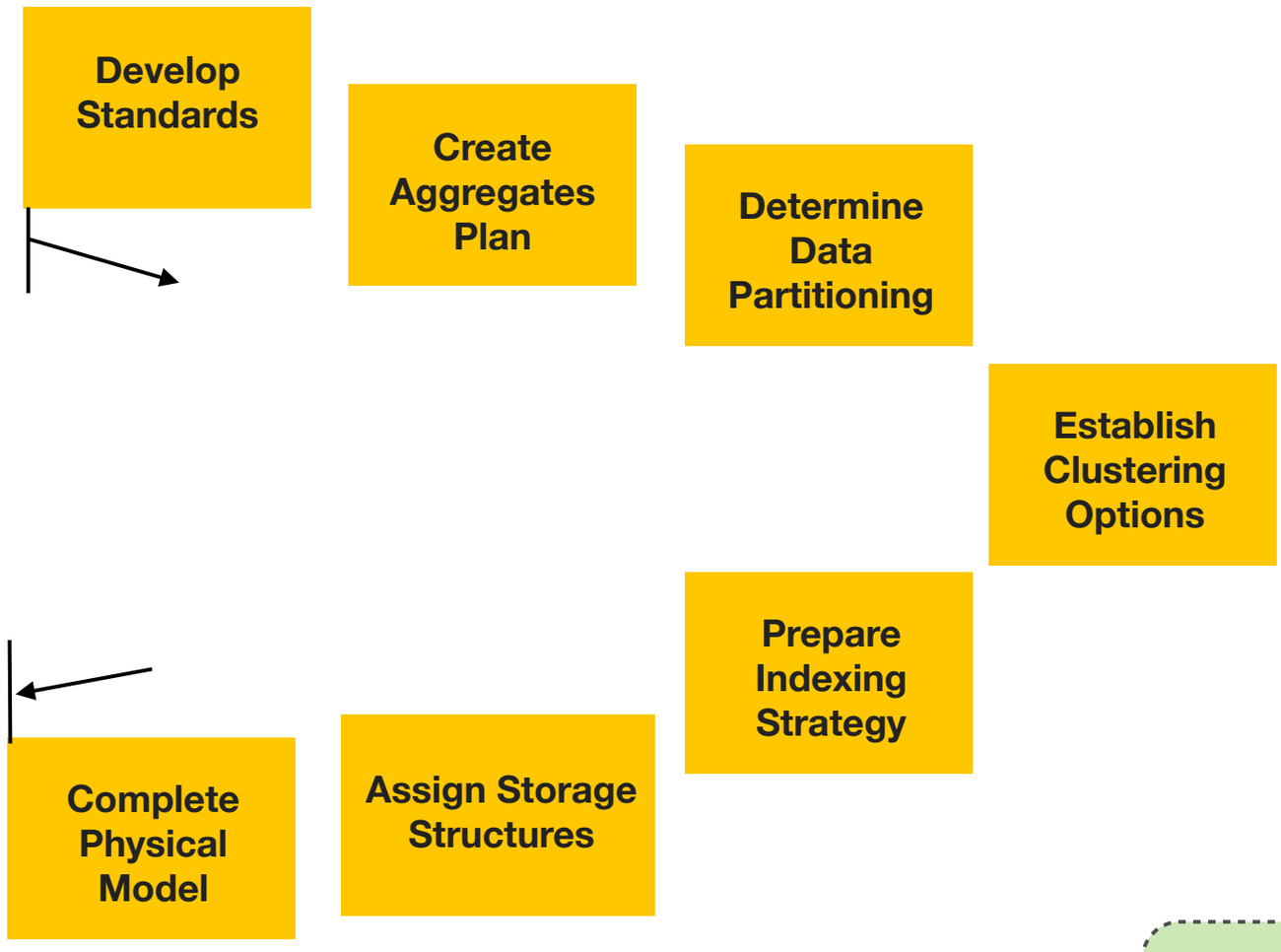


การสร้างแบบจำลองเชิงตรรกะของคลังข้อมูลจะทำการสร้าง dimensional model ที่จะแสดงความสัมพันธ์ของข้อมูลที่ควรจะเก็บอยู่ในฐานข้อมูลของคลังข้อมูล จากนั้นเราจะใช้แบบจำลองเชิงตรรกะที่สร้างขึ้นเพื่อสร้างแบบจำลองเชิงกายภาพของคลังข้อมูลต่อไป ซึ่งในบทนี้เราจะเน้นที่การออกแบบ การสร้าง และปัจจัยต่าง ๆ ที่เกี่ยวข้องกับการสร้างแบบจำลองเชิงกายภาพของคลังข้อมูล

SECTION 3

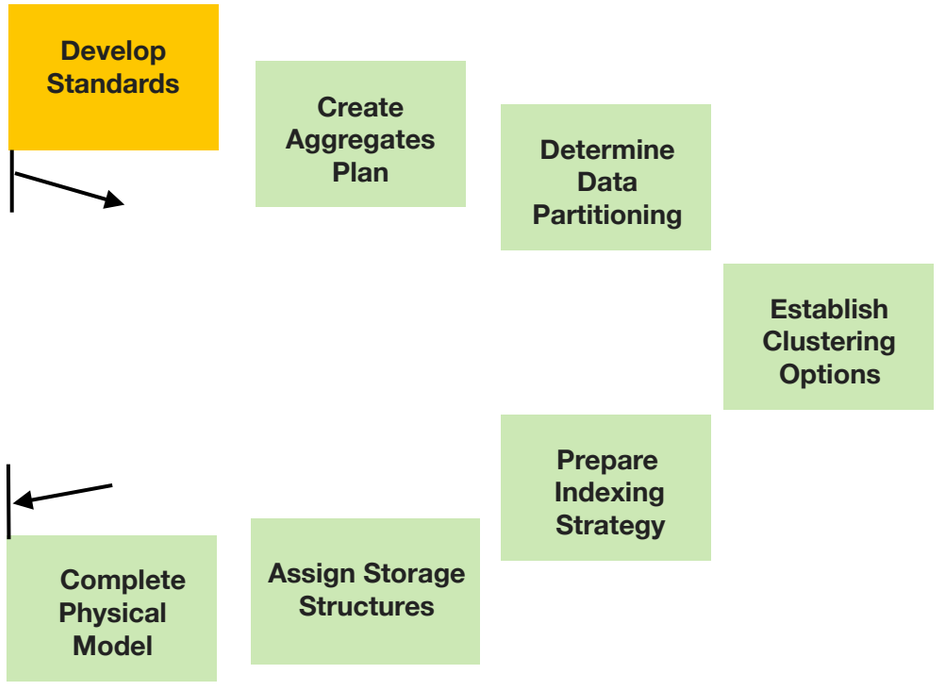
ขั้นตอนการออกสร้างแบบจำลองทางกายภาพ

ขั้นตอนการออกสร้างแบบจำลองทางกายภาพ



รูปที่ 12-1 ขั้นตอนการออกแบบแบบจำลองทางกายภาพ

ขั้นตอนการออกแบบแบบจำลองเชิงกายภาพจะประกอบไปด้วยขั้นตอนการทำงาน 7 ขั้นตอนด้วยกัน ดังแสดงในรูปที่ 12-1 ซึ่งจะแสดงถึงรายละเอียดของแต่ละขั้นตอนดังนี้

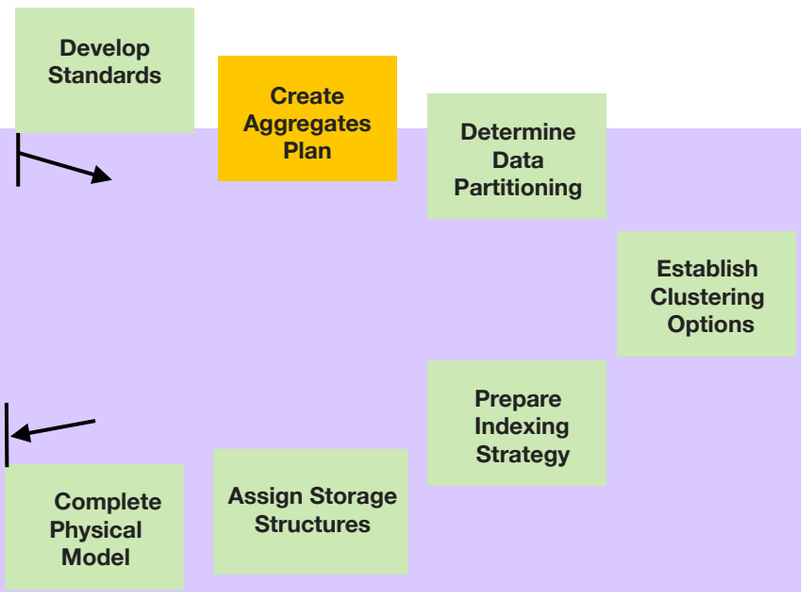


ขั้นตอนที่ 1 การสร้างมาตรฐาน (Develop Standards)

เป็นขั้นตอนที่จะทำให้สิ่งต่าง ๆ ในคลังข้อมูลเป็นมาตราฐาน เช่น การตั้งชื่อฐานข้อมูล ตาราง และ คอลัมน์ต่าง ๆให้เป็นมาตราฐานเดียวกัน ซึ่งจะสามารถช่วยลดความกำกวม เพิ่มความเข้าใจในเนื้อหาทางธุรกิจที่เกี่ยวข้องกับฐานข้อมูลนั้น ๆให้กับผู้ใช้ ผู้สร้างหรือผู้ดูแลคลังข้อมูล จะทำให้ผู้ใช้สามารถสร้างการสืบค้นข้อมูล (Query) ได้เอง และสามารถทำได้โดยง่าย การตั้งชื่อฐานข้อมูล ตาราง และคอลัมน์ให้เป็นมาตราฐานโดยส่วนใหญ่แล้วจะใช้คำหลาย ๆ คำประกอบกัน และจะเว้นวรรคระหว่างคำด้วยเครื่องหมาย “-” หรือ “_” นอกจากนี้ในการตั้งชื่อโดยใช้คำมาประสมกันจะเน้นที่คำแรกของชื่อซึ่งจะบ่งบอกถึงหัวข้อทางธุรกิจที่เกี่ยวข้องกับคำนั้น ๆ

ขั้นตอนที่ 2 การวางแผนการรวบยอดข้อมูล (Create Aggregates Plan)

ในการสืบค้นข้อมูลจากคลังข้อมูลจากผู้ใช้ร้อยละ 80 มักจะถามถึงข้อมูลที่เป็นแบบสรุปผล ถ้าคลังข้อมูลที่เราสร้างขึ้นมีการจัดเก็บข้อมูล ในระดับที่ละเอียดที่สุดเท่านั้น เมื่อผู้ใช้ทำการสร้างคิวรีสำหรับสืบค้นข้อมูล คลังข้อมูลจะต้องทำการอ่านข้อมูลที่อยู่ในระดับที่ละเอียดที่สุดทั้งหมด แล้วทำการหาผลสรุปตามความต้องการของผู้ใช้



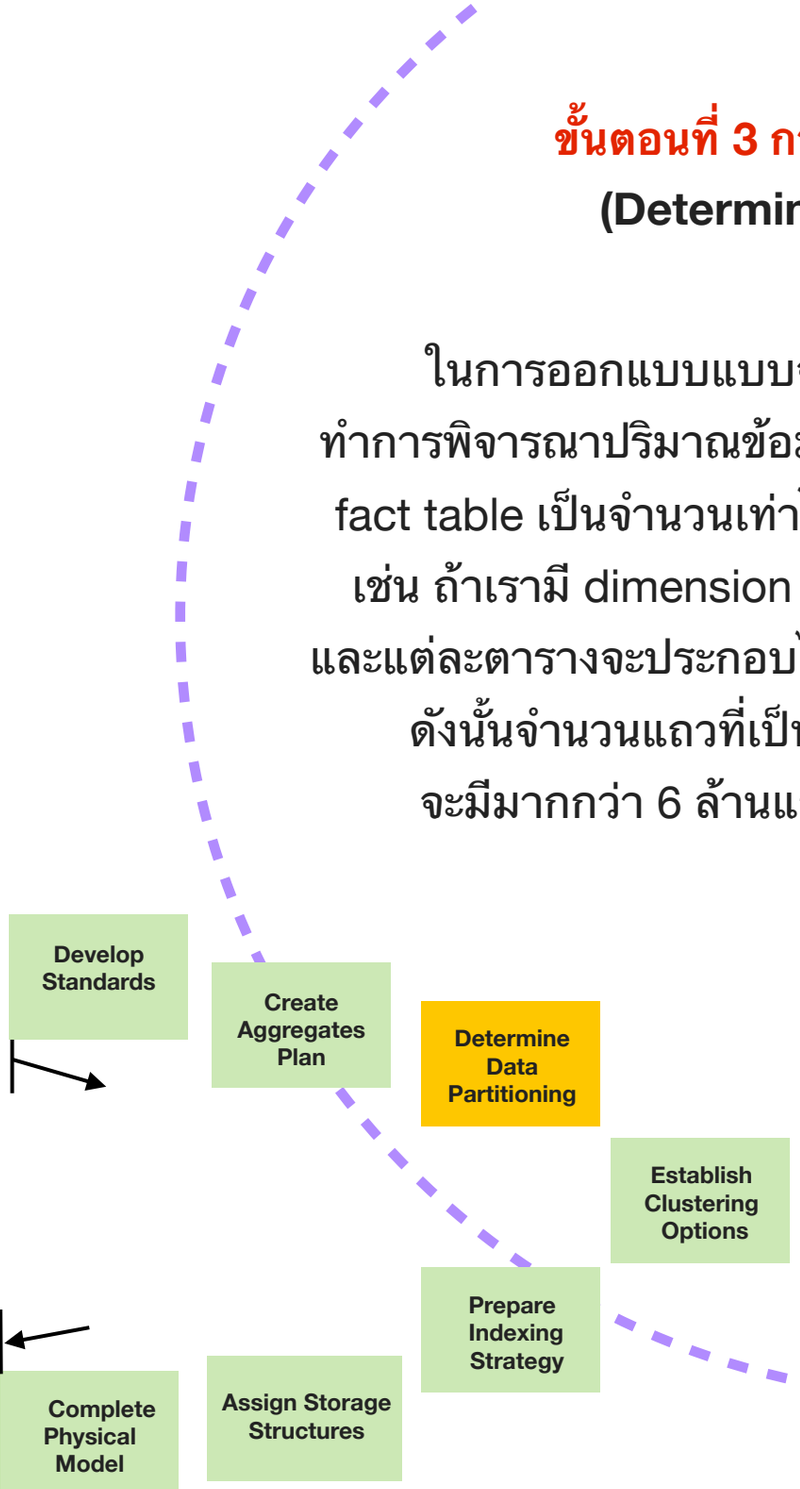
ลองพิจารณาการเรียกใช้ข้อมูลการขายของแต่ละรายการสินค้า ในแต่ละสาขา ในหนึ่งปี ถ้าเราเก็บข้อมูลในแต่ละครั้งของการซื้อสินค้าของลูกค้าไว้ในคลังข้อมูล เราจะต้องทำการอ่านข้อมูลเป็นจำนวนมากเป็นอันดับแรก จากนั้นค่อยทำการหาผลสรุปของข้อมูล ซึ่งกรณีดังกล่าว เราจะเห็นว่าประสิทธิภาพการค้นคืนข้อมูลไม่ค่อยจะสู้ดีนัก จึงมีคำถามที่ว่าเราจะสามารถเพิ่มประสิทธิภาพการค้นคืนข้อมูลได้อย่างไร?

สมมติว่าเรามีอีกหนึ่งตารางที่เก็บข้อมูลที่มีความละเอียดน้อยลง เช่น เก็บยอดขายของสินค้าในแต่ละสาขาจะช่วยให้การสืบค้นข้อมูลจากคลังข้อมูลเร็วขึ้นหรือไม่? ถ้าการสืบค้นเร็วขึ้น เราต้องทำการสร้างตารางของข้อมูลที่มีความละเอียดน้อยลงเป็นจำนวนเท่าใด? และข้อจำกัดมีอะไรบ้าง?

จากคำถามก่อนหน้า เราต้องวิเคราะห์ถึงความเป็นไปได้ในการสร้างตารางข้อมูลใหม่ที่มีความละเอียดลดลง (เรียกว่า aggregate table หรือ summary table) ซึ่งในการสร้างตารางใหม่ เราอาจเริ่มจากการวิเคราะห์นิยามความต้องการของผู้ใช้ที่เก็บไว้เป็นเมตาดาต้า จากนั้นเราจะมองที่แต่ละ dimension table ว่ามีลำดับชั้นความละเอียดของข้อมูลเป็นอย่างไรและระดับชั้นใดที่เป็นสิ่งสำคัญในการรวบรวมข้อมูลเข้าด้วยกันเพื่อแสดงผลลัพธ์การสืบค้นข้อมูล ซึ่งจากขั้นตอนการทำงานดังกล่าวเราต้องทำการวางแผนในการรวบยอดข้อมูล ซึ่งจะต้องคิดพิจารณาเกี่ยวกับชนิดของการรวบยอดที่ควรจะทำกับแต่ละระดับของข้อมูล

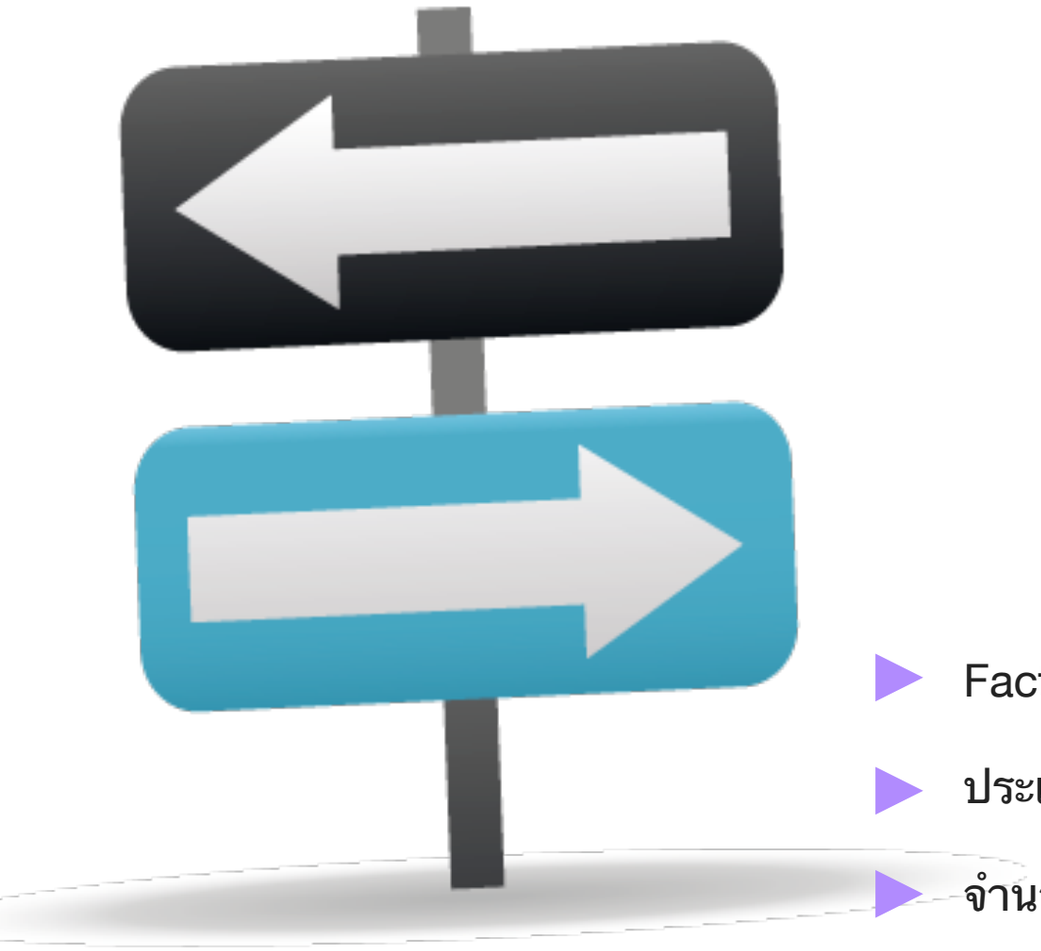
ขั้นตอนที่ 3 การแบ่งข้อมูลออกเป็นส่วนย่อยๆ
(Determine the Data Parititioning)

ในการออกแบบแบบจำลองเชิงกายภาพ เราจำเป็นต้องทำการพิจารณาปริมาณข้อมูลในคลังข้อมูลว่ามีจำนวนเรคคอร์ดใน fact table เป็นจำนวนเท่าไร ซึ่งเราสามารถคำนวณอย่างคร่าวได้ เช่น ถ้าเรามี dimension table 4 ตารางถูกเก็บไว้ในคลังข้อมูล และแต่ละตารางจะประกอบไปด้วยข้อมูลประมาณ 50 แถวโดยเฉลี่ย ดังนั้นจำนวนแถวที่เป็นไปได้ที่จะถูกเก็บอยู่ใน fact table จะมีมากกว่า 6 ล้านแถวด้วยกัน(≅ 50 × 50 × 50 × 50)

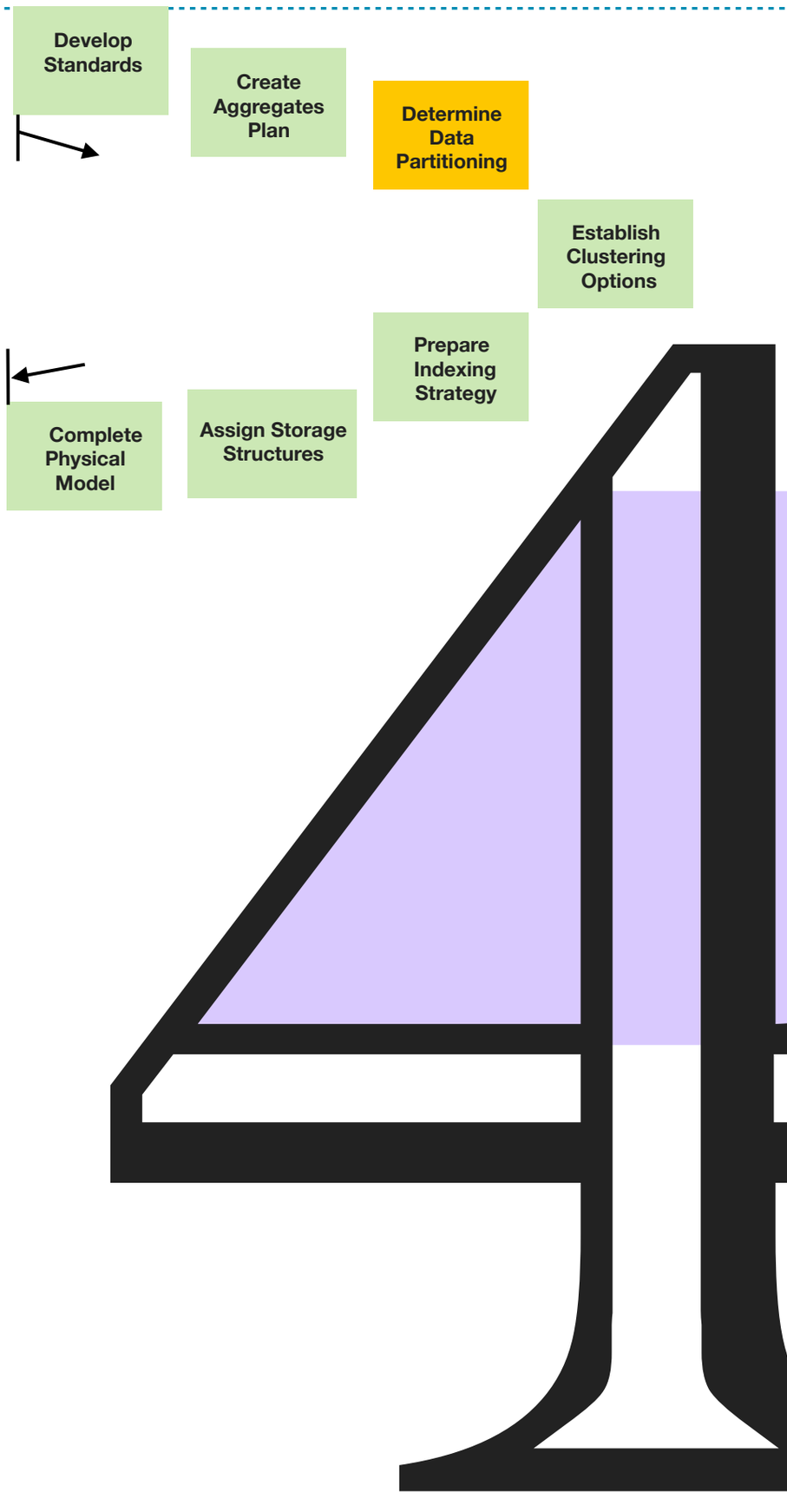


โดยปกติของคลังข้อมูลจะมีปริมาณข้อมูลใน fact table ค่อนข้างมาก ซึ่งจะทำให้ยากต่อการจัดการ ในแง่ของการสำรองข้อมูลและการกู้คืนข้อมูล ดังนั้นการแบ่งข้อมูลจากตารางที่มีข้อมูลมากออกเป็นตารางย่อย ๆ จะสามารถช่วยให้การบริหารจัดการได้ง่ายขึ้น

การพิจารณาถึงทางเลือกในการแบ่งข้อมูลออกเป็นส่วน ๆ เราจะต้องพิจารณาว่าเราจะทำการแบ่งตารางออกตามแนวนอนหรือตามแนวตั้ง ? นอกจากนั้นเราต้องทำการพิจารณาว่าตารางใดควรจะต้องทำการแบ่งข้อมูลออกเป็นส่วน ๆ ด้วย ซึ่งในการแบ่งข้อมูลเราจะต้องเขียน partitioning scheme ขึ้นมาก่อน ซึ่ง partitioning scheme จะประกอบไปด้วย



- Fact และ dimension tables ที่เลือกไว้สำหรับการแบ่งข้อมูลออกเป็นส่วน ๆ
- ประเภทของการแบ่งข้อมูลสำหรับแต่ละตาราง (แบ่งตามแนวตั้งหรือแบ่งตามแนวนอน)
- จำนวนของตารางที่ต้องการหลังจากการแบ่งข้อมูลตารางหนึ่งๆ
- เกณฑ์ในการแบ่งข้อมูลแต่ละตาราง (เช่น แบ่งข้อมูลตามกลุ่มของสินค้า เป็นต้น)
- คำอธิบายในการสร้างคิวรีสำหรับตารางข้อมูลย่อยที่ถูกแบ่งออกเป็นส่วน ๆ แล้ว



ขั้นตอนที่ 4 ทางเลือกในการสร้างกลุ่มของข้อมูล (Establish Clustering Options)

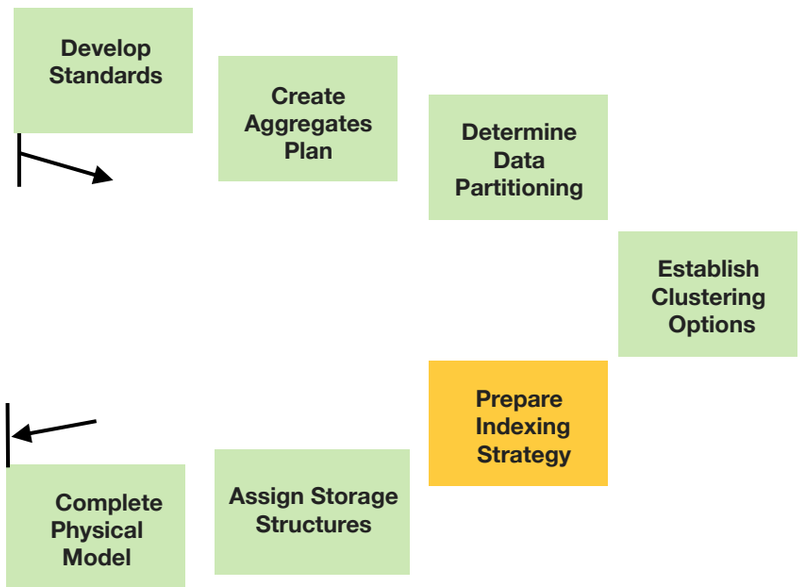
การใช้งานข้อมูลคลังข้อมูลของผู้ใช้แต่ละรายจะมีรูปแบบการใช้งาน ซึ่งโดยส่วนใหญ่จะเป็นรูปแบบการใช้งานคิวรีที่เกิดขึ้นบ่อย ๆ เมื่อเราทราบถึงรูปแบบในการใช้งานแล้ว เราจะต้องคิดพิจารณารูปแบบเหล่านั้นเพื่อนำมาปรับปรุงคุณภาพของการจัดเก็บข้อมูลที่จะช่วยให้สามารถเข้าถึงข้อมูลได้รวดเร็วยิ่งขึ้น โดยเทคนิคที่สามารถช่วยเพิ่มประสิทธิภาพในการจัดเก็บข้อมูล ได้แก่ การเก็บข้อมูลที่เกี่ยวเนื่องกันไว้ด้วยกัน ซึ่งจะเริ่มจากการพิจารณาข้อมูลในแต่ละตาราง จากนั้นทำการค้นหาตารางอื่น ๆ ที่เกี่ยวข้องกับตารางที่ถูกพิจารณา

การทำงานดังกล่าวจะทำให้เราสามารถได้ 2 ตารางที่มีความเกี่ยวเนื่องกันของข้อมูลตามรูปแบบการใช้งานของผู้ใช้ ขั้นตอนต่อไป เราต้องวางแผนเกี่ยวกับการจัดเก็บตารางที่มีความเกี่ยวเนื่องกันให้อยู่ใกล้กันมากที่สุดเท่าที่จะเป็นไปได้ ซึ่งอาจทำการเก็บข้อมูลไว้ในแฟ้มข้อมูลเดียวกัน โดยทำการเก็บข้อมูลสลับกัน

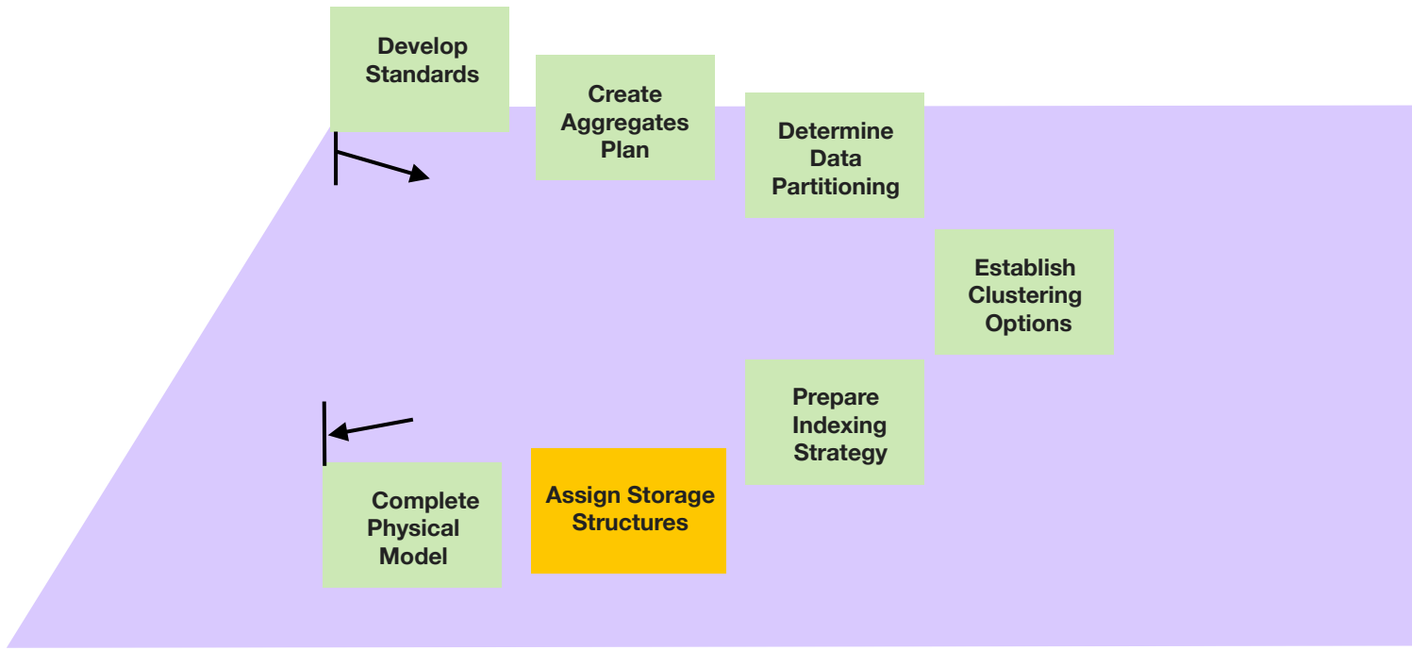
ตัวอย่างเช่น จัดเก็บข้อมูลเรคคอร์ดหนึ่ง ๆ จากตารางที่หนึ่ง แล้วตามด้วยทุก ๆ เรคคอร์ดจากตารางที่สองที่มีความเกี่ยวข้องกับเรคคอร์ดจากตารางที่หนึ่งที่ถูกเก็บไว้แล้ว จากนั้นทำการเก็บข้อมูลเรคคอร์ดใหม่จากตารางที่หนึ่งอีกครั้ง ทำการเก็บข้อมูลสลับไปสลับมาจนกระทั่งจัดเก็บข้อมูลไว้ทั้งหมด การเก็บข้อมูลด้วยวิธีนี้จะช่วยให้เราสามารถเรียกใช้ข้อมูลที่เชื่อมต่อกันได้ง่าย ซึ่งการเก็บข้อมูลที่ดีจะช่วยเพิ่มประสิทธิภาพในการใช้งานได้มากขึ้น

ขั้นตอนที่ 5 การเตรียมการสร้างดัชนีข้อมูล (Prepare an Indexing Strategy)

การสร้างดัชนีให้กับข้อมูลหรือตารางของข้อมูลเป็นขั้นตอนที่สำคัญมากสำหรับการออกแบบโมเดลเชิงกายภาพ ซึ่งการสร้างดัชนีจะช่วยเพิ่มประสิทธิภาพให้กับการเข้าถึง/ค้นคืนข้อมูล โดยก่อนที่จะทำการสร้างดัชนีเราจะต้องทำการวางแผนว่าเราจะสร้างดัชนีที่ตารางใด และที่คอลัมน์ใด ดังนั้นเราควรจะต้องให้เวลากับการออกแบบการสร้างดัชนีให้มาก รายละเอียดของวิธีการสร้างดัชนีให้กับข้อมูลจะกล่าวในส่วนต่อไปของบทนี้

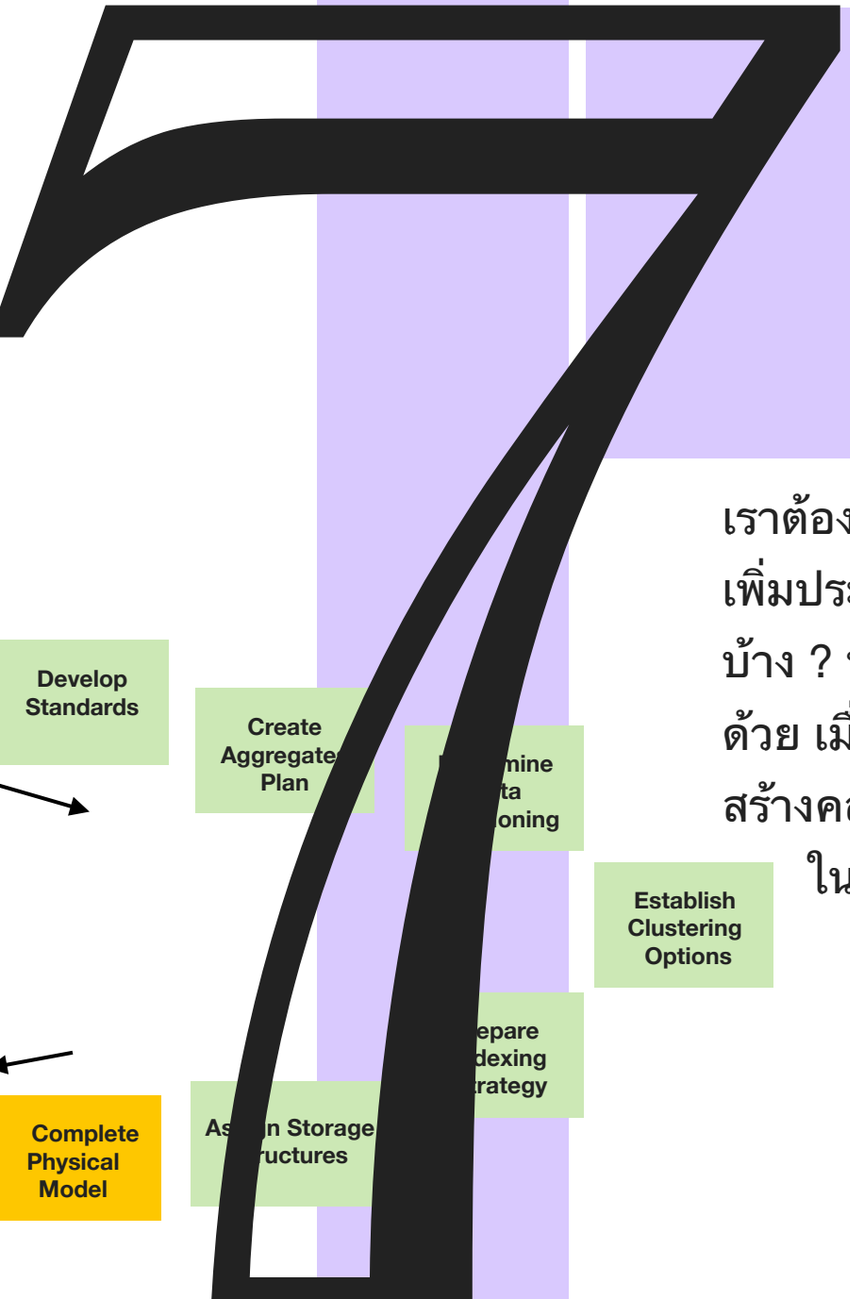


ขั้นตอนที่ 6 การกำหนดโครงสร้างการจัดเก็บข้อมูล (Assign Storage Strucutures)



ในการจัดเก็บข้อมูลเราจะต้องพิจารณาว่าเราควรเก็บข้อมูลไว้ที่ใด ? เราควรแบ่งแต่ละ physical file ออกเป็นกลุ่มก้อนของข้อมูลหรือไม่ ? ในการที่จะตอบคำถามเหล่านี้เราจะต้องทำการวางแผนเกี่ยวกับการจัดเก็บข้อมูล ซึ่งการเก็บข้อมูลในระบบคลังข้อมูลโดยส่วนใหญ่จะเป็นการเก็บข้อมูลลงใน physical files, temporary data extract files, staging area และแหล่งที่ใช้เก็บข้อมูลอื่น ๆ ดังนั้นเราจะต้องวางแผนเพื่อให้การจัดเก็บข้อมูลลงในทุกพื้นที่ข้างต้นมีความสอดคล้องกัน

ขั้นตอนที่ 7 การทำให้แบบจำลองทางกายภาพสมบูรณ์ (Complete Physical Model)



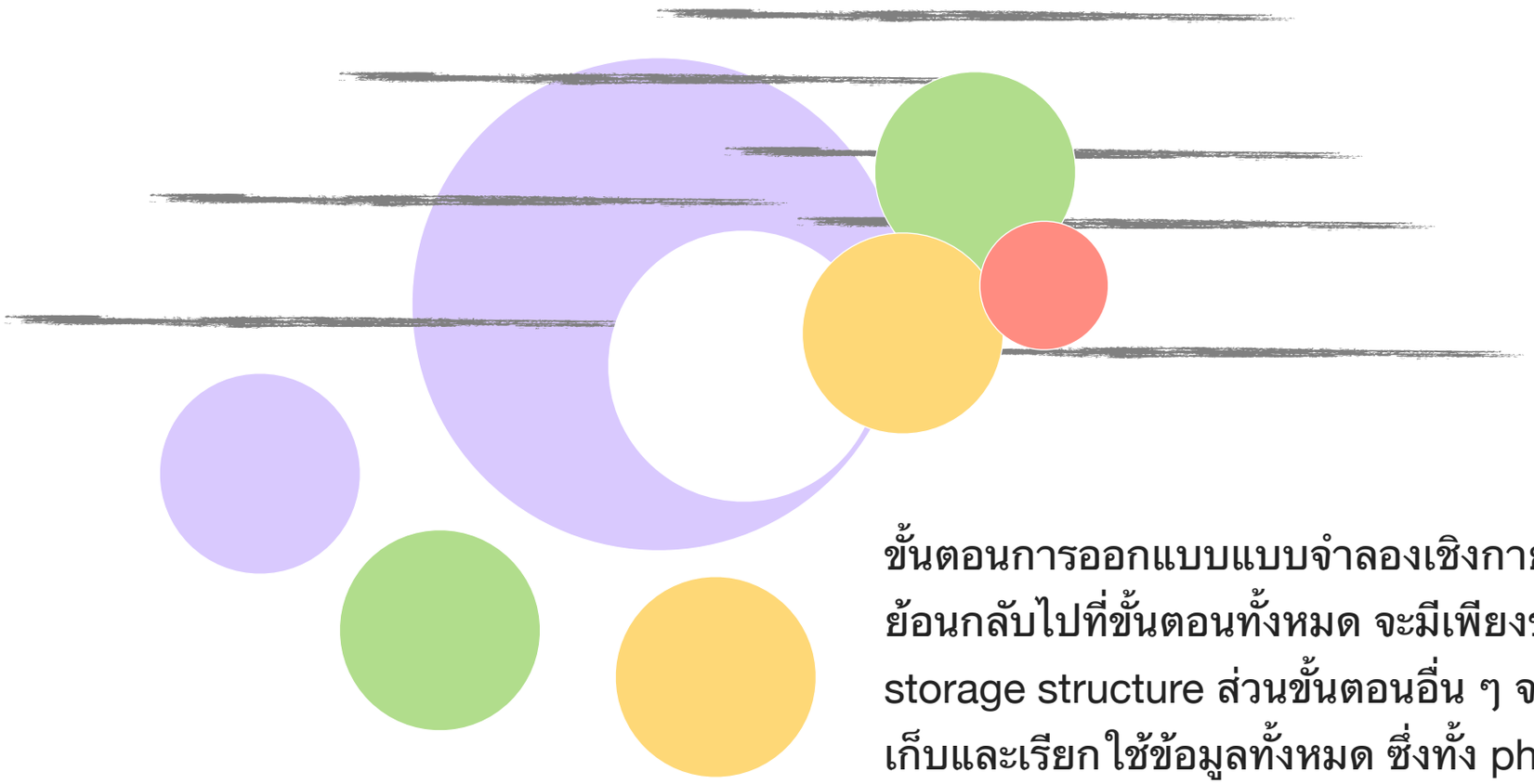
ขั้นตอนนี้เป็นขั้นตอนสุดท้ายที่จะทำการทบทวนและยืนยันความสมบูรณ์ของการทำงานใน 6 ขั้นตอนก่อนหน้า เมื่อมาถึงขั้นตอนนี้เราจะทราบถึงมาตราฐานในการตั้งชื่อฐานข้อมูล การตั้งชื่อตารางข้อมูล การตั้งชื่อแอทริบิวหรือฟิลด์ต่าง ๆ โดยที่เราต้องทำการพิจารณาว่าตารางที่ทำการรวมกันนั้นมีความจำเป็นหรือไม่ ? และมีตารางใดบ้างที่สมควรจะทำการแบ่งออกเป็นส่วนย่อย ๆ อีกบ้าง ?

เราต้องพิจารณาถึงการสร้างดัชนีให้กับข้อมูลด้วยว่าการสร้างดัชนีในขั้นตอนที่ผ่านมานั้นจะช่วยเพิ่มประสิทธิภาพในการเข้าถึงข้อมูลหรือไม่ ? หรือยังมีข้อบกพร่องใดเกิดขึ้นในการสร้างดัชนีอีกบ้าง ? นอกจากนี้เรายังต้องคิดถึงทางเลือกอื่น ๆ สำหรับการเพิ่มประสิทธิภาพในการทำงานอีกด้วย เมื่อเราทำการตรวจสอบกระบวนการทั้งหมดแล้ว เราจะได้ physical schema สำหรับการสร้างคลังข้อมูล หลังจากนั้นเราสามารถสร้าง physical structure ในดาต้าดิกชันนารี เพื่อช่วยในการสร้างหรือดูแลรักษาคลังข้อมูลต่อไป

SECTION 4

ปัจจัยที่ต้องพิจารณาในการออกแบบโมเดลทางกายภาพ

ปัจจัยที่ต้องพิจารณาในการออกแบบโมเดลทางกายภาพ



ขั้นตอนการออกแบบแบบจำลองเชิงกายภาพดังที่ได้กล่าวมาแล้ว เมื่อเรามองย้อนกลับไปที่ขั้นตอนทั้งหมด จะมีเพียงขั้นตอนเดียวที่เกี่ยวข้องกับ physical storage structure ส่วนขั้นตอนอื่น ๆ จะเกี่ยวข้องกับประสิทธิภาพในการจัดเก็บและเรียกใช้ข้อมูลทั้งหมด ซึ่งทั้ง physical storage structure และประสิทธิภาพการทำงานเป็น 2 ปัจจัยที่สำคัญมากในการสร้างคลังข้อมูล ในการที่จะสร้างแบบจำลองเชิงกายภาพจากแบบจำลองเชิงตรรกะที่สร้างขึ้นก่อนหน้าจะมีองค์ประกอบอีกมาให้ต้องคิดและพิจารณาอีกมากดังต่อไปนี้

2

วัตถุประสงค์ที่ 2
เพิ่มความมั่นใจในการยืดขยายต่อเติม
(Ensure Scalability)

1

วัตถุประสงค์ที่ 1
การเพิ่มประสิทธิภาพในการ
ทำงาน (Improve Performance)

3

วัตถุประสงค์ที่ 3
การจัดการเกี่ยวกับการจัดเก็บ
ข้อมูล (Manage Storage)

5

วัตถุประสงค์ที่ 5
เพิ่มความยืดหยุ่น (Design for flexibility)

4

วัตถุประสงค์ที่ 4
การเพิ่มประสิทธิภาพในการ
ทำงาน (Improve Performance)

วัตถุประสงค์ของการออกแบบโมเดลทางกายภาพ

เป้าหมายของการสร้างแบบจำลองเชิงตรรกะของฐานข้อมูล คือ การสร้างแบบจำลองแนวคิด (Conceptual model) ที่สะท้อนถึงรายละเอียดของข้อมูล และจะแสดงถึงองค์ประกอบทั้งหมดของข้อมูลรวมถึงความสัมพันธ์ของข้อมูลทั้งหมดด้วย แต่การสร้างแบบจำลองเชิงกายภาพจะมีวัตถุประสงค์ที่แตกต่างออกไป จะเน้นย้ำที่การทำงานที่เกี่ยวข้องกับระบบปฏิบัติการ ซอร์ฟแวร์ฐานข้อมูล ฮาร์ดแวร์ และแพลตฟอร์มคอมพิวเตอร์ ดั้งนั้นเราสามารถกล่าวได้ว่าแบบจำลองเชิงกายภาพจะสนใจเกี่ยวกับการที่จะสามารถทำงานได้อย่างไรมากกว่าที่จะสนใจในเรื่องของการเรียกดูข้อมูลจากโมเดลได้อย่างไร

Conceptual model

สมมติว่าถ้าเราต้องทำการสรุปผลข้อมูล เป้าหมายของการออกแบบแบบจำลองเชิงกายภาพจะเกี่ยวข้องกับการเพิ่มประสิทธิภาพในการจัดเก็บและเรียกใช้ข้อมูล ซึ่งเราจะต้องพิจารณาถึงความถี่/จำนวนครั้งในการใช้งานข้อมูลว่ามีการใช้งานบ่อยครั้งเพียงใด ปริมาณข้อมูลที่ต้องทำการจัดเก็บหรือเรียกดูเป็นเท่าไรและอื่น ๆ ในการออกแบบจำลองเชิงกายภาพ เราจำเป็นต้องใส่ใจกับปัจจัยที่เกี่ยวข้องและวิเคราะห์แต่ละขั้นตอนสำหรับการเพิ่มประสิทธิภาพโดยการพิจารณาวัตถุประสงค์ที่สำคัญดังต่อไปนี้

วัตถุประสงค์ที่ 1
การเพิ่มประสิทธิภาพในการทำงาน (Improve Performance)

โดยส่วนใหญ่ประสิทธิภาพการทำงานของระบบจะเกี่ยวข้องกับเวลาที่ใช้ในการค้นคืนข้อมูลให้กับคิวรีของผู้ใช้ ซึ่งในระบบการดำเนินงานทั่วไปควรจะสามารถสืบค้นข้อมูลได้ภายในเวลาไม่เกิน 3 วินาที แต่สำหรับคลังข้อมูลแล้วเวลาที่ใช้ในการค้นคืนข้อมูลจะมีความเข้มงวดน้อยกว่าระบบการดำเนินงาน กล่าวคือ เวลาที่ใช้สำหรับคลังข้อมูลจะมีความหลากหลายมากกว่า เริ่มตั้งแต่การใช้เวลาเพียงไม่กี่วินาทีไปจนถึง 2-3 นาที เวลาที่ใช้จะขึ้นอยู่กับปริมาณข้อมูลที่ต้องทำการประมวลผลเพื่อตอบคิวรีจากผู้ใช้

ดังนั้นเราต้องทำความเข้าใจกับผู้ใช้ว่าเวลาที่ใช้ในการค้นคืนข้อมูลจากคลังข้อมูลอาจนานกว่าการสืบค้นข้อมูลจากระบบการดำเนินงาน แต่อย่างไรก็ดี ในยุคปัจจุบันคลังข้อมูลมีการใช้ OLAP system ซึ่งทำให้การใช้เวลา 2-3 นาทีในการค้นคืนข้อมูลนั้นไม่สามารถยอมรับได้ ดังนั้นเราจึงต้องพยายามที่จะปรับปรุงประสิทธิภาพการทำงานเพื่อให้เวลาในการตอบคิวรีอยู่ในระดับที่ยอมรับได้

นอกจากนี้เราจะต้องมีการเฝ้าสังเกตุประสิทธิภาพของการทำงาน และจะต้องมีการปรับจูนประสิทธิภาพของคลังข้อมูลอย่างสม่ำเสมอ โดยจากการเฝ้าสังเกตุประสิทธิภาพของการทำงานจะกระทำโดยผู้ดูแลระบบซึ่งจะพิจารณาทั้งการออกแบบฐานข้อมูลในเชิงตรรกะการออกแบบแอพพลิเคชัน และ รูปแบบของคิวรีซึ่งทั้งหมดนี้จะส่งผลต่อประสิทธิภาพของการทำงานโดยรวม

วัตถุประสงค์ที่ 2
เพิ่มความมั่นใจในการยืดขยายต่อเติม
(Ensure Scalability)

เป้าหมายนี้เป็นเป้าหมายหลักของการออกแบบแบบจำลองเชิงกายภาพ เนื่องจากการใช้งานของคลังข้อมูลเพิ่มขึ้นตลอดเวลา จำนวนผู้ใช้ก็เพิ่มขึ้นอย่างรวดเร็ว และคิวรีที่ใช้ก็มีความซับซ้อนมากขึ้น ดังนั้นในการสร้างแบบจำลองเชิงกายภาพควรจะเน้นย้ำที่การใช้งานคลังข้อมูลที่เพิ่มขึ้น

วัตถุประสงค์ที่ 3
การจัดการเกี่ยวกับการจัดเก็บข้อมูล (Manage Storage)

การจัดการที่ดีที่เกี่ยวกับการจัดเก็บข้อมูลจะสามารถช่วยเพิ่มประสิทธิภาพการทำงานของคลังข้อมูลได้ โดยที่เราสามารถเพิ่มประสิทธิภาพโดยการเก็บข้อมูลจากตารางที่เกี่ยวเนื่องกันไว้ในไฟล์เดียวกัน หรือเราสามารถจัดการตารางใหญ่ๆ ได้ง่ายขึ้นโดยการเก็บข้อมูลส่วนต่าง ๆ ไว้ในที่จัดเก็บข้อมูลที่แตกต่างกัน

วัตถุประสงค์ที่ 4
การเพิ่มประสิทธิภาพในการทำงาน (Improve Performance)

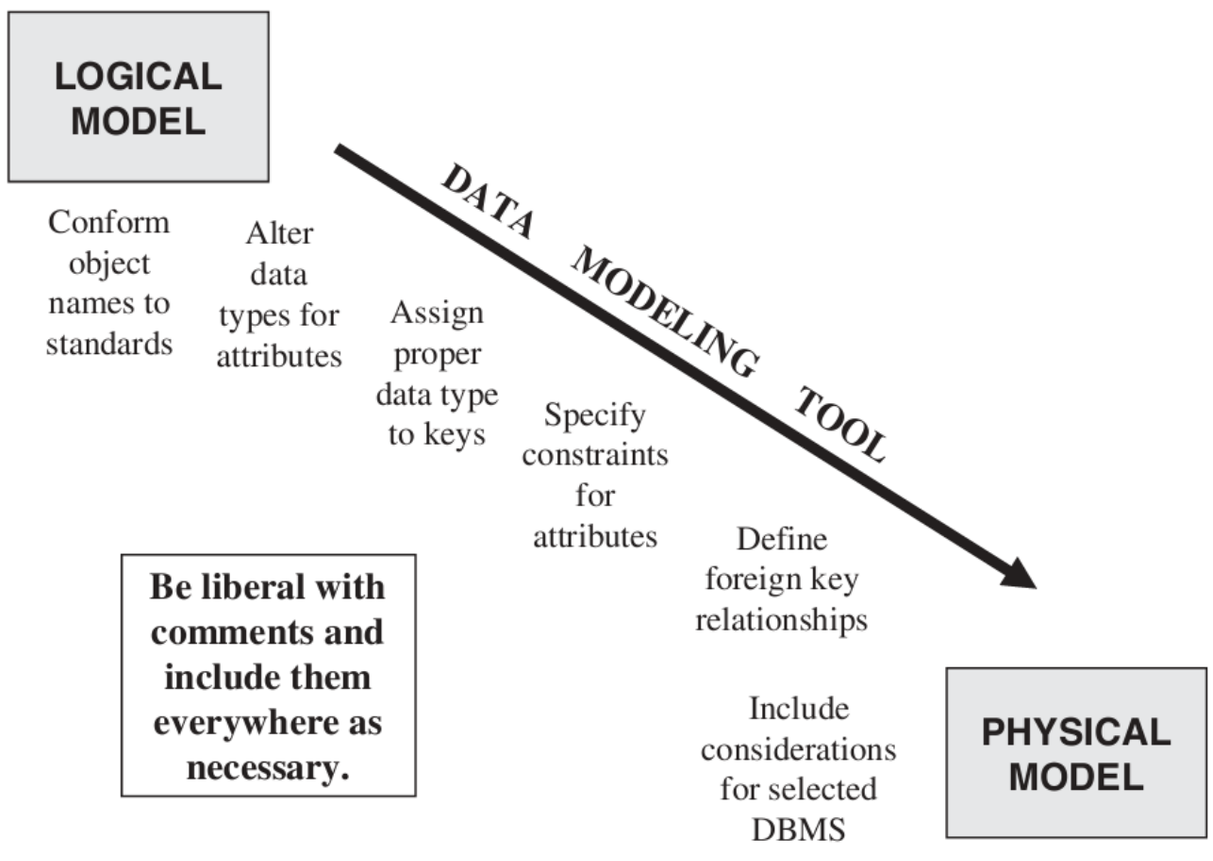
วัตถุประสงค์นี้จะครอบคลุมถึงกิจกรรมต่าง ๆ ที่จะทำให้การดูแลคลังข้อมูลสามารถทำได้โดยง่าย ตัวอย่างเช่น ความสะดวกของการดูแลคลังข้อมูลจะเกี่ยวข้องกับวิธีที่เหมาะสมในการจัดเรียงแถวของตารางในพื้นที่สำหรับจัดเก็บข้อมูลโดยไม่ทำการจัดระเบียบข้อมูลบ่อยจนเกินไป ความสะดวกอื่น ๆ ในการดูแลคลังข้อมูลจะเกี่ยวกับการการสำรองและการกู้คืนตารางต่างๆของฐานข้อมูล ดังนั้นเราต้องลองคิดพิจารณาวิธีการหรือขั้นตอนที่ต้องทำงานกับพื้นที่ในการจัดเก็บข้อมูล หรือ DBMS ที่จะช่วยให้การดูแลคลังข้อมูลมีความสะดวกมากขึ้น

วัตถุประสงค์ที่ 5
เพิ่มความยืดหยุ่น (Design for flexibility)

ในแง่มุมของการออกแบบแบบจำลองเชิงกายภาพ ความยืดหยุ่นจะหมายถึงการทำให้การออกแบบสามารถรองรับการทำงานต่างๆ ได้ โดยที่การทำงานของคลังข้อมูลอาจมีการเปลี่ยนแปลงเกิดขึ้นจากความต้องการของผู้ใช้หรือเทคโนโลยีที่เปลี่ยนแปลงไป ดังนั้น ในการออกแบบแบบจำลองเชิงกายภาพเราจะต้องออกแบบให้การทำงานมีความยืดหยุ่นต่อความต้องการในอนาคตด้วย

การออกแบบแบบจำลองทางกายภาพจากแบบจำลองเชิงตรรกะ

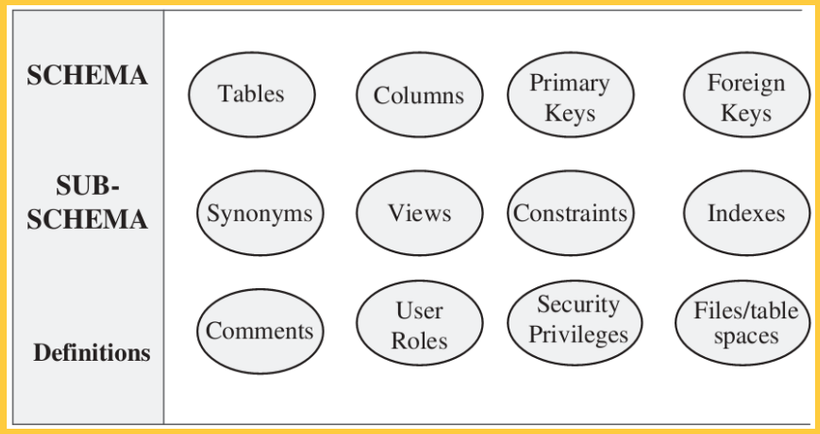
อย่างที่เราทราบกันดีว่า แบบจำลองเชิงตรรกะประกอบด้วย ตาราง แอทริบิว คีย์หลัก และความสัมพันธ์ของข้อมูล แต่สำหรับแบบจำลองเชิงกายภาพประกอบด้วยโครงสร้างของฐานข้อมูลและความสัมพันธ์ที่ถูกแสดงอยู่ในรูปแบบของ Database schema coded ของ data definition language (DDL) ของ DBMS โดยที่การสร้างแบบจำลองเชิงกายภาพจากแบบจำลองเชิงตรรกะจะแสดงในรูปที่ 12-2 ซึ่งจากรูปจะแสดงถึงขั้นตอนการทำงานจากแบบจำลองเชิงตรรกะไปสู่ฟังก์ชันการทำงานในแบบจำลองเชิงกายภาพ ซึ่งมีรายละเอียดดังนี้



รูปที่ 12-2 การออกแบบแบบจำลองทางกายภาพจากแบบจำลองเชิงตรรกะ

ส่วนประกอบของแบบจำลองทางกายภาพ

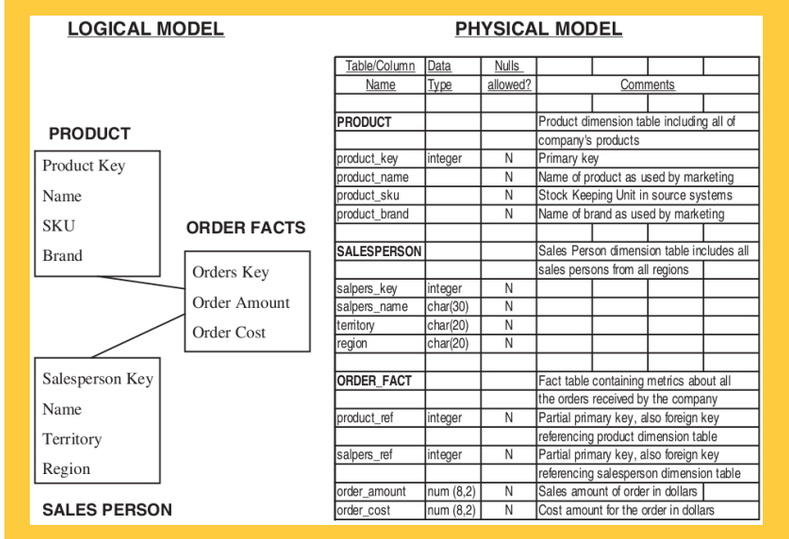
รายละเอียดของแบบจำลองเชิงกายภาพจะแสดงถึงเนื้อหาสาระของข้อมูลในระดับที่ใกล้ฮาร์ดแวร์มาก ซึ่งจะทำให้เราทราบถึงรายละเอียดต่าง ๆ เช่น ขนาดของไฟล์ที่ใช้เก็บข้อมูล ความยาวของแต่ละฟิลด์ที่ใช้ในการจัดเก็บข้อมูล คีย์หลัก คีย์รอง และอื่น ๆ รูปที่ 12-3 แสดงถึงส่วนประกอบหลักของแบบจำลองเชิงกายภาพที่แสดงส่วนประกอบในรูปแบบของดาต้าดิกชันนารีของ DBMS ผ่าน schemas และ subschemas เราสามารถใช้ data definition language ในการเขียน schema ต่าง ๆ ดังแสดงในรูปที่ 12-4 ใน schema จะแสดงถึงฐานข้อมูล ตาราง และคอลัมน์ในแต่ละตารางด้วย ดังนั้นเมื่อเราต้องการที่จะสร้างแบบจำลองเชิงกายภาพจากแบบจำลองเชิงตรรกะ เราอาจจำเป็นต้องเชื่อมโยงความสัมพันธ์ระหว่างส่วนประกอบต่าง ๆ จากทั้งสองโมเดลเข้าด้วยด้วยกัน ดังแสดงในรูปที่ 12-5



รูปที่ 12-3

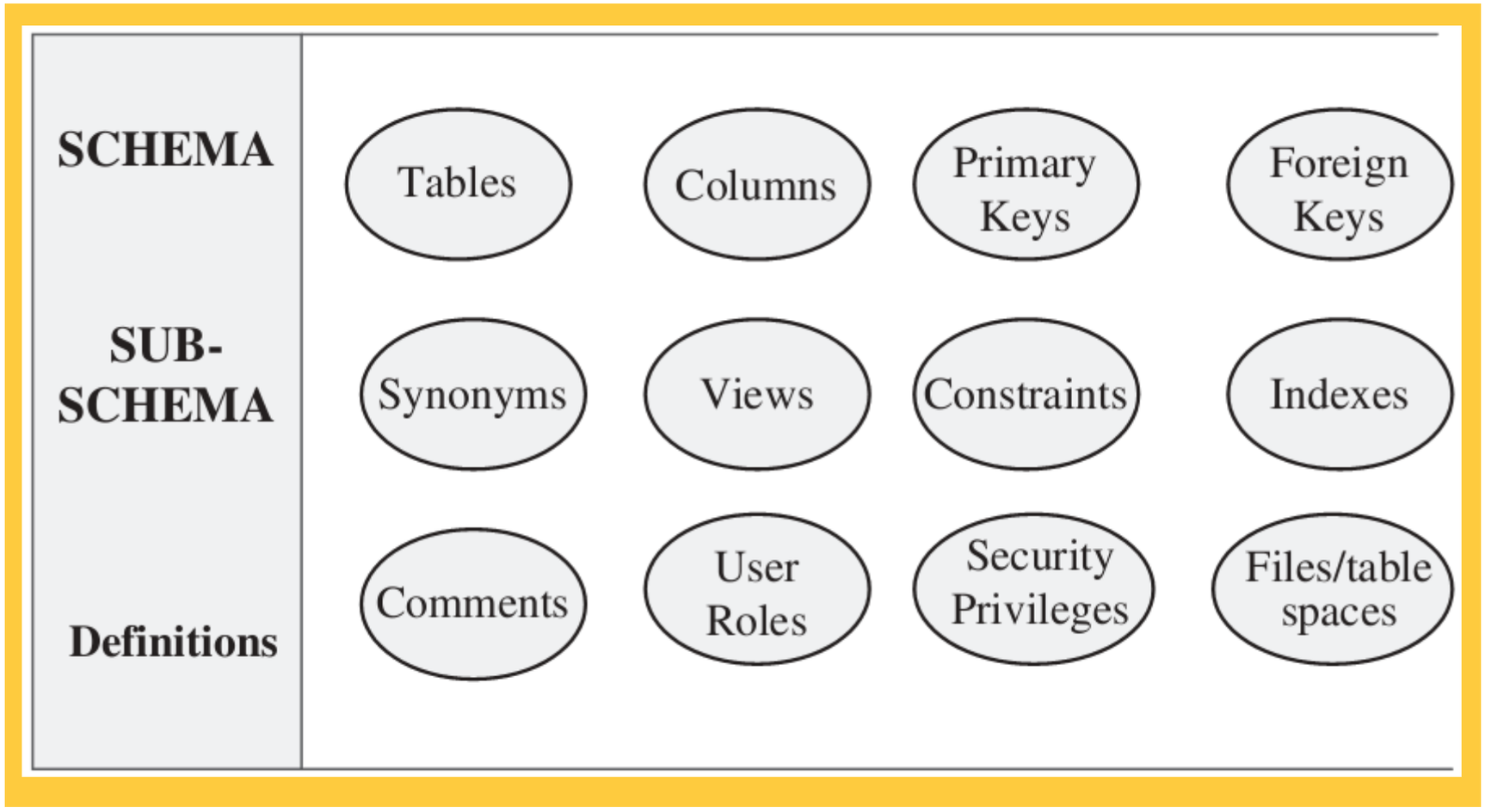
```
CREATE SCHEMA ORDER_ANALYSIS
  AUTHORIZATION SAMUEL_JOHNSON
  ……..
CREATE TABLE PRODUCT  (
  PRODUCT_KEY          CHARACTER (8)
                       PRIMARY KEY,
  PRODUCT_NAME         CHARACTER (25),
  PRODUCT_SKU          CHARACTER (20),
  PRODUCT_ BRAND       CHARACTER (25))

CREATE TABLE SALESPERSON  (
  SALPERS_KEY          CHARACTER (8)
                       PRIMARY KEY,
  SALPERS_NAME         CHARACTER (30),
  TERRITORY            CHARACTER (20),
  REGION               CHARACTER (20))

CREATE TABLE ORDER_FACT  (
  PRODUCT_REF          CHARACTER (8)
                       PRIMARY KEY,
  SALPERS_REF          CHARACTER (8)
                       PRIMARY KEY,
  ORDER_AMOUNT         NUMERIC (8,2),
  ORDER_COST           NUMERIC (8,2),
  FOREIGN KEY PRODUCT_REF
                       REFERENCES PRODUCT,
  FOREIGN KEY SALPERS_REF
                       REFERENCES SALESPERSON)
```

รูปที่ 12-4



Table/Column Name	Data Type	Nulls allowed?	Comments
PRODUCT			Product dimension table including all of company's products
product_key	integer	N	Primary key
product_name		N	Name of product as used by marketing
product_sku		N	Stock Keeping Unit in source systems
product_brand		N	Name of brand as used by marketing
SALESPERSON			Sales Person dimension table includes all sales persons from all regions
salpers_key	integer	N	
salpers_name	char(30)	N	
territory	char(20)	N	
region	char(20)	N	
ORDER_FACT			Fact table containing metrics about all the orders received by the company
product_ref	integer	N	Partial primary key, also foreign key referencing product dimension table
salpers_ref	integer	N	Partial primary key, also foreign key referencing salesperson dimension table
order_amount	num (8,2)	N	Sales amount of order in dollars
order_cost	num (8,2)	N	Cost amount for the order in dollars

รูปที่ 12-5



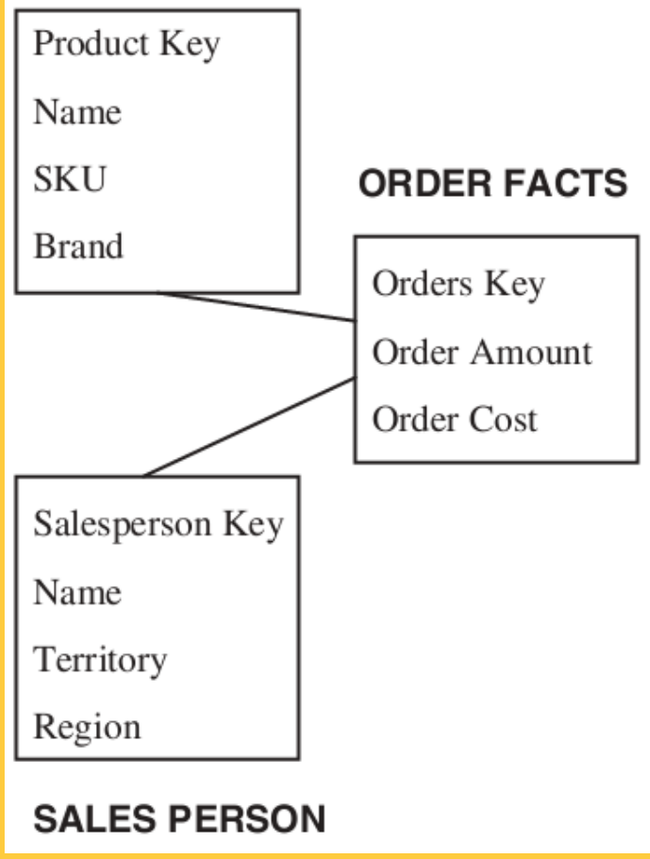
รูปที่ 12-3 ส่วนประกอบของแบบจำลองทางกายภาพ

```
CREATE SCHEMA ORDER_ANALYSIS
  AUTHORIZATION SAMUEL_JOHNSON
  ……...
CREATE TABLE PRODUCT  (
  PRODUCT_KEY          CHARACTER (8)
                       PRIMARY KEY,
  PRODUCT_NAME         CHARACTER (25),
  PRODUCT_SKU          CHARACTER (20),
  PRODUCT_ BRAND       CHARACTER (25))

CREATE TABLE SALESPERSON  (
  SALPERS_KEY          CHARACTER (8)
                       PRIMARY KEY,
  SALPERS_NAME         CHARACTER (30),
  TERRITORY            CHARACTER (20),
  REGION               CHARACTER (20))
```

```
CREATE TABLE ORDER_FACT  (
  PRODUCT_REF          CHARACTER (8)
                       PRIMARY KEY,
  SALPERS_REF          CHARACTER (8)
                       PRIMARY KEY,
  ORDER_AMOUNT         NUMERIC (8,2),
  ORDER_COST           NUMERIC (8,2),
  FOREIGN KEY PRODUCT_REF
                       REFERENCES PRODUCT,
  FOREIGN KEY SALPERS_REF
                       REFERENCES SALESPERSON)
```

รูปที่ 12-4 ตัวอย่างของ schema definition ใน SQL

LOGICAL MODEL



PHYSICAL MODEL

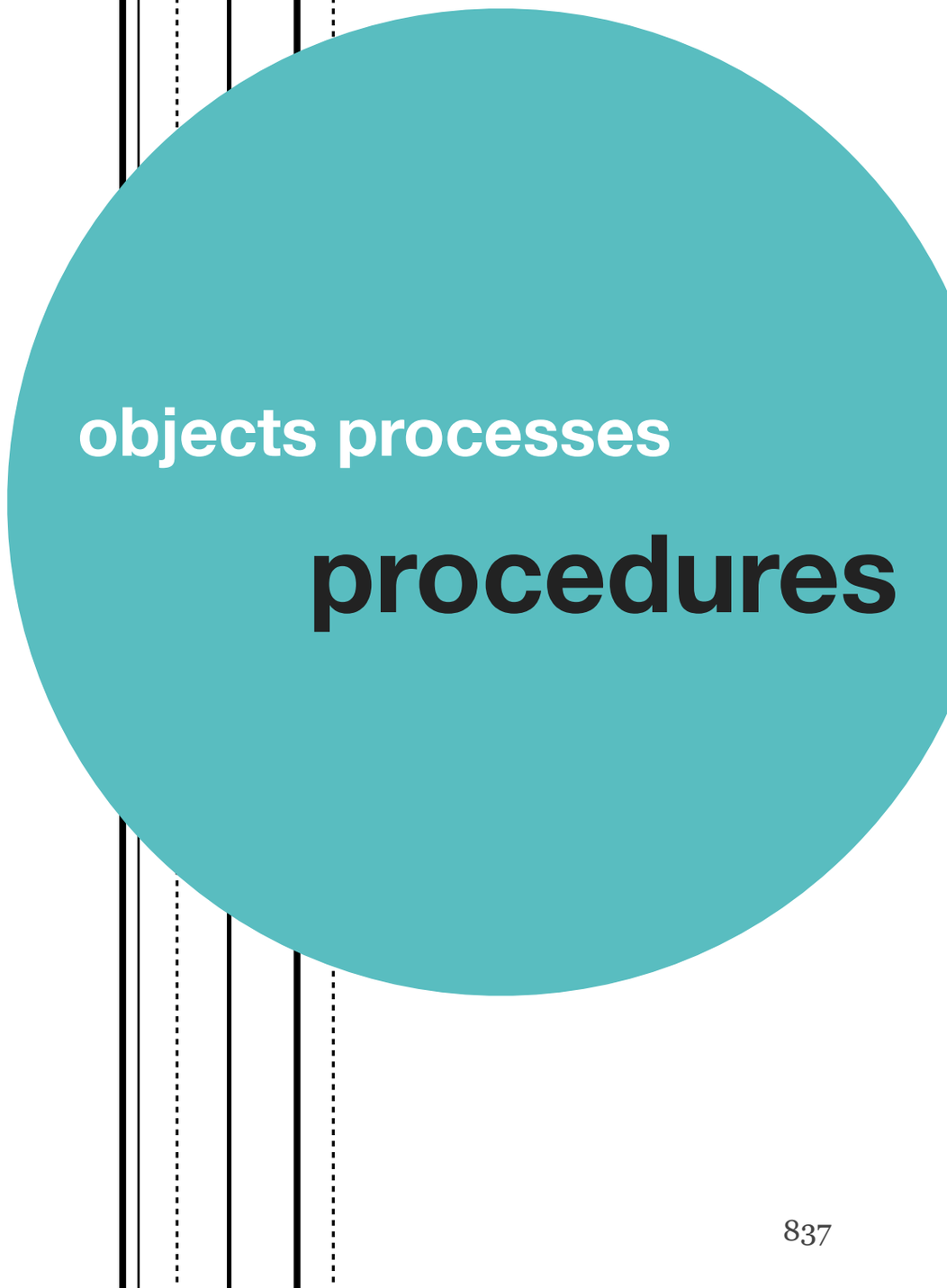
Table/Column Name	Data Type	Nulls allowed?	Comments
PRODUCT			Product dimension table including all of
			company's products
product_key	integer	N	Primary key
product_name		N	Name of product as used by marketing
product_sku		N	Stock Keeping Unit in source systems
product_brand		N	Name of brand as used by marketing
SALESPERSON			Sales Person dimension table includes all
			sales persons from all regions
salpers_key	integer	N	
salpers_name	char(30)	N	
territory	char(20)	N	
region	char(20)	N	
ORDER_FACT			Fact table containing metrics about all
			the orders received by the company
product_ref	integer	N	Partial primary key, also foreign key
			referencing product dimension table
salpers_ref	integer	N	Partial primary key, also foreign key
			referencing salesperson dimension table
order_amount	num (8,2)	N	Sales amount of order in dollars
order_cost	num (8,2)	N	Cost amount for the order in dollars

รูปที่ 12-5 ตัวอย่างแบบจำลองเชิงตรรกะและแบบจำลองทางกายภาพ

ความสำคัญของมาตรฐานของข้อมูล

มาตรฐานในคลังข้อมูลจะครอบคลุมถึง objects processes และ procedures เมื่อเราพิจารณาแบบจำลองเชิงกายภาพของคลังข้อมูล เราจะพบว่ามาตรฐานในการตั้งชื่อของ objects นั้นเป็นสิ่งสำคัญมาก ชื่อที่เป็นมาตรฐานจะทำให้เรามีชื่อที่มีความหมายที่สอดคล้องกันทั้งระบบ ซึ่งชื่อที่สอดคล้องกันจะช่วยเพิ่มประสิทธิภาพในการติดต่อสื่อสารระหว่างขั้นตอนการดำเนินงานต่าง ๆ ได้เป็นอย่างดี

อย่างที่เราทราบกันดี ในการใช้งานคลังข้อมูล ผู้ใช้จะมีส่วนร่วมโดยตรงในการเข้าถึงข้อมูลค่อนข้างมาก เนื่องจากผู้ใช้สามารถกำหนดคิวรีที่ต้องการเรียกดูข้อมูลเองได้ ดังนั้นถ้าเรามีขั้นตอนการสื่อสารที่ชัดเจนจะช่วยให้ผู้ใช้สามารถใช้งานได้สะดวกยิ่งขึ้น



การตั้งชื่อตารางและแอทริบิวต่าง ๆ ในฐานข้อมูล

ในการตั้งชื่อให้กับวัตถุใด ๆ ก็ตาม เช่น ตาราง แอทริบิว ขั้นตอนกระบวนการต่าง ๆ และอื่น ๆ เราจะต้องมีวิธีการที่ดีในการตั้งชื่อที่จะสื่อถึงความหมายและคำอธิบายของวัตถุนั้น ๆ

ตัวอย่างเช่น ชื่อของคอลัมน์หนึ่ง คือ customer_loan_balance จะทำให้เราทราบถึงข้อมูลจำนวนเงินคงเหลือในบัญชีของลูกค้าที่กู้ยืมเงินจากธนาคารจากตัวอย่างเราจะเห็นว่าชื่อของคอลัมน์นั้นถูกตั้งโดยใช้คำหลายๆคำมาเชื่อมต่อกันซึ่งจะสามารถสื่อความหมายได้ดีกว่าการใช้คำเพียงคำเดียว

ดังนั้นเราสามารถสร้างฟังก์ชันมาตรฐานสำหรับแต่ละคำในกลุ่มของคำที่ใช้ในการตั้งชื่อคอลัมน์ต่าง ๆ ได้ จากตัวอย่างข้างต้น คำแรกจะแสดงถึงหัวข้อหลัก (Primary subject) คำที่สามจะหมายถึงหมวดหมู่ของวัตถุ (Class of object) และคำที่สองจะเป็นคุณสมบัติของหมวดหมู่ (Qualify the class) โดยส่วนใหญ่ในการแยกคำในการตั้งชื่อจะใช้ “-” หรือ “_” ซึ่งทั้งสองตัวแบ่งจะสามารถใช้งานได้ดีกับ DBMS อีกด้วย

การตั้งชื่อในแบบจำลองเชิงตรรกะและแบบจำลองทางกายภาพ

ในการใช้งานคลังข้อมูลผู้วิเคราะห์ข้อมูลและผู้ออกแบบแบบจำลองเชิงตรรกะจะทำการติดต่อสื่อสารกับผู้อื่นโดยใช้ชื่อของวัตถุจากแบบจำลองเชิงตรรกะ แต่สำหรับการที่จะอ้างถึงตารางและคอลัมน์เพื่อทำการค้นคืนข้อมูล ผู้ใช้จะทำการติดต่อสื่อสารกับระบบโดยใช้ชื่อของวัตถุจากแบบจำลองเชิงกายภาพซึ่งจากการใช้งานดังกล่าว เราจะเห็นว่ามีการอ้างถึงชื่อของวัตถุในทั้งสองแบบจำลอง ดังนั้นเพื่อให้การตั้งชื่อวัตถุมีมาตรฐานเราควรจะต้องตั้งชื่อวัตถุเดียวกันกับวัตถุที่ปรากฏในแบบจำลองเชิงตรรกะและแบบจำลองเชิงกายภาพให้มีความเหมือนกัน

การตั้งชื่อแฟ้มข้อมูลและตารางใน Staging area

อย่างที่เราทราบกันดีว่า staging area เปรียบเสมือนพื้นที่สำหรับพักข้อมูลที่ได้จากการสกัดข้อมูลก่อนที่จะทำการถ่ายโอนข้อมูลไปยังคลังข้อมูล ซึ่งใน staging area เราอาจต้องทำการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล รวมถึงการรวมข้อมูลเข้าด้วยกัน แต่ด้วยเนื่องจากข้อมูลที่เข้าและออกจากคลังข้อมูลมีปริมาณค่อนข้างมากอาจประกอบไปด้วยหลายแฟ้มข้อมูล ซึ่งจะทำให้การพิจารณาข้อมูลใน staging area นั้นทำงานได้ค่อนข้างยาก

ดังนั้นถ้าเรามีการจัดการที่ดีที่สามารถบอกได้ว่าข้อมูลไฟล์ใด เก็บไว้สำหรับทำอะไรและการตั้งชื่อฟิลด์หรือแอททริบิวของข้อมูลเป็นไปอย่างมีมาตรฐานจะทำให้การทำงานสะดวกราบรื่นยิ่งขึ้น ลองพิจารณาข้อแนะนำดังต่อไปนี้เพื่อนำไปประยุกต์ใช้ในการตั้งชื่อไฟล์หรือตาราง ในคลังข้อมูล

Indicate the Process

จะเป็นการระบุวัตถุประสงค์การใช้งานให้กับแต่ละแฟ้มข้อมูล ถ้าแฟ้มข้อมูลเป็นผลลัพธ์ที่ได้หลังทำการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล เราควรจะตั้งชื่อแฟ้มที่บอกว่าเป็นผลลัพธ์จากการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลด้วย แต่ถ้าแฟ้มข้อมูลเป็นส่วนหนึ่งของข้อมูลที่มีการเปลี่ยนแปลงในแต่ละวัน เราก็ต้องทำให้ชื่อแฟ้มข้อมูลมีความชัดเจนด้วยเช่นกัน

Express the Purpose

สมมติว่าเรากำลังทำการกำหนดตารางเวลาสำหรับการอัพเดทข้อมูลประจำสัปดาห์ให้กับข้อมูลใน dimension table ที่เกี่ยวกับข้อมูลสินค้า เราจะต้องทราบถึงไฟล์ที่เป็นอินพุตก่อน ถ้าชื่อแฟ้มข้อมูลนั้นมีการบ่งบอกถึงวัตถุประสงค์ว่าต้องการที่จะทำอะไร จะเป็นการช่วยในการกำหนดตารางการอัพเดทข้อมูลได้มาก

ดังนั้นถ้าเรากำหนดมาตราฐานสำหรับชื่อไฟล์ต่างๆ ใน staging area โดยทำการเพิ่มวัตถุประสงค์เข้าไปในชื่อแฟ้มข้อมูลด้วย จะช่วยให้การทำงานเป็นไปอย่างสะดวกมากขึ้น

Examples

พิจารณาตัวอย่างชื่อแฟ้มข้อมูลใน staging area แล้วลองหาความหมายและมาตราฐานจากชื่อไฟล์เหล่านี้

- Sale_units_daily_stage
- Customer_daily_update
- Product_full_refresh
- Order_entryinitial_extract
- All_source_sales_extract
- Customer_nameaddr_daily_update

SECTION 5

การจัดเก็บข้อมูลทางกายภาพ

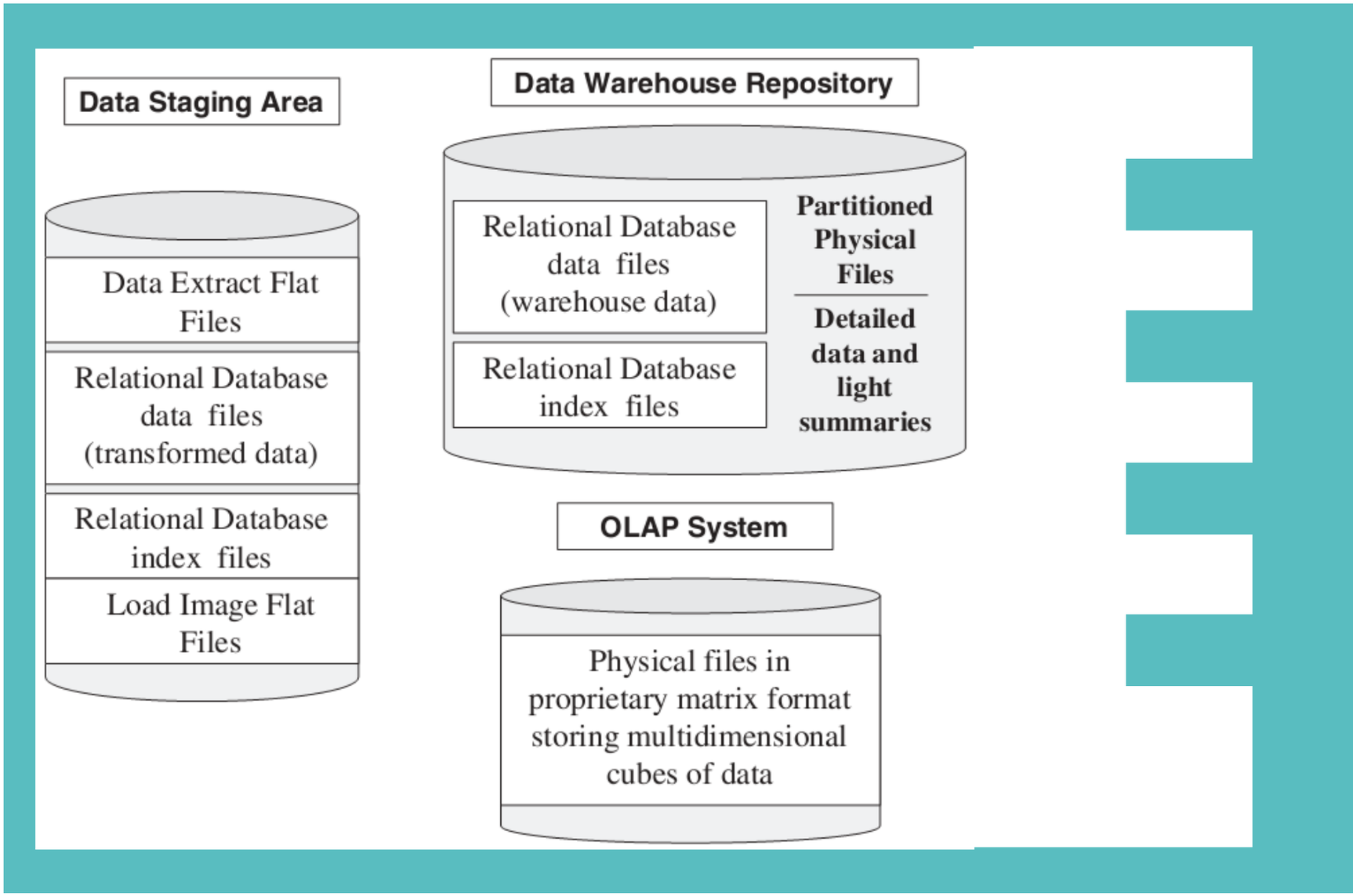
การจัดเก็บข้อมูลทางกายภาพ

ในการดำเนินการเกี่ยวกับการประมวลผลคิวรีหลังจากผู้ใช้ป้อนคิวรีแล้ว คิวรีที่ถูกสร้างขึ้นจะถูกตรวจสอบความถูกต้องของโครงสร้างและทำการตรวจสอบข้อมูลในคิวรีกับดาต้าดิกชันนารี หลังจากนั้น DBMS จะทำการแปลงคิวรีเพื่อค้นหาข้อมูลที่ต้องการจากข้อมูลตาราง แถว และคอลัมน์ที่เราต้องการ DBMS จะทำการเชื่อมโยงความต้องการข้อมูลเข้ากับ physical storage เพื่อเข้าถึงข้อมูลต่อไป โดยที่ประสิทธิภาพของการสืบค้นข้อมูลหรือการประมวลผลคิวรีจะเกี่ยวข้องกับ physical storage และการจัดเก็บข้อมูลโดยตรง ซึ่งการจัดเก็บข้อมูลอย่างมีประสิทธิภาพจะช่วยลดเวลาในการสืบค้นข้อมูลได้ค่อนมาก ซึ่งเราสามารถพิจาณาปัจจัยต่าง ๆ ได้ดังนี้

โครงสร้างการจัดเก็บข้อมูล

ข้อมูลที่จัดเก็บอยู่ในคลังข้อมูลจะถูกจัดเก็บอยู่ที่ staging area และถูกจัดเก็บอยู่ในฐานข้อมูลของคลังข้อมูลที่อยู่ในรูปของ ตารางข้อมูล หรือการทำดัชนีให้กับข้อมูลต่าง ๆ นอกจากนี้ยังรวมถึงข้อมูลที่มีหลายมิติซึ่งถูกเก็บอยู่ในระบบ OLAP ด้วย ซึ่งในการจัดเก็บข้อมูลเราจะต้องมองถึงประสิทธิภาพในการจัดเก็บและการเข้าถึงข้อมูลด้วย

ในการจัดเก็บข้อมูลจะมีคำถามที่ว่าเราจะสามารถจัดเก็บข้อมูลอย่างไรที่ทำให้ประสิทธิภาพในการทำงานอยู่ในเกณฑ์ที่ดีได้ ? ก่อนที่จะตอบคำถามเราเหล่านี้เราจะต้องเข้าใจถึงโครงสร้างข้อมูลเสียก่อน รูปที่ 12-6 แสดงถึงโครงสร้างทางกายภาพของการจัดเก็บข้อมูลทั้งใน staging area, data warehouse repository และระบบ OLAP จากรูปเราควรจะสังเกตุความแตกต่างของข้อมูลในแต่ละที่ด้วย จากนั้นเราจะทำการพิจารณาถึงการออกแบบโครงสร้างการจัดเก็บข้อมูลใน physical storage เช่น files, blocks และ records



รูปที่ 12-6 โครงสร้างการจัดเก็บข้อมูลในคลังข้อมูล

การทำให้การจัดเก็บข้อมูลมีประสิทธิภาพสูงสุด

อย่างที่เราทราบกันดีว่า ข้อมูลที่ถูกเก็บอยู่ใน physical storage จะถูกเก็บอยู่ในรูปแบบของแฟ้มข้อมูล ถ้าเราต้องการเก็บข้อมูลเกี่ยวกับ customer dimension table และ salesperson dimension table เราจะมีวิธีการเก็บข้อมูลอยู่ 2 วิธีด้วยกันคือ

1. ทำการเก็บข้อมูลแต่ละตารางแยกกันในแต่ละไฟล์

2. ถ้าเรคคอร์ดของทั้งสองตารางมีการเรียกใช้งานพร้อมกันบ่อยครั้ง เราอาจทำการเก็บข้อมูลทั้งสองตารางไว้ในไฟล์เดียวกัน

วิธีในการเก็บข้อมูลลงใน physical storage นั้นค่อนข้างหลากหลาย เราควรจะต้องเลือกวิธีการที่ดีที่จะสามารถเพิ่มประสิทธิภาพในการจัดเก็บข้อมูล แต่อย่างไรก็ตาม การเก็บข้อมูลลงใน physical storage จะเกี่ยวข้องกับคุณสมบัติและฟังก์ชันต่าง ๆ ของ DBMS ที่ใช้ในการจัดเก็บข้อมูลด้วย

ดังนั้นเราจะต้องพิจาณาถึงเทคนิคที่สามารถทำงานสอดคล้องกับ DMBS ลองพิจารณาเทคนิคดังต่อไปนี้เพื่อเพิ่มประสิทธิภาพในการจัดเก็บข้อมูลลงใน physical storage

การกำหนดขนาดของ

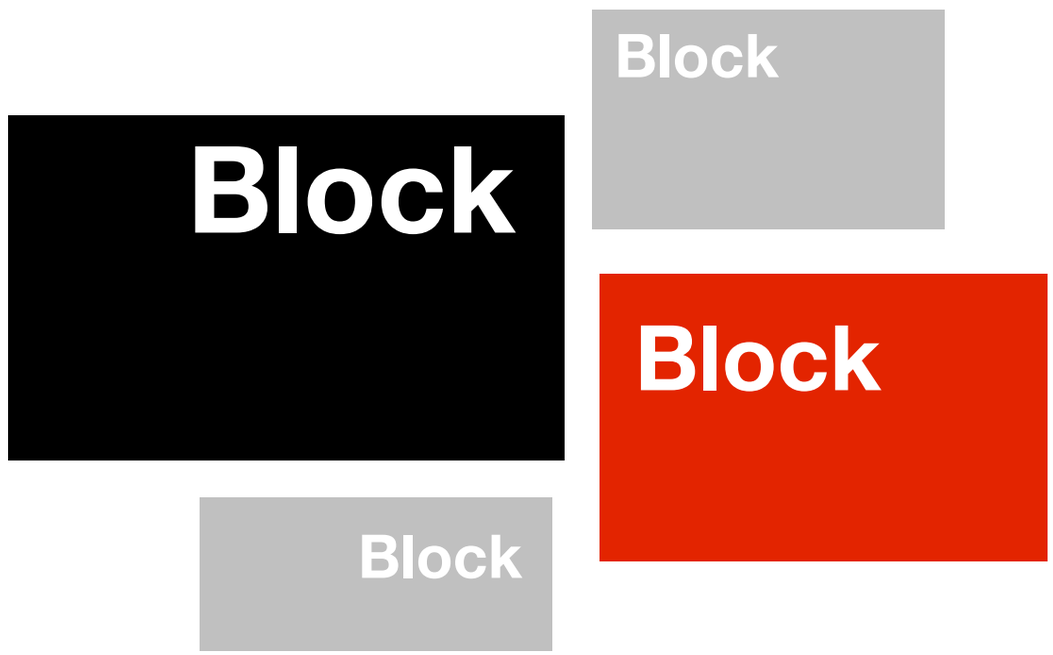
“Correct Block”

ใน physical storage เซตของเรคคอร์ดจะถูกเก็บอยู่ใน block หนึ่ง ๆ และเซตของ block จะถูกเก็บอยู่ในแฟ้มข้อมูลหนึ่งๆ ในการถ่ายโอนข้อมูลจากฐานข้อมูลไปยังหน่วยความจำจะทำการถ่ายโอนข้อมูลครั้งละ 1 block ซึ่งแต่ละ block จะประกอบด้วย block header (บอกถึงรายละเอียดต่างๆของ block) control information และเซตของเรคคอร์ดที่เก็บอยู่ใน block นั้นๆ โดยทั่วไปแล้วขนาดของ block ที่เก็บอยู่ในแต่ละ file จะมีขนาด 2KB และ 4KB ซึ่งเป็นขนาดมาตราฐานที่ใช้กันใน DBMS ถ้าผู้ใช้ต้องการเรียกดูข้อมูลใน block ที่ 10 ระบบปฏิบัติการจะทำการอ่าน block ทั้งหมดเพื่อที่จะได้ข้อมูลที่ต้องการ

ในการพิจารณาประสิทธิภาพของการจัดเก็บข้อมูล ถ้าเราลองเพิ่มขนาดของ block จะมีผลกระทบอะไรหรือไม่? เราจะสามารถเห็นประโยชน์โดยตรงคือ เราจะสามาถเก็บเรคคอร์ดในจำนวนที่มากขึ้นต่อหนึ่ง block ซึ่งจะทำให้สามารถลดจำนวน block ที่ต้องทำการ fetch ข้อมูลในแต่ละรอบการทำงาน อีกหนึ่งประโยชน์ที่จะได้รับคือเมื่อ block มีขนาดใหญ่ขึ้น นั่นคือจำนวน block จะลดลง ทำให้พื้นที่ที่ใช้ในการจัดเก็บ block header ลดลงด้วย

แต่อย่างไรก็ดี การเพิ่มขนาดของ block ก็มีข้อเสียที่อาจทำให้เกิดปัญหาเกี่ยวกับการจัดการหน่วยความจำได้ เนื่องจากระบบปฏิบัติการต้องทำการเก็บข้อมูลไว้ในหน่วยความจำเพิ่มขึ้นตามขนาดของ block ที่เพิ่มขึ้นด้วย แต่อย่างไรก็ดีคิวรีส่วนใหญ่ที่ดำเนินการกับคลังข้อมูลมักจะต้องยุ่งเกี่ยวกับข้อมูลเป็นจำนวนหลายแถวมากอยู่แล้ว จึงทำให้ไม่ได้รับผลกระทบจากปัญหาการจัดการหน่วยความจำเท่าที่ควร

“Correct Block”

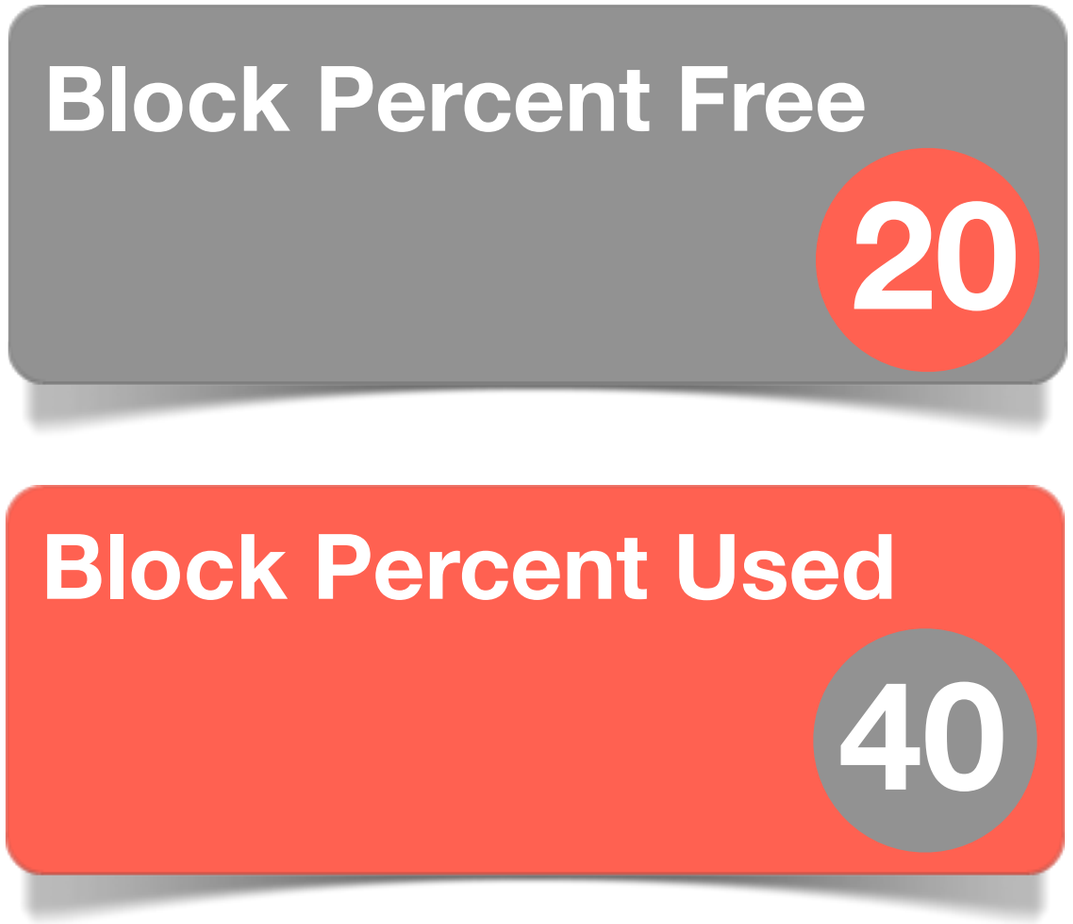


ปัญหาหนึ่งที่อาจเกิดขึ้นในการจัดเก็บข้อมูล คือ ตารางส่วนใหญ่ของคลังข้อมูลมักจะถูก denormalize เพื่อเพิ่มประสิทธิภาพในการเข้าถึงข้อมูล ซึ่งจะทำให้แต่ละเรคคอร์ดจะมีข้อมูลเป็นจำนวนมากหรือมีขนาดใหญ่มาก ในบางครั้งข้อมูลในเรคคอร์ดหนึ่ง ๆ อาจไม่สามารถเก็บอยู่ใน block เดียวได้ เราอาจต้องทำการแตก/แยกเรคคอร์ดไปเก็บไว้ในหลายๆ block โดยใช้ pointer ในการเชื่อมโยงข้อมูลระหว่าง block ซึ่งการใช้ pointer นั้นจะมีผลกระทบต่อประสิทธิภาพการทำงานอย่างหลีกเลี่ยงไม่ได้

เมื่อเราทราบถึงปัจจัยและปัญหาเกี่ยวกับขนาดของ block ที่ใช้ในการจัดเก็บข้อมูลแล้ว เราควรจะเลือกขนาดของ block ให้มีความเหมาะสม ถึงแม้ว่า block ที่มีขนาดใหญ่อาจทำให้ประสิทธิภาพในการทำงานดีขึ้นก็ตาม

การกำหนดพารามิเตอร์ต่าง ๆ ของแต่ละ block ข้อมูลให้มีความเหมาะสม

โดยส่วนใหญ่แล้วระบบจัดการฐานข้อมูลจะอนุญาตให้มีการกำหนดพารามิเตอร์เกี่ยวกับการใช้บล็อกข้อมูลให้มีค่าที่เหมาะสมเพื่อช่วยเพิ่มประสิทธิภาพในการทำงาน ดังนั้นเราจะต้องทำการหาค่าพารามิเตอร์ที่เหมาะสมเอง และวิธีการที่จะกำหนดค่าพารามิเตอร์เหล่านั้นจะขึ้นอยู่กับแต่ละซอร์ฟแวร์ที่ใช้ ซึ่งโดยทั่ว ๆ ไปแล้วการใช้งานเกี่ยวกับบล็อกจะมี 2 พารามิเตอร์ด้วยกัน ดังตัวอย่างดังต่อไปนี้



Block Percent Free 20

Block Percent Free — ในแต่ละบล็อกที่ถูกเก็บไว้ในระบบจัดการฐานข้อมูลจะมีพื้นที่ว่างซึ่งสงวนไว้สำหรับการเปลี่ยนแปลงหรืออัพเดทข้อมูลที่เกิดขึ้นจากตัวอย่างข้างต้น พื้นที่ว่างที่สงวนไว้จะเป็น 20 ซึ่งหมายความว่า 20% ของแต่ละบล็อกจะเป็นพื้นที่ที่สงวนไว้สำหรับการเปลี่ยนแปลงหรืออัพเดทข้อมูล

แต่อย่างไรก็ดี ข้อมูลสำหรับคลังข้อมูลนั้นแทบจะไม่มีการอัพเดทเท่าไรนัก เนื่องจากในการ initial load จะทำการเพิ่มข้อมูลทั้งหมดลงในคลังข้อมูล ส่วน incremental load ก็จะทำการเพิ่มข้อมูลเข้าสู่คลังข้อมูลด้วยเช่นกัน การอัพเดทข้อมูลจะเกิดขึ้นเมื่อข้อมูลใน dimension table มีการเปลี่ยนแปลงอย่างช้า ๆ (Slowly change) ดังนั้นในการกำหนดค่าพารามิเตอร์สำหรับพื้นที่ว่างที่สงวนไว้ ถ้าเรากำหนดไว้สูงก็จะทำให้เปลืองพื้นที่โดยไม่จำเป็น ดังนั้นเราควรจะกำหนดให้มีค่าน้อยที่สุดเท่าที่จะเป็นไปได้

Block Percent Used

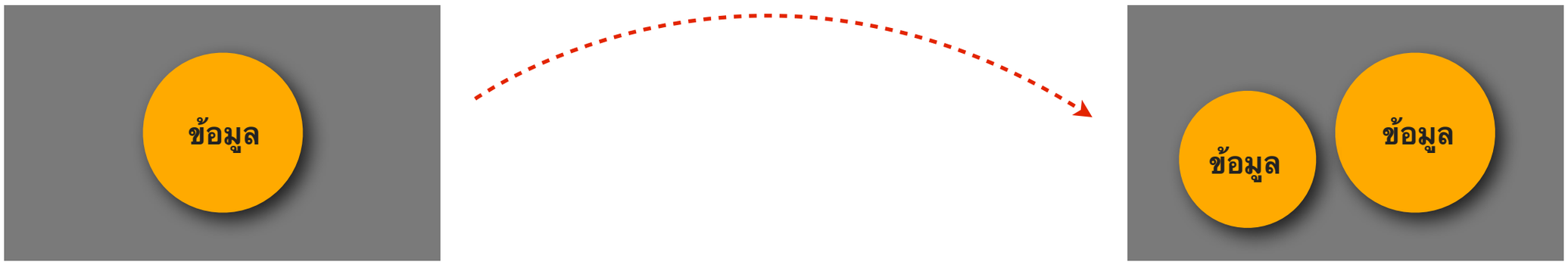
40

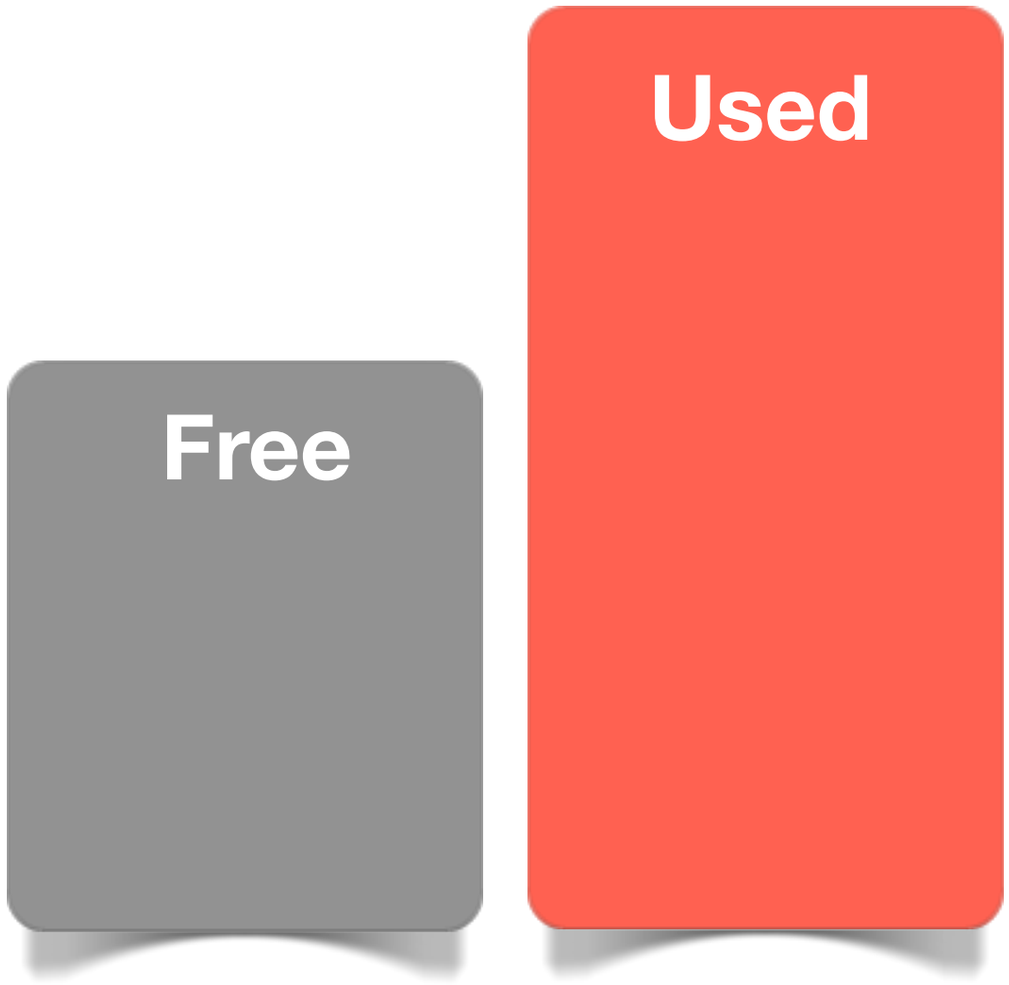
***Block Percent Used*—**พารามิเตอร์นี้จะเกี่ยวกับปริมาณข้อมูลที่ใช้ในแต่ละบล็อกที่จะส่งผลต่อการเพิ่มข้อมูลใหม่เข้ามาในบล็อกนั้น ๆ จากตัวอย่างข้างต้น ที่กำหนดไว้ที่ 40 นั้นหมายถึง เมื่อเรคคอร์ดถูกลบออกจากบล็อก ระบบจัดการฐานข้อมูลจะทำการตรวจสอบพื้นที่ที่เป็นพื้นที่ว่าง ว่ามีเกินกว่า 60% หรือไม่ ถ้าพื้นที่ว่างมีเกินกว่า 60% เราถึงจะสามารถใช้พื้นที่ว่างนั้นในการเก็บข้อมูลใหม่ที่มีการเพิ่มเข้ามาได้ ในการใช้งานคลังข้อมูล โดยส่วนใหญ่แล้วข้อมูลจะถูกเพิ่มเข้ามามากกว่าที่จะถูกลบออกไป

ดังนั้นเราควรจะต้องกำหนดให้ค่าของพารามิเตอร์นี้มีค่ามากที่สุดเท่าที่จะเป็นไปได้ นั่นหมายถึงเมื่อบล็อกนั้นมีพื้นที่ว่างน้อย ๆ เราก็จะสามารถเพิ่มข้อมูลเข้าไปในบล็อกนั้น ๆ ได้

การจัดการเกี่ยวกับการเคลื่อนย้ายข้อมูล

เมื่อเรคคอร์ดหนึ่ง ๆ ของบล็อกหนึ่ง ๆ มีการอัพเดทเกิดขึ้น แต่เผอิญว่าในบล็อกนั้นมีพื้นที่ไม่เพียงพอต่อข้อมูลที่เพิ่มขึ้นเมื่อทำการอัพเดทข้อมูล ระบบจัดการฐานข้อมูลส่วนใหญ่จะทำการเคลื่อนย้ายข้อมูลเรคคอร์ดที่มีการอัพเดททั้งหมดไปยังบล็อกอื่น แล้วทำการสร้างการเชื่อมโยงไปยังเรคคอร์ดที่ถูกเคลื่อนย้ายไป ซึ่งการเคลื่อนย้ายเรคคอร์ดจะส่งผลกระทบต่อประสิทธิภาพการทำงานเนื่องจากต้องทำการอ่านข้อมูลหลายบล็อกกว่าจะได้ข้อมูลที่ต้องการ แต่อย่างไรก็ดีปัญหานี้สามารถแก้ได้โดยการปรับค่าพารามิเตอร์





การจัดการการใช้งานบล็อกข้อมูล

ประสิทธิภาพของการเข้าถึงข้อมูลจาก physical storage จะต่ำลงเมื่อบล็อกของข้อมูลมีพื้นที่ว่างมาก เมื่อไหร่ก็ตามที่มีการทำคิวรีที่ต้องการเรียกดูข้อมูลทั้งหมดจากตารางใดตารางหนึ่ง ระบบจะต้องทำการอ่านข้อมูลจากหลายบล็อกมากขึ้นตามพื้นที่ว่างในแต่ละบล็อก ซึ่งเป็นเหตุให้ประสิทธิภาพการทำงานของคลังข้อมูลนั้นถดถอยลงไป ดังนั้นเพื่อให้ประสิทธิภาพการทำงานของการเข้าถึงข้อมูลอยู่ในเกณฑ์ดี เราจะต้องทำการปรับค่า block percent free ให้ต่ำลง และเพิ่มค่า block percent used ให้สูงขึ้น

การแก้ปัญหาการยืดขยายของข้อมูล

ในการเก็บข้อมูลลงแฟ้มข้อมูลที่อยู่ใน physical storage เราจะต้องทำการกำหนดขอบเขตของจำนวนข้อมูลที่สามารถเก็บได้ในแฟ้มข้อมูลหนึ่ง ๆ ในกรณีที่มีข้อมูลเต็มขอบเขตที่กำหนดไว้แต่เราต้องการเพิ่มเรคคอร์ดใหม่เข้าไปในแฟ้มข้อมูลหนึ่งเรคคอร์ด ระบบจัดการฐานข้อมูลจะหาขอบขอบเขตใหม่ที่มากกว่าเดิม **(เรียกว่า “Dynamic extension”)** และยอมให้ทำการเพิ่มเรคคอร์ดนั้นลงในแฟ้มข้อมูลแต่อย่างไรก็ตาม การหาขอบเขตใหม่ก็นำมาซึ่ง overhead ในการทำงาน

ดังนั้น เพื่อที่จะลดการทำงานของ dynamic extension เราควรกำหนดขอบเขตของจำนวนข้อมูลให้สามารถเก็บข้อมูลได้มาก ๆ ในตอนเริ่มต้น

การประยุกต์ใช้เทคนิคการแบ่งแฟ้มข้อมูลออกเป็นส่วน ๆ

ในการจัดเก็บข้อมูลลงใน physical storage เราสามารถการแบ่งแฟ้มข้อมูลออกเป็นส่วนๆ (File striping) จากนั้นเก็บแต่ละส่วนของแฟ้มข้อมูลไว้ใน physical devices ที่แยกจากกัน การทำ file striping จะช่วยให้เราสามารถเข้าถึงข้อมูลได้พร้อมๆกัน ซึ่งจะช่วยให้เราสามารถเพิ่มประสิทธิภาพในการเข้าถึงข้อมูลได้

การใช้เทคโนโลยี RAID ในการจัดเก็บข้อมูล

ในปัจจุบันคลังข้อมูลได้ประยุกต์ใช้ RAID เทคโนโลยี (Redundant array of inexpensive disks) ซึ่งประกอบไปด้วยดิสก์หลายตัวและมักจะใช้ในเซิร์ฟเวอร์ขนาดใหญ่เท่านั้น โดยเทคโนโลยีนี้จะสามารถทำงานได้ในขณะที่มีการกู้คืนข้อมูลที่เกิดความผิดพลาดจากดิสก์หนึ่ง ๆ ได้ เนื่องจาก RAID ทำการแบ่งข้อมูลออกเป็นส่วน ๆ จากนั้นทำการเขียนข้อมูลแต่ละส่วนไว้ในหลาย ๆ ดิสก์ ซึ่งจะทำให้ข้อมูลมีความซ้ำซ้อนกัน แต่เทคโนโลยีนี้ก็มีประโยชน์คือความคงทนต่อความผิดพลาด เนื่องจากมีความสามารถในการกู้คืนข้อมูลเมื่อมีความผิดพลาดในดิสก์หนึ่ง ๆ และสามารถสร้างข้อมูลที่มีความผิดพลาดขึ้นใหม่ได้ โดยที่ RAID จะมีคุณลักษณะดังนี้

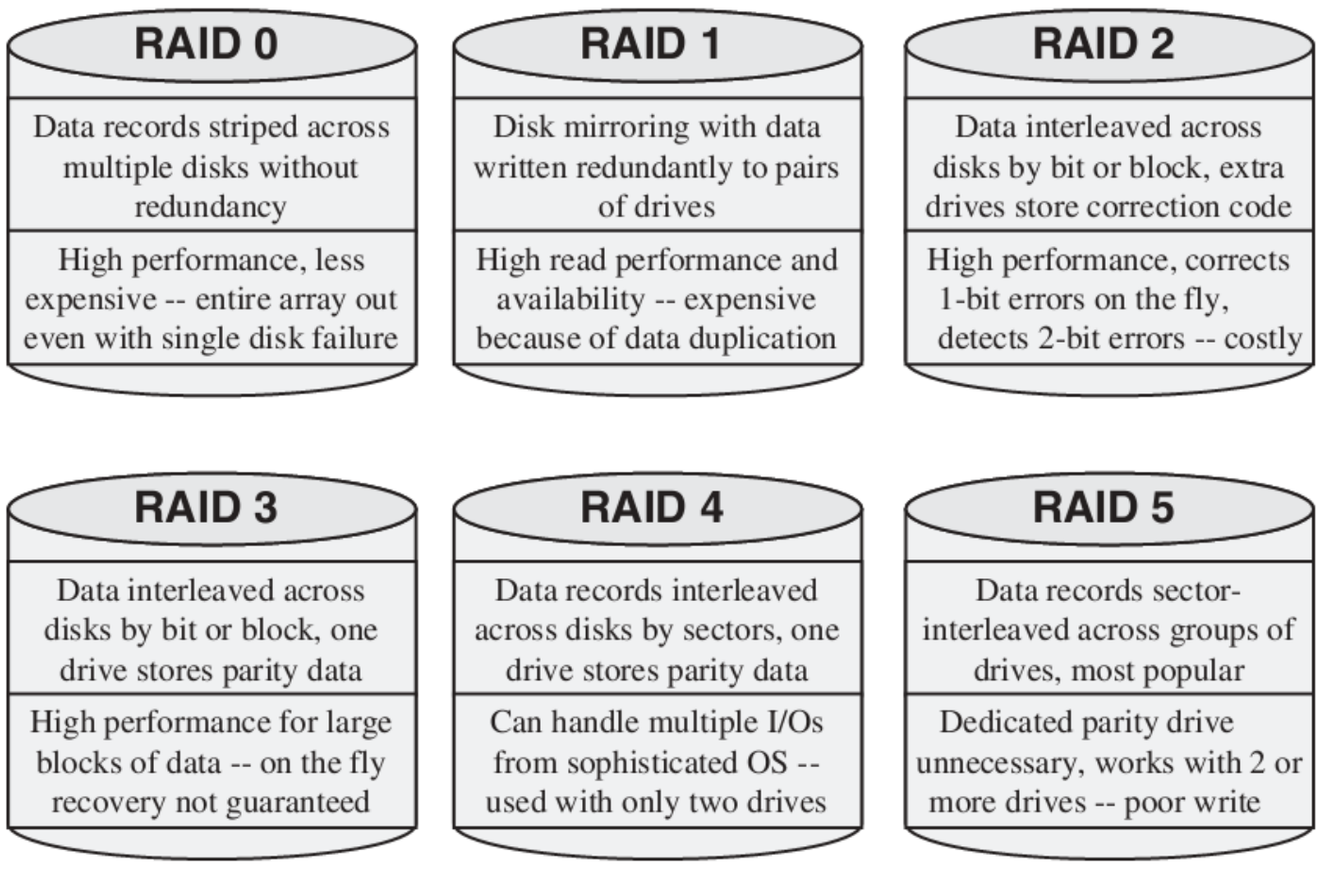
Disk mirroring —เขียนข้อมูลที่เหมือนกันลงในสองดิสก์ โดยที่ทั้งสองดิสก์จะถูกควบคุมโดย controller ตัวเดียวกัน

Disk deplexing —จะเหมือนกับ disk mirroring แต่จะแตกต่างที่ทั้งสองดิสก์จะถูกควบคุมโดย controller ที่ต่างกัน

Parity checking —จะทำการเพิ่ม parity bit (บิตภาวะคู่หรือคี่) ให้กับข้อมูลเพื่อใช้ในการตรวจสอบการส่งผ่านข้อมูลให้มีความถูกต้อง

Disk striping —จะทำการแพร่กระจายข้อมูลให้อยู่ในหลาย ๆ ดิสก์

RIAD จะถูกดำเนินการโดยการแบ่งออกเป็น 6 ระดับตั้งแต่ RAID-0 จนถึง RAID-5 ดังแสดงในรูปที่ 12-7 ซึ่งจะบอกถึงข้อดีและข้อเสียของแต่ละระดับด้วย



รูปที่ 12-7 เทคโนโลยี RAID

การประเมินพื้นที่ที่ใช้สำหรับจัดเก็บข้อมูล

ในการออกแบบหรือจัดการกับ physical storage เราจะต้องทำการประเมินขนาดของพื้นที่ที่จะใช้ในการจัดเก็บข้อมูล โดยที่เราจะต้องทราบถึงขนาดของพื้นที่ที่ใช้ในการจัดเก็บข้อมูลในขั้นตอนการถ่ายโอนข้อมูลครั้งแรก (initial load) และสำหรับการทำ incremental load เมื่อมีข้อมูลที่ต้องทำการเพิ่มให้กับคลังข้อมูล โดยที่การประเมินขนาดของพื้นที่จะสามารถดำเนินการได้ดังนี้

สำหรับแต่ละตารางในฐานข้อมูล เราจะต้องพิจารณาสิ่งเหล่านี้

- ทำการประเมินจำนวนแถวของข้อมูลที่จะทำการ initial load
- หาความยาวโดยเฉลี่ยของแต่ละแถว
- คาดคะเนจำนวนแถวที่จะเพิ่มขึ้นในแต่ละเดือน
- คาดคะเนขนาดเริ่มต้นของตารางโดยคิดเป็นเมกะไบต์ (MB)
- คำนวณขนาดของตารางในระยะ 6 เดือน และ 12 เดือน

สำหรับทุกตาราง เราจะต้องพิจารณาสิ่งเหล่านี้

- จำนวนดัชนีทั้งหมด เมื่อมีการสร้างดัชนีให้กับแต่ละตาราง
- จำนวนพื้นที่ที่ต้องการสำหรับการสร้างดัชนีในระยะเริ่มต้น 6 เดือน และ 12 เดือน

ทำการประเมิน

- พื้นที่ชั่วคราวที่ใช้สำหรับการเรียงและรวมข้อมูลเข้าด้วยกัน
- แฟ้มข้อมูลชั่วคราวที่ใช้ใน staging area ก่อนการถ่ายโอนข้อมูล
- แฟ้มข้อมูลที่ใช้ในการเก็บข้อมูลใน staging area

SECTION 6

การสร้างดัชนีในคลังข้อมูล

Index

การสร้างดัชนีในคลังข้อมูล

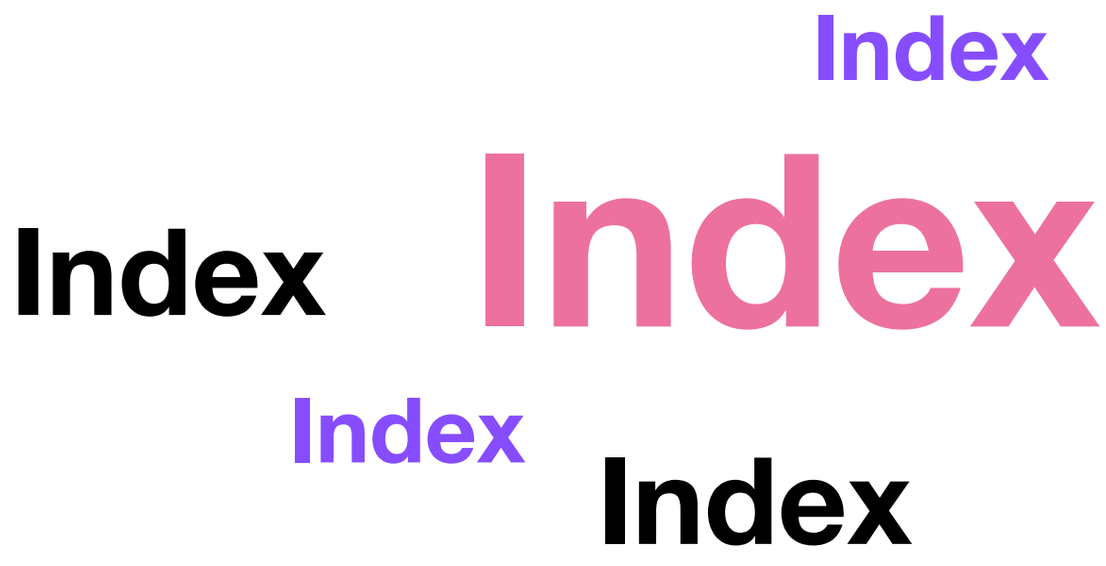
การสร้างดัชนี (index) ให้กับข้อมูลในตารางจะเป็นการเพิ่มประสิทธิภาพในการเข้าถึงข้อมูลที่ซึ่ง สามารถสร้างได้โดยการใช้ระบบจัดการฐานข้อมูลด้วยเช่นกัน โดยจะใช้เทคนิคต่างๆ มากมาย เช่น B-Tree indexes ที่ช่วยเพิ่มประสิทธิภาพในการค้นคืนข้อมูล Bitmapped index หรือการสร้างดัชนีให้กับตารางที่ถูกแบ่งออกเป็นส่วน ๆ (indexes on partition tables) และอื่น ๆ

ภาพรวมกว้าง ๆ ของ การสร้างดัชนี

ข้อมูลในคลังข้อมูลจะเป็นแบบ read-only ซึ่งจากคุณสมบัตินี้จะทำให้ผู้ใช้ไม่สามารถทำการอัพเดทหรือลบข้อมูลได้ และจะไม่สามารถเพิ่มข้อมูลให้กับคลังข้อมูลหลังจากการถ่ายโอนข้อมูลได้

ภาพรวมกว้าง ๆ ของ การสร้างดัชนี

จากคุณสมบัติดังกล่าวจะทำให้เราสามารถสร้างดัชนีให้กับแต่ละตารางได้เมื่อมีการถ่ายโอนข้อมูลจาก staging area เข้าสู่คลังข้อมูล โดยที่การสร้างดัชนีโดยส่วนใหญ่จะสร้างกับ dimension table และแต่ละครั้งในการสร้างดัชนีเราจะต้องพิจารณาถึงจำนวนดัชนีที่เราจะทำการสร้างในแต่ละตาราง โดยที่เมื่อขนาดของตารางเพิ่มขึ้น จำนวนดัชนีก็จะเพิ่มขึ้นด้วย และจำนวนพื้นที่ที่ใช้ก็จะเพิ่มขึ้นด้วยเช่นกัน ดังนั้นในการสร้างดัชนีให้กับตารางข้อมูลเราจะต้องพยายามทำให้ปัจจัยต่าง ๆ นั้นสมดุลและกำหนดจำนวนดัชนีต่อตารางหนึ่ง ๆ ด้วย



การสร้างดัชนี

การถ่ายโอนข้อมูล

เมื่อเรามีดัชนีเป็นจำนวนมากจะทำให้การถ่ายโอนข้อมูลเข้าสู่คลังข้อมูลช้าลง เนื่องจากเมื่อมีการเพิ่มเรคคอร์ดเข้าไปในแต่ละตารางเราจะต้องทำการสร้างดัชนีที่เกี่ยวข้องกับเรคคอร์ดนั้น ๆ ด้วย เนื่องจากปริมาณที่มากของดัชนีจะทำให้การทำ initial load นั้นค่อนข้างมีปัญหา เนื่องจากข้อมูลที่ต้องทำการ initial load นั้นมีปริมาณมาก เมื่อเราทำการเพิ่มข้อมูลแต่ละเรคคอร์ดเข้าสู่ตาราง เราจะต้องสร้างดัชนีให้กับเรคคอร์ดนั้น ๆ

แต่อย่างไรก็ดีเราสามารถหลีกเลี่ยงปัญหานี้ได้โดยทำการระงับการสร้างดัชนีระหว่างการโหลดข้อมูล โดยที่หลังจากการโหลดข้อมูลแล้วเราค่อยสร้างดัชนีให้กับข้อมูลในภายหลัง โดยการสร้างดัชนีหลังจากโหลดข้อมูลทั้งหมดแล้วจะใช้เวลาค่อนข้างมาก แต่ก็ยังไม่มากไปกว่าการสร้างดัชนีระหว่างการโหลดข้อมูล

การสร้างดัชนีสำหรับตารางที่มีข้อมูลเป็นจำนวนมาก

เมื่อตารางมีจำนวนเรคคอร์ดหลายล้านแถวซึ่งไม่สนับสนุนการมีหลายดัชนี ถ้าเราต้องการที่จะสร้างดัชนีขึ้นหลายดัชนี เราจะต้องทำการแยกตารางออกเป็นส่วน ๆ ก่อนแล้วจึงค่อยสร้างดัชนีให้กับตารางย่อยนั้น ๆ

การอ่านข้อมูลดัชนี

ในการค้นคืนข้อมูลจะทำการอ่านดัชนีเป็นอันดับแรก จากนั้นค่อยทำการอ่านข้อมูลที่สอดคล้องกับดัชนีที่เราทำการอ่านในภายหลัง โดยที่ระบบจัดการฐานข้อมูลจะทำการเลือกดัชนีที่ดีที่สุดจากดัชนีทั้งหมดก่อน ลองพิจารณาตัวอย่างดังต่อไปนี้ที่

ระบบจัดการฐานข้อมูลทำการสร้างดัชนีให้กับ 4 คอลัมน์ในตารางหนึ่ง ๆ และผู้ใช้ต้องการที่จะเรียกดูข้อมูลจาก 4 คอลัมน์นั้น และอีกหนึ่งคอลัมน์ในตารางเดียวกันที่ไม่ได้มีการสร้างดัชนี เราจะสามารถค้นคืนข้อมูลได้อย่างไร ? คำตอบคือ ระบบจัดการฐานข้อมูลจะใช้ดัชนีเพื่อค้นคืนเรคคอร์ดที่ต้องการ ซึ่งจะได้ข้อมูล 4 คอลัมน์ที่มีการสร้างดัชนี จากนั้นระบบจัดการฐานข้อมูลจะทำการค้นคืนข้อมูลอีกหนึ่งคอลัมน์ที่ไม่มีการสร้างดัชนี ในกรณีนี้เราควรจะสร้างเพิ่มคอลัมน์ที่ยังไม่มีดัชนีเป็นตัวชี้เข้ากับดัชนีของ 4 คอลัมน์ที่มีการสร้างดัชนีไว้แล้ว

จากการสร้างดัชนีดังกล่าวจะทำให้ระบบจัดการฐานข้อมูลทำการอ่านดัชนีและอ่านข้อมูลจากดัชนีที่เกี่ยวข้องกันโดยไม่ต้องทำการอ่านข้อมูลที่ไม่จำเป็น

การเลือกคอลัมน์สำหรับการสร้างดัชนี

ในการสร้างดัชนีให้กับข้อมูล เราจะต้องพบเจอกับคำถามต่างๆ มากมาย เช่น เราจะทำการเลือกคอลัมน์ในตารางที่ควรจะทำการสร้างดัชนีได้อย่างไร? คอลัมน์ใดที่จะให้ประสิทธิภาพสูงที่สุดเมื่อทำการสร้างดัชนีให้กับคอลัมน์นั้นๆ

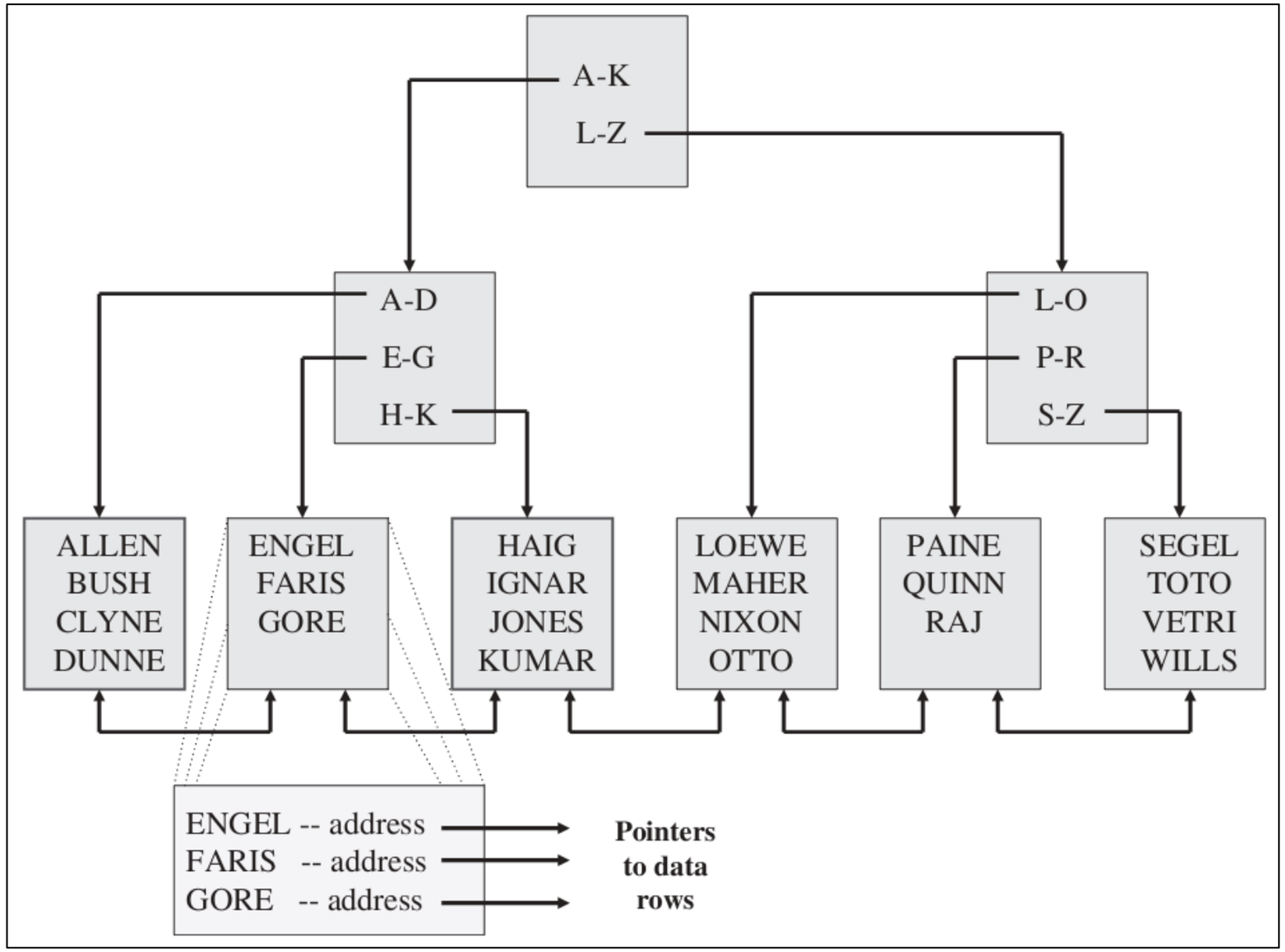
ในการสร้างดัชนีให้กับแต่ละคอลัมน์เราต้องต้องพิจารณาคิวรีที่ใช้ในการเข้าถึงข้อมูลจากคลังข้อมูล จากนั้นทำการตรวจสอบดูว่าคอลัมน์ใดเป็นคอลัมน์ที่ถูกใช้ค้นคืนข้อมูลบ่อยครั้ง ถ้าคิวรีเป็นจำนวนมากเรียกดูข้อมูลไลน์การผลิตสินค้า เราจะทำการเพิ่มไลน์การผลิตสินค้าไว้ในลิสต์ของคอลัมน์ที่จะทำการสร้างดัชนี จากนั้นค่อยมาพิจารณาคอลัมน์ทั้งหมดอีกครั้งหนึ่ง

ในการสร้างดัชนีนั้นผู้ดูแลระบบอาจสับสนว่าจะเริ่มทำการสร้างดัชนีได้อย่างไร หรือแต่ละตารางควรจะมีจำนวนดัชนีเป็นเท่าไหร่ และคอลัมน์ใดสมควรที่จะทำการสร้างดัชนีสำหรับการดำเนินการในครั้งแรก เนื่องจากในครั้งแรกไม่ทราบถึงการใช้งานของผู้ใช้ว่าต้องการเรียกดูข้อมูลอะไรบ้าง และความถี่ในการเรียกดูแต่ละข้อมูล เมื่อเราไม่ทราบข้อมูลเหล่านี้ เราควรจะต้องรอให้ผู้ใช้เริ่มใช้งานคลังข้อมูลไปก่อน จากนั้นค่อยทำการสร้างดัชนีจากข้อมูลการใช้งานของผู้ใช้ โดยที่ในตอนแรกเราจะทำการสร้างดัชนีให้กับคีย์หลักและคีย์รอง ของแต่ละตารางก่อน

การสร้างดัชนีโดยใช้
B-tree

โดยส่วนใหญ่ของระบบจัดการฐานข้อมูลจะใช้ B-tree (Balanced binary tree) ในการสร้างดัชนีให้กับข้อมูลผ่านทางการทำ code statement โดยใช้ data definition language (DDL) ซึ่งโดยปกติแล้วระบบจัดการฐานข้อมูลจะทำการสร้างดัชนีให้กับคีย์หลักโดยอัตโนมัติ โดยที่ B-tree นั้นเป็นเทคนิคที่ค่อนข้างจะดีกว่าเทคนิคอื่น ๆ เนื่องจากให้ความเร็วในการค้นคืนข้อมูล ให้ความสะดวกสะบายในการดูแลรักษาข้อมูล และความง่ายในการทำงาน

รูปที่ 12-8 จะแสดงถึงตัวอย่างของการสร้างดัชนีโดยใช้ B-tree ซึ่งเป็นโครงสร้างข้อมูลต้นไม้ที่ใช้สำหรับการสร้างดัชนีของชื่อคนที่เป็นข้อมูลทั้งหมดที่เก็บไว้ในคลังข้อมูล โดยที่ใน 2 ระดับแรกจะเป็นดัชนีที่จะชี้ไปยังข้อมูลในระดับถัดไปและระดับล่างสุดจะชี้ไปยังแถวของข้อมูลในตาราง

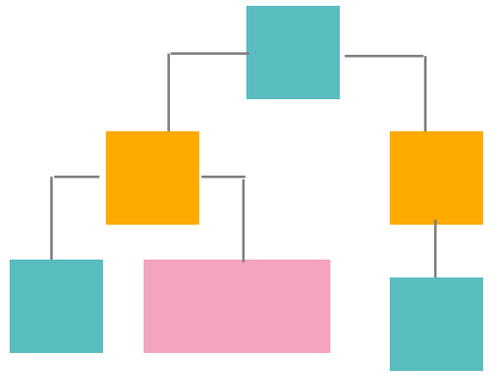


รูปที่ 12-8 ตัวอย่างการสร้างดัชนีโดยใช้ B-tree

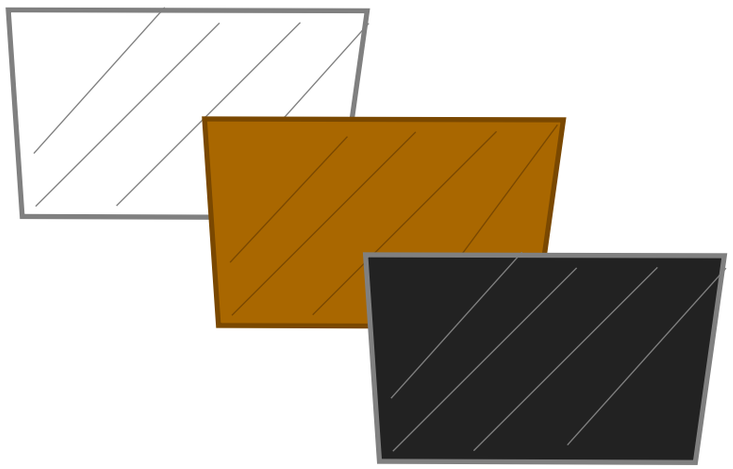
ถ้าในคอลัมน์หนึ่ง ๆ มีค่าที่ไม่ซ้ำกันเป็นจำนวนมากจะทำให้สมรรถนะในการสร้างดัชนีให้กับคอลัมน์นั้นสูง ตัวอย่างเช่น ใน dimension table ของพื้นที่หรือสถานที่ตั้งของร้านค้า ส่วนคอลัมน์ที่เก็บข้อมูลเกี่ยวกับเมืองจะมีหลายค่าที่ไม่ซ้ำกัน ซึ่งจะทำให้สมรรถนะในการสร้างดัชนีนั้นสูง โดยที่การสร้างดัชนีโดยใช้ B-tree นั้นจะเหมาะกับคอลัมน์ที่มีสมรรนะสูง ๆ เนื่องจากใน leaf nodes จะมีค่าที่ไม่ซ้ำกันเป็นจำนวนมากซึ่งนำไปสู่การอ้างถึงแถวของข้อมูลที่ไม่ซ้ำกัน จากคุณลักษณะของ B-tree ถ้าคอลัมน์มีสมรรถนะไม่สูงเราจะสมควรใช้ B-tree ในการสร้างดัชนีหรือไม่ ? ตัวอย่างเช่น คอลัมน์ที่เก็บชื่อของพนักงานซึ่งจะมีสมรรถนะไม่สูง เนื่องจากอาจมีชื่อพนักงานซ้ำกัน แต่อย่างไรก็ดี เราสามารถเพิ่มสมรรถนะของคอลัมน์นี้ได้ โดยการนำนามสกุลมาต่อกับชื่อ เพื่อให้ข้อมูลมีการซ้ำกันน้อยลง จากนั้นทำการสร้างดัชนีโดยใช้ B-tree กับคอลัมน์ที่ประกอบไปด้วยชื่อและนามสกุล

เมื่อไรก็ตามที่มีการสร้างดัชนีให้กับคอลัมน์ที่มาจากการรวมข้อมูลหลาย ๆ คอลัมน์เข้าด้วยกัน จะทำให้มีการใช้พื้นที่เพื่อเก็บข้อมูลค่อนข้างมาก (ต้องเก็บข้อมูลเยอะ จากเดิมเก็บข้อมูลชื่ออย่างเดียว ต้องเปลี่ยนเป็นเก็บชื่อและนามสกุลเข้าด้วยกัน) เนื่องจากคลังข้อมูลนั้นมีข้อมูลค่อนข้างมาก และคอลัมน์ที่เก็บอยู่ในคลังข้อมูลไม่ได้เป็นคอลัมน์ที่มีสมรรถนะสูงทั้งหมดอาจทำให้ index files นั้นมีขนาดใหญ่และอาจทำให้เกิดปัญหาได้ โดยที่ถ้าเราลองตรวจสอบคอลัมน์ที่เก็บอยู่ใน dimension table เราจะเห็นว่ามีคอลัมน์เป็นจำนวนมากที่มีสมรรถนะต่ำ

การสร้างดัชนีด้วย B-tree นั้นจะทำงานไม่ค่อยดีเมื่อข้อมูลหรือคอลัมน์มีสมรรถนะต่ำ ซึ่งนี่จะเป็นเหตุให้เราควรจะต้องพิจารณาถึงทางเลือกหรือเทคนิคอื่น ๆ ต่อไป



Bitmapped Index



การสร้างดัชนีด้วย Bitmap นั้นจะเหมาะกับข้อมูลหรือคอลัมน์ที่มีสมรรถนะต่ำ ซึ่ง bitmap จะเป็นกลุ่มของบิตที่เรียงลำดับกัน โดยที่ 1 บิตจะหมายถึงค่าหนึ่งค่าที่เกิดขึ้นในคอลัมน์นั้น ๆ

ตัวอย่างเช่น คอลัมน์ที่เก็บข้อมูลของสีของสินค้าในตารางสินค้ามีอยู่ 3 สีด้วยกันคือ สีขาว สีน้ำตาลอ่อน และสีดำ

ดังนั้น bitmap สำหรับแต่ละเรคคอร์ดจะประกอบไปด้วย 3 บิต ถ้าเรากำหนดให้บิตแรกหมายถึงสีขาว บิตที่สองหมายถึงสีน้ำตาลอ่อน และบิตที่สามหมายถึงสีดำ ซึ่งถ้าสินค้ามีสีเป็นสีใดใน 3 สี ค่าบิตของสีนั้นจะเป็น 1 และบิตที่เหลือจะเป็น 0 เช่น ถ้าสีของสินค้า A เป็นสีขาว ฉะนั้นค่าในคอลัมน์สีของสินค้า A คือ 100 ซึ่งหมายถึง สินค้า A มีสีขาว ลองพิจารณาตัวอย่างในรูปที่ 12-9 ซึ่งแสดงถึงข้อมูลการขายสินค้าและ bitmap ที่สอดคล้องกับแต่ละคอลัมน์ในตารางการขายข้อมูลสินค้า ซึ่งในแต่ละแถวจะมีข้อมูลที่เป็น bitmap ในคอลัมน์ต่าง ๆ เช่น รายการสินค้า สี และภูมิภาค เป็นต้น

Extract of Sales Data

Address or Rowid	Date	Product	Color	Region	Sale($)
00001BFE.0012.0111	15-Nov-00	Dishwasher	White	East	300
00001BFE.0013.0114	15-Nov-00	Dryer	Almond	West	450
00001BFF.0012.0115	16-Nov-00	Dishwasher	Almond	West	350
00001BFF.0012.0138	16-Nov-00	Washer	Black	North	550
00001BFF.0012.0145	17-Nov-00	Washer	White	South	500
00001BFF.0012.0157	17-Nov-00	Dryer	White	East	400
00001BFF.0014.0165	17-Nov-00	Washer	Almond	South	575

Bitmapped Index for Product Column

Ordered bits: Washer, Dryer, Dishwasher

Address or Rowid	Bitmap
00001BFE.0012.0111	001
00001BFE.0013.0114	010
00001BFF.0012.0115	001
00001BFF.0012.0138	100
00001BFF.0012.0145	100
00001BFF.0012.0157	010
00001BFF.0014.0165	100

Bitmapped Index for Color Column

Ordered bits: White, Almond, Black

Address or Rowid	Bitmap
00001BFE.0012.0111	100
00001BFE.0013.0114	010
00001BFF.0012.0115	010
00001BFF.0012.0138	001
00001BFF.0012.0145	100
00001BFF.0012.0157	100
00001BFF.0014.0165	010

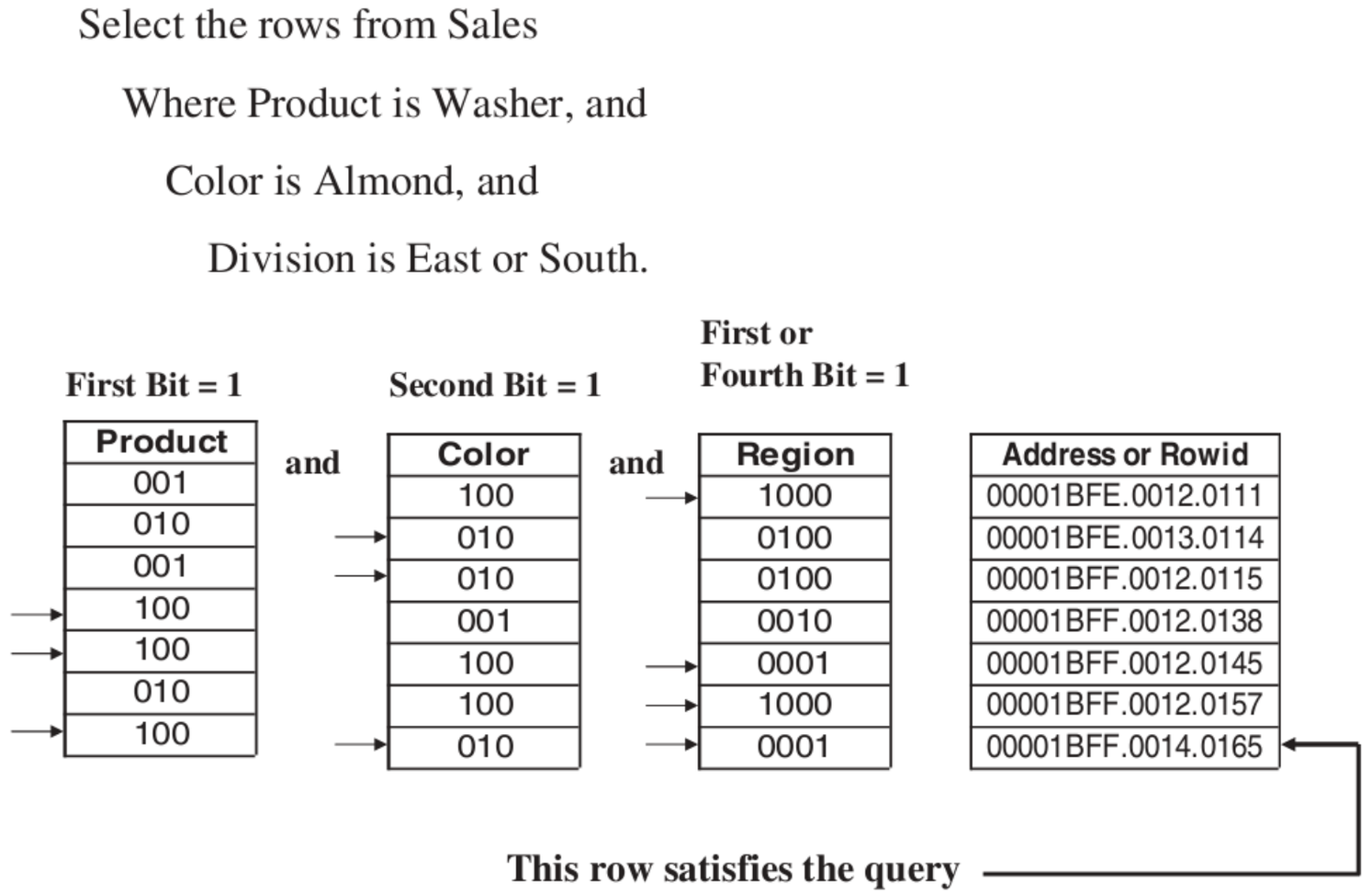
Bitmapped Index for Region Column

Ordered bits: East, West, North, South

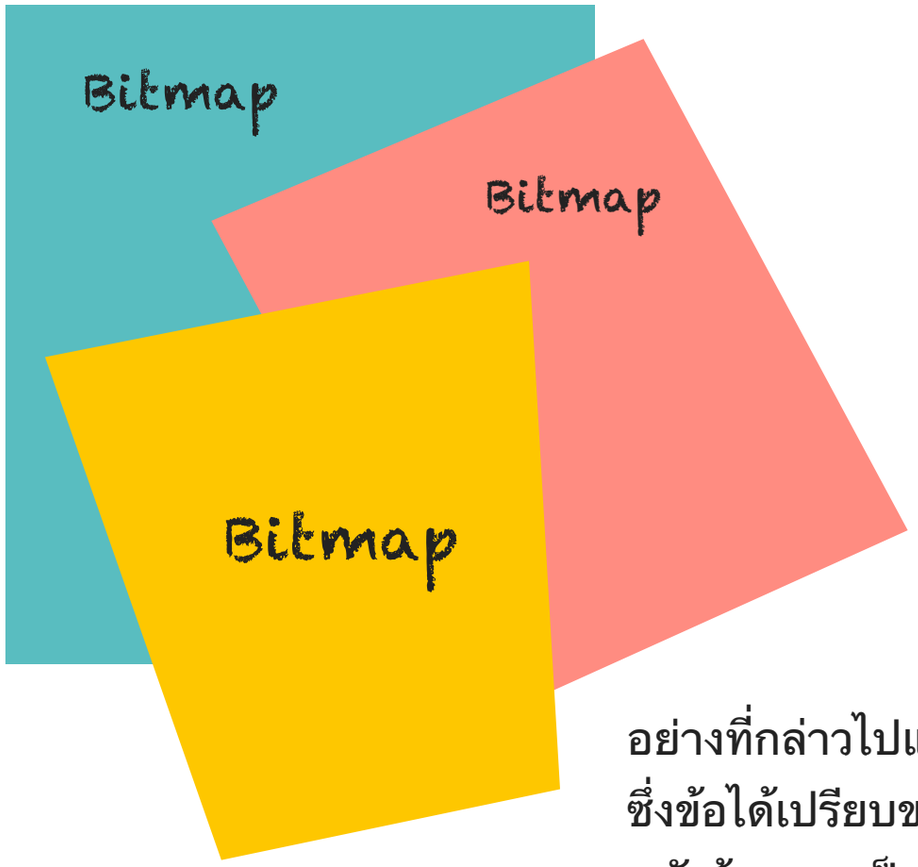
Address or Rowid	Bitmap
00001BFE.0012.0111	1000
00001BFE.0013.0114	0100
00001BFF.0012.0115	0100
00001BFF.0012.0138	0010
00001BFF.0012.0145	0001
00001BFF.0012.0157	1000
00001BFF.0014.0165	0001

รูปที่ 12-9 ตัวอย่างการสร้างดัชนีด้วย bit-mapped

ในการที่จะค้นคืนแถวที่เราต้องการ
เราจะต้องพิจารณาคิวรีสำหรับการเปรียบเทียบกับข้อมูล
ในตาราง เช่น จากข้อมูลในรูปที่ 12-9 ถ้าผู้ใช้ต้องการ
ดูข้อมูลการขายสินค้าชนิด “Washer” สี “Almond” และ
ขายที่ภูมิภาค “East or South” เราจะสามารถใช้
Boolean logic ในการค้นหาผลลัพธ์ได้ดังแสดง
ในรูปที่ 12-10



รูปที่ 12-10 การค้นคืนผลลัพธ์จาก bit-mapped index



อย่างที่กล่าวไปแล้วข้างต้นว่า bitmap จะเหมาะกับการสร้างดัชนีให้กับคอลัมน์ที่มีสมรรถนะต่ำ ซึ่งข้อได้เปรียบของวิธีการนี้จะอยู่ที่การใช้พื้นที่ที่น้อยกว่า B-tree โดยส่วนใหญ่แล้วข้อมูลในคลังข้อมูลจะเป็นแบบสมรรถนะต่ำ จึงเป็นเหตุให้การสร้างดัชนีให้กับข้อมูลนั้นค่อนข้างจะเหมาะสมกับการจัดเก็บข้อมูลในคลังข้อมูล แต่อย่างไรก็ดีการใช้ bitmap ในการสร้างดัชนีก็มีข้อเสีย เมื่อมีค่าใหม่เกิดขึ้นในคอลัมน์ เราต้องทำการสร้าง bitmap ใหม่ทั้งหมด ซึ่งเราจะต้องทำการอ่านข้อมูลทั้งหมดแล้วทำการอัพเดทข้อมูลเหล่านั้นทั้งหมด ในขณะที่ B-tree นั้นไม่ต้องทำการอ่านข้อมูลทั้งหมดเมื่อมีค่าใหม่เกิดขึ้น

การสร้างดัชนีใน Fact Table

เราทราบหรือไม่ว่าภายใน fact table ประกอบไปด้วยอะไรบ้าง ? และแต่ละคอลัมน์มีลักษณะเป็นอย่างไร? ลองพิจารณา star schema หนึ่ง ๆ ที่คีย์หลักของ fact table จะประกอบไปด้วยการรวมกันของคีย์หลักจาก dimension table เช่น ถ้าเรามี dimension table เป็นร้านค้า สินค้า เวลา และโปรโมชั่น คีย์หลักของ fact table จะประกอบไปด้วยคีย์หลักจากทั้ง 4 dimension เรียงต่อกัน ส่วนคอลัมน์อื่น ๆ จะเป็นตัวชี้วัดเช่น sale units, sale dollars, cost dollars และอื่น ๆ ดังนั้นเราสามารถสรุปได้ว่าใน fact table จะมีคอลัมน์อยู่ด้วยกัน 2 ประเภทคือ คีย์หลัก และตัวชี้วัดต่างๆ

ในการสร้างดัชนีให้กับคอลัมน์ต่าง ๆ ใน fact table จะมีเคล็ดลับดังต่อไปนี้

- ถ้าระบบจัดการฐานข้อมูลยังไม่ได้มีการสร้างดัชนีให้กับคีย์หลัก เราควรทำการสร้างดัชนีโดยการใช้ B-tree
- ควรออกแบบลำดับของการเรียงต่อกันของคีย์หลักใน fact table อย่างระมัดระวัง โดยที่พยามที่จะเลือกคีย์หลักของ dimension table ที่มีการอ้างถึงบ่อยอยู่ลำดับแรก ๆ เพื่อความสะดวกในการเข้าถึงข้อมูล
- ตรวจสอบแต่ละองค์ประกอบของคีย์ที่เรียงต่อกัน แล้วทำการสร้างดัชนีให้กับการจัดคีย์ตามหมวดหมู่โดยตั้งอยู่บนพื้นฐานของคิวรีที่ต้องการ
- ถ้า DBMS สามารถทำการรวมดัชนีของคีย์หลักจาก dimension เพื่อที่เรียกใช้ข้อมูลใน fact table เราอาจทำการสร้างดัชนีให้กับแต่ละองค์ประกอบของคีย์หลักใน fact table ได้
- อย่ามองข้ามการสร้างดัชนีให้กับคอลัมน์ที่เป็นตัวชี้วัด เช่น ถ้ามีคิวรีเป็นจำนวนมากร้องขอข้อมูลเกี่ยวกับ sale dollars คอลัมน์ “sale dollars” ควรจะเป็นคอลัมน์ที่น่าจะมีการสร้างดัชนีด้วย
- การสร้างดัชนีให้กับ fact table จะไม่ใช้ bitmap เนื่องจากข้อมูลใน fact table โดยส่วนใหญ่จะเป็นข้อมูลที่ไม่ซ้ำกัน ซึ่งจะทำให้คอลัมน์ต่าง ๆ มีสมรรถนะสูง ซึ่งไม่เหมาะกับ bitmap

การสร้างดัชนีใน Dimension Tables

คอลัมน์ใน dimension table นั้นจะใช้ในคิวรีต่าง ๆ ซึ่งคิวรีอาจเป็น “การขายสินค้าชนิด A ในเดือนมีนาคม ที่สาขาทางเหนือ เป็นจำนวนเท่าไหร่” ซึ่งจะทำการพิจารณาคอลัมน์สินค้า เดือน และภูมิภาค จาก 3 dimension ที่แตกต่างกันอาจเป็นคอลัมน์ที่ต้องทำการสร้างดัชนี ซึ่งเราจะต้องทำการตรวจสอบคอลัมน์ในแต่ละ dimension table อย่างละเอียดและทำการวางแผนในการสร้างดัชนีต่อไป ซึ่งการสร้างดัชนีให้กับคอลัมน์ใน dimension table จะช่วยเพิ่มประสิทธิภาพในการเข้าถึงข้อมูลได้

ในการสร้างดัชนีให้กับคอลัมน์ต่าง ๆ ใน dimension table จะมีเคล็ดลับดังต่อไปนี้

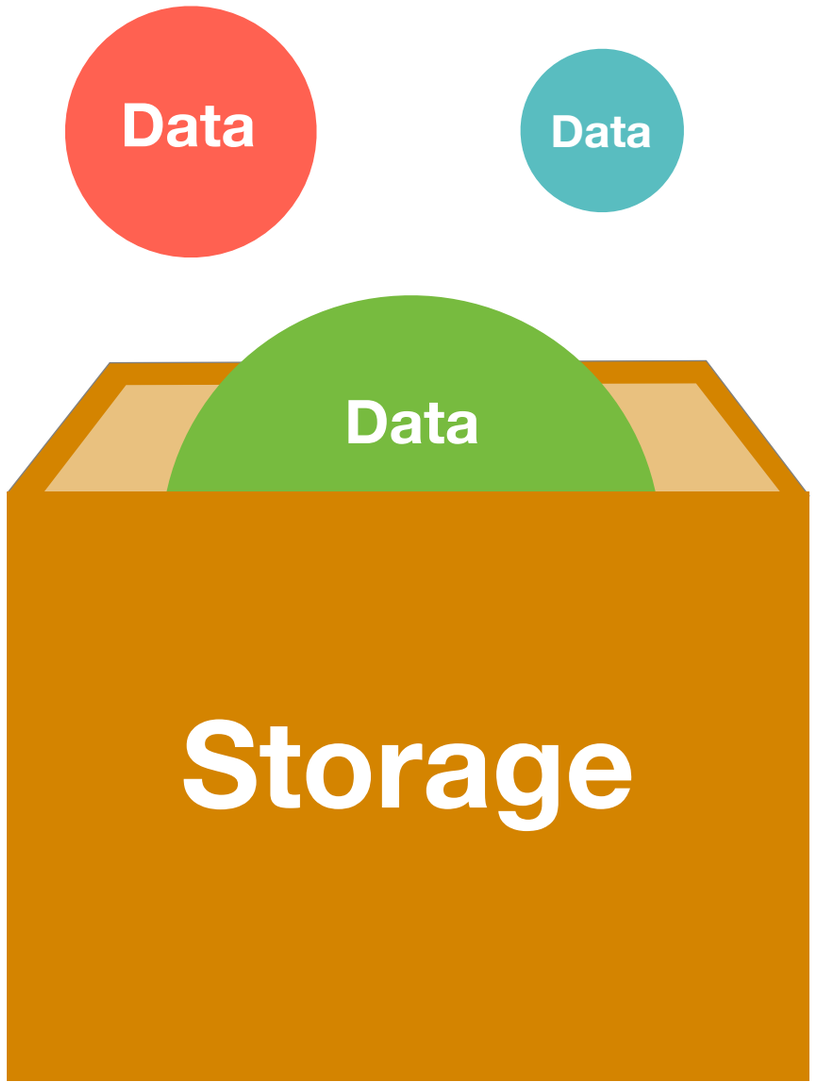
- สร้างดัชนีให้กับคีย์หลักที่เป็นคีย์เดียวโดยใช้ B-tree
- พิจารณาคอลัมน์ที่มักจะถูกใช้ในการให้ผลลัพธ์จากคิวรีโดยกำหนดให้คอลัมน์เหล่านั้นมีสิทธิที่จะทำการสร้างดัชนีโดยใช้ bitmap
- มองหาคอลัมน์ที่มักจะถูกใช้งานร่วมกันซึ่งคอลัมน์เหล่านั้นจะถูกเก็บไว้ในตารางขนาดใหญ่ โดยที่เราควรจะกำหนดวิธีในการจัดเรียงคอลัมน์นั้น ๆ และอาจทำการรวมคอลัมน์เหล่านั้นเข้าด้วยกันโดยทำการสร้างดัชนีให้กับคอลัมน์ที่ถูกรวมด้วย

SECTION 7

เทคนิคการเพิ่มประสิทธิภาพในการทำงานอื่น ๆ

เทคนิคการเพิ่มประสิทธิภาพในการทำงานอื่น ๆ

นอกเหนือจากการเพิ่มประสิทธิภาพในการทำคิวรีโดยการสร้างดัชนีแล้ว ยังมีวิธีอื่นที่จะเพิ่มประสิทธิภาพ ตัวอย่างเช่น การบีบอัดข้อมูลเมื่อทำการเขียนข้อมูลลงใน storage ซึ่งจะทำให้สามารถโหลดข้อมูลเข้าสู่บล็อกหนึ่ง ๆ ได้มากขึ้น และเป็นเหตุให้สามารถเข้าถึงข้อมูลได้มากขึ้นในการอ่านข้อมูลแต่ละครั้งอีกด้วย อีกวิธีหนึ่งที่ช่วยเพิ่มประสิทธิภาพ คือ การรวมตารางเข้าด้วยกัน ซึ่งจะทำให้สามารถเข้าถึงข้อมูลได้มากขึ้นในการอ่านข้อมูลแต่ละครั้ง และถ้าเรากำจัดข้อมูลที่ไม่ต้องการและไม่จำเป็นออกจากคลังข้อมูลอย่างสม่ำเสมอก็จะช่วยเพิ่มประสิทธิภาพในการทำงานได้ นอกจากนี้ยังมีวิธีอื่น ๆ ที่จะช่วยเพิ่มประสิทธิภาพในการทำงาน ซึ่งวิธีดังต่อไปนี้จะทำงานบนระบบจัดการฐานข้อมูลได้อีกด้วย



การแบ่งข้อมูลออกเป็นส่วน ๆ

โดยปกติแล้วคลังข้อมูลจะมีตารางที่มีข้อมูลเป็นจำนวนมาก เช่น fact table ที่อาจประกอบไปด้วยข้อมูลมากกว่า 1 ล้านแถว เมื่อตารางมีขนาดใหญ่มาก เราอาจพบกับปัญหาดังนี้

1

การถ่ายโอนข้อมูลของตารางขนาดใหญ่จะใช้เวลามาก

2

การสร้างดัชนีจะใช้เวลาหลายชั่วโมง

จากปัญหาดังกล่าว จะเป็นการดีถ้าเราสามารถแบ่งข้อมูลในตารางออกเป็นส่วนย่อย ๆ เมื่อข้อมูลแต่ละส่วนมีปริมาณน้อยจะทำให้การทำงานง่ายขึ้นและเร็วขึ้นด้วย โดยที่การแบ่งข้อมูลออกเป็นส่วน ๆ (Partitioning) จะหมายถึง การแบ่งตารางและดัชนีของตารางออกเป็นส่วนย่อยๆที่จะสามารถบริหารจัดการได้ โดยข้อมูลแต่ละส่วนของตารางจะเปรียบเสมือนวัตถุที่แตกต่างกัน โดยที่ก่อนการที่จะทำการแบ่งส่วนข้อมูลควรจะคิด วางแผน และดำเนินการในขั้นตอนการออกแบบคลังข้อมูล เนื่องจากถ้าทำการจัดเก็บข้อมูลลงฐานข้อมูลแล้วค่อยทำการแบ่งข้อมูลจะทำให้เสียเวลาในการทำงานค่อนข้างมาก เมื่อส่วนของข้อมูลหนึ่ง ๆ มีข้อมูลเพิ่มขึ้น เราสามารถแบ่งข้อมูลจากส่วนย่อยออกเป็นส่วนย่อย ๆ ที่เล็กกว่าเดิมได้ เมื่อทำการแบ่งข้อมูลออกเป็นส่วนๆแล้ว ในการเก็บข้อมูลแต่ละส่วนจะเก็บแยกจากกัน โดยจะเก็บแยกดิสก์กันเพื่อเพิ่มประสิทธิภาพในการเข้าถึงข้อมูล

ในการแบ่งส่วนข้อมูลจะมีอยู่ด้วยกัน 2 วิธีคือ

(1)
vertical splitting ซึ่งก็คือการแบ่งคอลัมน์ออกเป็นส่วนๆ ซึ่งแต่ละส่วนจะมีจำนวนแถวข้อมูลเท่า ๆ กันซึ่งจะเท่ากับตารางต้นแบบด้วย การทำ vertical splitting จะเหมาะกับตารางที่มีจำนวนแอ็ทริบิวค่อนข้างมาก และ

(2) horizontal splitting จะตรงกันข้ามกับ vertical splitting โดยในแต่ละส่วนจะประกอบไปด้วยข้อมูลเพียงบางแถวของตารางต้นแบบเท่านั้นแต่จำนวนคอลัมน์ยังคงครบถ้วน

จากที่กล่าวมาข้างต้นจะเห็นได้ว่า การแบ่งข้อมูลออกเป็นส่วน ๆ จะมีประโยชน์มากมายซึ่งสามารถแจกแจงประโยชน์เด่น ๆ ได้ดังนี้

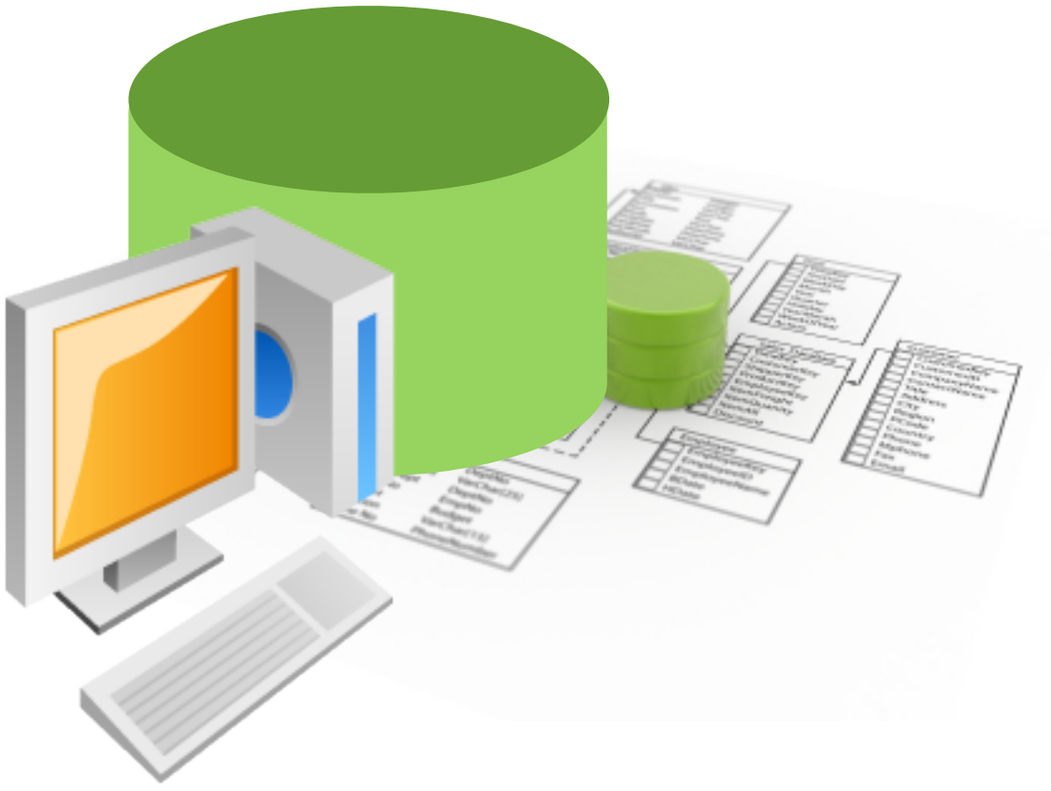
- ในการทำคิวรีจะทำให้ลดการอ่านข้อมูล โดยที่เราจะทำการอ่านข้อมูลเฉพาะส่วนที่จำเป็นเท่านั้น ซึ่งจะทำให้การทำคิวรีนั้นเร็วขึ้นเมื่อทำการเข้าถึงข้อมูลปริมาณไม่มาก
- เราสามารถจัดการกับส่วนข้อมูลแบบ off-line ได้ เราอาจแบ่งตารางเวลาสำหรับดูแลและจัดการกับข้อมูลในแต่ละส่วน นอกจากนี้การแบ่งส่วนข้อมูลจะช่วยให้เราสามารถทำการจัดการกับข้อมูลได้หลายส่วนพร้อม ๆ กันอีกด้วย
- การสร้างดัชนีจะเร็วขึ้น
- การถ่ายโอนข้อมูลเข้าสู่คลังข้อมูลจะง่ายและสามารถบริหารจัดการได้
- ความผิดพลาดของข้อมูลจะมีผลกระทบกับข้อมูลเพียงส่วนเดียวเท่านั้น
- การทำสำเนาข้อมูลและการกู้คืนข้อมูลสามารถทำงานได้เร็วขึ้น

การประมวลผลแบบขนาน

การประมวลผลคิวรีกับคลังข้อมูลโดยส่วนใหญ่จะต้องทำการเข้าถึงข้อมูลเป็นจำนวนมาก อาจต้องทำการหาผลรวมของข้อมูล หรือทำการเลือกข้อมูลจากหลาย ๆ เงื่อนไข เป็นต้น ในการเพิ่มประสิทธิภาพให้กับการประมวลผลคิวรีเราสามารถแบ่งขั้นตอนการทำงานออกเป็นส่วน ๆ จากนั้นทำแต่ละส่วนพร้อมกัน ซึ่งการทำงานที่พร้อมกันของทุกส่วนหรือบางส่วนของขั้นตอนทั้งหมดจะช่วยให้เราสามารถค้นหาผลลัพธ์ที่ต้องการได้เร็วขึ้น ในหลายระบบจัดการฐานข้อมูลได้มีการประมวลผลแบบขนานเตรียมไว้ให้ผู้ใช้ได้ใช้งาน ซึ่งในการใช้งานผู้ใช้ไม่จำเป็นต้องรู้เลยว่าจะต้องแบ่งขั้นตอนการทำงานออกเป็นส่วน ๆ ได้อย่างไร ระบบจัดการฐานข้อมูลจะทำการแบ่งส่วนขั้นตอนการทำงานให้

การประมวลผลแบบขนานสามารถประยุกต์ใช้ได้กับการถ่ายโอนข้อมูล และการปรับโครงสร้างของข้อมูล ซึ่งการทำงานของการประมวลผลแบบขนานควรจะทำงานควบคู่ไปกับ data partitioning schemes เพื่อทำให้ประสิทธิภาพของการประมวลผลแบบขนานเพิ่มขึ้น นอกจากนี้เรายังจำเป็นต้องพิจารณาถึงการประมวลผลแบบขนานในส่วนของฮาร์ดแวร์ด้วยซึ่งจะช่วยเพิ่มประสิทธิภาพการทำงานได้เป็นอย่างดี ตัวอย่างเช่น ทำการเก็บข้อมูล 2 ส่วน (จากการทำ data partitioning) ไว้ในอุปกรณ์จัดเก็บข้อมูลเดียวกัน ถ้าเราต้องการทำ parallel processing กับข้อมูล 2 ส่วนนั้น เป็นต้น

การสร้างผลสรุปของข้อมูล



อย่างที่เราทราบกันดีว่า คลังข้อมูลจะต้องเก็บข้อมูลทั้งที่เป็นข้อมูลที่แสดงรายละเอียดและข้อมูลที่เป็นผลสรุป ดังนั้นเราควรเลือกระดับความละเอียดของข้อมูลให้ดีเพื่อเพิ่มประสิทธิภาพในการจัดเก็บและเข้าถึงข้อมูล ถ้าเราต้องทำการเก็บข้อมูลเกี่ยวกับการขาย โดยเก็บยอดขายรายวันและรายเดือน แต่เผอิญว่าผู้ใช้มักจะต้องการเรียกดูข้อมูลยอดขายรายสัปดาห์บ่อยครั้ง เราจำเป็นต้องพิจารณาถึงการที่จะเก็บข้อมูลยอดขายในแต่ละสัปดาห์ด้วย ในอีกกรณีหนึ่ง ถ้าคลังข้อมูลเก็บยอดขายประจำสัปดาห์และประจำเดือน โดยไม่เก็บยอดขายในแต่ละวัน ทำให้การเรียกดูข้อมูลยอดขายในแต่ละวันไม่สามารถเรียกดูได้ ดังนั้นเราจำเป็นต้องเลือกระดับความละเอียดของข้อมูลให้เหมาะสมกับความต้องการของผู้ใช้งาน

rolling summary

การทำ rolling summary จะมีประโยชน์มากในคลังข้อมูล สมมติว่าเราต้องทำการเก็บข้อมูลแบ่งเป็นรายชั่วโมง รายวัน รายสัปดาห์ และรายเดือน เราต้องสร้างฟังก์ชันการทำงานของเราให้สามารถหาข้อมูลในระดับถัดไปจากระดับก่อนหน้า เช่น เราจะทำการรวมข้อมูลทุก ๆ ชั่วโมงเพื่อสร้างเป็นข้อมูลในแต่ละวัน และทำการรวมข้อมูลทุก ๆ 7 วัน เพื่อที่จะได้ข้อมูลรายสัปดาห์เป็นต้น

การตรวจสอบ Referentail Integrity

Referential Integrity constraints จะเป็นสิ่งที่ทำให้มั่นใจได้ว่าความสัมพันธ์ระหว่าง 2 ตารางนั้นมีความสมบูรณ์ ซึ่งกฎ referential integrity จะทำการควบคุมเกี่ยวกับ คีย์รองใน child table กับคีย์หลักใน parent table ให้มีความสอดคล้องกัน โดยที่แต่ละครั้งที่มีการเพิ่มหรือลบแถวในตาราง ระบบจัดการฐานข้อมูลจะทำการตรวจสอบว่าข้อมูลของตารางที่ถูกลบข้อมูลไปกับข้อมูลที่อยู่ในตารางอื่น ๆ ยังคงไว้ซึ่ง referential integrity หรือไม่ ซึ่งจะการตรวจสอบกฎ referential integrity จะเป็นการตรวจสอบว่า parent rows จะไม่ถูกลบเมื่อมี children rows อยู่ และ children row จะไม่ถูกเพิ่มเข้าสู่ฐานข้อมูลถ้าไม่มี parent row อยู่ก่อนแล้ว ซึ่งการตรวจสอบ referential integrity นั้นจะเป็นขั้นตอนที่มีความสำคัญในระบบการทำงานทั่ว ๆ ไปแต่ก็จะเป็นขั้นตอนที่ลดทอนประสิทธิภาพการทำงานด้วยเช่นกัน

ในการถ่ายโอนข้อมูลเข้าสู่คลังข้อมูล loading images จะถูกสร้างขึ้นใน staging area ซึ่งข้อมูลใน loading images นั้นจะได้มาจากขั้นตอนการสกัดข้อมูล การทำความสะอาดข้อมูล และการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล ซึ่งเป็นข้อมูลที่พร้อมจะทำการถ่ายโอนข้อมูลและเป็นข้อมูลที่มีการตรวจสอบความถูกต้องเกี่ยวกับความสัมพันธ์ในลักษณะ parent-child แล้ว

ดังนั้นเราจึงไม่ต้องทำการตรวจสอบข้อมูลในลักษณะของ referential integrity ในระหว่างการถ่ายโอนข้อมูลอีก ซึ่งจะทำให้เพิ่มประสิทธิภาพได้เป็นอย่างมาก

การกำหนดพารามิเตอร์ต่าง ๆ

ก่อนที่จะติดตั้งระบบฐานข้อมูล เราควรจะต้องวางแผนอย่างรอบคอบว่าจะกำหนดค่าพารามิเตอร์ต่าง ๆ อย่างไร ซึ่งในหลาย ๆ ครั้งจะพบว่าการกำหนดค่าพารามิเตอร์ที่ไม่เหมาะสมจะเป็นการลดทอนประสิทธิภาพของการทำงานได้ ดังนั้นผู้ดูแลคลังข้อมูลควรจะต้องใส่ใจในการเลือกค่าที่เหมาะสมสำหรับพารามิเตอร์ต่าง ๆ

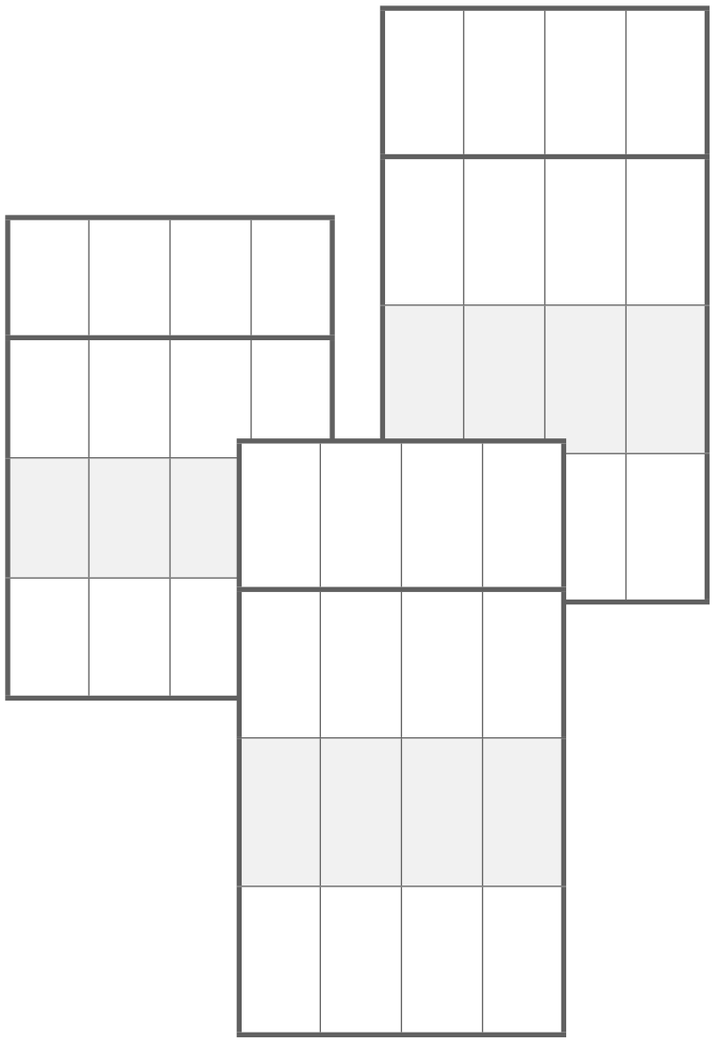
ตัวอย่างเช่น ถ้าเรากำหนดจำนวนผู้ใช้ที่สามารถใช้ได้พร้อมกันให้มีค่าน้อย ๆ อาจทำให้เกิดคอขวดเกิดขึ้นในการใช้งาน ผู้ใช้บางรายอาจต้องรอเพื่อที่จะใช้ฐานข้อมูลจนกระทั่งผู้ใช้รายก่อนหน้าใช้งานเสร็จสิ้น แต่ในกรณีที่กำหนดจำนวนผู้ใช้งานพร้อม ๆ กันให้มีค่าสูงก็จะทำให้เปลืองทรัพยากรของระบบโดยใช่เหตุ

ดังนั้นเราควรจะพิจารณาถึงการตรวจสอบความถี่ในการใช้งาน ซึ่งเป็นการพิจารณาช่วงเวลาระยะห่างระหว่างการใช้งาน 2 ครั้งว่าเป็นยาวนานเพียงใด ถ้าเป็นช่วงเวลาสั้น ๆ เราจะต้องใช้ทรัพยากรของระบบเพิ่มขึ้นเพื่อเตรียมพร้อมสำหรับการใช้งานจากผู้ใช้เป็นจำนวนมาก

จากตัวอย่างข้างต้น เราจำเป็นต้องทบทวนเกี่ยวกับการกำหนดค่าพารามิเตอร์ต่าง ๆ ให้มีความเหมาะสมเพื่อเพิ่มประสิทธิภาพการทำงาน

การจัดเก็บในลักษณะที่เป็น

Data Arrays



สมมติว่าในดาต้ามาร์ทของแผนกการเงินทำการเก็บข้อมูลยอดเงินคงเหลือในแต่ละเดือนของแต่ละบัญชีของบริษัท ในการทำนอร์มอลไลซ์ยอดเงินคงเหลือในแต่ละเดือนสำหรับ 1 ปีจะถูกเก็บอยู่ใน 12 แถวของข้อมูล ถ้าคิวรีโดยส่วนใหญ่ของผู้ใช้ต้องการเรียกดูข้อมูลของทุก ๆ เดือน เราจะทำการเพิ่มประสิทธิภาพการทำงานได้อย่างไร คำตอบคือเราสามารถสร้าง data array ที่เก็บข้อมูลทั้งหมดไว้ด้วยกัน ซึ่งแนวคิดของการทำ data array นั้นจะลดทอนประสิทธิภาพของการทำนอร์มอลไลซ์ แต่ก็เพิ่มประสิทธิภาพในการทำงานได้ ดังนั้นเราจะต้องพิจารณาให้ดีว่าข้อมูลนั้นเหมาะกับการทำ data array หรือไม่

SECTION 8

คำถามท้ายบท

1. จงแจงแจกวัตถุประสงค์ของการออกแบบแบบจำลองทางกายภาพ โดยเรียงลำดับจากวัตถุประสงค์ที่มีความสำคัญมากที่สุดไปยังน้อยที่สุด
2. อะไรคือส่วนประกอบของแบบจำลองทางกายภาพ
3. จงอธิบายเหตุผลว่าเพราะเหตุใดการตั้งชื่อให้เป็นมาตราฐานถึงเป็นสิ่งสำคัญสำหรับการสร้างคลังข้อมูล
4. จงอธิบายถึงเทคนิคในการเพิ่มประสิทธิภาพให้กับการจัดเก็บข้อมูลลงในฐานข้อมูล
5. จงยกตัวอย่างเหตุผลว่าเพราะเหตุใดการสร้างดัชนีแบบ B-tree จึงมีประสิทธิภาพมากกว่าเทคนิคอื่น ๆ
6. การแบ่งส่วนข้อมูลเป็นส่วนย่อย ๆ คืออะไร และสามารถมีส่วนช่วยในการเพิ่มประสิทธิภาพการทำงานของคลังข้อมูลได้อย่างไร

CHAPTER 13

การปรับใช้และการดูแลรักษาคลังข้อมูล

SECTION 1

แผนการสอนประจำบท

วัตถุประสงค์ของบทเรียน

- ศึกษาเกี่ยวกับบทบาทของขั้นตอนการปรับ ใช้คลังข้อมูล ในวงจรการพัฒนาคลังข้อมูล
- ศึกษาเกี่ยวกับกิจกรรมหลักของขั้นตอนการปรับ ใช้คลังข้อมูล
- ศึกษาเกี่ยวกับมาตราการรักษาความปลอดภัยสำหรับคลังข้อมูล
- ศึกษาเกี่ยวกับการสำรองและกู้คืนข้อมูล
- ศึกษาเกี่ยวกับการดูแล จัดการสิ่งต่างๆ และการเฝ้าตรวจสอบการทำงานของคลังข้อมูล

เนื้อหาของบทเรียน

เนื้อหาในบทนี้จะประกอบด้วย การปรับใช้คลังข้อมูล มาตราการความปลอดภัยสำหรับคลังข้อมูล การสำรองและกู้คืนข้อมูล การเติบโตของคลังข้อมูลและการบำรุงรักษา การจัดการต่าง ๆ กับคลังข้อมูล

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารประกอบการสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

กิจกรรมการเรียน-การสอน

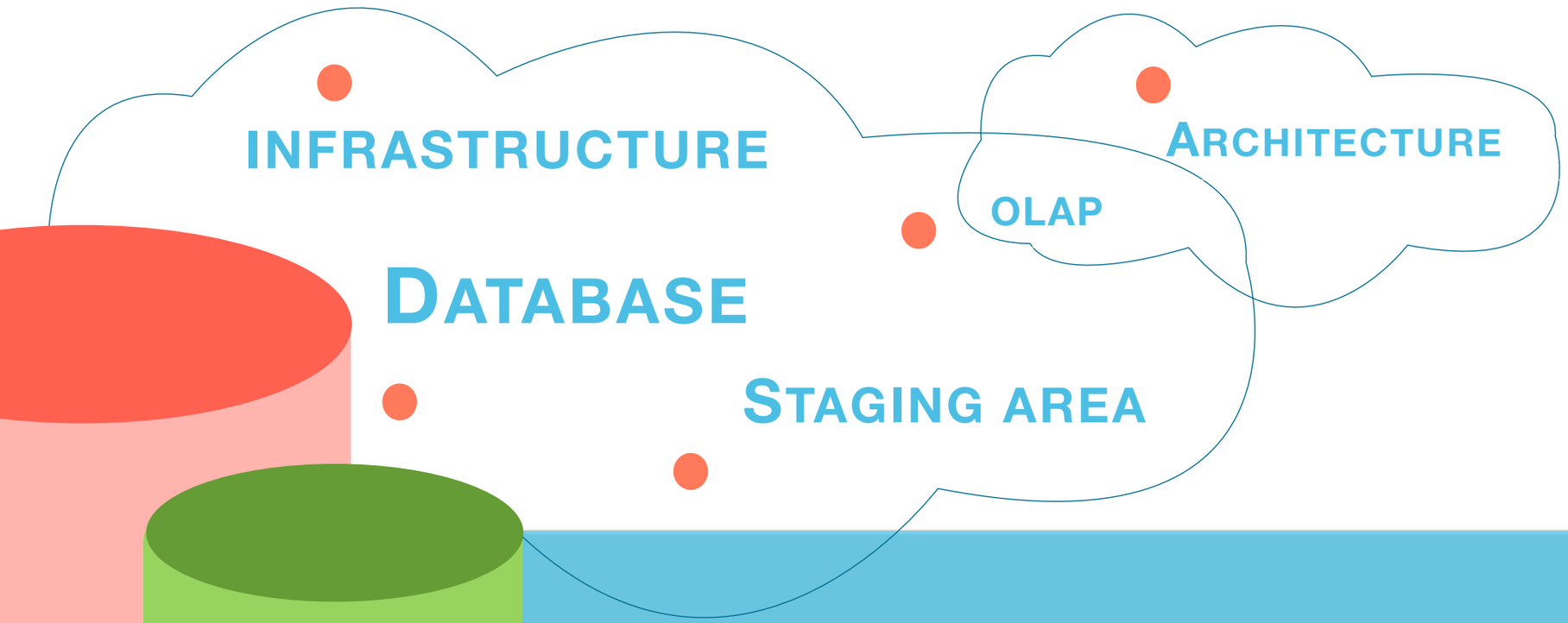
- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารประกอบการสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

การวัดและประเมินผล

- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

SECTION 2

บทนำ



เมื่อเราทำการสร้างคลังข้อมูลรุ่นแรกเสร็จสมบูรณ์ โดยการกำหนดโครงสร้างพื้นฐาน (INFRASTRUCTURE) สถาปัตยกรรม (ARCHITECTURE) ฐานข้อมูล (DATABASE) พื้นที่พักข้อมูล (STAGING AREA) ฟังก์ชันต่าง ๆ เกี่ยวกับการได้มาซึ่งข้อมูลที่ประกอบไปด้วย การสกัดข้อมูล การทำความสะอาดข้อมูลและการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล การสร้างแฟ้มข้อมูลสำหรับถ่ายโอนข้อมูล การสร้างและใช้งานเครื่องมือสำหรับการประมวลผลคิวรีและการสร้างรายงาน การประมวลผลโอแลป (OLAP) และการเรียกใช้คลังข้อมูลผ่านเว็บไซต์ หลังจากขั้นตอนทั้งหมดเสร็จสิ้น เราจะต้องทำการตรวจสอบความถูกต้องและความสมบูรณ์ของคลังข้อมูลก่อนที่จะเริ่มทำการปรับใช้คลังข้อมูลกับองค์กร

ซึ่งโดยปกติแล้วจะทำการตรวจสอบความสอดคล้องกันของฟังก์ชันหรือเครื่องมือต่าง ๆ ที่มาจากผู้ขายที่แตกต่างกันว่าสามารถทำงานร่วมกันได้หรือไม่ และจะทำการตรวจสอบฟังก์ชันการทำงานหลักซึ่งได้แก่ การสกัดข้อมูล การเปลี่ยนแปลงข้อมูล และการถ่ายโอนข้อมูล (อีทีแอล) ซึ่งจะทำการตรวจสอบดังนี้

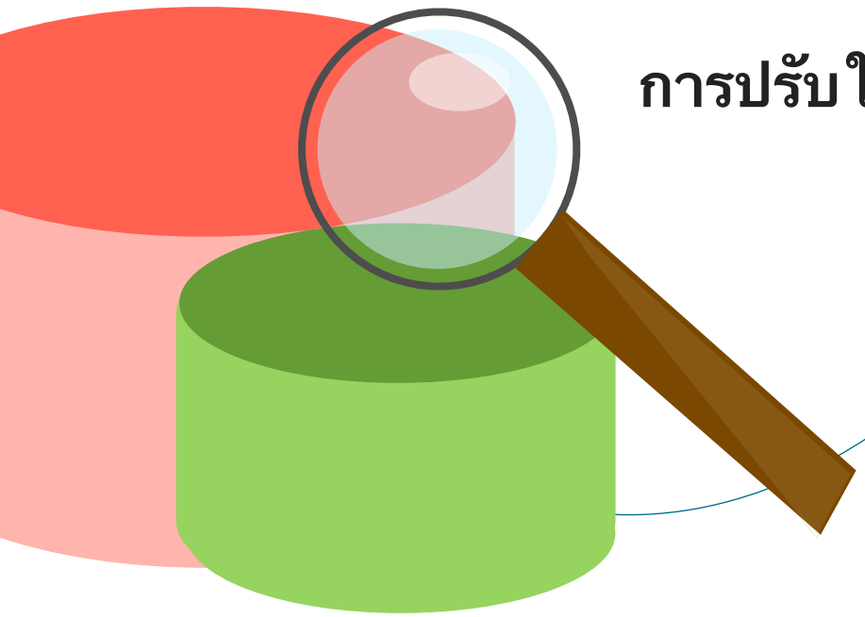
- การตรวจสอบการสกัดข้อมูลจะเป็นการตรวจสอบเพื่อให้แน่ใจว่าทุก ๆ ข้อมูลที่เกี่ยวข้องกับคลังข้อมูลได้ถูกสกัดออกมาจากระบบการดำเนินงานหรือแหล่งข้อมูลได้อย่างถูกต้องและครบถ้วน ซึ่งท้ายสุดจะได้ข้อมูลที่มีความสมบูรณ์ (Data completeness)

- การตรวจสอบการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลและการทำความสะอาดข้อมูลจะเป็นการตรวจสอบเพื่อให้แน่ใจว่าข้อมูลที่ถูกทำการเปลี่ยนแปลง/เปลี่ยนรูปแล้วนั้นมีความถูกต้องตามกฎทางธุรกิจหรือไม่ ซึ่งท้ายสุดเราจะได้ข้อมูลที่มีคุณภาพ (Data quality)

- การตรวจสอบการถ่ายโอนข้อมูลจะเป็นการตรวจสอบว่าการถ่ายโอนข้อมูลหลังจากการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลนั้นมีความถูกต้องหรือไม่ โดยจะทำการตรวจสอบเพื่อให้แน่ใจว่ามีการจัดเก็บข้อมูลลงใน dimension และ fact table อย่างถูกต้องและสมบูรณ์

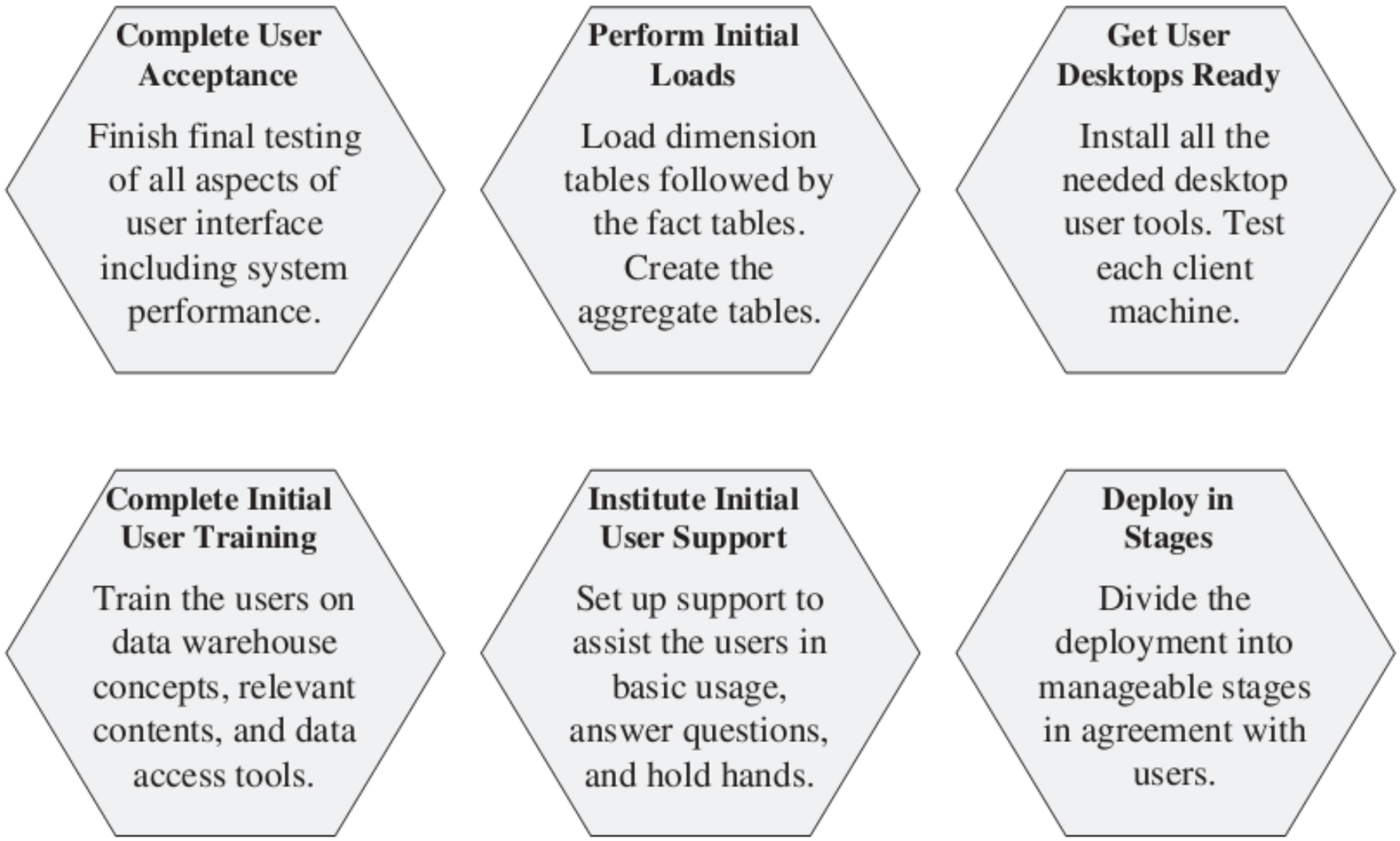
SECTION 3

การปรับใช้คลังข้อมูล



การปรับใช้คลังข้อมูล

หลังจากการตรวจสอบคลังข้อมูลข้างต้นแล้ว เราจะสามารถเริ่มทำการปรับใช้งานคลังข้อมูล และเริ่มทำการอบรมการใช้งานคลังข้อมูลให้กับผู้ใช้ รวมถึงเตรียมการจัดการต่าง ๆ เกี่ยวกับคลังข้อมูล เช่น การสร้างกลไลสำหรับการเก็บรวบรวมความคิดเห็นจากผู้ใช้งานที่มีการส่งต่อให้กับทีมผู้สร้างทราบถึงความเป็นไปของการปรับใช้คลังข้อมูล โดยที่การปรับใช้คลังข้อมูลจะมีกิจกรรมหรือการทำงานต่าง ๆ ดังแสดงในรูปที่ 13-1

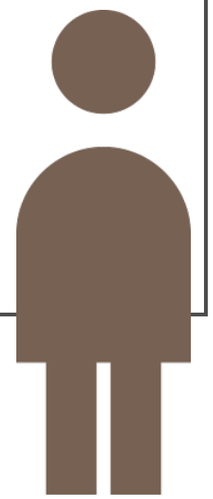


รูปที่ 13-1 ขั้นตอน/กิจกรรมสำหรับการปรับใช้คลังข้อมูล

การทำให้**คลังข้อมูล**เป็นที่ยอมรับของผู้ใช้

จะเปรียบเสมือนการตรวจสอบหรือตรวจรับคลังข้อมูลของผู้ใช้งาน โดยที่เราอาจทำการกำหนดให้ผู้ใช้ที่มีส่วนเกี่ยวกับการสร้างคลังข้อมูล เช่น ผู้ที่บอกถึงความต้องการให้กับทีมผู้สร้าง ทีมผู้บริหารที่ทำการตัดสินใจ ทีมไอทีที่จะดูแลบำรุงรักษา เป็นต้น

บุคคลเหล่านี้จะเป็นบุคคลกลุ่มแรกที่สามารถทดสอบการทำงานของฟังก์ชันต่าง ๆ ของคลังข้อมูล ถ้าคลังข้อมูลที่สร้างขึ้นได้รับการยอมรับจากบุคคลกลุ่มนี้แล้ว เราจะกำหนดให้ผู้ใช้เฉพาะทางที่ต้องทำงานเฉพาะหนึ่ง ๆ ทำการเริ่มต้นการใช้งานหรือทำการตรวจสอบคลังข้อมูลต่อไป จากนั้นจะอนุญาตให้ผู้ใช้ทั่วไปทำการทดสอบคลังข้อมูลต่อไป เมื่อผู้ใช้ทุกภาคส่วนยอมรับในฟังก์ชันการทำงานต่าง ๆ ของคลังข้อมูลแล้ว เราจะเลื่อนไปยังกิจกรรมต่อ ๆ ไปของการปรับใช้คลังข้อมูล



ในการตรวจสอบคลังข้อมูลโดยผู้ใช้จะมีข้อแนะนำหรือแนวทาง**ในการปฏิบัติดังนี้**

ในแต่ละส่วนงานหรือแต่ละแผนกควรกำหนดให้ผู้ใช้ทำการเลือกหรือกำหนดคิวรีทั่ว ๆ ไปเพื่อทดสอบการประมวลผลคิวรีของคลังข้อมูลว่าให้ผลลัพธ์หรือรายงานที่ถูกต้องหรือไม่ โดยหลังจากการประมวลผลคิวรีจากคลังข้อมูลจะทำการตรวจสอบผลลัพธ์ของแต่ละคิวรีโดยการประมวลผลคิวรีที่ระบบการดำเนินงานเพื่อสร้างเป็นรายงานที่มีเนื้อหาเกี่ยวข้องกับรายงานจากคลังข้อมูล จากนั้นทำการเปรียบเทียบระหว่างรายงาน 2 ฉบับ ว่ามีข้อมูลใดในรายงานจากคลังข้อมูลมีความแตกต่างจากรายงานจากระบบการดำเนินงานบ้าง แต่ก่อนที่เราจะทำการเปรียบเทียบเราจะต้องแน่ใจว่ารายงานที่ได้จากระบบการดำเนินงานนั้นเป็นรายงานที่มีความถูกต้องเสียก่อน ซึ่งเมื่อเราทราบถึงความแตกต่างระหว่างรายงาน 2 ฉบับแล้ว เราจำเป็นที่จะต้องแก้ไขให้รายงานมีความถูกต้องสืบไป

โดยปกติของการสร้างคลังข้อมูลจะมีการเตรียมการประมวลผลคิวรีและการสร้างรายงานแบบทั่ว ๆ ไป โดยการกำหนดรูปแบบรายงานแบบทั่ว ๆ ไปไว้ก่อนหน้า (predefined report) ซึ่งเมื่อเราเริ่มปรับใช้คลังข้อมูลเราควรจะต้องกำหนดให้ผู้ใช้ทำการทดสอบเกี่ยวกับการประมวลผลคิวรีหรือการสร้างรายงานจากรูปแบบที่กำหนดไว้ก่อนหน้าด้วย

ควรกำหนดให้ผู้ใช้ทำการทดสอบระบบ OLAP โดยทำการสร้าง multidimensional cube และทำการจัดเก็บ cubes เหล่านั้นไว้ใน multidimensional database (ในกรณีที่เราใช้ MOLAP) และ ต้องปล่อยให้ผู้ใช้ทำการเลือกการวิเคราะห์ข้อมูลที่ต้องการแล้วทำการทดสอบการวิเคราะห์แต่ละครั้งด้วยรายงานจากระบบการดำเนินงาน

ถ้าคลังข้อมูลที่เราสร้างขึ้นสามารถใช้งานผ่านเว็บไซต์ได้ (Web-enabled) เราจะต้องกำหนดให้ผู้ใช้ทำการทดสอบการใช้งานผ่านเว็บไซต์ด้วย

ควรจะต้องทำการทดสอบประสิทธิภาพของคลังข้อมูล ซึ่งประสิทธิภาพอาจหมายถึงเวลาในการคืนค่าผลลัพธ์หรือสร้างรายงานเมื่อผู้ใช้กำหนดและสั่งให้ทำการประมวลผลคิวรี ซึ่งโดยปกติแล้วคลังข้อมูลจะทำการประมวลผลคิวรีทั่ว ๆ ไปที่ไม่ซับซ้อนอะไรมากที่เวลา 3-5 วินาที โดยที่ในการทดสอบเราอาจต้องทำการหาค่าเฉลี่ยของเวลาที่ทำการประมวลผลคิวรีในลักษณะหนึ่ง ๆ แล้วส่งให้ผู้ใช้ทำการตรวจสอบว่าสามารถยอมรับเวลาที่ใช้ในการประมวลผลได้หรือไม่

จากคำแนะนำในการทดสอบคลังข้อมูลเบื้องต้น อาจยังมีการทดสอบในแง่มุมอื่น ๆที่ผู้ใช้ต้องการจะทดสอบหรืออาจจะมีผู้ใช้อีกเป็นจำนวนมากที่ต้องการทดสอบการใช้งานคลังข้อมูลเบื้องต้น ซึ่งในการทดสอบคลังข้อมูลโดยผู้ใช้ เราจะต้องทำอย่างจริงจังเพื่อให้ผู้ใช้ยอมรับฟังก์ชันการทำงานต่าง ๆ และขีดความสามารถของคลังข้อมูลที่สร้างขึ้น ไม่ใช่เพียงแต่การทดสอบและลงนามตรวจรับคลังข้อมูลจากทีม

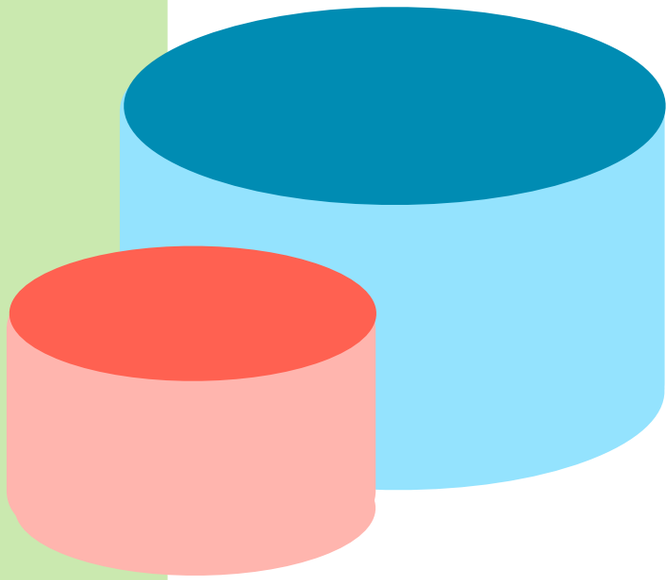
การถ่ายโอนข้อมูลครั้งแรก

หลังจากที่ผู้ใช้ยอมรับคลังข้อมูลที่สร้างขึ้นโดยการทดสอบต่างๆแล้ว ขั้นตอนต่อไปเราจะทำการทดสอบการถ่ายโอนข้อมูลครั้งแรก และหลังจากนั้นจะดำเนินการถ่ายโอนข้อมูลจากระบบการดำเนินงานเข้าสู่ staging area และฐานข้อมูลของคลังข้อมูลเป็นครั้งแรก ซึ่งในการถ่ายโอนข้อมูลครั้งแรกอาจจำเป็นต้องทำการถ่ายโอนข้อมูลเป็นจำนวนหลายแสนหรือหลายล้านเรคคอร์ดซึ่งจะทำให้อาจจะใช้เวลาในการถ่ายโอนข้อมูลหลายวันเลยทีเดียว ซึ่งในการถ่ายโอนข้อมูลครั้งแรกจะมีแนวทางการปฏิบัติต่าง ๆ ดังนี้

เนื่องจากข้อมูลที่ต้องทำการถ่ายโอนข้อมูลครั้งแรกมีเป็นจำนวนมาก ซึ่งอาจทำให้ใช้เวลาค่อนข้างมากในการถ่ายโอนข้อมูล ถ้าเกิดการถ่ายโอนข้อมูลเกิดการล้มเหลวหลังจากทำการถ่ายโอนข้อมูลไป 2-3 วันแล้ว ซึ่งสาเหตุอาจจะเกิดมาจากความล้มเหลวของระบบ และอื่น ๆ เพื่อที่จะแก้ปัญหาเราจะต้องทำการตรวจสอบถึงสาเหตุของการล้มเหลวของการถ่ายโอนข้อมูลแล้วทำการแก้ไข จากนั้นทำการถ่ายโอนข้อมูลต่อจากจุดสิ้นสุดเดิมที่การถ่ายโอนข้อมูลเกิดการล้มเหลวจนกระทั่งการถ่ายโอนข้อมูลครั้งแรกเสร็จสิ้น

ในการถ่ายโอนข้อมูลครั้งแรกเข้าสู่คลังข้อมูล เราจะต้องทำการถ่ายโอนข้อมูลลงใน dimension table และ fact table ต่าง ๆ แต่ด้วยเนื่องจากแต่ละเรคคอร์ดของ dimension table จะมีความสัมพันธ์แบบ one-to-many กับเรคคอร์ดหนึ่ง ๆ ของ fact table และคีย์หลักของ fact table นั้นจะสร้างมาจากการเรียงต่อกันของคีย์หลักของทุก ๆ dimension table ที่เกี่ยวข้องกับ fact table นั้น ๆ

ดังนั้น ในการถ่ายโอนข้อมูลเราจะต้องทำการจัดเก็บข้อมูลลงใน dimension table จนครบเสียก่อน แล้วจึงค่อยทำการจัดเก็บข้อมูลลงใน fact table ซึ่งในการจัดเก็บข้อมูลลงในแต่ละตารางอาจจะต้องมีการสร้างดัชนีสำหรับการจัดเก็บเพื่อเพิ่มประสิทธิภาพให้กับการเข้าถึงข้อมูลอีกด้วย ซึ่งการสร้างดัชนีควรจะทำการสร้างที่ staging area เพื่อลดขั้นตอนการทำงานของการถ่ายโอนข้อมูลที่ต้องทำการถ่ายโอนข้อมูลเป็นจำนวนมาก



การทำให้คอมพิวเตอร์ของผู้ใช้พร้อมใช้งานคลังข้อมูล

ก่อนที่จะปรับใช้คลังข้อมูลเราจะต้องทำให้เครื่องคอมพิวเตอร์ของผู้ใช้พร้อมสำหรับการใช้งานคลังข้อมูล โดยที่เราจะต้องทำการติดตั้งเครื่องมือต่าง ๆ ที่ใช้สำหรับการเข้าถึงข้อมูล (Data access tool) การติดตั้งและเชื่อมต่อเครือข่ายของแต่ละคอมพิวเตอร์เข้ากับเซิร์ฟเวอร์ของคลังข้อมูล และท้ายสุดคือการปรับแต่งเครื่องมือที่เป็น middle-ware ต่าง ๆ โดยที่ก่อนที่จะเริ่มดำเนินการต่าง ๆ เกี่ยวกับเครื่องคอมพิวเตอร์ของผู้ใช้ เราควรจะแจกแจงรายการที่ต้องทำกับคอมพิวเตอร์แต่ละเครื่องที่จะรวมถึงการติดตั้งซอร์ฟแวร์และเครื่องมือต่าง ๆ ในการส่งผ่าน/เข้าถึงข้อมูล และการติดตั้งหรือกำหนดสิ่งต่าง ๆ ของฮาร์ดแวร์ของแต่ละเครื่องคอมพิวเตอร์ของผู้ใช้

ซึ่งในการดำเนินการต่าง ๆ ถ้าเราทำการดำเนินการจากระยะไกล (remote) จะช่วยให้เราประหยัดเวลามากขึ้น ถ้าเรามีการวางแผนที่ดีจะทำให้เราไม่เสียเวลากับการดำเนินการมากนัก หลังจากที่เราติดตั้งและจัดการสิ่งต่าง ๆ กับคอมพิวเตอร์ของผู้ใช้ทั้งหมดแล้ว เราจะต้องทำการทดสอบการใช้งานคลังข้อมูลจากคอมพิวเตอร์เหล่านั้นซึ่งในการทดสอบเราอาจจะทำการสร้าง user และ password ให้กับแต่ละผู้ใช้ เพื่อทำการทดสอบด้วยตนเอง เมื่อเราแน่ใจว่าเครื่องคอมพิวเตอร์พร้อมใช้งานแล้ว จึงทำการอนุญาตให้ผู้ใช้สามารถใช้งานได้

การจัดอบรมให้กับผู้ใช้งานเริ่มแรก

ก่อนที่จะให้ผู้ใช้เริ่มใช้งานคลังข้อมูลเราต้องมีการให้ความรู้หรือจัดอบรมการใช้งานให้กับผู้ใช้เสียก่อน โดยที่เนื้อหาในการอบรมนั้นจะประกอบไปด้วย

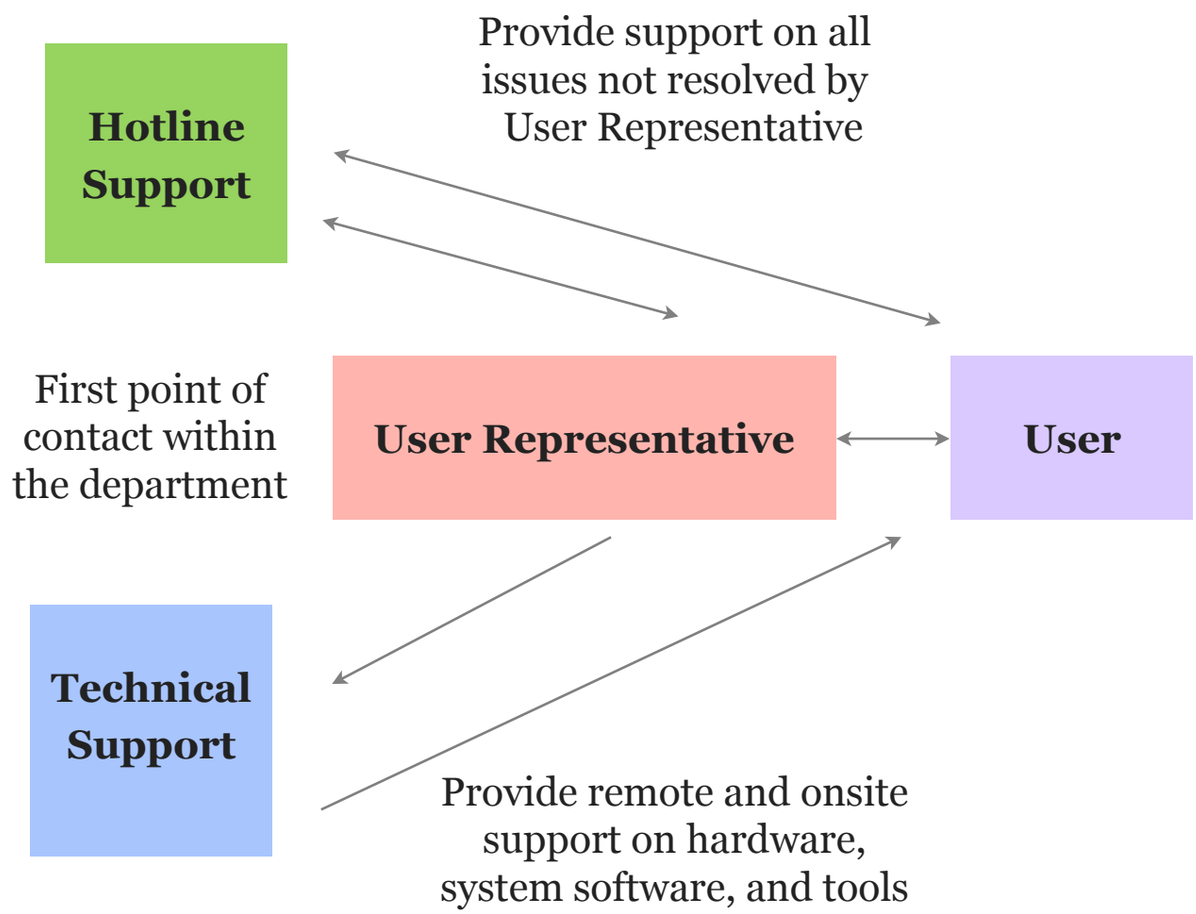
1. ส่วนประกอบต่าง ๆ ของข้อมูล
2. แอพลิเคชั่นต่าง ๆ
3. เครื่องมือต่าง ๆ ที่ผู้ใช้สามารถใช้งานได้

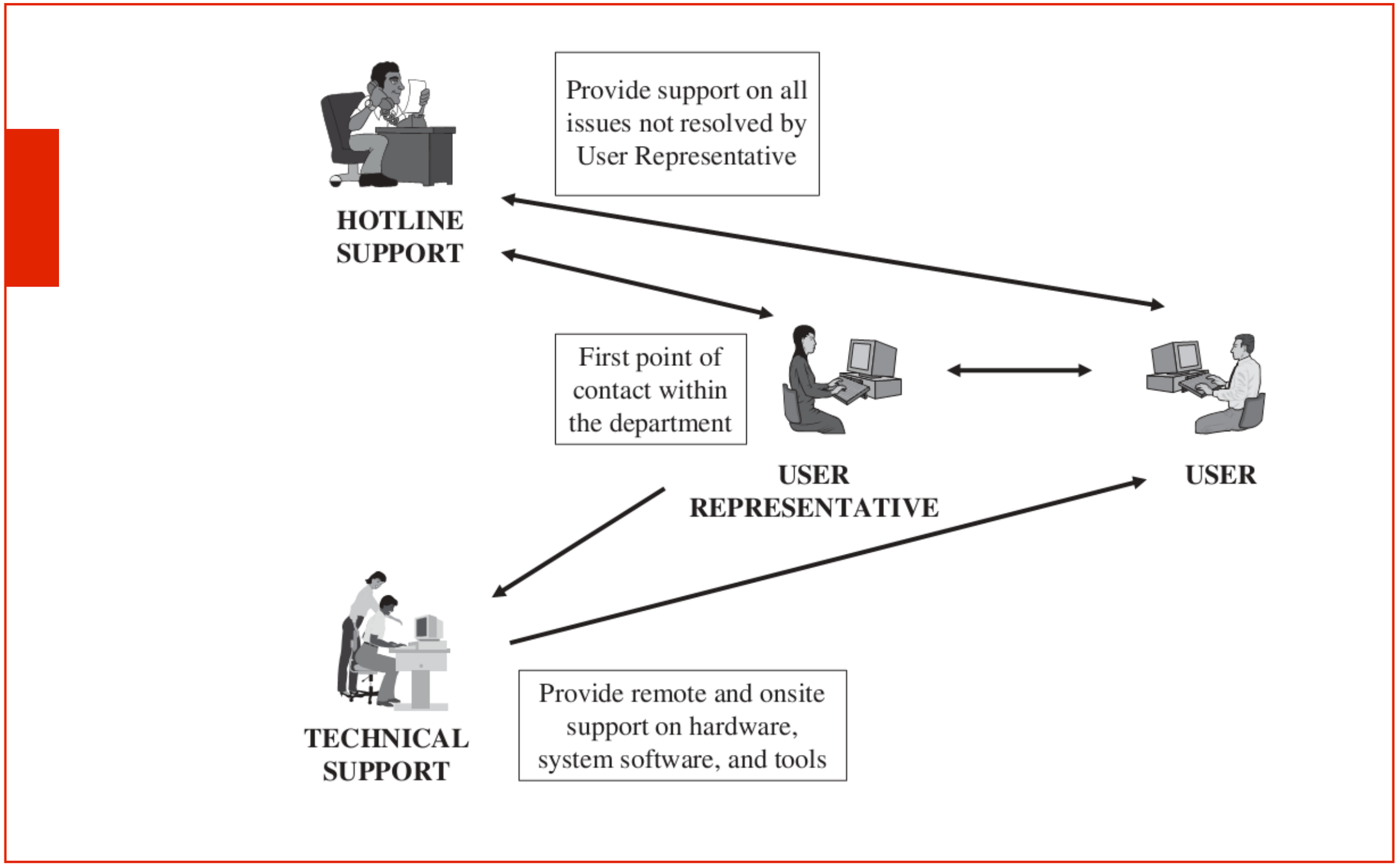
ซึ่งเนื้อหาในการอบรมนั้นควรจะถูกออกแบบมาในมุมมองของผู้ใช้ที่มีต่อคลังข้อมูล ซึ่งเราสามารถออกแบบการอบรมได้ดังต่อไปนี้

- แนวความคิดเกี่ยวกับการจัดเก็บข้อมูลและฐานข้อมูลเบื้องต้น
- คุณลักษณะ (feature) และฟังก์ชันการทำงานของคลังข้อมูลสำหรับผู้ใช้แต่ละกลุ่ม
- ข้อมูลและเนื้อหาสาระของข้อมูลในคลังข้อมูล
- การใช้เครื่องมือในการเข้าถึงข้อมูล
- แอพลิเคชันบนเว็บที่สามารถเข้าถึงข้อมูล/ส่งผ่านข้อมูลจากคลังข้อมูลไปยังผู้ใช้
- กลุ่มของคิวรีและรายงานที่มีการกำหนดไว้แล้ว
- ชนิดของการวิเคราะห์ที่สามารถดำเนินการได้จากคลังข้อมูล
- แม่แบบของคิวรี (Query template) และแนวทางการใช้แม่แบบเหล่านั้น
- รอบของการถ่ายโอนข้อมูลจากระบบการดำเนินงานไปยังคลังข้อมูล
- แนวทางในการช่วยเหลือเมื่อเกิดปัญหาในการใช้งานคลังข้อมูล

การจัดตั้งหน่วยงานเพื่อช่วยเหลือผู้ใช้งานคลังข้อมูล

ก่อนที่จะเริ่มใช้งานคลังข้อมูลเราควรจะต้องจัดเตรียมหรือวางแผนเกี่ยวกับแนวทางการช่วยเหลือผู้ใช้ในการใช้งานคลังข้อมูล ซึ่งในการจัดเตรียมนั้นเราอาจทำการจัดตั้งหน่วยงานหรือกำหนดบทบาทหน้าที่ให้กับพนักงานให้ทำหน้าที่ช่วยเหลือผู้ใช้งานคลังข้อมูล ซึ่งโดยส่วนใหญ่แล้วจะมีการช่วยเหลือคลังข้อมูลดังแสดงในรูปที่ 13-2 โดยที่เมื่อผู้ใช้พบเจออุปสรรคหรือปัญหาในการใช้งานคลังข้อมูล ผู้ใช้จะสามารถติดต่อไปยังตัวแทนผู้ใช้ (กล่าวคือ User presentative ที่ถูกอบรมมาอย่างดีและมีความรู้เกี่ยวกับคลังข้อมูลมากกว่าผู้ใช้ทั่ว ๆ ไป) เพื่อซักถามและแนวทางการแก้ไขปัญหา แต่ถ้าตัวแทนผู้ใช้ไม่สามารถตอบคำถามนั้น ๆ ได้จะทำการติดต่อและส่งต่อคำถามไปยัง hotline support หรือ ฝ่าย technical support เพื่อทำการตอบคำถามให้แก่ผู้ใช้ต่อไป





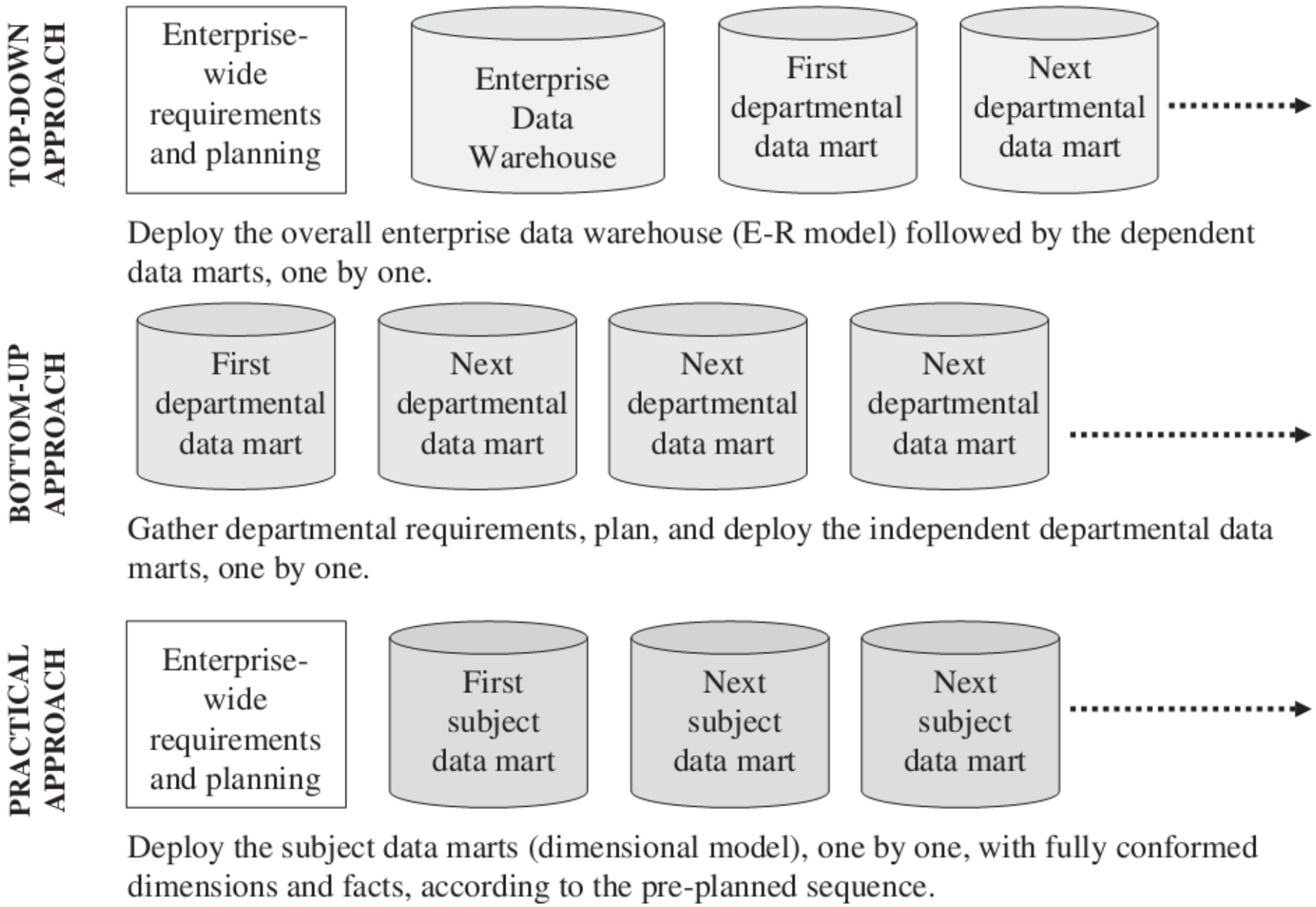
รูปที่ 13-2 การช่วยเหลือผู้ใช้งานคลังข้อมูล

การแบ่งส่วนการปรับใช้คลังข้อมูลออกเป็นส่วนๆ

ในการปรับใช้คลังข้อมูลเราจะต้องทำการเริ่มใช้งานคลังข้อมูลตามแนวทางการสร้างคลังข้อมูลซึ่งจากบทที่ 2 เราจะทราบว่าแนวทางการสร้างคลังข้อมูลจะประกอบไปด้วย 3 วิธีด้วยกันคือ (1) top-down (2) bottom-up และ (3) practical approach ซึ่งจากทั้ง 3 การสร้างที่มีการแตกต่างกัน เราจะสามารถปรับใช้คลังข้อมูลด้วยวิธีที่แตกต่างดังแสดงในรูปที่ 13-3

เมื่อเราทำการสร้างคลังข้อมูลด้วยวิธี top-down เราจะต้องทำการเก็บรวบรวมความต้องการของทั้งองค์รวมถึงทำการสร้างคลังข้อมูลของทั้งองค์กรก่อน (Enterprise data warehouse) จากนั้นค่อยทำการสร้างดาต้ามาร์ทสำหรับแต่ละแผนกต่อไป แต่สำหรับกรณีที่เราสร้างคลังข้อมูลด้วยวิธี bottom-up จะมีขั้นตอนที่น้อยกว่า top-down เล็กน้อย

ซึ่งจะเริ่มจากการสร้างและการปรับใช้แต่ละดาต้ามาร์ทจากนั้นทำการสร้างและปรับใช้ดาต้ามาร์ทต่อ ๆ ไป และท้ายสุดคือ practical approach ซึ่งจะต้องทำการเก็บรวบรวมความต้องการของทั้งองค์กรจะเริ่มสร้างแต่ละดาต้ามาร์ทตามลำดับความสำคัญและความต้องการของแต่ละแผนกต่อไป



รูปที่ 13-3 การปรับใช้คลังข้อมูลตามแนวทางการสร้างคลังข้อมูล

การปรับใช้ต้นแบบคลังข้อมูล

ในการปรับใช้คลังข้อมูล เรามีอีกทางเลือกหนึ่งนั่นคือ การปรับใช้ต้นแบบ (Pilot) ของคลังข้อมูลก่อนที่จะเริ่มการใช้งานจริง ซึ่ง ณ ปัจจุบันหลายบริษัทมักจะนิยมใช้การปรับใช้ต้นแบบในการสร้างคลังข้อมูลด้วยเหตุผลต่าง ๆ การปรับใช้ต้นแบบจะมีประโยชน์หลายข้อด้วยกัน คือ 1) ทำให้ผู้ใช้งานนั้นได้รับประสบการณ์ใหม่ๆ 2) ทำให้ผู้ใช้ได้รับรู้เกี่ยวกับเทคโนโลยีใหม่ที่สามารถประยุกต์ใช้เพื่อช่วยในการดำเนินธุรกิจ และ 3) ทำให้ทีมผู้สร้างสามารถทดสอบหรือพิสูจน์แนวความคิดของคลังข้อมูลกับผู้ใช้ได้ (Proof-of-concept) ว่ามีแนวความคิดตรงกับที่ผู้ใช้คาดหวังและสามารถช่วยเหลือในการดำเนินธุรกิจได้หรือไม่ เป็นต้น

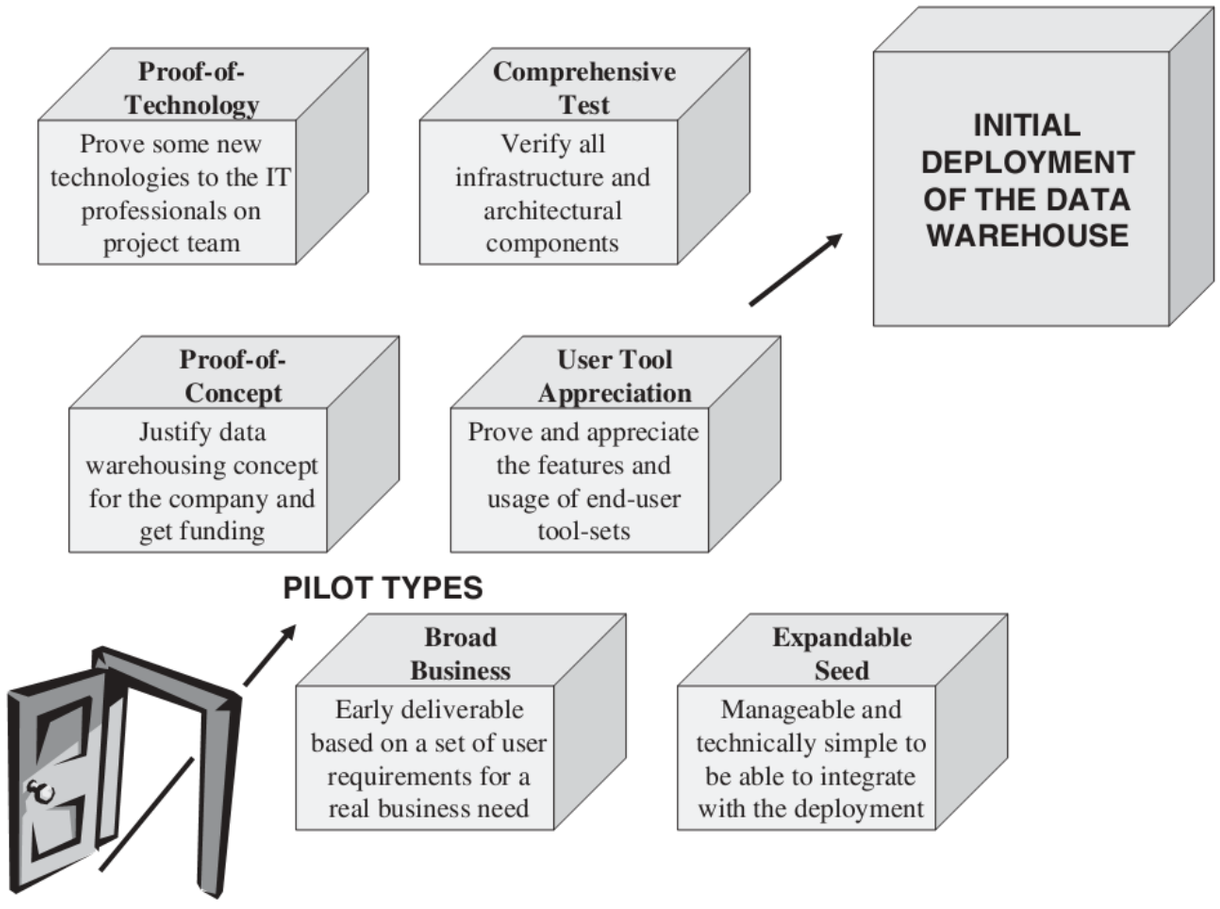
แต่อย่างไรก็ดีในการปรับใช้คลังข้อมูล เราไม่จำเป็นต้องปรับใช้ต้นแบบก่อนเสมอไป อาจมีหลายองค์กรที่มีผู้ใช้คลังข้อมูลที่เป็นนักวิเคราะห์เฉพาะทางที่ต้องการรายงานเฉพาะของแต่ละการดำเนินธุรกิจ ซึ่งเราอาจทำการอบรมและปรับใช้งานคลังข้อมูลได้โดยตรงโดยไม่ต้องใช้ต้นแบบคลังข้อมูล แต่ในบางสถานการณ์เราอาจจำเป็นต้องมีการประยุกต์ใช้ต้นแบบคลังข้อมูล การใช้ต้นแบบจะมีประโยชน์ก็ต่อเมื่อเรามีสิ่งแวดล้อมดังนี้

ผู้ใช้งานคลังข้อมูลยังไม่มีประสบการณ์เกี่ยวกับคลังข้อมูล เป็นผู้ใช้ที่ต้องการได้ประสบการณ์ใหม่ ๆ เกี่ยวกับเครื่องมือและเทคโนโลยี และนักวิเคราะห์ต้องการที่จะรับรู้เกี่ยวกับเกี่ยวกับคุณลักษณะและประโยชน์ของคลังข้อมูล

ทีมผู้สร้างต้องการที่จะแน่ใจว่าฟังก์ชันการทำงานอีทีแอลสามารถทำงานได้อย่างดี

ทีมผู้สร้างต้องการที่จะยืนยันว่าทุกส่วนประกอบของคลังข้อมูลทำงานสอดคล้องกัน เช่น โครงสร้างพื้นฐาน สถาปัตยกรรม การประมวลผลแบบขนาน การเชื่อมต่อ middleware การเข้าถึงข้อมูลผ่านเว็บไซต์ และการใช้ OLAP

ชนิดของต้นแบบที่มีการปรับใช้



รูปที่ 13-4 ชนิดของต้นแบบคลังข้อมูล

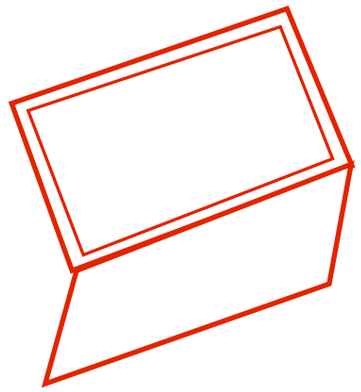
ก่อนที่เราจะทำการตัดสินใจว่าจะทำการปรับใช้ต้นแบบก่อนที่จะมีการปรับใช้คลังข้อมูลจริง เราจะต้องศึกษาถึงชนิดของต้นแบบที่มีอยู่หลายชนิดด้วยกัน ซึ่งแต่ละชนิดต่างก็มีเหตุผลและวัตถุประสงค์ของการทำงานที่แตกต่างกัน ลองพิจารณารูปที่ 13-4 ที่ประกอบไปด้วยต้นแบบ 6 ชนิด ซึ่งแต่ละชนิดจะมีรายละเอียดที่แตกต่างกันดังนี้

Proof-of-concept pilot

—จะเป็นต้นแบบที่ใช้สำหรับพิสูจน์แนวความคิดของคลังข้อมูลทั้งในแง่ของขอบเขตความสามารถของคลังข้อมูล และวิธีการใช้งานหรือวิธีการที่จะได้รับข้อมูลจากคลังข้อมูล เป็นต้น การ พิสูจน์จะสามารถทำได้โดยให้ผู้ใช้ทดลองใช้งานต้นแบบที่มีข้อมูลเพียงบางส่วน การพิสูจน์ในลักษณะนี้จะกระทำในตอนเริ่มต้นของการสร้างคลังข้อมูล โดยจะใช้เวลาไม่นาน ซึ่งโดยส่วนใหญ่แล้วจะใช้เวลาไม่เกิน 6 เดือน ในการสร้างตัวต้นแบบและการทดสอบการใช้งาน ซึ่งการสร้างตัวต้นแบบนี้จะช่วยให้ผู้ใช้สามารถเข้าใจได้ถึงการทำงานของคลังข้อมูลอย่างคร่าว ๆ และจะทำให้โปรเจคการสร้างคลังข้อมูลนั้นถูกอนุมัติให้จัดทำทั้งหมดได้เร็วขึ้น

Proof-of-technology pilot

—จะเป็นต้นแบบที่สร้างขึ้นเพื่อให้ทีมผู้สร้างได้ทำการทดสอบเกี่ยวกับเทคโนโลยีที่มีอยู่ 1 หรือ 2 เทคโนโลยี จากนั้นทำการเลือกว่าเทคโนโลยีที่เราควรจะเลือกใช้ในการสร้างคลังข้อมูล เช่น การพิสูจน์เกี่ยวกับเครื่องมือสำหรับสร้างแบบจำลองมิติต่างๆ (dimensional model) หรือ การตรวจสอบการทำงานและข้อดีของแต่ละเครื่องมือสำหรับสร้างอีทีแอลฟังก์ชัน เป็นต้น เมื่อเราทำการศึกษาถึงเทคโนโลยีและทำการเปรียบเทียบแล้ว เราต้องสามารถบอกได้ว่าเทคโนโลยีเหล่านั้นเพียงพอหรือเป็นที่พอใจสำหรับสร้างคลังข้อมูลหรือไม่ ซึ่งในการสร้างต้นแบบนั้นจะค่อนข้างจำกัด เราสามารถสร้างเป็นกลุ่มของซอร์ฟแวร์เล็ก ๆ หรืออาจจะเป็นการนำแต่ละซอร์ฟแวร์มาทำการทดสอบ เป็นต้น

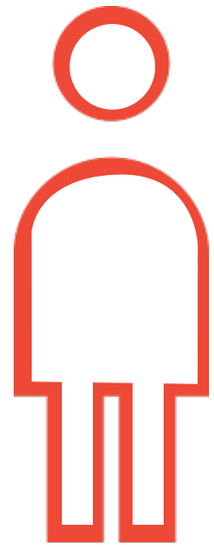


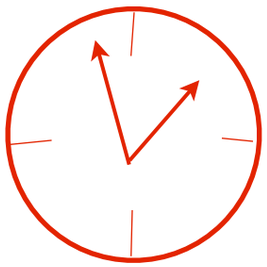
Comprehensive test pilot

—ต้นแบบนี้จะถูกสร้างขึ้นเพื่อทำการทดสอบการทำงานของส่วนประกอบต่าง ๆ ของโครงสร้างพื้นฐาน (Infrastructure) และ สถาปัตยกรรม (Architecture) ของคลังข้อมูล ว่าสามารถทำงานร่วมกันได้เป็นอย่างดีหรือไม่ ในการทดสอบจะทำการทดสอบกับข้อมูลจำนวนไม่มากนัก และจะเน้นที่การเคลื่อนที่หรือการไหลของข้อมูล (Data flow) ตั้งแต่ข้อมูลที่ถูกสกัดจากระบบการดำเนินงาน/แหล่งข้อมูลที่มีการส่งผ่านไปยัง staging area และท้ายสุดข้อมูลที่ส่งผ่านไปยังระบบการเข้าถึง/ส่งผ่านข้อมูล (Information delivery system) เป็นต้น การสร้างต้นแบบชนิดนี้จะช่วยให้ทีมผู้สร้างและผู้ใช้งานคลังข้อมูลได้ทราบถึงความซับซ้อนของการทำงานขั้นตอนต่างๆ และยังได้เก็บเกี่ยวประสบการณ์เกี่ยวกับเทคโนโลยี/เครื่องมือใหม่ ๆ อีกด้วย แต่ด้วยเนื่องจากต้นแบบชนิดนี้ต้องนำส่วนประกอบของฟังก์ชันการทำงานต่าง ๆ มาเชื่อมต่อกันเพื่อเฝ้าดูการเคลื่อนที่ของข้อมูล จึงเป็นเหตุให้กระบวนการสร้างและทดสอบต้นแบบจะต้องทำให้ต้นแบบที่สร้างขึ้นนั้นมีความเหมือนจริงมากที่สุด ซึ่งอาจทำให้ใช้เวลานานได้

User tool appreciation pilot

—ต้นแบบชนิดนี้จะเป็นต้นแบบที่ทำให้ผู้ใช้ได้เห็นลักษณะและรูปลักษณ์ของเครื่องมือที่พวกเขาจะต้องใช้ ต้นแบบนี้จะเน้นที่เครื่องมือต่าง ๆ ในการเข้าถึง/ส่งผ่านข้อมูล ซึ่งเป็นส่วนของหน้าจอ (Interface) ที่ใช้ติดต่อกับผู้ใช้ และจะเป็นต้นแบบที่มีการจัดเก็บเนื้อหาของข้อมูล (Data content) ไว้เบื้องหลังการที่จะทดสอบต้นแบบนี้จะอนุญาตให้ผู้ใช้ได้ทำการทดลองใช้ตัวต้นแบบเพื่อที่จะทำให้ผู้ใช้ได้ทราบถึงความสามารถและคุณสมบัติ/คุณลักษณะต่าง ๆ ของคลังข้อมูล



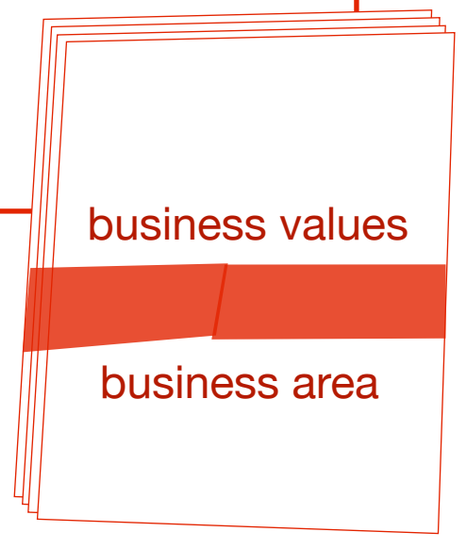


Broad Business pilot

—ต้นแบบชนิดนี้จะเกี่ยวข้องกับขอบเขตทางธุรกิจ โดยนำเอาความต้องการของผู้ใช้เป็นที่ตั้ง เมื่อเราพิจารณาถึงความต้องการจะเป็นความต้องการพิเศษที่ใช้กับการดำเนินกิจกรรมเฉพาะอย่าง เมื่อเราได้ความต้องการเป็นที่เรียบร้อยแล้ว เราจะทำการสร้างต้นแบบตามความต้องการเหล่านั้น เพื่อให้ผู้ใช้ได้เห็นภาพรวมกว้าง ๆ ของการใช้คลังข้อมูล ซึ่งในการสร้างต้นแบบนี้อาจมีข้อจำกัดทางด้านเวลาเพราะไม่ได้เป็นส่วนหลักของคลังข้อมูล ดังนั้นเมื่อเราได้รับความต้องการพิเศษจากคลังข้อมูลแล้ว เราจะต้องทำการวางแผนและกำหนดขอบเขตของตัวต้นแบบ โดยกำหนดให้ตัวต้นแบบมีขนาดเล็กเพื่อที่จะได้ทำการสร้างต้นแบบได้ตามเวลาที่กำหนด และสามารถเชื่อมต่อกับต้นแบบหรือส่วนประกอบอื่น ๆ ได้

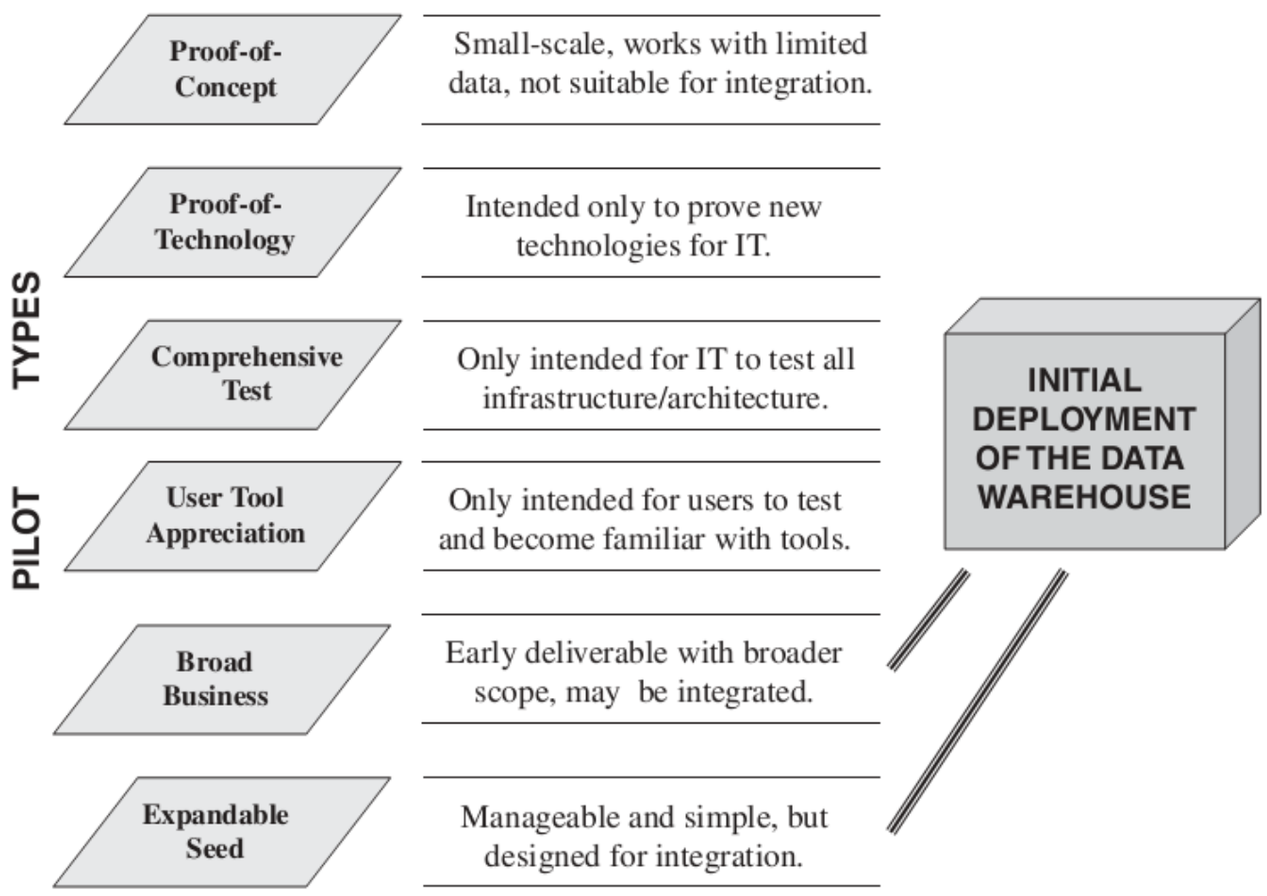
Expandable seed pilot

—ต้นแบบชนิดนี้จะเกี่ยวข้องกับ “business values” โดยจะเป็นต้นแบบที่ผู้ใช้สามารถทดลองใช้งานได้โดยไม่ต้องมีความรู้หรือใช้เทคนิคอะไรมาก เนื่องจากเป็นต้นแบบที่ค่อนข้างง่าย โดยในการสร้างต้นแบบนี้เราจะต้องทำการเลือก “business area” ที่ค่อนข้างง่าย มีประโยชน์ และจับต้องได้ง่าย จากนั้นทำการวางแผนเพื่อทำการต่อเติมเพิ่มขยายส่วนต่าง ๆ ของต้นแบบต่อไป



การเพิ่มขยายต่อเติมและการรวมต้นแบบเข้าด้วยกัน

หลังจากที่เราทำการสร้างต้นแบบเพื่อการทดสอบหรือตอบโจทย์ต่าง ๆ แล้ว เราจะไม่ปล่อยให้ต้นแบบที่สร้างขึ้นนั้นถูกทิ้งหรือไม่ได้ถูกใช้ เราสามารถนำต้นแบบเหล่านั้นมาพัฒนาให้มีความสามารถและความสมบูรณ์เพิ่มขึ้น และทำการรวบรวมต้นแบบเหล่านั้นเพื่อสร้างเป็นคลังข้อมูลที่แท้จริงได้ แต่อย่างไรก็ดีจากชนิดของต้นแบบที่มีความแตกต่างกันทั้งในแง่ของรายละเอียดและวัตถุประสงค์ของการทำงาน บางชนิดของต้นแบบอาจมีความเหมาะสมหรือความเป็นไปได้ที่จะพัฒนาต่อหรือรวมเข้ากับต้นแบบอื่นๆได้ แต่บางต้นแบบอาจไม่เหมาะสม ดังนั้นเราควรที่จะพิจารณาถึงความสามารถของต้นแบบแต่ละชนิดด้วย ดังแสดงในรูปที่ 13-5



รูปที่ 13-5 การรวมต้นแบบเพื่อสร้างคลังข้อมูล

SECTION 4

มาตรการความปลอดภัยสำหรับคลังข้อมูล

มาตรการความปลอดภัย
สำหรับคลังข้อมูล

คลังข้อมูลนั้นเปรียบเสมือนขุมทองข้อมูลขององค์กรหนึ่ง ๆ ที่มีความสามารถในการสร้างข้อมูลเชิงกลยุทธ์ที่มีคุณค่าต่อการดำเนินธุรกิจ ดังนั้นในการปรับใช้คลังข้อมูลเราจะต้องมีการกำหนดมาตรการความปลอดภัยของคลังข้อมูลด้วย โดยมาตรการความปลอดภัยของคลังข้อมูลจะมีความแตกต่างจากมาตรการรักษาความปลอดภัยของระบบการดำเนินงานที่มีการยืนยันหรือระบุตัวตนเพื่อเข้าใช้ระบบ ซึ่งในส่วนของมาตรการสำหรับคลังข้อมูลอาจจะเป็นการกำหนดบทบาทของผู้ใช้ (role/permission) ว่าผู้ใช้แต่ละกลุ่มหรือแต่ละหมวดหมู่จะสามารถเข้าถึงหรือเรียกดูข้อมูลส่วนใดได้บ้าง เป็นต้น

ดังนั้นเพื่อทำให้คลังข้อมูลมีความปลอดภัย เราจะต้องทำการกำหนดนโยบายทางด้านความปลอดภัยที่จะประกอบไปด้วยมาตรการต่าง ๆ ดังนี้

- การกำหนดขอบเขตของข้อมูลที่ต้องมีการรักษาความปลอดภัย เช่น ข้อมูลที่เป็นความลับหรือลับเฉพาะ เป็นต้น
- การวางแผนเกี่ยวกับความปลอดภัยทางกายภาพ เช่น ความปลอดภัยของเซิร์ฟเวอร์ เป็นต้น
- การวางแผนเกี่ยวกับความปลอดภัยทางด้านเครือข่ายและการเชื่อมต่อเครือข่าย
- มาตรการการเข้าถึงข้อมูลในฐานข้อมูล
- การกำหนดบทบาทของผู้ใช้
- มาตรการความปลอดภัยของข้อมูลที่เป็นผลสรุป
- มาตรการความปลอดภัยกับเมตาดาต้า
- มาตรการความปลอดภัยกับ OLAP
- มาตรการความปลอดภัยกับเว็บ
- แนวทางการแก้ไขเมื่อมีการล่วงละเมิดมาตรการความปลอดภัย

SECTION 5

การสำรองและกู้คืนข้อมูล

การสำรองและกู้คืนข้อมูล

อย่างที่เราทราบกันดีว่าทุกระบบสารสนเทศจะมีการสำรองข้อมูลเพื่อประกันความเสี่ยงของการสูญหายของข้อมูล คลังข้อมูลก็เป็นระบบสารสนเทศหนึ่งที่มีความจำเป็นในเรื่องของการสำรองข้อมูลด้วยเช่นกัน ด้วยเหตุที่ข้อมูลในคลังข้อมูลนั้นมีปริมาณค่อนข้างมาก ถ้าเราไม่มีมาตรการสำรองข้อมูลที่ดีเมื่อเกิดความผิดพลาดหรือความล้มเหลวขึ้นกับคลังข้อมูล เราจะต้องทำการสกัดและถ่ายโอนข้อมูลจากระบบปฏิบัติการใหม่ทั้งหมดซึ่งจะทำให้ใช้เวลานานมาก และในบางสถานการณ์ข้อมูลที่ค่อนข้างเก่ามากจะไม่ได้ถูกเก็บอยู่ในฐานข้อมูลของระบบการดำเนินงาน แต่จะเก็บไว้ที่อื่น เช่น เทป หรือพื้นที่สำหรับจัดเก็บข้อมูล (Secondary storage) ซึ่งเป็นเหตุให้เราไม่สามารถเรียกดูหรือเข้าถึงข้อมูลเหล่านั้นได้โดยตรง และเป็นเหตุให้การกู้คืนข้อมูลนั้นทำได้ค่อนข้างยากและใช้เวลานาน

Backup And Recovery

ดังนั้นเพื่อไม่ให้เกิดกรณีดังกล่าวขึ้นกับคลังข้อมูลเราควรที่จะต้องมีมาตรการสำรองและกู้คืนข้อมูลที่ดีเพื่อป้องกันการสูญหาย/สูญเสียข้อมูลสำคัญและเพื่อที่จะทำให้การกู้คืนข้อมูลนั้นสามารถดำเนินการได้อย่างมีประสิทธิภาพ

ในการดำเนินการสำรองและกู้คืนข้อมูลเราจะต้องพิจารณาสิ่งต่าง ๆ ที่มีผลต่อการสำรองและกู้คืนข้อมูลในหลาย ๆ แง่มุมด้วยกัน เช่น (1) ข้อมูลส่วนใดบ้างที่ต้องถูกสำรอง? (2) ควรมีการสำรองข้อมูลเมื่อไร? และ (3) การสำรองข้อมูลจะทำอย่างไร? เป็นต้น ดังนั้นเพื่อที่จะตอบคำถามเหล่านี้ เราควรจะต้องทำการกำหนดกลยุทธ์ในการสำรองและกู้คืนข้อมูล ซึ่งในการดำเนินการจะมีข้อแนะนำและเกร็ดเล็กเกร็ดน้อยต่าง ๆ ดังนี้

1

ทำการกำหนดว่าข้อมูลใดควรจะถูกสำรองบ้าง โดยทำการสร้างลิสต์สำหรับแจกแจงรายละเอียดเกี่ยวกับข้อมูลในฐานข้อมูลของผู้ใช้ ข้อมูลในฐานข้อมูลของระบบ และข้อมูลในล็อกไฟล์ของฐานข้อมูล (Database logs)

2

เราควรจะแยกข้อมูลระหว่างข้อมูลปัจจุบัน (Current data) และข้อมูลย้อนหลัง (historical data) ออกจากกัน เพื่อที่จะได้ไม่ต้องทำการสำรองข้อมูลย้อนหลังบ่อย ๆ ซึ่งจะทำให้การสำรองข้อมูลมีประสิทธิภาพมากขึ้น

3

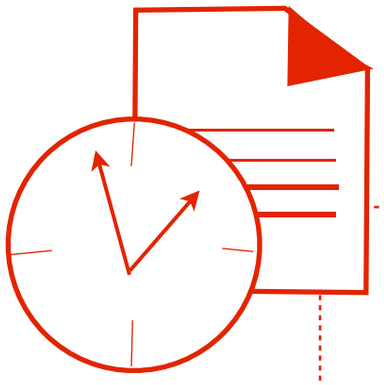
ต้องทำการวางแผนเกี่ยวกับช่วงเวลาหรือระยะที่จะทำการสำรองข้อมูลแต่ละครั้ง ถ้าเรามีการวางแผนในเรื่องของระยะเวลาที่ดีจะช่วยให้เราไม่ต้องทำการสำรองข้อมูลเป็นจำนวนมาก ซึ่งจะช่วยลดเวลาในการสำรองข้อมูลได้

4

เราควรจะมีการสำรอง “log file” นอกเหนือจากการสำรองข้อมูลทั้งหมด (Full backup) โดยที่ “log file” จะมีข้อมูลที่เกี่ยวกับแต่ละรายการ (transactions) ที่เกิดขึ้นหลังจากการสำรองข้อมูลทั้งหมดครั้งล่าสุด หรือรายการที่เกิดขึ้นหลังจากการสำรองข้อมูล “log file” ครั้งสุดท้าย

5

กระบวนการสำรองข้อมูลและการกู้คืนข้อมูลจากคลังข้อมูลจะต้องการการทำงานที่รวดเร็ว เนื่องจากมีคลังข้อมูลนั้นมีข้อมูลเป็นจำนวนมาก ดังนั้นถ้าเราตัดสินใจว่าจะใช้เครื่องมือต่าง ๆ ที่มีวางขายอยู่ในท้องตลาด เราควรพิจารณาถึงประสิทธิภาพของแต่ละเครื่องมือด้วย



การกำหนดตารางเวลาสำหรับการสำรองข้อมูล

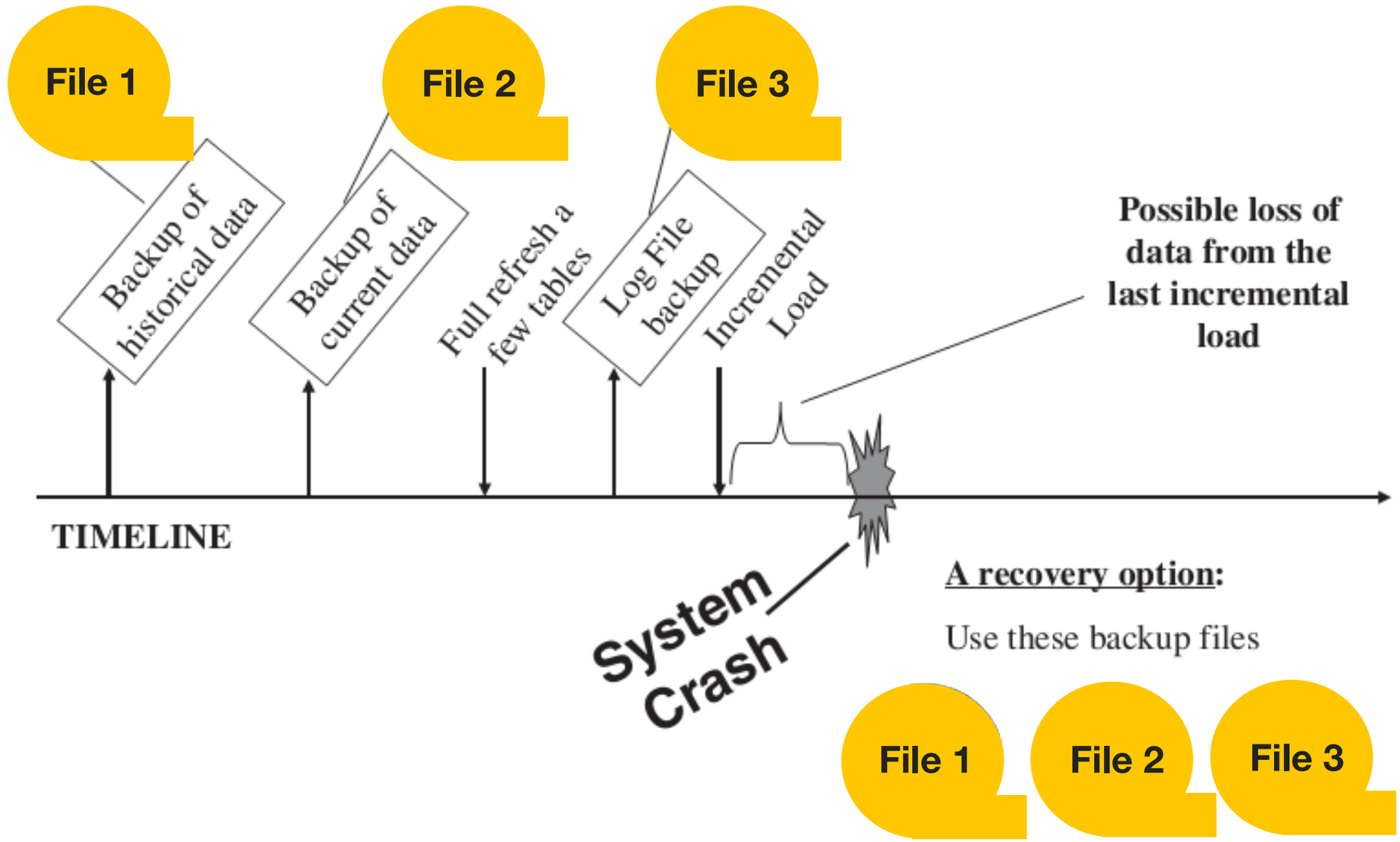
ในการสำรองข้อมูลมีหลายปัจจัยที่เราต้องพิจารณา เนื่องจากขนาดของข้อมูลในคลังข้อมูลมีค่อนข้างมาก การที่จะทำการสำรองข้อมูลทั้งหมดจะใช้เวลานานมาก หรือในอีกกรณีหนึ่งคือการสกัดข้อมูลที่อาจจะเกิดการสูญหายหรือข้อผิดพลาดจากระบบการดำเนินงาน/แหล่งข้อมูลใหม่แล้วทำการถ่ายโอนเข้าสู่คลังข้อมูลใหม่อีกครั้งหนึ่งก็ไม่ใช่ทางเลือกที่ดีเช่นกัน ดังนั้น เราจึงต้องคิดถึงปัจจัยต่าง ๆ ของการสำรองข้อมูลซึ่งมีข้อเท็จจริงดังนี้

- ระบบการดำเนินงานจะมีการสำรองข้อมูลในช่วงเวลากลางคืนเสียเป็นส่วนใหญ่ ส่วนคลังข้อมูลจะใช้เวลากลางคืนสำหรับการอัพเดทข้อมูล (Incremental load) เข้าสู่คลังข้อมูล ดังนั้นเราจำเป็นที่จะต้องคิดพิจารณาถึงช่วงเวลาที่เหมาะสมสำหรับการสำรองข้อมูล ซึ่งอาจจะเป็นช่วงเวลาเดียวกับการอัพเดทข้อมูลเข้าสู่คลังข้อมูล
- ถ้าคลังข้อมูลหนึ่งๆมีผู้ใช้อยู่หลายประเทศ อาจทำให้หาช่วงเวลาที่ไม่มีคนใช้คลังข้อมูลเลยได้ค่อนข้างยาก
- การกำหนดตารางเวลาสำหรับสำรองข้อมูลมักจะพบเจอกับคำถามต่าง ๆ ดังนี้ (1) เมื่อเกิดความผิดพลาดขึ้นผู้ใช้จะสามารถรอกระบวนการกู้คืนข้อมูลได้เป็นเวลาเท่าไร? (2) ในกรณีที่เลวร้ายที่สุดที่ไม่อาจจะกู้คืนข้อมูลได้ ผู้ใช้จะสามารถยอมรับความสูญเสียของข้อมูลได้เป็นจำนวนเท่าไร? (3) เมื่อมีข้อผิดพลาดเกิดขึ้น คลังข้อมูลจะยังสามารถทำงานได้อย่างมีประสิทธิภาพระหว่างการกู้คืนได้หรือไม่?

จากคำถามต่าง ๆ ข้างต้น เราควรจะต้องพิจารณาหรือกำหนดตารางเวลาสำหรับการสำรองข้อมูลอย่างละเอียดถี่ถ้วน ซึ่งการกำหนดตารางเวลาสำหรับการสำรองข้อมูลมักขึ้นอยู่กับสถานการณ์และความต้องการขององค์กร แต่อย่างไรก็ตามการสำรองข้อมูลสำหรับคลังข้อมูลจะมีแนวปฏิบัติที่คล้ายกันดังนี้

- ควรทำการแบ่งข้อมูลในคลังข้อมูลออกเป็น 2 ประเภทคือ“Active data” และ “Static data”
- ควรทำการกำหนดตารางเวลาในการสำรองข้อมูล “Active data” และ “Static data” คนละช่วงเวลากัน
- กำหนดให้มีการสำรองข้อมูล “Active data” บ่อย ๆ และ การสำรองข้อมูล “Static data” ไม่บ่อย
- กำหนดให้การสำรองข้อมูลสามารถทำควบคู่ไปกับการอัพเดทข้อมูลให้กับคลังข้อมูล (Incremental loads)

หลังจากทำการสำรองข้อมูลด้วยวิธีการและขั้นตอนต่าง ๆ แล้ว เมื่อเกิดความล้มเหลวของระบบเกิดขึ้นเราจะสามารถนำข้อมูลที่ทำการสำรองไว้กลับมาใช้ใหม่เพื่อทำให้คลังข้อมูลมีข้อมูลที่ครบถ้วนสมบูรณ์เช่นเดิม เพื่อเป็นการสรุปเกี่ยวกับการสำรองและกู้คืนข้อมูล ลองพิจารณาตัวอย่างดังรูปที่ 13-6 ซึ่งแสดงถึงการสำรองและการกู้คืนข้อมูลเมื่อเกิดปัญหาเกิดขึ้น

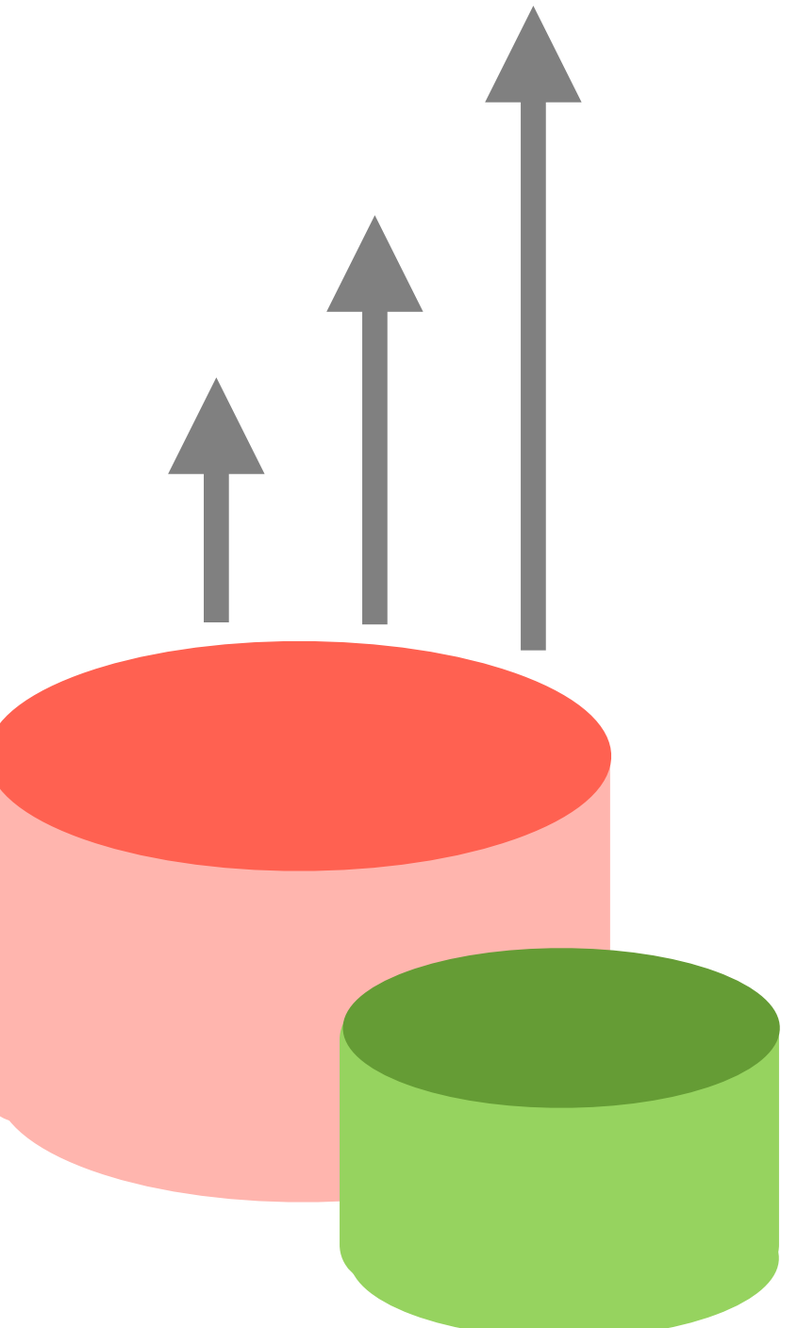


รูปที่ 13-6 การกู้คืนข้อมูล

SECTION 6

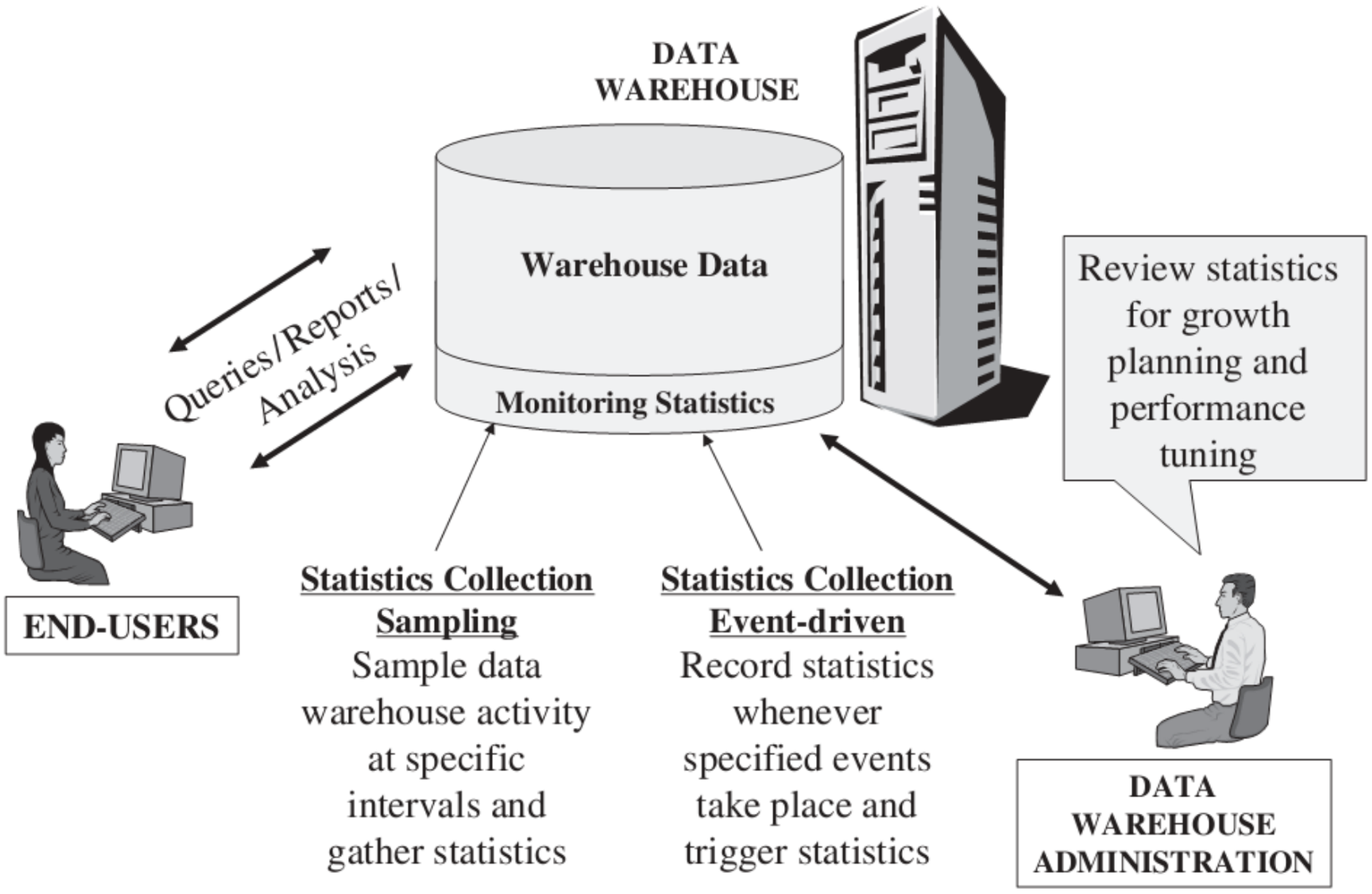
การเติบโตของคลังข้อมูลและการบำรุงรักษา

การเติบโตของคลังข้อมูล และการบำรุงรักษา



จากการปรับใช้คลังข้อมูลจะทำให้ผู้ใช้สามารถเข้าถึงข้อมูลและทำการวิเคราะห์ข้อมูลในแง่มุมต่าง ๆ ได้ และเพื่อให้การทำงานของคลังข้อมูลสามารถดำเนินไปได้อย่างดีและมีประสิทธิภาพ เราควรที่จะต้องทำการเฝ้าติดตามการทำงานของฟังก์ชันการทำงานของคลังข้อมูล ซึ่งจะต้องการฟังก์ชันการติดตามที่มีประสิทธิภาพที่จะสามารถรายงานสถานะของคลังข้อมูลให้กับทีมผู้สร้างหรือผู้ดูแลระบบ เพื่อนำข้อมูลเหล่านั้นไปใช้สำหรับวางแผนเพื่อการพัฒนาคลังข้อมูล

ดังแสดงตัวอย่างในรูปที่ 13-7 ที่จะใช้การเก็บข้อมูลเชิงสถิติที่ต่างจากการใช้งานคลังข้อมูล จากนั้นนำสถิติเหล่านั้นไปทำการวางแผนหรือปรับแก้การทำงานต่าง ๆ เพื่อเพิ่มประสิทธิภาพของคลังข้อมูลต่อไป



รูปที่ 13-7 การเฝ้าติดตามการทำงานของคลังข้อมูล

การเก็บรวบรวมสถิติต่างๆ

ข้อมูลเชิงสถิติที่เราได้รับจากการเฝ้าดูการทำงานของคลังข้อมูลจะเกี่ยวข้องกับข้อมูลการใช้ฮาร์ดแวร์และซอร์ฟแวร์ของคลังข้อมูล ซึ่งจากข้อมูลเชิงสถิติจะทำให้เรารู้ว่าคลังข้อมูลทำงานอย่างไร ดังนั้นเราจึงจำเป็นจะต้องมีการจัดเก็บข้อมูลเชิงสถิติจากการทำงานของคลังข้อมูลที่จะประกอบไปด้วยการจัดเก็บ 2 วิธี คือ

1) Sampling method และ 2) Event-driven method ตามลำดับ

Sampling method

การจัดเก็บข้อมูลเชิงสถิติแบบ Sampling method จะทำการวัดหรือตรวจสอบการดำเนินการของกิจกรรมต่าง ๆ ณ ช่วงเวลาหนึ่ง ๆ ซึ่งเราจะสามารถกำหนดช่วงเวลาเหล่านั้นได้ เช่น ถ้าเรากำหนดช่วงเวลาเป็น 10 นาที สำหรับการเฝ้าดูการใช้งานโปรเซสเซอร์ เราก็จะได้ข้อมูลการใช้งานโปรเซสเซอร์ในทุก ๆ 10 นาที เป็นต้น การเก็บข้อมูลเชิงสถิติแบบ sampling method นั้นจะส่งผลกระทบกับการทำงานของคลังข้อมูลค่อนข้างน้อย

event-driven method

ในส่วนของการจัดเก็บข้อมูลแบบ event-driven method จะเป็นการทำงานที่แตกต่างออกไป ซึ่งจะเป็นการจัดเก็บข้อมูลเชิงสถิติเมื่อมีกิจกรรมใดกิจกรรมหนึ่งที่เราสนใจเกิดขึ้น ตัวอย่างเช่น ถ้าเราต้องการเฝ้าดูเกี่ยวกับการสร้างดัชนีในการจัดเก็บข้อมูลลงตารางในฐานข้อมูล การจัดเก็บสถิติจะเกิดขึ้นเมื่อมีการอัพเดตข้อมูลลงในตารางซึ่งจะต้องมีการสร้างดัชนีด้วย การจัดเก็บข้อมูลแบบ even-driven methods จะทำการรบกวนการทำงานของคลังข้อมูลไม่มากนักแต่ก็จะมากกว่าการทำงานของ sampling methods

จากวิธีการจัดเก็บข้อมูลเชิงสถิติข้างต้น จะมี**คำถาม**ตามมาที่ว่า **เราควรจะใช้เครื่องมือใดในการจัดเก็บข้อมูลเชิงสถิติเหล่านั้น?**

คำตอบก็คือ เครื่องมือที่มาพร้อมกับเซิร์ฟเวอร์ฐานข้อมูล และเครื่องมือในระบบการดำเนินงานหรือเราอาจจะเพิ่มเครื่องมือในการจัดเก็บข้อมูลเชิงสถิติก็เป็นได้ ซึ่งในการเลือกใช้เครื่องมือนั้นจะขึ้นกับสิ่งแวดล้อมที่เรามีและความเข้ากันได้ของส่วนประกอบต่าง ๆ ของคลังข้อมูลที่เราทำการสร้างขึ้น

จากที่กล่าวทั้งหมดข้างต้นข้อมูลเชิงสถิติจะถูกใช้และสามารถจัดเก็บได้หลายวิธี ซึ่งจะสามารถใช้เครื่องมือต่าง ๆ ในการจัดเก็บได้ แต่ก่อนที่จะทำการจัดเก็บข้อมูลเชิงสถิติเราควรจะต้องทราบถึงชนิดของข้อมูลเชิงสถิติที่มีอยู่ด้วยกันหลายชนิดด้วยกันดังนี้

- ข้อมูลการใช้พื้นที่ใน physical disk storage
- จำนวนครั้งที่ระบบจัดการฐานข้อมูลต้องทำการค้นหาพื้นที่ในบล็อกต่างๆ เพื่อหา fragmentation
- กิจกรรมต่างๆที่เรียกใช้หน่วยความจำบัฟเฟอร์
- ประสิทธิภาพการทำงานของอินพุต-เอาท์พุต
- การจัดการหน่วยความจำ
- ขนาดของแต่ละตารางในฐานข้อมูล
- การเข้าถึงเรคคอร์ดใน fact table
- จำนวนคิวรีที่ถูกประมวลผลในช่วงเวลาหนึ่งๆระหว่างวัน
- เวลาที่ผู้ใช้แต่ละคนใช้งานคลังข้อมูล
- จำนวนผู้ใช้คลังข้อมูลในแต่ละวัน
- จำนวนผู้ใช้คลังข้อมูลสูงที่สุดในช่วงเวลาหนึ่งๆของแต่ละวัน
- ระยะเวลาที่ทำการ "incremental loads" ในแต่ละวัน
- จำนวนผู้ใช้งานที่ยังคงใช้งานคลังข้อมูลอยู่
- เวลาในการคืนค่าผลลัพธ์จากคิวรีของผู้ใช้
- จำนวนรายงานที่ต้องสร้างให้กับผู้ใช้ในแต่ละวัน
- จำนวนตารางในฐานข้อมูลของคลังข้อมูลที่ยังคงมีการใช้งาน

การใช้สถิติในการวางแผนเพิ่มการเติบโตของคลังข้อมูล

เมื่อเราปรับใช้คลังข้อมูลหลายเวอร์ชั่น จะทำให้คลังข้อมูลมีผู้ใช้มากขึ้น และคิวรีที่ต้องการประมวลผลมีความซับซ้อนมากขึ้น ซึ่งจากสถานะการที่เป็นอยู่ เราจะสามารถวางแผนเพื่อเพิ่มการเติบโตของคลังข้อมูลได้อย่างไร หรือเราจะสามารถตอบคำถามต่าง ๆ เหล่านี้ได้อย่างไร เช่น ทำไมประสิทธิภาพของการประมวลผลคิวรีถึงลดลง ทำไมคลังข้อมูลถึงเกิดความล้มเหลวเมื่อมีการขยายขนาดของตารางที่ใช้เก็บข้อมูล เป็นต้น

ซึ่งจากแนวทางการพัฒนาให้คลังข้อมูลเติบโตขึ้น เราจะต้องเฝ้าตรวจสอบข้อมูลเชิงสถิติที่เป็นเหมือนคำบอกใบ้ว่าเกิดอะไรขึ้นกับคลังข้อมูล เพื่อที่เราจะสามารถหาแนวทางในการพัฒนาคลังข้อมูลต่อไป ซึ่งหลังจากการจัดเก็บข้อมูลเชิงสถิติ เราอาจจะสามารถใช้ข้อมูลเหล่านั้นเพื่อจัดการกับการกระทำต่าง ๆ ดังนี้

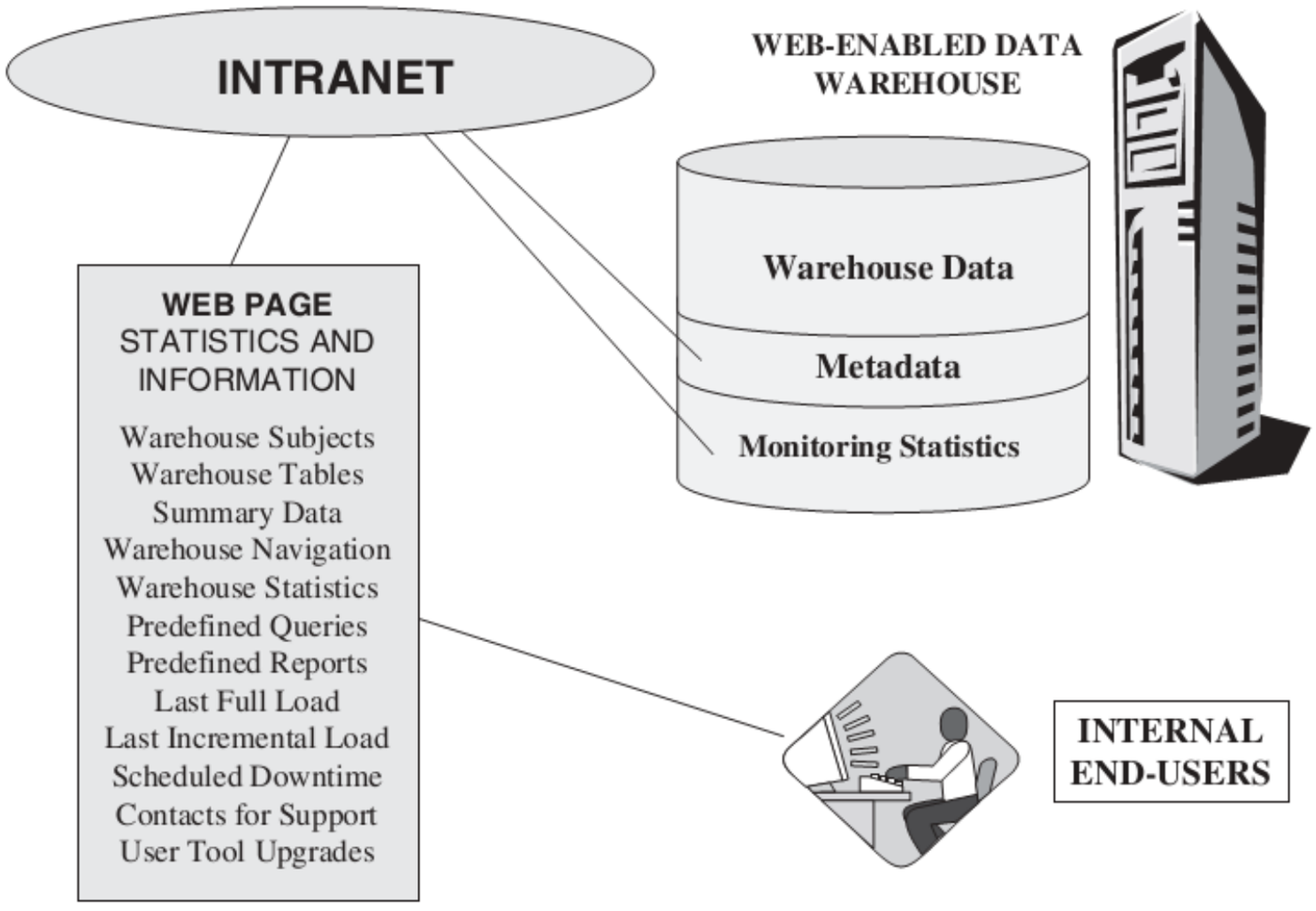
- การจองเนื้อที่ในดิสก์เพิ่มขึ้นสำหรับตารางข้อมูลในฐานข้อมูล
- การวางแผนสำหรับการใช้ในดิสก์ก้อนใหม่สำหรับตารางใหม่ที่จะถูกเพิ่มเข้าไปในฐานข้อมูล
- การปรับค่าพารามิเตอร์ที่เกี่ยวกับการจัดการบล็อกของข้อมูลในไฟล์เพื่อลด fragmentation ให้น้อยที่สุด
- การสร้างตารางสรุปรวบยอดที่สามารถตอบสนองคิวรีเป็นจำนวนมากที่ต้องการข้อมูลแบบผลสรุป
- การปรับเปลี่ยนแฟ้มข้อมูลใน staging area เพื่อที่จะสามารถรองรับข้อมูลได้มากขึ้น
- การเพิ่มหน่วยความจำเพื่อช่วยในเรื่องของการจัดการหน่วยความจำ
- การอัพเกรดเซิร์ฟเวอร์สำหรับฐานข้อมูล
- การปรับช่วงเวลาที่มีการใช้คลังข้อมูลหนาแน่นในระหว่าง 24 ชั่วโมงให้มีความสมดุล
- การแบ่งตารางออกเป็นส่วน ๆ เพื่อที่จะสามารถทำการถ่ายโอนข้อมูลแบบขนานได้ และยังช่วยในเรื่องของการสำรองข้อมูลอีกด้วย

การใช้สถิติในการปรับแต่ง
คลังข้อมูล

หลังจากที่เราทำการเก็บสถิติต่าง ๆ แล้วเราสามารถนำสถิติเหล่านั้นไปปรับปรุงประสิทธิภาพการทำงานของฟังก์ชันต่าง ๆ ได้ เช่น

- การปรับปรุงประสิทธิภาพของการประมวลผลคิวรี
- การปรับการกำหนดคิวรี
- การปรับการทำงานของ “incremental loads”
- การปรับความถี่ของ “OLAP loads”
- การปรับการทำงานของ OLAP
- การปรับปรุงการเรียกดูเนื้อหา/ข้อมูลจากคลังข้อมูล
- การปรับแบบฟอร์มของรายงาน
- การปรับปรุงการสร้างรายงานต่างๆ

จากที่กล่าวมาข้างต้นทั้งหมดจะเป็นการใช้ข้อมูลสถิติที่เกี่ยวข้องกับการใช้งานคลังข้อมูลที่ได้จากฟังก์ชันการเฝ้าดูการทำงานของคลังข้อมูลเมื่อได้ข้อมูลสถิติแล้ว ทีมผู้ดูแลคลังข้อมูลจะทำการประเมินสถิติเหล่านั้นแล้วคิดวิธีหรือกลยุทธ์ที่จะพัฒนาคลังข้อมูลให้มีประสิทธิภาพยิ่ง ๆ ขึ้นไป แต่อย่างไรก็ดี ข้อมูลสถิติยังมีอีกหลายแง่มุม ซึ่งในบางข้อมูลเราอาจจำเป็นต้องเปิดเผยข้อมูลสถิติเหล่านั้นให้แก่ผู้ใช้ ดังแสดงในรูปที่ 13-8 ข้อมูลเหล่านี้อาจจะเป็นข้อมูลที่ใช้บอกรายละเอียดต่าง ๆ ของคลังข้อมูล ซึ่งอาจจะส่งผลต่อการใช้งานคลังข้อมูลของผู้ใช้ได้



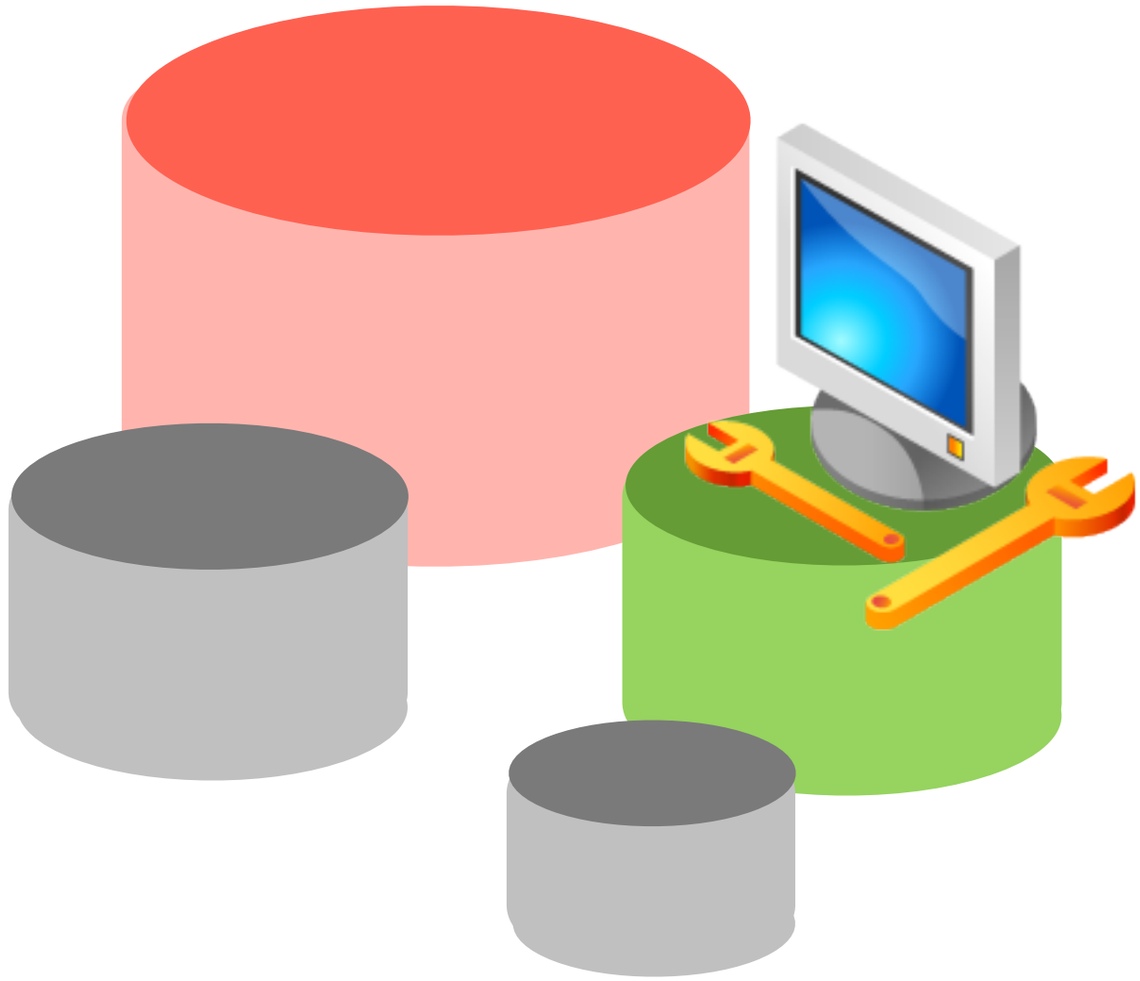
รูปที่ 13-8 ข้อมูลเชิงสถิติสำหรับผู้ใช้

SECTION 7

การจัดการต่าง ๆ กับคลังข้อมูล

การจัดการต่าง ๆ กับคลังข้อมูล

หลังจากที่เริ่มมีการปรับใช้งานคลังข้อมูลแล้ว เราจะสามารถจัดการสิ่งต่าง ๆ กับคลังข้อมูลได้ 2 รูปแบบ คือ (1) การจัดการโดยการดูแลรักษาคลังข้อมูล (Maintenance management) ที่พยายามทำให้ฟังก์ชันการทำงานต่าง ๆ ของคลังข้อมูลนั้นทำงานได้อย่างดีที่สุด และ (2) การจัดการความเปลี่ยนแปลง (Change management) ที่จะเน้นที่การเพิ่มประสิทธิภาพและการปรับแก้ไขคลังข้อมูลต่อไป



จากการจัดการทั้งสองประเภท หลังจากกรับใช้ข้อมูลแล้วเราจะสามารถจัดการกับสิ่งต่างได้มากมาย เช่น

- การจัดการกับการเติบโตของข้อมูล (Data growth management)
- การจัดการกับการจัดเก็บข้อมูล (Storage management)
- การจัดการกับเครือข่าย (Network management)
- การจัดการกับฟังก์ชันอีทีแอล (ETL management)
- การจัดการกับดาต้ามาร์ทใหม่ที่จะถูกสร้างขึ้นและปล่อยให้ใช้งาน (Management of future data marts releases)
- การเพิ่มประสิทธิภาพการเข้าถึง/ส่งผ่านข้อมูล (Enhancements to information delivery)
- การจัดการความปลอดภัย (Security administration)
- การจัดการการสำรองและกู้คืนข้อมูล (Backup and recovery management)
- การจัดการกับเว็บ (Web technology administration)
- การอัพเกรดแพลตฟอร์มต่างๆ (Platform upgrades)
- การจัดการการอบรมผู้ใช้ (Ongoing training)
- การจัดการเกี่ยวกับการสนับสุนนผู้ใช้งาน (User support)

ซึ่งจากการจัดการต่าง ๆ ที่มีมากมาย เราลองพิจารณาแต่ละการจัดการหลัก ๆ ดังนี้

อย่างที่เราทราบกันดีว่าแพลตฟอร์มการคำนวณของคลังข้อมูลนั้นจะเกี่ยวข้องกับฮาร์ดแวร์ ระบบปฏิบัติการ ระบบที่ติดต่อสื่อสารกับระบบอื่น ๆ หรือผู้ใช้งานและอื่น ๆ ซึ่งเมื่อเวลาผ่านไปเราอาจต้องทำการอัพเกรดแพลตฟอร์มที่เราใช้อยู่เพื่อรองรับการทำงานในปัจจุบันและอนาคต แต่ก่อนที่เราจะทำการอัพเกรดแพลตฟอร์ม เราจะต้องมีการวางแผนเกี่ยวกับการประยุกต์ใช้แพลตฟอร์มใหม่เข้ากับระบบคลังข้อมูลเดิม ถ้าเรามีการวางแผนที่ดีจะทำให้การประยุกต์ใช้แพลตฟอร์มใหม่จะไม่รบกวนหรือขัดขวางการทำงานของระบบคลังข้อมูลแต่อย่างไร ซึ่งในการอัพเกรดแพลตฟอร์มโดยส่วนใหญ่จะเริ่มจากผู้ขายพยายามที่จะบังคับให้เราทำการอัพเกรดตามกำหนดเวลาที่ผู้ขายเหล่านั้นได้ออกผลิตภัณฑ์ใหม่ ๆ แต่ถ้า ณ ช่วงเวลานั้นเราไม่สะดวกในการอัพเกรด เราจะต้องยืนกันกับผู้ขาย และเลื่อนเวลาในการอัพเกรดไปจนกว่าแพลตฟอร์มที่ใช้อยู่จะไม่สามารถทนต่อการทำงานได้

การจัดการกับการเติบโตของข้อมูล

โดยปกติของคลังข้อมูลจะมีข้อมูลอยู่เป็นจำนวนมาก ซึ่งถึงแม้ว่าข้อมูลจะเพิ่มขึ้นจากเดิมเพียงเล็กน้อย แต่ก็จะทำให้คลังข้อมูลนั้นมีข้อมูลเป็นจำนวนมากอยู่ดี ซึ่งเมื่อข้อมูลมีจำนวนเพิ่มขึ้นเราจะต้องจัดการกับข้อมูลเก่าที่มีอยู่ในคลังข้อมูลอยู่ก่อนหน้าแล้ว และข้อมูลใหม่ที่เพิ่งจะถูกเพิ่มเข้าไปด้วย ซึ่งในหลาย ๆ กรณี คลังข้อมูลอาจมีข้อมูลย้อนหลังเป็นจำนวนมากซึ่งเวลาผ่านไปข้อมูลเหล่านั้นอาจจะไม่ได้ใช้งาน ซึ่งเราจะต้องมีการจัดการกับสิ่งต่าง ๆ เหล่านี้ดังต่อไปนี้

เมื่อข้อมูลใน dimension table และ fact table เป็นข้อมูลที่มีรายละเอียดสูง เราอาจทำการรวบรวมข้อมูลเหล่านั้นให้เป็นผลสรุปของข้อมูล โดยทำการเก็บข้อมูลเฉพาะข้อมูลที่เป็นผลสรุปเท่านั้น

จำกัดการเรียกดูข้อมูลแบบเจาะลึกที่ไม่จำเป็นในบางมิติลง และทำการลบข้อมูลที่มีรายละเอียดสูงๆเหล่านั้นออกจากฐานข้อมูล

ทำการจำกัดปริมาณข้อมูลย้อนหลัง โดยทำการเคลื่อนย้ายข้อมูลที่เก่ามาก ๆ ออกจากฐานข้อมูลของคลังข้อมูล

- ข้อมูลที่ถูกจัดเก็บในฐานข้อมูลจะเพิ่มขึ้นตลอดเวลา ดังนั้นเราควรจะต้องคำนึงถึงการจัดการเกี่ยวกับการจัดเก็บข้อมูลด้วย ซึ่งจะมีแนวปฏิบัติดังนี้
- การพัฒนาเวอร์ชั่นใหม่ของคลังข้อมูลจะทำให้มีการจัดเก็บข้อมูลเพิ่มขึ้น ซึ่งจะทำให้เราต้องทำการวางแผนสำหรับข้อมูลที่เพิ่มขึ้นด้วย
- เราจะต้องมั่นใจว่าการสร้าง การติดตั้ง และการกำหนดพารามิเตอร์ต่าง ๆ ของการจัดเก็บข้อมูลนั้นมีความยืดหยุ่นและเราสามารถทำการปรับเปลี่ยนต่อเติมได้ และเราจะต้องสามารถเพิ่มขนาดของพื้นที่สำหรับจัดเก็บข้อมูลได้โดยทำการรบกวนการทำงานของคลังข้อมูลให้น้อยที่สุด
- เมื่อมีการเรียกใช้งาน/เข้าถึงข้อมูลเพิ่มขึ้น เราจะต้องวางแผนเกี่ยวกับการกระจายข้อมูลออกไปยังหลาย ๆ ดิสก์หรือหลายที่เพื่อลดคอขวดในการเข้าถึงข้อมูล
- ถ้าระบบคลังข้อมูลที่เราสร้างขึ้นเป็นแบบระบบการประมวลผลแบบกระจาย (Distributed system) ที่มีหลายเซิร์ฟเวอร์ที่มีการใช้ดิสก์ร่วมกัน เราจะต้องพิจารณาถึงการเชื่อมต่อของเซิร์ฟเวอร์ที่จะทำการติดต่อไปยังเซิร์ฟเวอร์ที่มีข้อมูลอยู่ โดยจะต้องทำให้การติดต่อนั้นมีประสิทธิภาพมากที่สุด
- ต้องมีกระบวนการในการย้ายข้อมูลจาก “bad storage sectors” ไปยังส่วนที่ใช้งานได้

การจัดการกับฟังก์ชันอีทีแอล

ETL

จะเป็นการจัดการกับฟังก์ชันการทำงานต่าง ๆ ของอีทีแอล โดยพยายามที่จะทำให้การทำงานของทุก ๆ ฟังก์ชันสามารถทำงานได้อย่างอัตโนมัติ โดยที่การจัดการเกี่ยวกับฟังก์ชันการทำงานต่าง ๆ ของอีทีแอลสามารถทำได้โดยการติดตั้งระบบแจ้งเตือนที่จะสามารถแจ้งเตือนผู้ดูแลคลังข้อมูลเมื่อมีเหตุการณ์ผิดปกติเกิดขึ้นกับฟังก์ชันอีทีแอล โดยที่ในการจัดการกับฟังก์ชันอีทีแอล เราจะสามารถจัดการกับสิ่งต่าง ๆ ได้ดังนี้

- ทำการสกัดข้อมูลให้ตรงเวลาที่ตั้งไว้ ซึ่งถ้า ณ ช่วงเวลาที่กำหนดแหล่งข้อมูลไม่สามารถให้บริการในการสกัดข้อมูลได้ เราจะต้องทำการปรับเปลี่ยนตารางเวลาของการสกัดข้อมูลจากแหล่งข้อมูลเสียใหม่
- เราต้องทำให้แน่ใจว่าในการทำสำเนาข้อมูล (ในกรณีต่าง ๆ เช่น ต้องการเก็บข้อมูลไว้ในแต่ละดาต้ามาร์ท) จะมีการตรวจสอบความถูกต้องของข้อมูลที่ถูกสำเนาด้วย
- เราต้องทำให้แน่ใจว่าการสกัดข้อมูลจากเรคคอร์ดหนึ่งๆของฐานข้อมูลในแหล่งข้อมูลไปยังเรคคอร์ดในแฟ้มข้อมูลที่ถูกสกัดแล้ว (Extracted files) มีความสอดคล้องกัน

- ทำการสกัดข้อมูลให้ตรงเวลาที่ตั้งไว้ ซึ่งถ้า ณ ช่วงเวลาที่กำหนดแหล่งข้อมูลไม่สามารถให้บริการในการสกัดข้อมูลได้ เราจะต้องทำการปรับเปลี่ยนตารางเวลาของการสกัดข้อมูลจากแหล่งข้อมูลเสียใหม่
- เราต้องทำให้แน่ใจว่าในการทำสำเนาข้อมูล (ในกรณีต่าง ๆ เช่น ต้องการเก็บข้อมูลไว้ในแต่ละดาต้ามาร์ท) จะมีการตรวจสอบความถูกต้องของข้อมูลที่ถูกสำเนาด้วย
- เราต้องทำให้แน่ใจว่าการสกัดข้อมูลจากเรคคอร์ดหนึ่งๆของฐานข้อมูลในแหล่งข้อมูลไปยังเรคคอร์ดในแฟ้มข้อมูลที่ถูกสกัดแล้ว (Extracted files) มีความสอดคล้องกัน
- เราต้องสร้างกระบวนการในการแก้ไขข้อผิดพลาดของฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล และการทำความสะอาดข้อมูล
- เราจะต้องสร้างกระบวนการในการตรวจสอบการสร้าง “load image” ซึ่งเป็นแฟ้มข้อมูลสำหรับใช้ในการถ่ายโอนข้อมูลจาก staging area ไปยังฐานข้อมูลของคลังข้อมูล และจะต้องมีกระบวนการในการตรวจสอบการสร้างคีย์ต่างของแต่ละเรคคอร์ดใน dimension และ fact table ด้วย
- เราจะต้องทำการตรวจสอบกระบวนการจัดการกับข้อมูลที่มีความเปลี่ยนแปลงเกิดขึ้นอย่างช้า ๆ
- เราจะต้องมั่นใจได้ว่าเราจะสามารถทำการ “incremental load” ในแต่ละวันได้ตรงตามเวลาที่กำหนดไว้

การปรับแก้แบบจำลองข้อมูล

เมื่อเราทำการขยายขอบเขตของคลังข้อมูลจะทำให้แบบจำลองข้อมูลนั้นเปลี่ยนแปลงไป ถ้าการต่อเติมนั้นประกอบไปด้วยดาต้ามาร์ทใหม่ที่มีหัวข้อใหม่ ๆ จะทำให้แบบจำลองข้อมูลมี fact table และ dimension tables เพิ่มขึ้น และอาจมี aggregate tables เพิ่มขึ้นอีกด้วย และเมื่อแบบจำลองข้อมูลมีการเปลี่ยนแปลงหรือปรับแก้การจัดเก็บข้อมูลก็จะต้องมีการเปลี่ยนแปลงไปด้วย ดังนั้นในการเปลี่ยนแปลงหรือขยายขอบเขตของคลังข้อมูลจะทำให้เกิดสิ่งต่าง ๆ ดังต่อไปนี้

- การแก้ไขเมตาดาต้า (Revisions to metadata)
- การเปลี่ยนแปลงของการออกแบบทางกายภาพ (Changes to the physical design)
- การจองพื้นที่สำหรับจัดเก็บข้อมูลที่เพิ่มขึ้น (Additional storage allocation)
- การแก้ไขฟังก์ชันการทำงานของอีทีแอล (Revision to ETL functions)
- การเพิ่มคิวรีและรายงานที่กำหนดไว้ก่อนหน้า (Additional predefined queries and preformatted reports)
- การแก้ไขระบบ OLAP (Revisions to the OLAP system)
- การเพิ่มเติมระบบรักษาความปลอดภัย (Additions to the security system)
- การเพิ่มเติมระบบสำรองและกู้คืนข้อมูล (Additions to the backup and recovery system)

การเพิ่มประสิทธิภาพการเข้าถึง/ส่งผ่านข้อมูล

เมื่อผู้ใช้ทำการใช้คลังข้อมูลนานขึ้นจะทำให้ผู้ใช้เหล่านั้นทำการสร้างคิวรีที่มีความซ้ำซ้อนมากขึ้น และ ประกอบกับในปัจจุบันมีการพัฒนาระบบที่ใช้สำหรับเข้าถึงข้อมูลหรือส่งผ่านข้อมูลไปยังผู้ใช้ที่มีประสิทธิภาพเพิ่มมากขึ้นเรื่อย ๆ ทั้งสองสิ่งนี้จะทำให้เราต้องพิจารณาถึงการปรับเปลี่ยนเครื่องมือที่ใช้สำหรับเข้าถึงหรือส่งผ่านข้อมูลไปยังผู้ใช้ด้วย ซึ่งในการปรับเปลี่ยนเครื่องมือสำหรับส่งผ่านข้อมูลจะมีแนวทางในการปฏิบัติดังนี้

- ก่อนที่จะทำการปรับเปลี่ยนเครื่องมือ เราจะต้องแน่ใจว่าเครื่องมือใหม่ที่จะใช้นั้นสามารถทำงานร่วมกับส่วนประกอบอื่น ๆ ของคลังข้อมูลได้
- ถ้าเราทำการติดตั้งเครื่องมือใหม่แล้วเราจะต้องค่อยๆบอกกล่าวผู้ใช้ค่อย ๆ ทำการเปลี่ยนเครื่องมือที่ใช้สำหรับเข้าถึงข้อมูล
- เมื่อทำการเปลี่ยนแปลงเครื่องมือ เราจะต้องมั่นใจได้ว่าเราจะสามารถเรียกดูหรือเรียกใช้เมตาดาต้าจากเครื่องมือนั้น ๆ ได้
- เราจะต้องทำการกำหนดตารางเวลาสำหรับอบรมการใช้งานเครื่องมือชิ้นใหม่ที่จะใช้ด้วย

การปรับแต่งสิ่งต่าง ๆ ในคลังข้อมูล

ในการปรับแก้คลังข้อมูลจะเน้นที่การเพิ่มประสิทธิภาพให้กับคลังข้อมูลจะมีแนวปฏิบัติดังต่อไปนี้

- ควรกำหนดช่วงเวลาในการตรวจสอบการใช้งานดัชนีต่าง ๆ และทำการลบดัชนีที่ไม่ถูกใช้งานทิ้ง
- ควรมีการเฝ้าดูประสิทธิภาพของการประมวลผลคิวรีในแต่ละวัน การตรวจสอบคิวรีที่มีการประมวลผลนาน ๆ
- ควรมีการวิเคราะห์การทำงานของการประมวลผลคิวรีที่มีการกำหนดไว้ก่อนหน้าแล้ว

ถึงแม้ว่าเราจะมีตารางเวลาที่แน่นอนสำหรับการปรับแต่งสิ่งต่างๆในคลังข้อมูล แต่เราสามารถปรับเปลี่ยนเวลาที่กำหนดไว้ได้ ถ้ามีปัญหาเกิดขึ้นหรืออาจจะมีเสียงบ่นต่าง ๆ จากผู้ใช้ที่เกี่ยวข้องกับคลังข้อมูล ดังนั้นทีมผู้ดูแลระบบอาจจำเป็นต้องเผื่อเวลาไว้สำหรับปรับแก้สิ่งต่าง ๆในคลังข้อมูลอย่างเร่งด่วนได้

SECTION 8

คำถามท้ายบท

1. จงอธิบายถึงขั้นตอนหลักในการปรับใช้คลังข้อมูลว่ามีอะไรบ้าง แต่ละขั้นตอนมีการทำงานอย่างไร
2. จงอธิบายขั้นตอนการทำงาน “User acceptance procedure” ว่ามีการทำงานอย่างไร ทำไมถึงสำคัญ
3. จงอธิบายถึงข้อดีของการใช้ต้นแบบในการปรับใช้คลังข้อมูล
4. ต้นแบบชนิด “proof-of-concept” คืออะไร มีประโยชน์อย่างไร จงอธิบาย
5. จงอธิบายถึงเงื่อนไข และปัจจัยที่ต้องพิจารณาสำหรับการสำรองและการกู้คืนข้อมูล
6. การจัดเก็บข้อมูลเชิงสถิติจะเก็บไว้ใช้ทำอะไร และสามารถจัดเก็บได้ด้วยวิธีใดบ้าง
7. เราจะสามารถจัดการกับข้อมูลที่เพิ่มขึ้นได้อย่างไร
8. เราสามารถจัดการกับคลังข้อมูลในแง่มุมใดได้บ้าง

บรรณานุกรม

Paulraj Ponniah, ***Data Warehousing Fundamentals for IT Professionals (2nd ed.)***, John Wiley & Sons, Inc., 2010.

Fon Silvers, ***Building and Maintaining a Data Warehouse***, CRC Press, 2008.

William H. Inmon, ***Building the Data Warehouse***, Fourth Edition, Wiley Publishing, Inc., 2005

Ralph Kimball and Joe Caserta, ***The Data Warehouse ETL Toolkit—Practical techniques for extracting, cleaning, conforming, and delivering data***, Wiley Publishing, Inc., 2004

Vincent Rainardi, ***Building a Data Warehouse—with examples in SQL server***, Apress, 2008