บทที่ 8 การสกัด การเปลี่ยนแปลง และการถ่ายโอนข้อมูล

วัตถุประสงค์

- ศึกษาเกี่ยวกับการสกัด การเปลี่ยนแปลง และถ่ายโอนข้อมูล รวมถึงสถาปัตยกรรมของอีทีแอล
- ศึกษาเกี่ยวกับลักษณะของข้อมูลที่มักพบบ่อยในระบบการดำเนินงาน
- ศึกษาเกี่ยวกับการสกัดข้อมูลด้วยวิธีการต่างๆ
- ศึกษาเกี่ยวกับฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล
- ศึกษาเกี่ยวกับการถ่ายโอนข้อมูลเข้าสู่คลังข้อมูลในลักษณะต่างๆ

เนื้อหาของบทเรียน

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

กิจกรรมการเรียนการสอน

- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารประกอบการสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารประกอบการสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

การวัดและประเมินผล

- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบฝึกหัดย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

จากบทก่อนๆหน้า เราจะทราบว่าการทำงานหลักของคลังข้อมูลจะสามารถถูกแบ่งได้เป็น 3 ส่วนใหญ่ๆ ด้วยกัน คือ (1) การเก็บรวบรวมและได้มาซึ่งข้อมูล (2) การจัดเก็บข้อมูล และ (3) การเข้าถึง/ส่งต่อข้อมูลสารสนเทศไปยังผู้ใช้ 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) ที่ได้จากขั้นตอนแรกตามกรรมวิธีต่างๆหรือตามฟังก์ชันการประมวลผลข้อมูลเบื้องต้น (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 เรา*จะต้องออกแบบการสกัดข้อมูลให้มีการใช้งานหรือรบกวนการทำงานของแหล่งข้อมูล/ระบบการดำเนินงานให้น้อยที่สุดเท่าที่จะเป็นไปได้* ซึ่งจะช่วยให้การทำงานของระบบการดำเนินงานนั้นยังคงสามารถดำเนินไปได้อย่างปกติ

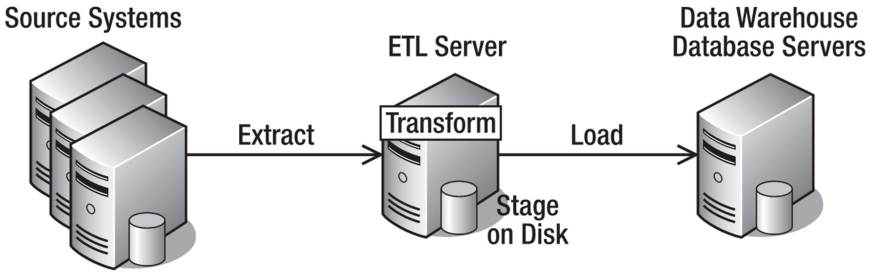
ในส่วนของฟังก์ชันที่สองซึ่งก็คือ การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลจะเริ่มการทำงานก็ต่อเมื่อข้อมูลที่ได้รับมาจากขั้นตอนการสกัดข้อมูลนั้นยังไม่มีคุณภาพหรือยังไม่เป็นมาตราฐานเพียงพอซึ่งข้อมูลที่สกัดได้ ซึ่งการด้อยคุณภาพของข้อมูลอาจมาจากข้อมูลที่มาจากหลายแหล่งข้อมูลที่มีความแตกต่างกันทั้งในแง่มุมของเทคโนโลยีหรือแพลตฟอร์มการคำนวณ ดังนั้น ฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลจะมีหน้าที่ในการทำความสะอาดข้อมูล รวมข้อมูลเข้าด้วยกัน ทำข้อมูลให้เป็นมาตราฐาน และอื่นๆ ซึ่งจากฟังก์ชันการทำงานดังกล่าว เราสามารถกล่าวได้ว่า ฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลเป็นฟังก์ชันที่ทำการประมวลผลข้อมูลเบื้องต้นก่อนที่จะนำข้อมูลเหล่านั้นไปเก็บไว้ในคลังข้อมูลต่อไป และหลังจากทำการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลแล้ว เราจะทำการถ่ายโอนข้อมูลเข้าสู่คลังข้อมูล ซึ่งฟังก์ชันการทำงานนี้เป็นอีกหนึ่งฟังก์ชันสุดท้ายของอีทีแอลที่ทำหน้าที่ในการถ่ายโอนข้อมูลที่ได้จากแหล่งข้อมูลและผ่านการประมวลผลแล้วไปจัดเก็บไว้ในคลังข้อมูลต่อไป ซึ่งในการถ่ายโอนข้อมูลเราจะต้องหาช่วงเวลาที่เหมาะสม โดยจะต้องไม่กระทบต่อการทำงานของระบบการดำเนินงานและระบบคลังข้อมูลมากนัก

จากที่กล่าวข้างต้น เราสามารถสรุปได้ว่า “อีทีแอล” นั้นเป็นขั้นตอนการโหลด/ถ่ายโอนข้อมูลจากระบบการดำเนินงานเข้าสู่คลังข้อมูล โดยการทำงานจะเริ่มจากการสกัดข้อมูลที่ผู้ใช้ต้องการจากแหล่งข้อมูล จากนั้นทำการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลให้มีคุณภาพหรือเป็นมาตราฐาน แล้วจึงทำการถ่ายโอนข้อมูลเข้าสู่คลังข้อมูล ซึ่งจากการทำงานทั้งหมดของคลังข้อมูล ฟังก์ชัน“อีทีแอล” จะเป็นขั้นตอนที่สำคัญมากในการสร้างคลังข้อมูลและเป็นขั้นตอนที่ใช้เวลามากที่สุดในการสร้างคลังข้อมูล ดังนั้นในการสร้างคลังข้อมูล เราจำเป็นที่จะต้องออกแบบฟังก์ชันอีทีแอลให้มีประสิทธิภาพมากที่สุด ซึ่งจะทำให้คลังข้อมูลที่เราสร้างขึ้นมีประสิทธิภาพที่ดีในการทำงานตามไปด้วย

สถาปัตยกรรมของอีทีแอล

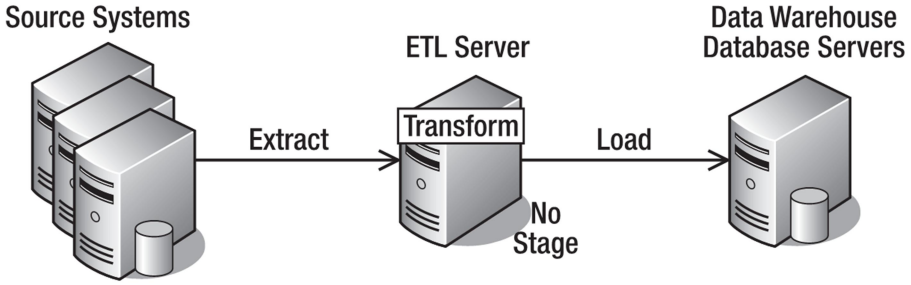
ขั้นตอนพื้นฐานของฟังก์ชัน “อีทีแอล” จะเริ่มจากการสกัดข้อมูลที่ต้องการเพียงบางส่วนจากแหล่งข้อมูลต่างๆแล้วจึงทำการประมวลผลกับข้อมูลนั้นๆ อาทิเช่น การรวบรวมข้อมูลเข้าด้วยกัน การแยกข้อมูลออกจากกัน การทำความสะอาดข้อมูล การปรับรูปแบบของข้อมูล และอื่นๆ โดยหลังจากทำการประมวลผลเบื้องต้นข้อมูลแล้วจะทำการถ่ายโอนข้อมูลที่ถูกประมวลผลเหล่านั้นเข้าสู่คลังข้อมูล ซึ่งจากการทำงานดังกล่าว เราควรที่จะต้องศึกษาเกี่ยวกับสถาปัตยกรรมที่เกี่ยวข้องกับฟังก์ชัน “อีทีแอล” ซึ่งจะมีอยู่ 2 ประเภทคือ

1. **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** กล่าวคือ การสกัดข้อมูลที่ต้องการเพียงบางส่วนจากแหล่งข้อมูล แล้วทำการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลในหน่วยความจำ (ดังแสดงในรูปที่ 8-2) แล้วจึงทำการถ่ายโอนข้อมูลไปยังคลังข้อมูลต่อไป



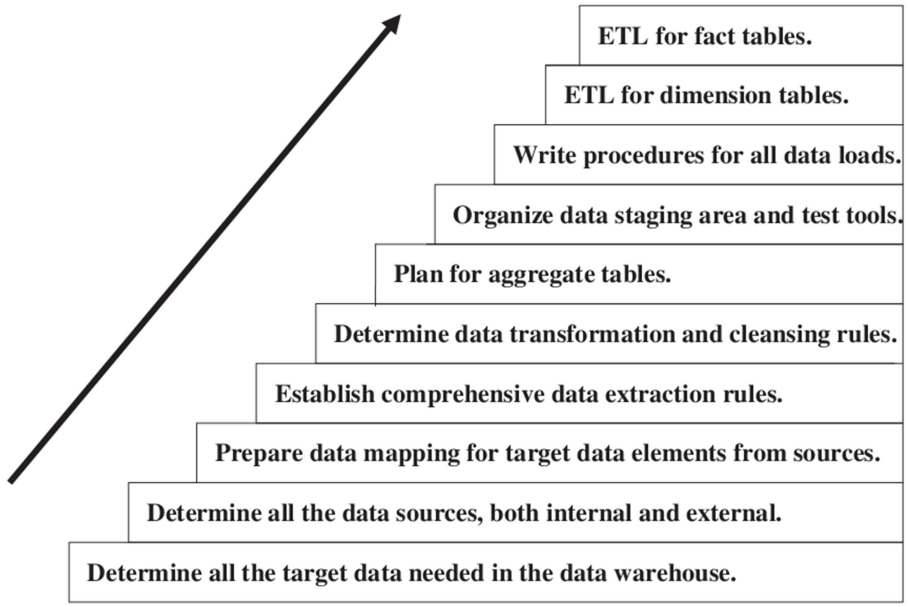
รูปที่ 8-2 การสกัดข้อมูลโดยการใช้หน่วยความจำ

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

ขั้นตอนการทำงานของอีทีแอล

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

- การวางแผนสำหรับการรวมรวมข้อมูลเข้าสู่ fact table
- การกำหนดกฎสำหรับการเปลี่ยนแปลง/เปลี่ยนรูปและการทำความสะอาด
- การสร้างกฎในการสกัดข้อมูล
- การเตรียมการเชื่อมโยงข้อมูลจากแหล่งข้อมูลเข้าสู่ dimension หรือ fact table
- การรวบรวมข้อมูลจากแหล่งข้อมูลต่างๆ ทั้งแหล่งข้อมูลภายในและภายนอก
- การกำหนดข้อมูลทั้งหมดที่ต้องการเก็บไว้ในคลังข้อมูล
- การรวมโครงสร้างข้อมูลจากหลายแหล่งข้อมูลไปเป็นข้อมูลเพียง row เดียวใน dimension หรือ fact table ที่ถูกเก็บไว้ในคลังข้อมูล
- การแยกโครงสร้างข้อมูลหนึ่งๆไปเป็นข้อมูลที่มีหลายโครงสร้างเพื่อสร้างเป็นข้อมูลหลายๆ row ที่ถูกเก็บไว้ในคลังข้อมูล
- การอ่านข้อมูลจากพจนานุกรมข้อมูลที่ถูกเก็บไว้ในแหล่งข้อมูล
- การอ่านข้อมูลจากหลายๆแหล่งข้อมูล 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

การสกัดข้อมูล

ในการสกัดข้อมูลจากระบบการดำเนินงาน เราจะต้องให้ความใส่ใจกับระบบการดำเนินงานของธุรกิจที่ซึ่งในปัจจุบันระบบเหล่านี้กระจายตัวอยู่ตามที่ต่างๆทั่วโลกและข้อมูลในระบบการดำเนินงานอาจเกิดความเปลี่ยนแปลงเกิดขึ้นได้ตลอด อาทิเช่น การเพิ่ม ลบหรืออัพเดทข้อมูล เป็นต้น ดังนั้นเมื่อเราทำการออกแบบการสกัดข้อมูลเราจะต้องให้ความสนใจกับสถานะของข้อมูลทั้งในระบบการดำเนินงาน และสถานะของคลังข้อมูลด้วย ซึ่งในตอนเริ่มต้นหลังจากทำการสร้างฐานข้อมูลและตารางต่างๆตามแบบจำลองมิติต่างๆแล้ว คลังข้อมูลจะยังไม่มีข้อมูลอยู่เลย โดยเราจะต้องทำการสกัดข้อมูลที่ต้องการทั้งหมดจากระบบการดำเนินงานเพื่อทำการถ่ายโอนข้อมูลเหล่านั้นไปยังคลังข้อมูล ซึ่งวิธีการถ่ายโอนข้อมูลจากการสกัดข้อมูลเข้าสู่คลังข้อมูลครั้งแรกจะเรียกว่า “Full load” หรือ “Initial load” ต่อมาเมื่อระบบการดำเนินงานมีข้อมูลเพิ่มขึ้นหรือมีการเปลี่ยนแปลง/อัพเดทข้อมูล เราจะต้องทำการสกัดข้อมูลที่ถูกเพิ่มขึ้นหรืออัพเดทเหล่านั้น เข้าสู่คลังข้อมูลในครั้งต่อๆไป ซึ่งวิธีการนี้เราจะเรียกว่า “(Ongoing) Incremental load” ซึ่งจากวิธีการดังกล่าว เราจะต้องทำการตั้งเวลาหรือกำหนดช่วงเวลาสำหรับการสกัดข้อมูลและทำการถ่ายโอนข้อมูล อาทิเช่น การสกัดข้อมูลสัปดาห์ละครั้ง เป็นต้น ซึ่งจากวิธีการดังกล่าว เราจะต้องทำการเสาะหาข้อมูลที่ถูกเพิ่มขึ้นหรือมีการเปลี่ยนแปลงในช่วงหนึ่งสัปดาห์ แล้วทำการสกัดข้อมูลที่ต้องการจากข้อมูลเหล่านั้น แล้วจึงถ่ายโอนไปยังคลังข้อมูลต่อไป

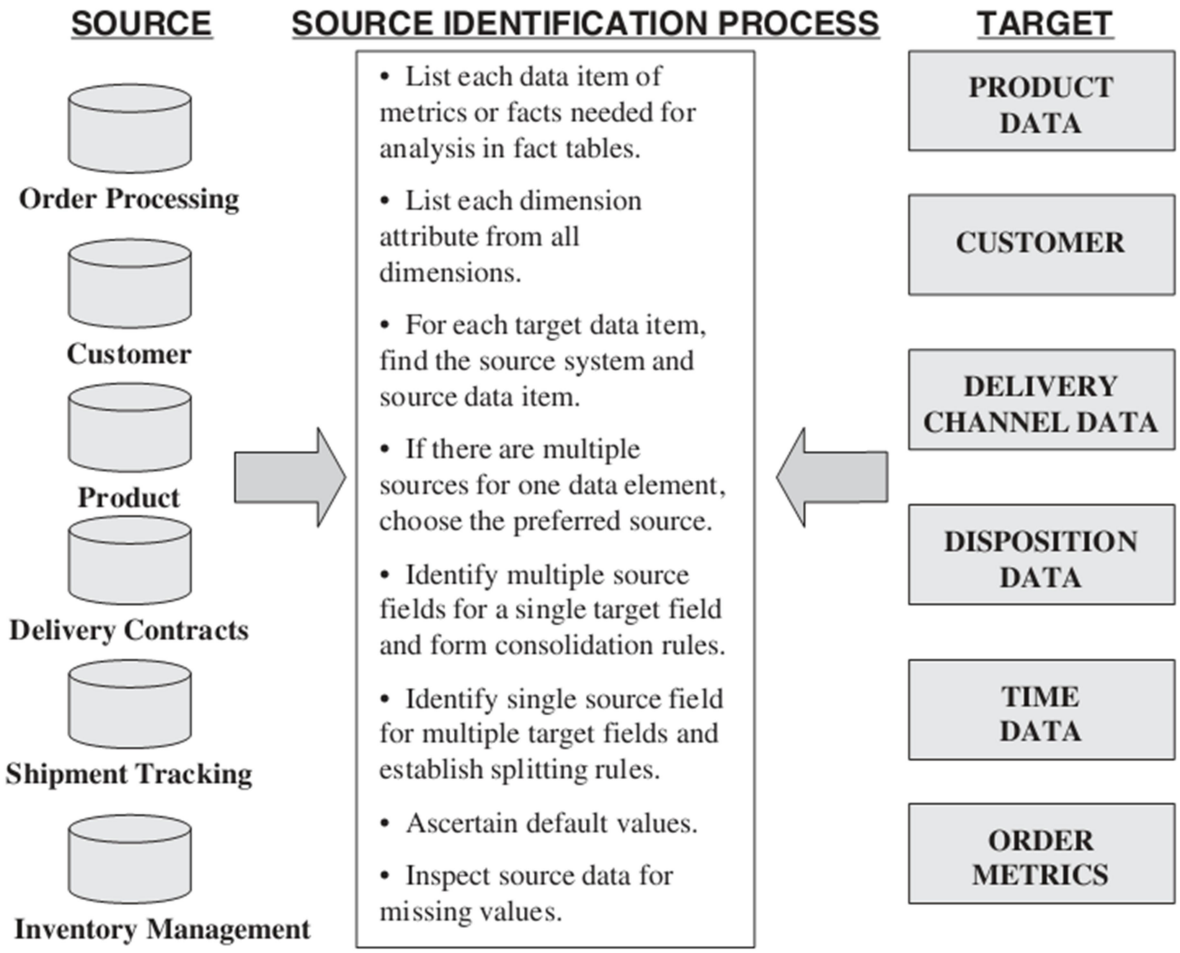
จากการถ่ายโอนข้อมูลทั้งสองวิธีข้างต้น จะทำให้การสกัดข้อมูลมีขั้นตอนการสกัดข้อมูลที่มีความยุ่งยากและซับซ้อนเพิ่มขึ้น ดังนั้นในการออกแบบฟังก์ชันการทำงานของคลังข้อมูล เราจะต้องพิจารณาถึงขั้นตอนการทำงานของวิธีการถ่ายโอนข้อมูลทั้งสอง และจำเป็นต้องพิจารณาขั้นตอนและปัจจัยต่างๆดังต่อไปนี้

- **Source Identification**—การระบุแหล่งข้อมูลและโครงสร้างของแหล่งข้อมูล ว่ามีข้อมูลที่เราต้องการหรือสนใจอยู่ที่ใดบ้าง อาทิเช่น 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 ซึ่งในขณะที่ขั้นตอนแรกทำงานขั้นตอนอื่นๆจะต้องรอเป็นลำดับ
- **Exception handling**—กำหนดวิธีในการจัดการกับเรคคอร์ดที่ไม่สามารถสกัดได้ หรือเหตุการณ์ผิดปกติที่อาจเกิดขึ้น

การระบุแหล่งข้อมูลที่สนใจ (Source Identification)

การระบุแหล่งข้อมูลจะเป็นการระบุถึงแหล่งข้อมูลอินพุตที่ต้องการว่าข้อมูลเหล่านั้นปรากฏหรือถูกจัดเก็บอยู่ในแหล่งข้อมูลใดบ้าง รวมถึงการทดสอบและการทวนสอบแหล่งข้อมูลที่ต้องทำการติดต่อว่ามีข้อมูลที่จำเป็นต่อการสร้างคลังข้อมูลหรือไม่ ซึ่งการระบุแหล่งข้อมูลนั้นมักจะนิยมใช้ในระบบที่มีแบ่งส่วนการเก็บข้อมูลออกเป็นส่วนๆตามดาต้ามาร์ทต่างๆ (ดังแสดงในรูปที่ 8-4) ซึ่งจะประกอบไปด้วยขั้นตอนต่างๆดังนี้

- ทำการกำหนดข้อมูลที่เป็นตัวชี้วัดหรือข้อเท็จจริงที่ต้องการสำหรับวิเคราะห์ข้อมูลใน fact table
- ทำการกำหนดข้อมูลแต่ละแอทริบิวที่เกี่ยวกับข้องทุกๆ dimension table
- ทำการหาแหล่งข้อมูลกับข้อมูลที่จะเก็บใน dimension และ fact table
- ในกรณีที่ข้อมูลหนึ่งๆ ถูกเก็บอยู่ในหลายแหล่งข้อมูล ต้องทำการเลือกแหล่งข้อมูลหนึ่งแหล่งที่จะทำการสกัดข้อมูล
- ทำการระบุว่าข้อมูลฟิลด์หนึ่งๆ ที่ต้องการเก็บไว้ในคลังข้อมูลนั้น มาจากหลายฟิลด์ของแหล่งข้อมูลหรือไม่ จากนั้นทำการสร้างกฎการรวมข้อมูล (Consolidation rules)
- ทำการระบุว่าข้อมูลหลายๆฟิลด์ที่ต้องการเก็บไว้ในคลังข้อมูลนั้นมาจากข้อมูลเพียงฟิลด์เดียวของแหล่งข้อมูลหรือไม่ จากนั้นทำการสร้างกฎการแยกข้อมูล (Splitting rules)
- ทำการกำหนดค่า default value สำหรับข้อมูลแต่ละ field ที่ต้องการเก็บไว้ในคลังข้อมูล
- ทำการตรวจสอบการขาดขายไปของข้อมูล



รูปที่ 8-4 ขั้นตอนการระบุแหล่งข้อมูลโดยละเอียด

เทคนิคในการสกัดข้อมูล

ก่อนที่จะทำการสกัดข้อมูลเราต้องทำความเข้าใจเกี่ยวกับธรรมชาติของข้อมูลที่ถูกเก็บอยู่ในระบบดำเนินการรวมถึงโครงสร้างการจัดเก็บข้อมูลด้วยเพื่อที่จะสามารถสกัดข้อมูลได้อย่างมีประสิทธิภาพ ซึ่งโดยปกติแล้ว ข้อมูลที่ถูกเก็บอยู่ในระบบทั่วๆไปจะมีการเปลี่ยนแปลงไปตามกาลเวลา ตัวอย่างเช่น ลูกค้าของบริษัทแห่งหนึ่งต้องย้ายที่อยู่จากรัฐนิวยอร์กไปยังแคลิฟอร์เนีย ระบบสำหรับจัดเก็บข้อมูลลูกค้าจะต้องทำการอัพเดทข้อมูลที่อยู่ของลูกค้า ถ้าโครงสร้างข้อมูลของระบบไม่มีการเก็บประวัติก่อนหน้า ระบบจะลบข้อมูลที่อยู่ในรัฐนิวยอร์กออกแล้วแทนที่ด้วยข้อมูลที่อยู่ในรัฐแคลิฟอร์เนีย ซึ่งการเก็บข้อมูลลักษณะนี้จะส่งผลกระทบโดยตรงต่อข้อมูลในคลังข้อมูล อาทิเช่น ถ้าเราต้องการวิเคราะห์ยอดขายในรัฐหนึ่งๆ 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

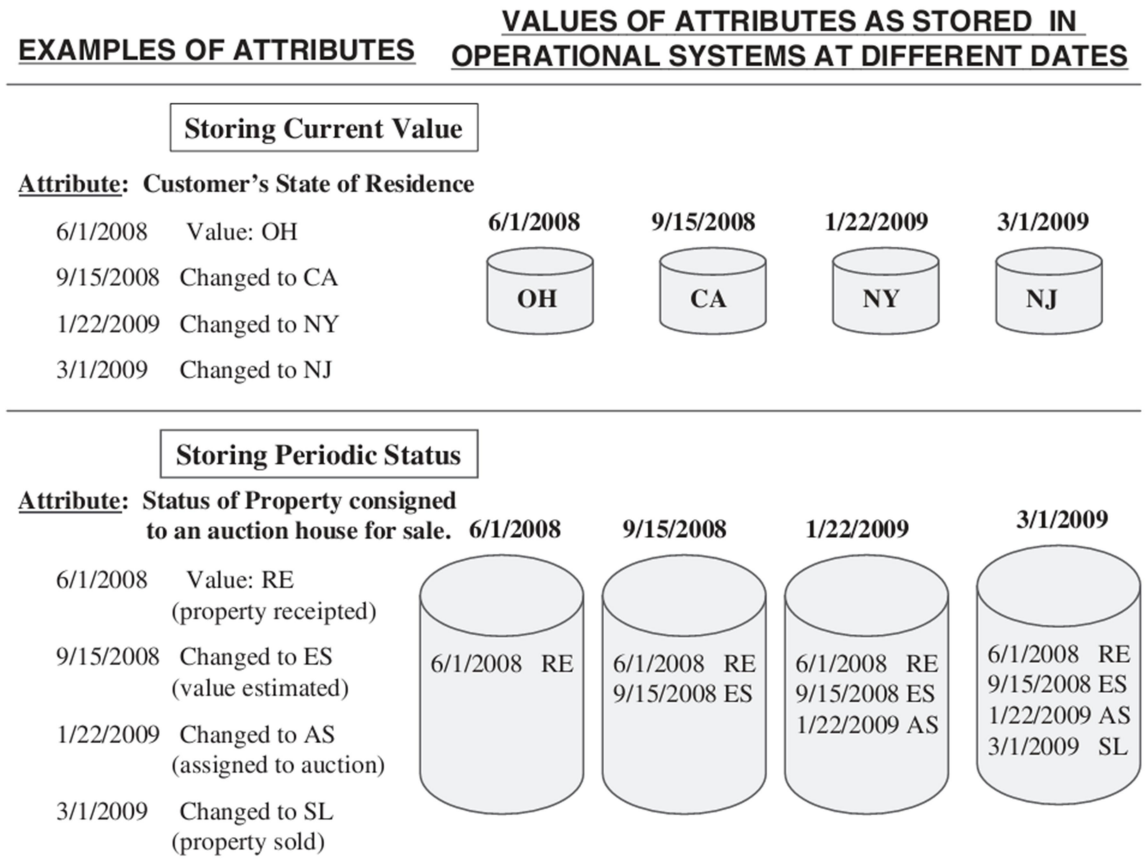
ข้อมูล ณ ปัจจุบัน (Current Value)

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

ข้อมูลที่มีการเปลี่ยนแปลงไปตามกาลเวลา (Periodic Status)

ข้อมูลลักษณะนี้จะมีการเก็บสถานะของข้อมูลเมื่อมีการเปลี่ยนแปลงเกิดขึ้นและรวมถึงการเพิ่มข้อมูลเข้าสู่ฐานข้อมูลในแต่ละครั้ง โดยที่ในแต่ละครั้งที่มีการเปลี่ยนแปลงเกิดขึ้นเราจะต้องทำการเก็บเวลาที่ใช้ในการอ้างอิงถึงการเปลี่ยนแปลงหรือการเพิ่มข้อมูลด้วย ซึ่งเวลาที่ถูกจัดเก็บไว้จะสามารถบอกได้ถึงลำดับการเกิดขึ้นข้อมูลในฐานข้อมูลและสถานะปัจจุบันของข้อมูลได้อีกด้วย โดยในการเก็บข้อมูลที่มีการเปลี่ยนแปลงไปตามการเวลานั้นจะเป็นการเก็บข้อมูลเหตุการณ์ที่เกิดขึ้นโดยจะมีเวลาที่เกี่ยวข้องเกิดขึ้นด้วย โดยที่การเก็บสถานะต่างๆของข้อมูลนั้นจะเก็บแยกไว้ในอีกฟิลด์หรือแอทริบิวหนึ่งๆ โดยการเก็บข้อมูลลักษณะนี้จะช่วยให้การสกัดข้อมูลที่ต้องการการวิเคราะห์หาสถิติย้อนหลังของข้อมูลสามารถทำงานได้ง่ายขึ้น

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



รูปที่ 8-5 ชนิดของข้อมูลในระบบการดำเนินงาน

"As is" หรือ static data คือ การดักจับ/เข้าถึงข้อมูล ณ เวลาช่วงเวลาหนึ่งๆที่กำหนด ซึ่งการทำงานของ "as is" จะคล้ายกับการทำ snapshot กับข้อมูลที่เกี่ยวข้อง ณ ช่วงเวลาหนึ่งๆโดยข้อมูลที่เราจะทำการดักจับนั้นจะเป็นข้อมูล ณ ปัจจุบัน หรือเป็นข้อมูลชั่วคราวที่สามารถเปลี่ยนแปลงได้ นอกจากนั้นยังทำการดักจับข้อมูลที่มีการเปลี่ยนแปลงไปตามกาลเวลาที่จะดักจับข้อมูลที่ถูกเพิ่มเข้าสู่ระบบและข้อมูลที่ถูกเปลี่ยนแปลงในช่วงเวลาที่กำหนด วิธีการสกัดข้อมูลแบบ "as is" มักถูกใช้ในการถ่ายโอนข้อมูลครั้งแรก (initial load) หรือการถ่ายโอนข้อมูลทั้งหมดจากระบบการดำเนินงานเข้าสู่คลังข้อมูล (full refresh) ตัวอย่างเช่น เมื่อเวลาผ่านไป มีการเปลี่ยนแปลงชื่อสินค้าในระบบการดำเนินงาน ดังนั้นถ้าเราทำการถ่ายโอนข้อมูลชื่อสินค้าเข้าสู่คลังข้อมูลทั้งหมดอีกรอบหนึ่ง โดยไม่เลือกว่าข้อมูลใดเคยถูกถ่ายโอนข้อมูลเข้าสู่คลังข้อมูลแล้ว จะช่วยให้การทำงานและการจัดการกับการทำงานสามารถดำเนินการได้โดยง่าย ดังนั้น เราจะสามารถใช้ "as is" ในการสกัดข้อมูลได้ก็ต่อเมื่อเราต้องการถ่ายโอนข้อมูลทั้งหมดเข้าสู่คลังข้อมูลอีกครั้งหนึ่ง

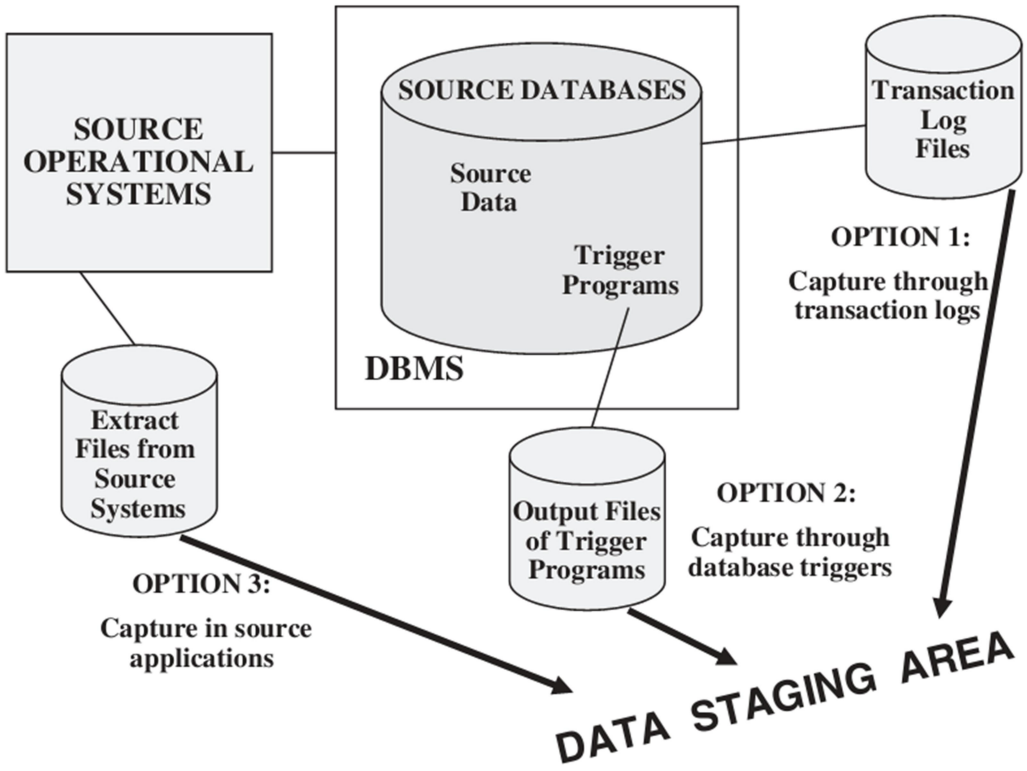
"Data of revisions" เป็นการสกัดข้อมูลแบบ incremental ซึ่งเป็นการสกัดข้อมูลที่มีการแก้ไขนับตั้งแต่การสกัดข้อมูลครั้งล่าสุด ถ้าข้อมูลจากแหล่งข้อมูลเป็นแบบข้อมูลชั่วคราว (เป็นข้อมูลไม่มีการเก็บสถานะของการเปลี่ยนแปลงของข้อมูล) การสกัดข้อมูลจะสามารถทำได้ยาก แต่ในทางกลับกัน ถ้าข้อมูลเป็นข้อมูลที่มีการเปลี่ยนแปลงไปตามกาลเวลา (เก็บสถานะของการเปลี่ยนแปลงด้วย) การสกัดข้อมูลจะสามารถ

ทำได้ง่าย 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)



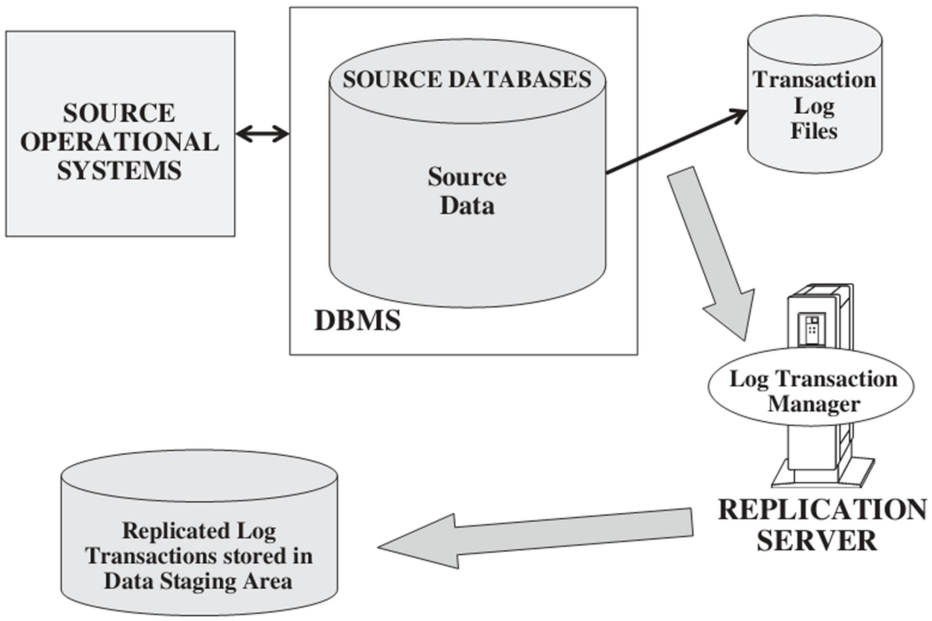
รูปที่ 8-6 ทางเลือกในการสกัดข้อมูลแบบทันที

การสกัดข้อมูลโดยใช้ล็อกไฟล์ของฐานข้อมูล (Capture through Transaction Logs)

วิธีการนี้จะเป็นการสกัดข้อมูลโดยใช้ล็อกไฟล์ที่ถูกจัดเก็บไว้ในระบบจัดการฐานข้อมูล (DBMS) ซึ่งโดยปกติของระบบจัดการฐานข้อมูลจะมีการเขียนข้อมูลลงล็อกไฟล์อยู่แล้วเมื่อมีการเพิ่ม ลบหรืออัพเดทข้อมูลเรคคอร์ดหนึ่งๆในฐานข้อมูล ซึ่งการจัดเก็บข้อมูลลงในล็อกไฟล์ของฐานข้อมูลจะทำเพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นกับการทำงานของระบบจัดการฐานข้อมูล ดังนั้น เมื่อระบบจัดการฐานข้อมูลมีการจัดเก็บข้อมูลลงในล็อกไฟล์จะทำให้เราสามารถทำการสกัดข้อมูลโดยทำการอ่านล็อกไฟล์เพื่อที่จะทราบถึงแต่ละรายการที่มีการกระทำกับฐานข้อมูล โดยการอ่านล็อกไฟล์จากระบบจัดการฐานข้อมูลนั้นจะไม่ทำให้การทำงานของระบบ

การดำเนินงานเพิ่มขึ้นเลย เนื่องจากล็อกไฟล์นั้นเป็นส่วนหนึ่งที่ต้องมีการจัดเก็บไว้ในฐานข้อมูลอยู่แล้ว แต่ถ้าเราใช้การจัดเก็บข้อมูลแบบอื่นๆ อาทิเช่น การทำดัชนี (index) หรือการใช้แฟ้มข้อมูลจะเป็นการเพิ่มการทำงานให้กับระบบการดำเนินงาน เนื่องจากต้องการสร้างดัชนีและทำการเขียนข้อมูลลงแฟ้มข้อมูลเอง

ในการสกัดข้อมูลโดยทำการอ่านล็อกไฟล์สามารถดำเนินการได้กับหลายแหล่งข้อมูลที่อาจเป็นระบบเป็นแบบกระจาย (Distributed system) ซึ่งโดยธรรมชาติของระบบแบบกระจายจะมีการทำสำเนาข้อมูล (Data replication) เพื่อให้แต่ละส่วนของระบบสามารถใช้ข้อมูลได้ทั้งหมด ดังนั้นในการสกัดข้อมูล เราอาจใช้เทคโนโลยีการทำสำเนาข้อมูลเพื่อช่วยในการหาความเปลี่ยนแปลงของข้อมูลที่ถูกจัดเก็บอยู่ในแหล่งข้อมูลแบบกระจาย ดังแสดงในรูปที่ 8-7



รูปที่ 8-7 การสกัดข้อมูลโดยใช้การทำเสานำข้อมูลร่วมกับการใช้ล็อกไฟล์

จากรูปที่ 8-7 เป็นการทำงานของการสกัดข้อมูลจากระบบกระจายโดยใช้การทำสำเนาข้อมูลต่างๆ ระบบและฐานข้อมูลอาจมีด้วยกันหลายแห่ง ซึ่งอาจทำให้มีล็อกไฟล์หลายไฟล์ตามไปด้วย ดังนั้นในการสกัดข้อมูลโดยใช้การทำสำเนาข้อมูลเป็นส่วนหนึ่งของขั้นตอนการทำงานจะมีขั้นตอนการทำงานดังต่อไปนี้

- ระบุตารางจากแหล่งข้อมูลที่ใช้ที่เก็บข้อมูลที่เราต้องการสกัด (Identify the source system database table)
- ระบุและกำหนดไฟล์เป้าหมายของ staging area (Identify and define target files in the staging area)
- สร้างการเชื่อมโยงระหว่างตารางข้อมูลและไฟล์เป้าหมาย (Create mapping between the source table and target files)
- ระบุโหมดการทำซ้ำ (Define the replication mode)

- กำหนดตารางเวลาสำหรับกระบวนการทำซ้ำ (Schedule the replication process)
- ทำการหาความแตกต่างของข้อมูลที่ต้องการจากล็อกไฟล์ (Capture the change from the transaction logs)
- ถ่ายโอนข้อมูลที่สามารถสกัดได้จากล็อกไฟล์ไปยังแฟ้มเป้าหมายใน staging area (Transfer captured data from logs to target files)
- ตรวจสอบความถูกต้องของการถ่ายโอนข้อมูล (Verify transfer of data changes)
- ทำการเก็บผลของการทำสำเนาข้อมูลไว้เป็นเมตาดาต้า (In metadata, document the outcome of replication)
- ดูแลรักษาคำจำกัดความของแหล่งข้อมูล ไฟล์เป้าหมาย และการเชื่อมโยง (Maintain definitions of sources, targets and mappings)

การสกัดข้อมูลโดยใช้ดาต้าเบสทริกเกอร์ (Capture through Database Triggers)

วิธีการนี้เป็นการสกัดข้อมูลจากระบบการดำเนินงานที่มีการใช้ฐานข้อมูลโดยใช้ทริกเกอร์ (Trigger) โดยที่ทริกเกอร์ คือ วิธีพิเศษในการจัดเก็บข้อมูล (หรือเราจะเรียกว่าโปรแกรมก็ได้) ซึ่งโดยปกติแล้วโปรแกรมทริกเกอร์จะถูกเก็บอยู่ในฐานข้อมูลและจะถูกเรียกใช้งานก็ต่อเมื่อมีเหตุการณ์ที่เรากำหนดไว้ล่วงหน้าแล้วเกิดขึ้น ในการสร้างโปรแกรมทริกเกอร์นั้น เราสามารถสร้างโปรแกรมทริกเกอร์สำหรับทุกๆเหตุการณ์ที่เกี่ยวข้องกับข้อมูลที่เราต้องการสกัดได้ โดยผลลัพธ์ที่ได้จากโปรแกรมทริกเกอร์จะถูกเขียนอยู่ในไฟล์ที่ถูกแยกไว้เพื่อนำไปใช้ในการสกัดข้อมูลเข้าสู่คลังข้อมูล ตัวอย่างเช่น ถ้าเราต้องการสกัดข้อมูลที่เป็นความเปลี่ยนแปลงของข้อมูลในตารางลูกค้า เราสามารถเขียนโปรแกรมทริกเกอร์เพื่อตรวจจับการเปลี่ยนแปลงทุกๆครั้งที่เกิดขึ้นกับข้อมูลลูกค้า อาทิเช่น การเพิ่ม ลบ และอัพเดทข้อมูลต่างๆ เป็นต้น โดยข้อมูลที่สกัดได้จากการใช้โปรแกรมทริกเกอร์จะมีความน่าเชื่อถือ ซึ่งเราสามารถทราบถึงข้อมูลก่อนและหลังการเปลี่ยนแปลง แต่อย่างไรก็ดี การสร้างและการดูแลโปรแกรมทริกเกอร์จะทำให้แหล่งข้อมูลหรือระบบดำเนินการมีการทำงานเพิ่มขึ้นจากเดิม ดังนั้น ในพิจารณาการใช้การสกัดข้อมูลโดยใช้โปรแกรมทริกเกอร์เราจะต้องพิจาณาถึงการทำงานที่เพิ่มขึ้นมาด้วย

การสกัดข้อมูลโดยทำการสร้างแอพพลิเคชันไว้ที่แหล่งข้อมูล (Capture in Source Applications)

วิธีนี้จะทำการสร้างโปรแกรมที่เกี่ยวข้องกับการสกัดข้อมูลที่เป็นโปรแกรมสำหรับเขียนข้อมูลลงในไฟล์และฐานข้อมูล โดยจะทำการเขียนข้อมูลลงในไฟล์และฐานข้อมูลทุกๆครั้งที่มีการเพิ่ม ลบ และอัพเดทข้อมูลตามลำดับ วิธีการนี้จะต่างกับสองวิธีข้างต้นเล็กน้อยตรงที่วิธีนี้สามารถประยุกต์ใช้กับระบบที่มีการเก็บข้อมูลหลากหลายไม่ว่าจะเป็น ฐานข้อมูล การทำดัชนี หรือแฟ้มข้อมูล เป็นต้น แต่อย่างไรก็ตามวิธีการนี้อาจจะให้ประสิทธิภาพของระบบดำเนินการหรือระบบของแหล่งข้อมูลลดลงเนื่องจากต้องมีขั้นตอนการเขียนข้อมูลลงไฟล์เพิ่มเข้ามาเพื่อช่วยในการหาความเปลี่ยนแปลงที่เกิดขึ้นกับข้อมูล

การสกัดข้อมูลแบบรอเวลา (Deferred Data Extraction)

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

การสกัดข้อมูลโดยใช้ข้อมูลวันและเวลา (Capture Based on Date and Time Stamp)

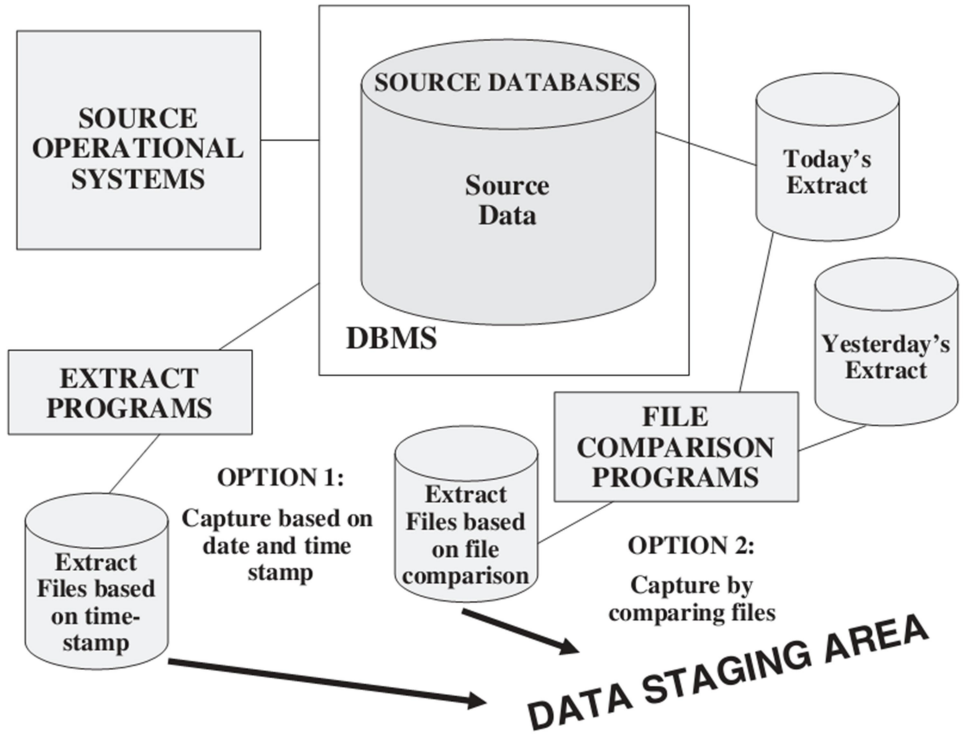
วิธีการนี้จะตั้งสมมติฐานที่ว่าข้อมูลแต่ละเรคคอร์ดในฐานข้อมูลจะมีข้อมูลเวลา (time stamp) แนบอยู่ด้วย ซึ่งข้อมูลเวลาจะเป็นเวลาในการเพิ่มหรืออัพเดทข้อมูลลงในฐานข้อมูล (ลบไม่มีเพราะเวลาจะถูกเก็บที่เรคคอร์ดในฐานข้อมูล เมื่อเรคคอร์ดถูกลบไปเราจะไม่สามารถเรียกดูข้อมูลเวลาได้) ในการเก็บข้อมูลเวลาจะช่วยให้เราสามารถเลือกเรคคอร์ดที่ต้องทำการสกัดข้อมูลได้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

การสกัดข้อมูลโดยใช้ข้อมูลเวลาอาจมีปัญหาเกิดขึ้นเมื่อมีการลบข้อมูลเกิดขึ้นกับระบบการดำเนินงาน ซึ่งจะทำให้เราไม่สามารถมองเห็นข้อมูลเวลาของข้อมูลนั้นๆได้ นี้จึงเป็นเหตุให้การสกัดข้อมูลไม่สามารถทราบถึงการลบข้อมูลที่เกิดขึ้นเหล่านั้นได้ จากปัญหาการลบข้อมูลเราสามารถดำเนินการแก้ไขปัญหาได้โดยทำการย้ายข้อมูลที่จะถูกลบไปยังตารางสำหรับข้อมูลที่จะถูกลบแทนที่จะทำการลบข้อมูลจริงๆ จากนั้นเมื่อถึงเวลาสำหรับการสกัดข้อมูล ข้อมูลในตารางสำหรับที่จะถูกลบจะถูกอ่านขึ้นมาเพื่อประมวลผลแล้วจึงค่อยทำการลบข้อมูลเหล่านั้นออกจากระบบการดำเนินงานจริงๆ ซึ่งวิธีดังกล่าวจะช่วยให้ข้อมูลในคลังข้อมูลมีความถูกต้องและตรงกับข้อมูลในแหล่งข้อมูล แต่ระบบการดำเนินงานจะต้องมีการทำงานที่เพิ่มขึ้น

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 ระบบอาจไม่สามารถใช้

วิธีนี้ในการสกัดข้อมูลได้ วิธีการนี้จะไม่เหมาะกับระบบทั่วๆไป แต่จะเหมาะกับระบบที่ไม่มีล็อกไฟล์และข้อมูลเวลาเท่านั้น



รูปที่ 8-8 วิธีสำหรับสกัดข้อมูลแบบรอเวลา

ประสิทธิภาพของการสกัดข้อมูลวิธีต่างๆ (Evaluation of the Techniques)

จากที่กล่าวมาข้างต้น เราได้ทราบถึงวิธีในการสกัดข้อมูลหลายวิธีด้วยกัน อาทิเช่น

- 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

การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล

ในการสกัดข้อมูลจากแหล่งข้อมูล ข้อมูลที่สกัดได้จะเป็นข้อมูลดิบ ซึ่งข้อมูลเหล่านี้อาจจะยังไม่สามารถนำไปประยุกต์ใช้กับคลังข้อมูลได้โดยตรง เนื่องจากคุณภาพของข้อมูลอาจยังไม่ดีพอต่อการตัดสินใจเชิงกลยุทธ์ ดังนั้นเราจึงต้องทำการปรับปรุงคุณภาพของข้อมูลให้ดีขึ้นเสียก่อนที่จะนำไปใช้ในคลังข้อมูลซึ่งก็คือ *“การเปลี่ยนแปลงหรือเปลี่ยนรูปข้อมูล (Data transformation)”* ที่ได้รับมาจากขั้นตอนการสกัดข้อมูล ในการเปลี่ยนแปลงข้อมูลนั้น ข้อมูลจะถูกทำการเปลี่ยนแปลงให้เป็นมาตราฐานมากขึ้น เนื่องจากข้อมูลที่สกัดได้อาจมาจากหลายแหล่งข้อมูล และแต่ละแหล่งข้อมูลอาจมีความโครงสร้างข้อมูลที่แตกต่างกัน การเปลี่ยนแปลงข้อมูลจะช่วยทำให้แน่ใจได้ว่า เมื่อทำการรวมข้อมูลเข้าด้วยกันแล้วข้อมูลที่ได้จะสามารถตอบสนองความต้องการทางธุรกิจได้

วัตถุประสงค์หลักของการเปลี่ยนแปลงข้อมูลจะเน้นที่การปรับปรุงคุณภาพของข้อมูลให้ดีขึ้น เมื่อทำการเปลี่ยนแปลงข้อมูลอาจทำให้ฟิลด์ (field) ต่างๆของข้อมูลเปลี่ยนแปลงไป หรืออาจทำให้โครงสร้างของข้อมูลมีการเปลี่ยนแปลง รวมทั้งยังส่งผลต่อประสิทธิภาพของการทำงานของคลังข้อมูลอีกด้วย ดังนั้นก่อนที่จะทำการเปลี่ยนแปลงข้อมูลเราควรจะต้องเข้าใจถึงข้อมูลที่มีอยู่ในแหล่งข้อมูล และเทคนิคต่างๆที่จะใช้ในการเปลี่ยนแปลงข้อมูลเสียก่อน

ขั้นตอนการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล

การเปลี่ยนแปลงข้อมูลสามารถแบ่งเป็นขั้นตอนย่อยได้หลายขั้นตอน ดังนี้

- **Selection** คือ การเลือกเรคคอร์ดทั้งหมดหรือกลุ่มของเรคคอร์ดเพียงบางกลุ่มจากแหล่งข้อมูล ซึ่งการทำ Selection จะเป็นขั้นตอนหนึ่งในการสกัดข้อมูล แต่ในบางกรณี การจัดวางองค์ประกอบของโครงสร้างข้อมูลจากแหล่งข้อมูลอาจไม่ตอบสนองต่อการเลือกข้อมูลที่สำคัญในขั้นตอนการสกัดข้อมูล ในกรณีเหล่านี้ เราควรจะทำการสกัดข้อมูลอีกครั้งหนึ่งโดยทำการสกัดเรคคอร์ดทั้งหมดก่อนจากนั้นค่อยทำการเลือกข้อมูลที่จะทำการเปลี่ยนแปลง
- **Splitting/Joining** คือ การแบ่งส่วนของข้อมูลที่ถูกเลือกไว้แล้วเพื่อที่จะทำการเปลี่ยนแปลงข้อมูล และการรวมข้อมูลที่ถูกเลือกไว้แล้วบางส่วนเข้าด้วยกัน โดยข้อมูลที่นำมารวมกันอาจมาจากหลายแหล่งข้อมูล
- **Conversion** คือ การทำให้ข้อมูลที่ถูกสกัดออกมาจากแหล่งข้อมูลเป็นมาตราฐานเดียวกัน และการทำให้ฟิลด์ต่างๆสามารถใช้งานได้ รวมถึงการทำให้ผู้ใช้เข้าใจในฟิลด์นั้นๆ
- **Summarization** คือ การสรุปข้อมูล กล่าวคือ ในบางสถานการณ์เราอาจพบว่าเป็นไปไม่ได้เลยที่จะเก็บข้อมูลที่มีรายละเอียดสูงมากๆในคลังข้อมูลที่สร้างขึ้น ซึ่งเหตุการณ์เหล่านี้อาจเกิดจากผู้ใช้ไม่ได้ต้องการข้อมูลที่มีรายละเอียดสูงสำหรับการวิเคราะห์ ตัวอย่างเช่น ซุปเปอร์มาร์เก็ตที่ซึ่งข้อมูลการขายที่มีความละเอียดสูงที่สุดจะเกิดขึ้นที่ขั้นตอนจ่ายเงิน ข้อมูลที่ละเอียดที่สุดที่สามารถเก็บได้ดังเช่น ยาสระผม ยี่ห้อแฟซ่า สูตรลดรังแค เป็นต้น ซึ่งข้อมูลที่ละเอียดมากๆเหล่านี้อาจไม่ใช่สิ่งที่ต้องการก็เป็นได้ ผู้จัดการอาจต้องการแค่ยอดขายของสินค้านั้นๆ ยอดขายในแต่ละสาขาของซุปเปอร์มาเก็ต (ในกรณีที่มีหลายสาขา) ยอดขายในแต่ละวัน ดังนั้น เมื่อมีกรณีแบบนี้เกิดขึ้น ขั้นตอนการเปลี่ยนแปลงข้อมูลจะต้องทำการสร้างผลสรุปของข้อมูลเพื่อให้สอดคล้องกับสิ่งที่ผู้จัดการต้องการ
- **Enrichment** คือ การจัดแจงฟิลด์ต่างๆที่มีอยู่ใหม่ รวมถึงการทำให้ฟิลด์นั้นๆเข้าใจได้ง่าย เพื่อทำให้ฟิลด์เหล่านั้นมีประโยชน์มากขึ้น เราอาจใช้ 1 ฟิลด์ (หรือมากกว่านั้น) จากเรคคอร์ดที่เป็นอินพุตเดียวกันเพื่อสร้างมุมมองของข้อมูลที่ขึ้นสำหรับคลังข้อมูล

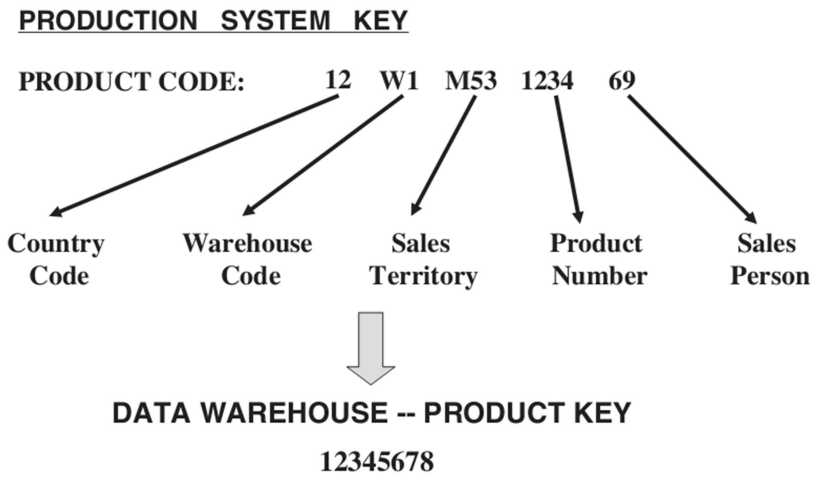
ประเภทของการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล

- **Format Revisions** คือ การเปลี่ยนชนิดของข้อมูลและความยาวของข้อมูลในบางฟิลด์ ตัวอย่างเช่น ชนิดของแพคเกจสินค้าอาจแสดงได้ด้วยโค้ดและชื่อ ซึ่งจะถูกเก็บอยู่ในฟิลด์ที่เป็นตัวเลขและข้อความในอีกทำนองหนึ่ง ความยาวของชนิดของแพคเกจสินค้าอาจแตกต่างกันระหว่างแหล่งข้อมูลที่แตกต่างกัน เราควรจะทำการสร้างมาตราฐานและทำการเปลี่ยนแปลงชนิดของข้อมูลไปเป็นข้อความเพื่อให้ข้อมูลดังกล่าวมีความหมายเพื่อที่ผู้ใช้จะสามารถเข้าใจได้ง่าย

- **Decoding of Fields** คือ การถอดรหัสลับที่อาจเกิดขึ้นในแหล่งข้อมูล เมื่อเราทำการสร้างคลังข้อมูลจากแหล่งข้อมูลหลายแหล่ง เราอาจจะพบเจอข้อมูลที่เป็นรหัสซึ่งอาจจะทำให้ผู้ใช้ไม่เข้าใจในรหัสเหล่านั้นได้ ตัวอย่างเช่น ข้อมูลเกี่ยวกับเพศ บางแหล่งข้อมูลจะเก็บข้อมูลเกี่ยวกับเพศ เป็น 1 และ 2 สำหรับเพศชายและหญิงตามลำดับ ในขณะที่บางแหล่งข้อมูลจะใช้การกำหนดค่าเกี่ยวกับเพศเป็น M และ F (บางครั้งอาจใช้ W) จากการสังเกตุระบบในปัจจุบันจะพบว่า มีหลายระบบที่ใช้รหัสลับแทนการอ้างถึงข้อมูลต่างๆทางธุรกิจ อาทิเช่น AC, IN, RE และ SU เมื่อเราพบเจอโค้ดในลักษณะแบบนี้เราจำเป็นต้องถอดรหัสลับและเปลี่ยนข้อมูลเหล่านั้นให้เป็นข้อมูลที่ผู้ใช้สามารถเข้าใจได้ จากตัวอย่างข้างต้น Active = AC, Inactive = IN, Regular = Re และ Suspended = SU เป็นต้น
- **Calculated and Derived Values** คือ การคำนวณหรือหาผลสรุปของข้อมูลและการนำข้อมูลเหล่านั้นไปใช้ ตัวอย่างเช่น ข้อมูลที่ถูกสกัดจากระบบการขายประกอบด้วยจำนวนยอดขาย และประมาณการค่าใช้จ่ายในการดำเนินงานจำแนกตามผลิตภัณฑ์ เราจะต้องทำการคำนวณต้นทุนทั้งหมดและอัตรากำไรก่อนที่จะเก็บข้อมูลลงในคลังข้อมูล เป็นต้น
- **Splitting of Single Fields** คือ การแยกฟิลด์หนึ่งๆออกเป็นหลายฟิลด์ ตัวอย่างเช่น ในระบบเดิมที่ทำการเก็บข้อมูลชื่อและที่อยู่ของลูกค้าและพนักงานในแบบที่เป็นข้อความ โดยที่ ชื่อแรก ชื่อกลาง และนามสกุล จะถูกเก็บอยู่ในฟิลด์เพียงฟิลด์เดียว หรือในบางระบบจะเก็บ เขต จังหวัด และรหัสไปรษณีย์ไว้ในฟิลด์เดียว เป็นต้น จากระบบเดิมที่กล่าวมาข้างต้น เราจำเป็นต้องเก็บองค์ประกอบของ ชื่อ และที่อยู่ ในคลังข้อมูลให้แยกออกจากกัน เพื่อ (1) ช่วยในเรื่องประสิทธิภาพในการดำเนินงานโดยใช้การสร้างดัชนี (indexing) กับองค์ประกอบนั้นๆ และ (2) ผู้ใช้คลังข้อมูลอาจต้องการวิเคราะห์ข้อมูลโดยใช้ข้อมูลที่ถูกแยกจากกระบวนการข้างต้น
- **Merging Information** คือ การผสานข้อมูลที่มีรายละเอียดแต่ละส่วนกระจายอยู่ที่หลายแหล่งข้อมูลเข้าด้วยกัน ตัวอย่างเช่น ข้อมูลเกี่ยวกับสินค้าที่อาจมาจากหลายแหล่งข้อมูล โค้ดและคำอธิบายสินค้าอาจมาจากแหล่งข้อมูลเพียงแหล่งเดียว ข้อมูลเกี่ยวกับประเภทของแพคเกจอาจจะพบในอีกแหล่งข้อมูลหนึ่ง ข้อมูลต้นทุนของสินค้าอาจจะได้มาจากอีกแหล่งข้อมูลหนึ่ง เราต้องทำการผสานข้อมูลของโค้ดของสินค้า คำอธิบายสินค้า ชนิดของแพคเกจของสินค้า และต้นทุนของสินค้าเข้าด้วยกันและเก็บไว้ในเอนทิตี (Entity) เดียวกัน
- **Character set conversion** คือ การแปลงเซตของตัวอักษรให้เป็นมาตราฐาน เพื่อใช้ในการจัดเก็บข้อมูลในคลังข้อมูล อาทิเช่น ถ้าเรามีเมนเฟรม (Mainframe) ที่เก็บระบบเดิม (แหล่งข้อมูล) โดยข้อมูลที่ได้จากระบบนี้จะอยู่ภายใต้อักขระแบบ EBCDIC จากระบบข้างต้น ถ้าเราต้องการที่จะสร้างคลังข้อมูลโดยใช้คอมพิวเตอร์ส่วนบุคคล (Personal Computer, PC) เราต้องทำการแปลงเซตของอักขระจาก EBCDIC ให้อยู่ในรูปของ ASCII ถึงจะเก็บข้อมูลเข้าสู่คลังข้อมูลได้
- **Conversion of Units of Measurements** คือ การแปลงหน่วยของมาตรวัด ในปัจจุบันหลายๆบริษัทได้กระจายการทำธุรกิจอยู่ในหลายๆประเทศ ถ้าทำธุรกิจกับประเทศในทวีปยุโรปหน่วยที่ใช้วัด

จะเป็นเมตร แต่ในบางประเทศอาจใช้หน่วยวัดเป็นเซนติเมตร ดังนั้นถ้าเราทำธุรกิจกับหลายๆประเทศ เราต้องทำการแปลงมาตรวัดให้อยู่ในมาตราฐานเดียวกัน

- **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
- **Summarization** คือ การสรุปข้อมูลเพื่อโอนถ่ายเข้าไปยังคลังข้อมูลแทนที่จะโอนถ่ายข้อมูลที่มีความละเอียดค่อนข้างมากเข้าไปโดยตรง ตัวอย่างเช่น ในการวิเคราะห์รูปแบบการขายของบริษัทบัตรเครดิต เราอาจจะไม่จำเป็นต้องเก็บข้อมูลแต่ละการทำธุรกรรมของบัตรใบหนึ่งๆ เราอาจจะต้องทำการหาผลสรุปของการธุรกรรมในหนึ่งวันต่อบัตรหนึ่งใบแล้วทำการเก็บข้อมูลเหล่านั้นไว้ในคลังข้อมูล
- **Key Restructuring** คือ การสร้างคีย์ใหม่แทนที่ของเดิม ตัวอย่างเช่น จากรูปที่ 8-10 ที่แสดงโค้ดของสินค้าซึ่งถูกกำหนดโดยสามารถบอกถึงแหล่งที่มาได้และมีความซับซ้อน ซึ่งถ้าเราใช้โค้ดของสินค้าลักษณะนี้เป็นคีย์หลัก (Primary key) อาจทำให้เกิดปัญหาในกรณีที่โค้ดของสินค้านี้ถูกเคลื่อนย้ายไปยังคลังข้อมูลอื่น ก็จะทำให้ส่วนของโค้ดของสินค้าในส่วนของโค้ดของคลังข้อมูลต้องเปลี่ยนไปซึ่งการเปลี่ยนแปลงนี้จะทำให้เกิดปัญหากับระบบดั้งเดิมที่จะต้องแก้ตามด้วย



รูปที่ 8-10 การสร้างคีย์ของข้อมูลขึ้นใหม่ (Key restructuring)

- **Deduplication** คือ การขจัดความซ้ำซ้อนของข้อมูล ตัวอย่างเช่น ในหลายๆบริษัท การเก็บข้อมูลของลูกค้าอาจจะทำการเก็บข้อมูลอยู่ในหลายเรคคอร์ด แต่สำหรับคลังข้อมูลแล้ว เราอาจต้องการที่จะเก็บข้อมูลเพียงเรคคอร์ดเดียวสำหรับลูกค้าหนึ่งคน เราจึงต้องทำการเชื่อมข้อมูลหลายเรคคอร์ดเหล่านั้นเข้าด้วยกันแล้วเก็บไว้ให้อยู่ในเรคคอร์ดเดียว

หลักการสร้างฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล

การทำการเปลี่ยนแปลงข้อมูลจะมีอยู่ 2 วิธีหลักในการดำเนินการ คือ (1) การใช้เครื่องมือ และ (2) การสร้างฟังก์ชันการเปลี่ยนแปลงข้อมูลขึ้นเอง ในการจะเลือกว่าจะใช้วิธีการใดในการสร้างฟังก์ชันการเปลี่ยนแปลงข้อมูลนั้น ผู้สร้างจะต้องพิจารณาปัจจัยต่างๆที่มีผลกระทบต่อการดำเนินการ ถ้าเราต้องการใช้เครื่องมือในการสร้าง เราอาจจะต้องใช้เวลาในการดำเนินงานต่างๆ อาทิเช่น ศึกษาเกี่ยวกับเครื่องมือนั้นๆ ปรับแต่งเครื่องมือเพื่อให้เหมาะสมกับคลังข้อมูลที่สร้างขึ้น ติดตั้งเครื่องมือที่จำเป็นต้องใช้ ฝึกอบรมทีมงานเกี่ยวกับเครื่องมือ และรวมเครื่องมือที่จะใช้เข้ากับคลังข้อมูล เครื่องมือสำหรับการเปลี่ยนแปลงข้อมูลอาจมีราคาแพง ถ้าของเขตของคลังข้อมูลที่เราจะสร้างขึ้นมีขนาดไม่ใหญ่มาก เราอาจจะไม่มีงบประมาณสำหรับเครื่องมือที่จะใช้ก็เป็นได้

การใช้เครื่องมือสำเร็จรูปในการสร้างฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล

ในหลายๆปีที่ผ่านมา มีเครื่องมือสำหรับการเปลี่ยนแปลงข้อมูลถูกพัฒนาอย่างแพร่หลายเพื่อรองรับหลายๆการทำงานและมีความยืดหยุ่นมากขึ้น การใช้เครื่องมือในการเปลี่ยนแปลงข้อมูลจะสามารถช่วยเพิ่มประสิทธิภาพในการทำงานรวมถึงการเพิ่มความถูกต้องของข้อมูลด้วย ข้อดีอย่างหนึ่งของการใช้เครื่องมือในการแปลงข้อมูล คือ เครื่องมือจะทำการเก็บเมตาดาต้า (Metadata) ให้เองอัตโนมัติ เมื่อเราทำการกำหนดค่าพารามิเตอร์ และกฎต่างๆ เครื่องมือจะทำการเก็บข้อมูลเหล่านั้นไว้เป็นเมตาดาต้าด้วย เมื่อเราทำการเปลี่ยนฟังก์ชันการเปลี่ยนแปลงข้อมูล ซึ่งอาจเกิดจากการเปลี่ยนแปลงของการดำเนินการทางธุรกิจ หรือนิยามของข้อมูล เราเพียงเลือกฟังก์ชันที่ต้องการเปลี่ยนจากฟังก์ชันเดิม เครื่องมือจะทำการปรับแก้เมตาดาต้าให้เองโดยอัตโนมัติ

การสร้างฟังก์ชันการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลขึ้นเอง

การสร้างฟังก์ชันการเปลี่ยนแปลงข้อมูลเอง ได้รับความนิยมเป็นอย่างมากในช่วงแรกของการพัฒนาคลังข้อมูล แต่เมื่อมีเครื่องมือวางขายอยู่ในตลาดไอที ความนิยมก็ลดลง แต่อย่างไรก็ดีการสร้างฟังก์ชันการเปลี่ยนแปลงข้อมูลเองก็ยังได้รับความนิยมสำหรับคลังข้อมูลที่มีขนาดเล็กๆ ในการสร้างฟังก์ชันต่างๆเองจะต้องมีนักวิเคราะห์และโปรแกรมเมอร์ที่มีความรู้อยู่แล้วและเชี่ยวชาญที่จะสามารถผลิตโปรแกรมและสคริปต์ได้ ข้อเสียของการสร้างฟังก์ชันการเปลี่ยนแปลงข้อมูลขึ้นเองจะเกี่ยวกับเมตาดาต้า นั่นคือ ผู้พัฒนาฟังก์ชันการเปลี่ยนแปลงข้อมูลจะต้องออกแบบว่าควรเก็บข้อมูลใดบ้างไว้ในเมตาดาต้า และต้องคิดเกี่ยวกับการดูแลรักษาเมตาดาต้าด้วย

การถ่ายโอนข้อมูลไปยังคลังข้อมูล

การถ่ายโอนข้อมูลจะเป็นการรับเอาข้อมูลที่สกัดได้จากแหล่งข้อมูลและทำการเปลี่ยนแปลงข้อมูลแล้วไปเก็บไว้ในคลังข้อมูล การถ่ายโอนข้อมูลนั้นสามารถทำได้หลายวิธีซึ่งมักจะเรียกการถ่ายโอนข้อมูลว่า การประ

ยุต์ใช้ข้อมูล (applying the data) การโหลดข้อมูล (loading the data) และการทำให้ข้อมูลมีความสดใหม่ (refreshing the data) ซึ่งสามารถอธิบายคร่าวๆได้ดังนี้

- Initial load คือ การถ่ายโอนข้อมูลจากแหล่งข้อมูลไปยังคลังข้อมูลครั้งแรก
- Incremental load คือ การเพิ่มข้อมูลที่มีการเปลี่ยนแปลงเข้าสู่คลังข้อมูลอย่างต่อเนื่องตามความจำเป็นในลักษณะที่เป็นระยะๆ
- Full refresh คือ การลบข้อมูลทั้งหมดออกจากคลังข้อมูล แล้วทำการถ่ายโอนข้อมูลที่มีความสดใหม่ลงไปในคลังข้อมูล ซึ่งการถ่ายโอนข้อมูลใหม่จะเหมือนกับการทำ initial load อีกครั้งหนึ่ง

ในการโหลดข้อมูลเข้าสู่คลังข้อมูลในแต่ละครั้งอาจมีข้อมูลที่ต้องทำการถ่ายโอนเป็นจำนวนมาก ซึ่งเป็นเหตุให้ใช้เวลาในการถ่ายโอนข้อมูลค่อนข้างมาก ดังนั้นเพื่อให้ไม่เกิดข้อผิดพลาดของการใช้งานคลังข้อมูล ผู้ดูแลระบบควรทำการปิดคลังข้อมูลระหว่างการถ่ายโอนข้อมูล ซึ่งในการโหลดข้อมูลนั้นเราควรหาช่วงเวลาที่เหมาะสมซึ่งไม่ส่งผลกระบบต่อผู้ใช้คลังข้อมูล ในการโหลดข้อมูลถ้าขนาดของข้อมูลมีจำนวนมากเราอาจจะพิจารณาที่จะแบ่งส่วนของการถ่ายโอนข้อมูลออกเป็นส่วนที่เล็กลงหลายๆส่วน เพื่อที่จะได้ใช้เวลาในถ่ายโอนในแต่ละครั้งน้อยลง การแบ่งส่วนของการถ่ายโอนข้อมูลมีข้อดีอยู่สองข้อ คือ (1) เราอาจจะสามารถทำการถ่ายโอนข้อมูลแบบขนานได้ (loads in parallel) และ (2) เราอาจจะสามารถทำการสำรองข้อมูลในส่วนที่ไม่ได้ใช้จากคลังข้อมูลได้ (back up data in data warehouse) แล้วทำการถ่ายโอนข้อมูลส่วนอื่นๆเข้าสู่คลังข้อมูลได้ ซึ่งในการโหลดข้อมูลในแต่ละครั้งเราไม่สามารถคาดคะเนเวลาที่ใช้ในการถ่ายโอนข้อมูลได้ โดยเฉพาะอย่างยิ่งการทำ initial load และ full refresh

ในการถ่ายโอนข้อมูล เราอาจจะไม่ได้ทำการถ่ายโอนข้อมูลสำเร็จทุกครั้งไป การถ่ายโอนเรคคอร์ดหนึ่งๆเข้าสู่ fact table ของคลังข้อมูลอาจมีปัญหาเกิดขึ้น เนื่องจาก concatenated key อาจผิด และไม่สอดคล้องกับ dimension tables เมื่อเกิดข้อผิดพลาดในการถ่ายโอนข้อมูลเกิดขึ้น เราจำเป็นต้องเตรียมกระบวนการสำหรับจัดการกับเรคคอร์ดที่ไม่ถูกโหลดเข้าสู่คลังข้อมูล และควรมีแผนสำหรับการประกันคุณภาพของการถ่ายโอนข้อมูล นอกจากความถูกต้องและความสมบูรณ์ของการโหลดข้อมูลแล้ว ยังมีปัจจัยหนึ่งที่สำคัญมาก คือ การเลือกวิธีในการถ่ายโอนข้อมูลที่ซึ่งจะต้องสอดคล้องกับสถาปัตยกรรมของคลังข้อมูลว่าเป็นแบบใด เช่น กรณีที่ staging area และฐานข้อมูลของคลังข้อมูลอยู่ในเซิร์ฟเวอร์เดียวกัน เราจะสามารถถ่ายโอนข้อมูลได้โดยตรงโดยไม่เสียเวลาในการถ่ายโอนข้อมูลมาก แต่สำหรับกรณีอื่นๆ 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)

Load

ถ้าคลังข้อมูลมีตารางเป้าหมาย (target table) ที่จะทำการถ่ายโอนข้อมูลอยู่แล้วและในตารางนั้นมีข้อมูลอยู่ การทำงานของวิธี Load จะทำการล้างข้อมูลที่มีอยู่และถ่ายโอนข้อมูลใหม่เข้าไปในตาราง แต่ในกรณีที่ตารางข้อมูลที่มีอยู่แล้วไม่มีข้อมูลอยู่ก่อนหน้าจะทำการโหลดข้อมูลใหม่เข้าไปทันที

Append

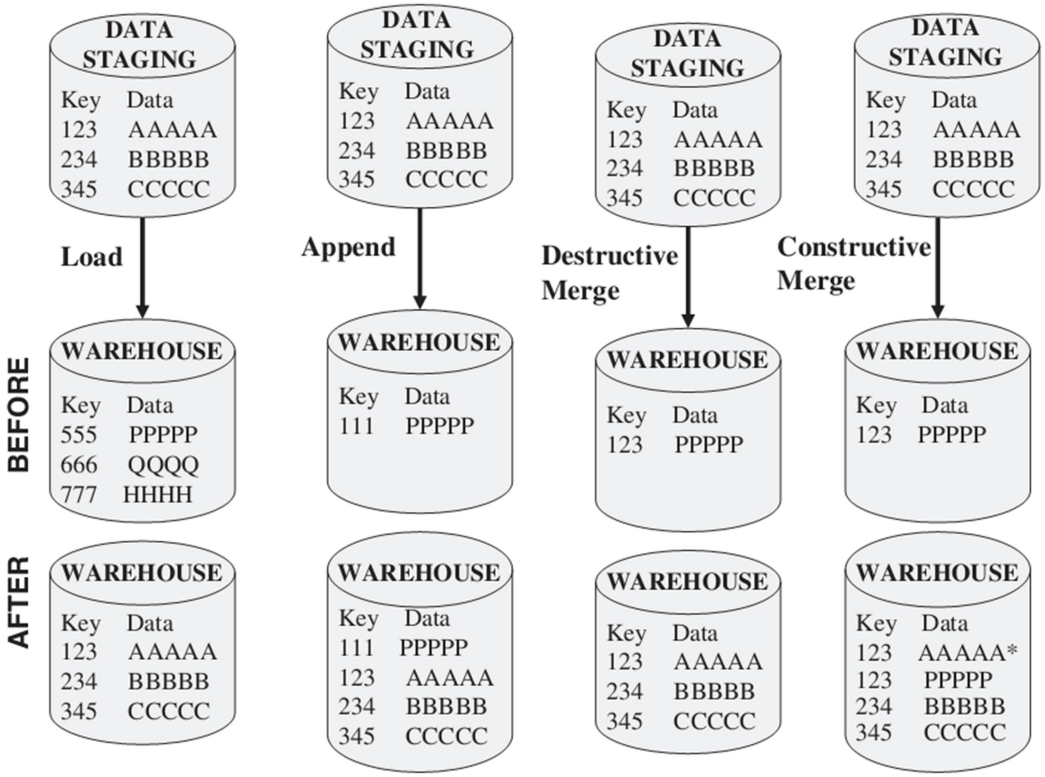
ถ้าคลังข้อมูลมีตารางเป้าหมายที่จะทำการถ่ายโอนข้อมูลอยู่แล้วและในตารางมีข้อมูลอยู่ก่อนแล้ว การ Append จะทำการถ่ายโอนข้อมูลเข้าสู่ตารางนั้นโดยไม่มีเงื่อนไข แต่ถ้ากรณีที่ตารางมีข้อมูลที่ต้องการถ่ายโอนอยู่แล้ว (มีข้อมูลซ้ำ) เราสามารถเลือกว่าจะปฏิบัติอย่างไร ระหว่างเพิ่มข้อมูลสู่ตารางเป้าหมายโดยยอมให้มีการซ้ำกันของข้อมูลหรือไม่ต้องทำการเพิ่มข้อมูลใหม่สู่ตารางเนื่องจากมีข้อมูลอยู่แล้วและหลีกเลี่ยงความซ้ำซ้อน

Destructive Merge

ถ้าคีย์หลัก (Primary key) ของเรคคอร์ดใหม่ที่จะทำการถ่ายโอนข้อมูลตรงกับคีย์หลักของข้อมูลที่มีอยู่แล้วในตารางเป้าหมาย Destructive Merge จะทำการอัพเดทข้อมูลเรคคอร์ดที่มีคีย์หลักซ้ำกับเรคคอร์ดใหม่ที่จะทำการถ่ายโอน แต่ถ้าเรคคอร์ดใหม่มีคีย์หลักไม่ซ้ำกับเรคคอร์ดใดๆที่อยู่ในตารางเป้าหมายจะทำการถ่ายโอนข้อมูลเข้าสู่ตาราง

Constructive Merge

ถ้าคีย์หลัก (Primary key) ของเรคคอร์ดใหม่ที่จะทำการถ่ายโอนข้อมูลตรงกับคีย์หลักของข้อมูลที่มีอยู่แล้วในตารางเป้าหมาย Constructive Merge จะทำการเก็บข้อมูลเก่าในตารางไว้ แล้วทำการถ่ายโอนข้อมูลใหม่เข้าสู่ตาราง และทำเครื่องหมายให้กับข้อมูลใหม่ที่ทำการถ่ายโอนว่าเป็นข้อมูลที่มาแทนข้อมูลเก่า



รูปที่ 8-11 เทคนิคการประยุกต์ใช้ข้อมูล

จากข้างต้น เราได้ทราบถึงวิธีในการประยุกต์ใช้ข้อมูลกับคลังข้อมูล จากนี้ไปเราจะมาพิจารณาว่าเราจะสามารถนำวิธีต่างๆมาใช้กับแต่ละชนิดของการถ่ายโอนข้อมูลได้อย่างไร

Initial Load

ในการทำ Initial load ถ้าเราสามารถทำการถ่ายโอนข้อมูลได้ภายในครั้งเดียว เราจะสามารถใช้วิธีการ load ในการถ่ายโอนข้อมูลได้โดยตรง แต่ถ้าเราทำการแบ่ง Initial load ออกเป็นการทำงานหลายๆครั้งเราจะต้องใช้วิธีการ load ในการถ่ายโอนข้อมูลครั้งแรกแล้วใช้วิธีการ append ในการถ่ายโอนข้อมูลครั้งถัดๆไป จากการที่เราทำการแบ่ง Initial load ออกเป็นการทำงานหลายๆครั้ง แสดงให้เห็นว่าข้อมูลที่ต้องทำการถ่ายโอนข้อมูลนั้นมีจำนวนมาก ดังนั้น การสร้างดัชนี (indexes) ของ Initial load จะใช้เวลาค่อนข้างมาก เราควรจะมองข้ามการสร้างดัชนีก่อนการถ่ายโอนข้อมูล เพื่อให้การโหลดนั้นสามารถทำงานได้อย่างรวดเร็ว จากนั้นค่อยทำการดัชนีขึ้นมาใหม่เมื่อการถ่ายโอนข้อมูลเสร็จสิ้น

Incremental Loads

ในหลายๆแอพพลิเคชันอาจมีความเปลี่ยนแปลงเกิดขึ้นในแหล่งข้อมูล ณ ช่วงเวลาหนึ่งๆ ซึ่งเราอาจจำเป็นต้องเก็บช่วงเวลาของการเปลี่ยนแปลงนั้นๆไว้ในคลังข้อมูล ถ้าช่วงเวลาเป็นส่วนหนึ่งของคีย์หลัก หรือถ้าช่วงเวลาถูกรวมอยู่ในการเปรียบเทียบระหว่างข้อมูลที่จะเข้ามาใหม่และข้อมูลที่มีอยู่เดิมในคลังข้อมูล Constructive Merge อาจถูกใช้ในการถ่ายโอนข้อมูล ซึ่งวิธีการนี้จะสามารถช่วยในการเก็บรักษาช่วงเวลาของการเปลี่ยนแปลงข้อมูล แต่ในกรณีที่ข้อมูลใน dimension table มีการเปลี่ยนแปลงอย่างช้าๆเรคคอร์ดที่

ถูกเก็บอยู่ในคลังข้อมูลควรจะถูกแทนที่ด้วยเรคคอร์ดที่จะเข้ามาใหม่ที่มีคีย์หลักตรงกัน ดังนั้นเราควรใช้ Destructive Merge ในการถ่ายโอนข้อมูล โดยที่การถ่ายโอนข้อมูลโดย Destructive Merge นั้นจะใช้งานได้กับทุกตารางที่เป็นเป้าหมายได้เมื่อข้อมูลเก่าๆไม่ได้มีความสำคัญในแอพพลิเคชัน

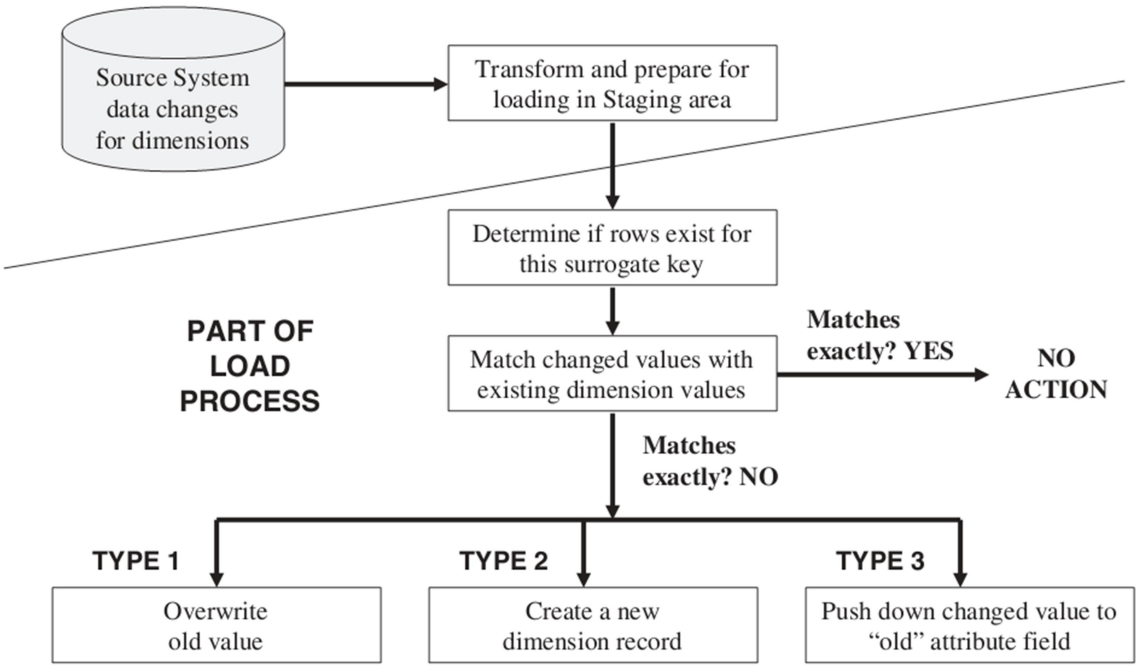
Full Refresh

วิธีในการถ่ายโอนข้อมูลของการทำ Full Refresh ก็จะเหมือนกับวิธีที่ใช้ใน Initial load ซึ่งก็คือจะใช้วิธี Load และ Append ในการถ่ายโอนข้อมูล การทำงานจะเริ่มจากการลบข้อมูลออกจากตารางในคลังข้อมูลก่อน จากนั้นใช้การ Load เพื่อทำการถ่ายโอนข้อมูลเข้าสู่ข้อมูลโดยตรง แต่ถ้าการถ่ายโอนข้อมูลมีการแบ่งส่วนการทำงานออกเป็นหลายๆครั้ง เราจะทำการใช้วิธีการ Load เพื่อทำการถ่ายโอนข้อมูลในครั้งแรก จากนั้นในครั้งต่อๆไปก็จะใช้วิธีการ Append ในการโหลดข้อมูลเข้าสู่คลังข้อมูล

การถ่ายโอนข้อมูลเข้าสู่ Dimension table

Dimension table ที่ถูกเก็บอยู่ในคลังข้อมูล อาทิเช่น customer dimension, product dimension และ time dimension จะมีแอทริบิวที่เป็นมาตราวัดอยู่ (measurement) อาทิเช่น sales และ costs ข้อมูลเหล่านี้เป็นข้อมูลที่สอดคล้องกับข้อมูลที่ถูกเก็บไว้ในแหล่งข้อมูล ในการสร้างคลังข้อมูล เราจะต้องทำการถ่ายโอนข้อมูลจากแหล่งข้อมูลเข้าสู่ dimension table ซึ่งมีวิธีการ 2 วิธีด้วยกันคือ 1) การทำ Initial loading ของตารางต่างๆ และ 2) การประยุกต์ใช้ข้อมูลที่มีการเปลี่ยนแปลง (applying change) จากสองวิธีที่ได้กล่าวข้างต้นยังมีอีกหนึ่งปัจจัยสำคัญที่เราจำเป็นต้องพิจารณาคือ*ความสอดคล้องกันระหว่างคีย์ของเรคคอร์ดจากแหล่งข้อมูลและคีย์ของเรคคอร์ดในคลังข้อมูล* อย่างที่เราทราบกันดีว่าเราไม่ได้ใช้คีย์ของเรคคอร์ดจากแหล่งข้อมูลไปเป็นคีย์ของข้อมูลในคลังข้อมูล เนื่องจากคีย์ของเรคคอร์ดในแหล่งข้อมูลอาจไม่มีความสมบูรณ์เพียงพอ ดังนั้นเมื่อเราทำการสกัดข้อมูลจากแหล่งข้อมูลแล้ว เราจะต้องสร้างระบบสำหรับการสร้างคีย์ขึ้นมาใหม่สำหรับใช้ในคลังข้อมูลด้วย ในการทำ Initial load และการโหลดครั้งต่อๆ 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 เราจะต้องทำการสร้างคีย์หลักให้กับเรคคอร์ดนั้นๆ ก่อนการโหลดข้อมูลเข้าสู่ fact table จะมีเคล็ดลับมากมายดังนี้

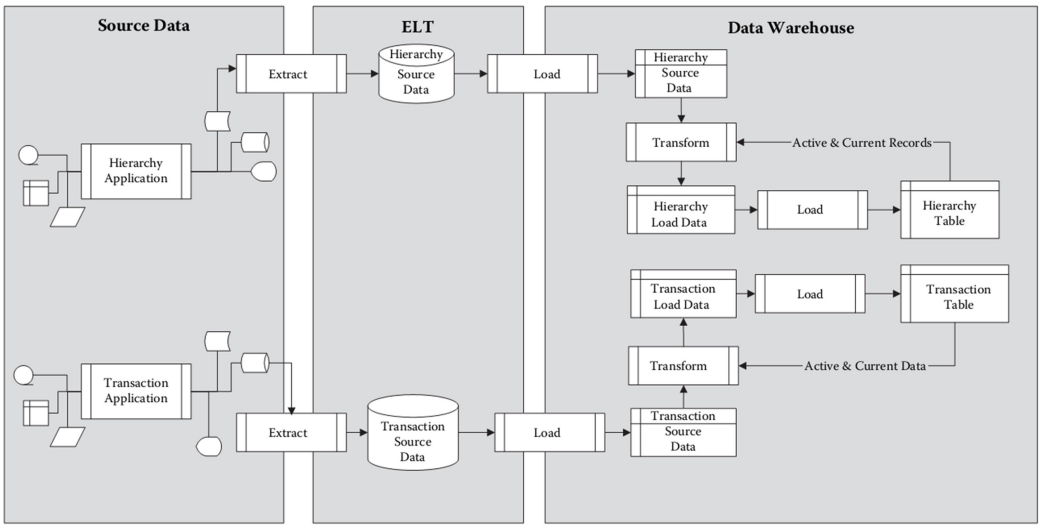
- ระบุข้อมูลทางประวัติศาสตร์ที่เป็นประโยชน์และน่าสนใจสำหรับคลังข้อมูล (Identify historical data useful and interesting for the data warehouse)
- กำหนดและปรับแต่งกฎเกณฑ์ทางธุรกิจที่ถูกสกัดออกมา (Define and refine extract business rules)
- เก็บข้อมูลการตรวจสอบทางสถิติเพื่อทำการสร้างการเชื่อมโยงกลับไปยังระบบการดำเนินงาน (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)

ข้อสังเกตในการถ่ายโอนข้อมูลเข้าสู่ 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)

การสลับตำแหน่งการทำงานจาก “อีทีแอล” เป็น “อีแอลที”

“อีแอลที” เป็นกระบวนการสกัดข้อมูลจากแหล่งข้อมูลแล้วทำการถ่ายโอนข้อมูลเข้าสู่คลังข้อมูล เช่นเดียวกับ “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)

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