บทที่ 2 แบบจำลองข้อมูล

วัตถุประสงค์ของบทเรียน

- ศึกษาเกี่ยวกับแบบจำลองข้อมูลและความสำคัญของแบบจำลองข้อมูล
- ศึกษาเกี่ยวกับส่วนประกอบของแบบจำลองข้อมูล
- ศึกษาเกี่ยวกับกฎเกณฑ์ทางธุรกิจและการประยุกต์ใช้กฎเกณฑ์ทางธุรกิจในการออกแบบฐานข้อมูล
- ศึกษาเกี่ยวกับสาเหตุของการคิดค้นแบบจำลองข้อมูล
- ศึกษาเกี่ยวกับการแบ่งประเภทของแบบจำลองข้อมูลตามระดับของความชัดเจนของข้อมูล

เนื้อหาของบทเรียน

แนวคิดพื้นฐานของแบบจำลองข้อมูล ขั้นตอนในการสร้างแบบจำลองข้อมูล โมเดลความสัมพันธ์แบบเอ็นทิตี้ อี-อาร์ไดอะแกรม

กิจกรรมการเรียน-การสอน

- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารคำสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

อุปกรณ์ที่ใช้ในการเรียน-การสอน

- เอกสารคำสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

การวัดและประเมินผล

- การตอบคำถามระหว่างการเรียน-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

จากบทที่แล้วเราจะทราบถึงเหตุผลที่การออกแบบฐานข้อมูลเป็นสิ่งสำคัญสำหรับการประยุกต์ใช้ระบบฐานข้อมูลในหลายๆแง่มุม ดังนั้น ในระหว่างขั้นตอนการออกแบบฐานข้อมูล ทีมผู้ออกแบบจะต้องเอาใจใส่กับรายละเอียดต่างๆ และพยายามที่จะหลีกเลี่ยงที่จะเกิดเหตุการณ์ที่ว่า ผู้ออกแบบ ผู้เขียนโปรแกรม และผู้ใช้งานระบบฐานข้อมูลมีมุมมองเกี่ยวกับข้อมูลหนึ่งๆที่ไม่เหมือนกันที่ซึ่งจะทำให้การออกแบบฐานข้อมูลไม่สามารถสะท้อนถึงขั้นตอนวิธีในการดำเนินธุรกิจได้ เพื่อหลีกเลี่ยงเหตุการณ์ดังกล่าว ผู้ออกแบบฐานข้อมูลควรจะทำการติดต่อสื่อสารกับผู้เขียนโปรแกรมและผู้ใช้งานระบบฐานข้อมูลบ่อยๆเพื่อที่จะทราบถึงธรรมชาติของข้อมูลและการประยุกต์ใช้งานข้อมูลนั้นๆได้อย่างชัดเจน และควรที่จะมีความเข้าใจเกี่ยวกับข้อมูลที่ตรงกันกับผู้เขียนโปรแกรมและผู้ใช้งานระบบฐานข้อมูล

นอกเหนือจากการติดต่อสื่อสารบ่อยๆแล้ว การประยุกต์ใช้แบบจำลองข้อมูล (data model) หรือแบบจำลองฐานข้อมูล (database model) ที่ได้จากการนิยามเอ็นทิตี้ (entity) และความสัมพันธ์ระหว่างเอ็นทิตี้จะสามารถทำให้ทุกฝ่ายมีความเข้าใจในข้อมูลได้ง่ายขึ้น และสามารถลดความซับซ้อนของการออกแบบฐานข้อมูลให้มีความเข้าใจง่ายมากขึ้นด้วย ดังนั้น เนื้อหาในบทนี้จะมุ่งเน้นที่การศึกษาเกี่ยวกับการสร้างแบบจำลองฐานข้อมูลที่ซึ่งจะทำให้เราสามารถนำไปประยุกต์ใช้ในการออกแบบฐานข้อมูลได้

2.1 แบบจำลองข้อมูลคือ?

การสร้างแบบจำลองข้อมูลหรือแบบจำลองฐานข้อมูลจะเป็นขั้นตอนแรกของการออกแบบฐานข้อมูลที่ซึ่งจะมุ่งเน้นที่การกำหนดโครงสร้างของฐานข้อมูลที่ซึ่งจะใช้ในการจัดเก็บและจัดการข้อมูลของผู้ใช้งานฐานข้อมูล ในหลายๆครั้งการสร้างแบบจำลองฐานข้องมูลอาจหมายถึงการระบุถึงแบบจำลองข้อมูสำหรับการกำหนดขอบเขตของปัญหา (problem domain) ที่เราจะพิจารณา แบบจำลองข้อมูลมักมีลักษณะเป็นแผนภาพที่ใช้แสดงโครงสร้างที่ซับซ้อนของฐานข้อมูล มีหน้าที่ในการช่วยให้ผู้ออกแบบฐานข้อมูลสามารถเข้าใจความซับซ้อนของข้อมูลที่ถูกใช้ในองค์กรต่างๆ นอกจากนั้น แบบจำลองข้อมูลมักจะแสดงถึงโครงสร้างของข้อมูลในฐานข้อมูลและคุณลักษณะของข้อมูลเหล่านั้น ข้อจำกัดต่างๆ การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล และอื่นๆที่ซึ่งสามารถสนับสนุนการกำหนดขอบเขตของปัญหา

การสร้างแบบจำลองฐานข้อมูลจะเป็นกระบวนการทำซ้ำแบบค่อยเป็นค่อยไป โดยในตอนเริ่มต้นเราอาจเริ่มต้นจากความเข้าอย่างๆง่ายๆเกี่ยวกับขอบเขตปัญหาและสามารถทำการสร้างแบบจำลองฐานข้อมูลอย่างง่ายตามความเข้าใจที่มี แต่หลังจากทำการติดต่อสื่อสารกับผู้ใช้และการพิจารณาเกี่ยวกับข้อมูลและขั้นตอนในการดำเนินธุรกิจต่างๆซ้ำแล้วซ้ำเล่าจะทำให้เรามีความเข้าใจเกี่ยวกับของเขตของปัญหามากยิ่งขึ้นและจะช่วยให้เราสามารถเพิ่มเติมรายละเอียดของแบบจำลองฐานข้อมูลได้มากขึ้น ในท้ายสุด เราจะได้แบบจำลองฐานข้อมูลที่สามารถตอบสนองต่อความต้องการของผู้ใช้ที่ซึ่งจะมีลักษณะคล้ายกับพิมพ์เขียวที่บรรจุไปด้วยวิธีการในการสร้างฐานข้อมูล โดยพิมพ์เขียวที่ได้จะมีลักษณะเป็นแผนภาพที่จะประกอบไปด้วย 1) คำอธิบายเกี่ยวกับโครงสร้างของข้อมูลที่ใช้ในการจัดเก็บข้อมูลที่ผู้ใช้ต้องการ 2) กฎที่เกี่ยวข้องกับการการันตีความสมบูรณ์ของข้อมูล (data integrity) และ 3) วิธีการในการจัดการข้อมูลที่ซึ่งจะสนับสนุนการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูล ตามลำดับ จากที่ได้กล่าวข้างต้น แบบจำลองข้อมูลทำหน้าที่เสมือนกับเครื่องมือ

ที่ใช้ในการติดต่อสื่อสารระหว่างผู้ออกแบบฐานข้อมูล ผู้เขียนโปรแกรม และผู้ใช้ระบบฐานข้อมูล ถ้าเราสามารถสร้างแบบจำลองข้อมูลที่ดี แบบจำลองนี้จะช่วยเพิ่มความเข้าใจและจะสามารถสะท้อนถึงโครงสร้างที่เหมาะสมสำหรับจัดเก็บข้อมูลตามแนวทางในการดำเนินธุรกิจ (หมายเหตุ ในการออกแบบฐานข้อมูล ผู้ออกแบบฐานข้อมูลควรที่จะต้องมีการตัดสินใจที่ดีที่ซึ่งจะช่วยให้สามารถสร้างแบบจำลองฐานข้อมูลที่ดีได้ การตัดสินใจที่ดีมักจะได้มาจากการทดลองและการประสบความล้มเหลว ดังนั้น ถ้าเราต้องการที่จะมีการตัดสินใจที่ดี เราจำเป็นที่จะต้องฝึกฝนเกี่ยวกับการออกแบบฐานข้อมูลบ่อยๆเพื่อที่จะได้รับประสบการณ์เพิ่มขึ้นจากการออกแบบฐานข้อมูลในแต่ละครั้ง)

2.2 ส่วนประกอบของแบบจำลองฐานข้อมูล

โดยส่วนใหญ่ของแบบจำลองฐานข้อมูลมักจะประกอบไปด้วย เอ็นทิตี้ (entities), แอทริบิว (attributes), ความสัมพันธ์ (relationships) และ ข้อจำกัดต่างๆ (constraints)

- *เอ็นทิตี้*—ถูกใช้แทนวัตถุต่างๆ สามารถเป็นอะไรก็ได้ เช่น คน สถานที่ สิ่งของ หรือเหตุการณ์ ที่ซึ่งจะเป็นข้อมูลที่จะถูกจัดเก็บอยู่ในฐานข้อมูล เนื่องจากเอ็นทิตี้หนึ่งจะใช้แทนข้อมูลชนิดหนึ่งๆ ดังนั้นแต่ละเอ็นทิตี้จะต้องมีความแตกต่างกัน และแต่ละเอ็นทิตี้จะต้องมีความเป็นเอกลักษณ์เสมอ (มีข้อมูลที่ไม่ซ้ำกัน) ตัวอย่างเช่น เอ็นทิตี้ของลูกค้าจะมีข้อมูลลูกค้าที่แตกต่างกันหลายคน โดยที่ลูกค้าคนหนึ่งจะมีข้อมูลที่มีความแตกต่างลูกค้าคนอื่นๆ—แดง อินทร์พรม จะมีชื่อและนามสกุลที่แตกต่างจากตุ๊กตา อารมณ์ดี เป็นต้น
- *แอทริบิว*—จะแสดงถึงคุณลักษณะของเอ็นทิตี้ ตัวอย่างเช่น ข้อมูลเอ็นทิตี้ลูกค้าถูกอธิบายด้วยแอทริบิวต่างๆ เช่น ชื่อ-นามสกุล เบอร์โทรศัทพ์ ที่อยู่ และเครดิตที่ได้รับจากบริษัท เป็นต้น แอทริบิวในระบบฐานข้อมูลจะมักลักษณะเหมือนกับฟิลด์ในแฟ้มข้อมูล
- *ความสัมพันธ์*—จะแสดงถึงความสัมพันธ์ระหว่างเอ็นทิตี้ ตัวอย่างเช่น ความสัมพันธ์ที่เกิดขึ้นระหว่างลูกค้าและพนักงานขายจะสามารถอธิบายได้ เช่น พนักงานขายหนึ่งคนสามารถดูแลลูกค้าได้หลายคน และลูกค้าคนหนึ่งๆอาจถูกดูแลโดยพนักงานเพียงคนเดียว จากความความสัมพันธ์ดังกล่าว เราสามารถแบ่งรูปแบบความสัมพันธ์ในแบบจำลองข้อมูลออกเป็น 3 ชนิดหลัก คือ 1) one-to-many—สามารถเขียนแทนได้เป็น 1:M หรือ 1..*, 2) many-to-many—สามารถเขียนแทนได้เป็น M:N, M:M หรือ *..* และ 3) one-to-one—สามารถเขียนแทนได้เป็น 1:1 หรือ 1..1 ตามลำดับ เพื่อให้มีความเข้าใจเกี่ยวกับรูปแบบความสัมพันธ์มากขึ้นลองพิจารณาตัวอย่างดังต่อไปนี้
    - ***One-to-many***—จิตรกรคนหนึ่งสามารถสร้างสรรค์ผลงานศิลปะได้หลายชิ้น แต่ผลงานศิลปะชิ้นหนึ่งๆจะสามารถถูกเขียนโดยจิตรกรเพียงคนเดียว จากความสัมพันธ์ข้างต้น เราสามารถเข้าใจโดยนัยได้ว่า จิตรกร (one) จะมีความสัมพันธ์กับผลงานศิลปะหลายชิ้น (many) ดังนั้น ในการออกแบบฐานข้อมูล เราจะแสดงความสัมพันธ์เกี่ยวกับ “จิตรกรสร้างสรรค์ผลงานศิลปะ” ได้เป็น “1:M”

- ***Many-to many***—พนักงานคนหนึ่งสามารถทำงานได้หลายหน้าที่ และหน้าที่ในการทำงานหนึ่งๆสามารถมีพนักงานหลายคนรับผิดชอบ ดังนั้น เราสามารถแสดงความสัมพันธ์เกี่ยวกับ “พนักงานทำงาน” ได้เป็น “M:M”
- ***One-to-one***—ในธุรกิจค้าปลีกที่มีร้านค้าหลายสาขา ที่ซึ่งแต่ละสาขาจะมีผู้จัดการสาขาเพียงหนึ่งคนเท่านั้น ดังนั้น เราสามารถแสดงความสัมพันธ์เกี่ยวกับ “ผู้จัดการสาขามีหน้าที่รับผิดชอบสาขาหนึ่งๆ” ได้เป็น “1:1”

หมายเหตุ การแสดงความสัมพันธ์ระหว่างข้อมูล 2 เอ็นทิตี้จะเป็นการแสดงความสัมพันธ์แบบ 2 ทิศทาง เช่น ลูกค้าคนหนึ่งสามารถสั่งซื้อสินค้าได้หลายครั้ง และ การสั่งซื้อสินค้าครั้งหนึ่งๆจะถูกสั่งซื้อโดยลูกค้าคนหนึ่งๆเท่านั้น

- *ข้อจำกัด*—จะเป็นข้อจำกัดที่เกี่ยวข้องกับข้อมูลที่ซึ่งจะสามารถช่วยเราได้แน่ใจเกี่ยวกับความสมบูรณ์ของข้อมูลได้ โดยส่วนใหญ่ของข้อจำกัดจะถูกแสดงอยู่ในรูปแบบของกฎต่างๆ ตัวอย่างเช่น เงินเดือนของพนักงานควรจะมีค่าอยู่ระหว่าง 6,000 ถึง 350,000 บาท หรือ เกรดเฉลี่ยของนักศึกษาควรจะมีค่าอยู่ระหว่าง 0.00 ถึง 4.00 เป็นต้น

จากส่วนประกอบทั้งหมดของแบบจำลองฐานข้อมูลข้างต้น จะมีคำถามที่ว่า เราจะสามารถระบุเกี่ยวกับเอ็นทิตี้ แอทริบิว ความสัมพันธ์ และข้อจำกัดต่างๆของข้อมูลที่เราสนใจได้อย่างไร?—ในการที่จะระบุส่วนประกอบต่างๆของแบบจำลองฐานข้อมูล เราควรที่จะมีความเข้าใจเกี่ยวกับกฎเกณฑ์ทางธุรกิจ (business rules) ของขอบเขตปัญหาที่เราสนใจเป็นอันดับแรก

2.3 กฎเกณฑ์ทางธุรกิจ

จากส่วนก่อนหน้า เราสามารถเปรียบการออกแบบฐานข้อมูลได้กับการเลือกหรือการกำหนดเอ็นทิตี้ แอทริบิว และความสัมพันธ์ต่างๆระหว่างเอ็นทิตี้ แต่ก่อนที่จะเริ่มกำหนดสิ่งต่างๆข้างต้น ผู้ออกแบบข้อมูลควรจะเข้าใจเกี่ยวกับชนิดของข้อมูลในองค์กร วิธีในการเรียกใช้ข้อมูล ช่วงเวลาที่มีการเรียกใช้ข้อมูล และยังรวมถึงเข้าใจภาพรวมและวิธีการดำเนินธุรกิจอย่างถ่องแท้ ข้อมูลในฐานข้อมูลจะเป็นข้อมูลที่มีความหมายก็ต่อเมื่อข้อมูลเหล่านั้นสามารถสะท้อนถึงกฎเกณฑ์ทางธุรกิจได้อย่างเหมาะสม กฎเกณฑ์ทางธุรกิจมักจะสั้นๆกะทัดรัดและมีคำอธิบายที่ชัดเจนเกี่ยวนโยบาย ขั้นตอน หรือหลักการขององค์กร

กฎเกณฑ์ทางธุรกิจที่ถูกกำหนดไว้อย่างถูกต้องและเหมาะสมจะถูกใช้ในการกำหนดส่วนประกอบต่างๆของแบบจำลองฐานข้อมูลซึ่งก็คือ เอ็นทิตี้ แอทริบิว ความสัมพันธ์ และข้อจำกัดต่างๆ แต่ในทางกลับกัน เมื่อเราทำการพิจารณาถึงส่วนประกอบต่างๆในฐานข้อมูล อาทิเช่น ความสัมพันธ์ระหว่างแอนทิตี้—พนักงานขายสามารถดูแลลูกค้าได้หลายคน และลูกค้าจะถูกดูแลโดยพนักงานขายเพียงคนเดียว—จะทำให้เราทราบถึงกฎเกณฑ์ทางธุรกิจด้วยเช่นกัน

กฎเกณฑ์ทางธุรกิจที่จะช่วยให้สามารถกำหนดเอ็นทิตี้ แอทริบิว ความสัมพันธ์ และข้อจำกัดได้อย่างมีประสิทธิภาพจะต้องเข้าใจง่ายและมีการตีความที่เหมือนกัน ดังแสดงในตัวอย่างดังต่อไปนี้

- ลูกค้าสามารถสั่งซื้อสินค้าได้หลายครั้ง
- ใบสั่งซื้อสินค้าจะเกิดจากลูกค้าคนหนึ่งๆสั่งซื้อสินค้า
- การเปิดอบรมจะกระทำได้ก็ต่อเมื่อมีพนักงานเข้าร่วมอยู่ระหว่าง 10 – 30 คน

กฎเกณฑ์ทางธุรกิจข้างต้นจะใช้ในการกำหนดเอ็นทิตี้ ความสัมพันธ์ และข้อจำกัดได้ดังนี้

- สำหรับ 2 กฎเกณฑ์ทางธุรกิจแรกจะสามารถใช้ในการกำหนด 2 เอ็นทิตี้ คือ ลูกค้าและใบสั่งซื้อสินค้า โดยทั้งสองเอ็นทิตี้นี้จะมีความสันพันธ์กันแบบ 1:M
- สำหรับกฎเกณฑ์ทางธุรกิจสุดท้ายจะสามารถใช้ในการกำหนด 2 เอ็นทิตี้ คือ พนักงานและการอบรม และยังสามารถใช้กำหนดข้อจำกัดของเอ็นทิตี้การอบรมที่ซึ่งการอบรมครั้งหนึ่งๆจะต้องมีพนักงานเข้าร่วมระหว่าง 10 – 30 คน ตามลำดับ

2.3.1 การค้นหาและทำความเข้าใจเกี่ยวกับกฎเกณฑ์ทางธุรกิจ

นอกเหนือจากการมีส่วนช่วยในการกำหนดเอ็นทิตี้ แอทริบิว ความสัมพันธ์ และข้อจำกัดต่างๆ กฎเกณฑ์ทางธุรกิจยังมีประโยชน์ในด้านอื่นๆอีกมากมาย อาทิเช่น 1) ช่วยในการสร้างมาตราฐานให้กับการพิจารณาข้อมูลในองค์กร 2) ช่วยเป็นเครื่องมือในการติดต่อสื่อสารระหว่างผู้ใช้หรือพนักงานกับกลุ่มผู้ออกแบบฐานข้อมูล 3) ช่วยทำให้ผู้ออกแบบฐานข้อมูลมีความเข้าใจเกี่ยวกับธรรมชาติ บทบาทและขอบเขตของข้อมูล และ 4) ช่วยทำให้ผู้ออกแบบฐานข้อมูลมีความเข้าใจเกี่ยวกับกระบวนการดำเนินธุรกิจ เป็นต้น

ถึงแม้ว่ากฎเกณฑ์ทางธุรกิจจะมีประโยชน์ที่หลากหลาย แต่ก็ไม่ใช่ว่าทุกกฎเกณฑ์ทางธุรกิจจะมีประโยชน์กับการออกแบบฐานข้อมูล ตัวอย่างเช่น กฎเกณฑ์ทางธุรกิจที่ว่า “จะไม่มีนักบินใดคนก็ตามที่ทำการบินนานกว่า 10 ชั่วโมงภายในช่วงเวลา 24 ชั่งโมง” กฎเกณฑ์ทางธุรกิจนี้ไม่สามารถใช้ในการสร้างแบบจำลองได้ แต่จะสามารถใช้เป็นเงื่อนไขในแอพพลิเคชันซอร์ฟแวร์ได้

กฎเกณฑ์ทางธุรกิจมักได้มาจากการสัมภาษณ์พนักงานในตำแหน่งต่างๆขององค์กร อาทิเช่น ผู้จัดการแผนกต่างๆ ผู้วางแผนและวางนโยบาย และเอกสารคู่มือที่บ่งบอกถึงวิธีการดำเนินธุรกิจขององค์กร รวมถึงมาตราฐานและการดำเนินงานต่างๆ แต่อย่างไรก็ตาม การสอบถามหรือได้รับกฎเกณฑ์ทางธุรกิจจากพนักงานทั่วๆไปอาจมีความไม่น่าเชื่อถือเท่าไหร่นัก เนื่องจากพนักงานแต่ละคนก็มีความเข้าใจในกฎเกณฑ์ธุรกิจที่แตกต่างกัน และเมื่อเราทำการสัมภาษณ์พนักงานหลายๆคนที่ทำงานในตำแหน่งเดียวกันอาจทำให้เราได้รับคำตอบหรือกฎทางธุรกิจที่แตกต่างกันก็เป็นได้ ดังนั้นเมื่อเกิดเหตุการณ์ดังกล่าวขึ้น ผู้ออกแบบฐานข้อมูลจะต้องขจัดความแตกต่างของกฎเกณฑ์ทางธุรกิจที่ได้รับ จากนั้นทำการตรวจสอบกฎเกณฑ์ที่ได้ทำการพิจารณาแล้วอีกครั้งหนึ่งเพื่อทำให้แน่ใจได้ว่ากฎทางธุรกิจที่ได้จะเป็นกฎที่มีความถูกต้องเหมาะสม

2.3.2 การปรับเปลี่ยนกฎเกณฑ์ทางธุรกิจไปเป็นส่วนประกอบของแบบจำลองข้อมูล

หลังจากเราได้รับกฎเกณฑ์ทางธุรกิจจากการสัมภาษณ์พนักงานระดับต่างๆในองค์แล้ว ขั้นตอนต่อไปเราจะต้องทำการปรับเปลี่ยนกฎเกณฑ์ทางธรุกิจไปเป็นเอ็นทิตี้ แอทริบิว ความสัมพันธ์ และข้อจำกัดต่างๆ โดยการปรับเปลี่ยนคำนามคำนามที่ปรากฏอยู่ในแต่ละกฎทางธุรกิจให้เป็นเอ็นทิตี้ของแบบจำนวนลองข้อมูล และทำการปรับเปลี่ยนคำกริยาที่เกี่ยวข้องกับคำนามที่ปรากฏก่อนหน้าไปเป็นความสัมพันธ์ระหว่างเอนทิตี้ ตัวอย่างเช่น กฎทางธุรกิจ “ลูกค้าคนหนึ่งๆสามารถเขียนใบสั่งของได้หลายใบ” ที่จะประกอบไปด้วย 2 คำนาม คือ ลูกค้าและใบสั่งของ และคำกริยา 1 คำ คือ เขียน ตามลำดับ จากกฎทางธุรกิจดังกล่าว เราสามารถสรุปได้ว่า 1) ลูกค้าและใบสั่งของจะเป็นวัตถุที่เราสนใจและสามารถปรับเปลี่ยนให้เป็นเอ็นทิตี้ได้ และ 2) “เขียน” จะเป็นความสัมพันธ์ระหว่างลูกค้ากับใบสั่งสินค้า

ในการที่จะระบุถึงความสัมพันธ์ระหว่างเอ็นทิตี้ เราจะต้องทำการพิจารณาความสัมพันธ์ทั้งสองด้าน (bidirectional relationship) ตัวอย่างเช่น กฎทางธุรกิจ “ลูกค้าคนหนึ่งๆสามารถเขียนใบสั่งของได้หลายใบ” จะมีส่วนเติมเต็มอีกกฎทางธุรกิจหนึ่ง คือ “ใบสั่งสินค้าใบหนึ่งๆจะถูกเขียนโดยลูกค้าคนหนึ่งๆเท่านั้น” จากกฎทางธุรกิจทั้งสองจะทำให้เราสามารถพิจารณาความสัมพันธ์ได้ทั้งสองด้านของเอ็นทิตี้ลูกค้าและเอ็นทิตี้ใบสั่งได้ โดยรูปแบบความสัมพันธ์จะเป็นแบบ 1:M ด้วยเหตุนี้ การระบุถึงความสัมพันธ์ระหว่าง 2 เอ็นทิตี้ A และ B ใดๆก็ตาม เราควรที่จะต้องตั้งคำถาม 2 คำถาม คือ 1) มีข้อมูลในเอ็นทิตี้ A เป็นจำนวนเท่าไร ที่มีความสัมพันธ์กับข้อมูลหนึ่งๆในเอ็นทิตี้ B ? และ 2) ข้อมูลในเอ็นทิตี้ B เป็นจำนวนเท่าไร ที่มีความสัมพันธ์กับข้อมูลหนึ่งๆในเอ็นทิตี้ A ? ตามลำดับ

2.3.3 การตั้งชื่อ

ระหว่างการปรับเปลี่ยนกฎเกณฑ์ทางธุรกิจให้เป็นเอ็นทิตี้ แอทริบิว ความสัมพันธ์ และข้อจำกัดต่างๆ ขั้นตอนการทำงานจะรวมถึงขั้นตอนการตั้งชื่อวัตถุต่างๆที่ซึ่งจะทำให้วัตถุที่เราทำการพิจารณามีความเป็นเอกลักษณ์และแตกต่างจากวัตถุอื่นๆในขอบเขตของปัญหาที่เราพิจารณา ดังนั้น เราควรจะต้องเอาใจใส่กับวิธีการในการตั้งชื่อวัตถุต่างๆที่เราจะทำการพิจารณา

ชื่อของเอ็นทิตี้ควรจะสามารถอธิบายได้ถึงวัตถุที่เราสนใจและควรที่จะใช้คำศัพท์ที่ผู้งานระบบฐานข้อมูลคุ้นเคย ในส่วนของการตั้งชื่อแอทริบิวก็ควรที่จะสามารถอธิบายได้ถึงข้อมูลเช่นกัน โดยเราอาจทำการตั้งชื่อด้วยคำนำหน้าของแอทริบิวหรือคำย่อของแอทริบิวนั้นๆ ตัวอย่างเช่น ในเอ็นทิตี้ CUSTOMER จะมีข้อมูลที่บ่งบอกถึงเครดิตที่ลูกค้าได้รับ (customer’s credit limit) ดังนั้น เราอาจทำการตั้งชื่อด้วยการนำคำนำหน้าของเอ็นทิตี้มารวมกับข้อมูลหลักที่สำคัญได้เป็น “CUS_CREDIT_LIMIT” ที่ซึ่งจะทำให้เราสามารถเข้าใจความหมายได้โดยง่าย การตั้งชื่อที่เหมาะสมจะเป็นการช่วยพัฒนาให้แบบจำลองที่เราสร้างขึ้นมีความสามารถในการสื่อสารกันระหว่างผู้ออกแบบฐานข้อมูล ผู้เขียนโปรแกรม และผู้ใช้ระบบฐานข้อมูลได้ดียิ่งขึ้น

2.4 วิวัฒนาการของแบบจำลองข้อมูล

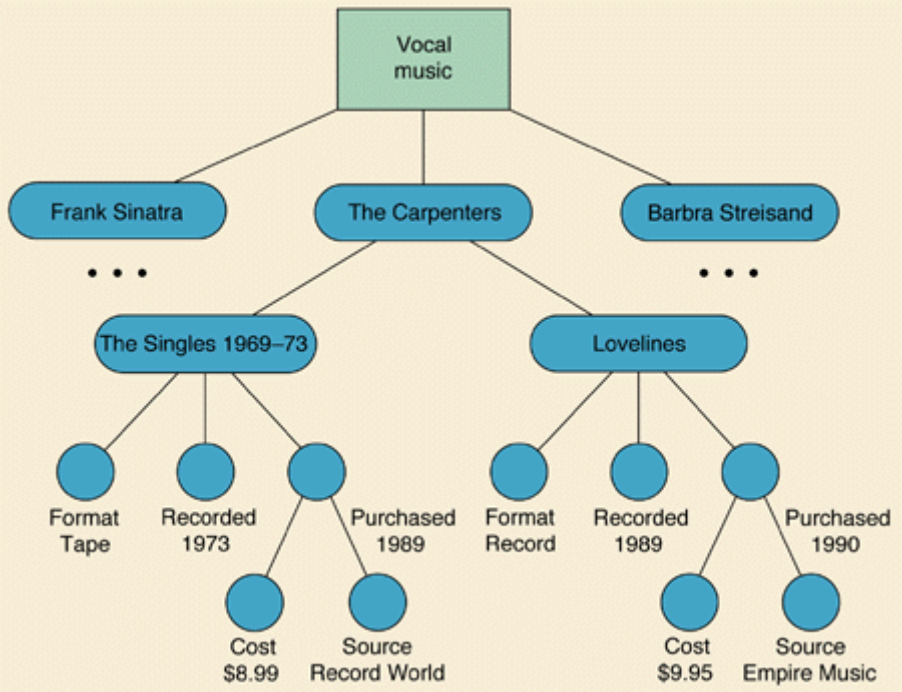
การค้นหาวิธีการในการจัดการข้อมูลที่ดีจะนำไปสู่การพัฒนาแบบจำลองข้อมูลที่หลากหลายที่ซึ่งจะมีวัตถุประสงค์ที่จะพัฒนาข้อบกพร่องของระบบแฟ้มข้อมูล โดยในการศึกษาเกี่ยวกับแบบจำลองข้อมูลเหล่านี้จะทำให้เราทราบถึงชนิดของโครงสร้างที่มีการประยุกต์ใช้ รวมถึงเทคโนโลยีที่ใช้ในการสร้างโครงสร้างเหล่านั้น แบบจำลองข้อมูลที่จะทำการศึกษาจะเป็นแบบจำลองที่มีลักษณะเป็นแผนภาพเช่นเดียวกับแบบจำลองฐานข้อมูล ดังนั้นเราจะทำการศึกษาแบบจำลองต่างๆที่สำคัญ ดังต่อไปนี้ (ดังแสดงในรูปที่ 2.1)

GENERATION	TIME	DATA MODEL	EXAMPLES	COMMENTS
First	1960s–1970s	File system	VMS/VSAM	Used mainly on IBM mainframe systems Managed records, not relationships
Second	1970s	Hierarchical and network	IMS ADABAS IDS-II	Early database systems Navigational access
Third	Mid-1970s to present	Relational	DB2 Oracle MS SQL-Server MySQL	Conceptual simplicity Entity relationship (ER) modeling and support for relational data modeling
Fourth	Mid-1980s to present	Object-oriented Object/relational (O/R)	Versant Objectivity/DB DB/2 UDB Oracle 11g	Object/relational supports object data types Star Schema support for data warehousing Web databases become common
Next generation	Present to future	XML Hybrid DBMS	dbXML Tamino DB2 UDB Oracle 11g MS SQL Server	Unstructured data support O/R model supports XML documents Hybrid DBMS adds an object front end to relational databases

รูปที่ 2.1 วิวัฒนาการของแบบจำลองข้อมูล

2.4.1 แบบจำลองแบบลำดับชั้นและแบบจำลองแบบเครือข่าย

แบบจำลองแบบลำดับชั้น (hierarchical model) ถูกพัฒนาขึ้นในยุค 1960 เพื่อจัดการกับข้อมูลปริมาณมากๆของโครงการผลิตที่มีความซับซ้อน อาทิ เช่น จรวดอพอลโลที่ลงจอดบนดวงจันทร์ในปี 1969 โครงสร้างพื้นฐานของแบบจำลองแบบลำดับชั้นจะสามารถแสดงได้ด้วย upside-down tree และโครงสร้างลำดับชั้นจะประกอบด้วยลำดับชั้นต่างๆ (level หรือ segment) ที่ซึ่งจะเหมือนกับชนิดของเรคคอร์ดในระบบแฟ้มข้อมูล โดยลำดับชั้นหนึ่งๆอาจประกอบไปด้วยหลาย segment ที่ซึ่ง segment ที่อยู่ระดับบนจะถูกเรียกว่า parent แต่ segment ที่อยู่ถัดลงมาจะเรียกว่า children ของ parent (หมายเหตุ parent หนึ่งๆ สามารถมีได้หลาย child แต่ในทางกลับกัน child หนึ่งๆจะสามารถมีได้เพียง parent เดียว) (ดังแสดงตัวอย่างดังรูปที่ 2.2)

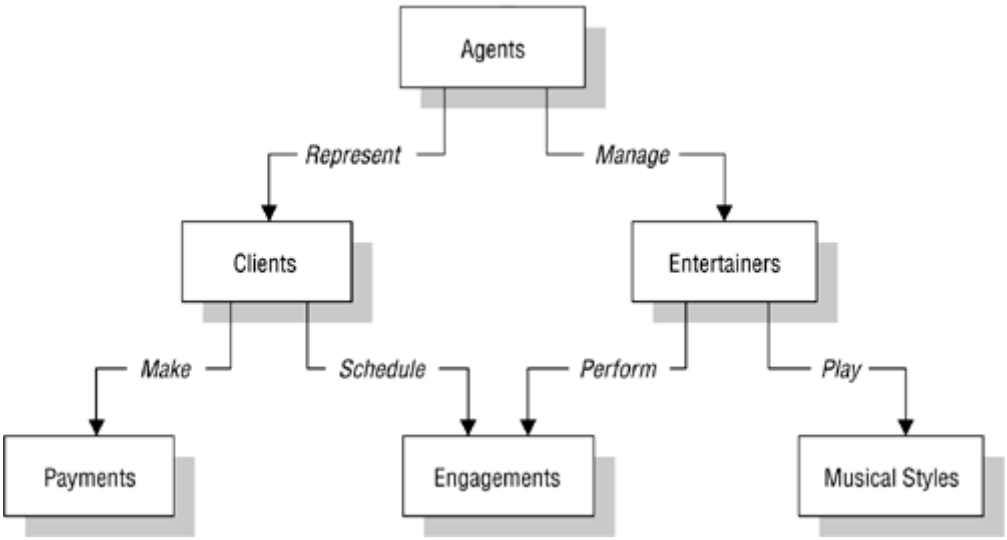


รูปที่ 2.2 ตัวอย่างแบบจำลองข้อมูลแบบหลายลำดับชั้น

แบบจำลองแบบเครือข่าย (network model) ได้ถูกสร้างขึ้นเพื่อเป็นตัวแทนของความสัมพันธ์ของข้อมูลที่มีความซับซ้อน แนวคิดสำหรับการพัฒนาแบบจำลองแบบเครือข่ายก็เพื่อที่จะพัฒนาประสิทธิภาพการทำงานให้เหนือกว่าแบบจำลองแบบหลายลำดับชั้น โดยจะมีการปรับปรุงประสิทธิภาพการทำงานของฐานข้อมูลและการกำหนดมาตราฐานต่างๆของฐานข้อมูล

ภายใต้แบบจำลองแบบเครือข่าย ผู้ใช้จะมองเห็นแบบจำลองดังกล่าวเป็นกลุ่มของเรคคอร์ดที่มีความสัมพันธ์เป็นแบบ 1:M นอกจากนั้น แบบจำลองแบบเครือข่ายจะยอมให้เรคคอร์ดหนึ่งๆมีได้มากกว่า 1 parent อีกด้วย (ดังแสดงในรูปที่ 2.3) แนวความคิดหลักของแบบจำลองแบบเครือข่ายสามารถกำหนดและนิยามได้ดังต่อไปนี้

- ***Schema***—จะเป็นโครงสร้างข้อมูลหรือแนวคิดของฐานข้อมูล รวมถึงความสัมพันธ์ของข้อมูลในแต่ละเอ็นทิตี้ ว่ามีความสัมพันธ์กันอย่างไร
- ***Subschema***—จะเป็นฐานข้อมูลส่วยย่อยๆที่จะสร้างข้อมูลที่ต้องการจากข้อมูลที่ถูกจัดเก็บอยู่ในฐานข้อมูล
- ***ภาษาในการจัดการข้อมูล (data management language, DML)***—จะเป็นภาษาที่ใช้นิยามหรือกำหนดสภาพแวดล้อมที่ข้อมูลจะถูกจัดการและถูกใช้งานภายใต้ฐานข้อมูล
- ***ภาษาในการนิยามข้อมูล (data definition language, DDL)***—จะเป็นภาษาที่ช่วยให้ผู้ดูและระบบฐานข้อมูลสามารถกำหนดส่วนประกอบของ schema ได้



รูปที่ 2.3 ตัวอย่างแบบจำลองแบบเครือข่าย

แต่อย่างไรก็ดี การประยุกต์ใช้แบบจำลองแบบเครือข่ายกลายเป็นเรื่องที่ยุ่งยากและไม่เหมาะสมเมื่อมีความต้องการข้อมูลสารสนเทศที่มากขึ้น และความต้องการฐานข้อมูลและแอพพลิเคชันที่มีความซับซ้อนมากขึ้น เนื่องจากแบบจำลองนี้ขาดความสามารถในการเข้าถึงข้อมูลแบบ ad hoc ที่ใช้สำหรับสร้างรายงานต่างๆ ดังนั้น จากปัญหาที่เกิดขึ้น จึงเป็นเหตุให้มีการพัฒนาแบบจำลองข้อมูลเชิงสัมพันธ์ในช่วงยุคทศวรรษ 1980

2.4.2 แบบจำลองข้อมูลเชิงสัมพันธ์

แบบจำลองข้อมูลเชิงสัมพันธ์ (relational model) ถูกพัฒนาขึ้นโดย E. F. Codd ในช่วงทศวรรษ 1970 ที่ซึ่งจะสามารถลดปัญหาและอุปสรรคของการสื่อสารระหว่างผู้ออกแบบและผู้ใช้งานระบบฐานข้อมูล โดยแนวความคิดพื้นฐานของแบบจำลองข้อมูลเชิงสัมพันธ์จะเป็นแนวคิดทางคณิตศาสตร์เรื่องความสัมพันธ์ (relation) ที่ซึ่งเราสามารถพิจารณาความสัมพันธ์เป็นเมทริกซ์หรือ***ตารางของข้อมูล***ที่ประกอบไปด้วยการรวมกันระว่างแถว (rows) และ คอลัมน์ (columns) แต่ละแถวของตารางจะถูกเรียกว่า ***tuple*** และแต่ละคอลัมน์จะถูกแทนด้วยแอทริบิวต่างๆ

การสร้างแบบจำลองข้อมูลเชิงสัมพันธ์จะสามารถดำเนินการผ่าน***ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (relational database management system, RDBMS)*** ที่ซึ่งจะมีความซับซ้อนค่อนข้างสูง RDBMS จะสามารถดำเนินการฟังก์ชั่นพื้นฐานได้เหมือนกับระบบจัดการฐานข้อมูลแบบลำดับชั้นและแบบเครือข่าย แต่ RDBMS จะมีการเพิ่มเติมประโยชน์ที่ค่อนข้างสำคัญ คือ ความสามารถในการปิดบังความซับซ้อนของแบบจำลองข้อมูลเชิงสัมพันธ์จากผู้ใช้งานฐานข้อมูล โดย RDBMS จะจัดการเกี่ยวกับการทำงานเชิงกายภาพทั้งหมด (ขั้นตอนการจัดเก็บและเรียกดูข้อมูลทั้งหมด) ที่ซึ่งจะทำให้ผู้ใช้งานจะมองเห็นฐานข้อมูลเชิงสัมพันธ์เป็นเพียงกลุ่มของตารางข้อมูลที่ใช้ในการจัดเก็บข้อมูลเท่านั้น และเมื่อไรก็ตามที่ผู้ใช้ต้องการที่จะดำเนินการกับข้อมูล ไม่ว่าจะเป็นการเพิ่ม ลบ และแก้ไขข้อมูล ก็จะสามารถทำได้โดยง่ายผ่านการเรียกใช้คิวรี

ตารางต่างๆในแบบจำลองเชิงสัมพันธ์ จะมีความสัมพันธ์กับตารางอื่นๆผ่านการแชร์แอทริบิวที่มีความเหมือนกัน ตัวอย่างเช่น ตารางลูกค้า (CUSTOMER) ในรูปที่ 2.4 จะมีข้อมูลรหัสพนักงานขายที่ดูแลพวกเขา ที่ซึ่งข้อมูลรหัสพนักงานก็จะถูกจัดเก็บในตารางพนักงานขาย (AGENT) ด้วยเช่นกัน การเชื่อมโยงกันระหว่างตารางข้อมูลลูกค้าและตารางพนักงานขายจะทำให้เราทราบถึงการเกี่ยวข้องกันของลูกค้าและพนักงานขาย ตัวอย่างเช่น เมื่อเราพิจารณาของมูลของ Anne Farriss จะทำให้เราทราบว่าจะมี Alex Alby เป็นพนักงานขายผู้ดูแล Anne เนื่องจากในข้อมูลเรคคอร์ดของ Anne Farriss จะประกอบไปด้วยแอทริบิว AGENT_CODE ที่มีค่าเป็น 501 และเมื่อนำค่า 501 มาเปรียบเทียบกับ AGENT_CODE ในตารางพนักงานขายจะทำให้เราทราบว่า 501 เป็นรหัสพนักงานของ Alex Alby เป็นต้น การเชื่อมโยงความสัมพันธ์ระหว่างตารางข้อมูลจะทำให้เราสามารถทราบถึงความสัมพันธ์ของข้อมูลระหว่างตารางต่างๆได้โดยง่าย

Table name: AGENT (first six attributes) **Database name: Ch02_InsureCo**

AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

Link through AGENT_CODE

Table name: CUSTOMER

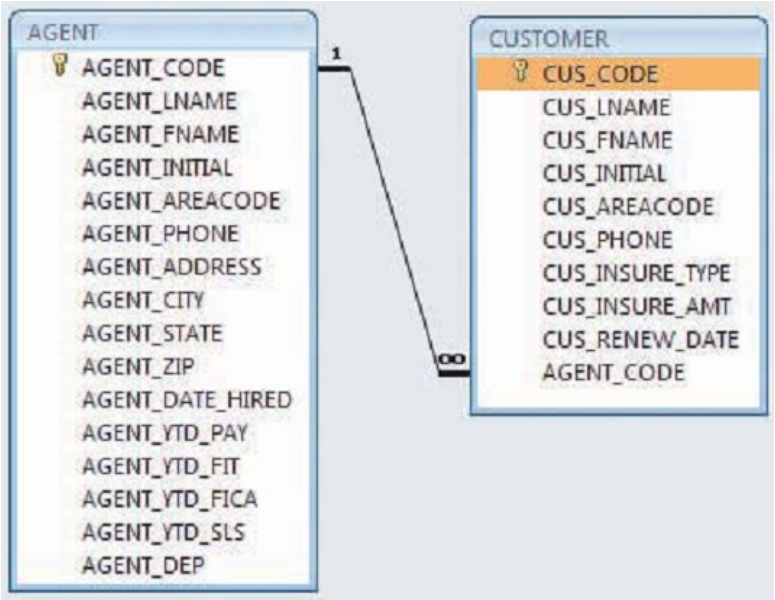
CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_INSURE_TYPE	CUS_INSURE_AMT	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	T1	100.00	05-Apr-2010	502
10011	Dunne	Leona	K	713	894-1238	T1	250.00	16-Jun-2010	501
10012	Smith	Kathy	W	615	894-2285	S2	150.00	29-Jan-2011	502
10013	Olowski	Paul	F	615	894-2180	S1	300.00	14-Oct-2010	502
10014	Orlando	Myron		615	222-1672	T1	100.00	28-Dec-2010	501
10015	O'Brian	Amy	B	713	442-3381	T2	850.00	22-Sep-2010	503
10016	Brown	James	G	615	297-1228	S1	120.00	25-Mar-2011	502
10017	Williams	George		615	290-2556	S1	250.00	17-Jul-2010	503
10018	Farriss	Anne	G	713	382-7185	T2	100.00	03-Dec-2010	501
10019	Smith	Olette	K	615	297-3809	S2	500.00	14-Mar-2011	503

รูปที่ 2.4 การเชื่อมโยงความสัมพันธ์ระหว่างข้อมูลในตารางต่างๆ

แบบจำลองข้อมูลจะมีการประยุกต์ใช้***แผนผังความสัมพันธ์ (Relational diagram)*** ในการแสดงถึงเอ็นทิตี้ แอทริบิวของเอ็นทิตี้ต่างๆ และความสัมพันธ์ระหว่างเอ็นทิตี้ต่างๆ (ทั้งแบบ 1:1, 1:M และ M:M) ของแบบจำลองข้อมูลเชิงสัมพันธ์ ตัวอย่างเช่น รูปที่ 2.5 จะเป็นแผนผังความสัมพันธ์ที่แสดงถึงเอ็นทิตี้ลูกค้าและเอ็นทิตี้พนักงานขายที่มีความสัมพันธ์กันแบบ 1:M โดยจากรูปจะมีการใช้สัญลักษณ์ ∞ ที่เอ็นทิตี้ลูกค้าที่ซึ่งจะแสดงความสัมพันธ์ในฝั่งของ “many” เนื่องจากพนักงานขายคนหนึ่งๆสามารถดูแลลูกค้าได้ขายคน แต่สำหรับเอ็นทิตี้พนักงานขายจะแสดงความสัมพันธ์ฝั่ง “1” เนื่องจากลูกค้าคนหนึ่งๆจะสามารถมีพนักงานขายดูแลได้คนเดียวเท่านั้น

การจัดเก็บข้อมูลในฐานข้อมูลเชิงสัมพันธ์ (relational database) จะมีการจัดเก็บข้อมูลไว้ในตารางที่ซึ่งจะมีลักษณะคล้ายกับการจัดเก็บข้อมูลด้วยแฟ้มข้อมูล แต่จะมีจุดหนึ่งทั้งสองวิธีมีความแตกต่างกัน นั่นคือ ตารางหนึ่งๆในฐานข้อมูลเชิงสัมพันธ์จะมีลักษณะเป็นแบบไม่ขึ้นกับข้อมูลและไม่ขึ้นกับโครงสร้างเนื่องจากตารางนั้นๆจะทำการเก็บเพียงแค่โครงสร้างเชิงตรรกะ (logical structure) ที่ซึ่งจะสนใจแค่ว่าตารางหนึ่งๆ ประกอบไปด้วยข้อมูลแอทริบิวอะไรบ้าง แต่การสลับตำแหน่งของแอทริบิวไม่มีผล แต่สำหรับการจัดเก็บข้อมูลในแฟ้มข้อมูลจะมีลักษณะเป็นแบบขึ้นกับโครงสร้างและขึ้นกับข้อมูล (ดังอธิบายในบทที่ 1) นอกจากเหตุผล

ข้างต้น ยังมีอีกเหตุผลหนึ่งที่การใช้แบบจำลองเชิงสัมพันธ์ได้รับความนิยมนั่นคือ RDBMS จะมีการประยุกต์ใช้ภาษาคิวรี (Structured Query Language, SQL) ที่มีประสิทธิภาพและมีความยืดหยุ่นในการเปลี่ยนคิวรีไปเป็นขั้นตอนวิธีในการเรียกดูข้อมูล



รูปที่ 2.5 ตัวอย่างแผนผังความสัมพันธ์ (relational diagram)

จากแนวความคิดของแบบจำลองข้อมูลเชิงสัมพันธ์และฐานข้อมูลเชิงสัมพันธ์จะทำให้แอพพลิเคชันที่สร้างขึ้นจากการประยุกต์ใช้ SQL และฐานข้อมูลเชิงสัมพันธ์ จะประกอบไปด้วย 3 ส่วน คือ 1) อินเทอร์เฟซ—จะทำหน้าที่เป็นตัวกลางระหว่างผู้ใช้และข้อมูล 2) ตารางข้อมูลในฐานข้อมูล—จะถูกใช้สำหรับจัดเก็บข้อมูลในฐานข้อมูล ที่ซึ่งแต่ละตารางจะเป็นตารางที่ไม่ขึ้นกับตารางอื่นๆ แต่แถวของตารางต่างๆ (ตารางที่ต่างกัน) จะเกี่ยวเนื่องกันด้วยแอทริบิวที่เหมือนกันได้ และ 3) เครื่องมือในการประมวลผล SQL (SQL engine)—จะเป็นส่วนหนึ่งในระบบจัดการฐานข้อมูลที่มีหน้าที่ประมวลผลคิวรี (คำสั่งร้องขอข้อมูล) ต่างๆ ซึ่งโดยปกติแล้วผู้ใช้งานระบบฐานข้อมูลจะใช้ SQL ในการสร้างโครงสร้างของตารางต่างๆ รวมถึงการจัดการการเข้าถึงข้อมูลและการดูแลจัดการต่างๆเกี่ยวกับตารางข้อมูล จากการใช้งาน SQL ดังกล่าว SQL engine จะทำหน้าที่ในการประมวลผลทุกๆคิวรีที่ถูกร้องขอจากผู้ใช้งาน

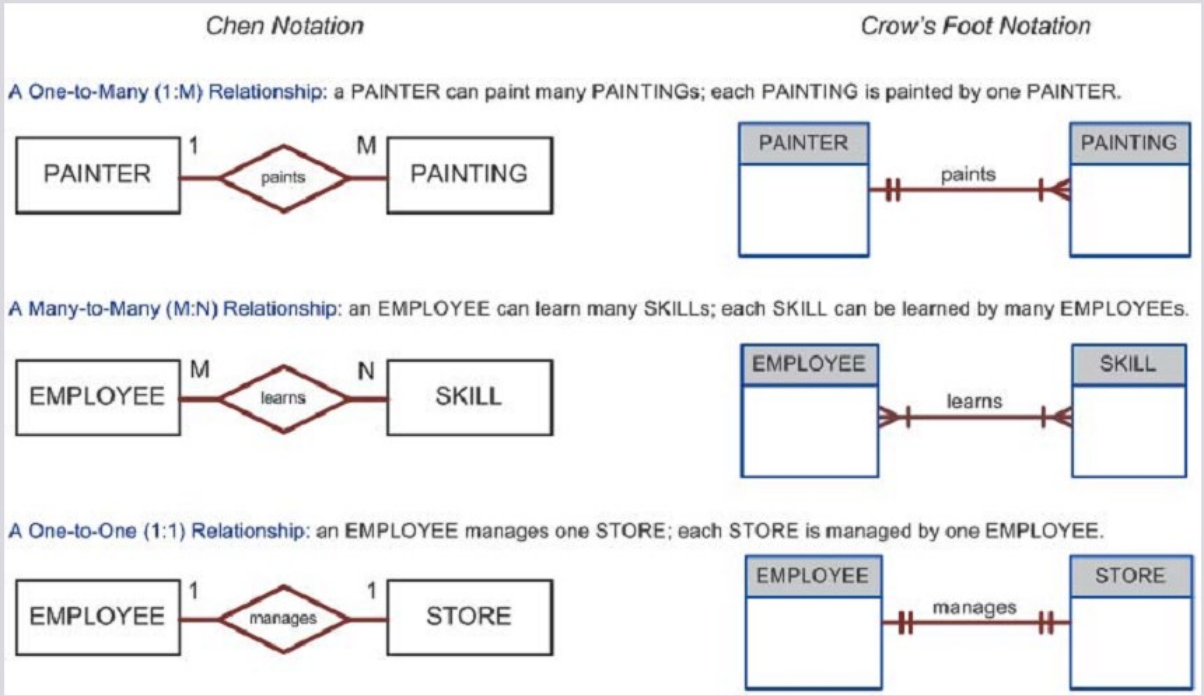
2.4.3 แบบจำลองข้อมูลความสัมพันธ์เอ็นทิตี้

ถึงแม้ว่าแบบจำลองข้อมูลเชิงสัมพันธ์จะถูกพัฒนาเพื่อปรับปรุงการทำงานของแบบจำลองแบบลำดับชั้นและแบบเครือข่าย แต่อย่างไรก็ตามแบบจำลองข้อมูลเชิงสัมพันธ์ยังขาดเค้าโครงที่จะทำให้การสร้างเครื่องมือสำหรับออกแบบฐานข้อมูลได้อย่างมีประสิทธิภาพ และด้วยเนื่องจากการแสดงโครงสร้างของแบบจำลองด้วยแผนภาพจะทำให้ผู้ออกแบบและผู้ใช้งานฐานข้อมูลสามารถเข้าใจแบบจำลองข้อมูลได้ง่ายกว่าการใช้ตัวอักษร ด้วยเหตุนี้ แบบจำลองความสัมพันธ์เอ็นทิตี้ (entity relationship (ER) model, ERM) ได้ถูกคิดค้นและถูกใช้เป็นภาษาฐานในการสร้างแบบจำลองข้อมูลสำหรับการออกแบบฐานข้อมูล

แบบจำลองความสัมพันธ์เอ็นทิตี้ถูกคิดค้นในปี 1976 โดย Peter Chen ที่ซึ่งจะใช้แผนภาพในการแสดงถึงเอ็นทีตี้และความสัมพันธ์ต่างๆในโครงสร้างของฐานข้อมูล แบบจำลองความสัมพันธ์จะมักจะแสดงอยู่ในรูปของแผนผังความสัมพันธ์เอ็นทีตี้ (entity relationship diagram, ERD) ที่ซึ่งจะเป็นแผนภาพที่จะแสดงส่วนประกอบต่างๆของแบบจำลองข้อมูล แบบจำลองความสัมพันธ์เอ็นทิตี้จะประกอบไปด้วย 2 ส่วน คือ

- *เอ็นทิตี้*—จะเป็นส่วนที่ใช้สำหรับจัดเก็บข้อมูล เอ็นทิตี้ใน ERD จะถูกแสดงด้วยสี่เหลี่ยมและชื่อของเอ็นทิตี้มักจะถูกเขียนด้วยตัวอักษรใหญ่อยู่ตรงกลางสี่เหลี่ยม แต่ละเอ็นทิตี้จะสามารถอธิบายได้ด้วยเซตของแอทริบิวที่บ่งบอกถึงคุณลักษณะของเอ็นทิตี้นั้นๆ ตัวอย่างเช่น เอ็นทิตี้พนักงานจะมีแอทริบิว รหัสพนักงาน หมายเลขบัตรประชาชน ชื่อ และนามสกุลเป็นคุณลักษณะหรือข้อมูลที่บ่งบอกถึงตัวตนของพนักงานคนหนึ่งๆ เมื่อเราทำการประยุกต์ใช้ ERD เอ็นทิตี้จะแสดงถึงตารางข้อมูลที่มีการเชื่อมโยงความสัมพันธ์ โดยแต่ละแถวในตารางจะเป็นค่าข้อมูลจริง (entity instance หรือ entity occurrence)
- *ความสัมพันธ์*—จะเป็นสิ่งที่ใช้อธิบายเกี่ยวกับความสัมพันธ์หรือการเชื่อมโยงกันของข้อมูล ความสัมพันธ์ส่วนใหญ่จะใช้ในการอธิบายความสัมพันธ์ของ 2 เอ็นทีตี้ที่ซึ่งจะแสดงผ่านความสัมพันธ์ทั้งแบบ one-to-many (1:M), many-to-many (M:M) และ one-to-one (1:1) ตามลำดับ ในแบบจำลองข้อมูลความสัมพันธ์เอ็นทิตี้จะมีการกำหนดชื่อความสัมพันธ์เพื่อแสดงถึงการกระทำ (action) ที่เป็นตัวกลางในการเชื่อมโยงความสัมพันธ์ของเอ็นทิตี้ต่างๆ ตัวอย่างเช่น จิตรกร***เขียน***ภาพ พนักงาน***เรียนรู้***ทักษะต่างๆ หรือ พนักงาน***จัดการ***ร้านค้าสาขาต่างๆ เป็นต้น

จากส่วนประกอบทั้ง 2 ของแบบจำลองข้อมูลความสัมพันธ์เอ็นทิตี้ เป็นเหตุให้ Chen และ Crow's Foot ได้คิดค้นวิธีในการแสดงความสัมพันธ์ระหว่างเอ็นทิตี้ ดังแสดงในรูปที่ 2.6 ทางฝั่งซ้ายของรูปจะเป็นแผนผังความสัมพันธ์เอ็นทีตี้ที่ถูกคิดค้นโดย Chen ที่ซึ่งเอ็นทิตี้จะถูกเขียนอยู่ในสี่เหลี่ยม และความสัมพันธ์ที่เชื่อมโยงระหว่างเอ็นทีตี้จะถูกเขียนอยู่ในสี่เหลี่ยมคางหมูที่เชื่อมต่อเส้นตรงเข้าหาเอ็นทิตี้ที่มีความสัมพันธ์กัน นอกจากนั้นยังมีการใช้ตัวอักษร 1 (one) และ M (many) ในการแสดงถึงลักษณะของความสัมพันธ์ ในส่วนของฝั่งขวาของรูปจะเป็นแผนผังความสัมพันธ์เอ็นทีตี้ที่ถูกคิดค้นโดย Crow's Foot ที่ซึ่งเอ็นทิตี้จะถูกเขียนอยู่ในสี่เหลี่ยมเช่นกัน แต่ในการแสดงถึงความสัมพันธ์ระหว่างเอ็นทิตี้จะใช้เส้นตรงเชื่อมกับเอ็นทิตี้ที่มีความสัมพันธ์กัน โดยชื่อของความสัมพันธ์จะถูกเขียนอยู่ข้างบนเส้น และจะใช้เส้นตรงคาดเส้นความสัมพันธ์ 1 เส้นเพื่อแสดงถึง 1 (one) และใช้สัญลักษณ์สามแฉกเพื่อแสดงถึง M (many)



รูปที่ 2.6 การนิยาม ERD จากแนวคิดของ Chen และ Crow's Foot

2.4.4 แบบจำลองข้อมูลเชิงวัตถุ

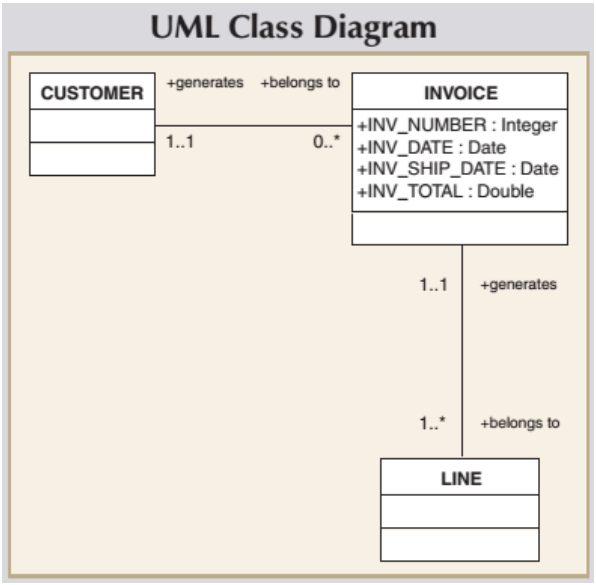
แบบจำลองข้อมูลเชิงวัตถุ (object-oriented data model, OODM) จะทำการจัดเก็บข้อมูลและความสัมพันธ์ต่างๆของข้อมูล ไว้ใน object ในการจัดเก็บข้อมูลต่างๆ OODM จะทำการประยุกต์ใช้ระบบจัดการฐานข้อมูลเชิงวัตถุ (object-oriented database management system, OODBMS) ที่ซึ่งจะทำให้เราสามารถจัดเก็บและเรียกดูข้อมูลได้อย่างสะดวกและรวดเร็ว

OODM จะมีแนวทางในการกำหนดและใช้เอ็นทิตี้ที่แตกต่างจากแบบจำลองข้อมูลอื่นๆ โดย object หนึ่งๆจะถูกอธิบายได้ด้วยข้อเท็จจริงหนึ่งๆ และยังทำการจัดเก็บความสัมพันธ์ระหว่าง object ที่พิจารณากับ object อื่นๆอีกด้วย โดยการจัดเก็บข้อเท็จจริงใน object จะทำให้เราทราบถึงความหมายได้มากขึ้นและสามารถแสดงถึงข้อมูลที่มีความซับซ้อนได้มากขึ้นด้วยเช่นกัน ด้วยเหตุนี้เราจึงได้สามารถกล่าวได้ว่า OODM จะเป็นแบบจำลองข้อมูลที่แสดงความหมาย (semantic data model) เนื่องจากการที่เราได้รับทราบความหมายที่มากขึ้นนั่นเอง นอกจากนั้น OODM ได้มีการพัฒนาให้ object หนึ่งๆสามารถบรรจุไปด้วยการดำเนินการทั้งหมดที่สามารถเกิดขึ้นได้กับ object ได้ แบบจำลองเชิงวัตถุจะประกอบไปด้วยส่วนย่อยต่างๆ ดังนี้

- *Object*---จะคล้ายคลึงกับเอ็นทิตี้ในแบบจำลองข้อมูลต่างๆ แต่ object หนึ่งๆจะใช้ในการแสดงถึงการเกิดขึ้นครั้งหนึ่งๆของข้อมูล (one occurrence/instance) ในเอ็นทิตี้เท่านั้น
- *แอทริบิว*—จะเป็นสิ่งที่ใช้อธิบายคุณลักษณะของ object ตัวอย่างเช่น object ของบุคคลหนึ่งๆ (Person) อาจประกอบไปด้วยแอทริบิว รหัสประจำตัวประชาชน ชื่อ-สกุล ที่อยู่ และ วันเกิด เป็นต้น

- *คลาส (Class)*—จะเป็นกลุ่มของ objects ที่มีความเหมือนกันในแอทริบิว เราสามารถพิจารณาได้ว่าคลาสหนึ่งๆจะมีลักษณะเหมือนกับเอ็นทิตี้ในแบบจำลองข้อมูลเชิงสัมพันธ์ แต่คลาสจะมีความแตกต่างจากเอ็นทิตี้ตรงคลาสหนึ่งๆจะมีการจัดเก็บ method รวมอยู่ด้วย ภายใต้แบบจำลองเชิงวัตถุ method จะถูกพิจารณาเป็นพฤติกรรมของ object (object's behavior) โดย method ในคลาสจะเป็นการดำเนินต่างๆกับ object ในคลาสนั้นๆ ตัวอย่างเช่น การเลือกชื่อของบุคคล การเปลี่ยนชื่อบุคคล การพิมพ์ที่อยู่ของบุคคล เป็นต้น
- *Class hierarchy*—จะเป็นลำดับชั้นของคลาสที่เกิดจากการจัดลำดับชั้นของคลาสต่างๆ ลำดับชั้นของคลาสจะมีลักษณะเป็น upside-down tree (คล้ายกับแบบจำลองแบบลำดับชั้น) โดยแต่ละคลาสจะมีเพียงแค่ parent เดียวเท่านั้น ตัวอย่างเช่น คลาสลูกค้าและพนักงานจะมี parent ที่เหมือนกันคือ คลาสบุคคล เป็นต้น
- *Inheritance*—เป็นคุณลักษณะหนึ่งของ object ภายในลำดับชั้นคลาสที่ซึ่งจะทำการเรียกใช้แอทริบิวและ method จากคลาสที่อยู่เหนือขึ้นไปจากคลาสที่เราทำการพิจารณา ตัวอย่างเช่น คลาสลูกค้าและพนักงานอาจถูกสร้างเป็นคลาสย่อยของคลาสบุคคล ด้วยเหตุนี้ คลาสลูกค้าและพนักงานจะสามารถเรียกใช้แอทริบิวและ method ของคลาสบุคคลที่เป็น parent (อยู่ตำแหน่งเหนือขึ้นไปในลำดับชั้นของคลาส) ได้

ในการออกแบบแบบจำลองเชิงวัตถุ เราจะสามารถประยุกต์ใช้แผนผัง UML (Unified Modeling Language) ในการแสดงข้อมูลและความสัมพันธ์ระหว่างข้อมูลต่างๆ UML จะเป็นภาษาที่อยู่ภายใต้แนวความคิดเชิงวัตถุที่ใช้อธิบายถึงแผนผังและสัญลักษณ์ต่างๆที่ใช้ในการออกแบบแบบจำลองข้อมูล ดังแสดงในรูปที่ 2.7 ที่ซึ่งจะเป็นการแสดงให้เห็นถึงการจัดเก็บข้อมูลและความสัมพันธ์ต่างๆของข้อมูลใบแจ้งราคาสินค้าแก่ลูกค้าคนหนึ่งๆโดยใช้ UML



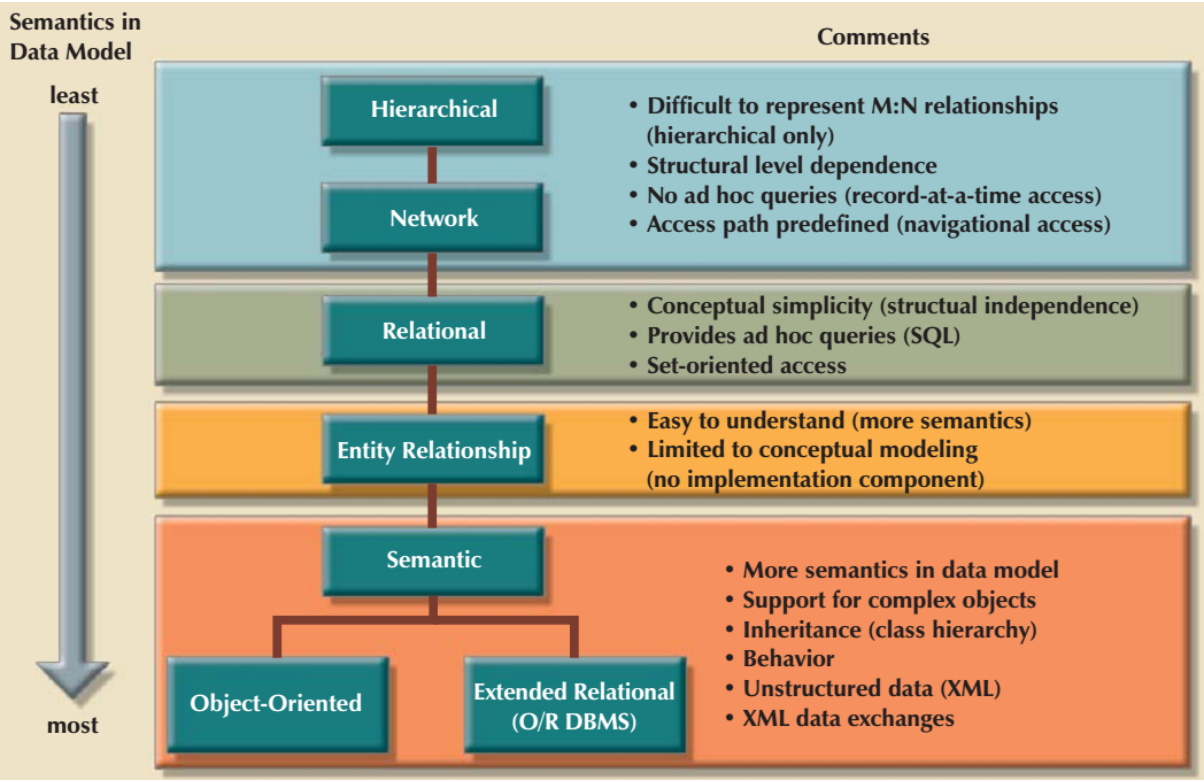
รูปที่ 2.7 การแสดงข้อมูลและความสัมพันธ์ด้วย UML

2.4.5 แบบจำลองเชิงสัมพันธ์ที่ถูกพัฒนาเพิ่มเติม

เนื่องจากข้อมูลต่างๆมีความซับซ้อนมากขึ้นตามยุคสมัย ด้วยเหตุนี้จึงทำให้ผู้ผลิตซอร์ฟแวร์ระบบจัดการฐานข้อมูลมีแนวคิดที่จะพัฒนาขีดความสามารถของแบบจำลองข้อมูลเชิงสัมพันธ์เพิ่มเติม (extended relational data model, ERDM) ที่ซึ่งจะทำการเพิ่มคุณลักษณะบางประการของแบบจำลองเชิงวัตถุเพื่อทำให้ฐานข้อมูลเชิงสัมพันธ์สนับสนุนการทำงานเชิงวัตถุ เช่น object การขยายชนิดของข้อมูลภายใต้แนวความคิดของคลาสและ inheritance เป็นต้น จากการพัฒนาข้างต้นจึงเป็นเหตุให้มีผู้คนให้ความสนใจกับระบบจัดการฐานข้อมูลเชิงวัตถุ/สัมพันธ์ (object/relation database management system, O/R DBMS) มากขึ้น นอกจากนั้น O/R DBMS ยังจะสามารถสนับสนุนการทำงานร่วมกับเอกสาร XML (eXtensible Markup Language—เป็นมาตราฐานในการแลกเปลี่ยนข้อมูลแบบไม่มีโครงสร้าง กึ่งโครงสร้าง และแบบมีโครงสร้าง ที่ซึ่งจะทำให้เราสามารถเชื่อมต่อระบบที่ใช้กับระบบต่างๆที่มีความแตกต่างทางด้านสถาปัตยกรรมของระบบได้

2.4.6 สรุปเกี่ยวกับแบบจำลองข้อมูล

วิวัฒนาการของระบบจัดการฐานข้อมูลยังคงมีอย่างต่อเนื่อง เนื่องจากยังคงมีความต้องการที่จะมีวิธีการสร้างแบบจำลองใหม่ๆที่ซึ่งสามารถเพิ่มการรองรับข้อมูลที่มีความซับซ้อน โดยบทสรุปของแบบจำลองข้อมูลที่ถูกคิดค้นและถูกใช้งานกันอย่างกว้างขวางจะสามารถแสดงได้ในรูปที่ 2.8



รูปที่ 2.8 วิวัฒนาการของแบบจำลองข้อมูล

แบบจำลองทั้งหมดในรูปที่ 2.8 จะมีคุณลักษณะคล้ายกันบางประการ ที่ซึ่งจะเป็นลักษณะที่แบบจำลองพึงมี โดยสามารถอธิบายได้ดังนี้

- แบบจำลองข้อมูลจะต้องสามารถแสดงถึงระดับของความง่ายของกรอบความคิดโดยไม่สูญเสียความครบถ้วนของความหมายของฐานข้อมูล มันอาจจะไม่ถูกต้องนักถ้าเรามีแบบจำลองข้อมูลที่ให้กรอบความคิดที่ค่อนข้างยากกว่าความเป็นจริง
- แบบจำลองจะต้องสามารถให้กรอบความคิดที่ใกล้เคียงความจริงมากที่สุดเท่าที่จะเป็นไปได้ เป้าหมายนี้จะสามารถทำได้โดยการเพิ่มความหมาย (semantic) ให้กับการแสดงถึงข้อมูลให้มากขึ้น (ความหมายจะเกี่ยวข้องกับพฤติกรรมของข้อมูล)
- การทำการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลจะต้องเป็นไปตามกฎความสอดคล้องและความสมบูรณ์ของข้อมูล

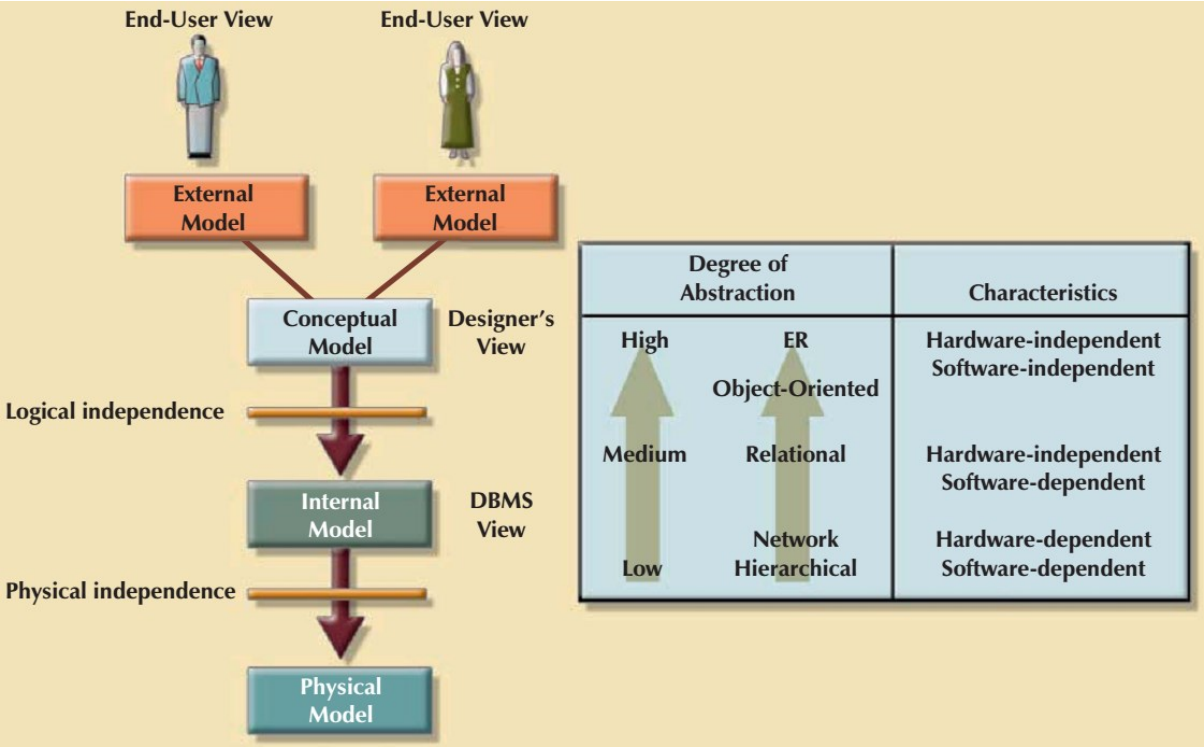
การคิดหรือออกแบบแบบจำลองข้อมูล ณ ปัจจุบันจะมุ่งเน้นที่การแก้ไขจุดด้วยของแบบจำลองก่อนหน้า โดยแบบจำลองข้อมูลแบบเครือข่ายถูกคิดค้นเพื่อแทนที่แบบจำลองข้อมูลแบบลำดับชั้น โดยแบบจำลองข้อมูลแบบเครือข่ายจะสามารถทำการปรับเปลี่ยนความสัมพันธ์ที่ซับซ้อนให้มีความง่ายมากขึ้น ในทางกลับกันแบบจำลองข้อมูลเชิงสัมพันธ์จะมีประโยชน์ที่มากกว่าแบบจำลองแบบเครือข่ายในหลายๆแง่มุม เช่น การแสดงข้อมูลด้วยวิธีการที่ง่าย การลดทอนความไม่สอดคล้องของข้อมูล และการประยุกต์ใช้ภาษาคิวรีในการเข้าถึงและจัดการกับข้อมูล แบบจำลองเชิงวัตถุได้ถูกพัฒนาขึ้นเพื่อจัดการกับข้อมูลที่มีความซับซ้อนภายใต้แนวความคิดของการจัดการเกี่ยวกับความหมายของข้อมูล ขณะที่แบบจำลองความสัมพันธ์เชิงวัตถุจะเป็นแบบจำลองข้อมูลเชิงสัมพันธ์ที่มีการเพิ่มเติมแนวความคิดเชิงวัตถุ ที่ซึ่งจะช่วยให้สามารถจัดการกับข้อมูลที่มีความซับซ้อนได้และยังสามารถสนับสนุนการทำงานร่วมกับ XML ได้อีกด้วย จากที่กล่าวข้างต้นเราสามารถสรุปเกี่ยวกับข้อดี-เสียของแบบจำลองต่างๆได้ดังรูปที่ 2.9

DATA MODEL	DATA INDEPENDENCE	STRUCTURAL INDEPENDENCE	ADVANTAGES	DISADVANTAGES
Hierarchical	Yes	No	1. It promotes data sharing. 2. Parent/Child relationship promotes conceptual simplicity. 3. Database security is provided and enforced by DBMS. 4. Parent/Child relationship promotes data integrity. 5. It is efficient with 1:M relationships.	1. Complex implementation requires knowledge of physical data storage characteristics. 2. Navigational system yields complex application development, management, and use; requires knowledge of hierarchical path. 3. Changes in structure require changes in all application programs. 4. There are implementation limitations (no multiparent or M:N relationships). 5. There is no data definition or data manipulation language in the DBMS. 6. There is a lack of standards.
Network	Yes	No	1. Conceptual simplicity is at least equal to that of the hierarchical model. 2. It handles more relationship types, such as M:N and multiparent. 3. Data access is more flexible than in hierarchical and file system models. 4. Data Owner/Member relationship promotes data integrity. 5. There is conformance to standards. 6. It includes data definition language (DDL) and data manipulation language (DML) in DBMS.	1. System complexity limits efficiency—still a navigational system. 2. Navigational system yields complex implementation, application development, and management. 3. Structural changes require changes in all application programs.
Relational	Yes	Yes	1. Structural independence is promoted by the use of independent tables. Changes in a table's structure do not affect data access or application programs. 2. Tabular view substantially improves conceptual simplicity, thereby promoting easier database design, implementation, management, and use. 3. Ad hoc query capability is based on SQL. 4. Powerful RDBMS isolates the end user from physical-level details and improves implementation and management simplicity.	1. The RDBMS requires substantial hardware and system software overhead. 2. Conceptual simplicity gives relatively untrained people the tools to use a good system poorly, and if unchecked, it may produce the same data anomalies found in file systems. 3. It may promote "islands of information" problems as individuals and departments can easily develop their own applications.
Entity relationship	Yes	Yes	1. Visual modeling yields exceptional conceptual simplicity. 2. Visual representation makes it an effective communication tool. 3. It is integrated with dominant relational model.	1. There is limited constraint representation. 2. There is limited relationship representation. 3. There is no data manipulation language. 4. Loss of information content occurs when attributes are removed from entities to avoid crowded displays. (This limitation has been addressed in subsequent graphical versions.)
Object-oriented	Yes	Yes	1. Semantic content is added. 2. Visual representation includes semantic content. 3. Inheritance promotes data integrity.	1. Slow development of standards caused vendors to supply their own enhancements, thus eliminating a widely accepted standard. 2. It is a complex navigational system. 3. There is a steep learning curve. 4. High system overhead slows transactions.

รูปที่ 2.9 ข้อดี-เสียของแบบจำลองข้อมูลต่างๆ

2.5 ระดับความชัดเจนของข้อมูล

ในการออกแบบฐานข้อมูลมักจะเป็นกระบวนการแบบค่อยเป็นค่อยไป โดยอาจเริ่มจากการพิจารณาข้อมูลและการดำเนินการต่างๆที่ค่อนข้างจะเป็นนามธรรม (ข้อมูลและการดำเนินการที่มีความชัดเจนค่อนข้างน้อย) จากนั้นจะทำการเพิ่มรายละเอียดไปเรื่อยๆจนกว่าฐานข้อมูลจะมีความสมบูรณ์ ที่ซึ่งสามารถตอบสนองต่อการดำเนินธุรกิจได้ เมื่อพิจารณาจากลักษณะของข้อมูลที่มีความชัดเจนแตกต่างกันในแต่ละรอบของการพิจารณาข้อมูลจะทำให้เราสามารถแบ่งระดับความชัดเจนของข้อมูล (level of data abstraction) ได้เป็น 3 ระดับ คือ external, conceptual และ internal ตามลำดับ ดังแสดงในรูปที่ 2.10 จะแสดงถึงการประยุกต์ใช้หรือการขึ้นตอนการประมวลผลตามระดับความชัดเจนข้างต้น นอกจากนั้นยังมีการเพิ่มเติมส่วนที่เป็นการออกแบบทางกายภาพ (physical) ที่ซึ่งจะเน้นที่การดำเนินการต่างๆ

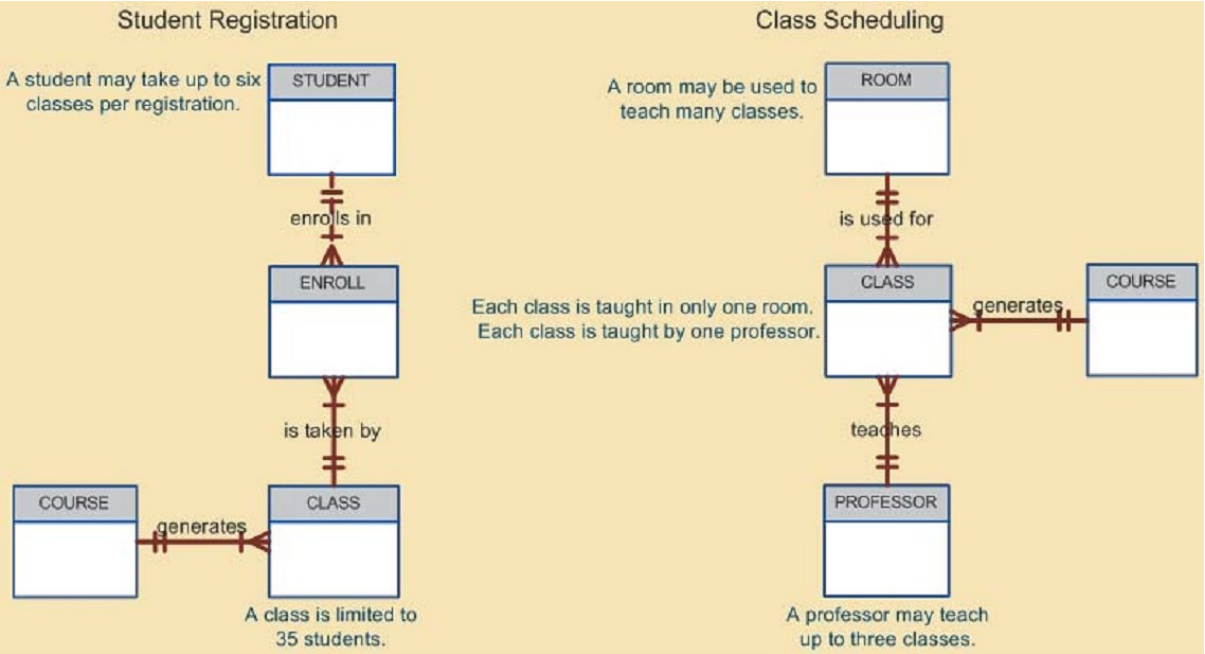


รูปที่ 2.10 ระดับความชัดเจนของข้อมูล

- *external*—จะเป็นข้อมูลที่เกิดจากมุมมองของผู้ใช้ที่มีหน้าที่ในการจัดการข้อมูลและทำการสร้างข้อมูลสารสนเทศ โดยทั่วไปแล้วบริษัทหรือองค์กรหนึ่งๆจะมีการแบ่งการดำเนินธุรกิจออกเป็นส่วนย่อยๆ เช่น การขาย การเงิน และการตลาด เป็นต้น จากการแบ่งส่วนย่อยดังกล่าวจะทำให้ผู้ใช้มักจะทำงานหรือใช้งานระบบฐานข้อมูลเพื่อการดำเนินธุรกิจในส่วนหนึ่งๆและจะมีมุมมองกับข้อมูลในส่วนที่ตัวเองได้รับมอบหมาย จากมุมมองของผู้ใช้ เราสามารถทำการสร้างแบบจำลองข้อมูลด้วยการ

ประยุกต์ใช้แผนผังความสัมพันธ์เอ็นทิตี้ที่ซึ่งมักจะเรียกว่า external schema ที่ซึ่งจะเป็นแผนผังในมุมมองของผู้ใช้

ในการที่จะมีความเข้าใจเกี่ยวกับ external schema มากขึ้น ลองพิจารณาตัวอย่างข้อมูลเกี่ยวกับการลงทะเบียนเรียนและการจัดการจัดเรียนในสถานศึกษาที่ซึ่งจะถูกแบ่งเป็น 2 การดำเนินการย่อย ดังแสดงในรูปที่ 2.11 จะประกอบไปด้วย 2 external schema (1 schema สำหรับ 1 การดำเนินการย่อย) ที่จะประกอบไปด้วยเอ็นทิตี้ ความสัมพันธ์ กระบวนการ และเงื่อนไขต่างๆของการดำเนินการ

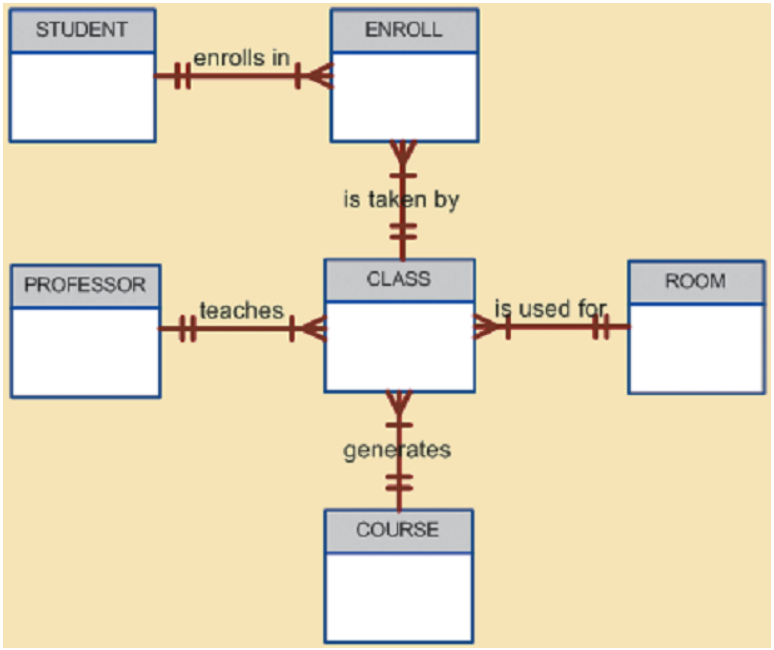


รูปที่ 2.11 ตัวอย่าง external schema

การประยุกต์ใช้มุมมองแบบ external ในการแสดงถึงส่วนย่อยๆของฐานข้อมูลจะมีประโยชน์หลายประการดังนี้

- ช่วยให้สามารถระบุหรือจำกัดขอบเขตของข้อมูลที่สนับสนุนการดำเนินการหนึ่งๆของการดำเนินธุรกิจได้โดยง่าย
- ช่วยให้ผู้ที่ออกแบบแบบจำลองข้อมูลสามารถให้ความคิดเห็นเกี่ยวกับแบบจำลองที่สร้างขึ้นได้ง่ายขึ้น เมื่อทำการประยุกต์ใช้ external จะทำให้เราสามารถตรวจสอบเพื่อให้แน่ใจได้ว่าแบบจำลองที่ออกแบบสามารถสนับสนุนการทำงานทั้งหมดตามที่ต้องการ และมีความสอดคล้องกับเงื่อนไขต่างๆในการดำเนินธุรกิจ
- ช่วยให้เราสามารถเพิ่มเงื่อนไขหรือมาตราการเกี่ยวกับความปลอดภัยในการจัดการกับข้อมูลได้
- ช่วยให้สามารถพัฒนาโปรแกรมได้ง่ายขึ้น

- *conceptual*—จะเป็นมุมมองหรือแบบจำลองที่ถูกสร้างขึ้นหลังจากทำการสร้างมุมมองแบบ external จากนั้น มุมมองแบบ conceptual จะทำการรวบรวมการดำเนินการต่างๆ หรือ external schema ย่อยๆ (คือการรวมเอ็นทิตี้ ความสัมพันธ์ เงื่อนไข และกระบวนการต่างๆ) ให้เป็นหนึ่งเดียว ที่ซึ่งจะทำให้เรามองเห็นภาพรวมทั้งหมดของทุกการดำเนินการในองค์กร แบบจำลองแบบ conceptual จะทำการประยุกต์ใช้แผนผังความสัมพันธ์เอ็นทิตี้ (entity-relationship diagram) ในการแสดงถึงข้อมูลและความสัมพันธ์ต่างๆที่ซึ่งจะถูกเรียกว่า conceptual schema (ตัวอย่าง conceptual schema จะแสดงในรูปที่ 2.12 ที่ซึ่งจะเป็นการรวมกันของ external schema ย่อย 2 schema ในรูปที่ 2.10)



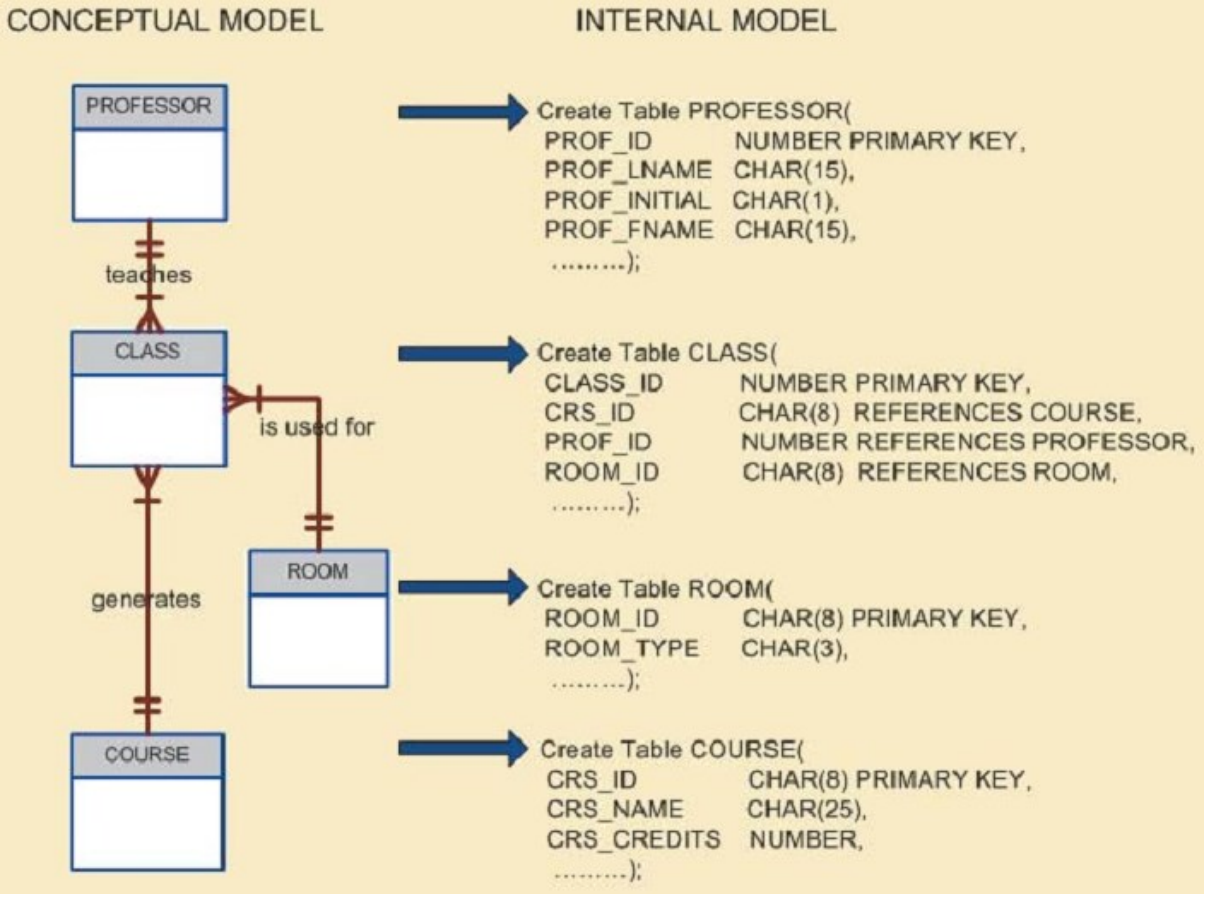
รูปที่ 2.12 ตัวอย่าง conceptual schema

การประยุกต์ใช้ conceptual model จะมีประโยชน์หลายประการด้วยกัน เช่น 1) สามารถจัดเตรียมมุมมองข้อมูลที่สามารถเข้าใจได้ง่าย 2) ไม่ขึ้นกับทั้งซอร์ฟแวร์ (ไม่ขึ้นกับระบบจัดการฐานข้อมูลใดๆ จะสามารถเรียกใช้ระบบใดก็ได้) และฮาร์ดแวร์ในการพัฒนาระบบฐานข้อมูล ด้วยเหตุนี้จะทำให้การเปลี่ยนระบบจัดการฐานข้อมูลหรือปรับเปลี่ยนฮาร์ดแวร์บางส่วนจะไม่ส่งผลกระทบต่อการออกแบบฐานข้อมูลภายใต้ conceptual model

- *internal*—หลังจากทำการสร้าง conceptual model เราจะต้องทำการเลือกระบบจัดการฐานข้อมูลที่จะใช้สำหรับจัดเก็บและจัดการต่างๆกับข้อมูล และหลังจากการเลือกระบบจัดการฐานข้อมูลแล้ว จะเป็นการทำงานของ internal model ที่ซึ่งจะเป็นการแปลงแนวความคิดจาก conceptual model ที่สร้างขึ้นไปเป็นโครงสร้างที่จะใช้ในการจัดเก็บข้อมูลในระบบจัดการฐานข้อมูล หรือ เราอาจกล่าวอีกนัยหนึ่งได้ว่า internal model จะต้องการให้ผู้ออกแบบฐานข้อมูล

ทำการเชื่อมโยงส่วนประกอบต่างๆทั้งเอ็นทิตี้ ความสัมพันธ์ เงื่อนไข และขั้นตอนต่างๆใน conceptual model ไปยังระบบจัดการฐานข้อมูลที่ทำการเลือกไว้แล้ว ภายใต้แนวคิดของ internal model จะทำการประยุกต์ใช้ internal schema เพื่อแสดงรายละเอียดทั้งหมดของโครงสร้างข้อมูล รูปที่ 2.13 จะแสดงตัวอย่างของ internal schema ที่ถูกแปลงจาก conceptual model ด้วยการประยุกต์ใช้ SQL ที่ซึ่งมาภาษามาตราฐานสำหรับฐานข้อมูลเชิงสัมพันธ์

Internal model จะมีการขึ้นกับซอร์ฟแวร์ เนื่องจากเราต้องทำการเลือกซอร์ฟแวร์ระบบจัดการฐานข้อมูลเป็นอันดับแรก ดังนั้นเมื่อเราทำการปรับเปลี่ยนซอร์ฟแวร์ระบบจัดการฐานข้อมูล เราจะต้องทำการปรับเปลี่ยน internal model ให้มีความสอดคล้องกับคุณลักษณะและความต้องการของซอร์ฟแวร์ที่เราปรับเปลี่ยนด้วยเช่นกัน (แต่การปรับเปลี่ยน internal model จะไม่ส่งผลต่อ conceptual model) แต่ในทางกลับกัน internal model ยังคงคุณสมบัติการไม่ขึ้นกับฮาร์ดแวร์ ที่ซึ่งการปรับเปลี่ยนอุปกรณ์ฮาร์ดแวร์ใดๆจะไม่ส่งผลต่อ internal model เลย



รูปที่ 2.13 การปรับเปลี่ยนจาก conceptual model ไปเป็น internal model

- *physical*—จะเป็นแบบจำลองที่ทำงานในระดับต่ำสุดที่ซึ่งจะเป็นแบบจำลองที่อธิบายถึงวิธีการในการจัดเก็บข้อมูลลองในอุปกรณ์จัดเก็บข้อมูล (เช่น ดิสก์ หรือ เทป) ในการออกแบบ physical model จะต้องการทราบถึงรายละเอียดหรือข้อจำกัดของอุปกรณ์จัดเก็บข้อมูล และวิธีในการเข้าถึง

ข้อมูลที่ผู้ออกแบบฐานข้อมูลต้องการ ที่ซึ่งจะทำให้ physical model จะต้องขึ้นอยู่กับทั้งซอร์ฟแวร์และฮาร์ดแวร์ โดยโครงสร้างของการจัดเก็บข้อมูลจะต้องขึ้นกับซอร์ฟแวร์และจะต้องขึ้นกับชนิดของอุปกรณ์สำหรับจัดเก็บข้อมูล นอกจากความต้องการข้างต้นแล้ว ขั้นตอนการออกแบบ physical model ยังต้องการให้ผู้ออกแบบโมเดลที่ซึ่งมีความรู้เกี่ยวกับซอร์ฟแวร์และฮาร์ดแวร์ที่ใช้ในการสร้างระบบฐานข้อมูลอีกด้วย

แต่อย่างไรก็ตาม ในแอพพลิเคชันทั่วๆไป ผู้ออกแบบฐานข้อมูลมักจะให้ความสำคัญกับการออกแบบจำลองที่บ่งบอกถึงข้อมูลที่จะจัดเก็บในฐานข้องมูล (logical model—อาทิเช่น แบบจำลองความสัมพันธ์ แบบจำลองความสัมพันธ์เอ็นทิตี้ เป็นต้น) มากกว่าการพิจารณาถึงรายละเอียดของ physical model ที่ซึ่งจะสามารถช่วยให้เราสามารถปรับจูนเพิ่มประสิทธิภาพการทำงานได้

จากเนื้อหาที่กล่าวเกี่ยวกับการทำงานของแบบจำลองต่างๆทั้ง external, conceptual, internal และ physical models เราจะสามารถทราบถึงคุณลักษณะ รายละเอียดและขั้นตอนการทำงานต่างๆ ดังนั้นเราสามารถสรุปคุณลักษณะเกี่ยวกับระดับของความชัดเจนของข้อมูลในแบบจำลองได้ดังรูปที่ 2.14

MODEL	DEGREE OF ABSTRACTION	FOCUS	INDEPENDENT OF
External	High	End-user views	Hardware and software
Conceptual	↕	Global view of data (database model–independent)	Hardware and software
Internal		Specific database model	Hardware
Physical	Low	Storage and access methods	Neither hardware nor software

รูปที่ 2.14 ระดับของความชัดเจนของข้อมูล

คำถามท้ายบท

1. จงอธิบายถึงความสำคัญของแบบจำลองข้อมูล
2. กฎเกณฑ์ทางธุรกิจคืออะไร และ อะไรคือวัตุประสงค์หลักของการประยุกต์ใช้กฎเกณฑ์ทางธุรกิจ
3. เราจะสามารถทำการปรับเปลี่ยนกฎเกณฑ์ทางธุรกิจไปเป็นส่วนประกอบของแบบจำลองข้อมูลได้อย่างไร
4. จงอธิบายคุณลักษณะพื้นฐานของแบบจำลองข้อมูลเชิงสัมพันธ์และความสำคัญของคุณลักษณะเหล่านี้ต่อผู้ใช้งานระบบฐานข้อมูลและผู้ออกแบบฐานข้อมูล
5. จงอธิบายวิธีการที่แบบจำลองข้อมูลความสัมพันธ์เอ็นทิตี้จะสามารถช่วยในการออกแบบโครงสร้างฐานข้อมูลเชิงสัมพันธ์ได้
6. จงสร้างแผนผังความสัมพันธ์เอ็นทิตี้จากกฎเกณฑ์ทางธุรกิจดังต่อไปนี้ “ลูกค้าคนหนึ่งๆสามารถทำการชำระเงินได้หลายครั้ง แต่สำหรับการชำระเงินครั้งหนึ่งๆ จะสามารถทำได้โดยลูกค้าคนหนึ่งๆเท่านั้น”
7. เมื่อทำการพิจารณาเกี่ยวกับการขึ้นกับโครงสร้างและการขึ้นกับข้อมูล—จงเปรียบเทียบระบบแฟ้มข้อมูลกับแบบจำลองข้อมูลทั้ง 5 ที่อธิบายในบทนี้ภายใต้ 2 แง่มุมข้างต้น

8. จงยกตัวอย่างของรูปแบบความสัมพันธ์ทั้ง 3 ชนิด
9. ตารางคืออะไร และมีบทบาทสำคัญอย่างไรในแบบจำลองเชิงสัมพันธ์
10. แผนผังความสัมพันธ์เอ็นทิตี้คืออะไร จงยกตัวอย่างประกอบ