บทที่ 7

แบบจำลองข้อมูลโดยใช้แบบจำลองอีอาร์

แบบจำลองเชิงแนวคิดมีความสำคัญต่อการออกแบบโปรแกรมประยุกต์ที่เชื่อมต่อกับฐานข้อมูล โดยปกติแล้ว โปรแกรมประยุกต์เป็นส่วนต่อประสานกับผู้ใช้เพื่อให้ผู้ใช้สามารถเชื่อมต่อกับฐานข้อมูล เพื่อสืบค้นข้อมูล หรือปรับปรุงข้อมูลในฐานข้อมูลได้สะดวก ตัวอย่างเช่น ระบบฐานข้อมูลของธนาคาร มีโปรแกรมประยุกต์ที่อยู่ในรูปแบบของส่วนต่อประสานกราฟิก เพื่อให้เจ้าหน้าที่ธนาคารใช้เชื่อมต่อกับฐานข้อมูลของธนาคารที่ใช้จัดเก็บข้อมูลบัญชีของลูกค้าที่ทำธุรกรรมต่างๆกับธนาคาร เช่น การฝาก-ถอนเงิน เป็นต้น จากตัวอย่างข้างต้น ส่วนที่สำคัญคือ การออกแบบฐานข้อมูล การออกแบบโปรแกรม และการทดสอบโปรแกรม ซึ่งโดยทั่วไปแล้ว การออกแบบฐานข้อมูลจะเป็นหน้าที่ของคนที่ออกแบบฐานข้อมูล (Database Designer) สำหรับส่วนของการออกแบบโปรแกรม และการทดสอบโปรแกรมจะเป็นหน้าที่ของวิศวกรซอฟต์แวร์ (Software Engineer) ในบทนี้ จะเน้นเรื่องของการออกแบบฐานข้อมูลเชิงแนวคิดโดยใช้แบบจำลองข้อมูลเชิงสัมพันธ์ (Entity-Relationship Model) หรือเรียกสั้นๆว่า แบบจำลองข้อมูลอีอาร์ (ER Model) ซึ่งเป็นแบบจำลองที่นิยมใช้ในแบบจำลองข้อมูลเชิงแนวคิดระดับสูง โดยที่นำเสนอแนวคิดผ่านทางอีอาร์ไดอะแกรม (ER Diagram)

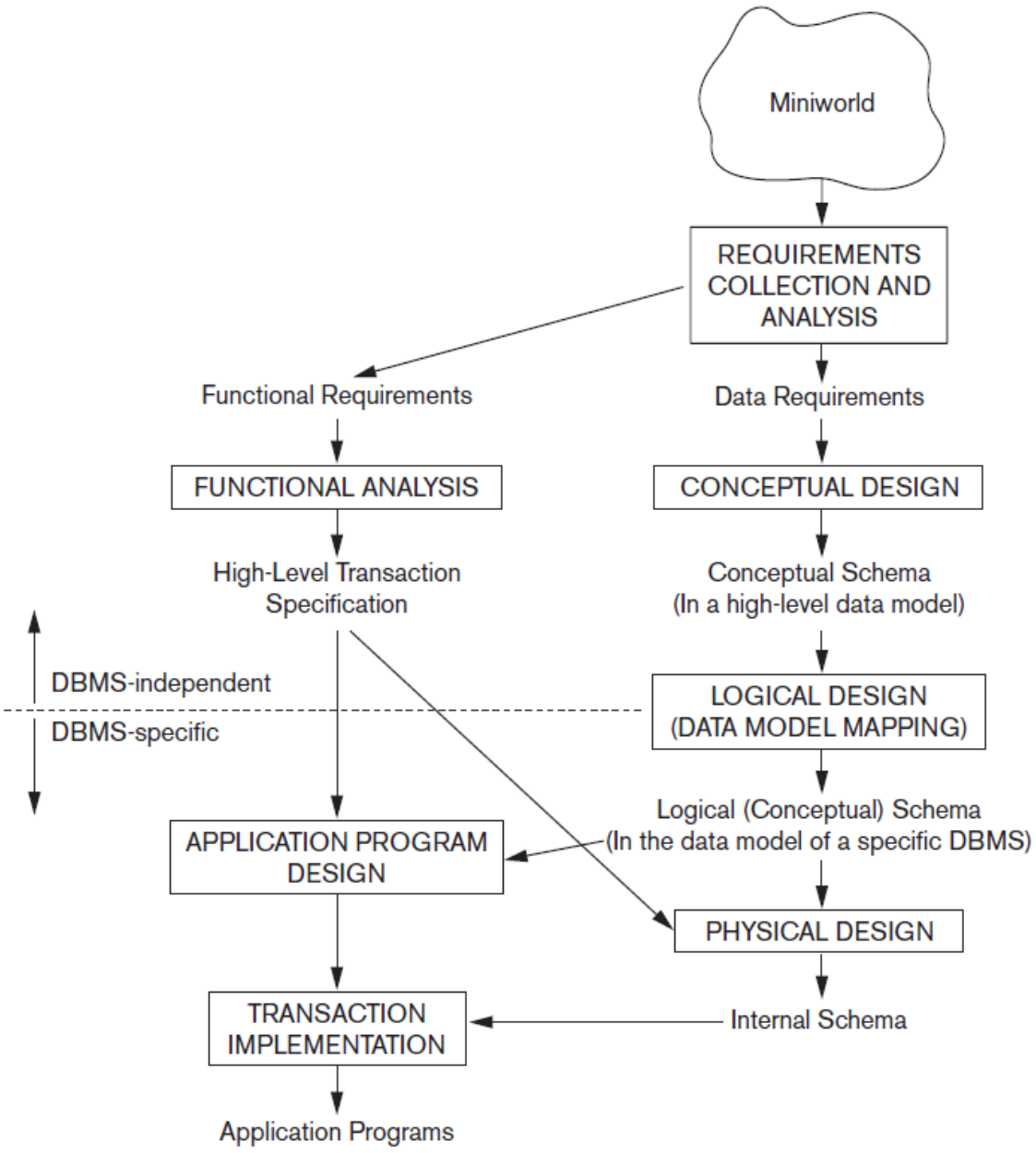
7.1 ขั้นตอนการออกแบบฐานข้อมูล

ในการออกแบบฐานข้อมูล มีลำดับขั้นตอนดังแสดงในรูปที่ 7.1 และรายละเอียดแต่ละขั้นตอน มีดังต่อไปนี้

1) ผู้ออกแบบฐานข้อมูล สัมภาษณ์คนที่ใช้ฐานข้อมูล เพื่อจะได้เข้าใจความต้องการของผู้ใช้ และจะได้ดำเนินการจัดทำเอกสารที่เกี่ยวข้องกับฐานข้อมูลให้ถูกต้อง ผลลัพธ์ที่ได้จากการสัมภาษณ์คือข้อมูลที่ผู้ใช้ต้องการ
2) กำหนดฟังก์ชันต่างๆของแอปพลิเคชัน ตามความต้องการของผู้ใช้ ในการออกแบบฟังก์ชันต่างๆ สามารถนำเสนอในรูปแบบของไดอะแกรมที่แสดงถึงการไหลของข้อมูล (Data Flow Diagram) หรือยูสเคสไดอะแกรม (Use Case Diagram) เป็นต้น
3) ออกแบบฐานข้อมูลในระดับแนวคิด ซึ่งเป็นแบบจำลองข้อมูลระดับสูง เพื่ออธิบายข้อมูลตามความต้องการของผู้ใช้ฐานข้อมูล รายละเอียดในการออกแบบจะกล่าวถึงในหัวข้อ 7.2

4) เปลี่ยนจากแบบจำลองข้อมูลระดับสูงไปเป็นแบบจำลองข้อมูลระดับแนวคิด โดยการออกแบบเชิงตรรกะ หรือเรียกว่า Data Modeling Mapping ผลลัพธ์ที่ได้จากการออกแบบเชิงตรรกะคือโครงสร้างฐานข้อมูล (Database Schema)

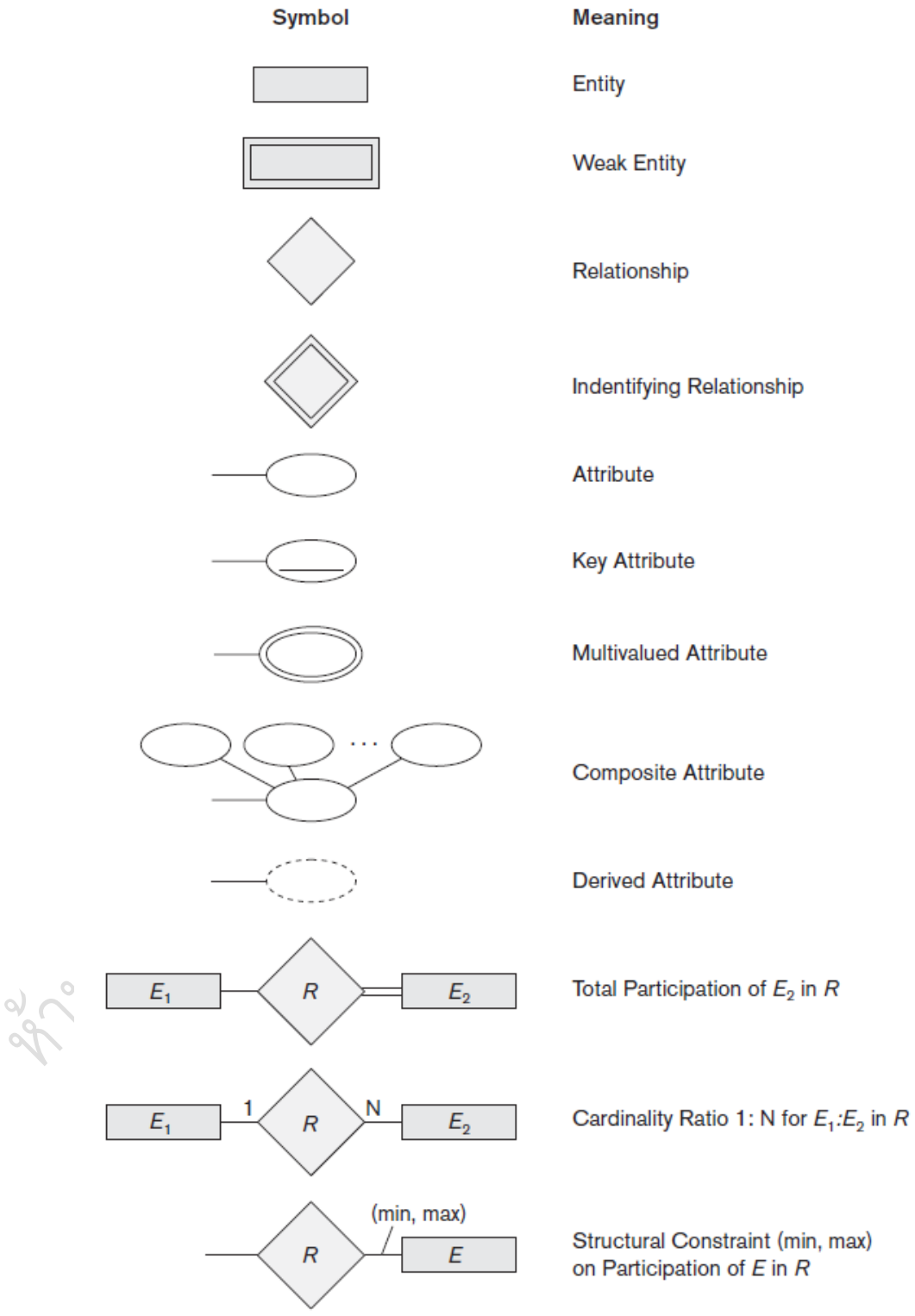
5) การออกแบบเชิงกายภาพ เป็นการกำหนดโครงสร้างเชิงกายภาพ คือออกแบบที่ที่ใช้สำหรับจัดเก็บข้อมูล องค์ประกอบของไฟล์ข้อมูล ดัชนี วิธีการเข้าถึงข้อมูล และพารามิเตอร์ (Parameter) ต่างๆที่เกี่ยวข้องกับการออกแบบเชิงกายภาพ นอกจากนี้การออกแบบโปรแกรมประยุกต์จะทำไปพร้อมๆกันโดยวิศวกรซอฟต์แวร์ และหลังจากนั้นจะทำการติดตั้งระบบ



รูปที่ 7.1 ขั้นตอนการออกแบบฐานข้อมูล [1]

7.2 การออกแบบฐานข้อมูลในระดับแนวคิด

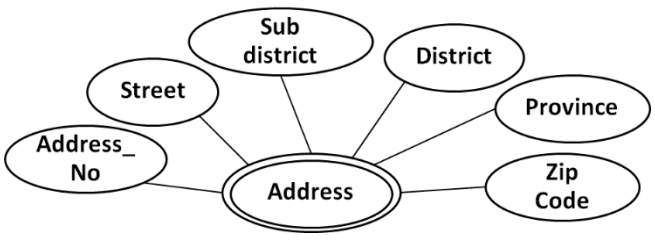
ในการออกแบบฐานข้อมูล สามารถนำเสนอแนวคิดในการออกแบบผ่านทางอีอาร์ไดอะแกรม ซึ่งเป็นแบบจำลองข้อมูลระดับแนวคิด (Conceptual Data Model) สัญลักษณ์ และความหมายขององค์ประกอบต่างๆที่อยู่ในอีอาร์ไดอะแกรม แสดงในรูปที่ 7.2



รูปที่ 7.2 สัญลักษณ์ที่ใช้ในอีอาร์ไดอะแกรม [1]

จากรูป 7.2 ความหมายของสัญลักษณ์ต่างๆมีรายละเอียดดังต่อไปนี้

1) Regular Entity หรือ Strong Entity คือ สิ่งต่างๆที่เกิดขึ้นจริงในโลกที่ดำรงอยู่ได้ด้วยตัวเอง หรือเป็นอิสระจากสิ่งอื่น ในการออกแบบฐานข้อมูล สิ่งที่ต้องคิดถึงสิ่งแรกคือ เอนทิตี ซึ่งใช้นำเสนอ สิ่งที่เกิดขึ้น สิ่งที่เราสนใจ หรือสิ่งที่สำคัญที่ต้องการจัดเก็บลงในฐานข้อมูล ไม่จำเป็นต้องจัดเก็บทุกสิ่งที่เกิดขึ้น เช่น ถ้าเรากำลังนึกถึงสิ่งที่เกิดขึ้นในระบบการขายสินค้าออนไลน์ เอนทิตี ที่ต้องนึกถึงเป็นอย่างแรกคือ ลูกค้า และสินค้า เป็นต้น ถ้าในระบบไม่ได้สนใจเกี่ยวกับโปรโมชัน (Promotion) ก็ไม่จำเป็นต้องจัดเก็บข้อมูลโปรโมชันเข้าไปในฐานข้อมูล
2) Attribute คือ คุณสมบัติเฉพาะที่สามารถใช้อธิบายเอนทิตีได้ เช่น คุณสมบัติเฉพาะที่บ่งบอกถึงนักศึกษา คือ รหัสนักศึกษา ชื่อนักศึกษา เป็นต้น ทั้งนี้ให้นึกถึงเฉพาะคุณสมบัติที่เราสนใจ หรือสำคัญต่อระบบของเราเท่านั้น เช่น คุณสมบัติของนักศึกษาที่ไม่เกี่ยวกับระบบลงทะเบียน เช่น งานอดิเรก น้ำหนัก ส่วนสูง ซึ่งไม่เกี่ยวข้องกับระบบเหล่านี้ ก็ไม่จำเป็นต้องจัดเก็บลงในฐานข้อมูล แอททริบิวต์มีหลายประเภท ดังต่อไปนี้
    - Simple Attribute or Atomic Attribute คือ แอททริบิวต์ที่ไม่สามารถแบ่งแยกได้อีก เช่น รหัสนักศึกษา รหัสไปรษณีย์ เป็นต้น
    - Composite Attribute คือ แอททริบิวต์ที่สามารถแบ่งแยกออกเป็นส่วนที่ย่อยลงไปได้อีก เช่น ที่อยู่ สามารถแบ่งย่อยออกเป็น บ้านเลขที่ ถนน ตำบล อำเภอ จังหวัด รหัสไปรษณีย์ เป็นต้น
    - Multivalued Attribute คือ แอททริบิวต์ที่มีค่าข้อมูลได้หลายค่า เช่น ระบบลงทะเบียนต้องการจัดเก็บข้อมูลอาคารเรียนของแต่ละคณะ ซึ่งแต่ละคณะมีอาคารเรียนหลายหลัง เป็นต้น ซึ่งต่างจากแอททริบิวต์ที่เป็น Simple Attribute เช่น รหัสนักศึกษาของแต่ละคนจะมีแค่ 1 รหัสเท่านั้น
    - Derived Attribute คือ แอททริบิวต์ที่สามารถหาค่าข้อมูลได้จากแอททริบิวต์อื่น เช่น อายุ สามารถคำนวณจาก วันที่ปัจจุบัน ลบด้วย ค่าข้อมูลในแอททริบิวต์วันเกิด หรือ แอททริบิวต์ จีพีเอ (Grade Point Average: G.P.A.) ที่สามารถคำนวณได้จากรายวิชาที่นักศึกษาลงทะเบียน และเกรดที่ได้ เป็นต้น
    - Complex Attribute คือ แอททริบิวต์ที่เกิดจากการผสมผสานระหว่าง Composite Attribute และ Multivalued Attribute เช่น บางระบบต้องการจัดเก็บข้อมูลที่อยู่มากกว่า 1 ที่ และที่อยู่เป็น Composite Attribute ดังตัวอย่างแสดงด้านล่าง



- Key Attribute คือ แอททริบิวต์ในเอนทิตีที่มีค่าข้อมูลไม่ซ้ำกัน เช่น รหัสนักศึกษาของแต่ละคนจะมีค่าไม่ซ้ำกัน ในทุกๆเอนทิตีจะต้องมีแอททริบิวต์ที่เป็นคีย์แอททริบิวต์

3) Relationship คือ ความสัมพันธ์ที่เกิดขึ้นระหว่างเอนทิตี เช่น ถ้ามีเอนทิตี ลูกค้า กับ สินค้า ให้มองว่าสองเอนทิตีนี้มีความสัมพันธ์กัน หรือไม่ ถ้าสัมพันธ์ สัมพันธ์กันอย่างไร โดยตั้งชื่อความสัมพันธ์ให้สอดคล้องกับความต้องการในการจัดเก็บข้อมูลในฐานข้อมูล เช่น ให้เราตั้งคำถามก่อนว่าต้องการทราบหรือไม่ว่ามีลูกค้าคนใดซื้อสินค้าอะไรในระบบ ถ้าต้องการทราบ เราก็สร้างความสัมพันธ์ระหว่าง ลูกค้า กับ สินค้า ดังนี้คือ ลูกค้า ซื้อ สินค้า และให้มองว่ามีข้อมูลที่ต้องจัดเก็บเพิ่มเติมเกี่ยวกับการซื้อสินค้า เช่น ลูกค้าซื้อสินค้าวันไหน ซื้อกี่ชิ้น เป็นต้น หากเราต้องการจัดเก็บ เราก็ออกแบบข้อมูลเพิ่มเติมที่เกี่ยวกับการซื้อ และข้อมูลเหล่านี้จะถูกเชื่อมโยงกับความสัมพันธ์ โดยปกติแล้ว ความสัมพันธ์เกิดขึ้นเนื่องจากแอททริบิวต์ ในเอนทิตีหนึ่ง มีการอ้างอิงถึงข้อมูลในเอนทิตีอื่น เช่น ดังที่กล่าวมาแล้วคือ ลูกค้ารหัส xxx ซื้อสินค้ารหัส yyy เป็นต้น รายละเอียดของความสัมพันธ์ มีดังต่อไปนี้ คือ

- Relation Degree คือ จำนวนของเอนทิตี ที่เชื่อมต่อกับความสัมพันธ์ เช่น 2 (Binary Relationship) 3 (Ternary Relationship) หรือ n (n-ary Relationship) โดยที่ n คือตัวเลขจำนวนเอนทีตีที่สัมพันธ์กันมีค่าตั้งแต่ 4 เป็นต้นไป ตัวอย่างความสัมพันธ์ของ 2 เอนทิตี และความสัมพันธ์ของ 3 เอนทิตี ดังแสดงในรูปที่ 7.3
- การตั้งชื่อความสัมพันธ์ ควรตั้งตามบทบาทหน้าที่ของความสัมพันธ์ ชื่อความสัมพันธ์แต่ละชื่อ ใช้บอกบทบาท การมีส่วนร่วม หรือมีความเกี่ยวข้องกันระหว่างเอนทิตีแต่ละคู่หรือแต่ละกลุ่ม (กรณีที่เป็นความสัมพันธ์ตั้งแต่ 3 เอนทิตี ขึ้นไป)
- Recursive Relationship คือ การที่เอนทิตีเดียวกันมีความสัมพันธ์กันเอง ในลักษณะนี้เกิดจากการที่มีแอททริบิวต์ 2 แอททริบิวต์ที่มีลักษณะเหมือนกัน แต่มีบทบาทต่างกัน เช่น ในเอนทิตีพนักงาน มีหมายเลขประกันสังคม 2 หมายเลข หมายเลขหนึ่งมีบทบาทเป็นหมายเลขประกันสังคมของพนักงาน (SSN) แต่อีกหมายเลขหนึ่งมีบทบาทเป็นหมายเลขประกันสังคมของหัวหน้างาน (Super_SSN) เป็นต้น ดังตัวอย่างแสดงในรูปที่ 7.4
- Cardinality Ratio คือ อัตราส่วนความสัมพันธ์ระหว่างข้อมูลของสองเอนทิตี ซึ่งค่าที่ระบุในแต่ละฝั่งจะเป็นค่าอัตราส่วนที่มากที่สุดที่สัมพันธ์กับฝั่งตรงข้าม ซึ่งค่าที่เป็นไปได้มีดังนี้คือ
    - One-to-one or 1:1 คือความสัมพันธ์แบบ หนึ่งต่อหนึ่ง
    - One-to-many or 1:N คือความสัมพันธ์แบบ หนึ่งต่อหลาย
    - Many-to-many or M:N คือความสัมพันธ์แบบ หลายต่อหลาย

ในการกำหนดอัตราส่วนความสัมพันธ์ ให้ทำสองฝั่ง คือ จากซ้ายไปขวา และจากขวาไปซ้าย และคงไว้ซึ่งตัวเลขที่อยู่ที่ปลายลูกศรทั้งสองด้าน ดังตัวอย่างแสดงด้านล่าง ตัวอย่างเช่น อัตราส่วนความสัมพันธ์ระหว่าง พนักงาน (Employee) บริหาร (manage) แผนก (Department) มีการดำเนินการตามลำดับขั้นตอนดังต่อไปนี้

1) กำหนดอัตราส่วนความสัมพันธ์ จากซ้ายไปขวา คือ พนักงาน 1 คน บริหาร แผนกได้ 1 แผนก
2) กำหนดอัตราส่วนความสัมพันธ์ จากขวาไปซ้าย คือ แผนก 1 แผนก ถูกบริหารโดยพนักงาน 1 คน
3) คงตัวเลขที่อยู่ที่ปลายลูกศรทั้งสองด้าน ดังนั้น พนักงาน บริหาร แผนก มีอัตราส่วนความสัมพันธ์เป็นแบบ One-to-one ดังแสดงในรูปที่ 7.5

ตัวอย่างของอัตราส่วนความสัมพันธ์แบบ One-to-many แสดงดังรูปที่ 7.6 สำหรับอัตราส่วนความสัมพันธ์แบบ Many-to-many ไม่ควรระบุจำนวนตัวเลขเป็น N ทั้งสองฝั่ง เนื่องจากค่าจำนวนตัวเลขที่อยู่ในแต่ละฝั่งไม่จำเป็นต้องมีค่าเท่ากัน จึงควรระบุฝั่งหนึ่งเป็น M และอีกฝั่งหนึ่งเป็น N ดังตัวอย่างแสดงในรูปที่ 7.7

- Participation Constraint คือ ข้อกำหนดของการมีส่วนร่วม ซึ่งมี 2 ลักษณะ คือ
  1) Total Participation ใช้สัญลักษณ์เส้นความสัมพันธ์แบบเส้นคู่ คือ ลักษณะที่ทุกข้อมูลของเอนทิตีที่อยู่ฝั่งที่มีความสัมพันธ์แบบเส้นคู่มีส่วนร่วมกับข้อมูลที่อยู่ฝั่งตรงข้าม เช่น ความสัมพันธ์ระหว่าง พนักงาน บริหาร แผนก โดยที่ฝั่งแผนกเป็นความสัมพันธ์แบบมีส่วนร่วมทั้งหมด หมายความว่า ทุกๆแผนกจะต้องมีผู้จัดการ (พนักงานที่อยู่อีกฝั่งหนึ่ง) มาบริหาร ดังตัวอย่างรูปที่แสดงด้านล่าง

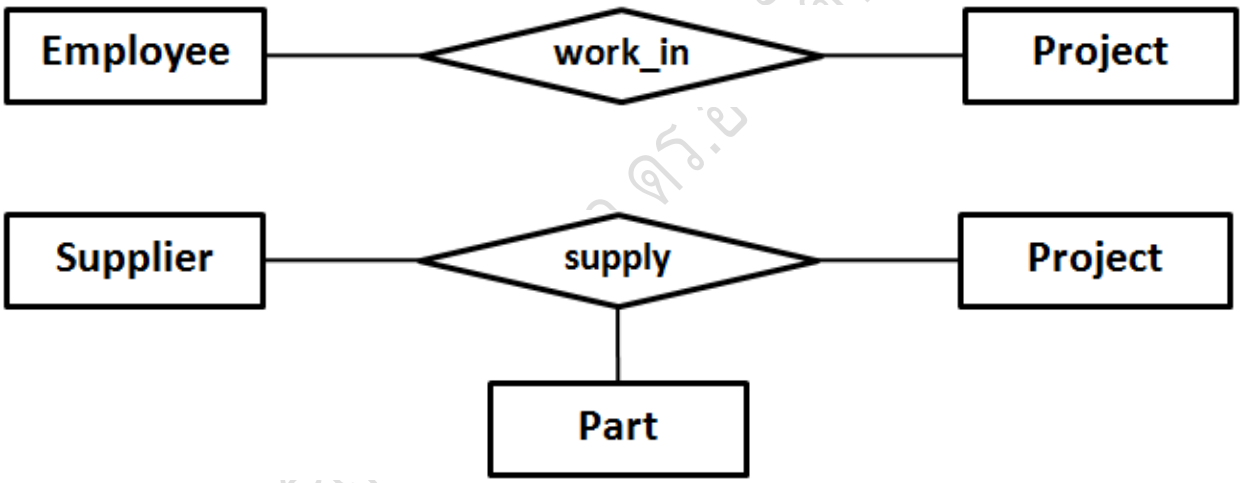
  2) Partial Participation ใช้สัญลักษณ์เส้นความสัมพันธ์แบบเส้นเดียว คือ ลักษณะที่ข้อมูลของเอนทิตีที่อยู่ฝั่งที่มีความสัมพันธ์แบบเส้นเดียวอาจจะมี หรือไม่มีส่วนร่วมกับข้อมูลที่อยู่ฝั่งตรงข้าม เช่น ความสัมพันธ์ระหว่าง พนักงาน บริหาร แผนก โดยที่ฝั่งพนักงานเป็นความสัมพันธ์แบบมีส่วนร่วมบางส่วน หมายความว่า มีพนักงานบางคนเท่านั้นที่ได้บริหารแผนก ดังตัวอย่างรูปที่แสดงด้านบน

4) Weak Entity คือ เอนทิตีที่ไม่สามารถอยู่ได้ด้วยตัวเอง ถ้าไม่มีเอนทิตีที่เป็นเจ้าของ (Owner) หรือเอนทิตีที่เป็นต้นกำเนิด (Parent) ก็จะไม่มี วีคเอนทิตี เช่น ถ้าไม่มีรายการสั่งซื้อสินค้า

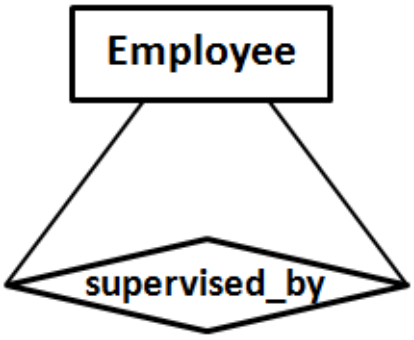
(Order) ก็จะไม่มีรายละเอียดการสั่งซื้อสินค้า (Order Item) เป็นต้น โดยปกติแล้ว วีคเอนทิตี จะไม่มี คีย์แอททริบิวต์ (Key Attribute) เป็นของตัวเอง แต่จะมี Partial Key (แอททริบิวต์ที่ขีดเส้นใต้ด้วยเส้นประ) คือ แอททริบิวต์ที่ใช้แบ่งแยกความแตกต่าง หรือบ่งบอกความเป็นเอกลักษณ์ของวีคเอนทิตี วีคเอนทิตีแตกต่างจากสตรองเอนทิตีตรงที่ ข้อมูลแต่ละเรคคอร์ดในวีคเอนทิตีต้องบ่งบอกได้ว่าใคร/สิ่งใด เป็นเจ้าของ หรือใคร/สิ่งไหนเป็นต้นกำเนิด

5) Identifying Relationship ใช้สำหรับเชื่อมโยงความสัมพันธ์ระหว่าง วีคเอนทิตี กับ เอนทิตีที่เป็นเจ้าของ ซึ่งข้อกำหนดของการมีส่วนร่วมทางฝั่งที่เป็นวีคเอนทิตีจะเป็นแบบมีส่วนร่วมทั้งหมด (Total Participation) เสมอ เนื่องจาก ถ้าไม่มีเอนทิตีเจ้าของก็จะไม่มีวีคเอนทิตี ดังนั้น เมื่อมีข้อมูลวีคเอนทิตี จึงถูกเชื่อมโยง หรือมีส่วนร่วมกับข้อมูลฝั่งเจ้าของทุกเรคคอร์ด

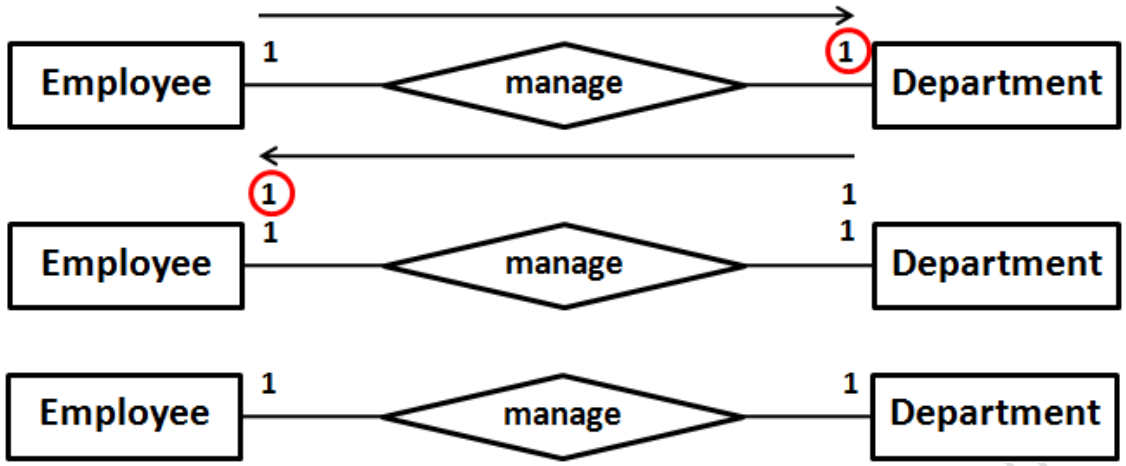
ตัวอย่างอีอาร์ไดอะแกรมของฐานข้อมูลบริษัท (Company Database) แสดงดังรูปที่ 7.8



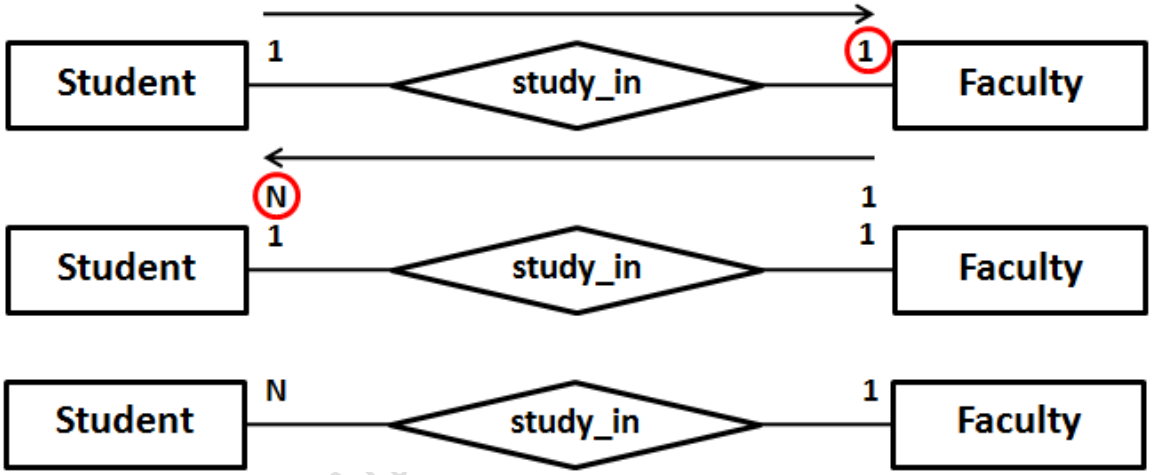
รูปที่ 7.3 ความสัมพันธ์ระหว่างสองเอนทิตี และความสัมพันธ์ระหว่างสามเอนทิตี



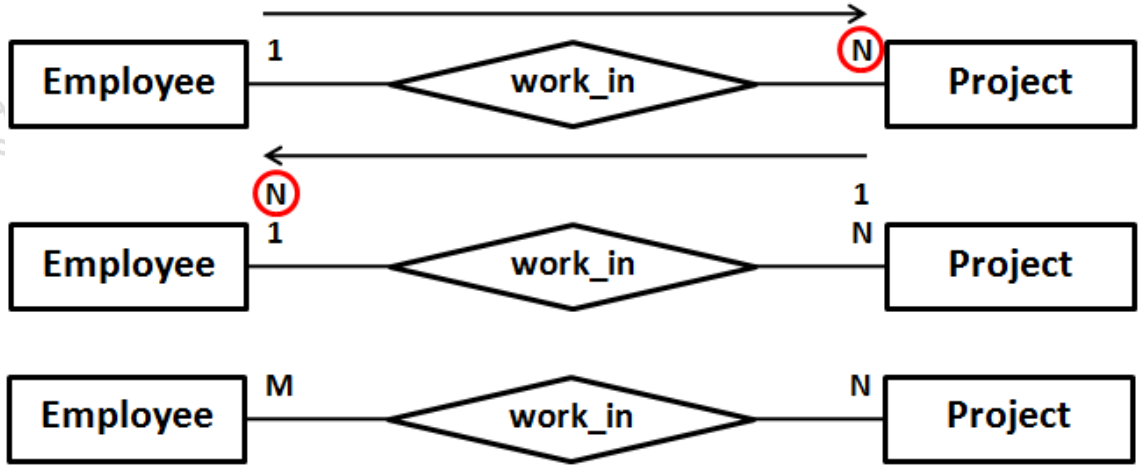
รูปที่ 7.4 ตัวอย่างเอนทิตีที่มีความสัมพันธ์กับตัวเอง



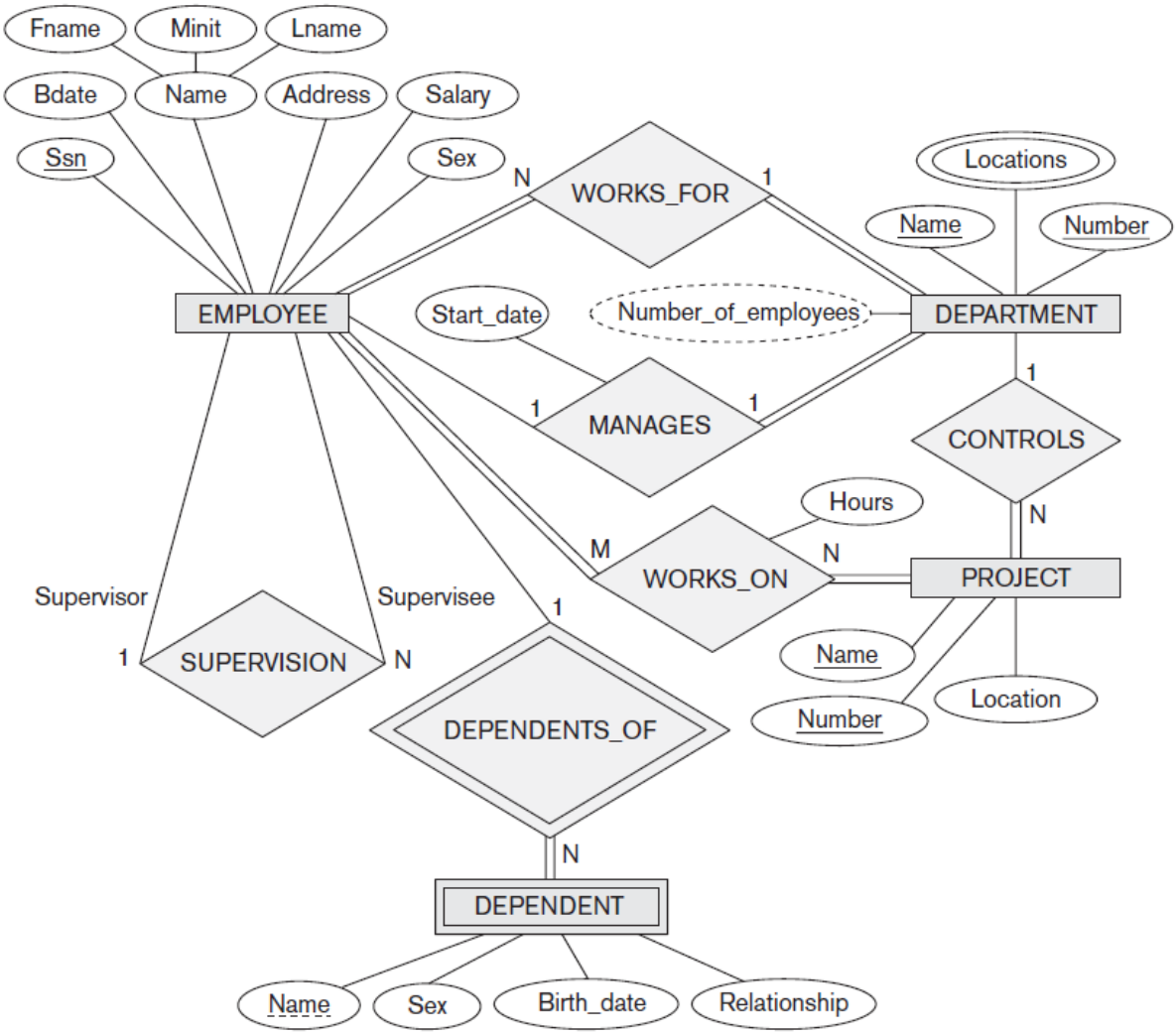
รูปที่ 7.5 ตัวอย่างของอัตราส่วนความสัมพันธ์แบบ One-to-one



รูปที่ 7.6 ตัวอย่างของอัตราส่วนความสัมพันธ์แบบ One-to-many



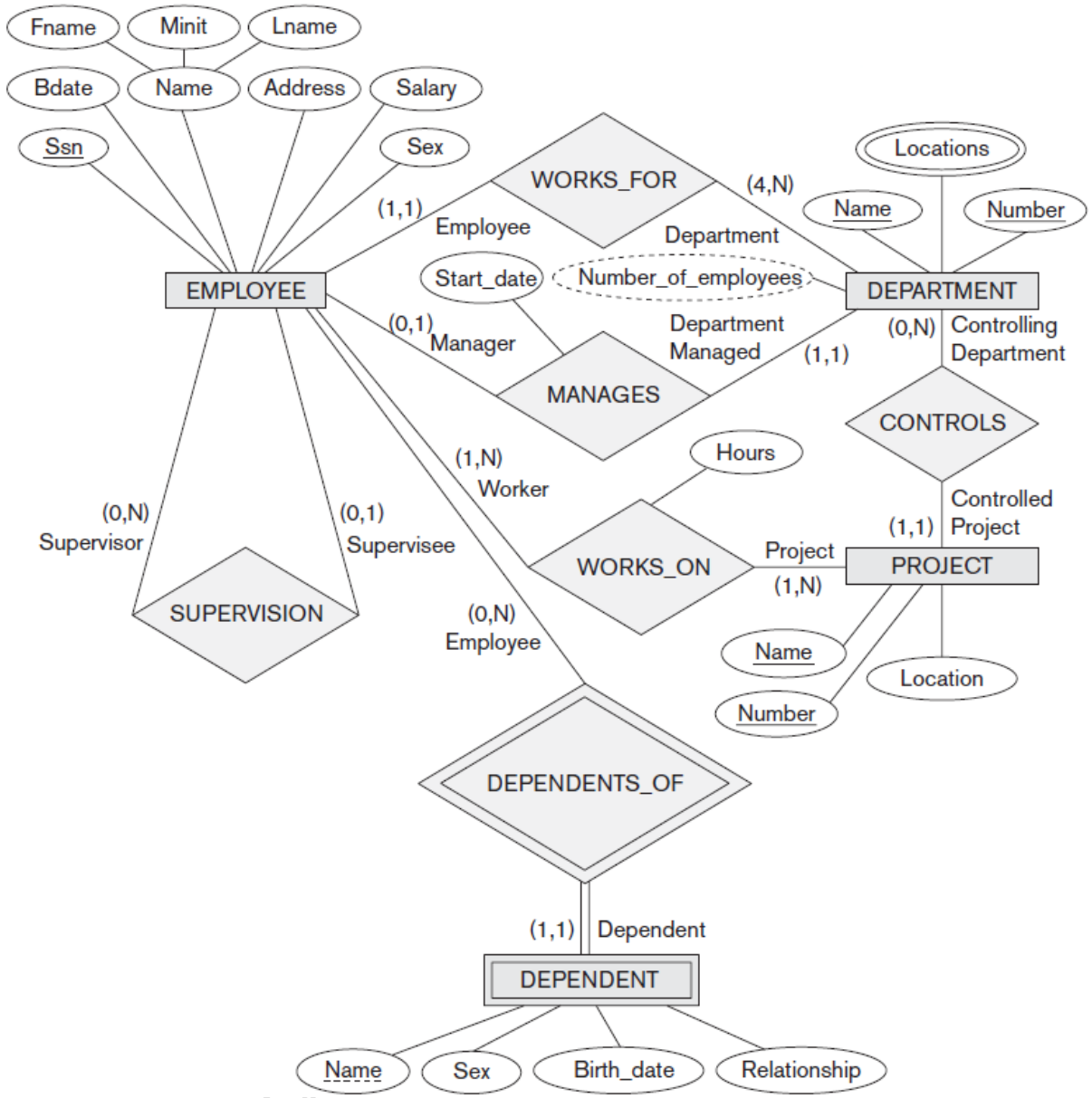
รูปที่ 7.7 ตัวอย่างของอัตราส่วนความสัมพันธ์แบบ Many-to-many



รูปที่ 7.8 ตัวอย่างอีอาร์ไดอะแกรมของฐานข้อมูลบริษัท [1]

การตั้งชื่อให้กับองค์ประกอบต่างๆที่อยู่ในอีอาร์ไดอะแกรม ควรตั้งชื่อให้เหมาะสม โดยมีแนวทางในการตั้งชื่อดังนี้

1) การตั้งชื่อองค์ประกอบต่างๆ เช่น เอนทิตี ความสัมพันธ์ แอททริบิวต์ และส่วนอื่นๆ ควรตั้งชื่อให้สื่อความหมายให้ชัดเจน
2) การตั้งชื่อเอนทิตี ควรใช้คำนาม
3) การตั้งชื่อความสัมพันธ์ ควรใช้คำกริยา
4) การตั้งชื่อความสัมพันธ์ระหว่างสองเอนทิตี ให้ตั้งในลักษณะที่อ่านจากซ้ายไปขวา และอ่านจากบนลงล่าง เช่น พนักงาน ทำงานใน โครงการ (Employee work_on Project) หรือ โครงการ ถูกกำหนดให้ พนักงาน (Project is_assigned_to Employee) เป็นต้น



รูปที่ 7.9 ตัวอย่างค่าต่ำสุดและค่าสูงสุดของอัตราส่วนความสัมพันธ์ระหว่างข้อมูล [1]

7.3 แนวคิดในการออกแบบอีอาร์ไดอะแกรม

ในการออกแบบอีอาร์ไดอะแกรมมีแนวคิดในการออกแบบ ดังขั้นตอนต่อไปนี้

1) ตอนแรกให้นึกถึง แอททริบิวต์ในแต่ละเอนทิตี ต่อจากนั้นให้ดูว่า แอททริบิวต์ในเอนทิตีหนึ่งมีความสัมพันธ์กับข้อมูลในเอนทิตีอื่นหรือไม่ หากมีความสัมพันธ์กัน ให้สร้างความสัมพันธ์โดยดูจากแอททริบิวต์ที่มีความเกี่ยวข้องกัน เช่น ลูกค้า ซื้อ สินค้า แอททริบิวต์ที่มีการอ้างอิง หรือเชื่อมโยงระหว่าง 2 เอนทิตีนี้คือ ลูกค้าคนไหน (อ้างอิงโดย รหัสลูกค้า) ซื้อ สินค้าอะไร (อ้างอิงโดย รหัสสินค้า) เป็นต้น

2) แอททริบิวต์ที่มีส่วนร่วมอยู่ในหลายๆ เอนทิตี เช่น แอททริบิวต์ของแผนก จะเกี่ยวข้องกับเอนทิตีพนักงาน เช่น พนักงานสังกัดแผนกใด หรือ เอนทิตีโครงการ เช่น แต่ละโครงการอยู่ภายใต้การควบคุมโดยแผนกใด เป็นต้น ให้แยกแอททริบิวต์ดังกล่าวออกจาก เอนทิตีต่างๆ และนำไปสร้างเป็น เอนทิตีแผนกแทน ในทางกลับกัน ถ้าในเอนทิตีมีเพียง 1 แอททริบิวต์ และสัมพันธ์กับ 1 เอนทิตี เท่านั้น ให้ย้ายแอททริบิวต์ดังกล่าวไปไว้ในเอนทิตีที่มีความสัมพันธ์กัน และยุบ เอนทิตีที่มี 1 แอททริบิวต์ ดังกล่าว

3) Cardinality Ratio และ Participation Constraints สามารถถูกแทนที่ด้วยค่า (min, max) ที่ใช้บ่งบอกค่าต่ำสุด และสูงสุดที่เป็นไปได้ของอัตราส่วนความสัมพันธ์ระหว่างเอนทิตี ดังแสดงในรูปที่ 7.9 ที่นำอีอาร์ไดอะแกรมในรูปที่ 7.8 มาแปลงให้อยู่ในรูป (min, max) และกำหนดชื่อบทบาทหน้าที่ (Role Name) ของแต่ละฝั่ง ด้วยแนวทางดังต่อไปนี้

   - สำหรับ Partial Paticipation (ความสัมพันธ์ที่เป็นเส้นเดี่ยว) เนื่องจากการมีส่วนร่วมของข้อมูลกับอีกฝั่งหนึ่งอยู่ในลักษณะที่มีส่วนร่วมบางข้อมูล เพราะฉะนั้น อาจจะมีส่วนร่วม หรือไม่มีส่วนร่วมเลยก็ได้ ดังนั้น ค่าต่ำสุดที่เป็นไปได้คือตั้งแต่ 0 เป็นต้นไป ส่วนค่าสูงสุดคงค่าเดิมที่อยู่บนเส้น เนื่องจาก ค่าเดิมที่อยู่บนเส้นคือค่าการมีส่วนร่วมที่เป็นไปได้สูงสุด
   - สำหรับ Total Participation (ความสัมพันธ์ที่เป็นเส้นคู่) เนื่องจากการมีส่วนร่วมของข้อมูลกับอีกฝั่งหนึ่งอยู่ในลักษณะที่มีส่วนร่วมทั้งหมด เพราะฉะนั้น อย่างน้อยต้องมีส่วนร่วม 1 เรคคอร์ด ดังนั้น ค่าต่ำสุดที่เป็นไปได้คือตั้งแต่ 1 เป็นต้นไป ส่วนค่าสูงสุดคงค่าเดิมที่อยู่บนเส้น เนื่องจาก ค่าเดิมที่อยู่บนเส้นคือค่าการมีส่วนร่วมที่เป็นไปได้สูงสุด

7.4 แนวทางในการออกแบบอีอาร์ไดอะแกรม

ในการออกแบบอีอาร์ไดอะแกรม สามารถออกแบบได้ 2 แนวทาง ดังต่อไปนี้คือ จากบนลงล่าง (Top-Down) หรือ จากล่างขึ้นบน (Bottom Up) ดังรายละเอียดต่อไปนี้

1) การออกแบบจากบนลงล่าง คือ การนึกถึงสิ่งต่างๆที่อยู่ในระบบที่ต้องการจัดเก็บ โดยที่ยังไม่ต้องนึกถึงรายละเอียดของสิ่งต่างๆเหล่านั้น เช่น หากต้องการสร้างระบบการลงทะเบียนเรียนของนักศึกษา ให้เรานึกถึง นักศึกษา รายวิชา ผู้สอน คณะ สาขาวิชา เป็นต้น ซึ่งสิ่งต่างๆเหล่านี้คือ เอนทิตีของระบบ หลังจากนั้น ให้นึกถึงคุณสมบัติเฉพาะ หรือแอททริบิวต์ ของแต่ละเอนทิตีว่าเราสนใจ จะจัดเก็บแอททริบิวต์อะไรบ้างที่สำคัญ และจำเป็นต่อระบบ
2) การออกแบบจากล่างขึ้นบน คือ ให้นึกถึงสิ่งต่างๆในระบบ และรายละเอียดของสิ่งต่างๆที่ต้องการจัดเก็บในระบบ หรือรายละเอียดต่างๆที่เกี่ยวข้องกับระบบเท่าที่เป็นไปได้ เช่น หากต้องการสร้างระบบการลงทะเบียนเรียนของนักศึกษา ให้นึกถึง นักศึกษา และรายละเอียดที่เกี่ยวข้องกับนักศึกษา ไปพร้อมๆกัน ยกตัวอย่างเช่น รหัสนักศึกษา ชื่อ-สกุล ที่อยู่ รายวิชาที่

ลงทะเบียน ปีการศึกษา ภาคการศึกษาที่ลงทะเบียน เกรดที่ได้ GPA ของแต่ละภาคการศึกษา ใครเป็นผู้สอนในแต่ละรายวิชา คณะที่นักศึกษาสังกัด สาขาวิชาที่นักศึกษาสังกัด เป็นต้น หลังจากนั้น ให้ทำการแยกกลุ่มของรายละเอียดที่กล่าวมาข้างต้น เช่น อะไรที่อยู่ในกลุ่มของนักศึกษา กลุ่มรายวิชา กลุ่มคณะ หลังจากนั้น ให้ตั้งชื่อกลุ่ม ซึ่งจะได้ 1 เอนทิตี ต่อ 1 กลุ่ม

หลังจากที่ได้ เอนทิตี และแอททริบิวต์ แล้ว ทั้งสองแนวทาง ให้ดำเนินการตามลำดับขั้นตอนดังต่อไป

- ให้วิเคราะห์เอนทิตีที่ได้ทั้งหมดว่าควรจะอยู่ในลักษณะ Regular Entity หรือ Weak Entity
- แต่ละ Entity ให้จำแนกประเภทของแอททริบิวต์แต่ละตัวตามที่กล่าวไว้ในหัวข้อ 7.2 และให้หาว่า Attribute ใดสามารถเป็น Key Attribute ได้
- หาความสัมพันธ์ระหว่าง Entity และตั้งชื่อ Relationship ให้เหมาะสม
- กำหนดค่า Cardinality Ratio
- กำหนดความสัมพันธ์ในแต่ละฝั่งว่าอยู่ในรูป Partial Participation หรือ Total Participation
- แปลง Cardinality Ratio และ Participation Constraint ให้อยู่ในรูป (min, max)

7.5 ความสัมพันธ์ระหว่างเอนทิตีที่มีดีกรีความสัมพันธ์ตั้งแต่สามขึ้นไป

ในหัวข้อนี้ จะกล่าวถึงความแตกต่างระหว่างความสัมพันธ์ที่มีดีกรีสอง (Binary Relationship) และที่มีดีกรีมากกว่า 2 คือตั้งแต่ 3 ขึ้นไป (Ternary Relation and n-ary Relationship) เมื่อใดที่จะเลือกใช้ดีกรี 2 และเมื่อใดจะเลือกใช้ดีกรีสูงกว่า 2 และจะกำหนดข้อบังคับสำหรับความสัมพันธ์ที่มีดีกรีมากกว่า 2 ได้อย่างไร

เครื่องหมายที่ใช้นำเสนอความสัมพันธ์ของข้อมูล 3 เอนทีตี (Ternary Relationship) ในอีอาร์ไดอะแกรม แสดงดังในรูปที่ 7.10 (a) ความสัมพันธ์ที่ชื่อ SUPPLY ในรูปที่ 7.10(a) เชื่อมโยงข้อมูลจาก 3 เอนทิตี คือ SUPPLIER PROJECT และ PART ซึ่งความสัมพันธ์ SUPPLY ในที่นี้มีความหมายว่า SUPPLIER (ผู้ผลิต) SUPPLY (จัดส่ง) PART (ชิ้นส่วน) ให้กับ PROJECT (โครงการ) โดยทั่วไปแล้วสำหรับความสัมพันธ์ที่มีดีกรี n (n-ary Relationship) ก็จะมีการเชื่อมโยงจากความสัมพันธ์ไปยังเอนทิตีที่สัมพันธ์กัน โดยมีเส้นเชื่อมโยงจากความสัมพันธ์จำนวน n เส้น

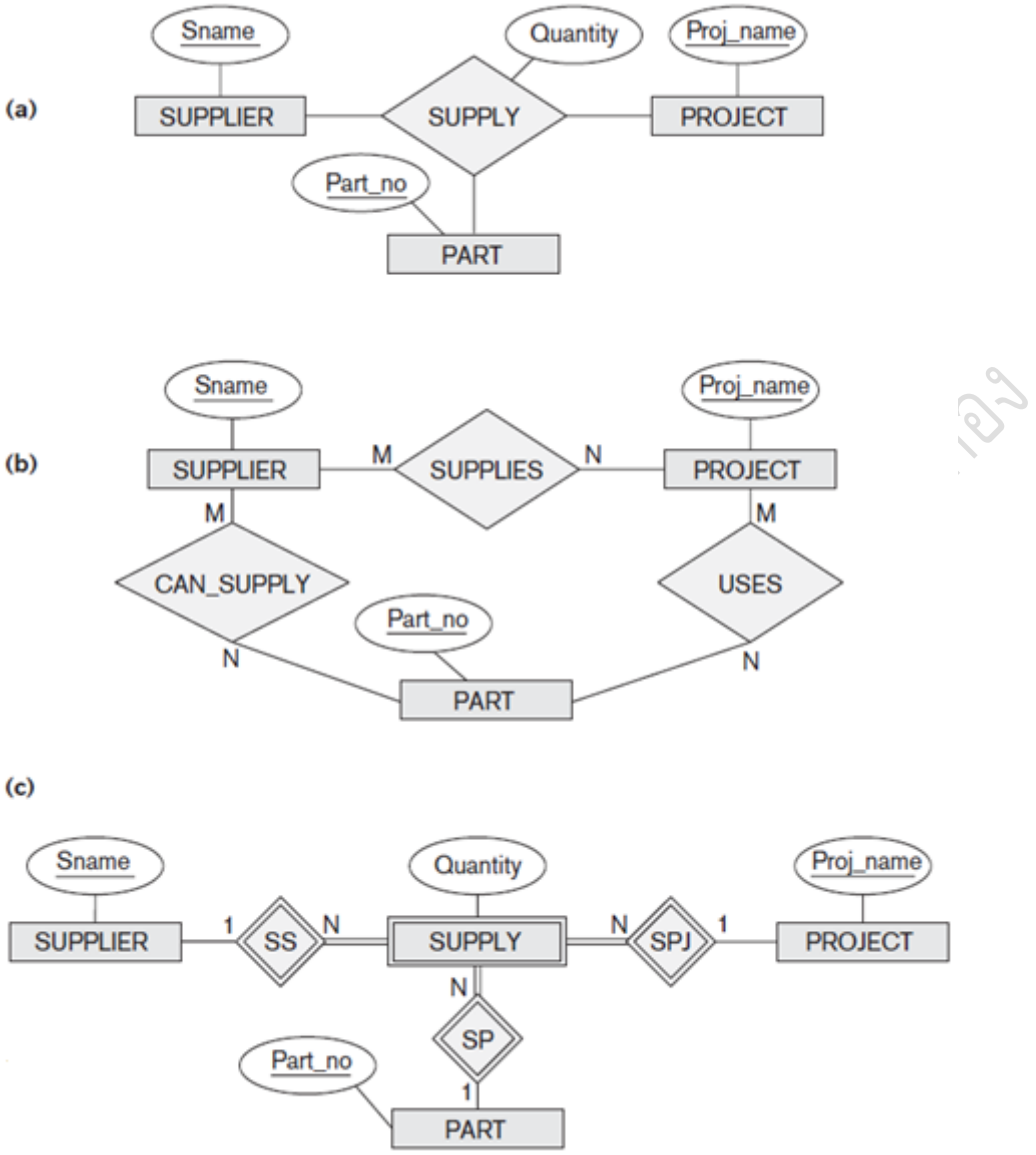
โดยปกติแล้ว การนำเสนอความสัมพันธ์แบบ Ternary Relationship มีความแตกต่างจาก การนำเสนอความสัมพันธ์แบบ Binary Relationship แบบ 3 คู่ ดังตัวอย่างแสดงในรูปที่ 7.10(b) ซึ่งในรูป 7.10(a) นำเสนอความสัมพันธ์ของ SUPPLIER PART และ PROJECT ด้วยความสัมพันธ์ชื่อ SUPPLY ซึ่งอยู่ในลักษณะ Ternary Relationship แต่ในรูปที่ 7.10(b) นำเสนอความสัมพันธ์ระหว่าง SUPPLIER PART และ PROJECT ด้วย 3 คู่ของ Binary Relationship โดยแต่ละคู่ มีรายละเอียด ดังต่อไปนี้ คือ

1) คู่แรก คือ ระหว่าง เอนทิตี SUPPLIER (ผู้ผลิต) กับ PART (ชิ้นส่วน) ด้วยความสัมพันธ์สามารถจัดส่ง (CAN_SUPPLY) ดังนี้คือ ผู้ผลิต สามารถจัดส่ง ชิ้นส่วน (SUPPLIER CAN_SUPPLY PART)
2) คู่ที่สอง คือ ระหว่าง เอนทิตี SUPPLIER กับ PROJECT (โครงการ) ด้วยความสัมพันธ์จัดส่งให้กับ (SUPPIES) ดังนี้คือ ผู้ผลิต จัดส่งให้กับ โครงการ (SUPPLIER SUPPIES PROJECT)
3) คู่ที่สาม คือ ระหว่าง เอนทิตี PROJECT กับ PART ด้วยความสัมพันธ์ ใช้ (USES) ดังนี้คือ โครงการ ใช้ ชิ้นส่วน (PROJECT USES PART)

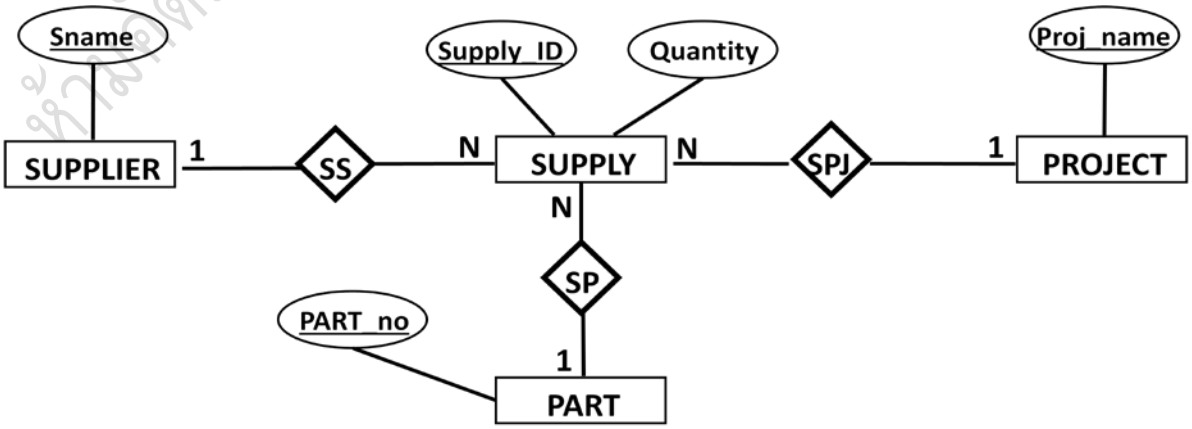
ทั้งนี้ในการออกแบบฐานข้อมูล ขึ้นอยู่กับลักษณะของความสัมพันธ์ หากความสัมพันธ์เชื่อมโยงมากกว่า 2 เอนทิตีจริง ให้นำเสนอในรูป Ternary Relationship หรือ n-ary Relationship ดังตัวอย่างแสดงในรูปที่ 7.10 (a) หากความสัมพันธ์ของเอนทิตีแต่ละคู่เป็นความสัมพันธ์ที่แตกต่างกัน ให้แยกความสัมพันธ์ออกเป็น Binary Relationship 3 คู่ ดังตัวอย่างแสดงในรูปที่ 7.10 (b) ดังนั้นในการออกแบบให้ตัดสินใจให้สอดคล้องกับความหมายของความสัมพันธ์เป็นหลัก สำหรับ n-ary Relationship อาจจะประกอบด้วย Ternary Relationship และ 1 ถึง 2 หรือมากกว่า 2 คู่ของ Binary Relationship ซึ่งทั้งนี้ขึ้นอยู่กับความหมายของความสัมพันธ์ของแต่ละกลุ่ม

เครื่องมือที่ใช้ในการออกแบบฐานข้อมูลเชิงสัมพันธ์บางตัว อนุญาตให้ออกแบบความสัมพันธ์ในลักษณะ Binary Relationship เท่านั้น ในกรณีที่มีดีกรีความสัมพันธ์มากกว่า 2 ตัวอย่างเช่น SUPPLY ในรูป 7.10 (a) ให้นำเสนอความสัมพันธ์ SUPPLY อยู่ในรูปของ Weak Entity แทน และสร้าง Identifying Relationship 3 คู่ ดังแสดงในรูปที่ 7.10 (c) โดยที่ Weak Entity ชื่อ SUPPLY ไม่มี Partial Key สำหรับ Owner ของ SUPPLY คือ SUPPLIER PART และ PROJECT ดังนั้น SUPPLY จะประกอบด้วย Key ของ Owner ซึ่งประกอบด้วย Sname PART_no และ Proj_name

สำหรับความสัมพันธ์ชื่อ SUPPLY มีอีก 1 ทางเลือกคือ แปลง SUPPLY ให้เป็น Regular Entity โดยการสร้าง Surrogate Key ชื่อ SUPPLY_ID เพื่อให้ SUPPLY มี Key Attribute เป็นของตัวเอง และ Supply_ID สามารถใช้แทน Sname PART_no และ Proj_name ดังแสดงในรูปที่ 7.11 และหลังจาก Mapping Key ทั้งสามตัวจะกลายเป็น Foreign Key ในเอนทิตี SUPPLY



รูปที่ 7.10 ตัวอย่างการนำเสนอความสัมพันธ์ของ 3 เอนทิตี



รูปที่ 7.11 การแปลง SUPPLY ที่อยู่ในรูป Weak Entity ให้เป็น Regular Entity