

# บทที่ 4 แบบจำลองเชิงสัมพันธ์เอ็นทิตี

## วัตถุประสงค์ของบทเรียน

- ศึกษาเกี่ยวกับส่วนประกอบที่สำคัญของแบบจำลองข้อมูลเชิงสัมพันธ์เอ็นทิตี
- ศึกษาเกี่ยวกับวิธีการในการนิยาม การกลั่นกรอง และการรวมเอ็นทิตีต่างๆเข้าด้วยกัน
- ศึกษาสาเหตุที่ส่วนประกอบของแผนผังเชิงสัมพันธ์เอ็นทิตีจะมีความเกี่ยวข้องเนื่องกับการออกแบบและการดำเนินการสร้างฐานข้อมูล
- ศึกษาเกี่ยวกับการปรับปรุงแนวความคิดให้ตรงกันระหว่างบุคคลที่มีส่วนเกี่ยวข้องกับการออกแบบฐานข้อมูลสำหรับแอปพลิเคชันต่างๆ

## เนื้อหาของบทเรียน

แบบจำลองข้อมูลเชิงสัมพันธ์เอ็นทิตี Connectivity และ Cardinality ระดับความเข้มของความสัมพันธ์ การมีส่วนร่วมกับความสัมพันธ์ ระดับความสัมพันธ์ recursive relationship การสร้างเอ็นทิตีใหม่สำหรับรูปแบบความสัมพันธ์แบบ M:N ขั้นตอนการออกแบบแผนภาพเชิงสัมพันธ์เอ็นทิตี

## กิจกรรมการเรียนรู้-การสอน

- อธิบายพร้อมยกตัวอย่างประกอบ
- ศึกษาจากเอกสารคำสอน
- ฝึกปฏิบัติการตามที่มอบหมาย
- ทำแบบฝึกหัดท้ายบท

## อุปกรณ์ที่ใช้ในการเรียนรู้-การสอน

- เอกสารคำสอน
- เครื่องคอมพิวเตอร์
- เครื่องฉายภาพสไลด์

## การวัดและประเมินผล

- การตอบคำถามระหว่างการเรียนรู้-การสอน
- การทำแบบทดสอบย่อยท้ายบท
- การตรวจงานตามที่มอบหมาย

การออกแบบแบบจำลองข้อมูลจะเป็นเพียงส่วนแรกหรือขั้นตอนแรกของการออกแบบฐานข้อมูลเท่านั้น นอกเหนือจากการออกแบบดังกล่าวแล้ว ยังมีขั้นตอนอีกเป็นจำนวนมาก อาทิเช่น การปรับเปลี่ยนแบบจำลองให้เป็นฐานข้อมูลจริง การตรวจสอบความถูกต้องของข้อมูลหลังการสร้างฐานข้อมูล การปรับปรุงการทำงานต่างๆเพื่อเพิ่มประสิทธิภาพการทำงานให้กับฐานข้อมูล และอื่นๆ แต่อย่างไรก็ดี ถึงแม้จะมีขั้นตอนในการออกแบบหลายขั้นตอน การออกแบบแบบจำลองข้อมูลก็ยังคงเป็นขั้นตอนที่สำคัญและมีประโยชน์ค่อนข้างมาก ที่ซึ่งจะทำให้เราเข้าใจทั้งภาพรวมและรายละเอียดของข้อมูลและโครงสร้างการจัดเก็บข้อมูล ดังนั้น ในบทนี้เราจะทำการศึกษาเกี่ยวกับแบบจำลองข้อมูลเชิงสัมพันธ์เอนทิตี (entity relationship model, ERM) ที่ซึ่งจะทำการประยุกต์ใช้แผนภาพความสัมพันธ์เอนทิตี (entity relationship diagram, ER diagram หรือ ERD) เพื่อแสดงถึงส่วนประกอบต่างๆของโครงสร้างการจัดเก็บข้อมูล อาทิเช่น เอนทิตีและความสัมพันธ์ต่างๆระหว่างเอนทิตี นอกจากนั้นเราจะทำการศึกษาเกี่ยวกับการวิเคราะห์แผนภาพความสัมพันธ์เอนทิตีเพื่อทำการพิจารณาว่าแผนภาพใดบ้างเป็นแผนภาพที่ดี และแผนภาพใดบ้างเป็นแผนภาพที่ไม่ดี อันนำมาซึ่งความเข้าใจในการออกแบบแผนภาพความสัมพันธ์เอนทิตีที่ซึ่งจะช่วยให้เราสามารถออกแบบฐานข้อมูลที่ดีได้

## 4.1 แบบจำลองข้อมูลเชิงสัมพันธ์เอนทิตี

จากเนื้อหาในบทก่อนๆหน้าจะทำให้เราทราบว่าแบบจำลองข้อมูลเชิงสัมพันธ์เอนทิตีจะทำการประยุกต์ใช้ ERD ในการแสดงถึงโครงสร้างของการจัดเก็บข้อมูล ERD หนึ่งๆจะประกอบไปด้วยเอนทิตี แอทริบิวต์ และความสัมพันธ์ระหว่างเอนทิตี นอกจากนั้น เรายังทราบอีกว่า ERD จะมีวิธีการแสดงการออกแบบได้หลายวิธี เช่น 1) ERD ที่คิดค้นจาก Chen, 2) ERD ที่คิดค้นจาก Crow's Foot และ 3) การประยุกต์ใช้ UML ตามลำดับ ดังนั้นเนื้อหาในบทนี้จะแสดงถึงตัวอย่างและวิธีการออกแบบ ERD สองวิธีแรกข้างต้นที่ซึ่งจะทำให้เราเห็นข้อดี-เสีย และการประยุกต์ใช้งานวิธีการต่างๆมากขึ้นและยังสามารถช่วยให้เราประยุกต์ใช้วิธีการออกแบบ ERD ได้อย่างถูกต้อง

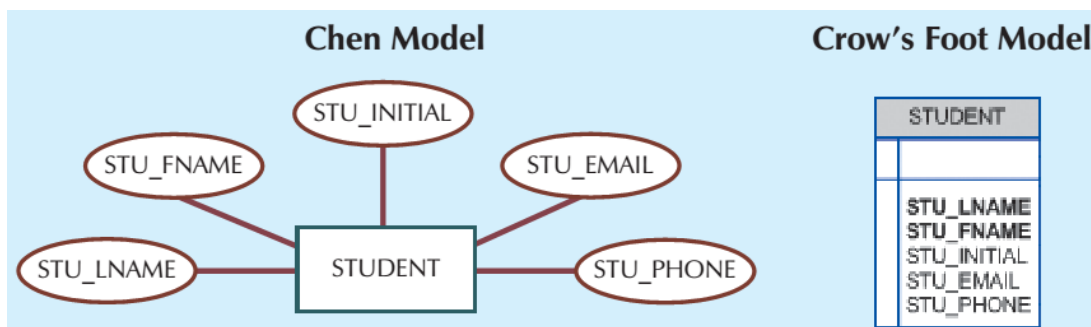
### 4.1.1 เอนทิตี

เอนทิตีจะเป็นสิ่งที่ใช้แทนวัตถุหนึ่งๆที่ซึ่งจะสอดคล้องกับตารางข้อมูลหนึ่งในฐานข้อมูล แต่ถ้าเราทำการพิจารณาถึงข้อมูลแถวหนึ่งที่ปรากฏในตารางข้อมูลหนึ่งๆ(เอนทิตีหนึ่งๆ) เราจะเรียกข้อมูลแถวหนึ่งๆว่า entity instance หรือ entity occurrence ตามลำดับ ดังนั้น เมื่อเราทำการออกแบบ ERD เราจะต้องทำการกำหนดเอนทิตีเพื่อใช้แทนวัตถุที่ต้องการจัดเก็บข้อมูล เอนทิตีหนึ่งๆมักจะเขียนอยู่ในรูปสี่เหลี่ยม และชื่อของเอนทิตีมักเขียนด้วยตัวอักษรพิมพ์ใหญ่ทั้งหมด (ดังแสดงในรูป 4.1)

### 4.1.2 แอทริบิวต์

แอทริบิวต์จะเป็นสิ่งที่บ่งบอกถึงลักษณะต่างๆของเอนทิตี โดยเอนทิตีหนึ่งๆสามารถมีได้หลายแอทริบิวต์ เช่น เอนทิตีของนักเรียน (STUDENT) จะประกอบไปด้วยแอทริบิวต์ ชื่อ (STU\_FNAME), ชื่อกลาง (STU\_INITIAL) และ นามสกุล (STU\_LNAME) เป็นต้น หลังจากทำการระบุถึงแอทริบิวต์ต่างๆที่ใช้ในการบ่ง

บอกถึงลักษณะของเอนทิตีหนึ่งๆแล้ว เราสามารถออกแบบ ERD ตามกรอบความคิดของ Chen และ Crow's Foot ได้ดังรูป 4.1 ที่ซึ่งจะมีวิธีการแสดงที่แสดงถึงเอนทิตีและแอททริบิวในเอนทิตีที่แตกต่างกัน โดยจากรูปเราจะสังเกตได้ว่าแต่ละแอททริบิวที่ถูกแสดงภายใต้แนวความคิดของ Chen จะถูกเขียนชื่อของแอททริบิวนั้นๆอยู่ในวงรีที่ซึ่งเชื่อมต่อกับสี่เหลี่ยมที่เป็นแสดงถึงเอนทิตีและชื่อเอนทิตี แต่ในส่วนของแนวความคิดของ Crow's Foot จะทำการสร้างกล่องสี่เหลี่ยมภายใต้กล่องสี่เหลี่ยมที่แสดงถึงชื่อเอนทิตี โดยเมื่อเปรียบเทียบวิธีการแสดงเอนทิตีและแอททริบิว เราจะสังเกตได้ว่าวิธีการของ Chen จะใช้พื้นที่ค่อนข้างมาก ด้วยเหตุนี้จึงทำให้ซอฟต์แวร์ที่ใช้ในการสร้าง ERD จะทำการประยุกต์ใช้แนวความคิดของ Crow's Foot ในการแสดงถึงเอนทิตีและแอททริบิวต่างๆเพื่อลดการใช้พื้นที่ในการแสดงถึงเอนทิตีหนึ่งๆ



รูปที่ 4.1 ตัวอย่างการออกแบบเอนทิตี STUDENT จากแนวคิดของ Chen และ Crow's Foot

ในการกำหนดแอททริบิวให้กับเอนทิตีหนึ่งๆ เราจะต้องพิจารณาว่าแอททริบิวนั้นๆมีความจำเป็นหรือมีความสำคัญต่อการดำเนินการต่างๆภายในองค์กรหนึ่งๆหรือไม่ ด้วยเหตุนี้ เราทำการแบ่งประเภทของแอททริบิวเป็นสองประเภทย่อยดังนี้

- **แอททริบิวที่สำคัญ (required attribute)**—จะเป็นแอททริบิวที่จำเป็นต้องมีค่าปรากฏในแอททริบิวนั้นๆ ห้ามเป็นค่าว่าง (NULL) โดยเด็ดขาด โดยจากรูป 4.1 เราจะสังเกตได้ว่ามีสองแอททริบิวใน ERD แบบ Crow's Foot ที่ถูกเน้นย้ำ (ด้วยการทำให้ชื่อแอททริบิวเป็นตัวหนา) ที่ซึ่งจะเป็นแอททริบิวที่สำคัญ นั่นคือ STU\_LNAME และ STU\_FNAME ที่ซึ่งจะเป็นการเน้นย้ำว่านักเรียนคนหนึ่งๆจะต้องมีชื่อและนามสกุล แต่สำหรับแอททริบิวอื่นๆจะเป็นแอททริบิวที่ไม่สำคัญเนื่องจากนักเรียนคนหนึ่งๆอาจไม่มีชื่อกลาง (STU\_INITIAL) อาจยังไม่มีเบอร์โทรศัพท์ (STU\_PHONE) และ/หรืออีเมลล์ (STU\_EMAIL) ตามลำดับ
- **แอททริบิวที่เป็นส่วนเสริม (optional attribute)**—จะเป็นแอททริบิวไม่สำคัญหรือมีความสำคัญน้อย ดังนั้น เราอาจยอมให้ค่าที่ปรากฏในแอททริบิวเหล่านี้มีค่าว่างได้

ในแอททริบิวหนึ่งๆจะมีขอบเขตของค่าที่ปรากฏ (attribute domain) ที่ซึ่งขอบเขตของแอททริบิวจะสามารถเป็นค่าใดก็ได้ ตัวอย่างเช่น ขอบเขตของแอททริบิวเกรดเฉลี่ย (GPA) จะมีค่าอยู่ระหว่าง (0.0 – 4.0) เนื่องจากเกรดเฉลี่ยต่ำสุดที่เป็นไปได้คือ 0.0 และเกรดเฉลี่ยสูงสุดที่เป็นไปได้คือ 4.0 อีกตัวอย่างคือ ขอบเขต

ของแอทริบิวต์ (GENDER) ของนักเรียนจะสามารถมีค่าที่เป็นไปได้ 2 ค่าคือ M หรือ F ที่ซึ่งจะแสดงถึงเพศชายและเพศหญิง เป็นต้น นอกเหนือจากการกำหนดขอบเขตให้กับแอทริบิวต์หนึ่งๆแล้ว แอทริบิวต์ต่างๆที่ไม่ใช่แอทริบิวต์เดียวกันสามารถมีขอบเขตร่วมกันหรือขอบเขตเหมือนกันได้ ตัวอย่างเช่น แอทริบิวต์ที่อยู่ของนักเรียนและ แอทริบิวต์ที่อยู่ของอาจารย์สามารถมีขอบเขตแอทริบิวต์เหมือนกันได้ เนื่องจากอาจมีที่อยู่ใกล้เคียงหรืออยู่เดียวกันได้

นอกเหนือจากการกำหนดขอบเขตให้กับแต่ละแอทริบิวต์แล้ว แบบจำลองข้อมูลเชิงสัมพันธ์เอ็นทีดีจะทำการประยุกต์ใช้ *identifiers (primary keys, PKs)* ที่ซึ่งจะเป็นแอทริบิวต์หนึ่งๆหรือกลุ่มของแอทริบิวต์ที่ใช้สำหรับแยกความแตกต่างของ entity instance หนึ่งๆจาก entity instance อื่นๆ (แสดงถึงความเป็นเอกลักษณ์ของ entity instance หนึ่งๆ) ตัวอย่างเช่น ตารางข้อมูลในรูป 4.2 จะทำการกำหนดให้แอทริบิวต์ CLASS\_CODE ทำหน้าที่เป็น primary key ของตารางข้อมูล แต่อย่างไรก็ตามเราอาจทำการประยุกต์ใช้ **composite identifier** ที่ซึ่งจะเป็นกลุ่มของแอทริบิวต์ที่มากกว่าหนึ่งแอทริบิวต์ทำหน้าที่เป็น primary key ของตารางได้ ตัวอย่างเช่น กำหนดให้การรวมกันของแอทริบิวต์ CRS\_CODE และ CLASS\_SECTION ทำหน้าที่เป็น primary key แทนที่แอทริบิวต์ CLASS\_CODE ได้

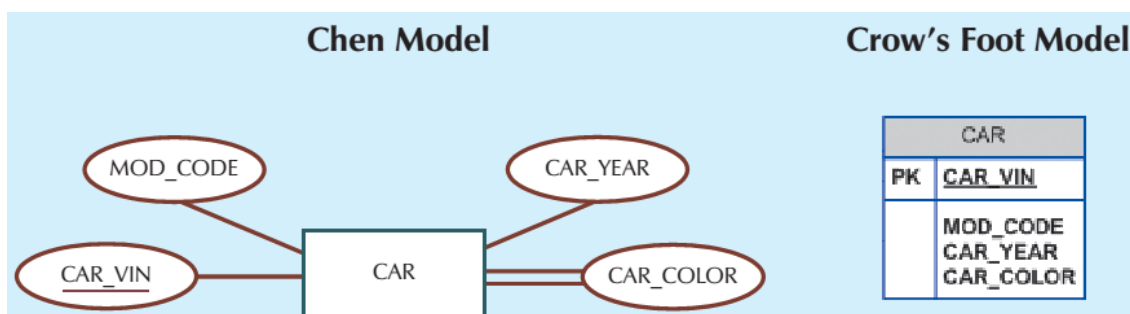
| CLASS_CODE | CRS_CODE | CLASS_SECTION | CLASS_TIME           | ROOM_CODE | PROF_NUM |
|------------|----------|---------------|----------------------|-----------|----------|
| 10012      | ACCT-211 | 1             | MWF 8:00-8:50 a.m.   | BUS311    | 105      |
| 10013      | ACCT-211 | 2             | MWF 9:00-9:50 a.m.   | BUS200    | 105      |
| 10014      | ACCT-211 | 3             | TTh 2:30-3:45 p.m.   | BUS252    | 342      |
| 10015      | ACCT-212 | 1             | MWF 10:00-10:50 a.m. | BUS311    | 301      |
| 10016      | ACCT-212 | 2             | Th 6:00-8:40 p.m.    | BUS252    | 301      |
| 10017      | CIS-220  | 1             | MWF 9:00-9:50 a.m.   | KLR209    | 228      |
| 10018      | CIS-220  | 2             | MWF 9:00-9:50 a.m.   | KLR211    | 114      |
| 10019      | CIS-220  | 3             | MWF 10:00-10:50 a.m. | KLR209    | 228      |
| 10020      | CIS-420  | 1             | W 6:00-8:40 p.m.     | KLR209    | 162      |
| 10021      | QM-261   | 1             | MWF 8:00-8:50 a.m.   | KLR200    | 114      |
| 10022      | QM-261   | 2             | TTh 1:00-2:15 p.m.   | KLR200    | 114      |
| 10023      | QM-362   | 1             | MWF 11:00-11:50 a.m. | KLR200    | 162      |
| 10024      | QM-362   | 2             | TTh 2:30-3:45 p.m.   | KLR200    | 162      |
| 10025      | MATH-243 | 1             | Th 6:00-8:40 p.m.    | DRE155    | 325      |

รูปที่ 4.2 ตัวอย่างตารางข้อมูลชั้นเรียน (CLASS)

นอกเหนือจากการแบ่งประเภทของแอทริบิวต์ข้างต้น ที่ซึ่งจะแบ่งออกเป็นสองประเภทหลักคือ แอทริบิวต์ที่สำคัญ และแอทริบิวต์ที่เป็นส่วนเสริม เรายังสามารถแบ่งแอทริบิวต์ออกเป็นประเภทอื่นๆได้ดังนี้

- **composite attribute**—จะเป็นแอทริบิวต์ที่เกิดจากการแบ่งส่วนของแอทริบิวต์หนึ่งเพื่อให้ได้แอทริบิวต์อื่นๆเพิ่มเติม ตัวอย่างเช่น แอทริบิวต์ที่อยู่จะสามารถแบ่งเป็นแอทริบิวต์ถนน ตำบล อำเภอ จังหวัด รหัสไปรษณีย์ได้ หรือในส่วนของแอทริบิวต์หมายเลขโทรศัพท์จะสามารถแบ่งเป็นแอทริบิวต์ย่อยตามรหัสเขตและหมายเลขโทรศัพท์ปกติได้

- **simple attribute**—จะเป็นแอทริบิวต์ที่ไม่สามารถแบ่งออกเป็นส่วนย่อยๆหรือแอทริบิวต์ย่อยได้ ตัวอย่างเช่น แอทริบิวต์อายุ เพศ สถานะทางสังคม (หมายเหตุ—เพื่อให้มีความสะดวกในการเข้าถึงและจัดการกับข้อมูล เราควรที่จะต้องทำการปรับเปลี่ยน composite attribute ให้กลายเป็น simple attribute)
- **single-valued attribute**—จะเป็นแอทริบิวต์ที่มีค่าที่ปรากฏในแอทริบิวต์เพียงค่าเดียวเท่านั้น เช่น บุคคลหนึ่งๆสามารถมีหมายเลขบัตรประชาชนได้หมายเลขเดียวเท่านั้น หรือสินค้าหนึ่งๆจะมีรหัสสินค้าเพียงรหัสเดียวเท่านั้น โดยแอทริบิวต์ที่มีลักษณะเป็นแบบ single-valued attribute จะสามารถเป็นแอทริบิวต์แบบ composite attribute หรือ simple attribute ได้
- **multivalued attribute**—จะเป็นแอทริบิวต์ที่มีค่าที่ปรากฏในแอทริบิวต์ได้หลายค่า ตัวอย่างเช่น บุคคลหนึ่งๆสามารถทำการศึกษาในหลายมหาวิทยาลัย (หนึ่งมหาวิทยาลัยสำหรับระดับปริญญาตรี และอีกหนึ่งมหาวิทยาลัยสำหรับระดับปริญญาโท และ/หรือ ปริญญาเอก เป็นต้น) บ้านหลังหนึ่งๆอาจมีหลายเบอร์โทรศัพท์ รถคันหนึ่งๆอาจมีหลายสี (สีหลังคา สีตัวถัง สีกันชนหน้า-หลัง เป็นต้น) ดังนั้น ในการออกแบบ ERD เราจะสามารถดำเนินการได้ดังรูป 4.3 ที่ซึ่งการออกแบบตามแนวความคิดของ Chen จะทำการใช้สองเส้นตรงเพื่อบ่งบอกถึงแอทริบิวต์ที่เป็น multivalued attribute (แอทริบิวต์ CAR\_COLOR จะมีลักษณะเป็น multivalued attribute) แต่สำหรับการออกแบบตามแนวความคิดของ Crow's Foot จะไม่บ่งบอกถึงแอทริบิวต์ที่มีลักษณะเป็น multivalued attribute เลย



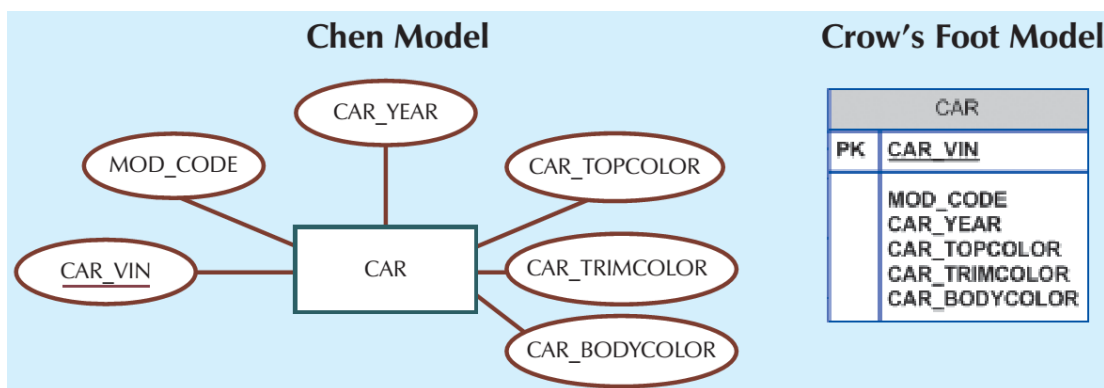
รูปที่ 4.3 ตัวอย่างการแสดงถึง multivalued attribute ใน ERD

การดำเนินการจัดเก็บข้อมูลใน multivalued attribute เราจะไม่สามารถจัดเก็บข้อมูลเหล่านี้ได้โดยตรง เนื่องจากข้อมูลที่ปรากฏในคอลัมน์หนึ่งของแถวข้อมูลหนึ่งๆจะแสดงถึงข้อมูลเพียงข้อมูลเดียวเท่านั้น ด้วยเหตุนี้ เราจึงต้องทำการเลือกใช้วิธีในการจัดการกับ multivalued attribute จากหนึ่งในสองวิธี ดังนี้

- **การเพิ่มแอทริบิวต์ใหม่ให้กับตารางข้อมูลเดิม**—โดยเราอาจทำการเพิ่มแอทริบิวต์ในตารางข้อมูลให้มีจำนวนเท่ากับจำนวนค่าที่สามารถปรากฏได้ใน multivalued attribute (หนึ่งแอทริบิวต์ใช้สำหรับจัดเก็บหนึ่งค่าข้อมูล) ตัวอย่างเช่น แอทริบิวต์ CAR\_COLOR ของเอ็นทิตีรถยนต์ (CAR) จะมีลักษณะเป็น multivalued attribute ที่

ซึ่งจะสามารถทำการเพิ่มแอททริบิวต์เพิ่มเติมได้เป็น CAR\_TOPCOLOR (สีหลังคา), CAR\_BODYCOLOR (สีตัวถัง), และ CAR\_TRIMCOLOR (สีกันชนหน้า-หลัง) ดังแสดงในรูป 4.4

การเพิ่มแอททริบิวต์ต่างๆดูเหมือนจะสามารถทำงานได้ดี แต่การประยุกต์ใช้งานวิธีนี้จะก่อให้เกิดปัญหาเกี่ยวกับโครงสร้างของข้อมูลที่ถูกจัดเก็บในตารางหนึ่งๆ ตัวอย่างเช่น ถ้าเรามีการเพิ่มเติมสีของรถยนต์ในส่วนประกอบอื่นๆ—เช่น สีของโลโก้รถยนต์ที่ซึ่งจะถูกเพิ่มให้กับรถเพียงบางรุ่นเท่านั้น เราจะต้องทำการปรับเปลี่ยนโครงสร้างข้อมูลของตารางข้อมูลด้วยการเพิ่มแอททริบิวต์ใหม่เข้าไปในตาราง และเราต้องทำการกำหนดให้ค่าของข้อมูลกรุ่นใดๆก็ตามที่ไม่มีสีในส่วนดังกล่าวให้มีค่าเป็น NULL หรือ N/A (not applicable) นอกจากนี้ การสร้างแอททริบิวต์ยังมีปัญหาเพิ่มเติม คือ ถ้าในเอ็นทิตีพนักงานมีการจัดเก็บข้อมูลระดับการศึกษาและใบประกาศนียบัตรต่างๆ และถ้ามีพนักงานจำนวนหนึ่ง (เป็นจำนวนน้อย) ที่มีข้อมูลการศึกษาและประกาศนียบัตรทั้งสิ้น 10 ข้อมูล ในขณะที่พนักงานส่วนใหญ่มีข้อมูลเหล่านี้น้อยกว่า 10 หรืออาจไม่มีเลย ด้วยเหตุนี้เราต้องจำเป็นที่จะต้องทำการกำหนดให้ตารางพนักงานมีแอททริบิวต์ที่ใช้จัดเก็บข้อมูลระดับการศึกษาและใบประกาศนียบัตรต่างๆเป็นจำนวน 10 แอททริบิวต์ แต่ค่าที่ปรากฏในแอททริบิวต์เหล่านี้ของพนักงานส่วนใหญ่จะมีค่าเป็น NULL เนื่องจากพนักงานเหล่านี้ไม่มีข้อมูลการศึกษาและ/หรือประกาศนียบัตร หรืออาจมีเป็นจำนวนน้อย จากการปรากฏขึ้นของค่า NULL เป็นจำนวนมากในแอททริบิวต์ต่างๆ เราจะสังเกตเห็นได้ว่าวิธีการเพิ่มแอททริบิวต์สำหรับจัดเก็บข้อมูลหลายๆค่าข้อมูลในแอททริบิวต์ที่มีลักษณะเป็น multivalued attribute อาจเป็นวิธีการที่ไม่เหมาะสม



รูปที่ 4.4 การเพิ่มแอททริบิวต์จากแอททริบิวต์ที่มีลักษณะเป็น multivalued attribute

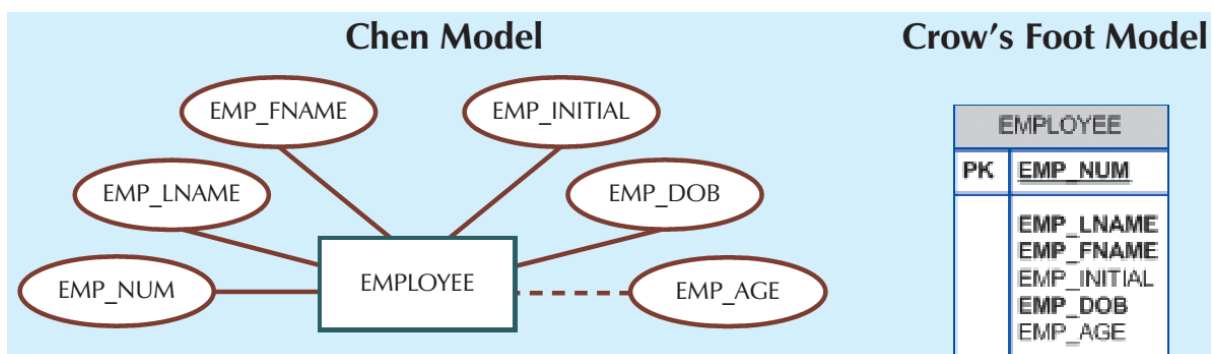
- **การเพิ่มเอ็นทิตีใหม่**—ที่ประกอบไปด้วยส่วนประกอบต่างๆของ multivalued attribute เอ็นทิตีใหม่ที่ถูกรสร้างขึ้นจะมีความสัมพันธ์ในลักษณะ 1:M กับเอ็นทิตีเดิม ตัวอย่างเช่น การเพิ่มตาราง CAR\_COLOR ในรูปที่ 4.5 ที่ซึ่งจะประกอบไปด้วยสามแอท

รีบิว คือ CAR\_VIN, COL\_SECTION, COL\_COLOR ตามลำดับ การเพิ่มตารางจะช่วยให้เราสามารถจัดการกับการเพิ่มเติมของข้อมูลได้โดยง่ายโดยไม่ต้องปรับเปลี่ยนโครงสร้างของการจัดเก็บข้อมูลแต่อย่างใด



รูปที่ 4.5 การเพิ่มเอ็นทิตีสำหรับจัดเก็บข้อมูลในแอทริบิวที่มีลักษณะเป็น multivalued attribute

- **derived attribute**—จะเป็นแอทริบิวที่ข้อมูลในแอทริบิวจะได้มาจากการประมวลผลข้อมูลในแอทริบิวอื่นๆ และจะเป็นแอทริบิวที่ไม่ต้องการพื้นที่ในฐานข้อมูลสำหรับการจัดเก็บข้อมูลแต่อย่างใด เราสามารถได้รับข้อมูลใน derived attribute จากอัลกอริทึมสำหรับประมวลผลข้อมูลหนึ่งๆได้ ตัวอย่างเช่น ข้อมูลในแอทริบิว EMP\_AGE (อายุของพนักงาน) อาจคำนวณได้จากข้อมูลวันที่ ณ ปัจจุบันลบกับข้อมูลในแอทริบิว EMP\_DOB (วันเกิดของพนักงาน) แล้วหารด้วย 365; ข้อมูลราคารวมของการซื้อสินค้าจะคำนวณได้จากผลคูณระหว่างจำนวนสินค้าที่ถูกซื้อกับราคาต่อชิ้นสินค้า เป็นต้น การแสดงถึง derived attribute จะสามารถแสดงได้ดังรูปที่ 4.6 ที่ซึ่งภายใต้กรอบความคิดของ Chen จะประยุกต์ใช้เส้นประเพื่อแสดงถึง derived attribute



รูปที่ 4.6 การแสดงถึงแอทริบิวที่มีลักษณะเป็น derived attribute ใน ERD

การประยุกต์ใช้ derived attribute จะยอมให้เราสามารถเลือกได้ว่าเราจะทำจัดเก็บข้อมูลไว้ในฐานข้อมูลหรือไม่ ที่ซึ่งการจัดเก็บและไม่จัดเก็บจะมีข้อดีและข้อเสียที่แตกต่างกันดังแสดงในรูปที่ 4.7



|                     | DERIVED ATTRIBUTE                                                                                                                        |                                                                                               |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
|                     | STORED                                                                                                                                   | NOT STORED                                                                                    |
| <b>Advantage</b>    | Saves CPU processing cycles<br>Saves data access time<br>Data value is readily available<br>Can be used to keep track of historical data | Saves storage space<br>Computation always yields current value                                |
| <b>Disadvantage</b> | Requires constant maintenance to ensure derived value is current, especially if any values used in the calculation change                | Uses CPU processing cycles<br>Increases data access time<br>Adds coding complexity to queries |

รูปที่ 4.7 ข้อดีและข้อเสียสำหรับการจัดเก็บ derived attribute ในฐานข้อมูล

### 4.1.3 ความสัมพันธ์

ความสัมพันธ์จะแสดงถึงความสัมพันธ์ระหว่างเอนทิตี ที่ซึ่งเอนทิตีที่เกี่ยวข้องกับความสัมพันธจะถูกเรียกว่า **participants** และแต่ละความสัมพันธ์จะมีชื่อความสัมพันธ์ที่ซึ่งบ่งบอกความหมายของความสัมพันธ์ ตัวอย่างเช่น <a STUDENT takes a CLASS>, <a PROFESSOR teaches a CLASS>, <a DEPARTMENT employs a PROFESSOR> เป็นต้น

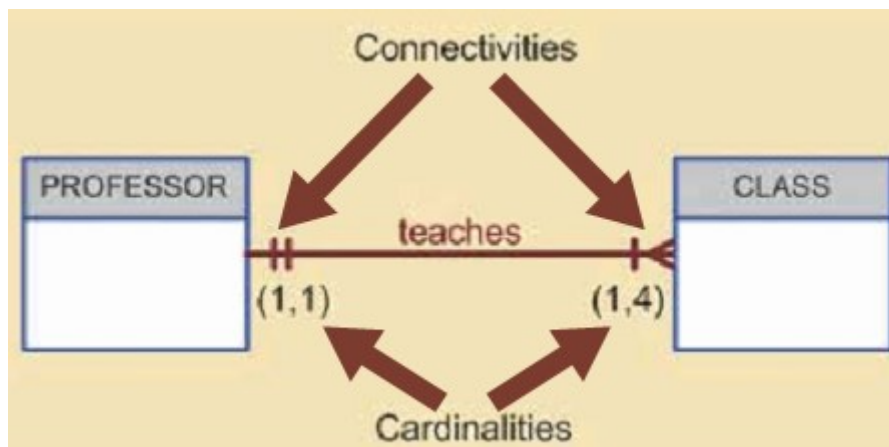
ความสัมพันธ์ระหว่างสองเอนทิตีใดๆจะถูกแสดงในสองทิศทาง เช่น 1) ลูกค้าคนหนึ่งๆสามารถเขียนใบสั่งสินค้าได้หลายใบ และ 2) ใบสั่งสินค้าใบหนึ่งๆจะถูกเขียนโดยลูกค้าคนหนึ่งๆเท่านั้น ความสัมพันธ์ทั้งสองด้านจะทำให้เราทราบถึงรูปแบบของความสัมพันธ์ได้อย่างง่ายดาย (จากตัวอย่างข้างต้น รูปแบบความสัมพันธ์จะเป็นแบบ 1:M) แต่ถ้าสมมติว่าเราทราบถึงความสัมพันธ์เพียงด้านเดียว มักจะเป็นการยากที่จะทราบถึงรูปแบบของความสัมพันธ์ ตัวอย่างเช่น “กองพลหนึ่งๆมักถูกบริหารจัดการโดยนายพลคนหนึ่งๆ” จากความสัมพันธ์ดังกล่าว เราไม่สามารถทราบได้ว่ารูปแบบความสัมพันธ์จะเป็นแบบ 1:1 หรือ 1:M ดังนั้นเราควรที่จะมีการตั้งคำถามที่ว่า “นายพลคนหนึ่งๆสามารถบริหารจัดการได้หลายกองพลหรือไม่ ?” ถ้าคำตอบคือใช่ จะทำให้เราสามารถสรุปได้ว่ารูปแบบความสัมพันธ์ระหว่างนายพลและกองพลจะมีรูปแบบเป็น 1:M แต่ถ้าคำตอบคือไม่ จะทำให้รูปแบบความสัมพันธ์มีรูปแบบเป็น 1:1

### 4.1.4 Connectivity และ Cardinality

อย่างที่เราทราบก่อนหน้านี้แล้วว่าความสัมพันธ์ระหว่างเอนทิตีจะมี 3 รูปแบบคือ one-to-one, one-to-many และ many-to-many ตามลำดับ ด้วยเหตุนี้ จึงเป็นเหตุให้ “**connectivity**” จะเป็นเครื่องหมายที่ซึ่งบ่งบอกถึงรูปแบบความสัมพันธ์ที่เชื่อมโยงระหว่างเอนทิตีต่างๆ (ดังแสดงในรูป 4.8) แต่ในส่วน of “**cardinality**” จะเป็นสิ่งที่บ่งบอกถึงจำนวนแถวข้อมูล (entity occurrence) ต่ำสุดและสูงสุดในตารางข้อมูลหนึ่งๆที่ซึ่งจะมีความสัมพันธ์กับข้อมูลแถวหนึ่งในอีกตารางข้อมูลหนึ่งๆ ตัวอย่างเช่น ERD ในรูป 4.8 จะแสดง cardinality ของความสัมพันธ์ระหว่างเอนทิตี PROFESSOR และ CLASS ที่ซึ่งจะบ่งบอกถึง cardinality (1,4) ในฝั่งของเอนทิตี CLASS จะแสดงว่าอาจารย์คนหนึ่งๆจะต้องสอนอย่างน้อย 1 ชั้นเรียนและ



สามารถสอนได้มากที่สุด 4 ชั้นเรียน แต่ในส่วนของ cardinality (1,1) ในฝั่งของ PROFESSOR จะแสดงว่าแต่ละชั้นเรียนจะถูกสอนโดยอาจารย์เพียงคนเดียวเท่านั้น (หมายเหตุ—cardinality (1, N) ของเอ็นทิตีหนึ่งๆจะหมายถึงการที่ไม่มีข้อจำกัดเกี่ยวกับจำนวนสูงสุดของแถวข้อมูลที่จะมีความสัมพันธ์กับข้อมูลแถวหนึ่งในอีกตารางหนึ่ง ตัวอย่างเช่น ถ้า cardinality (1, N) ปรากฏขึ้นในฝั่งของเอ็นทิตี CLASS จะแสดงว่าอาจารย์คนหนึ่งจะต้องสอนอย่างน้อย 1 ชั้นเรียนและสามารถสอนได้ไม่จำกัดชั้นเรียน)



รูปที่ 4.8 ตัวอย่าง connectivity และ cardinality ใน ERD

#### 4.1.5 ระดับความเข้มของความสัมพันธ์

แนวความคิดของระดับความเข้มของความสัมพันธ์ (relationship strength) จะขึ้นกับการนิยามและการประยุกต์ใช้ primary key ของตารางหนึ่งๆเพื่อเป็น foreign key ของอีกตารางหนึ่งๆอย่างไร ในหลายๆครั้งที่เราอาจประยุกต์ใช้ foreign key เพื่อเป็นส่วนประกอบของ primary key ของตารางข้อมูลหนึ่งๆ จากการดำเนินการดังกล่าว เราจะสามารถแบ่งประเภทของระดับความเข้มของความสัมพันธ์ได้เป็น 2 ประเภทดังนี้

- **weak relationship (non-identifying relationship)**—จะเป็นความสัมพันธ์ที่ซึ่ง primary key ในเอ็นทิตี A ใดๆจะทำหน้าที่เป็น foreign key ในเอ็นทิตี B แต่ไม่ได้ทำหน้าที่เป็นหรือเป็นส่วนประกอบของ primary key ในเอ็นทิตี B—ตัวอย่างเช่น ความสัมพันธ์ดังรูป 4.9 ที่จะแสดงถึงความสัมพันธ์ระหว่างเอ็นทิตี COURSE และ CLASS ที่ซึ่งเอ็นทิตี COURSE จะกำหนดให้แอทริบิวต์ CRS\_CODE ทำหน้าที่เป็น primary key และเอ็นทิตี CLASS จะกำหนดให้แอทริบิวต์ CRS\_CODE ทำหน้าที่เป็น foreign key เพื่อเชื่อมโยงความสัมพันธ์ระหว่างสองเอ็นทิตีนี้ แต่อย่างไรก็ตาม แอทริบิวต์ CRS\_CODE ในเอ็นทิตี CLASS ไม่ได้ทำหน้าที่เป็นหรือเป็นส่วนประกอบหนึ่งของ primary key ในเอ็นทิตี CLASS ดังนั้น เราจะสามารถสรุปได้ว่าเอ็นทิตี COURSE และเอ็นทิตี CLASS จะมีความสัมพันธ์ในลักษณะ weak

relationship โดยจากรูป 4.9 เราจะสังเกตได้ว่า ERD ที่ถูกเขียนอยู่ในรูปแบบของ Crow's Foot จะประยุกต์ใช้เส้นประเพื่อแสดงถึง weak relationship

- **strong relationship (identifying relationship)**—จะเป็นความสัมพันธ์ที่ซึ่ง primary key ในเอนทิตี A ใดๆจะทำหน้าที่เป็น foreign key ในเอนทิตี B และยังทำหน้าที่เป็นหรือเป็นส่วนประกอบหนึ่งของ primary key ในเอนทิตี B ด้วยเช่นกัน ตัวอย่างเช่น ความสัมพันธ์ดังรูป 4.10 ที่จะแสดงถึงความสัมพันธ์ระหว่างเอนทิตี COURSE และ CLASS ที่ซึ่งจะเป็นความสัมพันธ์แบบ strong relationship เนื่องจากแอททริบิวต์ CRS\_CODE ที่ทำหน้าที่เป็น primary key ในเอนทิตี COURSE จะทำหน้าที่เป็น foreign key และเป็นส่วนประกอบหนึ่งของ primary key ในเอนทิตี CLASS ด้วยเช่นกัน โดยจากรูป 4.10 เราจะสังเกตได้ว่า ERD ที่ถูกเขียนในรูปแบบของ Crow's Foot จะประยุกต์ใช้เส้นตรง (หนา) เพื่อแสดงถึง strong relationship

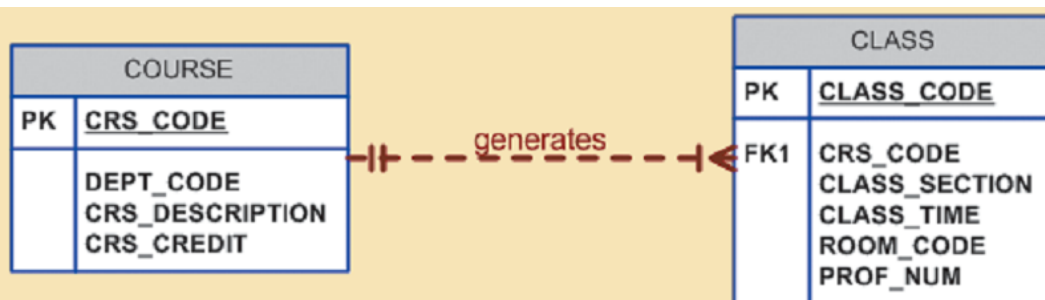


Table name: COURSE

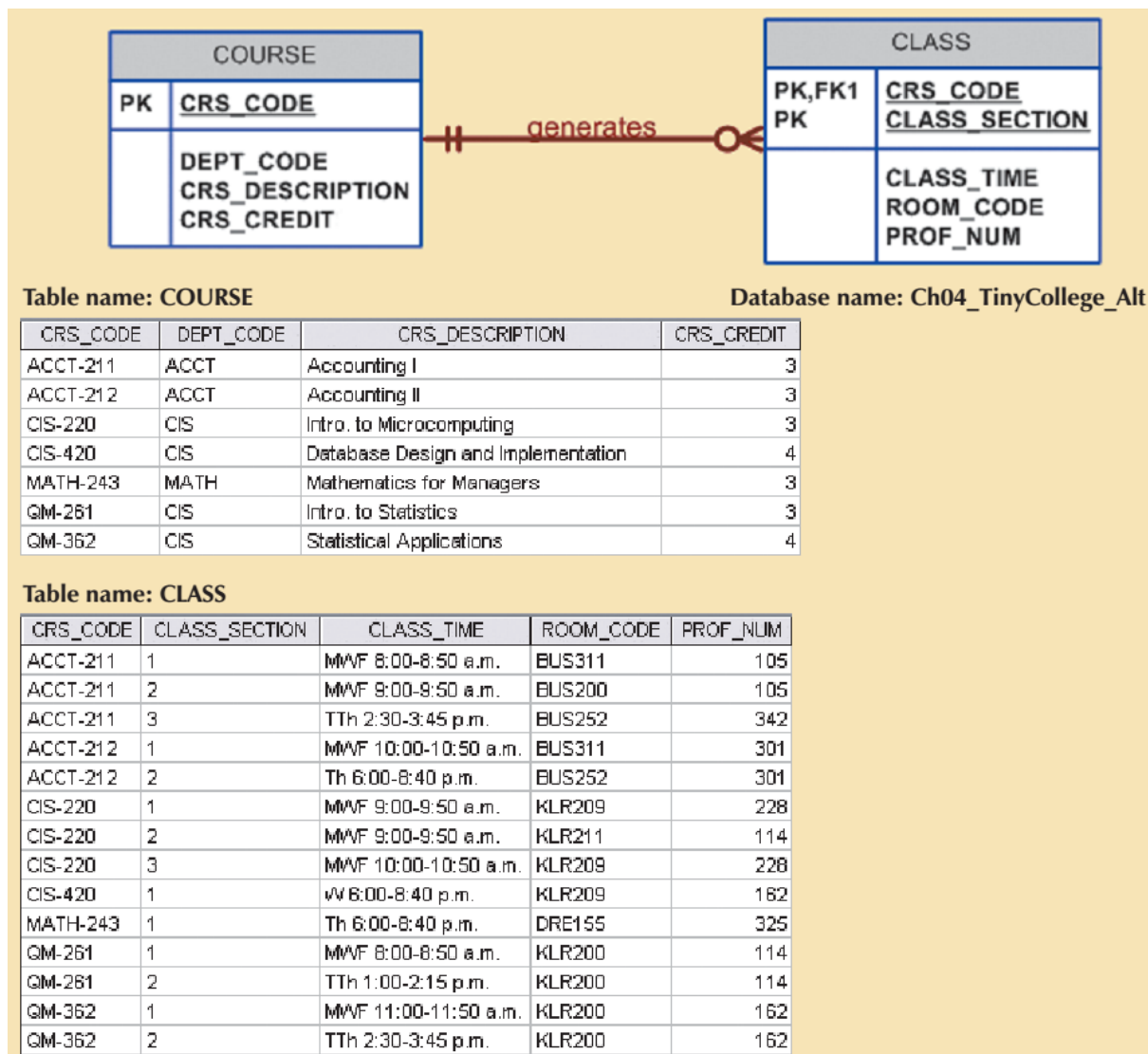
Database name: Ch04\_TinyCollege

| CRS_CODE | DEPT_CODE | CRS_DESCRIPTION                    | CRS_CREDIT |
|----------|-----------|------------------------------------|------------|
| ACCT-211 | ACCT      | Accounting I                       | 3          |
| ACCT-212 | ACCT      | Accounting II                      | 3          |
| CIS-220  | CIS       | Intro. to Microcomputing           | 3          |
| CIS-420  | CIS       | Database Design and Implementation | 4          |
| MATH-243 | MATH      | Mathematics for Managers           | 3          |
| QM-261   | CIS       | Intro. to Statistics               | 3          |
| QM-362   | CIS       | Statistical Applications           | 4          |

Table name: CLASS

| CLASS_CODE | CRS_CODE | CLASS_SECTION | CLASS_TIME             | ROOM_CODE | PROF_NUM |
|------------|----------|---------------|------------------------|-----------|----------|
| 10012      | ACCT-211 | 1             | M/W/F 8:00-8:50 a.m.   | BUS311    | 105      |
| 10013      | ACCT-211 | 2             | M/W/F 9:00-9:50 a.m.   | BUS200    | 105      |
| 10014      | ACCT-211 | 3             | TTh 2:30-3:45 p.m.     | BUS252    | 342      |
| 10015      | ACCT-212 | 1             | M/W/F 10:00-10:50 a.m. | BUS311    | 301      |
| 10016      | ACCT-212 | 2             | Th 6:00-8:40 p.m.      | BUS252    | 301      |
| 10017      | CIS-220  | 1             | M/W/F 9:00-9:50 a.m.   | KLR209    | 228      |
| 10018      | CIS-220  | 2             | M/W/F 9:00-9:50 a.m.   | KLR211    | 114      |
| 10019      | CIS-220  | 3             | M/W/F 10:00-10:50 a.m. | KLR209    | 228      |
| 10020      | CIS-420  | 1             | W 6:00-8:40 p.m.       | KLR209    | 162      |
| 10021      | QM-261   | 1             | M/W/F 8:00-8:50 a.m.   | KLR200    | 114      |
| 10022      | QM-261   | 2             | TTh 1:00-2:15 p.m.     | KLR200    | 114      |
| 10023      | QM-362   | 1             | M/W/F 11:00-11:50 a.m. | KLR200    | 162      |
| 10024      | QM-362   | 2             | TTh 2:30-3:45 p.m.     | KLR200    | 162      |
| 10025      | MATH-243 | 1             | Th 6:00-8:40 p.m.      | DRE155    | 325      |

รูป 4.9 ตัวอย่างความสัมพันธ์แบบ weak relationship



รูป 4.10 ตัวอย่างความสัมพันธ์แบบ strong relationship

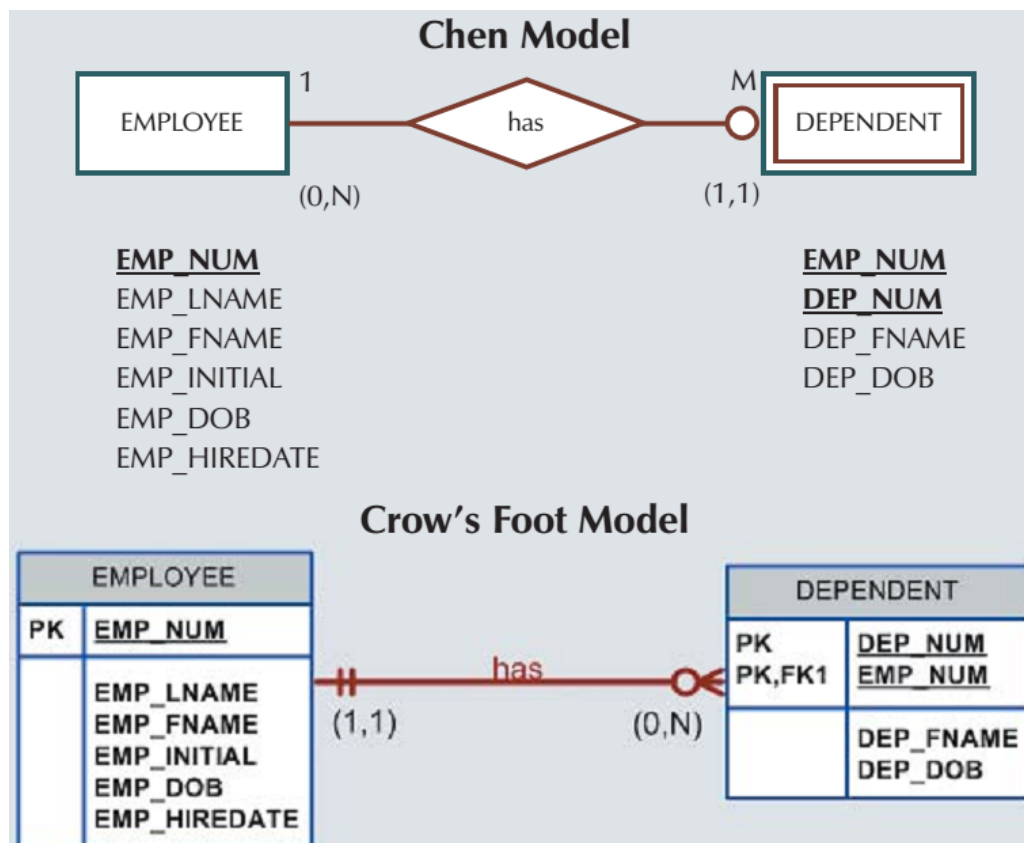
ในการที่จะสร้างและจัดข้อมูลในตารางข้อมูลที่มีความสัมพันธ์แบบ strong relationship เราจะต้องทำการพิจารณาถึง “ลำดับ” ของการสร้างและการจัดเก็บข้อมูล จากตัวอย่างในรูป 4.10 เราจะสังเกตได้ว่าเมื่อมี COURSE หนึ่งๆจะก่อให้เกิด CLASS หนึ่งๆ ดังนั้น เราควรที่จะต้องทำการสร้างตาราง COURSE ก่อน จากนั้นจึงทำการสร้างตาราง CLASS การดำเนินการดังกล่าวจะทำให้ไม่เกิดเหตุการณ์ที่ว่า foreign key ในตาราง CLASS จะทำการอ้างอิงไปถึงตารางที่ไม่มีตัวตน ในส่วนของการจัดเก็บข้อมูลก็เช่นกัน เราควรที่จะต้องจัดเก็บข้อมูลในเอ็นทิตี COURSE (ในฝั่งของ 1) ก่อนเอ็นทิตี CLASS (ในฝั่งของ M) เพื่อหลีกเลี่ยงปัญหาข้อผิดพลาดเกี่ยวกับ referential integrity

#### 4.1.6 weak entity

Weak entity จะเป็นเอ็นทิตีที่ซึ่ง 1) จะไม่สามารถเกิดขึ้นได้ถ้าไม่มีการเชื่อมโยงความสัมพันธ์กับเอ็นทิตีอื่น (เรียกว่า existence-dependence) และ 2) primary key ของเอ็นทิตีจะต้องมีแอทริบิวที่เป็น

foreign key ในเอ็นทิตีเป็นส่วนประกอบ (หมายเหตุ—primary key อาจกำหนดจากกลุ่มของ foreign key ในเอ็นทิตี หรืออาจกำหนดจาก foreign key เพียงบางส่วนผนวกกับแอทริบิวต์อื่นๆเพิ่มเติม) ตัวอย่างเช่น บริษัทประกันชีวิตที่มีนโยบายที่จะทำประกันให้แก่ญาติของพนักงาน ที่ซึ่งพนักงานคนหนึ่งๆอาจจะมีญาติหรือไม่ก็ได้ แต่ญาติคนหนึ่งๆจะต้องมีความเกี่ยวข้องกับพนักงานคนหนึ่งๆ ดังนั้น เราสามารถสรุปได้ว่าบุคคลหนึ่งๆจะไม่สามารถทำประกันตามนโยบายของบริษัทได้ถ้าไม่ได้เป็นญาติกับพนักงานคนหนึ่งๆ

รูป 4.11 จะแสดงความสัมพันธ์ระหว่างเอ็นทิตีพนักงาน (EMPLOYEE) และญาติของพนักงาน (DEPENDENT) โดยที่ ERD ที่ถูกเขียนในรูปแบบของ Chen จะใช้กรอบสี่เหลี่ยมสองชั้นเพื่อแสดงถึง weak entity และ ในรูป 4.12 จะแสดงถึงการดำเนินการสร้างความสัมพันธ์ระหว่างเอ็นทิตี DEPENDENT ที่เป็น weak entity กับเอ็นทิตี EMPLOYEE ที่ซึ่งในตารางข้อมูล DEPENDENT จะมี primary key ที่เป็น composite key ที่เกิดจากการรวมกันระหว่างแอทริบิว EMP\_NUM (foreign key) และ DEP\_NUM



รูปที่ 4.11 ตัวอย่าง weak entity ใน ERD

| Table name: EMPLOYEE |            |           |             |           | Database name: Ch04_ShortCo |
|----------------------|------------|-----------|-------------|-----------|-----------------------------|
| EMP_NUM              | EMP_LNAME  | EMP_FNAME | EMP_INITIAL | EMP_DOB   | EMP_HIREDATE                |
| 1001                 | Callifante | Jeanine   | J           | 12-Mar-64 | 25-May-97                   |
| 1002                 | Smithson   | William   | K           | 23-Nov-70 | 26-May-97                   |
| 1003                 | Washington | Herman    | H           | 15-Aug-68 | 28-May-97                   |
| 1004                 | Chen       | Lydia     | B           | 23-Mar-74 | 15-Oct-98                   |
| 1005                 | Johnson    | Melanie   |             | 28-Sep-66 | 20-Dec-98                   |
| 1006                 | Ortega     | Jorge     | G           | 12-Jul-79 | 05-Jan-02                   |
| 1007                 | O'Donnell  | Peter     | D           | 10-Jun-71 | 23-Jun-02                   |
| 1008                 | Brzenski   | Barbara   | A           | 12-Feb-70 | 01-Nov-03                   |

| Table name: DEPENDENT |         |           |           |
|-----------------------|---------|-----------|-----------|
| EMP_NUM               | DEP_NUM | DEP_FNAME | DEP_DOB   |
| 1001                  | 1       | Annelise  | 05-Dec-97 |
| 1001                  | 2       | Jorge     | 30-Sep-02 |
| 1003                  | 1       | Suzanne   | 25-Jan-04 |
| 1006                  | 1       | Carlos    | 25-May-01 |
| 1008                  | 1       | Michael   | 19-Feb-95 |
| 1008                  | 2       | George    | 27-Jun-98 |
| 1008                  | 3       | Katherine | 18-Aug-03 |

รูปที่ 4.12 ตัวอย่าง weak entity ที่มีความสัมพันธ์แบบ strong relationship

#### 4.1.7 การมีส่วนร่วมกับความสัมพันธ์

อย่างที่เราราบติว่า ความสัมพันธ์ระหว่างเอนิตี้หนึ่งๆจะเป็นแบบสองทิศทาง ดังนั้น เราจึงควรที่จะทำการกำหนด connectivity เพื่อที่จะทราบถึงรูปแบบความสัมพันธ์ และทำการกำหนด cardinality เพื่อทราบถึงจำนวนน้อยสุดและมากสุดของแถวข้อมูลในเอนิตี้หนึ่งๆที่ซึ่งจะมีความสัมพันธ์กับข้อมูลแถวหนึ่งในอีกเอนิตี้หนึ่งๆ นอกจากนั้น เราควรที่จะต้องพิจารณาถึงการมีส่วนร่วมกับความสัมพันธ์ (relationship participation) ของเอนิตี้ที่ซึ่งจะสามารถแบ่งได้เป็น 2 ประเภท ดังนี้

- optional participation—จะเป็นความสัมพันธ์ของเอนิตี้ที่ซึ่งการปรากฏขึ้นของข้อมูลครั้งหนึ่งในเอนิตี้หนึ่งๆจะไม่ต้องการให้มีการปรากฏขึ้นของข้อมูลในอีกเอนิตี้หนึ่ง ตัวอย่างเช่น “รายวิชา (COURSE) ก่อให้เกิดชั้นเรียน (CLASS)” แต่เราจะสังเกตความจริงได้ว่า “รายวิชาหนึ่งอาจมีหรือไม่มีชั้นเรียนที่เกี่ยวข้องกันก็ได้” ดังนั้น เราสามารถสรุปได้ว่าการปรากฏขึ้นของข้อมูลหนึ่งๆ (ข้อมูลแถวหนึ่งๆ) ในตาราง COURSE จะไม่ต้องการให้มีการปรากฏขึ้นของข้อมูลในตาราง CLASS ที่ซึ่งจะทำให้เอนิตี้ CLASS จะถูกพิจารณาเป็น optional participation ของความสัมพันธ์กับเอนิตี้ COURSE (หมายเหตุ—การแสดงถึง optional participation ในรูป 4.10 จะแสดงในรูปแบบของ Crow’s Foot ที่ซึ่งจะประยุกต์ใช้วงกลมในฝั่งของเอนิตี้ที่มีลักษณะเป็น optional participation)
- mandatory participation—จะเป็นความสัมพันธ์ของเอนิตี้ที่ซึ่งการปรากฏขึ้นของข้อมูลครั้งหนึ่งในเอนิตี้หนึ่งๆจะต้องการให้มีการปรากฏขึ้นของข้อมูลในอีกเอนิตี้หนึ่ง โดยเมื่อเราพิจารณาที่เอนิตี้หนึ่งๆใน ERD ใดๆแล้วไม่พบสัญลักษณ์วงกลม เราจะสามารถสันนิษฐานได้ว่าเอนิตี้ที่พิจารณาจะมี



ส่วนร่วมกับความสัมพันธ์แบบ mandatory participation ที่ซึ่งจะการใช้การขีดคร่อมเส้นเชื่อมโยงความสัมพันธ์หนึ่งซิด ดังแสดงในรูป 4.11 ภายใต้รูปแบบของ Crow's Foot

เนื่องด้วยรูปแบบความสัมพันธ์มีหลากหลายประกอบกับประเภทของการมีส่วนร่วมกับความสัมพันธ์อีกสองลักษณะ เราจะสามารถแสดง connectivity และ participation ได้หลายลักษณะดังแสดงในรูป 4.13

| CROW'S FOOT SYMBOL                                                                | CARDINALITY | COMMENT                                |
|-----------------------------------------------------------------------------------|-------------|----------------------------------------|
|  | (0,N)       | Zero or many. Many side is optional.   |
|  | (1,N)       | One or many. Many side is mandatory.   |
|  | (1,1)       | One and only one. 1 side is mandatory. |
|  | (0,1)       | Zero or one. 1 side is optional.       |

รูปที่ 4.13 สัญลักษณ์เกี่ยวกับ cardinality ในรูปแบบของ Crow's Foot

เนื่องจากการกำหนดความมีส่วนร่วมกับความสัมพันธ์เป็นสิ่งสำคัญในกระบวนการออกแบบฐานข้อมูล ดังนั้น ลองพิจารณาตัวอย่างของมหาวิทยาลัยที่ทำการจ้างอาจารย์ที่จะทำงานวิจัยเพียงอย่างเดียวโดยไม่ต้องสอนหนังสือแต่อย่างใด ถ้าเราพิจารณาที่ความสัมพันธ์ “อาจารย์ (PROFESSOR) สอนในชั้นเรียนต่างๆ (CLASS)” แต่อาจมีอาจารย์บางคนที่ไม่จำเป็นต้องสอนหนังสือในชั้นเรียนต่างๆด้วยเช่นกัน ด้วยเหตุนี้ PROFESSOR จะมีส่วนร่วมกับความสัมพันธ์ในแบบ mandatory participation กับเอ็นทิตี CLASS ดังนั้น ERD ในรูป 4.14 จะแสดง cardinality ของ CLASS ที่สัมพันธ์กับ PROFESSOR โดยจะมี cardinality เป็น (0, 3) ที่จะแสดงให้เห็นว่าอาจารย์คนหนึ่งๆอาจไม่จำเป็นต้องสอนในชั้นเรียน แต่ถ้าจำเป็นต้องสอนจะสามารถสอนได้ไม่เกิน 3 ชั้นเรียน แต่ในส่วนของเอ็นทิตี CLASS จะเกี่ยวเนื่องกับ PROFESSOR เพียงคนเดียวเท่านั้น (cardinality เป็น 1,1)

ความไม่เข้าใจความแตกต่างระหว่าง mandatory และ optional participation สามารถก่อให้เกิดปัญหาตามมา ตัวอย่างเช่น มหาวิทยาลัยจะมีหลายวิชาที่เปิดสอน แต่ละรายวิชาจะสามารถมีได้หลายชั้นเรียน ดังนั้น เราสามารถทำการวิเคราะห์ความสัมพันธ์ของเอ็นทิตีรายวิชา (COURSE) และชั้นเรียน (CLASS) ได้เป็น “รายวิชาก่อให้เกิดชั้นเรียน” และ “จะไม่มีชั้นเรียนใดเลยที่ไม่เกี่ยวเนื่องกับรายวิชา” จากความสัมพันธ์ที่วิเคราะห์ได้ เราจะสามารถสรุปได้ว่าเอ็นทิตี COURSE จะมีความเกี่ยวเนื่องกับความสัมพันธ์กับเอ็นทิตี CLASS แบบ mandatory participation แต่เราสามารถแสดงถึงการมีส่วนร่วมในความสัมพันธ์ของเอ็นทิตี CLASS ได้ 2 กรณีดังนี้ (ดังแสดงในรูป 4.15 และ 4.16)

- การมีส่วนร่วมกับความสัมพันธ์แบบ optional ของเอ็นทิตี CLASS จะทำให้มีความเป็นไปได้ที่ภาควิชาหนึ่งจะทำการสร้างเอ็นทิตี COURSE จากนั้นจึงทำการสร้างเอ็นทิตี CLASS ซึ่งการ

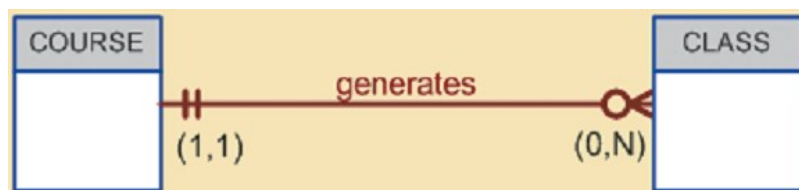
ดำเนินการดังกล่าวจะสอดคล้องกับความจริงที่อาจจะมีบาง COURSE ที่ไม่มีการเปิดชั้นเรียนเพื่อสอน เช่น บาง COURSE อาจถูกสอนสองปีครั้ง หรือปีละครั้ง เป็นต้น

- การมีส่วนร่วมกับความสัมพันธ์แบบ mandatory ของเอนทิตี CLASS จะถูกกำหนดจากความหมายของความสัมพันธ์ “แต่ละ COURSE ก่อให้เกิด CLASS” โดยแต่ละ COURSE จำเป็นที่จะต้องก่อให้เกิด CLASS อย่างน้อยหนึ่ง CLASS ดังนั้นเมื่อมี COURSE หนึ่งๆปรากฏจะต้องมี CLASS หนึ่งๆปรากฏด้วยเสมอ ซึ่งอาจทำให้ลักษณะการดำเนินงานหรือเงื่อนไขในการดำเนินงานเปลี่ยนแปลงไป

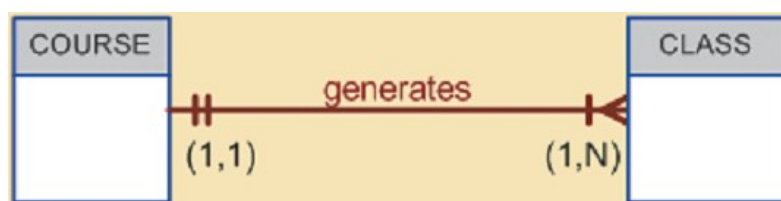
จากการกำหนดเกี่ยวกับการมีส่วนร่วมกับความความสัมพันธ์ทั้งสองกรณีข้างต้น เราจะสังเกตได้ว่าการกำหนดให้เอนทิตี CLASS มีส่วนร่วมกับความสัมพันธ์แบบ mandatory participation จะเป็นสิ่งที่ไม่เหมาะสม เนื่องจากจะทำให้รูปแบบความสัมพันธ์ไม่ยืดหยุ่นและอาจจะไม่สอดคล้องกับการดำเนินงานจริง ด้วยเหตุนี้ เราควรที่จะพิจารณาถึงการมีส่วนร่วมในความสัมพันธ์ของเอนทิตีต่างๆอย่างถี่ถ้วน



รูปที่ 4.14 การมีส่วนร่วมกับความสัมพันธ์แบบ optional participation



รูปที่ 4.15 เอนทิตี CLASS มีส่วนร่วมกับความสัมพันธ์แบบ optional



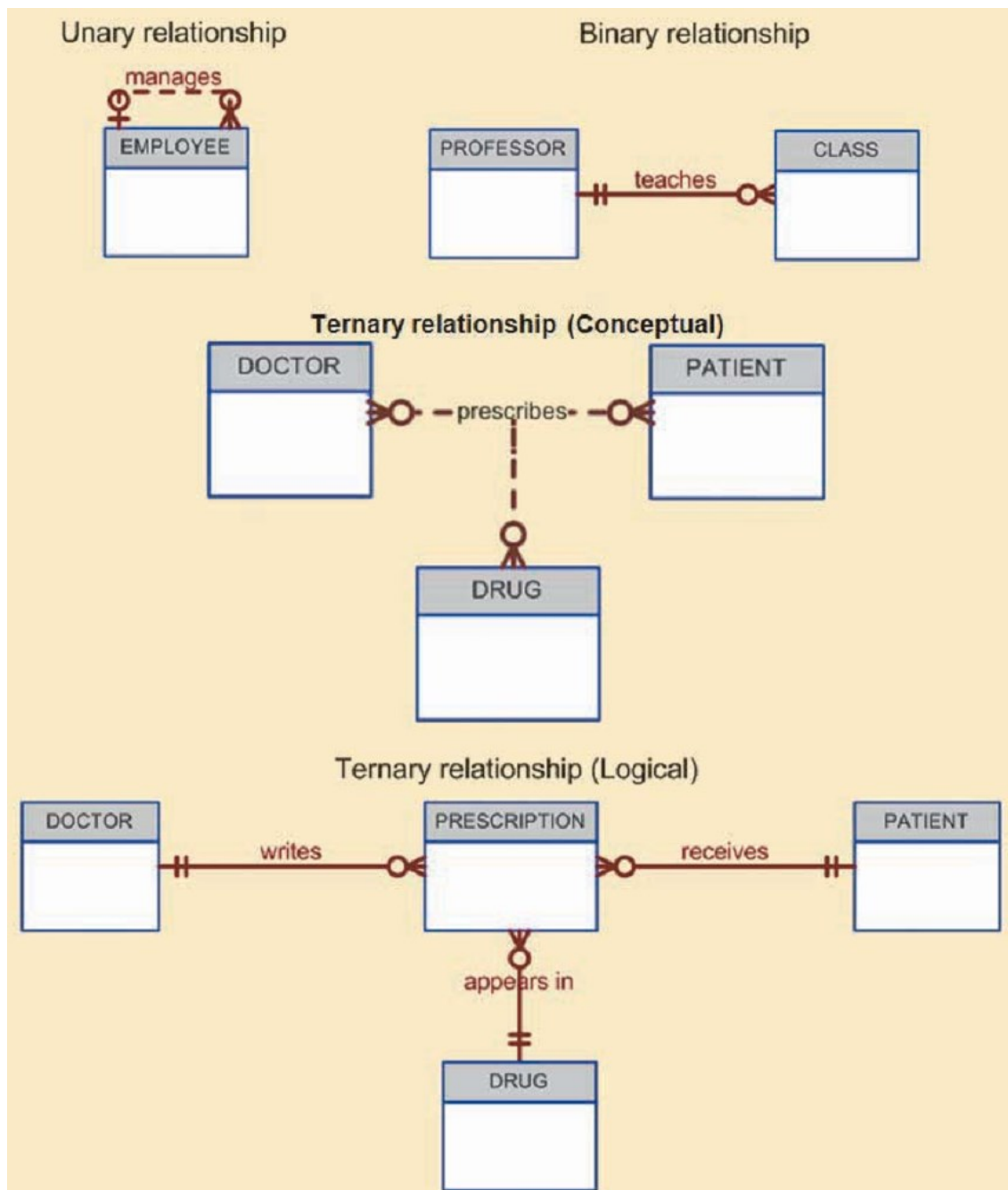
รูปที่ 4.16 เอนทิตี CLASS มีส่วนร่วมกับความสัมพันธ์แบบ mandatory

#### 4.1.8 ระดับความสัมพันธ์

ระดับความสัมพันธ์ (relationship degree) จะแสดงถึงจำนวนเอนทิตีที่มีส่วนร่วมกับความสัมพันธ์หนึ่งๆที่ซึ่งจะมีความสัมพันธ์ทั้งสิ้น 3 ระดับดังแสดงในรูป 4.17

- Unary relationship จะเป็นความสัมพันธ์ระหว่างแถวข้อมูลในเอนทิตีหนึ่งๆที่มีความสัมพันธ์กันเองภายในเอนทิตี ตัวอย่างเช่น “พนักงาน (EMPLOYEE) หนึ่งๆจะเป็นหัวหน้าของพนักงานหลายคน

(EMPLOYEEs)” จะเป็นความสัมพันธ์ภายในเอนทิตีหนึ่งๆ หรือที่เรียกอีกอย่างหนึ่งว่า recursive relationship ที่ซึ่งจะเป็นความสัมพันธ์ระหว่างพนักงานกับพนักงานภายในเอนทิตี EMPLOYEE



รูปที่ 4.17 ระดับความสัมพันธ์ทั้ง 3 ระดับ

- Binary relationship จะเป็นความสัมพันธ์ระหว่าง 2 เอนทิตีใดๆ เช่น ความสัมพันธ์ระหว่างเอนทิตี PROFESSOR และ เอนทิตี CLASS ที่ซึ่ง “อาจารย์คนหนึ่งๆจะต้องสอนในชั้นเรียนหนึ่งๆหรือมากกว่านั้น”

- Ternary relationship จะเป็นความสัมพันธ์ระหว่าง 3 เอนทิตีที่แตกต่างกัน ตัวอย่างเช่น หมอ (DOCTOR) คนหนึ่งๆจะสามารถเขียนใบสั่งยา (PRESCRIPTION) ได้หลายใบ คนไข้ (PATIENT) คนหนึ่งๆอาจได้รับใบสั่งยาหลายใบ และใบสั่งยาหนึ่งๆอาจมียา (DRUG) หลายชนิด ตามลำดับ จากความสัมพันธ์ดังกล่าวจะเป็นความสัมพันธ์ระหว่างเอนทิตี DOCTOR, PATIENT และ DRUG ที่มีความสัมพันธ์กันผ่านใบสั่งยา ด้วยเหตุนี้เราจึงเรียกความสัมพันธ์ของทั้ง 3 เอนทิตีนี้ว่า ternary relationship แต่อย่างไรก็ตาม การที่เอนทิตีต่างๆมีความสัมพันธ์กับใบสั่งยาจะทำให้ใบสั่งยามีข้อมูลต่างๆคือ หมอผู้ออกใบสั่งยา คนไข้ผู้รับยา และตัวยาที่ถูกสั่ง เป็นต้น ด้วยเหตุนี้เราจึงจำเป็นที่จะต้องสร้างเอนทิตีใบสั่งยา (PRESCRIPTION) เพื่อจัดเก็บรายละเอียดต่างๆของใบสั่งยา (ดังแสดงในรูป 4.17)

#### 4.1.9 recursive relationship

Recursive relation จะเป็นความสัมพันธ์ที่เกิดขึ้นระหว่างแถวข้อมูลในเอนทิตีหนึ่งๆ ตัวอย่างเช่น

- Unary relation ที่มีความสัมพันธ์ในรูปแบบ 1:M สามารถแสดงได้โดย “พนักงานคนหนึ่งๆจะสามารถจัดการ/เป็นหัวหน้าของพนักงานหลายๆคน และพนักงานแต่ละคนจะเป็นลูกน้องของพนักงานคนหนึ่งๆ”
- Unary relation ที่มีความสัมพันธ์ในรูปแบบ 1:1 สามารถแสดงได้โดย “พนักงานคนหนึ่งๆอาจแต่งงานกับพนักงานอีกคนหนึ่ง”
- Unary relation ที่มีความสัมพันธ์ในรูปแบบ M:N อาจสามารถแสดงได้โดย “รายวิชาหนึ่งๆอาจเป็นบูรพวิชาของรายวิชาต่างๆ และรายวิชาต่างๆอาจมีหลายบูรพวิชา”

จากตัวอย่างทั้ง 3 ข้างต้น เราสามารถพิจารณาถึงรูปแบบความสัมพันธ์ของ unary relation (ดังแสดงในรูป 4.18) ที่จะมีรายละเอียดการดำเนินการดังต่อไปนี้



รูปที่ 4.18 รูปแบบความสัมพันธ์ของ unary relationship

- ความสัมพันธ์แบบ 1:1 จะสามารถดำเนินการสร้างตารางข้อมูลเพื่อใช้ในการเก็บข้อมูลดังแสดงในรูป 4.19 ที่ซึ่งเราจะสามารถทราบถึงข้อมูลพนักงานที่แต่งงานกัน เช่น James Ramirez แต่งงานกับ Louise Ramirez และสามารถข้อมูลอีกนัยหนึ่งได้ว่าใครแต่งงานกับ James Ramirez เป็นต้น
- ความสัมพันธ์แบบ 1:M จะสามารถดำเนินการสร้างตารางข้อมูลเพื่อใช้ในการเก็บข้อมูลดังแสดงในรูป 4.20 ที่ซึ่งจะแสดงถึงการที่อุปกรณ์หนึ่งๆจะสามารถมีส่วนประกอบย่อยเป็นอุปกรณ์ต่างๆได้หลายชิ้น (อุปกรณ์หนึ่งๆจะเกิดจากการประกอบกันของอุปกรณ์หลายๆชิ้น) ตัวอย่างเช่น อุปกรณ์รหัส C-130 (rotor assembly) จะประกอบไปด้วยหลายส่วนประกอบย่อย แต่แต่ละส่วนประกอบหนึ่งๆจะสามารถถูกใช้เพื่อประกอบเป็น rotor assembly หนึ่งๆ แต่ถ้าอุปกรณ์หนึ่งๆสามารถเป็นส่วนประกอบย่อยของอุปกรณ์หลายชิ้นจะทำให้รูปแบบความสัมพันธ์เปลี่ยนแปลงจาก 1:M ไปเป็น M:N ที่ซึ่งเราจะต้องทำการสร้างตารางข้อมูลใหม่เพื่อใช้ในการจัดเก็บข้อมูลที่มีความสัมพันธ์กัน (ดังแสดงในรูป 4.21)

| Database name: CH04_PartCo |           |           |            |
|----------------------------|-----------|-----------|------------|
| Table name: EMPLOYEE_V1    |           |           |            |
| EMP_NUM                    | EMP_LNAME | EMP_FNAME | EMP_SPOUSE |
| 345                        | Ramirez   | James     | 347        |
| 346                        | Jones     | Anne      | 349        |
| 347                        | Ramirez   | Louise    | 345        |
| 348                        | Delaney   | Robert    |            |
| 349                        | Shapiro   | Anton     | 346        |

รูปที่ 4.19 recursive relationship แบบ 1:1 ของความสัมพันธ์ EMPLOYEE แต่งงานกับ EMPLOYEE

| Table name: PART_V1 |                             |               |                   | Database name: CH04_PartCo |
|---------------------|-----------------------------|---------------|-------------------|----------------------------|
| PART_CODE           | PART_DESCRIPTION            | PART_IN_STOCK | PART_UNITS_NEEDED | PART_OF_PART               |
| AA21-6              | 2.5 cm. washer, 1.0 mm. rim | 432           | 4                 | C-130                      |
| AB-121              | Cotter pin, copper          | 1034          | 2                 | C-130                      |
| C-130               | Rotor assembly              | 36            |                   |                            |
| E128                | 2.5 cm. steel shank         | 128           | 1                 | C-130                      |
| X10                 | 10.25 cm. rotor blade       | 345           | 4                 | C-130                      |
| X34AW               | 2.5 cm. hex nut             | 879           | 2                 | C-130                      |

รูปที่ 4.20 recursive relationship แบบ 1:M ของความสัมพันธ์ PART เป็นส่วนประกอบย่อยของ PART

| Table name: COMPONENT |           |                   | Database name: Ch04_PartCo |
|-----------------------|-----------|-------------------|----------------------------|
| COMP_CODE             | PART_CODE | COMP_PARTS_NEEDED |                            |
| C-130                 | AA21-6    | 4                 |                            |
| C-130                 | AB-121    | 2                 |                            |
| C-130                 | E129      | 1                 |                            |
| C-131A2               | E129      | 1                 |                            |
| C-130                 | X10       | 4                 |                            |
| C-131A2               | X10       | 1                 |                            |
| C-130                 | X34AW     | 2                 |                            |
| C-131A2               | X34AW     | 2                 |                            |

| Table name: PART |                             |               |
|------------------|-----------------------------|---------------|
| PART_CODE        | PART_DESCRIPTION            | PART_IN_STOCK |
| AA21-6           | 2.5 cm. washer, 1.0 mm. rim | 432           |
| AB-121           | Cotter pin, copper          | 1034          |
| C-130            | Rotor assembly              | 36            |
| E129             | 2.5 cm. steel shank         | 128           |
| X10              | 10.25 cm. rotor blade       | 345           |
| X34AW            | 2.5 cm. hex nut             | 879           |

รูปที่ 4.21 recursive relationship แบบ M:M ของความสัมพันธ์ PART เป็นส่วนประกอบย่อยของ PART

- ความสัมพันธ์แบบ M:N จะสามารถดำเนินการสร้างตารางข้อมูลเพื่อใช้ในการเก็บข้อมูลดังแสดงในรูป 4.22 ที่ซึ่งจะแสดงความสัมพันธ์ “รายวิชาหนึ่งๆจะสามารถเป็นบุรพวิชาของหลายรายวิชา และรายวิชาหนึ่งๆสามารถมีได้หลายบุรพวิชา” ที่ซึ่งจะต้องทำการสร้างตารางข้อมูลใหม่เพื่อใช้ในการจัดเก็บข้อมูลความสัมพันธ์ระหว่างรายวิชาหนึ่งๆและรายวิชาที่เป็นบุรพวิชาทั้งหมด

| Table name: COURSE |           |                                    |            | Database name: Ch04_TinyCollege |
|--------------------|-----------|------------------------------------|------------|---------------------------------|
| CRS_CODE           | DEPT_CODE | CRS_DESCRIPTION                    | CRS_CREDIT |                                 |
| ACCT-211           | ACCT      | Accounting I                       | 3          |                                 |
| ACCT-212           | ACCT      | Accounting II                      | 3          |                                 |
| CIS-220            | CIS       | Intro. to Microcomputing           | 3          |                                 |
| CIS-420            | CIS       | Database Design and Implementation | 4          |                                 |
| MATH-243           | MATH      | Mathematics for Managers           | 3          |                                 |
| QM-261             | CIS       | Intro. to Statistics               | 3          |                                 |
| QM-362             | CIS       | Statistical Applications           | 4          |                                 |

| Table name: PREREQ |          |
|--------------------|----------|
| CRS_CODE           | PRE_TAKE |
| CIS-420            | CIS-220  |
| QM-261             | MATH-243 |
| QM-362             | MATH-243 |
| QM-362             | QM-261   |

รูปที่ 4.22 recursive relationship แบบ M:M ของความสัมพันธ์ COURSE จะเป็นบุรพวิชาของ COURSE

#### 4.1.10 การสร้างเอนทิตีใหม่สำหรับรูปแบบความสัมพันธ์แบบ M:N

อย่างที่เรทราบจากบทที่ 3 ที่ว่าถ้าความสัมพันธ์ระหว่างสองเอนทิตีใดๆมีรูปแบบเป็น M:N เราจะต้องทำการสร้างเอนทิตี (ตารางข้อมูล) ใหม่เพื่อเชื่อมโยงถึงความสัมพันธ์ระหว่างสองเอนทิตีหลัก ตารางใหม่ที่สร้างขึ้นจะถูกเรียกว่า “associative entity” หรือ “composite entity” ที่ซึ่งจะประกอบไปด้วย primary key ของทั้งสองเอนทิตีหลักและข้อมูลแอททริบิวต์อื่นๆเพิ่มเติม ตัวอย่างเช่น การลงทะเบียนของนิสิตที่ซึ่งมีกฎเกณฑ์ทางธุรกิจเป็น “นักเรียนคนหนึ่งๆสามารถลงทะเบียนเรียนได้หลายชั้นรายวิชา (ชั้นเรียน) และรายวิชาหนึ่งๆจะเปิดโอกาสให้นักเรียนลงทะเบียนเรียนได้หลายคน” จากการสังเกตกฎเกณฑ์ทางธุรกิจนี้เราจะสังเกตได้ว่าความสัมพันธ์ระหว่างนักเรียน (STUDENT) และ ชั้นเรียน (CLASS) จะมีความสัมพันธ์แบบ M:N ด้วยเหตุนี้เราจำเป็นที่จะต้องสร้างเอนทิตีใหม่ที่เป็น composite entity เพื่อเชื่อมโยงความสัมพันธ์ของสองเอนทิตีหลัก ดังแสดงในรูป 4.23

ในรูป 4.23 เราจะสังเกตได้ว่าตารางข้อมูล ENROLL จะเป็น composite entity ที่ประยุกต์ใช้ primary key ของเอนทิตี STUDENT และ CLASS ประกอบกันเป็น primary key นอกจากนั้นตาราง ENROLL ยังมีข้อมูลอื่นๆ (ที่ไม่ได้เป็น primary key และ/หรือ foreign key) ที่บ่งบอกถึงข้อมูลต่างๆ เช่นเกรดที่ได้จากการเรียนและการลงทะเบียน จำนวนครั้งที่นิสิตคนหนึ่งๆขาดเรียน และข้อมูลอื่นๆเกี่ยวกับพฤติกรรมของนักเรียนในชั้นเรียน นอกเหนือจากการกำหนด primary key และแอททริบิวต์ให้กับเอนทิตีใหม่ที่เกิดขึ้น เราควรที่จะต้องพิจารณาถึงรูปแบบความสัมพันธ์และการมีส่วนร่วมกับความสัมพันธ์ที่ซึ่งเราจะสังเกตได้ว่า CLASS หนึ่งๆอาจยังคงมีอยู่ แม้ว่าจะไม่มีนักเรียนลงทะเบียนเรียนใน CLASS นั้นๆเลย ดังนั้น เราควรที่จะประยุกต์ใช้สัญลักษณ์วงกลม (สัญลักษณ์สำหรับ optional participation) เพื่อแสดงถึงการมีส่วนร่วมกับความสัมพันธ์ของเอนทิตี STUDENT (ดังแสดงในรูป 4.24) โดยจากรูป 4.24 เราจะต้องทำการปรับเปลี่ยนความสัมพันธ์ระหว่างเอนทิตี STUDENT และ CLASS ที่มีลักษณะเป็น M:N ให้เป็นความสัมพันธ์แบบ 1:M สองความสัมพันธ์คือ 1 (STUDENT) : M (ENROLL) และ M (ENROLL) : 1 (CLASS) (ดังแสดงในรูป 4.25) จากรูปแบบความสัมพันธ์ดังกล่าว เราจะสังเกตได้ว่าชั้นเรียนหนึ่งๆจะไม่ปรากฏในตาราง ENROLL ถ้าไม่มีนักเรียนลงทะเบียนเรียนในชั้นเรียนนั้นๆ และด้วยเนื่องจากชั้นเรียนหนึ่งๆจะไม่ปรากฏในตาราง ENROLL จะทำให้เอนทิตี ENROLL กลายเป็น optional participation ภายใต้ความสัมพันธ์กับเอนทิตี CLASS และในทำนองเดียวกัน นักเรียนคนหนึ่งๆอาจยังไม่ทำการลงทะเบียนในตอนเข้าใหม่ ด้วยเหตุนี้จะทำให้เอนทิตี ENROLL เป็น optional participation ในความสัมพันธ์กับเอนทิตี STUDENT



Table name: STUDENT

| STU_NUM | STU_LNAME |
|---------|-----------|
| 321452  | Bowser    |
| 324257  | Smithson  |

Database name: Ch04\_CollegeTry

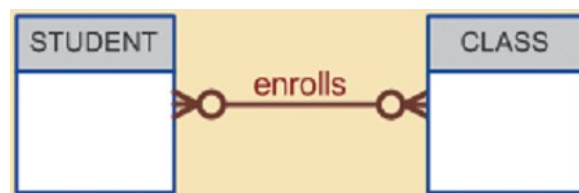
Table name: ENROLL

| CLASS_CODE | STU_NUM | ENROLL_GRADE |
|------------|---------|--------------|
| 10014      | 321452  | C            |
| 10014      | 324257  | B            |
| 10018      | 321452  | A            |
| 10018      | 324257  | B            |
| 10021      | 321452  | C            |
| 10021      | 324257  | C            |

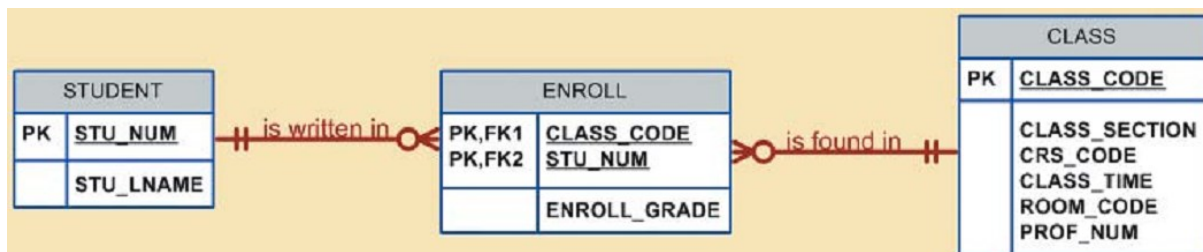
Table name: CLASS

| CLASS_CODE | CRS_CODE | CLASS_SECTION | CLASS_TIME          | CLASS_ROOM | PROF_NUM |
|------------|----------|---------------|---------------------|------------|----------|
| 10014      | ACCT-211 | 3             | TTh 2:30-3:45 p.m.  | BUS252     | 342      |
| 10018      | CIS-220  | 2             | MWTF 9:00-9:50 a.m. | KLR211     | 114      |
| 10021      | QM-261   | 1             | MWTF 8:00-8:50 a.m. | KLR200     | 114      |

รูป 4.23 การปรับเปลี่ยนรูปแบบความสัมพันธ์จาก M:N ไปเป็น 1:M



รูปที่ 4.24 ความสัมพันธ์ระหว่างเอนทิตี STUDENT และ CLASS ที่เป็นแบบ M:N



รูปที่ 4.25 composite entity ใน ERD

## 4.2 ขั้นตอนการออกแบบแผนภาพเชิงสัมพันธ์เอนทิตี

จากที่เราทราบในบทก่อนๆ หน้า กระบวนการออกแบบฐานข้อมูลจะเป็นกระบวนการดำเนินการซ้ำๆ กับกระบวนการดังต่อไปนี้

- การสร้างคำอธิบายต่างๆ เกี่ยวกับการดำเนินการต่างๆ ภายในองค์กร
- การกำหนดกฎเกณฑ์ทางธุรกิจจากคำอธิบายต่างๆ ข้างต้น
- การระบุถึงเอนทิตีหลักและความสัมพันธ์ระหว่างเอนทิตีจากกฎเกณฑ์ทางธุรกิจต่างๆ

- การสร้างแผนภาพเชิงสัมพันธ์เอ็นทิตีเบื้องต้น
- การระบุถึงแอทริบิวต์ต่างๆ primary key และ foreign key ภายในเอ็นทิตีต่างๆ
- การตรวจสอบและแก้ไขปรับปรุงแผนภาพเชิงสัมพันธ์เอ็นทิตีให้มีความสมบูรณ์มากยิ่งขึ้น

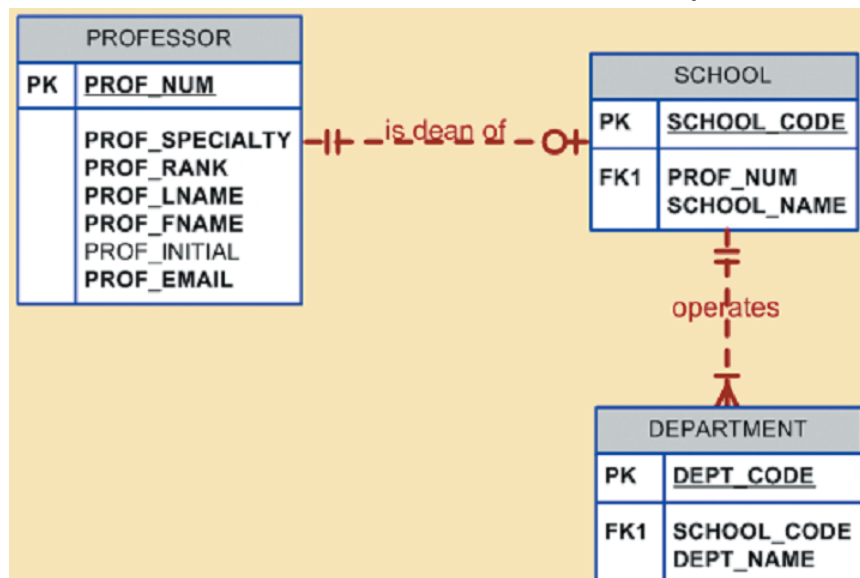
ในระหว่างกระบวนการตรวจสอบและแก้ไขปรับปรุงแผนภาพเชิงสัมพันธ์เอ็นทิตี (ขั้นตอนสุดท้าย) อาจมีเอ็นทิตี แอทริบิวต์ และความสัมพันธ์ต่างๆที่ยังไม่ถูกพิจารณา ด้วยเหตุนี้กระบวนการออกแบบฐานข้อมูลจึงจำเป็นต้องดำเนินการซ้ำเพื่อที่จะได้แผนภาพเชิงสัมพันธ์เอ็นทิตีที่สมบูรณ์ ขั้นตอนการออกแบบฐานข้อมูลจะเสร็จสิ้นก็ต่อเมื่อผู้ใช้และผู้ออกแบบมีความเห็นตรงกัน/ยอมรับแผนภาพเชิงสัมพันธ์เอ็นทิตีและฟังก์ชันต่างๆ ภายใต้กรอบการดำเนินงานขององค์กร (หมายเหตุ—ในระหว่างการเรียนรู้การดำเนินการต่างๆขององค์กร ผู้ออกแบบจะต้องสอบถาม/สัมภาษณ์พนักงานในองค์กรเพื่อที่จะมีความเข้าใจเกี่ยวกับวิธีการ/วัตถุประสงค์ของการดำเนินงานต่างๆ นอกจากนั้น ผู้ออกแบบควรที่จะต้องพิจารณาแบบฟอร์ม (เอกสาร) ต่างๆและรายงานต่างๆที่ถูกประยุกต์ใช้ในการดำเนินการในแต่ละวันด้วยเช่นกัน)

ในการที่จะเข้าใจเกี่ยวกับกระบวนการในการออกแบบฐานข้อมูลและแผนภาพเชิงสัมพันธ์เอ็นทิตี ลองพิจารณาการสัมภาษณ์พนักงานในมหาวิทยาลัยเบื้องต้น ที่ซึ่งเราจะสามารถทราบถึงกฎเกณฑ์ทางธุรกิจดังต่อไปนี้

1. มหาวิทยาลัยจะแบ่งออกเป็นวิทยาลัยต่างๆ เช่น วิทยาลัยทางธุรกิจ วิทยาลัยทางศิลปะ วิทยาลัยทางศิลปะและวิทยาศาสตร์ วิทยาลัยทางการศึกษา และวิทยาลัยด้านวิทยาศาสตร์ประยุกต์ แต่ละวิทยาลัยจะถูกบริหารจัดการโดยหัวหน้าวิทยาลัยที่ซึ่งเป็นอาจารย์คนหนึ่งๆ อาจารย์คนหนึ่งๆจะสามารถเป็นหัวหน้าวิทยาลัยได้เพียงวิทยาลัยเดียวเท่านั้น และอาจารย์คนหนึ่งๆไม่จำเป็นต้องทำหน้าที่เป็นหัวหน้าวิทยาลัย จากข้อมูลข้างต้น เราจะสามารถระบุได้ถึงเอ็นทิตี PROFESSOR และเอ็นทิตี SCHOOL ที่ซึ่งจะมีความสัมพันธ์แบบ 1:1 นอกจากนั้นเรายังสามารถระบุถึง cardinality ในฝั่งของ PROFESSOR และในฝั่งของ SCHOOL ได้เป็น (1,1) และ (0,1) ตามลำดับ
2. แต่ละวิทยาลัยจะมีหลายภาควิชา ตัวอย่างเช่น วิทยาลัยทางธุรกิจจะมีภาควิชาการบัญชี ภาควิชาการจัดการและการตลาด ภาควิชาธุรกิจและการเงิน และภาควิชาระบบเทคโนโลยีสารสนเทศ จากข้อมูลข้างต้น เราจะสังเกตได้ว่า cardinality ของการที่วิทยาลัยหนึ่งๆจะมีภาควิชามีค่าเป็น (1, N) ที่ซึ่งวิทยาลัยหนึ่งๆจะต้องมีอย่างน้อยหนึ่งภาควิชา แต่จะมีจำนวนภาควิชาสูงสุดแบบไม่จำกัด ในทางกลับกัน เราจะสังเกตได้อีกว่า cardinality ของการที่ภาควิชาหนึ่งๆจะต้องสังกัดภายใต้วิทยาลัยหนึ่งๆจะมีค่าเป็น (1,1) จากกฎทางธุรกิจข้อ 1 และ ข้อ 2 เราจะสามารถออกแบบแผนภาพเชิงสัมพันธ์เอ็นทิตีได้ดังรูป 4.26
3. แต่ละภาควิชาจะมีรายวิชาต่างๆที่เปิดสอน ตัวอย่างเช่น ภาควิชาการจัดการและการตลาดจะมีรายวิชาต่างๆที่เปิดสอน เช่น Introduction to Management, Principle of

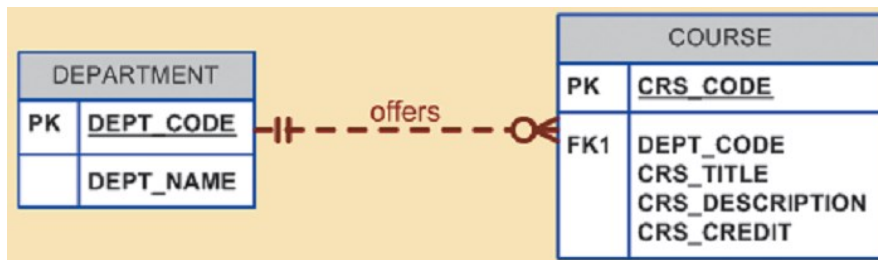
Marketing และ Production Management แต่อย่างไรก็ตาม อาจมีบางภาควิชาที่ไม่มีรายวิชาเปิดสอนเนื่องจากเป็นภาควิชาที่เน้นในด้านการดำเนินการวิจัย ด้วยเหตุนี้จึงทำให้เราสามารถระบุถึงความสัมพันธ์ระหว่างรายวิชา (COURSE) และภาควิชา (DEPARTMENT) ได้ดังรูป 4.27 ที่ซึ่งเอ็นทิตี COURSE จะเป็น optional participation ภายใต้ความสัมพันธ์กับเอ็นทิตี CLASS

- แต่ละรายวิชา (COURSE) จะสามารถมีได้หลายชั้นเรียน (CLASS) และแต่ละชั้นเรียนจะถูกสอนโดยอาจารย์ (PROFESSOR) คนหนึ่งๆ ณ ห้องเรียนหนึ่งๆ ณ เวลาหนึ่งๆ จากกฎเกณฑ์ทางธุรกิจข้างต้นจะทำให้เราทราบถึงความสัมพันธ์ระหว่างเอ็นทิตี COURSE และเอ็นทิตี CLASS ที่ซึ่งจะเป็นความสัมพันธ์ในรูปแบบ 1:M แต่อย่างไรก็ตาม อาจมีบางรายวิชาในภาควิชาที่ไม่ถูกเปิดสอนในชั้นเรียนหนึ่งๆ ดังนั้น เอ็นทิตี CLASS จะเป็น optional participation ภายใต้ความสัมพันธ์กับเอ็นทิตี COURSE (ดังแสดงในรูป 4.28)

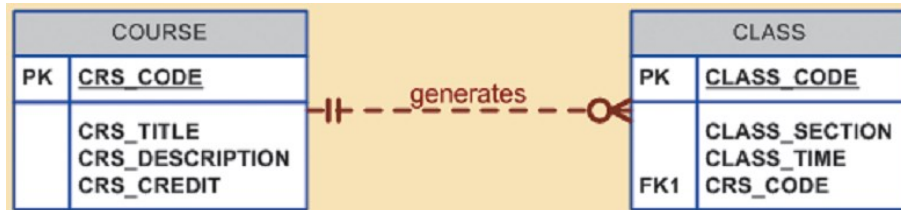


รูป 4.26 ERD แสดงความสัมพันธ์ระหว่างเอ็นทิตี PROFESSOR SCHOOL และ DEPARTMENT

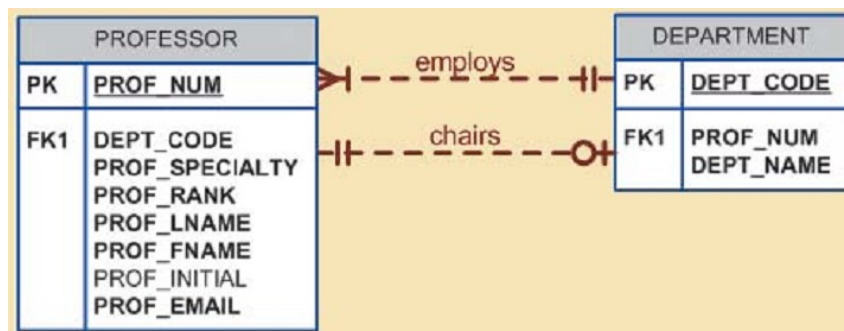
- แต่ละภาควิชาจะต้องมีอาจารย์ในสังกัดอย่างน้อยหนึ่งคน ในภาควิชาหนึ่งๆจะต้องมีอาจารย์คนหนึ่งคนในสังกัดที่ซึ่งจะทำหน้าที่หัวหน้าภาควิชา แต่อาจารย์ในสังกัดคนหนึ่งๆไม่จำเป็นต้องทำหน้าที่เป็นหัวหน้าภาควิชา ด้วยเหตุนี้จึงทำให้เอ็นทิตี PROFESSOR และเอ็นทิตี DEPARTMENT มีความสัมพันธ์ 2 ความสัมพันธ์คือ 1) อาจารย์คนหนึ่งๆจะต้องสังกัดภาควิชาหนึ่งๆ และภาควิชาหนึ่งๆจะต้องมีอาจารย์ภายในสังกัดอย่างน้อยหนึ่งคน และ 2) ภาควิชาหนึ่งๆจะมีอาจารย์คนหนึ่งๆที่ทำหน้าเป็นหัวหน้าภาควิชา และอาจารย์คนหนึ่งๆไม่จำเป็นต้องเป็นหัวหน้าภาควิชา เมื่อพิจารณาถึงความสัมพันธ์ที่เกี่ยวข้องกับการทำหน้าที่เป็นหัวหน้าภาควิชา เราจะสังเกตได้ว่าเอ็นทิตี DEPARTMENT จะเป็น optional participation ภายใต้ความสัมพันธ์ดังกล่าว (ดังแสดงในรูป 4.29)



รูปที่ 4.27 ERD แสดงความสัมพันธ์ระหว่างเอนทิตี DEPARTMENT และ COURSE

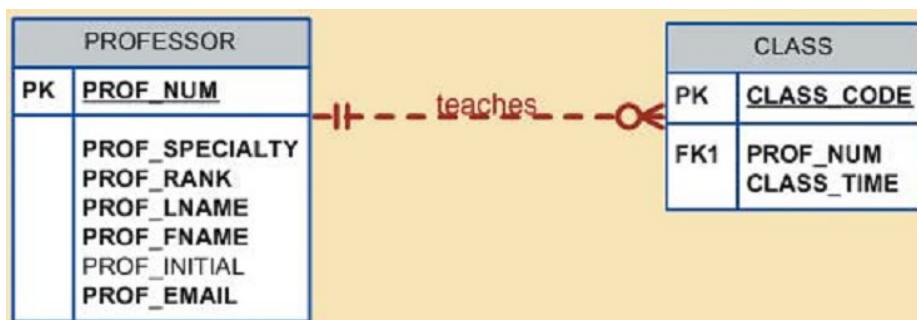


รูปที่ 4.28 ERD แสดงความสัมพันธ์ระหว่างเอนทิตี COURSE และ CLASS



รูปที่ 4.29 ERD แสดงความสัมพันธ์ระหว่างเอนทิตี PROFESSOR และ DEPARTMENT

6. อาจารย์คนหนึ่งๆสามารถเลือกที่จะดำเนินการวิจัยเพียงอย่างเดียวโดยไม่ต้องสอนหนังสือแต่อย่างใด แต่ถ้าอาจารย์ต้องการสอนหนังสือจะมีสิทธิ์สอนได้ไม่เกิน 4 ชั้นเรียน ด้วยเหตุนี้จึงทำให้ความสัมพันธ์ระหว่างเอนทิตี PROFESSOR และ CLASS จะเป็น 1:M ที่ซึ่งเอนทิตี CLASS จะเป็น optional participation ภายใต้ความสัมพันธ์ดังกล่าว (ดังแสดงในรูป 4.30)

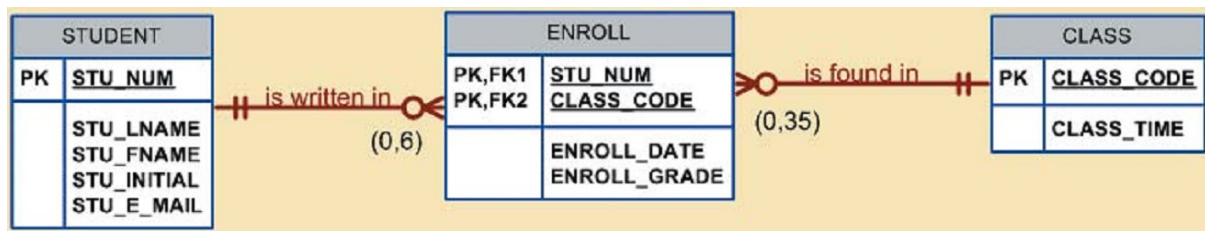


รูปที่ 4.30 ERD แสดงความสัมพันธ์ระหว่างเอนทิตี PROFESSOR และ CLASS

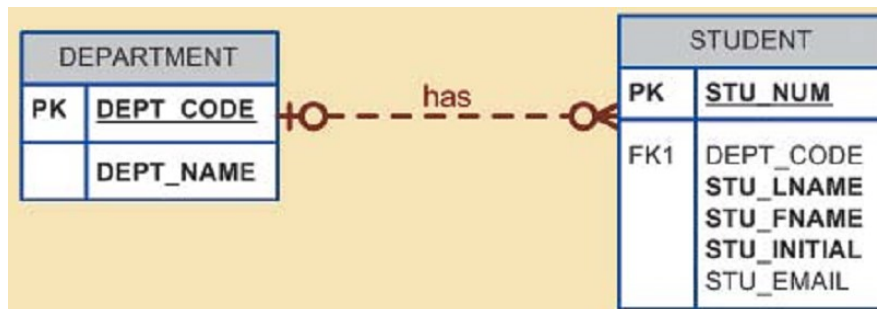
7. นักเรียนคนหนึ่งๆอาจลงทะเบียนเรียนในหลายชั้นเรียน แต่จะสามารถลงทะเบียนเรียนวิชาหนึ่งได้ครั้งเดียวในภาคการศึกษาหนึ่งๆเท่านั้น ตัวอย่างเช่น นักเรียนคนหนึ่งอาจทำการลงทะเบียนเรียน 5 รายวิชา คือ Statistics, Accounting, English, Database และ History เป็นต้น แต่นักเรียนไม่สามารถลงทะเบียนรายวิชา Statistics ซ้ำกันในภาคการศึกษาเดียวกันได้ นอกจากนั้น นักเรียนแต่ละคนจะไม่สามารถลงทะเบียนเรียนเกิน 6 รายวิชาได้ และแต่ละชั้นเรียนจะมีนักเรียนลงทะเบียนเรียนได้ไม่เกิน 35 คน จากข้อมูลทั้งหมดข้างต้น เราจะสามารถระบุได้ว่าความสัมพันธ์ระหว่างเอนทิตี STUDENT และ CLASS จะเป็นความสัมพันธ์ในลักษณะ M:N แต่อย่างไรก็ตาม อาจมีกรณีที่ CLASS หนึ่งๆอาจไม่มีนักเรียนลงทะเบียนเรียนเลยก็เป็นได้ ที่ซึ่งจะทำให้เอนทิตี STUDENT จะเป็น optional participation ภายใต้ความสัมพันธ์กับเอนทิตี CLASS และจากความสัมพันธ์ในลักษณะ M:N เราจะต้องทำการสร้างเอนทิตี ENROLL ขึ้นใหม่ เพื่อปรับเปลี่ยนรูปแบบความสัมพันธ์จาก M:N ไปเป็น 1:M ดังแสดงในรูป 4.31 ที่ซึ่งจะเป็นสองความสัมพันธ์ในรูปแบบ 1:M คือ 1 (STUDENT) : M (ENROLL) และ M (ENROLL) : 1 (CLASS) และเมื่อเราพิจารณาที่ความสัมพันธ์ทั้งสองเราจะสามารถกำหนด cardinality ของการลงทะเบียนเรียนได้เป็น (0,6) และสามารถกำหนด cardinality ของการลงทะเบียนเรียนวิชาหนึ่งๆได้เป็น (0, 35) ตามลำดับ
8. แต่ละภาควิชาจะมีนักเรียนในสังกัดหลายคน แต่นักเรียนคนหนึ่งๆจะสังกัดได้เพียงภาควิชาเดียวเท่านั้น แต่อย่างไรก็ตาม อาจมีนิสิตบางรายที่ยังไม่มีสังกัดในตอนเริ่มแรก ด้วยเหตุนี้จึงทำให้เอนทิตี DEPARTMENT จะเป็น optional participation ภายใต้ความสัมพันธ์กับเอนทิตี STUDENT (ดังแสดงในรูป 4.32)
9. นักเรียนแต่ละคนจะมีอาจารย์ที่ปรึกษาคนหนึ่งๆ แต่อาจารย์ที่ปรึกษาคนหนึ่งๆจะต้องเป็นอาจารย์ที่ซึ่งจะสามารถให้คำปรึกษาแก่นักเรียนได้หลายคน อาจารย์คนหนึ่งๆอาจไม่จำเป็นต้องทำหน้าที่เป็นอาจารย์ที่ปรึกษาของนักเรียนใดเลยก็เป็นได้ ด้วยเหตุนี้ เอนทิตีนักเรียน (STUDENT) จะมีความสัมพันธ์กับเอนทิตีอาจารย์ (PROFESSOR) ที่ซึ่ง STUDENT จะเป็น optional participation ภายใต้ความสัมพันธ์ดังกล่าว (ดังแสดงในรูป 4.33)
10. ชั้นเรียนหนึ่งๆจะถูกสอนในห้องเรียนหนึ่งๆภายในอาคารหนึ่งๆที่ซึ่งอาคารหนึ่งๆจะมีห้องเรียนหลายห้อง ด้วยเหตุนี้ เราจะสามารถพิจารณาความสัมพันธ์ระหว่างอาคาร (BUILDING) ห้องเรียน (ROOM) และชั้นเรียน (CLASS) ได้ดังแสดงในรูป 4.34

จากกฎเกณฑ์ทางธุรกิจทั้ง 10 ข้อข้างต้น เราจะสามารถระบุได้ถึงเอนทิตี SCHOOL, COURSE, DEPARTMENT, CLASS, PROFESSOR, STUDENT, BUILDING, ROOM และ ENROLL (เป็นเอนทิตีที่ถูกสร้างขึ้นใหม่จากความสัมพันธ์ระหว่างเอนทิตี STUDENT และ CLASS) นอกจากนั้น เรายังสามารถกำหนด

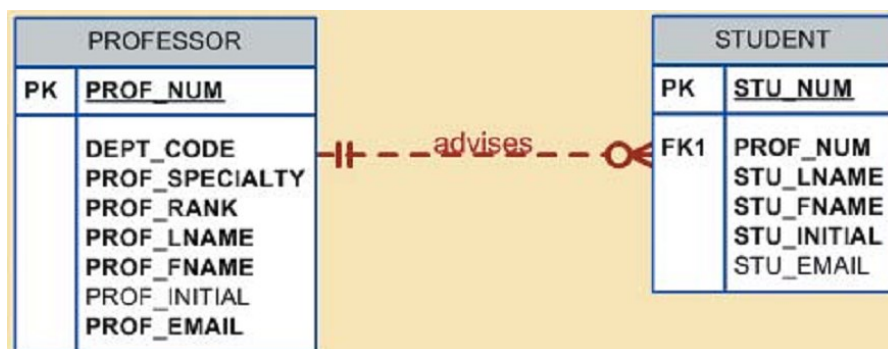
รูปแบบความสัมพันธ์ connectivity และ cardinality จากกฎเกณฑ์ทางธุรกิจดังกล่าวได้ดังแสดงในรูป 4.35 และ 4.36



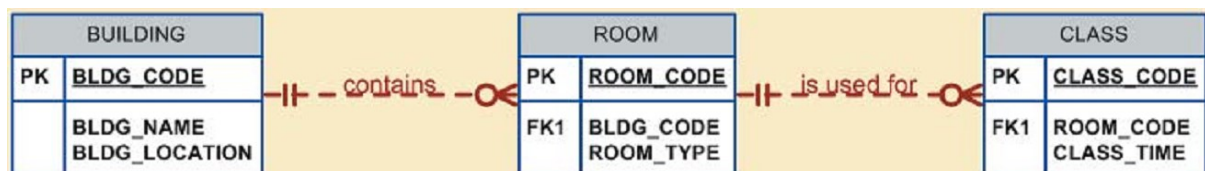
รูปที่ 4.31 ERD แสดงการสร้างเอ็นทิตี ENROLL ใหม่



รูปที่ 4.32 ERD แสดงความสัมพันธ์ระหว่างเอ็นทิตี DEPARTMENT และ STUDENT



รูปที่ 4.33 ERD แสดงความสัมพันธ์ระหว่างเอ็นทิตี PROFESSOR และ STUDENT

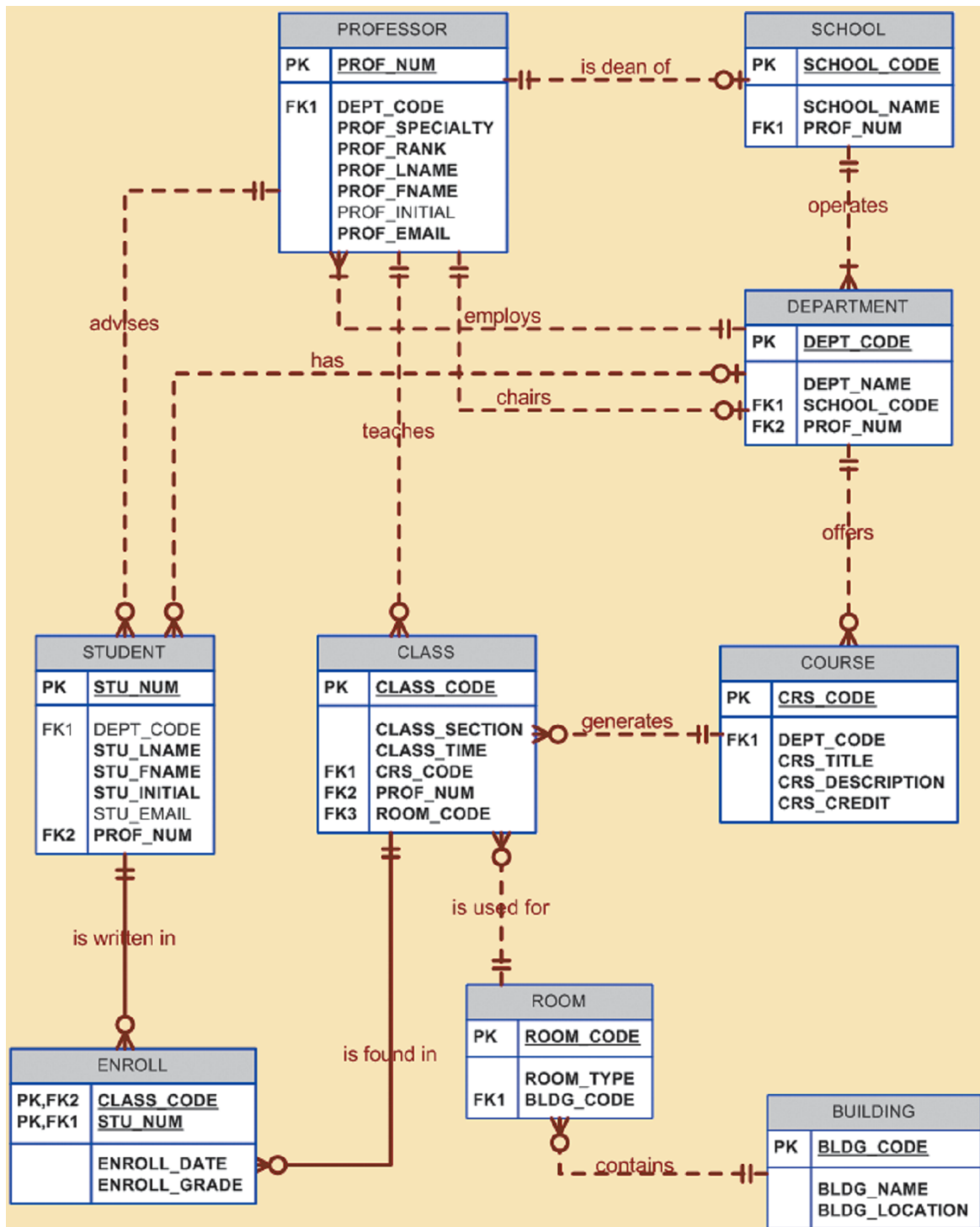


รูปที่ 4.34 ERD แสดงความสัมพันธ์ระหว่างเอ็นทิตี BUILDING ROOM และ CLASS

| ENTITY                                                                                                | RELATIONSHIP | CONNECTIVITY | ENTITY     |
|-------------------------------------------------------------------------------------------------------|--------------|--------------|------------|
| SCHOOL                                                                                                | operates     | 1:M          | DEPARTMENT |
| DEPARTMENT                                                                                            | has          | 1:M          | STUDENT    |
| DEPARTMENT                                                                                            | employs      | 1:M          | PROFESSOR  |
| DEPARTMENT                                                                                            | offers       | 1:M          | COURSE     |
| COURSE                                                                                                | generates    | 1:M          | CLASS      |
| PROFESSOR                                                                                             | is dean of   | 1:1          | SCHOOL     |
| PROFESSOR                                                                                             | chairs       | 1:1          | DEPARTMENT |
| PROFESSOR                                                                                             | teaches      | 1:M          | CLASS      |
| PROFESSOR                                                                                             | advises      | 1:M          | STUDENT    |
| STUDENT                                                                                               | enrolls in   | M:N          | CLASS      |
| BUILDING                                                                                              | contains     | 1:M          | ROOM       |
| ROOM                                                                                                  | is used for  | 1:M          | CLASS      |
| Note: ENROLL is the composite entity that implements the M:N relationship "STUDENT enrolls in CLASS." |              |              |            |

รูปที่ 4.35 รายละเอียดเอนทิตีและความสัมพันธ์จากกฎเกณฑ์ทางธุรกิจทั้ง 10 ข้อ



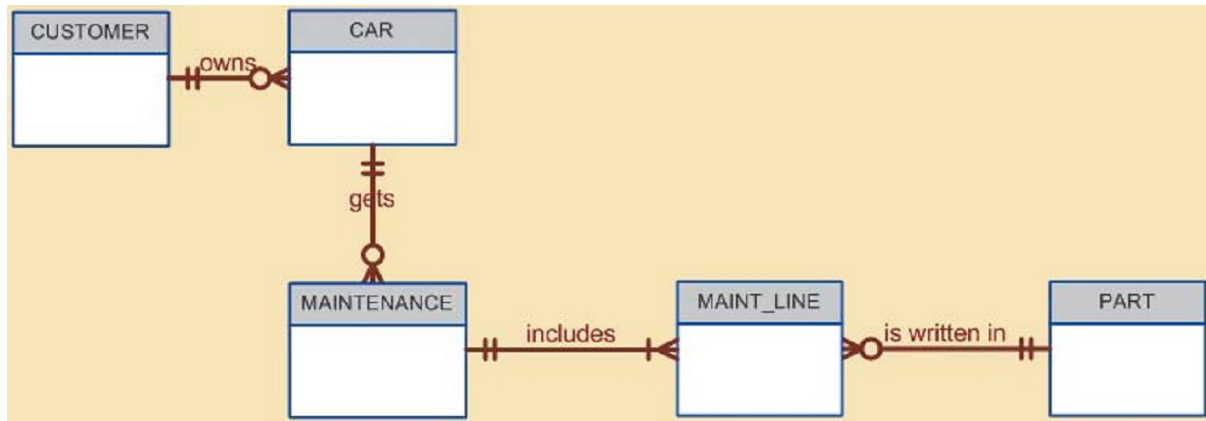


รูปที่ 4.36 ERD ที่สมบูรณ์จากกฎเกณฑ์ทางธุรกิจทั้ง 10 ข้อ

## คำถามท้ายบท

- เงื่อนไขใดที่สามารถตัดสินได้ว่าเอนทิตีหนึ่งๆจะเป็น weak entity? จงยกตัวอย่างประกอบ
- Strong (identifying) relationship คืออะไร? และเราจะสามารถแสดงถึง strong relationship ภายใต้รูปแบบของ Crow's Foot ได้อย่างไร?

3. จงพิจารณากฎเกณฑ์ทางธุรกิจ “an employer may have many degrees” แล้วอธิบายเกี่ยวกับ เอ็นทิตี แอททริบิว และความสัมพันธ์จากกฎเกณฑ์ทางธุรกิจข้างต้น
4. Composite entity คืออะไร? เราจะสามารถประยุกต์ใช้ composite key ได้อย่างไร?
5. จงพิจารณา ERD ดังรูปแล้วตอบคำถามต่อไปนี้



- a. จงเขียนกฎเกณฑ์ทางธุรกิจที่สะท้อนถึงเอ็นทิตีและความสัมพันธ์ต่างๆ
- b. ระบุถึง cardinality ทั้งหมดใน ERD
6. Recursive relationship คืออะไร? จงยกตัวอย่างประกอบ
7. เราจะสามารถแสดงส่วนประกอบต่างๆดังต่อไปนี้ ใน ERD ภายใต้รูปแบบของ Crow's Foot ได้อย่างไร
  - a. เอ็นทิตี
  - b. Cardinality (0, N)
  - c. Weak relationship
  - d. Strong relationship
8. จงอธิบายถึงความแตกต่างระหว่าง composite key และ composite attribute?
9. เมื่อเราต้องทำการจัดเก็บข้อมูลแอททริบิวที่มีลักษณะเป็นแบบ multivalued attribute เราจะสามารถจัดการได้อย่างไร?
10. Derived attribute มีลักษณะอย่างไร? จงยกตัวอย่างประกอบ