**ใบความรู้ที่5**

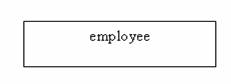
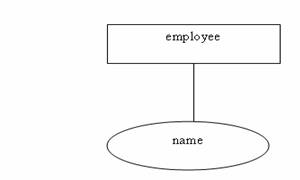
**การวิเคราะห์และออกแบบระบบ**

**การสร้างโมเดลความสัมพันธ์ระหว่างข้อมูล : ER-DIAGRAM**

**แนวคิดเกี่ยวกับ ER-DIAGRAM**   
ER-DIAGRAM ประกอบด้วยองค์ประกอบพื้นฐานดังนี้

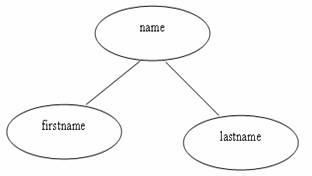
* เอนทิตี้ (Entity) เป็นวัตถุ หรือสิ่งของที่เราสนใจในระบบงานนั้น ๆ
* แอททริบิว (Attribute) เป็นคุณสมบัติของวัตถุที่เราสนใจ
* ความสัมพันธ์ (Relationship) คือ ความสัมพันธ์ระหว่างเอนทิตี้

**เอนทิตี้ (Entity)**  
เอนทิตี้  หมายถึง สิ่งของหรือวัตถุที่เราสนใจ ซึ่งอาจจับต้องได้และเป็นได้ทั้งนามธรรม โดยทั่วไป เอนทิตี้จะมีลักษณะที่แยกออกจากกันไป เช่น เอนทิตี้พนักงาน จะแยกออกเป็นของพนักงานเลย เอนทิตี้เงินเดือนของพนักงานคนหนึ่งก็อาจเป็นเอนทิตี้หนึ่งในระบบของโรงงาน   
โดยทั่วไปแล้ว เอนทิตี้จะมีกลุ่มที่บอกคุณสมบัติที่บอกลักษณะของเอนทิตี้ เช่น พนักงานมีรหัส ชื่อ นามสกุล และแผนก โดยจะมีค่าของคุณสมบัติบางกลุ่มที่ทำให้สามารถแยกเอนทิตี้ออกจากเอนทิตี้อื่นได้ เช่น รหัสพนักงานที่จะไม่มีพนักงานคนไหนใช้ซ้ำกันเลย เราเรียกค่าวของคุณสมบัติกลุ่มนี้ว่าเป็นคีย์ของเอนทิตี้

รูปสัญลักษณ์ของเอนทิตี้ คือ รูปสี่เหลี่ยมผืนผ้า ตัวอย่างเช่น   
  
**แอททริบิวท์ (Attribute)**    
**Attribute** คือ คุณสมบัติของวัตถุหรือสิ่งของที่เราสนใจ โดยอธิบายรายละเอียดต่าง ๆ ที่เกี่ยวข้องกับลักษณะของเอนทิตี้ โดยคุณสมบัตินี้มีอยู่ในทุกเอนทิตี้ เช่น ชื่อ นามสกุล ที่อยู่ แผนก เป็น Attribute ของเอนทิตี้พนักงาน  
โดยทั่วไปแล้วโมเดลข้อมูล เรามักจะพบว่า Attribute มีลักษณะข้อมูลพื้นฐานอยู่โดยที่ไม่ต้องมีคำอธิบายมากมาย และ Attribute ก็ไม่สามารถอยู่แบบโดด ๆ ได้โดยที่ไม่มีเอนทิตี้หรือความสัมพันธ์  
รูปสัญลักษณ์ของ Attribute คือ รูปวงรีโดยที่จะมีเส้นเชื่อมต่อกับเอนทิตี้ ตัวอย่างเช่น  


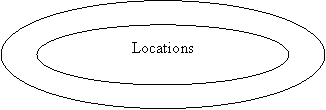
ชนิดของ Attribute สามารถแบ่งออกได้หลายลักษณะดังนี้  
**Simple และ Composite**

* Simple Attribute คือ Attribute ที่ไม่สามารถแยกออกเป็นส่วนย่อยได้เช่น รหัส
* Composite Attribute คือ Attribute ที่สามารถแยกออกเป็นส่วนย่อยได้เช่น ชื่อ อาจจะประกอบด้วยชื่อต้น และชื่อสกุล เป็นต้น โดยยกตัวอย่างเช่น



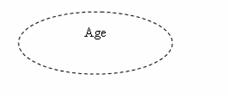
**Single – valued และ Multi – valued attribute**

* Single – valued คือ ค่าของเอนทิตี้ที่สามารถมีได้แค่ค่าเดียว เช่น วันเกิด สำหรับพนักงานแล้วสามารถมีได้เพียงค่าเดียว จึงให้สัญลักษณ์ของ Attribute ปกติ
* Multi – valued คือ ค่าที่เป็นไปได้มากกว่า 1 ค่า เช่น ทำเลที่ตั้งของโรงงานสามารถมีได้มากกว่า 1 แห่ง
* รูปสัญลักษณ์ที่ใช้จะเป็นรูปวงรีซ้อนกัน 2 รูป โดยจะยกตัวอย่างเช่น



**Stored และ Derived attribute**

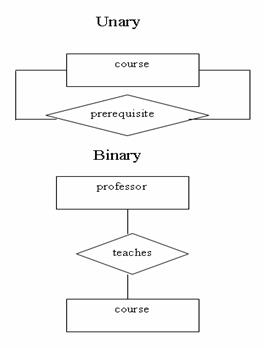
* Stored Attribute จะเป็น Attribute ที่เก็บอยู่ในฐานข้อมูล เช่น วันเกิด ใช้สัญลักษณ์ปกติ
* Derived Attribute เป็น Attribute ที่เกิดจากการคำนวณ เช่น อายุ เกิดจากการคำนวณวันเกิดกับช่วงเวลาปัจจุบัน
* รูปสัญลักษณ์ คือ รูปวงรีมีเส้นประรอบ ๆ โดยจะยกตัวอย่าง เช่น

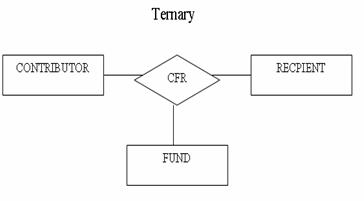


**ความสัมพันธ์ (Relationship)**   
เอนทิตี้แต่จะต้องมีความสัมพันธ์ร่วมกัน โดยจะมีชื่อแสดงความสัมพันธ์ร่วมกันซึ่งจะใช้รูปภาพสัญลักษณ์สี่เหลี่ยมรูปว่าวแสดงความสัมพันธ์ระหว่างเอนทิต  
ี้และระบุชื่อความสัมพันธ์ลงในสี่เหลี่ยม ดังตัวอย่างเช่น รูปนี้แสดงให้เห็นถึงความสัมพันธ์ระหว่างเอนทิตี้อาจารย์กับกลุ่มเรียน  


**ระดับชั้นของความสัมพันธ์ (Relationships Degree)** จะบอกถึงความสัมพันธ์ระหว่างเอนทิตี้ มีดังนี้

* ความสัมพันธ์เอนทิตี้เดียว (Unary Relationships) หมายถึง เอนทิตี้หนึ่ง ๆ จะมีความสัมพันธ์กับตัวมันเอง
* ความสัมพันธ์สองเอนทิตี้ (Binary Relationships) หมายถึง เอนทิตี้สองเอนทิตี้จะมีความสัมพันธ์กัน
* ความสัมพันธ์สามเอนทิตี้(Ternary Relationships) หมายถึง เอนทิตี้สามเอนทิตี้มีความสัมพันธ์กัน

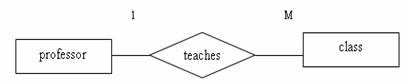




**ภาพที่ 10.1 แสดงตัวอย่างของระดับชั้นของข้อความ**

**การระบุตำแหน่งความสัมพันธ์ระหว่างเอนทิตี้ (Connectivity)**   
การระบุตำแหน่งความสัมพันธ์ระหว่างเอนทิตี้ (Connectivity) ว่าเป็นแบบหนึ่งต่อหนึ่ง (One to One Relationships) , แบบหนึ่งต่อกลุ่ม (One to Many Relationships) หรือ แบบกลุ่มต่อกลุ่ม (Many to Many Relationships) นั้นจะใช้ Connectivity เพื่อระบุตำแหน่ง 1, M หรือ N ไว้ข้างใดของเอนทิตี้



**ภาพที่ 10.2 แสดงความสัมพันธ์แบบ One to One Relationships**  
จากตัวอย่างนี้ จะแสดงความสัมพันธ์ระหว่างนักศึกษากับสัญญาเงินกู้ โดยที่นักศึกษาหนึ่งคนทำสัญญาเงินกู้ได้เพียงครั้งเดียว สัญญาการกู้เงินแต่ละฉบับถูกลงชื่อกู้ได้จากหนักศึกษาเพียงคนเดียวเท่านั้น ความสัมพันธ์การกู้เงินที่เชื่อมระหว่างนักศึกษาและสัญญากู้เงินจึงเป็นแบบ 1-1   


**ภาพที่ 10.3 แสดงความสัมพันธ์แบบ One to Many Relationships**

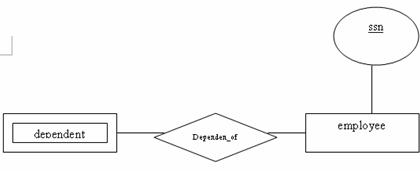
                จากตัวอย่างนี้ จะประกอบด้วยเอนทิตี้อาจารย์กับเอนทิตี้กลุ่มเรียน มีความสัมพันธ์แบบหนึ่งต่อกลุ่ม หมายความว่า อาจารย์จะสอนได้หลายกลุ่มเรียน แต่ละกลุ่มเรียนจะมีอาจารย์สอนได้เพียงคนเดียวไว้ด้านเอนทิตี้อาจารย์และตัวอักษร M ไว้ด้านเอนทิตี้กลุ่มเรียน  


**ภาพที่ 10.4 แสดงความสัมพันธ์แบบ Many to Many Relationships**

                    จากตัวอย่างนี้ ประกอบด้วยเอนทิตี้นักเรียนกับเอนทิตี้วิชาเรียน โดยที่นักศึกษาแต่ละคนลงทะเบียนเรียนวิชาได้มากกว่า 1 วิชา แต่ละวิชามีนักศึกษาได้มากกว่า 1 คน ความสัมพันธ์ขอลการลงทะเบียนของนักศึกษากับวิชาเป็นแบบ N: M

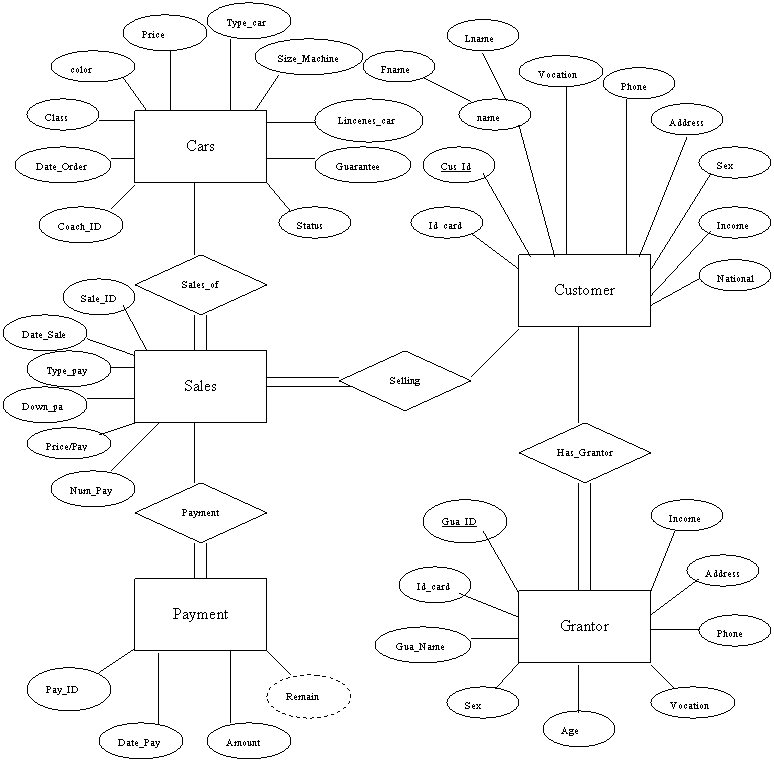
**Keys**

* Super Key : Attribute หรือกลุ่มของ Attribute ซึ่งมีค่าแตกต่างกันไปในแต่ละเอนทิตี้ สามารถระบุเอนทิตี้เฉพาะตัวหนึ่ง ๆ ได้
* Candidate Key : Subset ที่เล็กที่สุดของ Super Key ที่สามารถระบุเฉพาะเอนทิตี้นั้นได้
* Primary Key : Candidate Key ที่ถูกเลือกให้เป็นตัวระบุหรือ Identity เอนทิตี้เฉพาะตัว

**Strong VS Weak Entity Sets**  
บางครั้งเราอาจพบว่าเอนทิตี้ที่มี (Primary Key) ประกอบจาก Primary Key ของ Entity Set อื่น ๆนั่นคือ เอนทิตี้ไม่มี  Primary Key  หรือ Attribute เพียงพอในการสร้าง  Primary Key ได้ด้วยตนเองเราเรียกเอนทิตี้นี้ว่า Weak Entity Set ดังนั้น หากจะระบุถึงเอนทิตี้นี้ได้จะต้องผูกสัมพันธ์กับบางเอนทิตี้ผ่าน Primary Key เป็นของเอนทิตี้ที่สัมพันธ์กับ Weak Entity ที่มี Primary Key ว่าเป็น Strong Entity Set เราพบว่า Weak Entity นั้นจะต้องสัมพันธ์เกี่ยวข้องแบบ Total Participate กับ Strong Entity เสมอ  
ตัวอย่างเช่น ความสัมพันธ์ของพนักงานและคนในอุปการะ โดยที่คนในอุปการะหนึ่งคนเกี่ยวข้องโดยตรงกับพนักงานหนึ่งคนเท่านั้น แต่พนักงานอาจไม่มีหรือมีมากว่าหนึ่งคนในอุปการะ ซึ่งเราจะพบว่าความสัมพันธ์ระหว่าง Weak กับ Strong Entity จะเป็นแบบกลุ่มต่อหนึ่ง  


**ภาพที่ 10.5 แสดงความสัมพันธ์แบบกลุ่มต่อหนึ่ง**

**การแปลง E-R MODEL ให้อยู่ในรูปแบบโครงสร้างฐานข้อมูล**  
การแปลง E-R MODEL ให้อยู่ในรูปแบบโครงสร้างฐานข้อมูลหรือตารางของข้อมูลมีกฎดังนี้  
1.แปลงเอนทิตี้ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง  (One to One Relationships) ไปเป็นตาราง โดยแทนที่หนึ่งเอนทิตี้เป็นหนึ่งตาราง Attribute แต่ละเอนทิตี้เป็นฟิลด์หรือคอลัมน์แต่ละตาราง  
2.แปลงเอนทิตี้ที่มีความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One to Many Relationships) ไปเป็นตาราง โดยด้านเอนทิตี้ที่เป็นตัวเลข 1 นั้นสามารถแปลงเป็นตารางได้ทันที Attribute ของเอนทิตี้นั้นจะเป็นฟิลด์ของตารางทันที ส่วนด้านเอนทิตี้ที่เป็นตัวอักษร M ให้แผลงเอนทิตี้เป็นตารางโดยมี Attribute ของเอนทิตี้ตัวมันเอง และนำคีย์หลักของเอนทิตี้ที่เป็นเลข 1 มาใส่ฟิลด์ของตารางนั้นด้วย  
3.แปลงเอนทิตี้ที่มีความสัมพันธ์แบบกลุ่มต่อกลุ่ม  (Many to Many Relationships) ไปเป็นตารางโดยสร้างเอนทิตี้กลาง (Composite Entity) เอนทิตี้กลางจะนำคีย์หลักของทั้งสองตารางมาเป็นคีย์หลักของเอนทิตี้กลางด้วย ส่วนเอนทิตี้ทั้งสองที่อยู่ระหว่างเอนทิตี้กลางก็แปลงเป็นตารางได้ โดยนำเอา Attribute ของแต่ละเอนทิตี้ไปเป็นฟิลด์ (ทำตามกฎของความสัมพันธ์แบบหนึ่งต่อหนึ่ง)



**ภาพที่ 10.6 ตัวอย่างของระบบซื้อขายรถยนต์**

|  |  |  |  |
| --- | --- | --- | --- |
| สัญลักษณ์ | ความหมาย | สัญลักษณ์ | ความหมาย |
| http://learningsystem.6te.net/assets/images/clip_image001_0001.gif | Entity set | http://learningsystem.6te.net/assets/images/clip_image002_0001.gif | Discriminator key attribute |
| http://learningsystem.6te.net/assets/images/clip_image003_0001.gif | Weak entity set | http://learningsystem.6te.net/assets/images/clip_image004_0001.gif | Composite attribute |
| http://learningsystem.6te.net/assets/images/clip_image005_0008.gif | Relationship set | http://learningsystem.6te.net/assets/images/clip_image006_0002.gif | Derived attribute |
| http://learningsystem.6te.net/assets/images/clip_image007_0004.gif | Identifying relationship set | http://learningsystem.6te.net/assets/images/clip_image008_0002.gif | Key attribute |
| http://learningsystem.6te.net/assets/images/clip_image009_0003.gif | Attribute | http://learningsystem.6te.net/assets/images/clip_image010_0003.gif | Multi valued attribute |

**ภาพที่ 10.7 แสดงสัญลักษณ์ของ E-RMODEL**

**คำศัพท์  บทที่ 10**  
**การเริ่มต้นโครงการและการศึกษาเบื้องต้น**

|  |  |
| --- | --- |
| Entity | วัตถุ หรือวิ่งของที่ราสนใจ |
| Attribute | คุณสมบัติของวัตถุที่เราสนใจ |
| Relationship | ความสัมพันธ์ระหว่าง เอนทิตี้ |
| Simple Attribute | Attribute ที่ไม่สามารถแยกออกเป็นส่วนย่อย |
| Composite Attribute | Attribute ที่สามารถแยกออกเป็นส่วนย่อย |
| Single-valued | ค่าของเอนทิตี้ที่สามารถมีได้แค่ค่าเดียว |
| Multi-valued | ค่าที่เป็นได้มากกว่า 1 ค่า |
| Stored Attribute | Attribute ที่เก็บอยู่ในฐานข้อมูล |
| Derived Attribute | Attribute ที่เกิดจากการคำนวณ |
| Relationships Degree | ระดับชั้นของความสัมพันธ์ |
| Unary Relationships | ความสัมพันธ์เอนทิตี้เดียว |
| Binary Relationships | ความสัมพันธ์สองเอนทิตี้ |
| Ternary Relationships | ความสัมพันธ์สามเอนทิตี้ |
| Connectivity | การระบุความสัมพันธ์ระหว่างเอนทิตี้ |
| One to One Relationships | ความสัมพันธ์แบบหนึ่งต่อหนึ่ง |
| One to Many Relationships | ความสัมพันธ์แบบหนึ่งต่อกลุ่ม |
| Many to Many Relationships | ความสัมพันธ์แบบกลุ่มต่อกลุ่ม |
| Super Key | กลุ่มของ Attribute ซึ่งมีค่าแตกต่างกัน |
| Candidate Key | Subset ที่เล็กที่สุดของ Super Key |
| Primary Key | Candidate Key ที่ถูกเลือกให้เป็นตัวระบุ |
| Composite Entity | เอนทิตี้กลาง |
| Weak Entity | เอนทิตี้ที่ไม่มี Primary Key เป็นของตนเอง |
| Strong Entity | เอนทิตี้ที่มี Primary Key เป็นของตนเอง |