บทที่ 4 โมเดลเชิงสัมพันธ์

<u>หัวข้อเรื่อง</u>

4.1 โมเดลข้อมูลเชิงสัมพันธ์ (Entity Relationship Model)
4.2 เอนทิตี้ (Entity)
4.3 แอททริบิวต์ (Attribute)
4.4 ความสัมพันธ์ระหว่างเอนทิตี้ (Entity Relationship)
4.5 ความเกี่ยวข้องกันระหว่างเอนทิตี้กับความสัมพันธ์ (Participation)
4.6 การแปลง E-R Diagram ไปเป็นตารางเก็บข้อมูล
4.7 การเปลี่ยนจากความสัมพันธ์เป็นตาราง

สรุปบทเรียน
ประเมินผลหลังเรียน

<u>จุดประสงค์ทั่วไป</u>

- เพื่อให้มีความรู้ความเข้าใจเกี่ยวกับระบบฐานข้อมูลเบื้องต้น

<u>จุดประสงค์เชิงพฤติกรรม</u>

เมื่อศึกษาบทที่ 4 จบแล้ว ผู้เรียนสามารถ

1. บอกสัญลักษณ์ที่ใช้ใน E-R diagram
2. บอกสัญลักษณ์ที่ใช้แทนความสัมพันธ์ของข้อมูลได้
3. สามารถอธิบายความสัมพันธ์ของเอนทิตี้แบบต่างๆ ในฐานข้อมูลเชิงสัมพันธ์ได้
4. สามารถแปลง E-R Diagram ให้เป็นตารางในการเก็บข้อมูลได้
5. สามารถเปลี่ยนความสัมพันธ์เป็นตารางได้

4.1 โมเดลข้อมูลเชิงสัมพันธ์ (Entity Relationship Model)

โมเดลข้อมูลเชิงสัมพันธ์ (Entity Relationship Model) หรือ E-R Model ถูกนำเสนอโดย Chen (1976) และได้รับการพัฒนาจะสมบูรณ์ในปี 1979 เพื่อใช้ในการอธิบายฐานข้อมูลในรูปแบบแผนภาพ เรียกว่า E-R diagram นั้นคือมนุษย์พยายาม เขียนสร้างที่เป็นจินตภาพ (ในที่นี้คือมุมมองที่เขามองฐานข้อมูล) ให้เป็นกายภาพ ในรูปแผนภาพ แผนภาพ ER ดังกล่าวใช้พื้นฐานของโมเดลข้อมูลเชิงตรรกะ (Logical Data Model) ของ Fleming ที่เสนอไว้ในปี 1989

เป็นวิธีการแสดงความต้องการสารสนเทศในระบบธุรกิจให้เป็นแผนภาพ ในขั้นตอนการวิเคราะห์และออกแบบระบบฐานข้อมูล เน้นตัวข้อมูลที่มีอยู่จริงโดยไม่คำนึงถึงรายละเอียดในการ ติดตั้ง, ความต้องการพิเศษอื่นในแง่การใช้งาน และความเร็วในการสืบค้นข้อมูล หรืออาจกล่าวอีกนัยหนึ่งว่าโมเดลข้อมูลเชิงตรรกะเป็นการสร้างโครงร่างวิวของ ผู้ใช้ (Skeletal User View) จะแสดงข้อมูลในขอบเขตที่ผู้ออกแบบสนใจโดยมีสิ่งที่ต้องกำหนดเป็นพื้นฐาน ได้แก่ เอนทิตี้, รีเลชั่นชิป, แอททริบิวต์ ในแง่ของ ER-Diagram ประกอบด้วย

เอนทิตี้ (Entity) คือ สิ่งที่มีอยู่จริง จับต้องได้ หรือเป็นจินตภาพที่แสดงความเป็นหนึ่งเดียว เมื่อกล่าวถึงแล้วทุกคนเข้าใจตรงกัน เช่น สินค้า, วัตถุดิบ, แผนกต่าง ๆ, การสั่งซื้อ
รีเลชั่นชิป (Relationship)คือความสัมพันธ์ซึ่งเป็นลักษณะการเกี่ยวพันกันระหว่างเอนทิตี้หนึ่งกับตัว มันเองหรือ เอนทิตี้อื่น อาจเป็นความสัมพันธ์ที่มากกว่า 2 เอนทิตี้ก็ได้ เช่น แผนกจัดซื้อทำการสั่งซื้อสินค้าหรือวัตถุดิบ
แอททริบิวต์ (Attribute) คือกลุ่มของค่าความจริงใด ๆ ที่เป็นรายละเอียดของเอนทิตี้ซึ่งแสดงลักษณะ และ คุณสมบัติของเอนทิตี้ ทำให้เข้าใจเอนทิตี้ได้ลึกซึ้งยิ่งขึ้น และเป็นสิ่งที่ไม่สามารถแตกย่อยลงไปได้อีกโดยไม่เสียความหมายไป เช่น รหัสสินค้า, สถานที่เก็บ, ชื่อสินค้า, ราคา
นอกจากนั้นยังมีการระบุด้วยว่าแอททริบิวต์ใดเป็นคีย์กำหนดกฎข้อบังคับต่าง ๆ ของเอนทิตี้และรีเลชั่นชิป

4.2 เอนทิตี้ (Entity)

เอนทิตี้ คือ สิ่งที่คงอยู่ สามารถระบุได้ ในความจริง เช่น บุคคล เหตุการณ์ สถานที่ โดยทั่วไป เอนทิตี้ มักจะอยู่ในรูปของนาม เอนทิตี้สามารถมีพรอพเพอตี้ หรือคุณสมบัติ ได้หลายอย่าง เช่น บุคคล มีคุณสมบัติได้หลายอย่าง เช่น ชื่อ ที่อยู่ อายุ เพศ วุฒิการศึกษา ฯลฯ

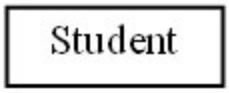
4.2.1 Regular Entity

Regular Entity หรือ เอนทิตี้ทั่วๆไป ที่ตัวเอนทิตี้เองมีคุณสมบัติ ที่สามารถ กำหนด ให้เป็นเอกลักษณ์เฉพาะกับเอนทิตี้ได้ คือมีแอททริบิวต์ใดแอททริบิวต์หนึ่งที่แยก ความแตกต่างของข้อมูลแต่ละทูเพิลได้

เอนทิตี้ STUDENT

SID	Sname	Address	Grade
4739010001	ประสาน สมพงษ์	111 หมู่ 5 สามเสน กรุงเทพฯ	3.5
4739010002	อำนาจ รักดี	222 หมู่ 2 บางซื่อ กรุงเทพฯ	2.8
4739010003	ประนอม รักการอ่าน	333 หมู่ 1 หนองจอก กรุงเทพฯ	3.0

สัญลักษณ์ของ Regular Entity



รูปที่ 4.1 ตัวอย่าง Regular Entity และสัญลักษณ์

จากรูปที่ 4.1 เป็นตัวอย่างของ Regular Entity ที่ชื่อว่า STUDENT ซึ่งสัญลักษณ์จะเขียน แทนด้วยสี่เหลี่ยมผืนผ้า และมีชื่อของเอนทิตี้อยู่ภายใน

4.2.2 Weak Entity Weak Entity หรือเอนทิตี้อ่อนแอ คือเอนทิตี้ที่ไม่สามารถคงอยู่ หรือมีคุณสมบัติที่แยกความ แตกต่างของข้อมูล แต่ละทูเพิลได้ด้วยคุณสมบัติของตัวเองได้
ตัวอย่างของ Weak Entity คือเวลาเข้าออกของพนักงาน

เอนทิตี้ TimeStamp

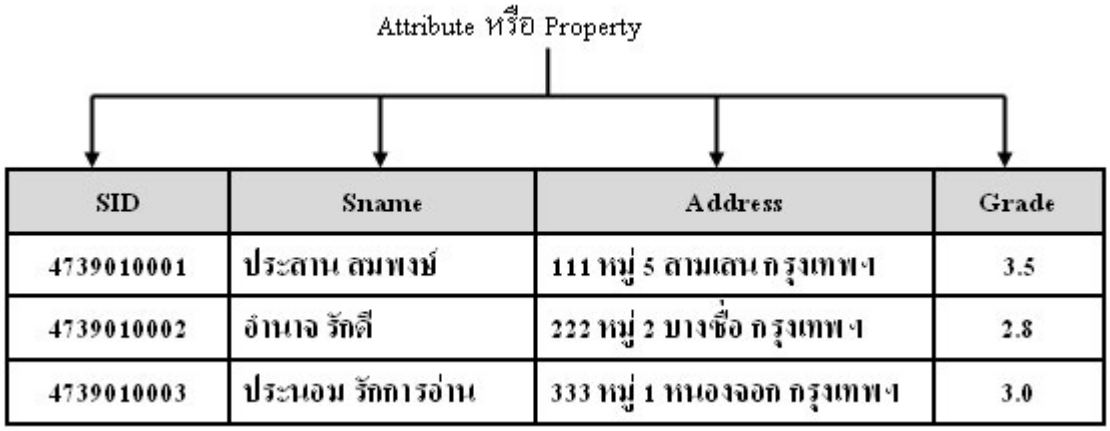
EmpID	Date	TimeIn	TimeOut
47035501	2/7/47	7:30	17:30
47035502	2/7/47	8:00	18:00
470355503	2/7/47	7:45	17:45

รูปที่ 4.2 ตัวอย่าง Weak Entity และสัญลักษณ์

จากตัวอย่าง เอนทิตี้ Timestamp มีคุณสมบัติคือวันที่และเวลาเข้า เวลาออกเท่านั้น ซึ่งถ้ามีคุณสมบัติเพียงเท่านี้ จะไม่สามารถแยกความแตกต่างของข้อมูลในแต่ละทูเพิลได้เลย ดังนั้น จึงต้องมี การนำแอททริบิวต์ EmpID มารวมเข้ากับ เอนทิตี้ Timestamp เพื่อให้สามารถแยก ความแตกต่าง ของข้อมูลในแต่ละทูเพิลได้

4.3 แอททริบิวต์ (Attribute)

แอททริบิวต์ หรือ พรอพเพอตี้ (Property) คือคุณสมบัติต่างๆ ของเอนทิตี้ ซึ่งในแต่ละเอนทิตี้สามารถมี คุณสมบัติต่างๆ ได้มากมาย ขึ้นอยู่กับว่าเราต้องการจัดเก็บข้อมูลอะไรบ้าง เช่น คุณสมบัติของพนักงาน คือ จะมีรหัสพนักงาน ชื่อ-นามสกุล ที่อยู่ เงินเดือน ฯลฯ เป็นต้น

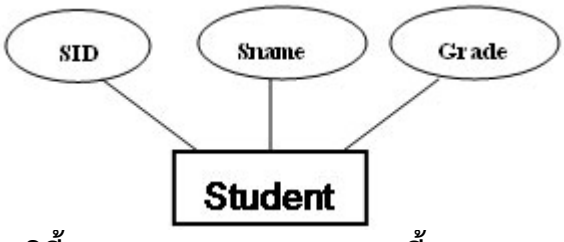


รูปที่ 4.3 ตัวอย่างของพรอพเพอตี้

ประเภทของพรอพเพอตี้ มีดังนี้

4.3.1 Simple Property

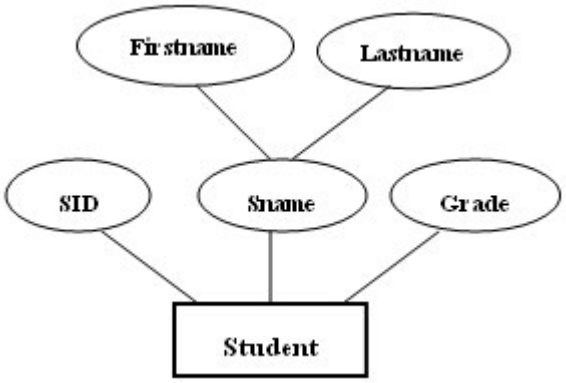
คือพรอพเพอตี้ ที่ค่าภายในพรอพเพอตี้นั้น จะไม่แบ่งแยกย่อยออกมาอีก เช่น รหัสนักศึกษา ชื่อ เกรด เป็นต้น สัญลักษณ์ที่ใช้ คือวงรี ดังรูปที่ 4.6 เป็นตัวอย่างเอนทิตี้ Student ซึ่งมีพรอพเพอตี้ คือ SID, Sname และ Grade



รูปที่ 4.4 เอนทิตี้ Student และ พรอพเพอตี้ SID, Sname, Grade

4.3.2 Composite Property

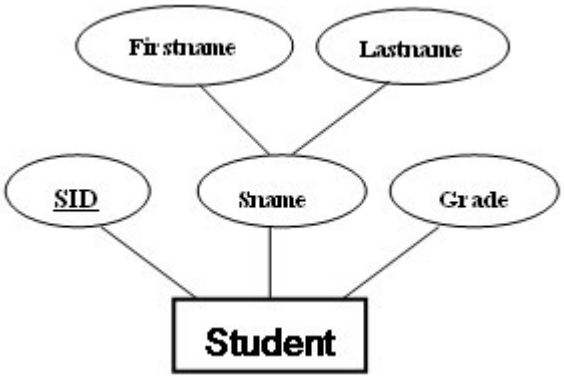
คือพรอพเพอตี้ ที่ค่าภายในพรอพเพอตี้นั้น สามารถแบ่งแยกย่อยออกมาได้อีก เช่น ชื่อ ที่อยู่ เป็นต้น ซึ่งสามารถแบ่งแยกย่อยเป็น ชื่อ และนามสกุล ส่วนที่อยู่สามารถแบ่งแยกย่อยได้เป็น เลขที่ ถนน อำเภอ จังหวัด เป็นต้น จากรูปที่ 4.5 แสดงพรอพเพอตี้ Sname ซึ่งประกอบด้วย FirstName และ LastName



รูปที่ 4.5 เอนทิตี้ Student และ Composite Property Sname

4.3.3 Key Property

คือพรอพเพอตี้ ที่ใช้แยกความแตกต่างของข้อมูลในแต่ละทูเพิล ซึ่ง ค่าของพรอพเพอตี้ ที่ทำหน้าที่เป็น Key นั้นจะมีค่าไม่ซ้ำกัน ดังแสดงในรูปที่ 4.6 เอนทิตี้ Student มีพรอพเพอตี้ SID เป็น Key ซึ่งจะเขียนเส้นใต้ที่พรอพเพอตี้ที่ทำหน้าที่เป็น Key



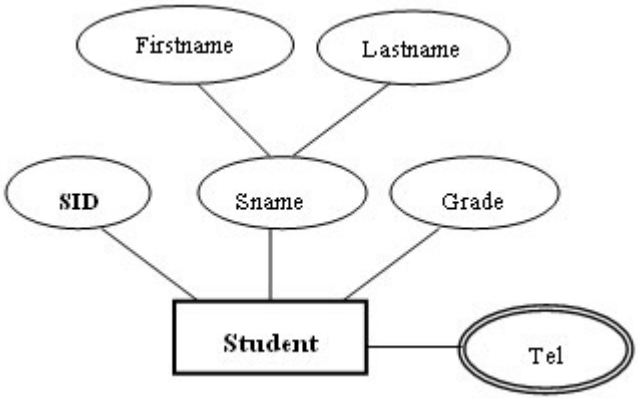
รูปที่ 4.6 เอนทิตี้ Student และ Key พรอพเพอตี้

4.3.4 Single-Valued Property

คือพรอพเพอตี้ ที่ค่าข้อมูลของพรอพเพอตี้นั้น มีอยู่เพียงค่าเดียว เช่น SID,Sname,Grade ดังแสดงในรูปที่ 4.7

4.3.5 Multi-Valued Property

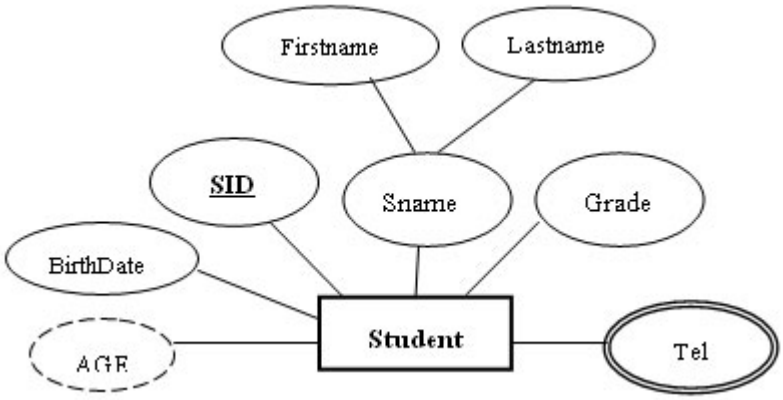
คือพรอพเพอตี้ ที่ค่าข้อมูลของพรอพเพอตี้นั้น มีอยู่ได้หลายค่า เช่น หมายเลขโทรศัพท์ ดังแสดงในรูปที่ 4.7



รูปที่ 4.7 Single-Valued Property และ Multi-Valued Property

4.3.6 Derived Property

คือพรอพเพอตี้ที่ได้จากการนำค่าของพรอพเพอตี้อื่นมาคำนวณ เช่น อายุ ได้จากการ นำวัน เดือน ปีเกิด มาคำนวณ สัญลักษณ์แสดงดังตัวอย่างรูปที่ 4.8 โดยใช้วงรีเป็นเส้นประ

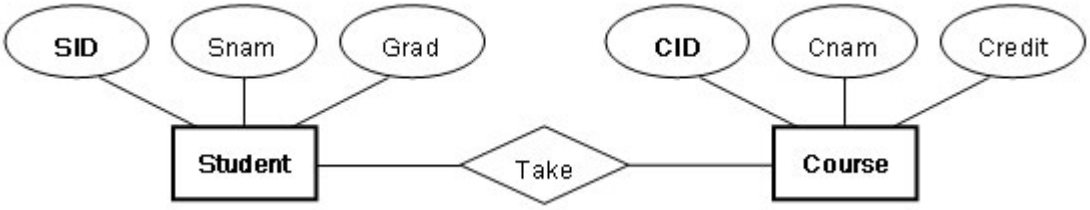


รูปที่ 4.8 Derived Property

จากรูปที่ 4.8 สาเหตุที่แอททริบิวต์ AGE ต้องใช้การคำนวณจากวันเดือนปีเกิด เนื่องจากถ้าเก็บ ค่าอายุโดยตรง จะต้องทำการแก้ไขข้อมูลอายุทุกปี เนื่องจากข้อมูลจะเปลี่ยนไปเรื่อยๆ แต่หากคำนวณ จากวันเดือน ปีเกิด ข้อมูลอายุจะถูกต้องอยู่เสมอ

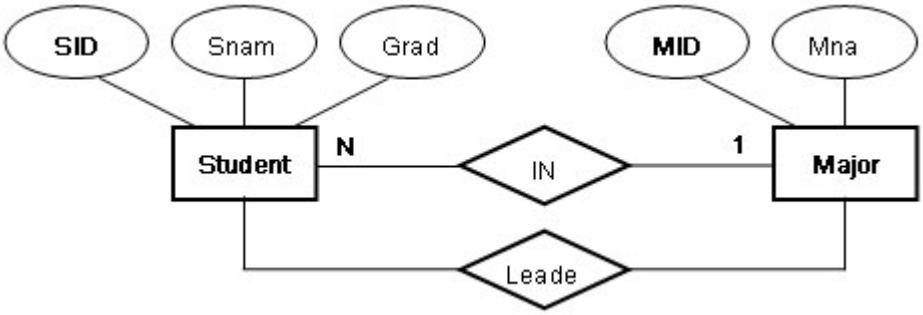
4.4 ความสัมพันธ์ระหว่างเอนทิตี้ (Entity Relationship)

ความสัมพันธ์ของเอนทิตี้คือการอธิบายความสัมพันธ์ระหว่างเอนทิตี้ ว่ามีความสัมพันธ์ของ ข้อมูลกันอย่างไร ใช้สัญลักษณ์สี่เหลี่ยมข้าวหลามตัด และมีเส้นโยง ความสัมพันธ์ระหว่างเอนทิตี้ ดังตัวอย่างรูปที่ 4.9



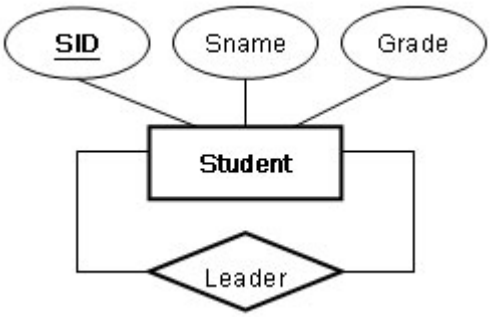
รูปที่ 4.9 ความสัมพันธ์ของข้อมูล

เอนทิตี้ สามารถมีความสัมพันธ์กันได้มากกว่าหนึ่งความสัมพันธ์ เช่นตัวอย่างดังรูปที่ 4.10 เป็นความสัมพันธ์ของเอนทิตี้ Student และ เอนทิตี้ Major ซึ่งมีความสัมพันธ์ 2 แบบ คือ ความสัมพันธ์ Leader และ ความสัมพันธ์ IN ซึ่งบอกว่านักศึกษาเป็นหัวหน้าสาขาวิชาใด และบอกความความ สัมพันธ์ว่านักศึกษาเรียนอยู่ในสาขาวิชาใด



รูปที่ 4.10 ความสัมพันธ์ของเอนทิตี้สามารถมีความสัมพันธ์ได้มากกว่าหนึ่ง

เอนทิตี้ สามารถมีความสัมพันธ์กับตัวเองได้ ซึ่งเรียกว่า ความสัมพันธ์แบบ Recursive ดังตัวอย่างในรูปที่ 4.11 ซึ่งเอนทิตี้ Student สามารถมีความสัมพันธ์ Leader กับตัวเองได้ คือ นักศึกษาเป็นหัวหน้าของนักศึกษา จึงสามารถเขียนความสัมพันธ์ระหว่างนักศึกษากับนักศึกษาได้ โดยใช้เอนทิตี้เดียวคือเอนทิตี้ Student

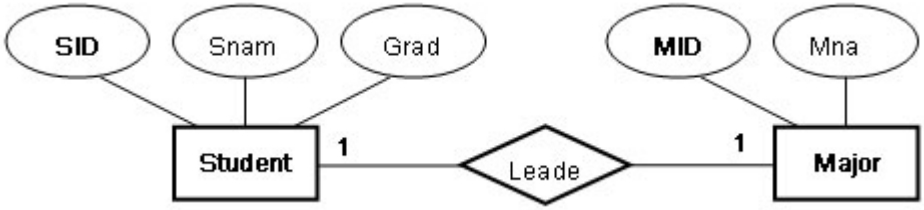


รูปที่ 4.11 ความสัมพันธ์ของเอนทิตี้ แบบ Recursive

ความสัมพันธ์ของเอนทิตี้ สามารถระบุได้ 3 รูปแบบ ดังนี้

4.4.1 ความสัมพันธ์ของเอนทิตี้ แบบหนึ่งต่อหนึ่ง (one to one, 1:1 Relationship)

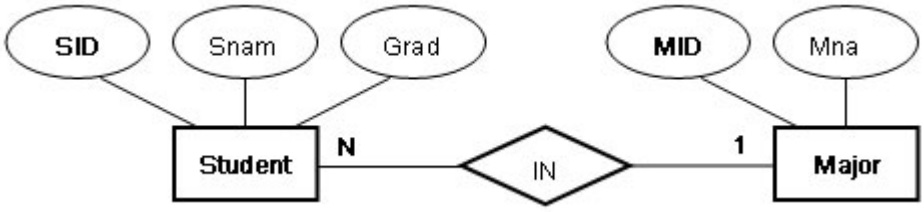
ความสัมพันธ์ของเอนทิตี้ แบบหนึ่งต่อหนึ่ง คือ ความสัมพันธ์ที่ข้อมูลของเอนทิตี้หนึ่ง จะมีความสัมพันธ์กับข้อมูลในเอนทิตี้หนึ่ง เพียง ข้อมูลเดียว ซึ่งใช้ตัวเลข กำกับที่เส้นเพื่อระบุความสัมพันธ์ ดังตัวอย่างรูปที่ 4.12 เป็นความสัมพันธ์ของเอนทิตี้ Student และ เอนทิตี้ Major ซึ่งกำหนดให้แต่ละสาขาวิชามีหัวหน้าห้องได้เพียงคนเดียว และนักศึกษาที่เป็นหัวหน้าห้อง จะเป็นหัวหน้าห้องได้เพียงห้องเดียวเท่านั้น



รูปที่ 4.12 ความสัมพันธ์ของข้อมูลแบบ 1 : 1

4.4.2 ความสัมพันธ์ของเอนทิตี้ แบบหนึ่งต่อหลาย (one to many, 1:N Relationship)

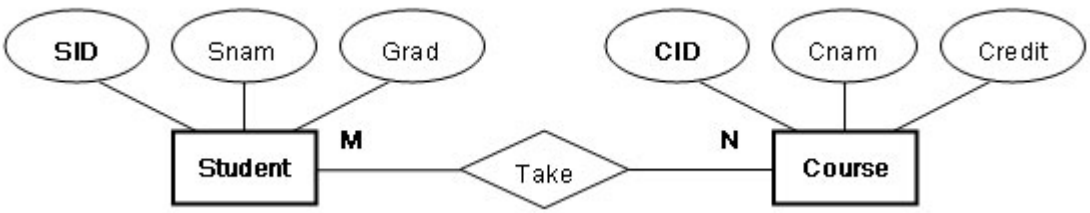
ความสัมพันธ์ของเอนทิตี้แบบหนึ่งต่อหลาย คือ ความสัมพันธ์ที่ข้อมูลของเอนทิตี้ทางด้านหนึ่ง จะมีความสัมพันธ์กับข้อมูลในเอนทิตี้อื่นได้เพียงข้อมูลเดียว แต่เอนทิตี้ที่อยู่ทางด้านกลุ่ม สามารถมีความสัมพันธ์ของข้อมูลกับเอนทิตี้ที่อยู่ทางด้านหนึ่งได้หลายข้อมูล ดังตัวอย่างรูปที่ 4.13 เป็นความสัมพันธ์ของเอนทิตี้ Student และ เอนทิตี้ Major ซึ่งกำหนดให้นักศึกษาแต่ละคน จะต้องสังกัดอยู่ในสาขาวิชาใดสาขาวิชาหนึ่งเท่านั้น แต่ในสาขาวิชาสามารถ มีนักศึกษาสังกัดอยู่ ในสาขานั้นได้มากกว่าหนึ่งคน



รูปที่ 4.13 ความสัมพันธ์ของข้อมูลแบบ 1 : N

4.4.3 ความสัมพันธ์ของเอนทิตี้ แบบหลายต่อหลาย (many to many, M : N relationship)

ความสัมพันธ์ของเอนทิตี้ แบบกลุ่มต่อกลุ่ม คือ ความสัมพันธ์ที่ข้อมูลของเอนทิตี้ทางด้านหนึ่ง จะมีความสัมพันธ์กับข้อมูลในเอนทิตี้อื่นหลายข้อมูล ดังตัวอย่างรูปที่ 4.14 เป็นความสัมพันธ์ของเอนทิตี้ Student และ เอนทิตี้ Course ซึ่งนักศึกษาแต่ละคน สามารถลงทะเบียนเรียนได้หลายวิชา และแต่ละวิชาสามารถมีนักศึกษาลงทะเบียนเรียนได้หลายคน



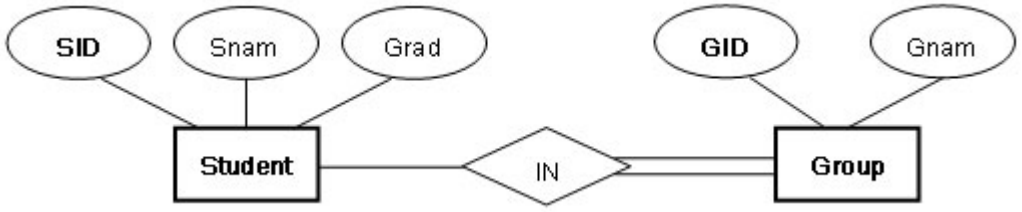
รูปที่ 4.14 ความสัมพันธ์ของข้อมูลแบบ M : N

4.5 ความเกี่ยวข้องกันระหว่างเอนทิตี้กับความสัมพันธ์ (Participation)

Participation คือ ความเกี่ยวข้องกันระหว่างเอนทิตี้กับความสัมพันธ์ ว่ามีลักษณะ ความเกี่ยวข้องกันอย่างไร ซึ่ง มีอยู่ด้วยกัน 2 ลักษณะคือ

1) Total Participation คือ ทุกข้อมูลภายในเอนทิตี้ จะต้องมีความสัมพันธ์กับข้อมูล ในอีกเอนทิตี้ หนึ่ง ตัวอย่างแสดงดังรูปที่ 4.15 แสดงความสัมพันธ์ของนักศึกษากับชมรม ซึ่งกำหนดว่าทุกชมรม จะต้องมีนักศึกษาสังกัดชมรมนั้น ไม่มีไม่ได้ ใช้สัญลักษณ์คือใช้เส้น แสดงความสัมพันธ์ 2 เส้นคู่กัน

2) Partial Participation คือ ข้อมูลภายในเอนทิตี้ อาจจะมีความสัมพันธ์กับข้อมูล ในอีกเอนทิตี้หนึ่ง หรือไม่มีก็ได้ ตัวอย่างแสดงดังรูปที่ 4.15 แสดงความสัมพันธ์ของนักศึกษากับชมรม ซึ่งนักศึกษาบางคนอาจจะไม่สังกัดชมรมใดก็ได้ ซึ่งเขียนแทนโดย ใช้เส้นแสดงความสัมพันธ์ เพียงเส้นเดียว

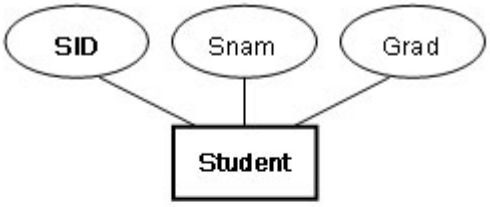


รูปที่ 4.15 ความสัมพันธ์แบบ Total และ Partial Participation

4.6 การแปลง E-R Diagram ไปเป็นตารางเก็บข้อมูล (Mapping ER Diagram)

วิธีการเปลี่ยนจาก E-R Diagram เป็นตารางในการเก็บข้อมูลสามารถสรุปขั้นตอนและตัวอย่างได้ดังนี้

1) Strong Entity สามารถเปลี่ยนเป็นตารางได้เลย โดยมีจำนวนแอททริบิวต์ (คอลัมน์) ของตาราง ตามจำนวนพรอพเพอตี้ของเอนทิตี้ ถ้าพรอพเพอตี้เป็นแบบ Simple Property และให้คีย์ของพรอพเพอตี้ทำหน้าที่ เป็นไพรมารี่คีย์ของตาราง



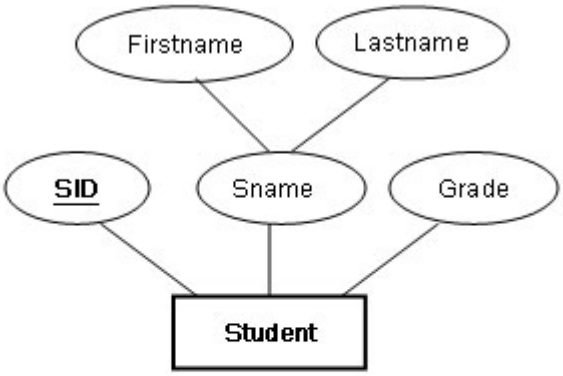
ตาราง Student

SID	Sname	Grade

รูปที่ 4.16 การแปลงจาก Strong Entity เป็นตารางเก็บข้อมูล

จากรูปที่ 4.16 จะเห็นว่าชื่อของพรอพเพอตี้ จะนำมาตั้งเป็นชื่อแอทริบิวต์ของตารางได้เลย และชื่อของเอนทิตี้จะนำมากำหนดเป็นชื่อของตาราง

2) กรณีถ้าพรอพเพอตี้เป็นแบบ Composite Property ให้แยกพรอพเพอตี้ย่อยที่รวมกันเป็น Composite Property ออกเป็นแต่ละแอททริบิวต์ของตารางได้เลย ดังตัวอย่างรูปที่ 4.17

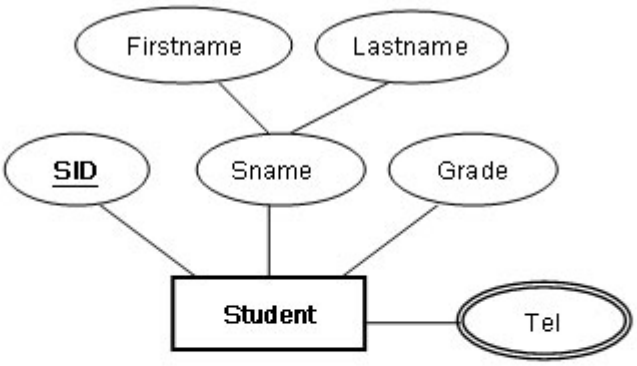


ตาราง Student

SID	Firstname	lastname	Grade

รูปที่ 4.17 การแปลงจาก Strong Entity ที่มี Composite Property เป็นตารางเก็บข้อมูล

3) กรณีถ้าพรอพเพอตี้เป็นแบบ Multivalue Property ให้นำค่าพรอพเพอตี้ที่เป็น Multivalue แยกออกมาสร้างเป็นตารางใหม่ และนำค่าคีย์ของเอนทิตี้เดิม มารวมกับพรอพเพอตี้ที่เป็น Multivalue แล้วกำหนดให้เป็นไพรมารี่คีย์ของตารางที่แยกออกมาสร้างใหม่ ดังตัวอย่างรูปที่ 4.18



ตาราง Student

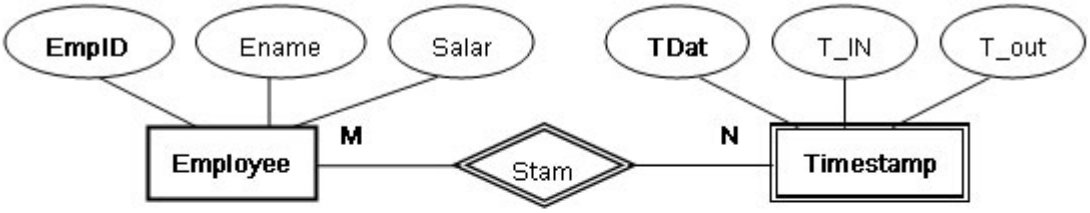
SID	Firstname	lastname	Grade

ตาราง Tel

SID	Tel

รูปที่ 4.18 การแปลง Multivalue Property เป็นตารางเก็บข้อมูล

4) Weak Entity มีหลักการเช่นเดียวกับ Strong Entity แต่ให้นำค่าพรอพเพอตี้ ที่ทำหน้าที่ เป็นคีย์ของ Strong Entity ที่ Weak Entity นั้นมีความสัมพันธ์อยู่ มารวมกับพรอพเพอตี้ของ Weak Entity เอง พร้อมทั้งกำหนดให้เป็นค่าไพรมารี่คีย์ของตารางด้วย ดังตัวอย่างรูปที่ 4.19



ตาราง TimeStamp

EmpID	TDate	T_In	T_out

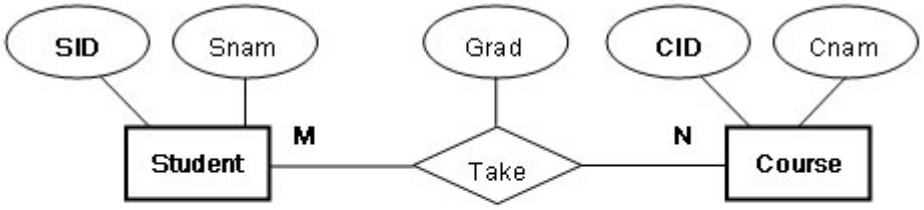
รูปที่ 4.19 การแปลง Weak Entity เป็นตารางเก็บข้อมูล

4.7 การเปลี่ยนจากความสัมพันธ์เป็นตาราง

การเปลี่ยนจากความสัมพันธ์เป็นตาราง มีแนวทางในการทำดังนี้

1) ความสัมพันธ์เป็นแบบกลุ่มต่อกลุ่ม (M : N)

ให้นำความสัมพันธ์นั้นมาสร้างเป็นตารางใหม่ โดยนำไพรมารี่คีย์ของเอนทิตี้ที่มีความสัมพันธ์กับความสัมพันธ์นั้นมากำหนด เป็นพรอพเพอตี้ของตารางใหม่ พร้อมทั้งกำหนดให้ทำหน้าที่เป็นไพรมารี่คีย์ หากความสัมพันธ์นั้นมีพรอพเพอตี้ของความสัมพันธ์ด้วย ก็นำพรอพเพอตี้นั้นมาเป็น แอททริบิวต์ของตารางใหม่ด้วย ตัวอย่างดังรูปที่ 4.20



ตาราง Student

SID	Sname

ตาราง Take

SID	CID	Grade

ตาราง Course

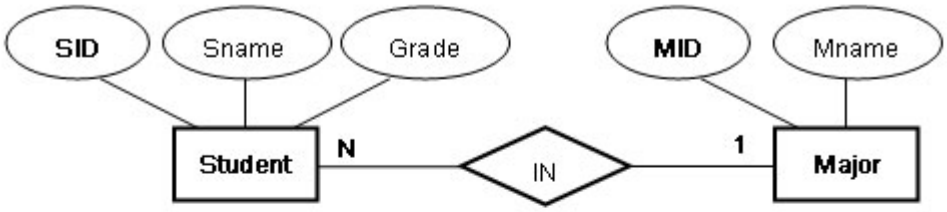
CID	Cname

รูปที่ 4.20 การแปลงจากความสัมพันธ์ เป็นตารางเก็บข้อมูล กรณีความสัมพันธ์แบบกลุ่มต่อกลุ่ม

จากรูปที่ 4.20 จะเห็นว่าตารางที่สร้างมาจากความสัมพันธ์คือตาราง Take ซึ่งได้จากการ นำไพรมารี่คีย์ ของตาราง Student คือ SID และ ไพรมารี่คีย์ของตาราง Course คือ CID มารวมกันแล้วกำหนดให้พรอพเพอตี้ทั้งคู่รวมกันเป็นไพรมารี่คีย์ของตาราง Take และมีแอตทริบิวส์ ของความสัมพันธ์เองก็คือ Grade

2) ความสัมพันธ์เป็นแบบหนึ่งต่อกลุ่ม (1 : N)

ให้นำค่าไพรมารี่คีย์ของเอนทิตี้ที่มีความสัมพันธ์ทางด้าน หนึ่ง มาเป็นแอททริบิวต์ หนึ่ง ของเอนทิตี้ทางด้านกลุ่ม โดยไม่ต้องมีการสร้างตารางของความสัมพันธ์ขึ้นมาใหม่ ตัวอย่างดังรูปที่ 4.21



ตาราง Student

SID	Sname	Grade	MID

ตาราง Major

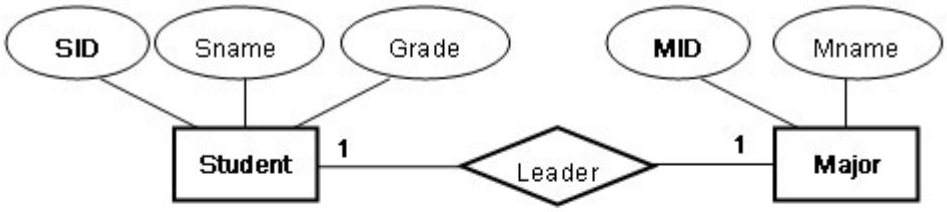
MID	Mname

รูปที่ 4.21 การแปลงจากความสัมพันธ์ เป็นตารางเก็บข้อมูล กรณีความสัมพันธ์แบบหนึ่งต่อกลุ่ม

จากรูปที่ 4.21 จะเห็นว่าที่ตาราง Student จะมีการนำเอาแอททริบิวต์ MID ซึ่งเป็นค่าไพรมารี่คีย์ของตาราง Major มาเป็นแอททริบิวต์หนึ่งของตาราง Student เพื่อใช้สำหรับการ เชื่อมโยงความสัมพันธ์ของทั้งสองตารางนั่นเอง ซึ่งค่า MID ในตาราง Student ก็จะทำหน้าที่เป็น คีย์นอก (Foreign Key) ของตาราง

3) ความสัมพันธ์เป็นแบบหนึ่งต่อหนึ่ง (1 : 1)

ให้นำค่าไพรมารี่คีย์ของเอนทิตี้ที่มีความสัมพันธ์กัน มาเป็นแอททริบิวต์ หนึ่งของอีกเอนทิตี้ ที่มีความสัมพันธ์กัน โดยจะนำจากเอนทิตี้ด้านใดก็ได้ โดยไม่ต้องมีการสร้างตารางของ ความสัมพันธ์ขึ้นมาใหม่ ตัวอย่างดังรูปที่ 4.22



ตาราง Student

SID	Sname	Grade	Leader_of_ MID

ตาราง Major

MID	Mname

รูปที่ 4.22 การแปลงจากความสัมพันธ์ เป็นตารางเก็บข้อมูล กรณีความสัมพันธ์แบบหนึ่งต่อหนึ่ง

จากรูปที่ 4.22 ในกรณีแรกเป็นการนำแอททริบิวต์ MID ซึ่งเป็นไพรมารี่คีย์ของเอนทิตี้ Major มาเป็นแอททริบิวต์หนึ่งของเอนทิตี้ Student โดยตั้งชื่อว่า Leader of MID ส่วนกรณีที่สอง จะนำค่า SID ของเอนทิตี้ Student มาเป็นแอททริบิวต์ของเอนทิตี้ Major โดยตั้งชื่อว่า SID Leader ซึ่งในการใช้งานจริง สามารถที่จะเลือกแบบใดก็ได้

สรุปบทที่ 4

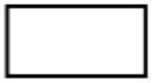
โมเดลข้อมูลเชิงสัมพันธ์ (Entity Relationship Model) เป็นการอธิบายโครงร่างของฐานข้อมูลผ่านทางแผนภาพ E-R Diagram เปรียบเสมือนเป็นพิมพ์เขียวของระบบฐานข้อมูล ซึ่งประกอบด้วยองค์ประกอบสำคัญ คือ เอนทิตี้ (Entity), รีเลชั่นชิป (Relationship) และ แอททริบิวส์ (Attribute)

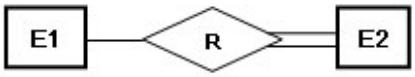
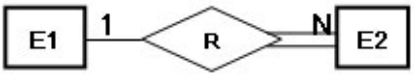
เอนทิตี้ (Entity) คือสิ่งที่มีอยู่จริง จับต้องได้ หรือเป็นจินตภาพที่แสดงความเป็นหนึ่งเดียว เมื่อกล่าวถึงแล้วทุกคนเข้าใจตรงกัน เช่น สินค้า, พนักงาน, นักศึกษา, การสั่งซื้อ เป็นต้น

รีเลชั่นชิป (Relationship) คือความสัมพันธ์ซึ่งเป็นลักษณะการเกี่ยวพันกันระหว่างเอนทิตี้หนึ่งกับตัว มันเองหรือเอนทิตี้อื่น อาจเป็นความสัมพันธ์ที่มากกว่า 2 เอนทิตี้ก็ได้ เช่น แผนกจัดซื้อทำการสั่งซื้อสินค้าหรือวัตถุดิบ

แอททริบิวส์ (Attribute) หรือ พรอพเพอตี้ (Property) คือกลุ่มของค่าความจริงใด ๆ ที่เป็นรายละเอียดของเอนทิตีซึ่งแสดงลักษณะและคุณสมบัติของเอนทิตี้ เช่น รหัสสินค้า, สถานที่เก็บ, ชื่อสินค้า, ราคา นอกจากนั้นยังมีการระบุด้วยว่าแอตตริบิวใดเป็นคีย์กำหนดกฎข้อบังคับต่าง ๆ ของเอนทิตี้และรีเลชั่นชิป

สัญลักษณ์ที่ใช้ในแผนภาพ E-R diagram

สัญลักษณ์	ความหมาย	สัญลักษณ์	ความหมาย
	Entity		Weak Entity
	Relationship		Weak Relationship
	Attribute		Multi value Attribute
	Primary Key Attribute		Derived Attribute

สัญลักษณ์	ความหมาย
	Total Participation E2 in R
	Cardinality Ratio 1:N E1:E2 in R