

Chapter 8

Data Model

(Structured System Approach)



ดร.สันติภูษณ์ นรบิน

เรียบเรียง

อ.วไลลักษณ์ วงษ์รัตน์

Content

- 1) Data Model
- 2) Entity Relationship Diagram : ERD
- 3) ERD Notation
- 4) ERD Component
- 5) Creating ERD

1 – Data Model



Data Model

- ❑ หมายถึง แบบจำลองที่แสดงให้เห็นข้อมูลและความสัมพันธ์ระหว่างข้อมูลที่เกิดขึ้นทั้งหมดในระบบ
- ❑ เครื่องมือที่ใช้ตามแนวทางเชิงโครงสร้างคือ “แผนภาพแสดงความสัมพันธ์ระหว่างข้อมูล (Entity Relationship Diagram : ERD)”
- ❑ เครื่องมือที่ใช้ตามแนวทางเชิงวัตถุคือ “Class Diagram”

Data Model (ต่อ)

- ❑ แบบจำลองข้อมูลใน Analysis Phase ถือเป็น “การออกแบบฐานข้อมูลในระดับแนวคิด (Conceptual Database Design)”
- ❑ กระบวนการออกแบบฐานข้อมูลจะแบ่งเป็น 3 ระดับ
 - ระดับแนวคิด (Conceptual)
 - ระดับตรรกะ (Logical)
 - ระดับกายภาพ (Physical)
- ❑ ERD ที่ได้จากขั้นตอนนี้เป็น Conceptual Model เพื่อนำไปใช้ในการออกแบบในระดับตรรกะต่อไป

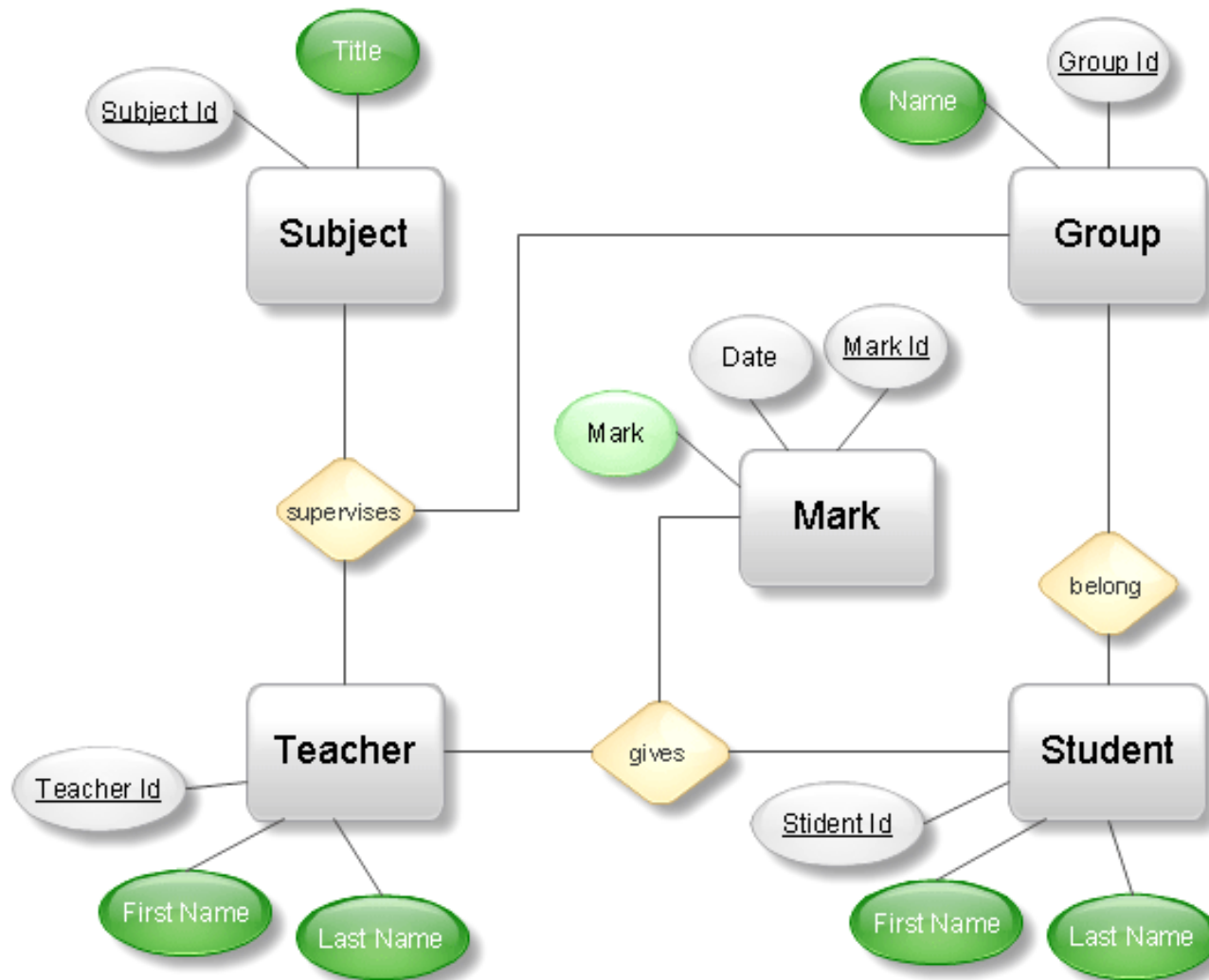
2 – Entity Relationship



แผนภาพแสดงความสัมพันธ์ระหว่างข้อมูล

- ❑ หมายถึง แผนภาพที่ใช้เป็นเครื่องมือสำหรับจำลองข้อมูล
- ❑ ประกอบด้วย
 - Entity (แทนกลุ่มของข้อมูลที่เป็นเรื่องเดียวกัน/เกี่ยวข้องกัน)
 - ความสัมพันธ์ระหว่างข้อมูล (Relationship) ที่เกิดขึ้นทั้งหมดในระบบ

Example : ERD











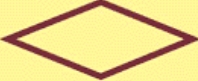
















3 – ERD Notation



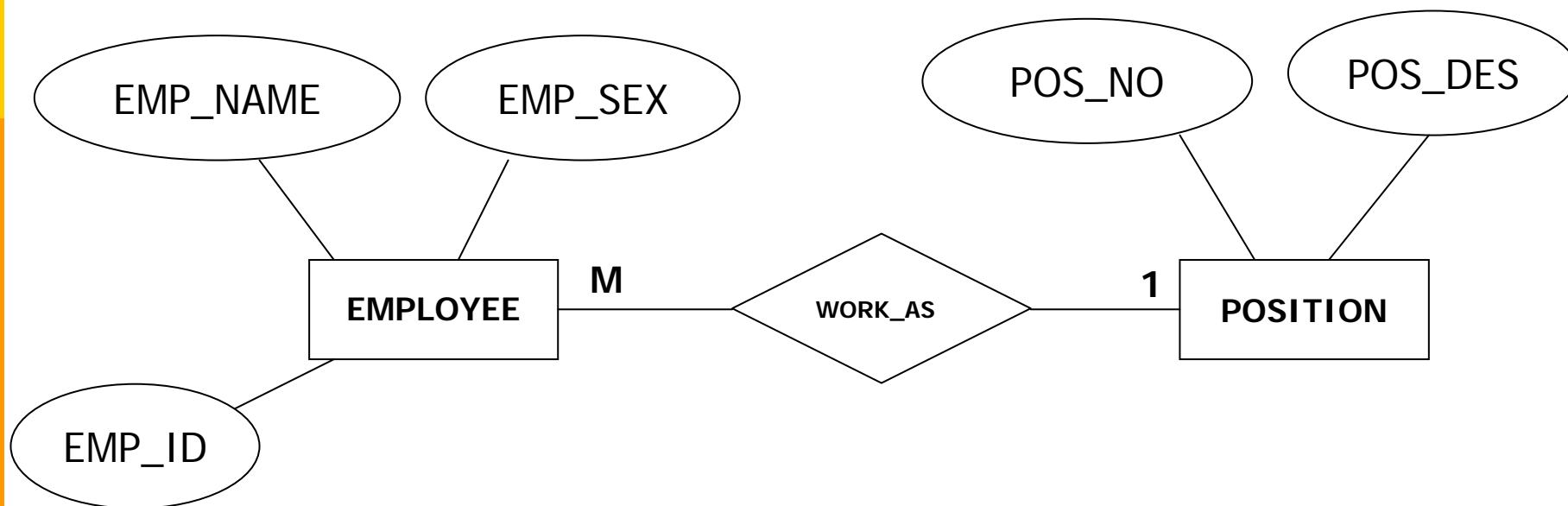
สัญลักษณ์ที่ใช้ใน ERD

- สัญลักษณ์ที่ใช้ในแผนภาพ E-R Diagram ที่ใช้ในการจำลองแบบข้อมูลมีหลายรูปแบบ

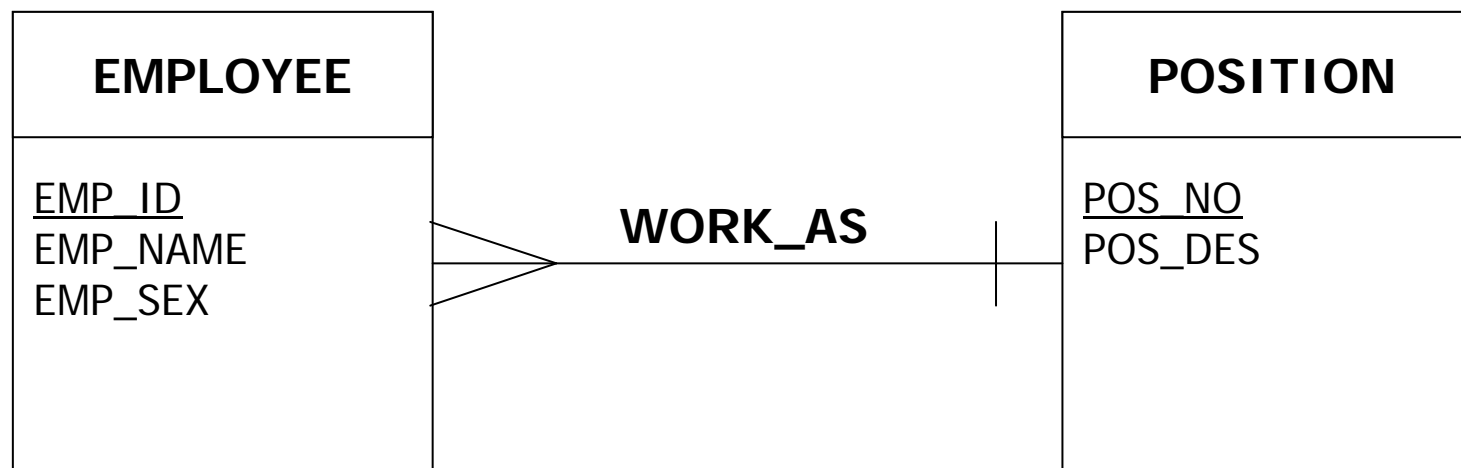
A Comparison of ER Modeling Symbols

	Chen	Crow's Foot	Rein85	IDEF1X
Entity				
Relationship line				
Relationship				
Option symbol				
One (1) symbol	1			
Many (M) symbol	M			
Composite entity				
Weak entity				

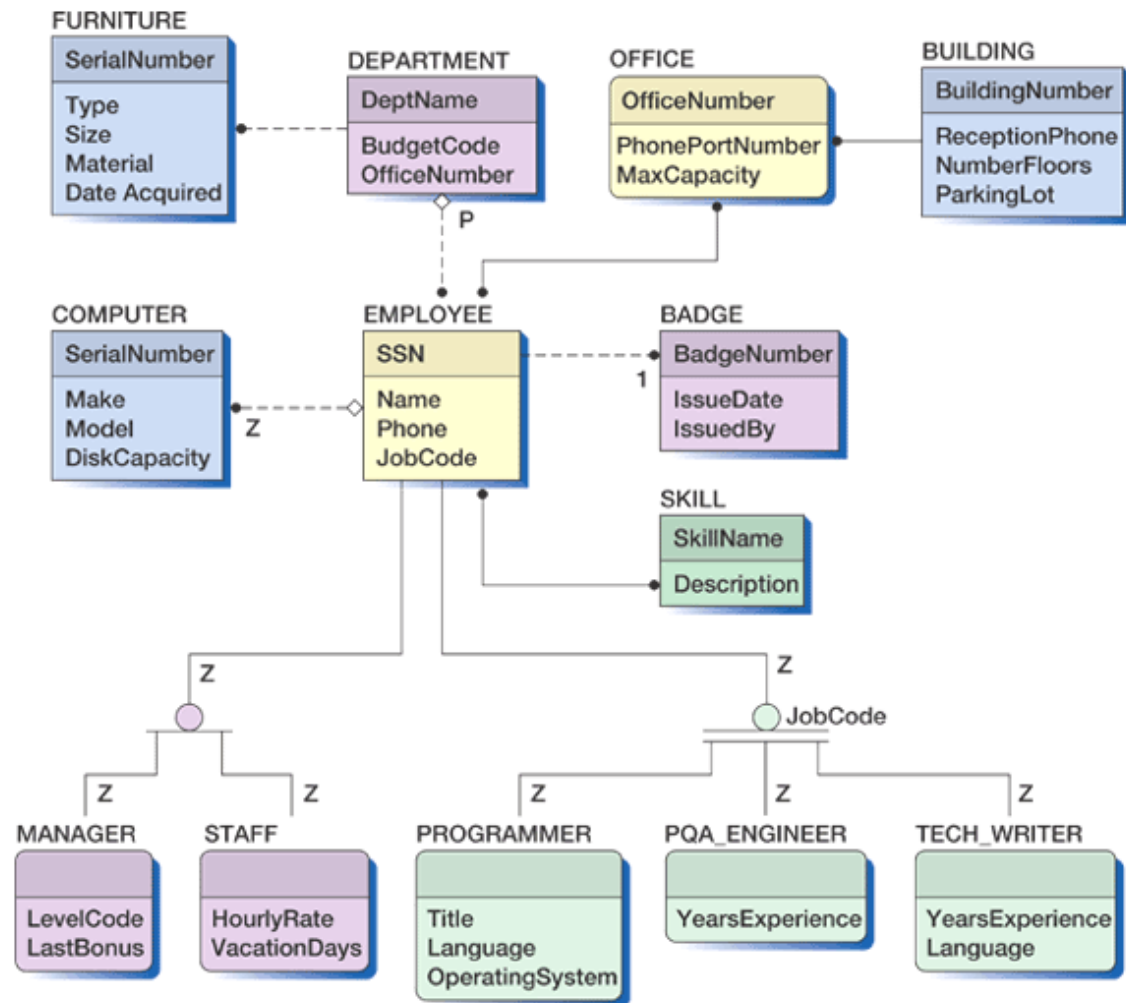
ตัวอย่าง ERD ของรูปแบบ Chen's Model



ตัวอย่าง ERD ของรูปแบบ Crow's Foot Model



ตัวอย่าง ERD ของรูปแบบ IDEF1X



4 - ERD Component



องค์ประกอบของ ERD

□ การสร้างแผนภาพแสดงความสัมพันธ์ระหว่างข้อมูลนั้นมีองค์ประกอบต่างๆ ดังนี้

4.1 Entities

4.2 Attributes

4.3 Relationships

4.4 Associative Entity

4.1 Entities

- ❑ คือ คน สถานที่ วัตถุ เหตุการณ์ หรือแนวคิดที่แวดล้อมผู้ใช้ในระบบที่ต้องการพัฒนา
- ❑ สามารถเป็นได้ทั้งสิ่งที่จับต้องได้และสิ่งที่จับต้องไม่ได้ในระบบ
- ❑ ตัวอย่างเช่น ระบบโรงเรียน ซึ่งจะประกอบด้วย
 - Entity นักเรียน (Student)
 - Entity อาจารย์ (Teacher)
 - Entity หลักสูตร (Course)
 - Entity ห้องเรียน (Room) เป็นต้น

หมวดหมู่ของ Entities

- (1) หมวดบุคคล (Person)
(Employee, Student, Patient, Customer, Department, Division)
- (2) หมวดสถานที่ (Place)
(State, Region, Country, Branch, Building, Room, Campus)
- (3) หมวดเหตุการณ์ (Event)
(Sale, Registration, Order, Invoice, Flight)
- (4) หมวดสิ่งของ (Object)
(Machine, Building, Automobile, Product)
- (5) หมวดของแนวคิด
(Account, Course, Work)

Entity Type

- ❑ คือ กลุ่มของออบเจกต์หรือแนวคิดรวบยอด (Concept) ที่ถูกกำหนดขึ้นเพื่อเป็นตัวแทนข้อมูลในโลกของความเป็นจริง โดยที่ออบเจกต์ที่อยู่ในกลุ่มเดียวกัน จะมีคุณสมบัติ (Property) เหมือนกัน

Entity Instance

- ❑ คือสมาชิกของ Entity Type มีความเป็นเอกลักษณ์ที่สามารถระบุความแตกต่างได้

Views of Stored Data

Object-Oriented	Entity-Relationship	Relational Database
Class	Entity Type	Table
Object	Entity Instance	Row
Attribute	Attribute	Column

ชนิดของ Entity

4.1.1 Regular Entity

4.1.2 Weak Entity

4.1.1 Regular Entity

- ❑ หรือ Strong Entity เป็น Entity ที่ประกอบด้วยสมาชิกที่มีคุณสมบัติ ซึ่งบ่งบอกถึงเอกลักษณ์ของแต่ละสมาชิกนั้น
- ❑ เช่น Entity ประชากร (POPULATION) ซึ่งสมาชิกภายใน Entity นี้ได้แก่ ประชากรแต่ละคนในประเทศไทยที่มีหมายเลขบัตรประชาชนไม่ซ้ำกันเลย เป็นต้น
- ❑ สำหรับสัญลักษณ์ที่ใช้แทน Entity ประเภทนี้ คือรูปสี่เหลี่ยมผืนผ้า โดยมีชื่อของ Entity นั้น อยู่ภายใน ดังรูป

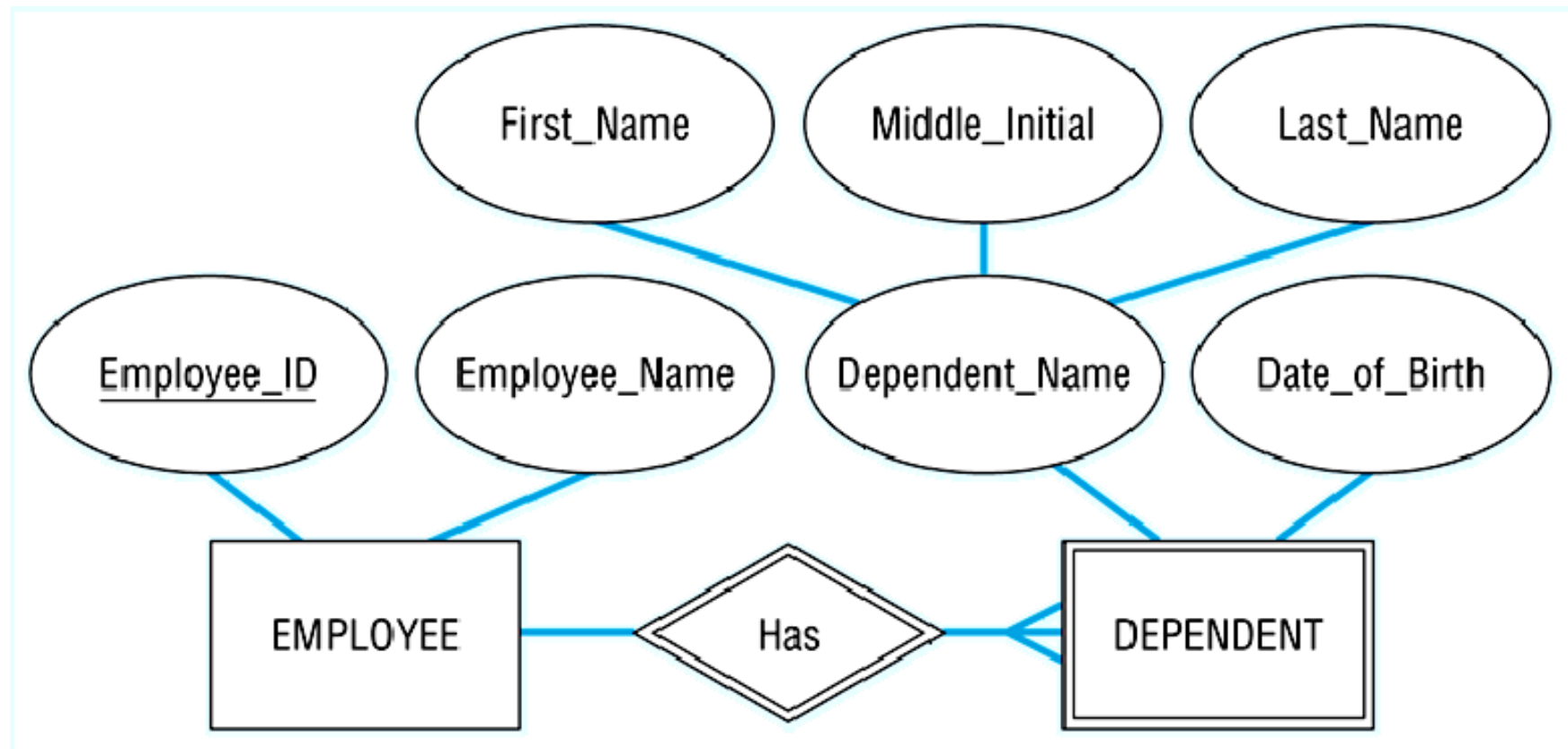


POPULATION

4.1.2 Weak Entity

- ❑ คือ Entity ที่มีลักษณะตรงกันข้ามกับ Regular Entity
- ❑ สมาชิกของ Entity ประเภทนี้ จะสามารถมีคุณสมบัติที่บ่งบอกถึงเอกลักษณ์ของแต่ละสมาชิกได้นั้น จะต้องอาศัยคุณสมบัติใดคุณสมบัติหนึ่งของ Regular Entity มาประกอบกับคุณสมบัติของ Weak Entity เอง

Example of a weak entity



Example of a weak entity (Crow's Foot Model)



4.2 Attributes (Property/Element/Field)

- ❑ หมายถึง คุณสมบัติหรือลักษณะของ Entity หรือ Relationship ที่สนใจ
- ❑ ตัวอย่างที่ 1 Entity “นิสิต” จะมีคุณสมบัติหรือมีลักษณะ (Attributes) ดังนี้
 - รหัสนิสิต
 - ชื่อ-สกุล
 - วันเดือนปีเกิด
 - ที่อยู่
 - คณะ
 - สาขาวิชา

4.2 Attributes (ต่อ)

- ❑ ตัวอย่างที่ 2 Entity “Employee” มี Attribute ที่ทำให้ทราบว่าถ้ามีสิ่งเหล่านี้แล้วจึงเรียกได้ว่า “Employee” ได้แก่
 - Employee_ID
 - Employee_Name
 - Address
 - Skill

ประเภทของ Attribute

4.2.1 Simple Attribute

4.2.2 Composite Attribute

4.2.3 Identifier/Key

4.2.4 Single-Valued Attribute

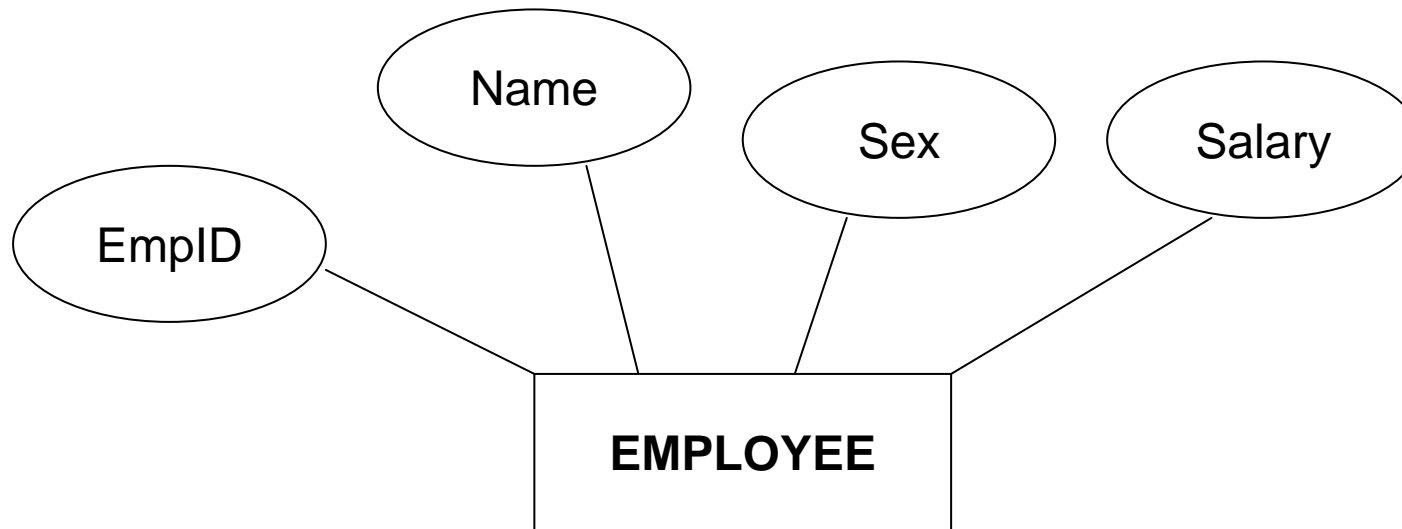
4.2.5 Multi-Valued Attribute

4.2.6 Derived Attribute

4.2.1 Simple Attribute

- ❑ คือ Attribute ที่ค่าภายใน Attribute นั้นไม่สามารถแบ่งย่อยได้อีก เช่น เพศ เงินเดือน อายุ จังหวัด เป็นต้น
- ❑ สำหรับสัญลักษณ์ที่ใช้แทน Attribute ประเภทนี้ ได้แก่ วงรีที่มีเส้นเชื่อมต่อไปยัง Entity ที่เป็นเจ้าของ Attribute นั้น โดยมีชื่อของ Attribute นั้นอยู่ภายใน เช่น Attribute “EmpID”, “NAME”, “SEX” และ “SALARY” ของ Entity “EMPLOYEE”

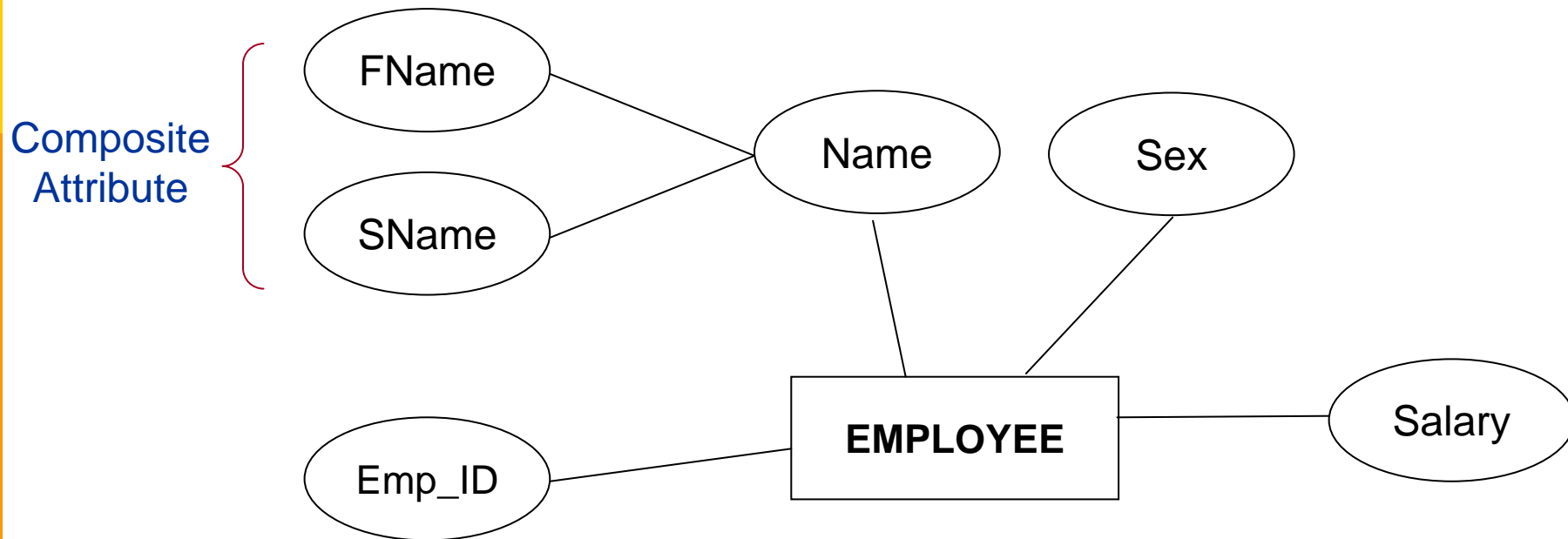
ตัวอย่าง Simple Attribute



4.2.2 Composite Attribute

- ❑ คือ Attribute ที่ค่าภายใน Attribute นั้น สามารถแยกเป็น Attribute ย่อยได้อีก ซึ่งมีลักษณะตรงข้ามกับ Simple Attribute
- ❑ ตัวอย่างที่ 1 Attribute “ชื่อ” ที่สามารถแบ่งย่อยออกเป็น “คำนำหน้าชื่อ”, “ชื่อ” และ “นามสกุล”
- ❑ ตัวอย่างที่ 2 Attribute “ที่อยู่” ที่สามารถแบ่งย่อยออกเป็น “เลขที่บ้าน”, “ซอย” , “ถนน” , “แขวง/ตำบล”, “เขต/อำเภอ”, “จังหวัด” เป็นต้นดังรูป

ตัวอย่าง Composite Attribute



4.2.3 Identifier/Key

- ❑ Identifier หรือ Key คือ Attribute หรือกลุ่มของ Attribute ที่มีค่าในแต่ละ Attribute ของ Entity ไม่ซ้ำกันเลย ซึ่งถูกนำมาใช้กำหนดความเป็นเอกลักษณ์ให้กับแต่ละ Attribute ใน Entity
- ❑ ตัวอย่าง Attribute “EmpID” ของ Entity “Employee” ที่ใช้แทนรหัสประจำตัวพนักงาน
- ❑ Identifier/Key สามารถจำแนกได้ 3 ประเภทดังนี้
 - (1) Candidates Keys
 - (2) Primary Key
 - (3) Foreign Key

(1) Candidates Keys

- ❑ คือ Attribute ใด ๆ หรือกลุ่มของ Attribute ที่ใช้ระบุความแตกต่างของแต่ละเรคอร์ด ดังนั้น Candidate Key จึงต้องเป็นค่าที่ไม่ซ้ำกัน
- ❑ Entity “Employee” มี Candidate Key คือ “EmpID” และ “IDCard”

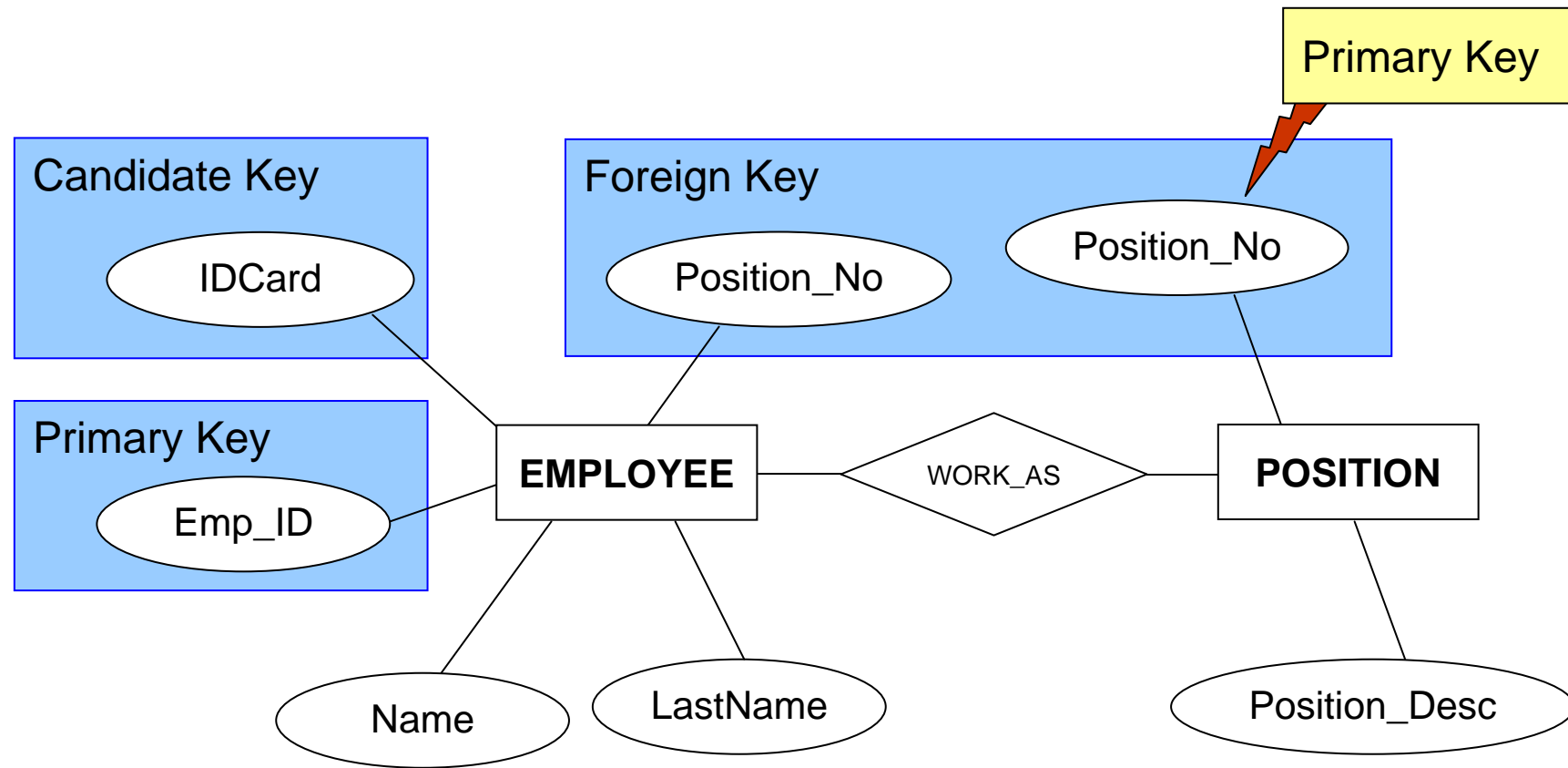
(2) Primary Key

- ❑ คือ Candidate Key ที่ถูกเลือกให้เป็น Key หลักที่มีค่าของสมาชิกใน Attribute ไม่ซ้ำกันเลย
- ❑ การที่เลือก Key ที่มีค่าไม่ซ้ำกันเลยมาเป็น Primary Key เพื่อให้ Primary Key นี้สามารถไประบุค่าในอีก Attribute อื่นเพื่อประโยชน์ในการค้นหาข้อมูลได้โดยไม่เกิดข้อมูลซ้ำซ้อนกัน

(3) Foreign Key

- ❑ คือ Primary Key ของ Entity หนึ่งที่สามารถระบุค่าสมาชิกของอีก Entity หนึ่งที่มีความสัมพันธ์กันได้ แสดงดังรูป

Example : Candidates Keys, Primary Key, Foreign Key



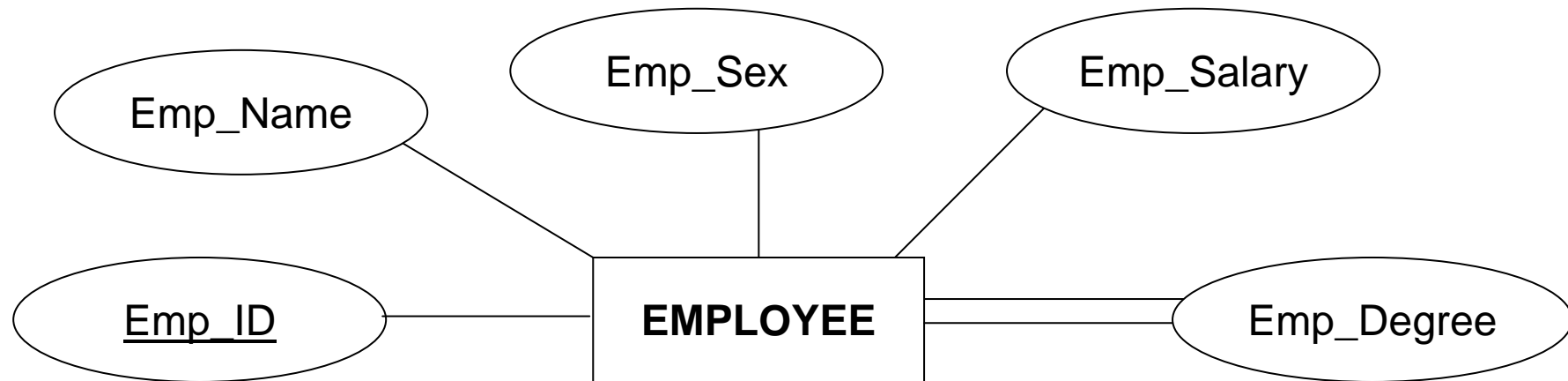
4.2.4 Single-Valued Attribute

- ❑ คือ Attribute ที่มีค่าของข้อมูลภายใต้ Attribute ใด Attribute หนึ่งเพียงค่าเดียว
- ❑ เช่น Attribute “Salary” ซึ่งที่ใช้เก็บเงินเดือนของพนักงาน และพนักงานแต่ละคนจะมีเงินเดือนเพียงค่าเดียว

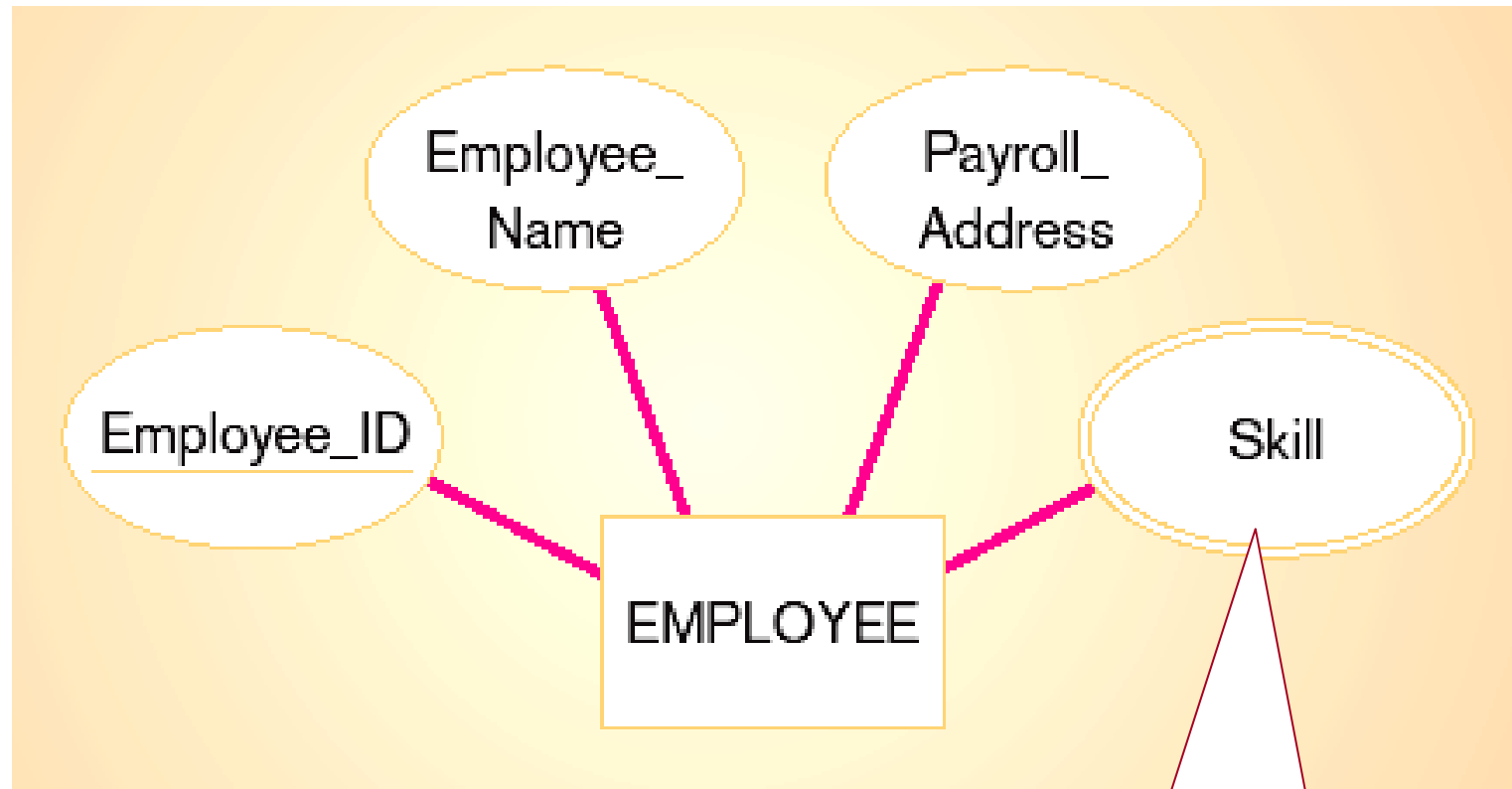
4.2.5 Multi-Valued Attribute

- ❑ คือ Attribute ที่มีค่าของข้อมูลได้หลายค่าภายใต้ค่าของ Attribute ใด Attribute หนึ่ง
- ❑ เช่น Attribute “DEGREE” ที่ใช้ระบุระดับการศึกษาของพนักงานแต่ละคน ซึ่งพนักงานแต่ละคน จะมีระดับการศึกษาได้หลายระดับ
- ❑ กลุ่มข้อมูลที่สามารถมีหลายค่าได้นี้ (Multi-Valued Attribute) เรียกอีกอย่างหนึ่งว่า “Repeating Group”
- ❑ สัญลักษณ์ที่ใช้แทน Attribute ประเภทนี้ จะใช้เส้น 2 เส้นเชื่อมระหว่างรูปภาพของ Attribute กับ Entity ดังรูป

Example : Multi-Valued Attribute



Example : Multi-Valued Attribute



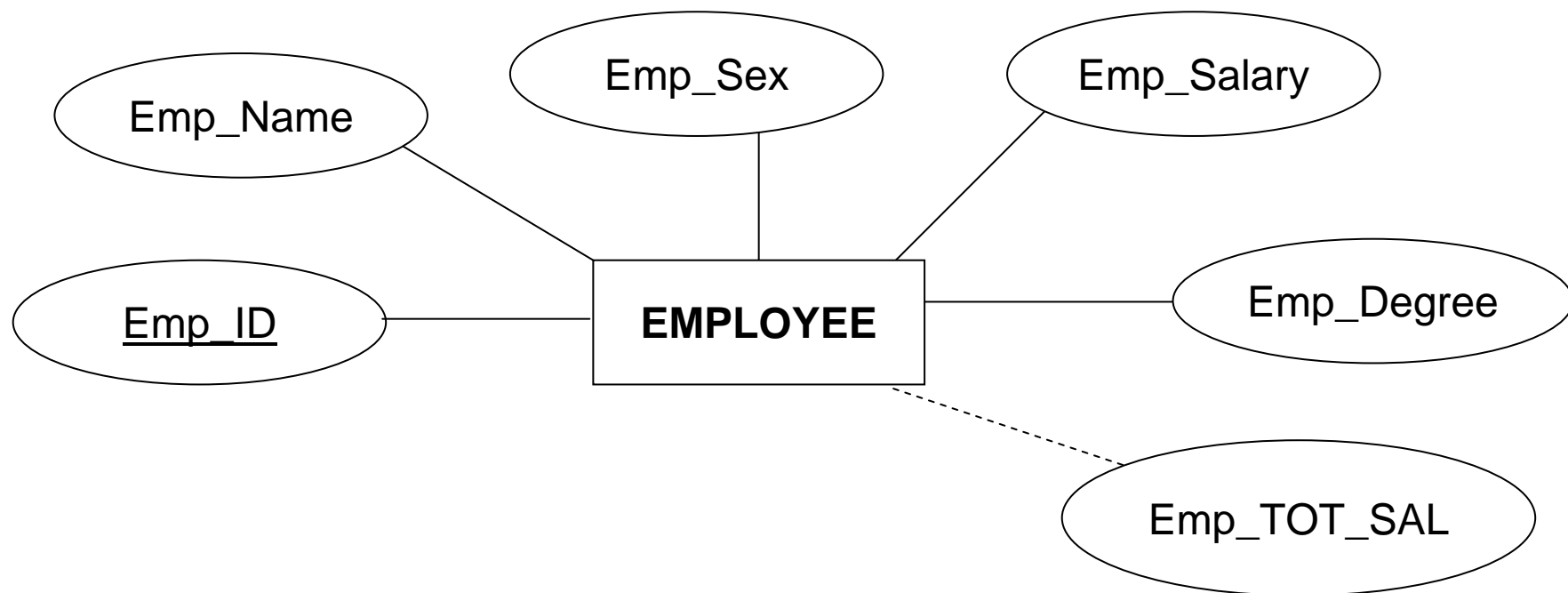
Multi-valued attribute...
an employee may have
more than one skill.

4.2.6 Derived Attribute

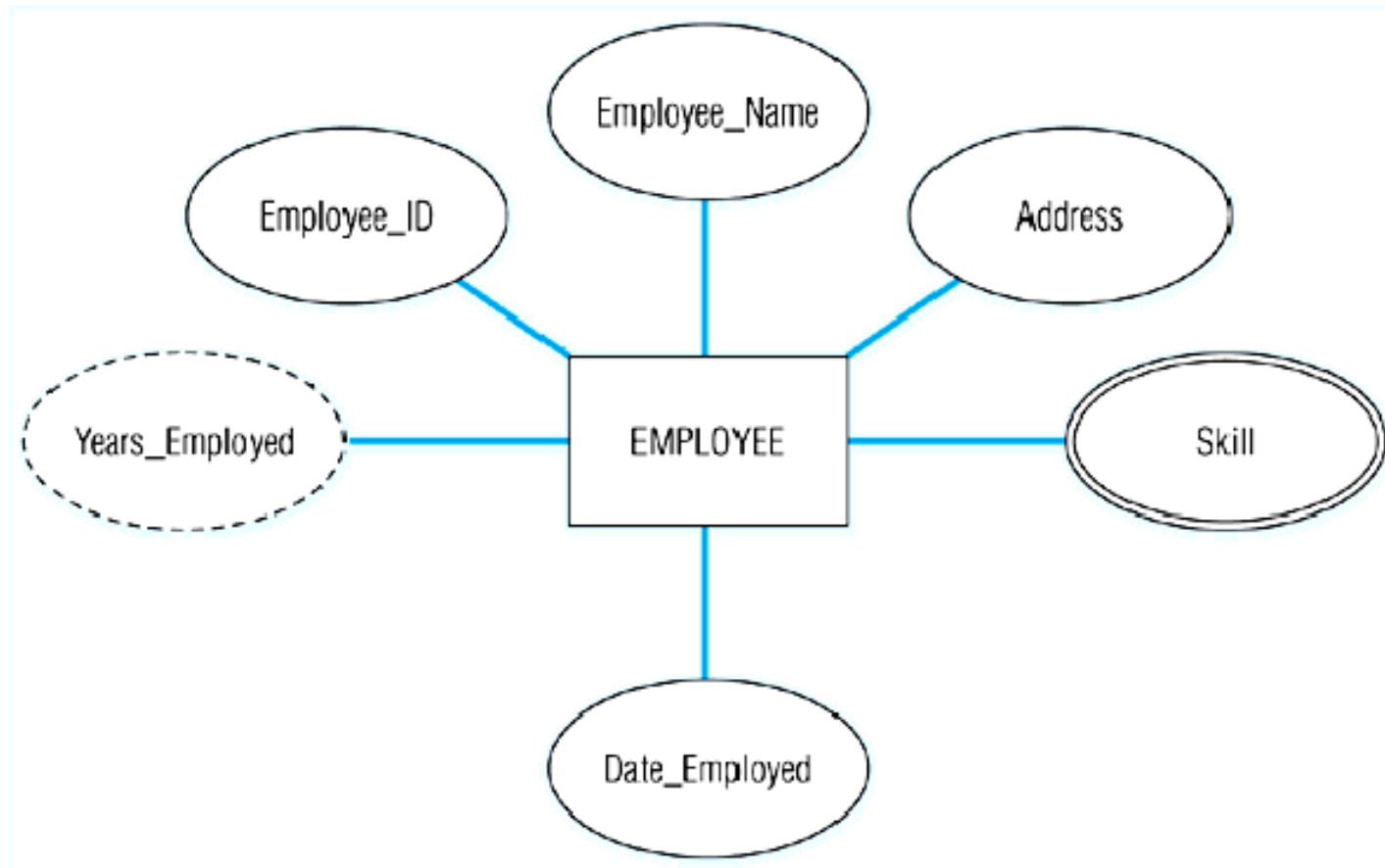
- ❑ คือ Attribute ที่ค่าของข้อมูลได้มาจากการนำเอาค่าของ Attribute อื่นมาทำการคำนวณ ซึ่งค่าของ Attribute ประเภทนี้จะต้องเปลี่ยนแปลงทุกครั้ง เมื่อมีการเปลี่ยนแปลงค่าของ Attribute ที่ถูกนำค่ามาคำนวณ
- ❑ เช่น Attribute “Emp_TOT_SAL” ของ Entity “Employee” ที่ใช้เก็บเงินเดือนทั้งหมดของพนักงานแต่ละคนเพื่อนำไปคำนวณภาษี ซึ่งได้มาจากผลรวมของค่าใน Attribute “INCOME” ของ Entity “MTHLY_SALARY” ซึ่งเป็นเงินเดือนที่พนักงานแต่ละคนได้รับในแต่ละเดือน

Example : Derived Attribute

- ❑ สัญลักษณ์ที่ใช้แทน Attribute ประเภทนี้จะใช้สัญลักษณ์เส้นประเชื่อมระหว่าง Entity และ Attribute ดังรูป



Entity with a multivalued attribute (Skill) and derived attribute (Years_Employed)



4.3 Relationship

- ❑ คือ ความสัมพันธ์ระหว่าง Entity 1 Entity ขึ้นไป โดยความสัมพันธ์ที่แท้จริงนั้นเกิดขึ้นจากสมาชิกของ Entity ที่สัมพันธ์กัน
- ❑ องค์ประกอบของความสัมพันธ์ใน ERD ประกอบด้วย

4.3.1 Degree of a Relationships

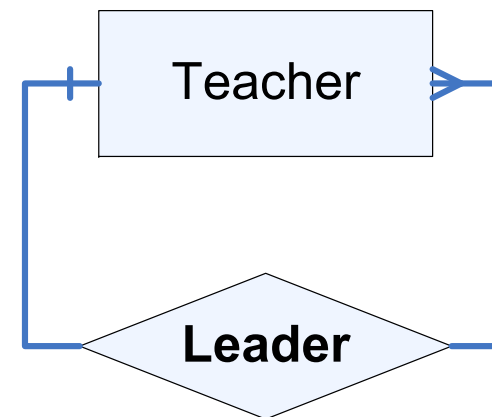
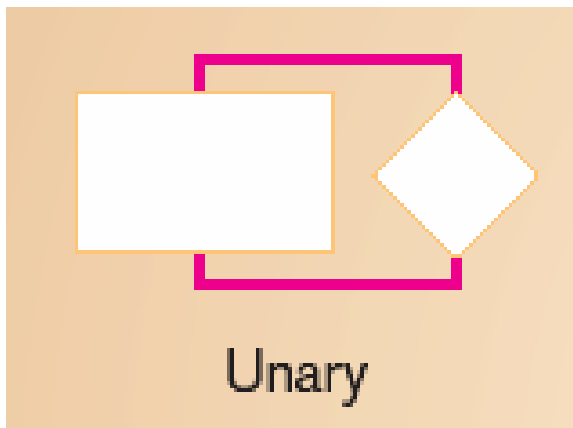
4.3.2 Connectivity of a Relationships

4.3.1 Degree of Relationships

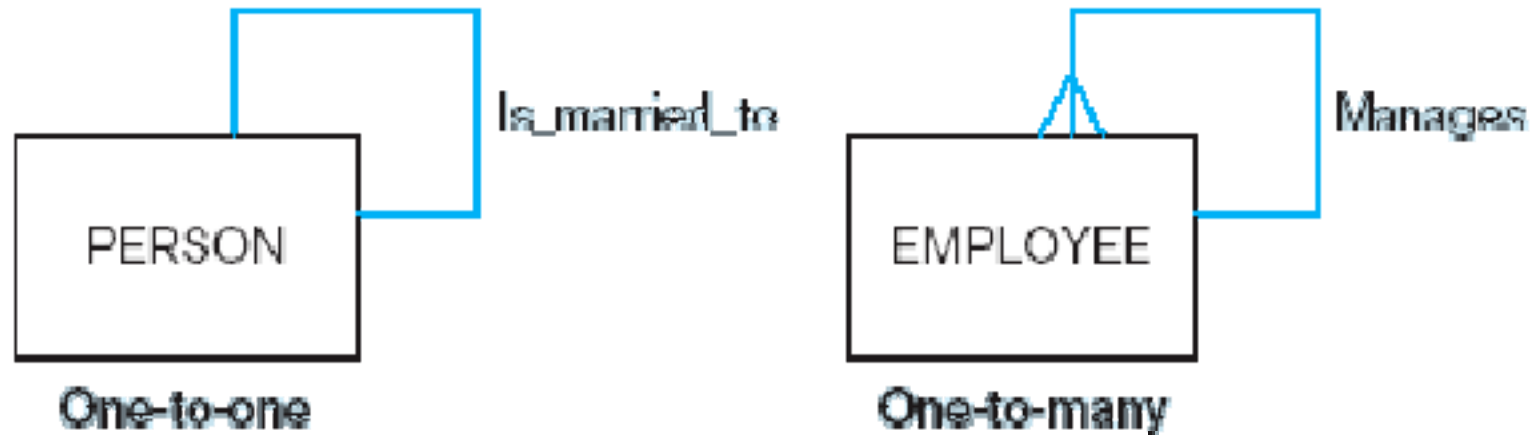
- ❑ คือจำนวน Entity ที่มีความสัมพันธ์กัน
- ❑ แบ่งเป็น 3 ระดับ
 - Unary Relationship
 - Binary Relationship
 - Ternary Relationship

Unary Relationship

- เป็นความสัมพันธ์ที่เกิดขึ้นระหว่างสมาชิกภายใน Entity เดียวกัน

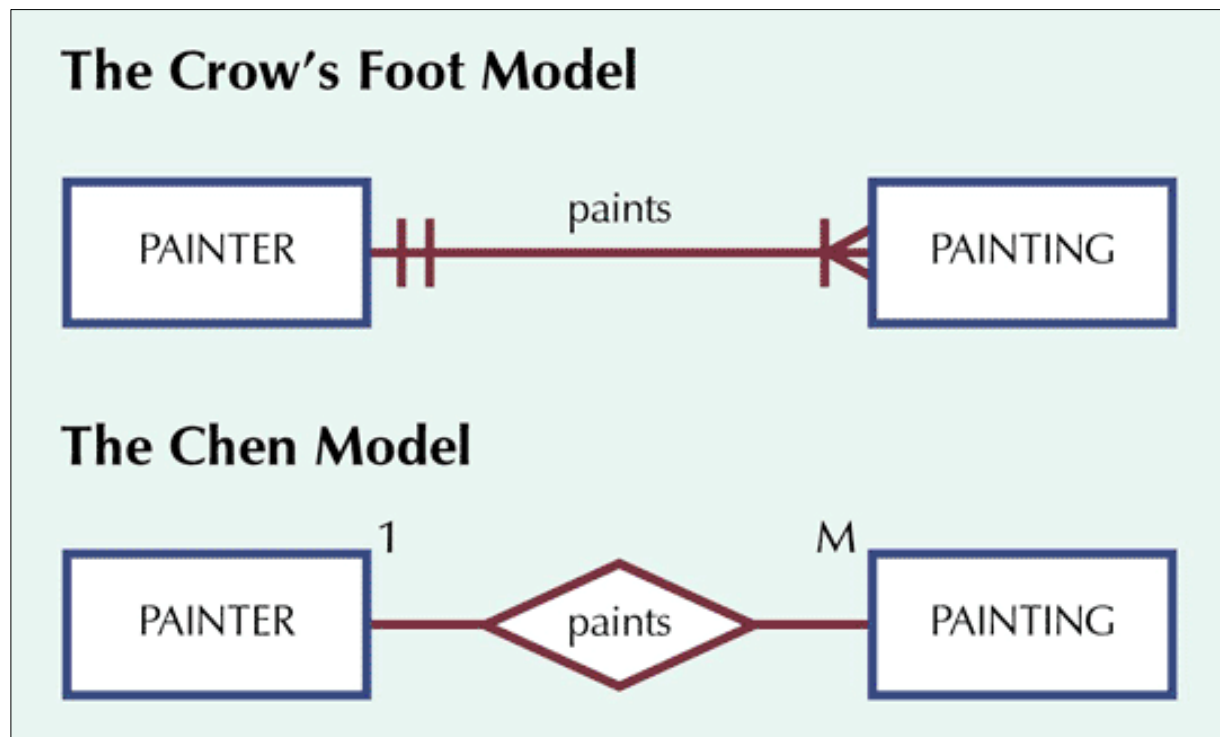


Examples of Unary relationships



Binary Relationship

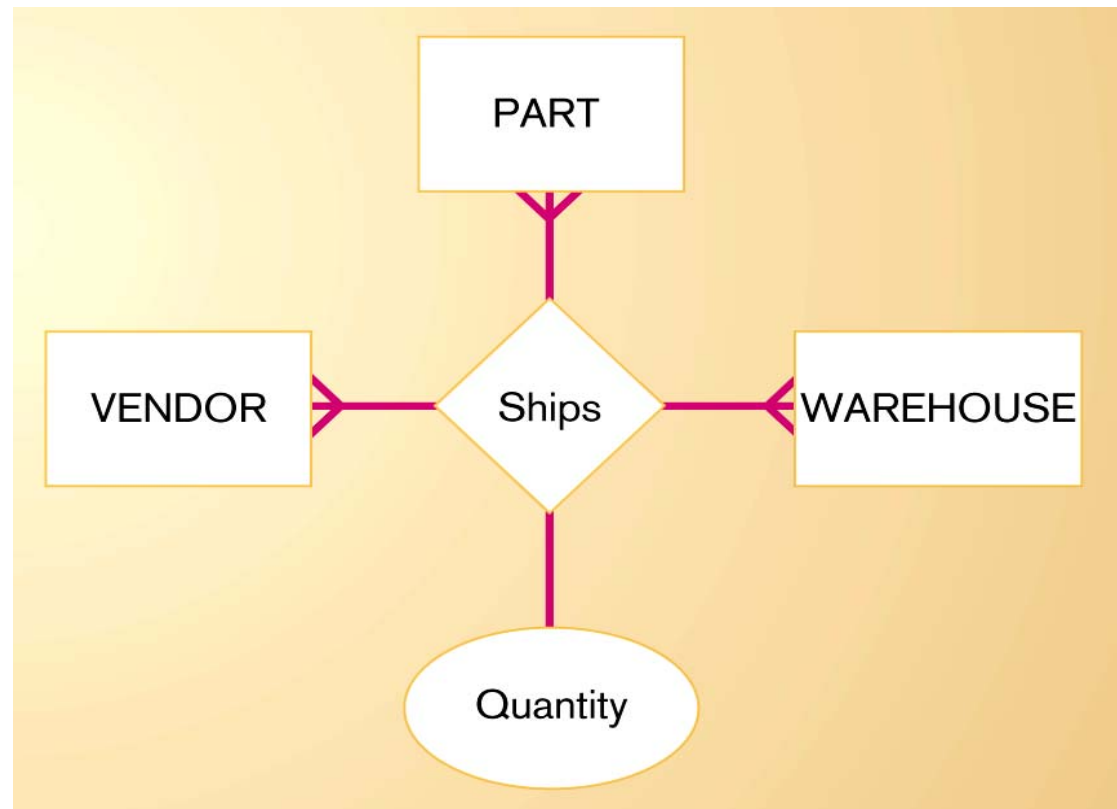
- เป็นความสัมพันธ์ที่เกิดขึ้นระหว่าง 2 Entity



Ternary Relationship

- ❑ เป็นความสัมพันธ์ระหว่าง Entity 3 Entity เกิดขึ้นเนื่องจาก Binary Relationship ไม่สามารถอธิบายความสัมพันธ์ของ Entity ในงานจริงได้อย่างสมบูรณ์
- ❑ กรณีที่มีความสัมพันธ์มากกว่า 3 Entity ขึ้นไปจะเรียกว่า “N-ary Relationship”

Example of Ternary relationships

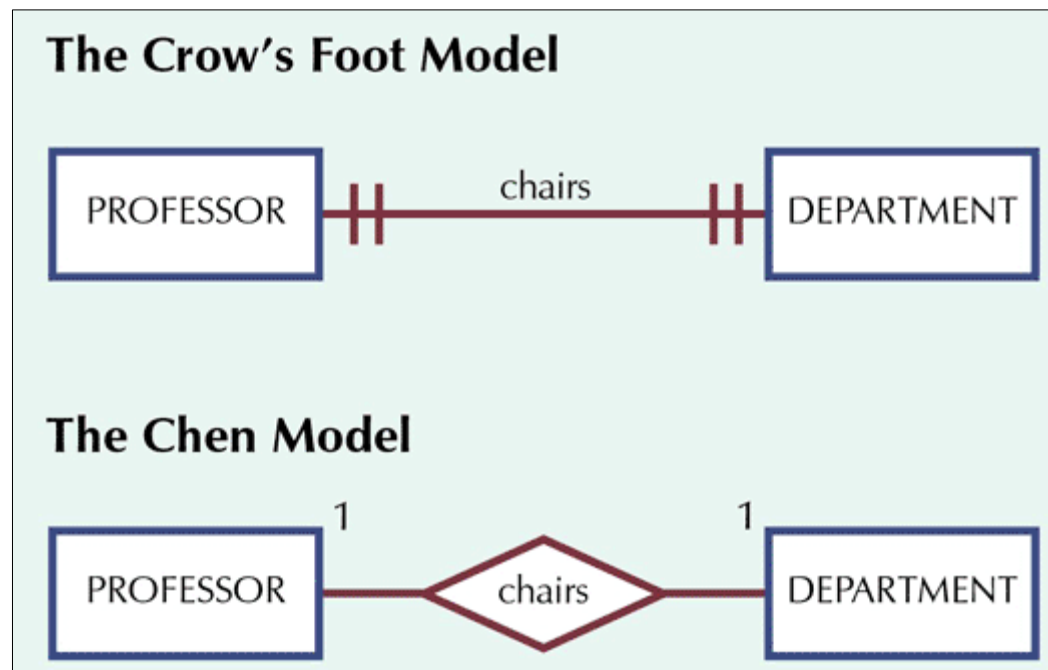


4.3.2 Connectivity of a Relationships

- ❑ เป็นการจำแนกประเภทความสัมพันธ์ของ Entity ตามจำนวนสมาชิกของแต่ละ Entity ที่สัมพันธ์กัน
- ❑ แบ่งเป็น 3 ชนิด
 - One-to-One Relationship
 - One-to-Many Relationship
 - Many-to-Many Relationship

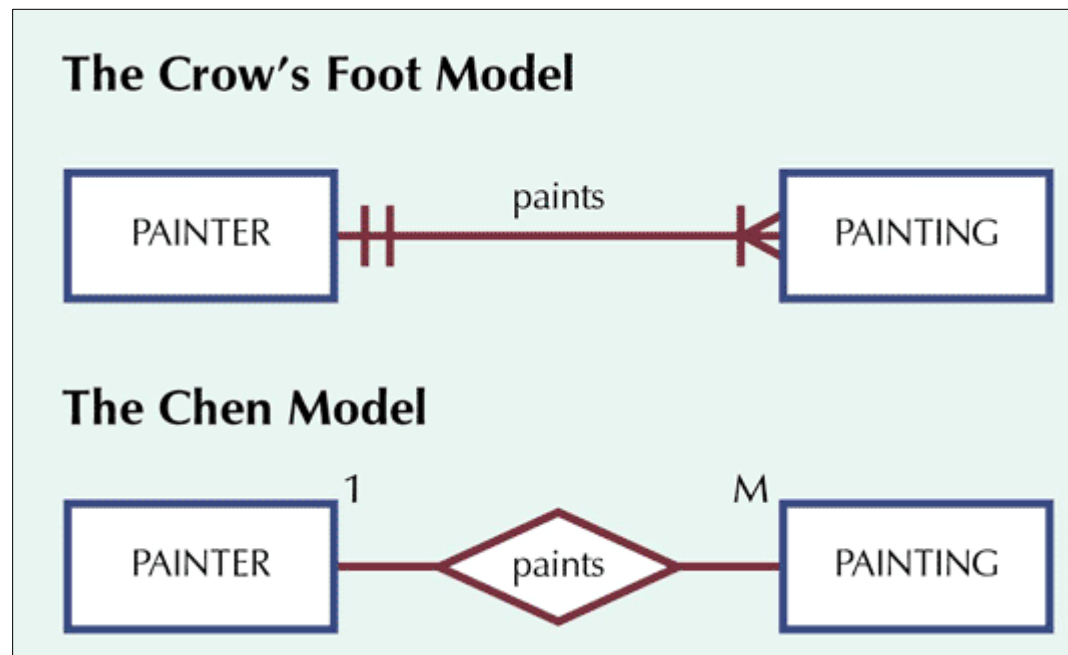
One-to-One Relationship

- เป็นความสัมพันธ์ที่แต่ละสมาชิกของ Entity หนึ่ง มีความสัมพันธ์กับสมาชิกของอีก Entity หนึ่งได้เพียงสมาชิกเดียว



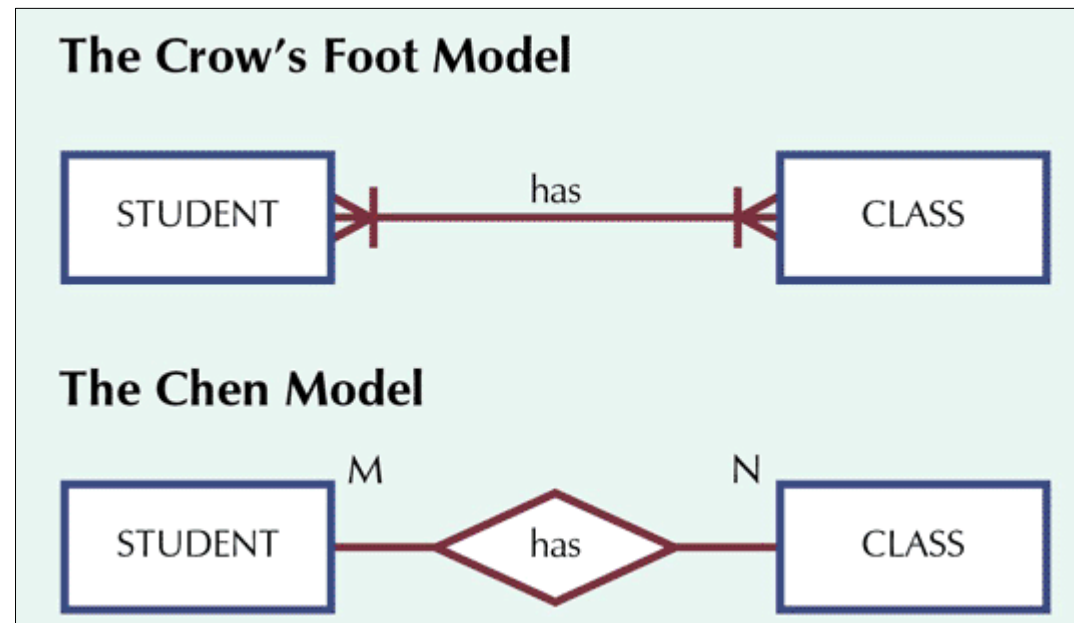
One-to-Many Relationship

- เป็นความสัมพันธ์ที่แต่ละสมาชิกของ Entity หนึ่ง มีความสัมพันธ์กับสมาชิกของอีก Entity หนึ่งได้มากกว่า 1 จำนวนสมาชิก



Many-to-Many Relationship

- เป็นความสัมพันธ์ที่ Entity หนึ่ง ซึ่งมีสมาชิกมากกว่า 1 จำนวน ไปมีความสัมพันธ์กับสมาชิกของอีก Entity หนึ่งได้มากกว่า 1 จำนวน

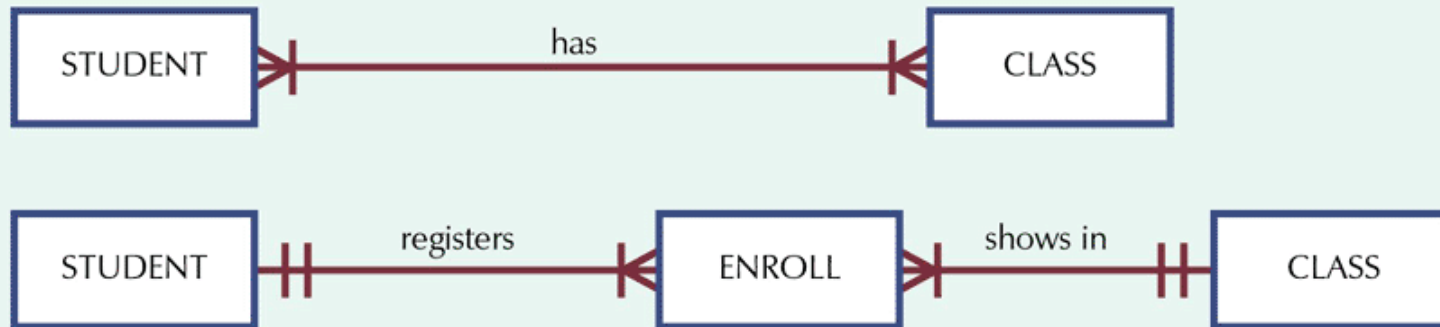


4.4 Associative Entity

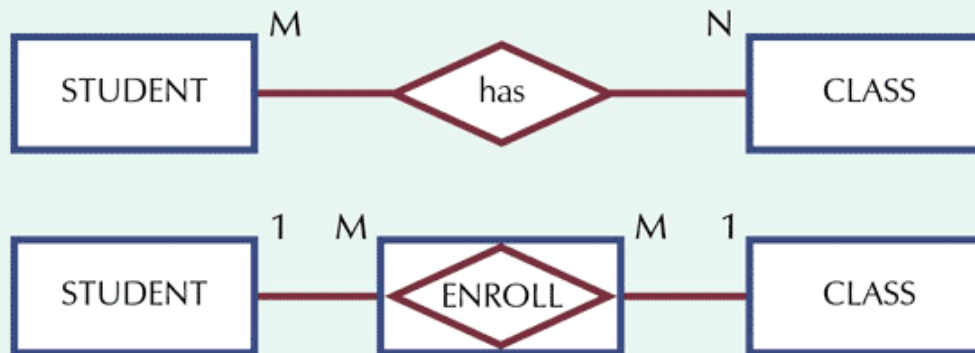
- ❑ กรณีที่ Relationship เป็นแบบ Many-to-Many มักจะมี Entity อีกหนึ่ง Entity แอบแฝงอยู่ ซึ่ง Entity นั้นคือ Relationship นั่นเอง
- ❑ เนื่องจากความสัมพันธ์แบบ Many-to-Many ทำให้ Relationship ที่เกิดขึ้นมี Attribute ใหม่เกิดขึ้นเพื่อแสดงความ เป็นเอกลักษณ์ของความสัมพันธ์เองด้วย

Example of Associative Entity

The Crow's Foot Model



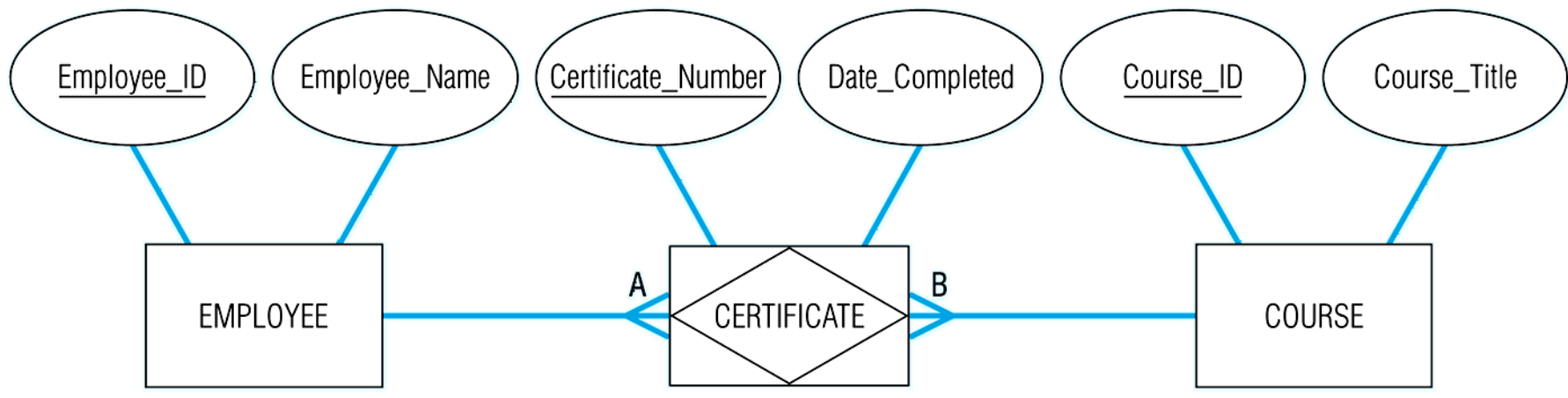
The Chen Model



การแปลง Relationship เป็น Associative Entity

- ❑ นอกจากจะมี Attribute ใหม่แล้ว จะต้องประกอบด้วย Attribute ที่เป็น Primary Key ของ Entity ที่ทำให้เกิดความสัมพันธ์นั้นด้วย

An associative entity



5 - Creating ERD



วิธีสร้าง ERD

□ แบ่งเป็น 2 แนวทาง

5.1 Top-down Approach

5.2 Bottom-up Approach

5.1 Top-down Approach

- ❑ เป็นการสร้าง ERD โดยอาศัยความเข้าใจในกระบวนการทำงานของระบบเป็นอย่างดี แล้วนำมาประกอบกับข้อมูล Requirements ที่เก็บรวบรวมได้
- ❑ ขั้นตอน
 - กำหนด Entity ทั้งหมดของระบบ
 - สร้าง Relationship ระหว่าง Entity
 - กำหนด Attribute และ Primary Key

Example : การสร้าง ERD ด้วยวิธี Top-down Approach

- ❑ มหาวิทยาลัยแห่งหนึ่ง เปิดสอนหลักสูตรปริญญาตรีหลายคณะ
- ❑ แต่ละคณะเปิดสอนหลายรายวิชา ซึ่งทำการสอนโดยอาจารย์ที่มีคุณภาพ
- ❑ การเปิดสอนแต่ละรายวิชานั้น จะต้องทำเรื่องขอเสนอเปิดสอนรายวิชานั้น ๆ
- ❑ แต่ละรายวิชาจะสามารถเปิดสอนได้ต่อเมื่อมีนักศึกษาสมัครลงทะเบียนในรายวิชานั้นอย่างน้อย 20 คน
- ❑ อาจารย์ 1 ท่านสามารถสอนได้หลายวิชา
- ❑ และห้องเรียนแต่ละห้องสามารถใช้สอนวิชาต่าง ๆ ได้หลายวิชา

ขั้นตอนที่ 1 กำหนด Entity ทั้งหมดของระบบ

- ❑ คณะ (Faculty)
- ❑ รายวิชา (Course)
- ❑ อาจารย์ (Teacher)
- ❑ นักศึกษา (Student)
- ❑ ห้องเรียน (Room)

ขั้นตอนที่ 2 สร้าง Relationship ระหว่าง Entity

- ❑ ทางคณะ (Faculty) จะต้องเสนอเรื่อง (Offer) เพื่อขอเปิดวิชาเรียน (Course)
- ❑ แต่ละรายวิชา (Course) จะต้องมียาจารย์ (Teacher) เป็นผู้ทำการสอน (Teach)
- ❑ ห้องเรียน (Room) เปิด (Open) ทำการเรียนการสอนทุกรายวิชา (Course)
- ❑ รายวิชาใด ๆ (Course) จะสามารถเปิดสอนได้ต้องมี (Have) นักศึกษา (Student) ลงทะเบียนเรียนอย่างน้อย 20 คน



คณะ (Faculty) จะต้องเสนอเรื่อง (Offer) เพื่อขอเปิดวิชาเรียน (Course)

แต่ละรายวิชา (Course) จะต้องมามีอาจารย์ (Teacher) เป็นผู้ทำการสอน (Teach)

ห้องเรียน (Room) เปิด (Open) ทำการเรียนการสอนทุกรายวิชา (Course)

รายวิชาใด ๆ (Course) จะสามารถเปิดสอนได้ต้องมี (Have)
นักศึกษา (Student) ลงทะเบียนเรียนอย่างน้อย 20 คน

กำหนดเงื่อนไข (Constraints) ความสัมพันธ์ระหว่าง Entity

- ❑ แต่ละคณะสามารถเสนออธิบดีเพื่อขอเปิดสอนได้หลายรายวิชา แต่รายวิชา 1 รายวิชา จะต้องอยู่ในคณะเดียวกันเท่านั้น กล่าวคือ ชื่อรายวิชาจะซ้ำกับคณะอื่นไม่ได้
- ❑ อาจารย์ 1 ท่าน สามารถสอนได้หลายวิชา และ 1 รายวิชาสามารถมีอาจารย์สอนได้หลายท่านเช่นกัน
- ❑ ห้องเรียน 1 ห้องสามารถเปิดทำการเรียนการสอนได้หลายรายวิชา และ 1 รายวิชาสามารถใช้ห้องเรียนหลายห้องเพื่อทำการเรียนการสอนได้เช่นเดียวกัน
- ❑ รายวิชา 1 รายวิชาจะสามารถเปิดได้ จะต้องมื่อนักศึกษาลงทะเบียนอย่างน้อย 20 คน และนักเรียน 1 คนสามารถมีวิชาเรียนได้หลายวิชา

แต่ละคณะสามารถเสนออธิบดีเพื่อขอเปิดสอนได้หลายรายวิชา แต่รายวิชา 1 รายวิชา
จะต้องอยู่ในคณะเดียวกันเท่านั้น กล่าวคือ ชื่อรายวิชาจะซ้ำกับคณะอื่นไม่ได้



อาจารย์ 1 ท่าน สามารถสอนได้หลายวิชา และ 1 รายวิชาสามารถ
มีอาจารย์สอนได้หลายท่านเช่นกัน



ห้องเรียน 1 ห้องสามารถเปิดทำการเรียนการสอนได้หลายรายวิชา และ 1 รายวิชาสามารถใช้ห้องเรียนหลายห้องเพื่อทำการเรียนการสอนได้เช่นเดียวกัน

รายวิชา 1 รายวิชาจะสามารถเปิดได้ จะต้องมื่อนักศึกษาลงทะเบียน
อย่างน้อย 20 คนและนักเรียน 1 คนสามารถมีวิชาเรียนได้หลายวิชา

ขั้นตอนที่ 3 กำหนด Attribute ให้แต่ละ Entity และกำหนด Primary Key

- ❑ Faculty (FAC_ID, FAC_NAME)
โดยที่ FAC_ID เป็น PK
- ❑ Course (Course_ID, Course_Name)
โดยที่ Course_ID เป็น PK
- ❑ Teacher (Teacher_ID, Teacher_Name)
โดยที่ Teacher_ID เป็น PK
- ❑ Student (STD_ID, STD_SEX, STD_NAME)
โดยที่ STD_ID เป็น PK
- ❑ Room (Room_No)
โดยที่ Room_No เป็น PK

5.2 Bottom-up Approach

- ❑ เป็นการสร้างแบบจำลองข้อมูลจากเอกสารที่มีอยู่ในองค์กร
เช่น แบบฟอร์ม รายงาน หรือหน้าจอแสดงผลของระบบงาน
ปัจจุบัน เอกสารเหล่านี้จะปรากฏเป็น Data Flow อยู่บน DFD
- ❑ ดังนั้น SA จึงสามารถสร้างแบบจำลองข้อมูลได้จาก DFD และ
เอกสารอื่น ๆ เพิ่มเติมเพื่อสร้างความเข้าใจให้ชัดเจนยิ่งขึ้น
- ❑ ขั้นตอน
 - กำหนด Entity ทั้งหมดของระบบ
 - สร้าง Relationship ระหว่าง Entity
 - กำหนด Attribute และ Primary Key

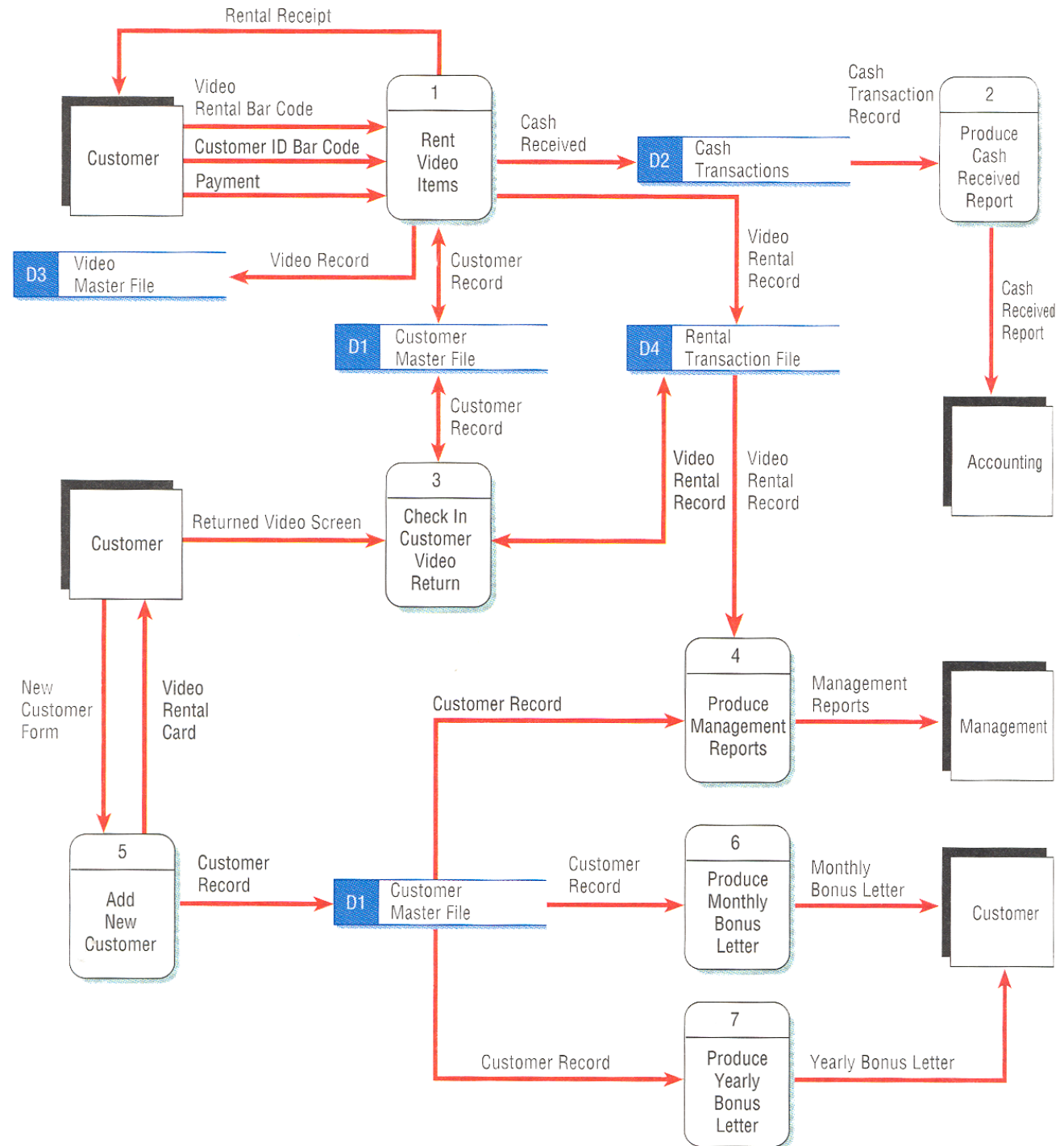
Example :

DFD Level 0

for

Video Rental

System



ขั้นตอนที่ 1 กำหนด Entity ทั้งหมดของระบบ

- ❑ พิจารณา Data Store ที่ปรากฏบน DFD
- ❑ พบว่ามี 4 Entity ได้แก่
 - Customer
 - Cash
 - Video
 - Rental

ขั้นตอนที่ 2 สร้าง Relationship ระหว่าง Entity

- ❑ ลูกค้า (Customer) ทำ (Create) รายการเช่า (Rental) กับทางร้าน โดยที่ลูกค้า 1 คน เข้าได้หลายรายการ
- ❑ รายการเช่า (Rental) แต่ละรายการจะมี (Have) ข้อมูลวิดีโอ (Video) ตั้งแต่ 1 เรื่องขึ้นไป
- ❑ รายการเช่า (Rental) แต่ละรายการจะได้รับ (Receive) การจ่ายเงิน (Cash) ค่าเช่า

ขั้นตอนที่ 3 กำหนด Attribute และ Primary Key

- ❑ Customer (CustomerID, Name, Lastname, Address, Phone)
โดยที่ CustomerID เป็น PK
- ❑ Video (VideoID, Title, VideoType, Charge)
โดยที่ VideoID เป็น PK
- ❑ Rental (RentalID, DateRent, CustomerID, VideoID)
โดยที่ RentalID เป็น PK
- ❑ Cash (CashID, RentalID, Amount)
โดยที่ CashID เป็น PK