



# DSI320 Data Architecture

## วช.320 สถาปัตยกรรมการจัดการข้อมูล

ผู้ช่วยศาสตราจารย์ ดร.สมเกียรติ โกศลสมบัติ

สาขาวิชาวิทยาศาสตร์และนวัตกรรมข้อมูล

วิทยาลัยสหวิทยาการ มหาวิทยาลัยธรรมศาสตร์

# Contents

---

- Data Management
- Introduction to EA
- Entity-Relationship Model (ER Model)
- Mapping ER Diagram
- Data Modeling
- References

# Data Management



# Enterprise Architecture (EA)

---

- Enterprise Architecture (EA) is a practice that involved the design, planning, and management of an organization's business processes, information system, and infrastructure, with the goal of aligning all of these elements with the organization's goals and strategies.
- EA frameworks provide a structured approach for defining and implementing an enterprise architecture, and for using that architecture to guide the organization's transformation and growth.
- Some common elements of enterprise architecture include business architecture, data architecture, application architecture, technology architecture, and security architecture.

# Data Architecture

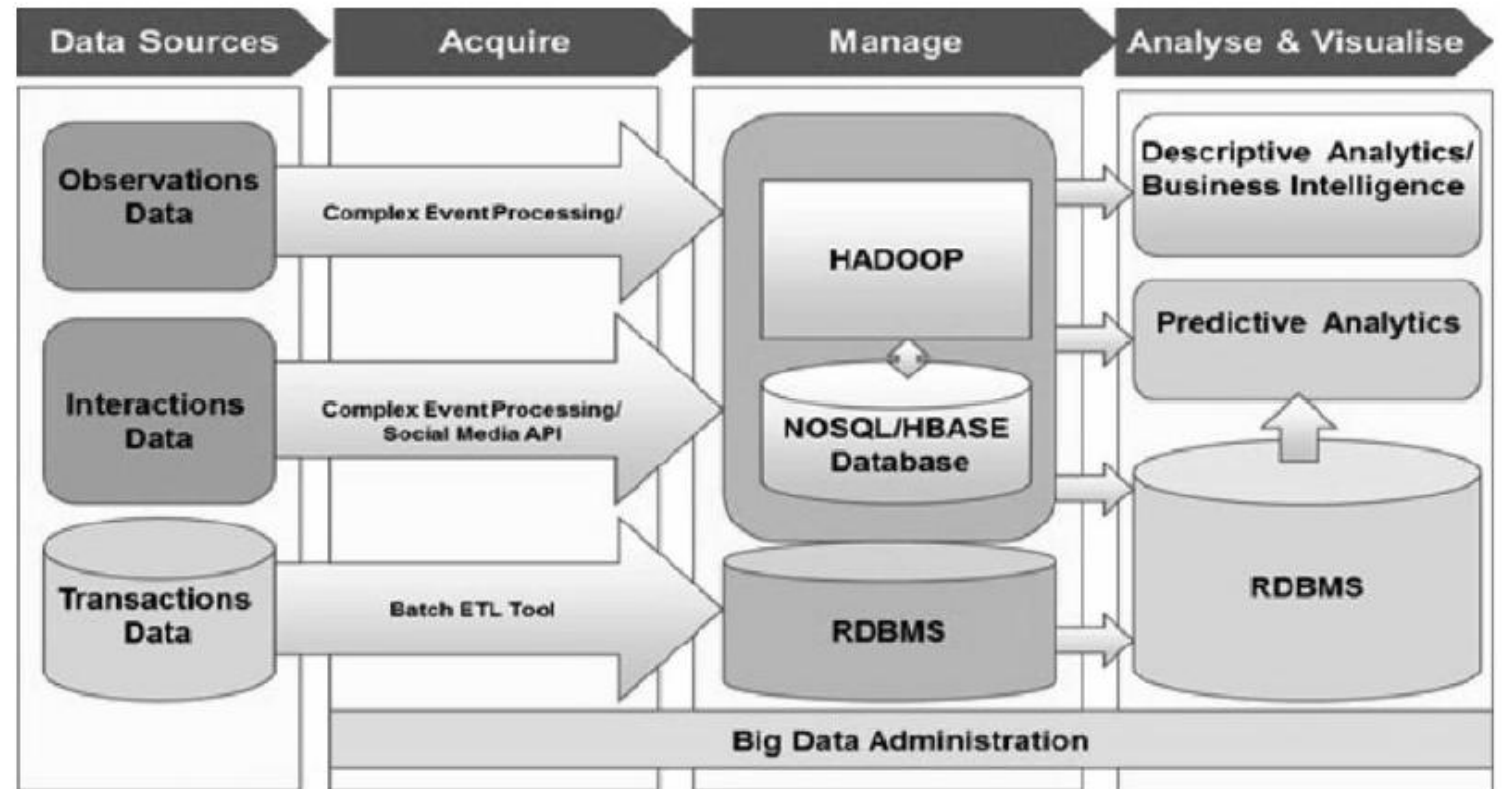
---

- สถาปัตยกรรมฐานข้อมูล (Database Architecture)
- หรือแนวคิดที่ใช้อธิบายโครงสร้างองค์ประกอบหลักของระบบข้อมูลภายในฐานข้อมูล การจัดเก็บข้อมูลภายในฐานข้อมูล รวมถึงการติดต่อกับส่วนอื่น ๆ ที่เกี่ยวข้อง

# DA Diagram

---

- The data architecture diagram is the diagrammatic representation of how the data is managed throughout the whole data life from consuming to disposing of securely.



# ER Diagram

---

แบบจำลองความสัมพันธ์ Entity

# Entity Relationship Model

---

- โมเดลข้อมูลเชิงสัมพันธ์ (ER Model)
  - Entity
  - Attribute
  - Key
  - Relationship
  - Cardinality



# Entity

- An entity can be a person, place, event, or object that is relevant to a given system.
- For example, a school system may include students, teachers, major courses, subjects, fees, and other items.
- Entities are represented in ER diagrams by a rectangle and named using singular **nouns**.



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

# Weak Entity

- A weak entity is an entity that **depends on the existence of another entity**.
- In more technical terms it can be defined as an entity that cannot be identified by its own attributes.
- It uses a foreign key combined with its attributed to form the primary key.
- The order item will be meaningless without an order, so it depends on the existence of the order.



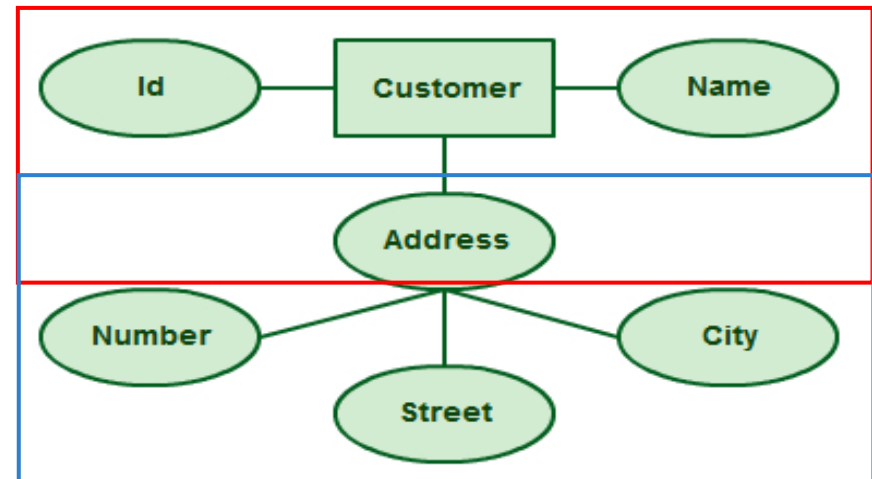
EmpID	Date	TimeIn	TimeOut
47035501	2/7/47	7:30	17:30
47035502	2/7/47	8:00	18:00
47035503	2/7/47	7:45	17:45

# Attribute

- An attribute is a property, trait, or characteristic of an entity, relationship, or another attribute.
- For example, Name is an attribute of the entity Inventory Item.
- An entity can have as many attributes as necessary.
- Meanwhile, attributes can also have their own specific attributes.
- For example, the attribute “customer address” can have the attributes number, street, city, and state.

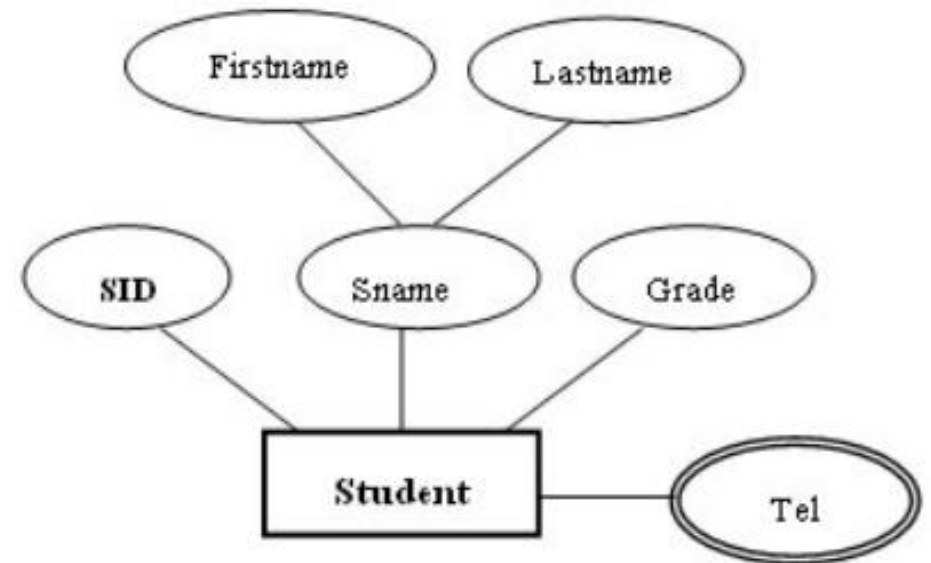
Simple Property

Composite Property



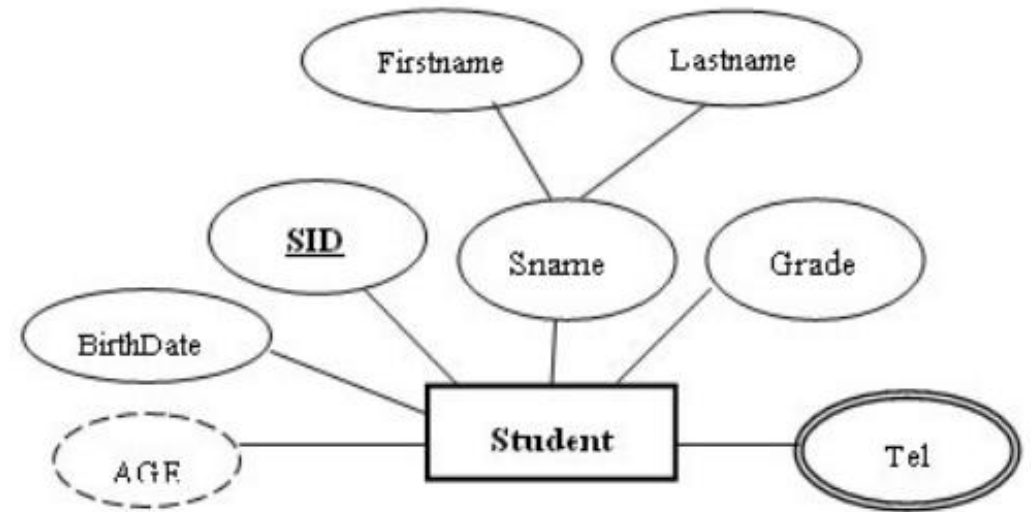
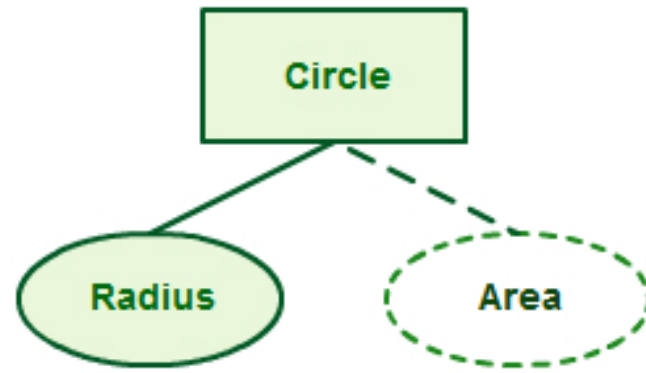
# Multivalued Attribute

- If an attribute can have more than one value it is called a multi-valued attribute.
- It is important to note that this is different from an attribute having its own attributes.
- For example, a teacher entity can have multiple subject values.



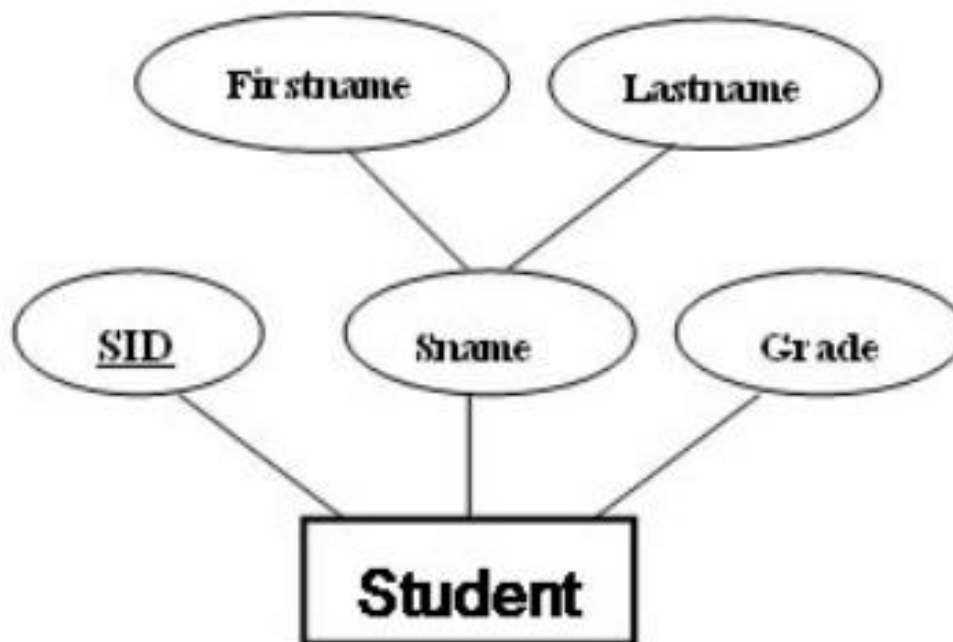
# Derived Attribute

- An attribute based on another attribute.
- For example, for a circle, the area can be derived from the radius.



# Key

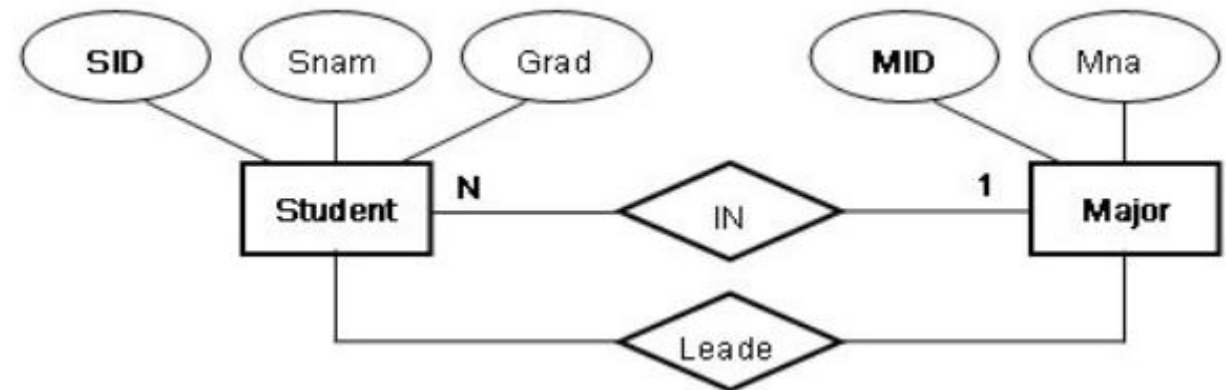
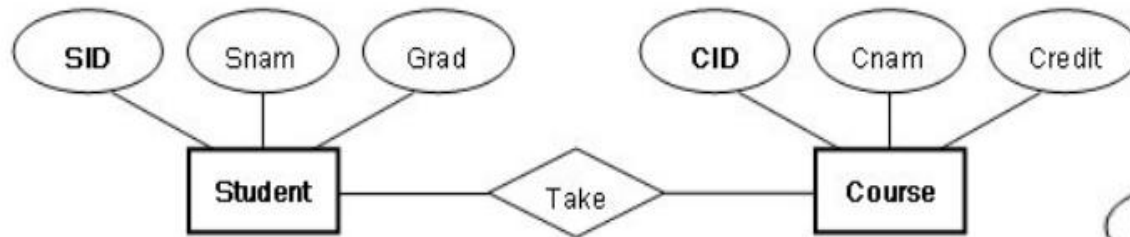
- ไม่ซ้ำกัน
- Primary
- Secondary



# Relationship

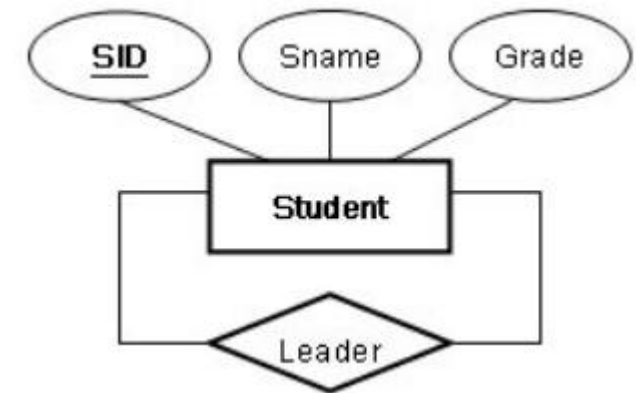
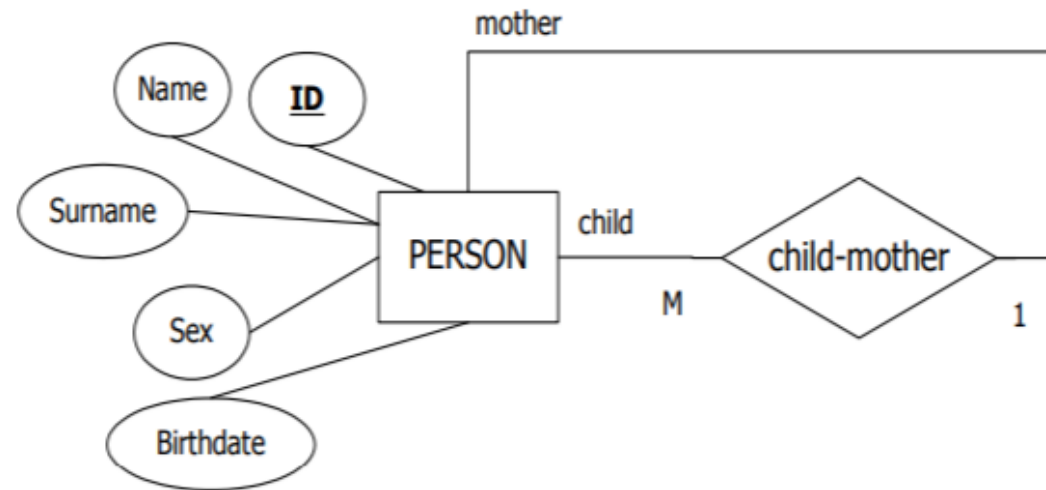
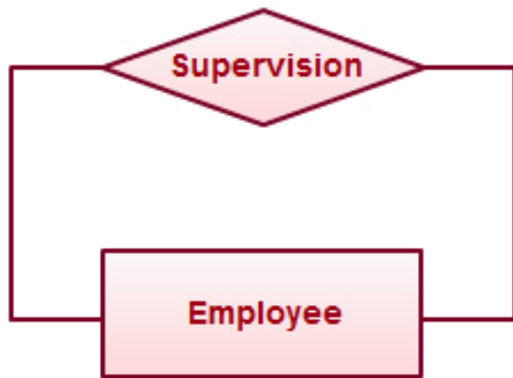


- A relationship describes how entities interact.
- For example, the entity “Carpenter” may be related to the entity “table” by the relationship “builds” or “makes”.
- Relationships are represented by diamond shapes and are labeled using **verbs**.



# Recursive Relationship

- If the same entity participates more than once in a relationship it is known as a recursive relationship.
- In the below example an employee can be a supervisor and be supervised, so there is a recursive relationship.





# Cardinality

- These two further defines relationships between entities by placing the relationship in the context of numbers.
- For example, In an email system, one account can have multiple contacts.
- The relationship, in this case, follows a “one to many” model.



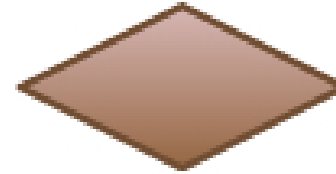
# Entity Relationship Symbols and Notations



**Entity**



**Attribute**



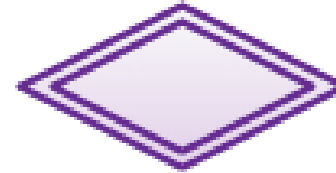
**Relationship**



**Weak  
Entity**



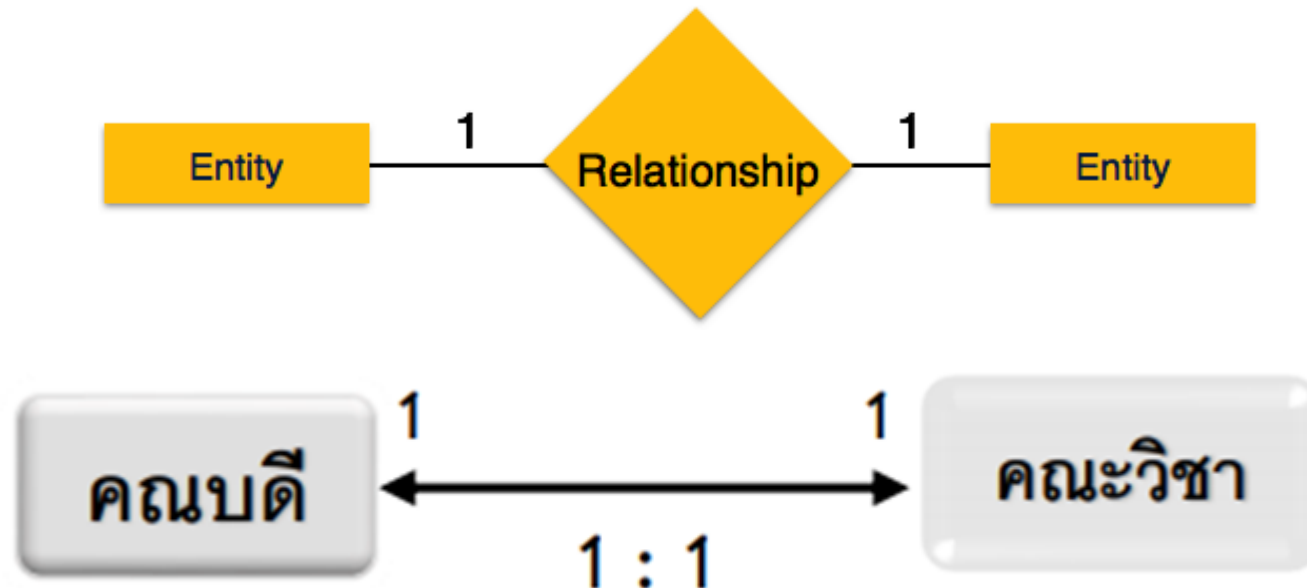
**Multivalued  
Attribute**



**Weak  
Relationship**

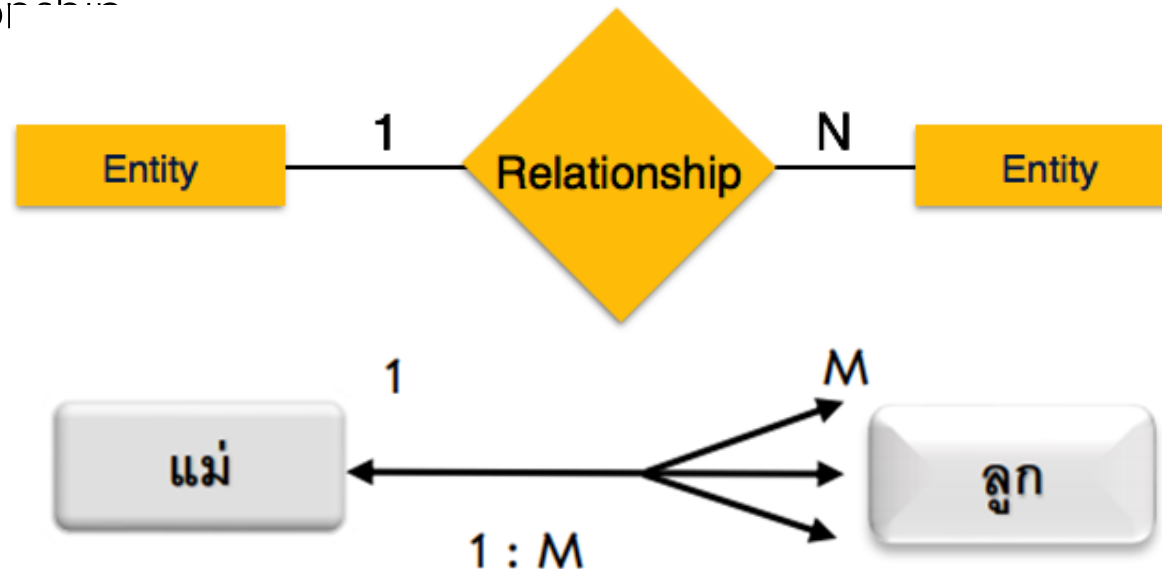
# Binary Relationship and Cardinality

- A relationship where two entities are participating is called a binary relationship.
- Cardinality is the number of instance of an entity from a relation that can be associated with the relation.
- One-to-one – When only one instance of an entity is associated with the relationship, it is marked as '1:1'.



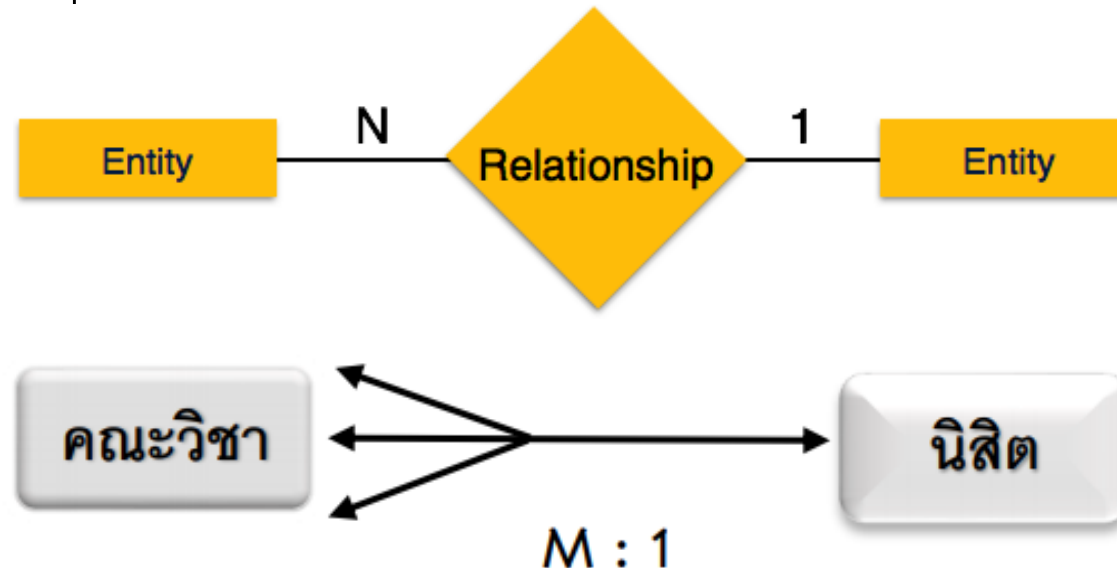
# Binary Relationship and Cardinality

- One-to-many – When more than one instance of an entity is associated with a relationship, it is marked as '1:N'.
- The following image reflects that only one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship.
- It depicts one-to-many relationship



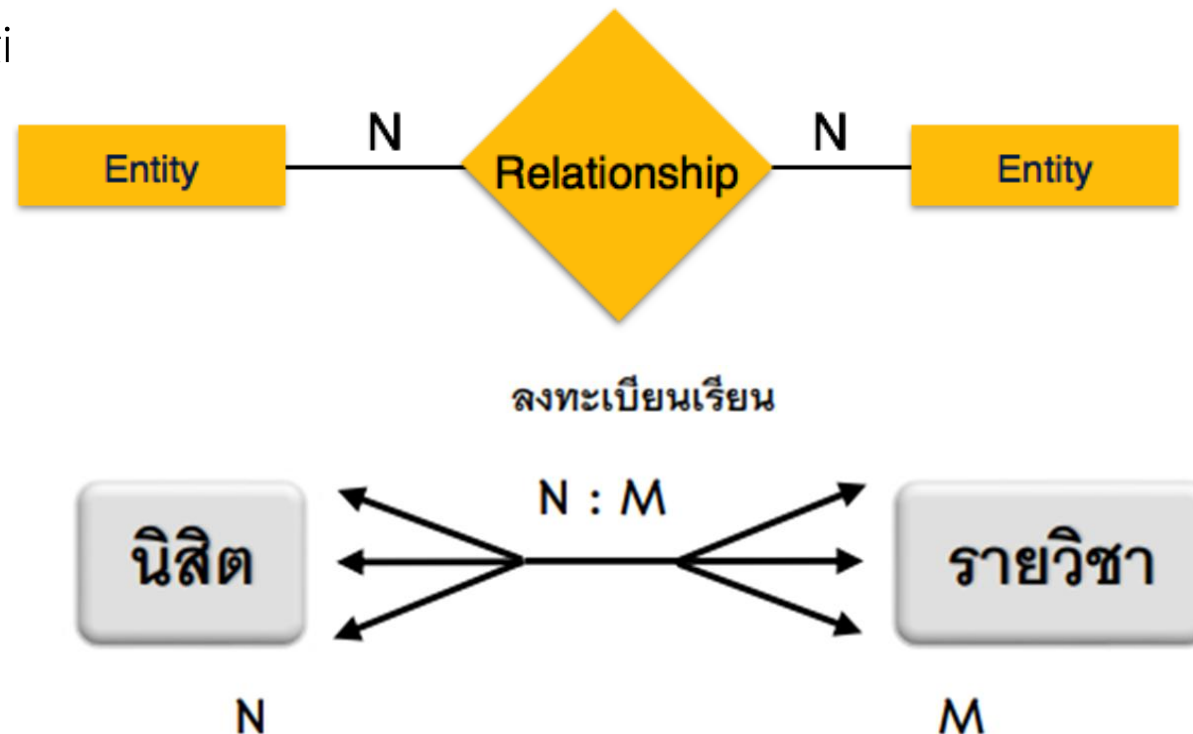
# Binary Relationship and Cardinality

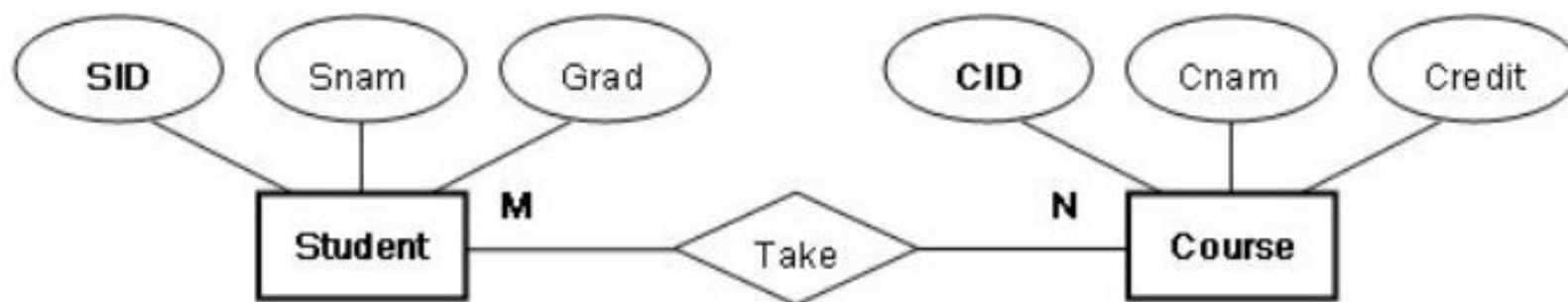
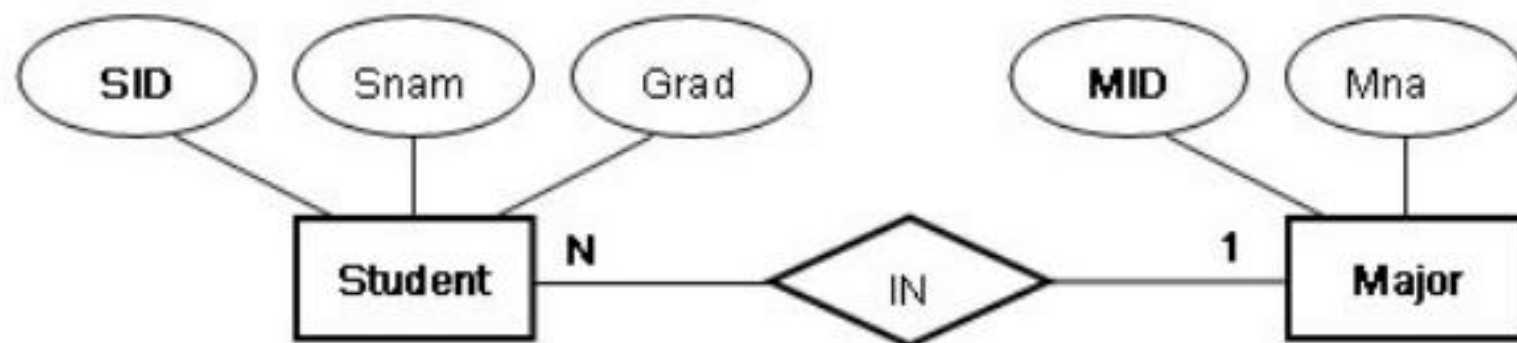
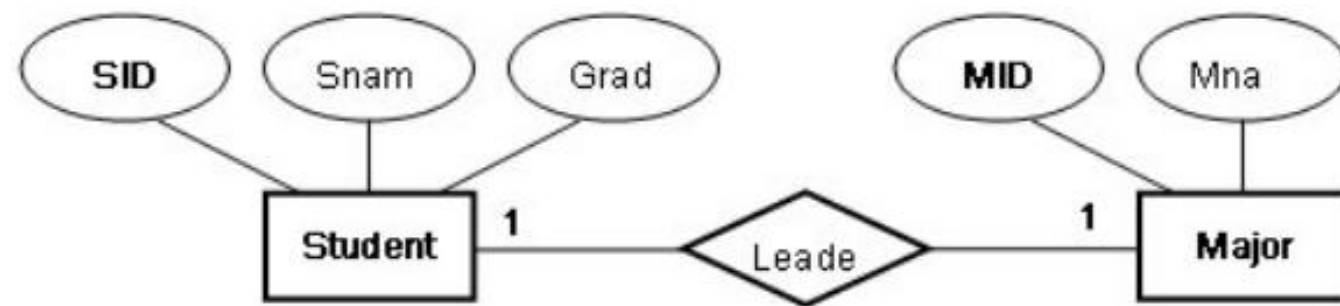
- Many-to-one – When more than one instance of entity is associated with the relationship, it is marked as 'N:1'.
- The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship.
- It depicts many-to-one relationship.



# Binary Relationship and Cardinality

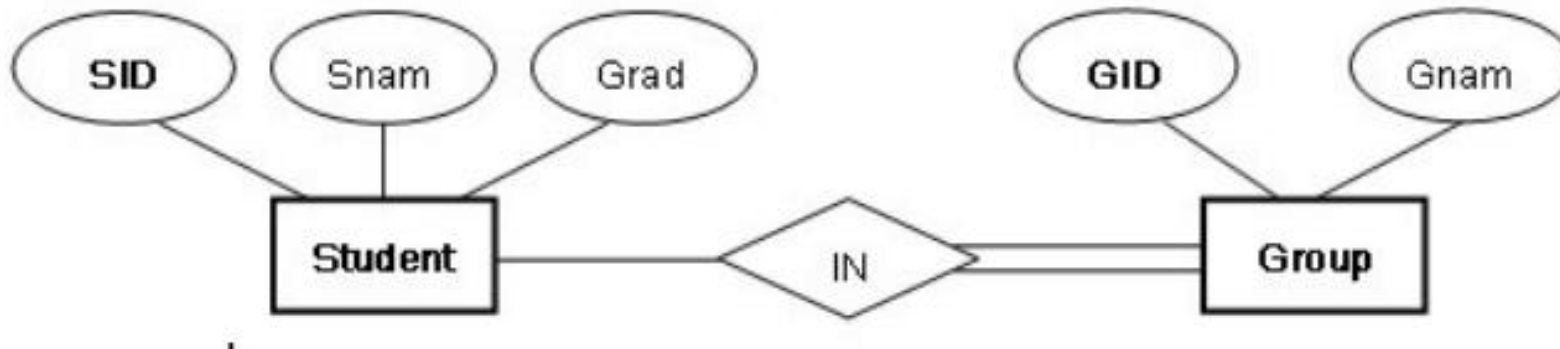
- Many-to-many – The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship.
- It depicts many-to-many relati





# Participation

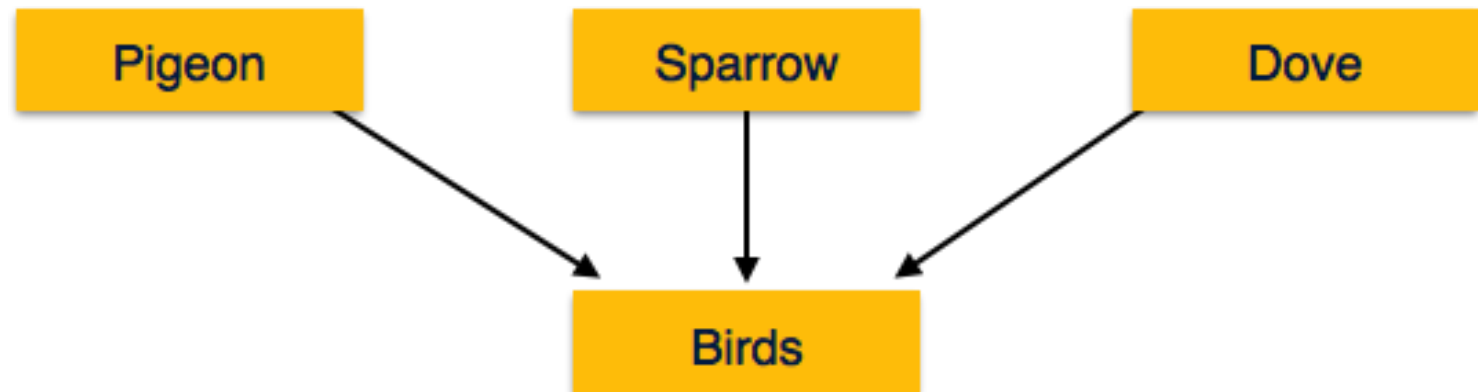
- Total Participation
  - ทุกข้อมูลใน Entity จะต้องมีความสัมพันธ์กับข้อมูลในอีก Entity หนึ่ง
- Partial Participation
  - ข้อมูลใน Entity อาจจะไม่มีความสัมพันธ์กับข้อมูลในอีก Entity หนึ่งหรือไม่ก็ได้





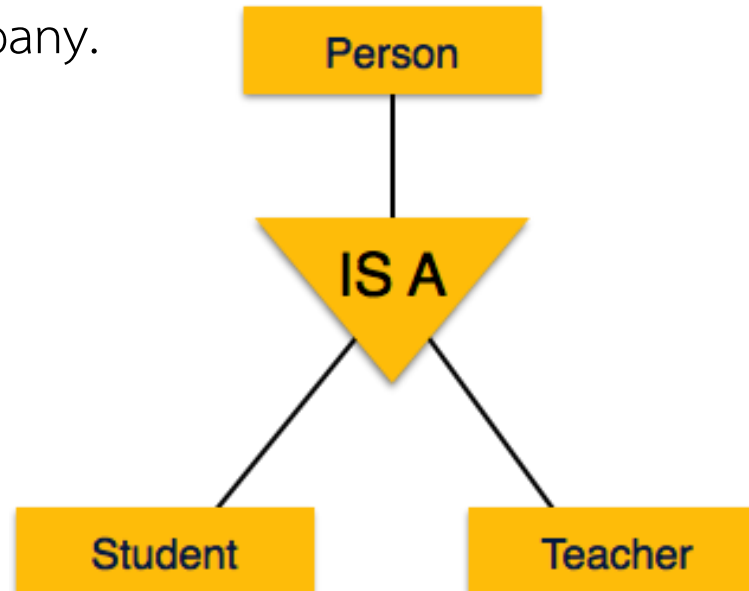
# Generalization

- The process of generalizing entities, where the generalized entities contain the properties of all the generalized entities, is called generalization.
- In generalization, a number of entities are brought together into one generalized entity based on their similar characteristics.
- For example, pigeon, house sparrow, crow and dove can all be generalized as Birds.



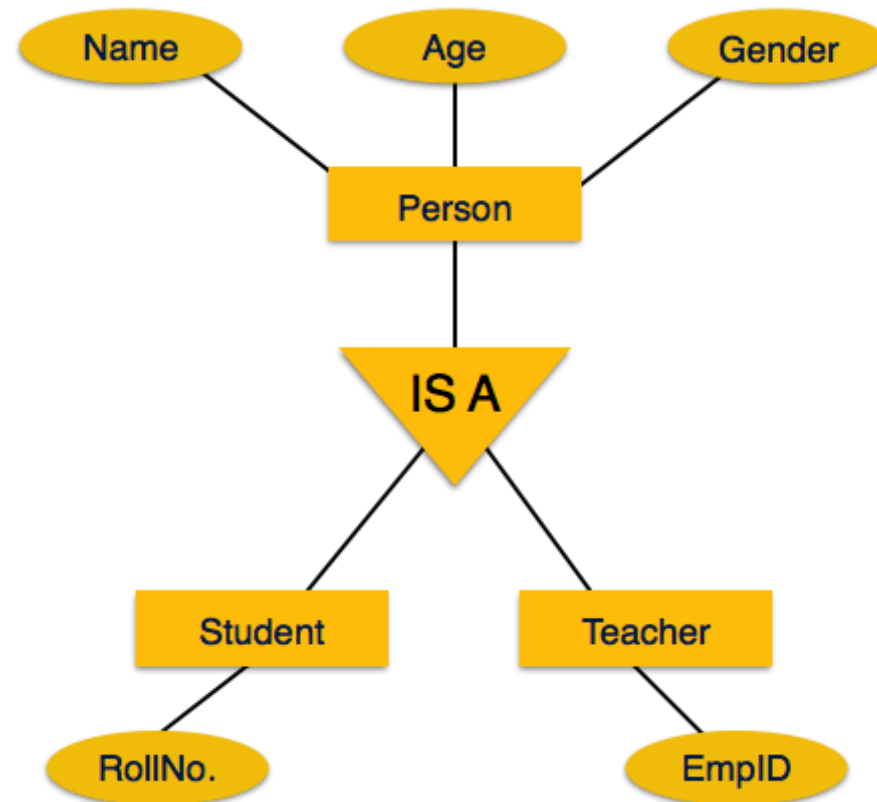
# Specialization

- Specialization is the opposite of generalization.
- In specialization, a group of entities is divided into sub-groups based on their characteristics.
- These properties are common in all persons, human beings.
- But in a company, persons can be identified as employee, employer, customer, or vendor, based on what role they play in the company.

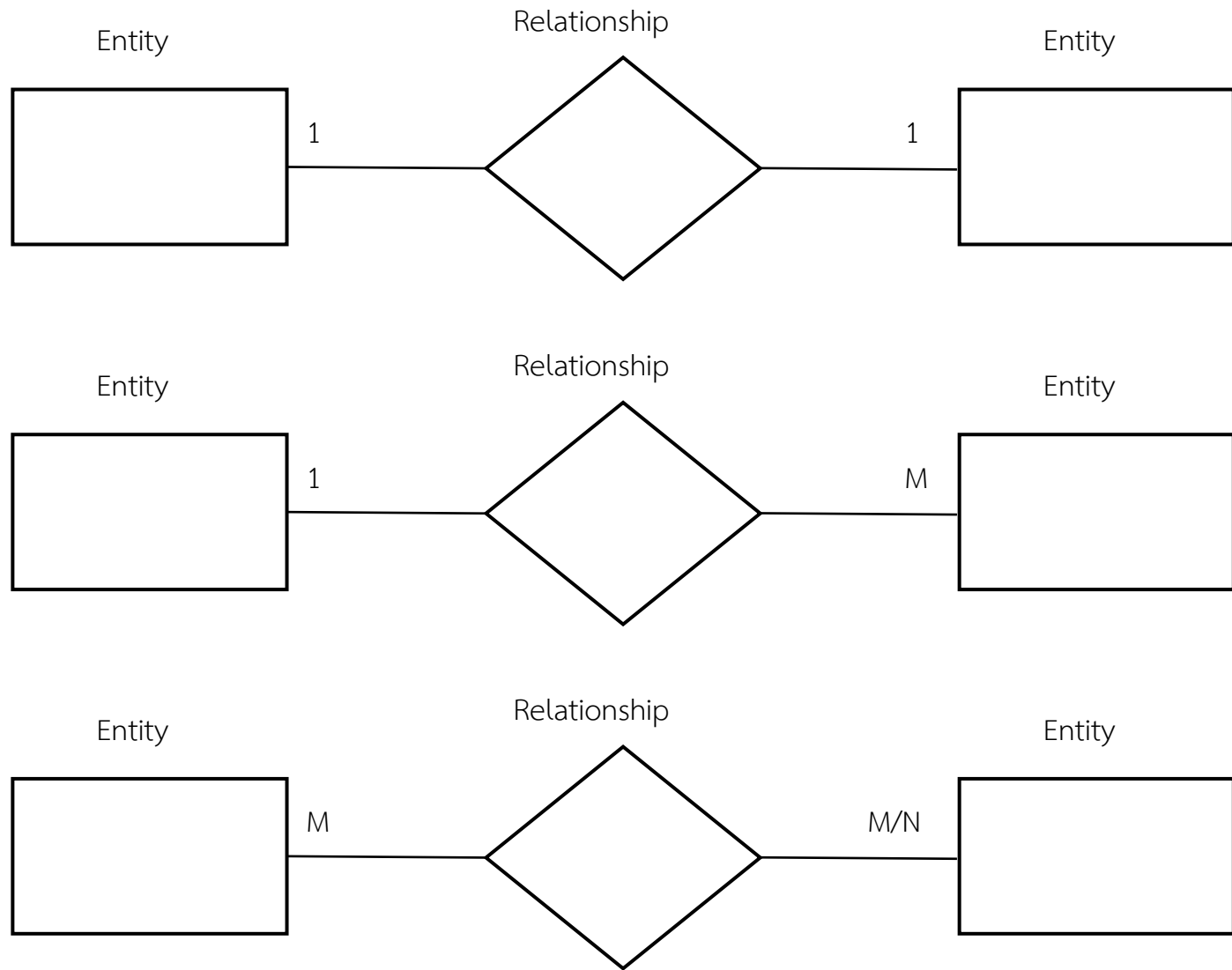


# Inheritance

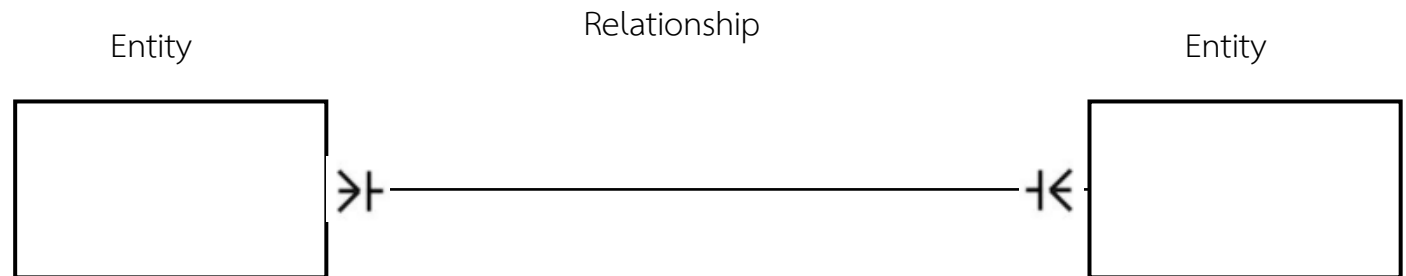
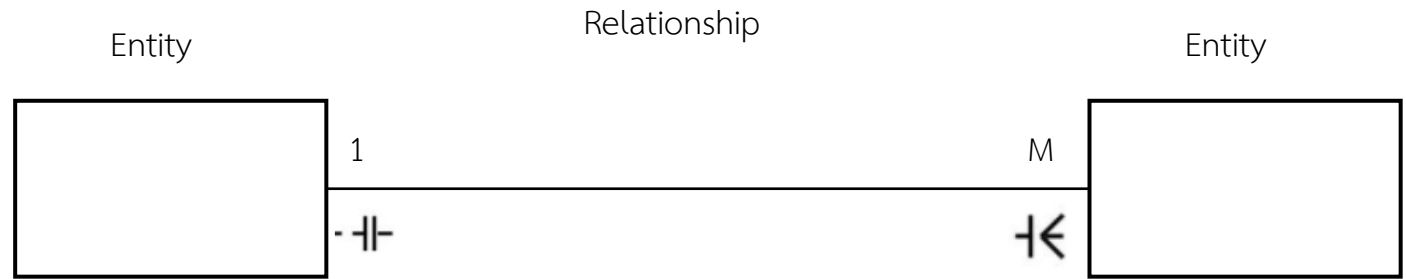
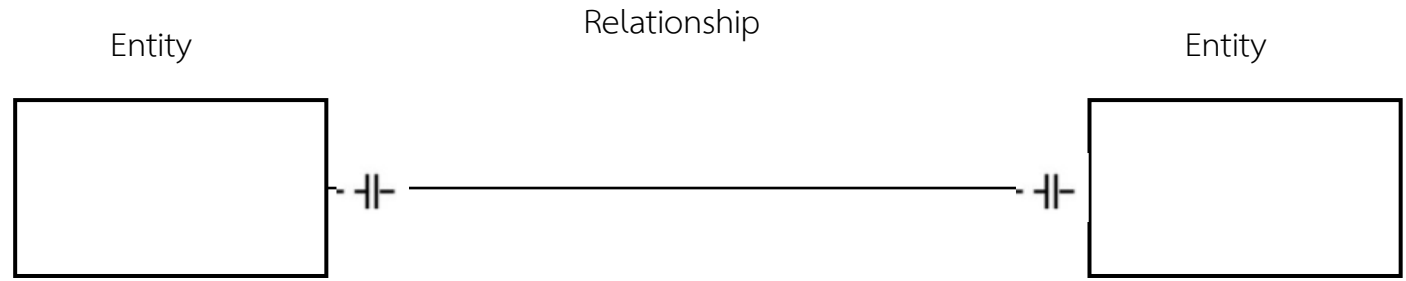
- Inheritance is an important feature of Generalization and Specialization.
- It allows lower-level entities to inherit the attributes of higher-level entities.



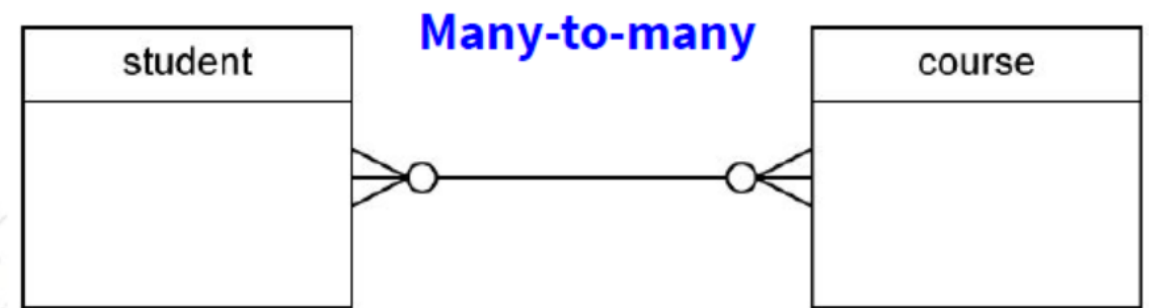
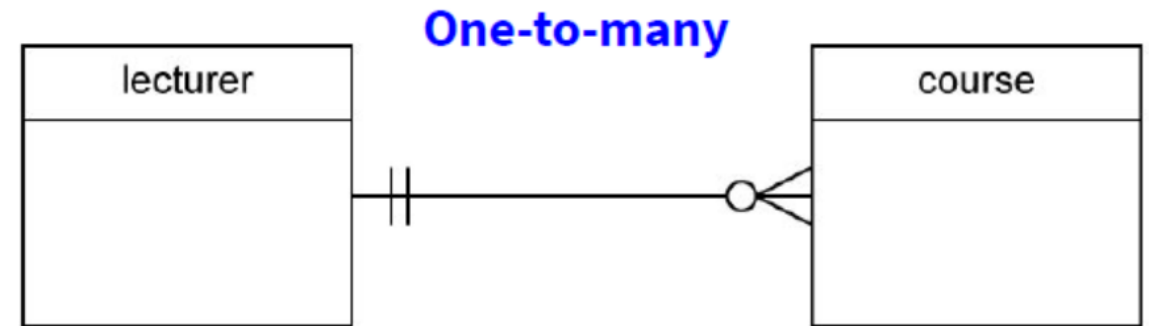
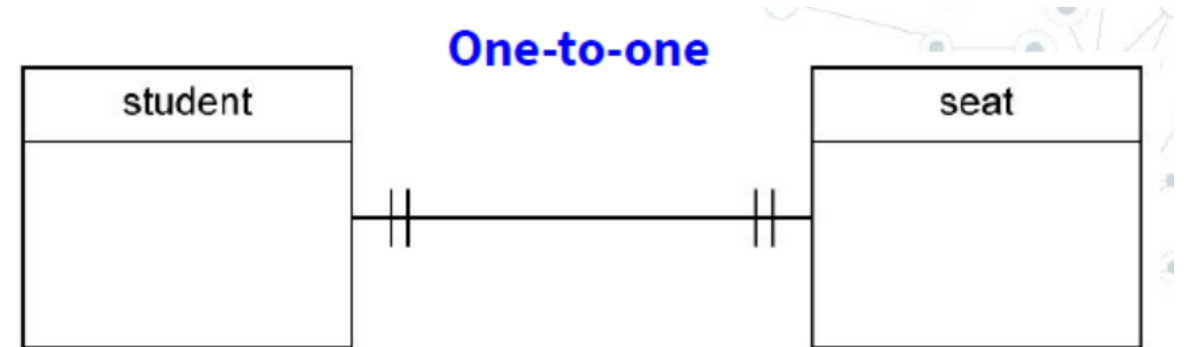
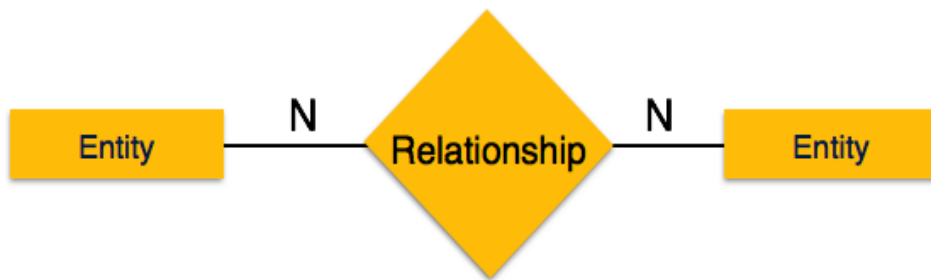
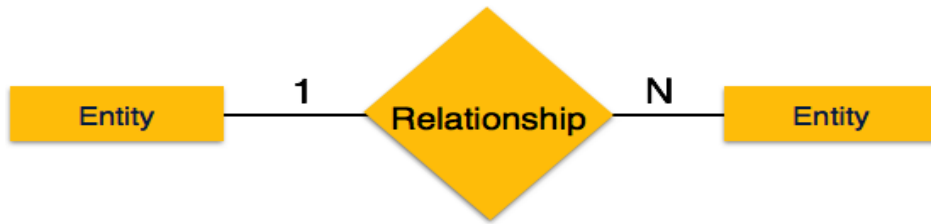
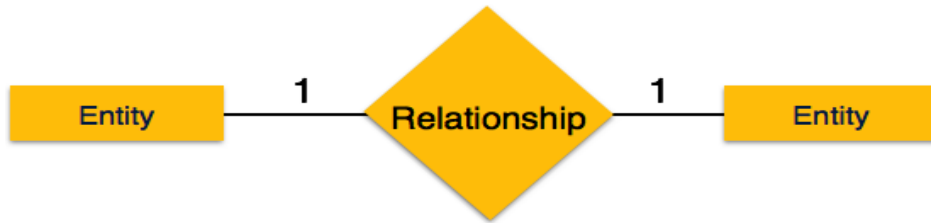
# ส่วนประกอบ ER Model (Peter Chen)



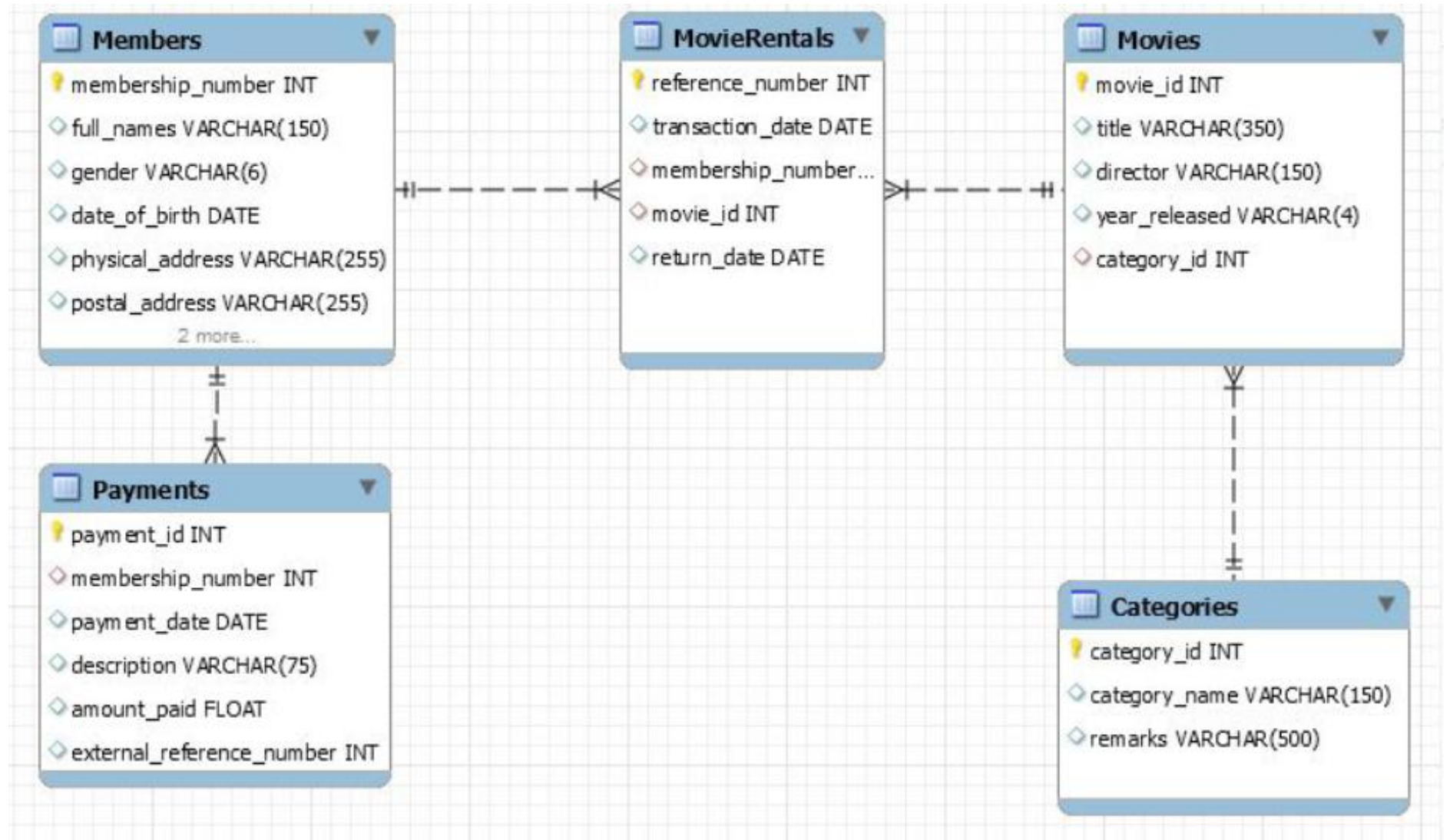
# ส่วนประกอบ ER Model (Crow's Foot)



- Chen and Crow's Foot Notation



# Crow's Foot Example

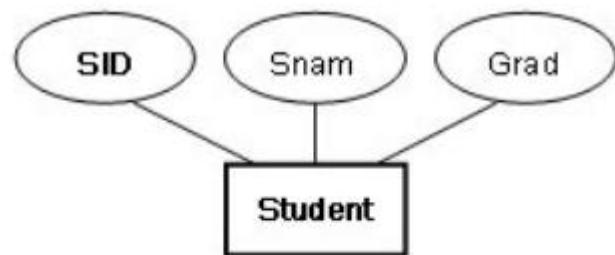


# Mapping ER Diagram

---

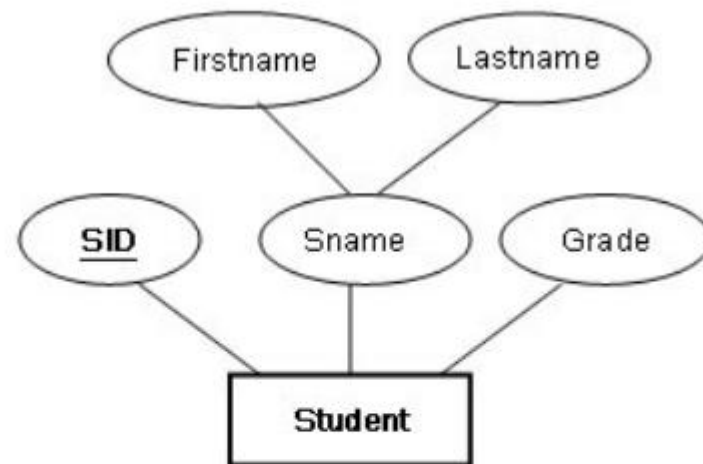
การแปลง ER Diagram เป็นตารางข้อมูล





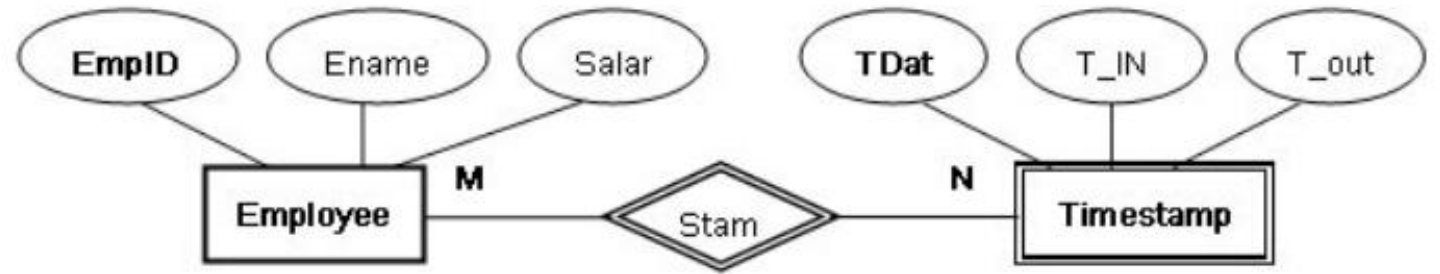
ပုံ ၁၁.၁၁ Student

<u>SID</u>	Sname	Grade



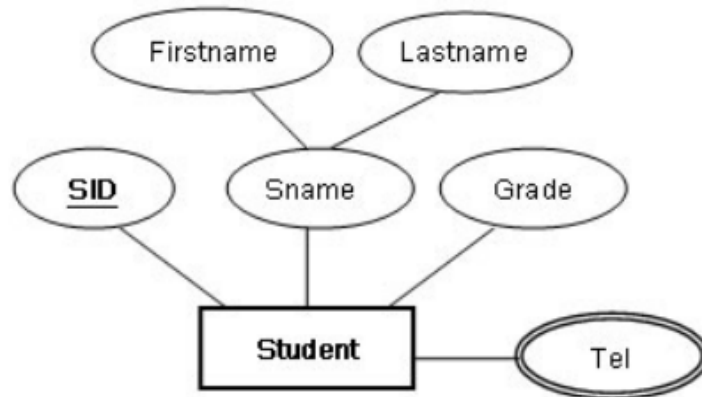
ပုံ ၁၁.၁၂ Student

<u>SID</u>	Firstname	lastname	Grade



ဖော်ပြချက် TimeStamp

<u>EmpID</u>	<u>TDate</u>	T_In	T_out

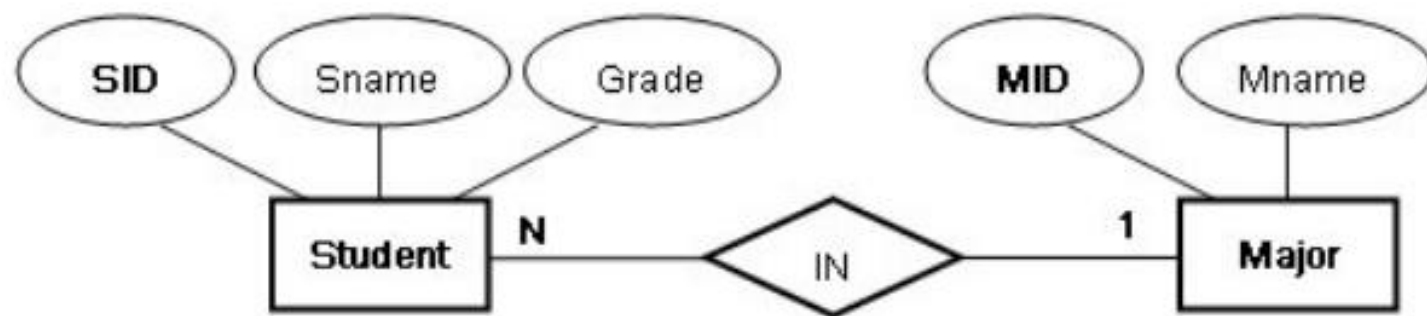


ဖော်ပြချက် Student

<u>SID</u>	Firstname	lastname	Grade

ဖော်ပြချက် Tel

<u>SID</u>	Tel

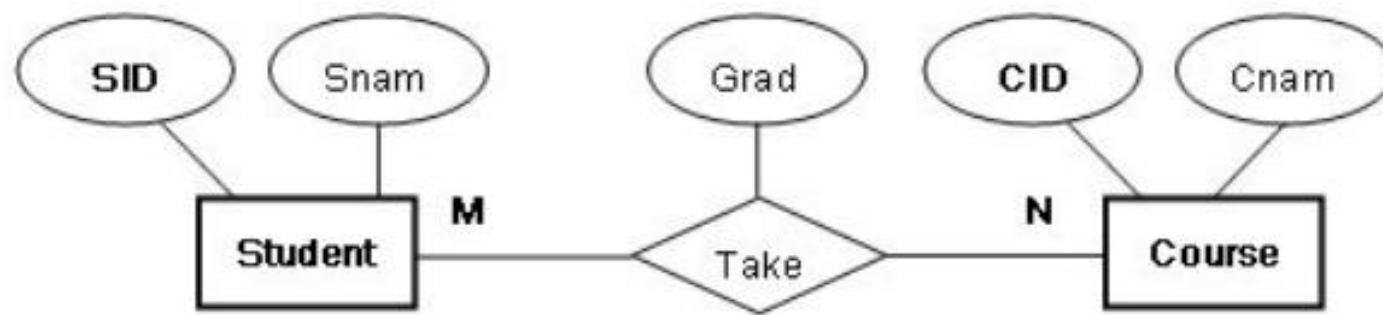


የሰነድ Student

<u>SID</u>	Sname	Grade	MID

የሰነድ Major

<u>MID</u>	Mname



የተማሪ Student

<u>SID</u>	Sname

የተማሪ Take

<u>SID</u>	CID	Grade

የምረቃ Course

<u>CID</u>	Cname

# Data Modeling

---

แบบจำลองข้อมูล

# Data Modeling

---

- Data modeling is the **process of creating a data model for the data to be stored in a database.**
- This data model is a conceptual representation of data objects, the associations between different data objects, and the rules.
- Data modeling **helps in the visual representation of data** and enforces business rules, regulatory compliances, and government policies on the data.
- Data models ensure consistency in naming conventions, default values, semantics, security while ensuring quality of the data.

# Data Model

---

- The Data model is defined as an **abstract model** that organizes data description, data semantics, and consistency constraints of data.
- The data model emphasizes on **what data is needed and how it should be organized instead of what operations** will be performed on data.
- Data model is **like an architect's building plan**, which helps to build conceptual models and set a relationship between data items.

# Why use Data Model

---

- Ensures that all data objects required by the database are accurately represented.
- A data model helps design the database at the conceptual, physical and logical levels.
- Data model structure helps to define the relational tables, primary and foreign keys and stored procedures.
- It provides a clear picture of the base data and can be used by database developers to create a physical database.
- It is also helpful to identify missing and redundant data.



# Types of Data Model

---

- Conceptual Data Model (WHAT)
- Logical Data Model (HOW)
- Physical Data Model (HOW)

# Types of Data Model

---

- **Conceptual Data Model**: This data model defines **WHAT** the system contains.
- This model is typically created by **Business Stakeholders and Data Architects**.
- The purpose is to **organize, scope and define business concepts and rules**.

# Types of Data Model

---

- Logical Data Model: Defines HOW the system should be implemented regardless of the DBMS.
- This model is typically created by Data Architects and Business Analysts.
- The purpose is to developed technical map of rules and data structures.

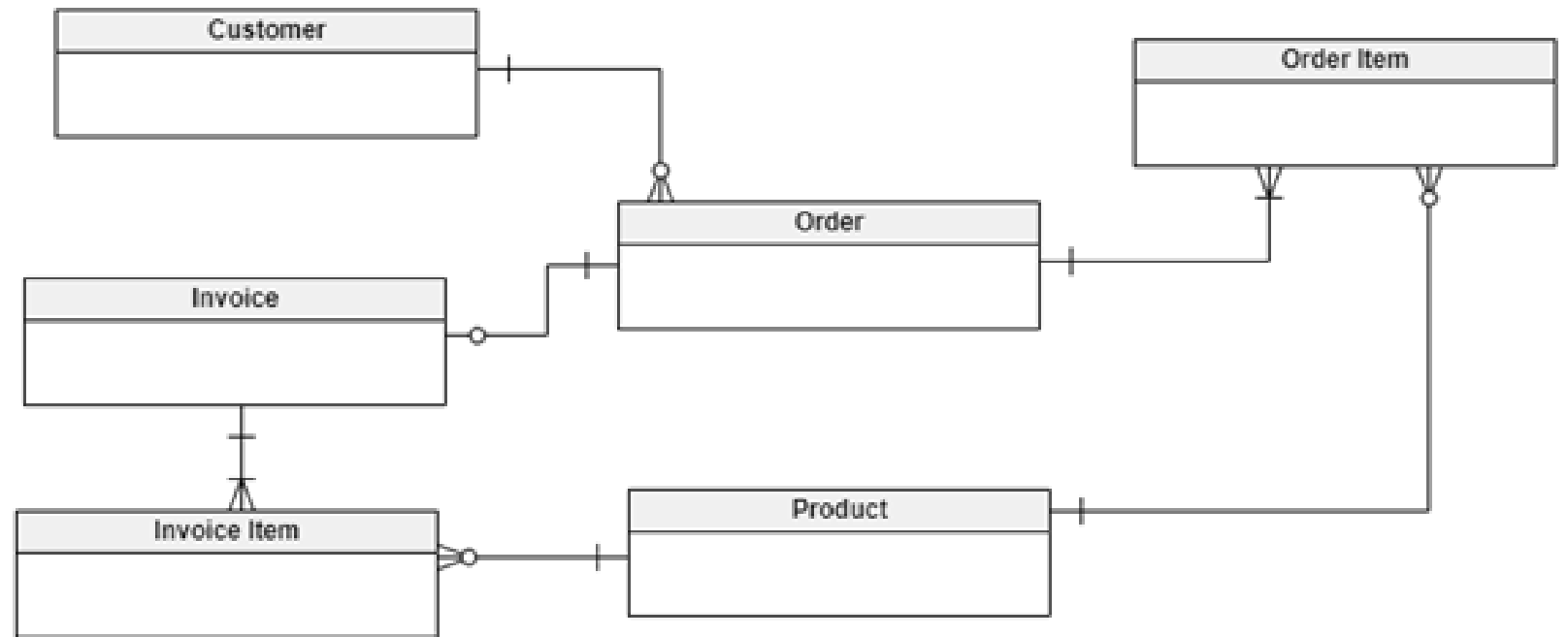
# Types of Data Model

---

- **Physical Data Model**: This Data Model describes **HOW** the system will be implemented using a specific DBMS system.
- This model is typically created by **DBA and developers**.
- The purpose is actual implementation of the database.

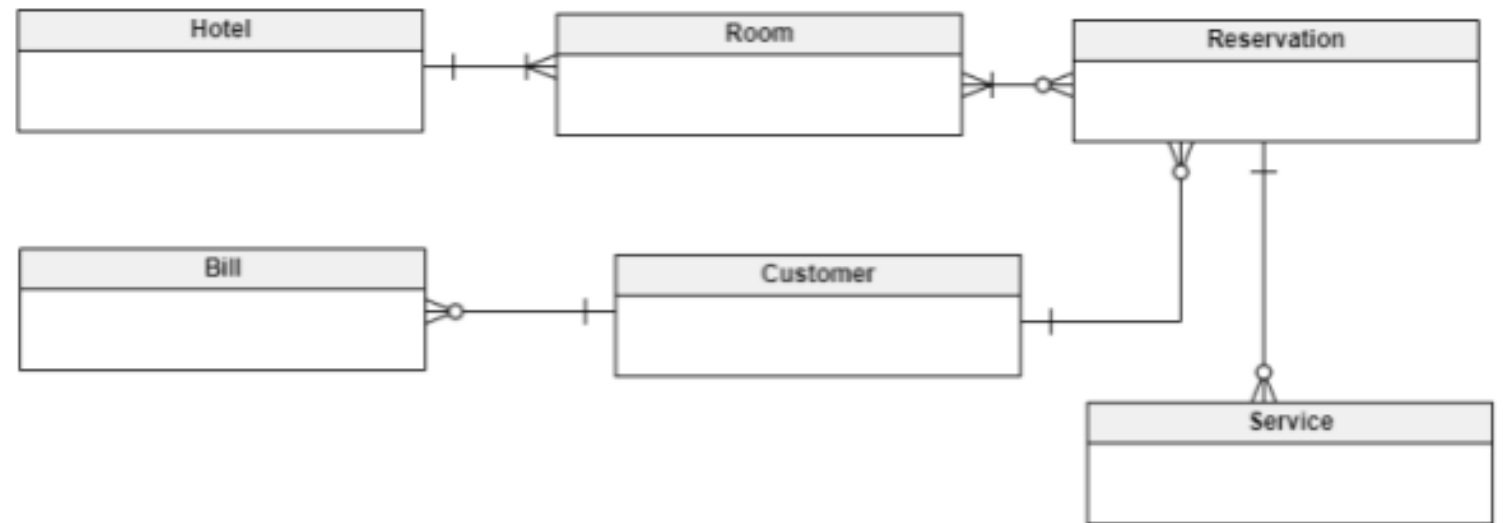
# Conceptual Data Model

---



# Conceptual Data Model

---



# References

---

- <https://www.pinterest.com/pin/191825265356325311/>
- แบบจำลองข้อมูล (Database Models)
- <https://creately.com/blog/diagrams/er-diagrams-tutorial/>
- [https://www.tutorialspoint.com/dbms/er\\_model\\_basic\\_concepts.htm](https://www.tutorialspoint.com/dbms/er_model_basic_concepts.htm)
- <https://segment.com/blog/data-modeling/>
- [https://pirun.ku.ac.th/~faastwc/02739422-6 2/pttx/chapter04%20Relational%20Model.pdf](https://pirun.ku.ac.th/~faastwc/02739422-6%20pttx/chapter04%20Relational%20Model.pdf)