# Database Security

CT069-3-3


## Entity Relationship Modelling

# Learning Outcomes

**At the end of this module, YOU should be able to :**

- Explain the entity relationship diagram (ERD)  terminologies and notations
- Create an ERD based on a case study or requirements given
- Start work on your assignment question

# Key Terms you must be able to use

**If you have mastered this topic, you should be able to use the following terms correctly :**

- Entity , Attribute
- Relationship , Participation, Cardinality

# Entity

- Entity is anything **(a person, a place, a thing or an event)** about which data are to be collected and stored.
- Entities may be physical objects such as customers or products, but entities also can be abstractions, such as flight routes or musical concert.

Customer

Product

# Relationship

- Relationship is an **association between the entities**.
- Example: A customer purchases items.

- Customer → Entity
- Product → Entity
- Purchase → Relationship

Customer

Purchase

Product

# Attribute

- Attribute is a **characteristics of an entity or relationship**.

Customer

| First Name |
| --- |
| Last Name |
| Address |
| Phone No. |
| Email |

Purchase

| Purchase Date |
| --- |
| Quantity |
| Total Price |

Product

| Product Name |
| --- |
| Description |
| Category |
| Unit Price |
| Quantity in Stock |

**Attributes**

# Purpose of ERD ?

A·P·U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

- ERD is to help us to capture the business requirements related to data
- By having an ERD, we can model data that is required and the relationship between the data

# Entity Relationship Diagram

- ER Model is commonly represented by Chen or Crow's Foot models/diagrams
- To draw an ER diagram, one needs:

# Notation of Entity in ERD

## In both Chen and Crow's Foot models

| | |
|---|---|
| Entity | **Rectangular box** labelled with the name of the entity type<br><br>The entity type name should be **singular and in uppercase letters**<br><br>Use underscore in place of space for entity type name |

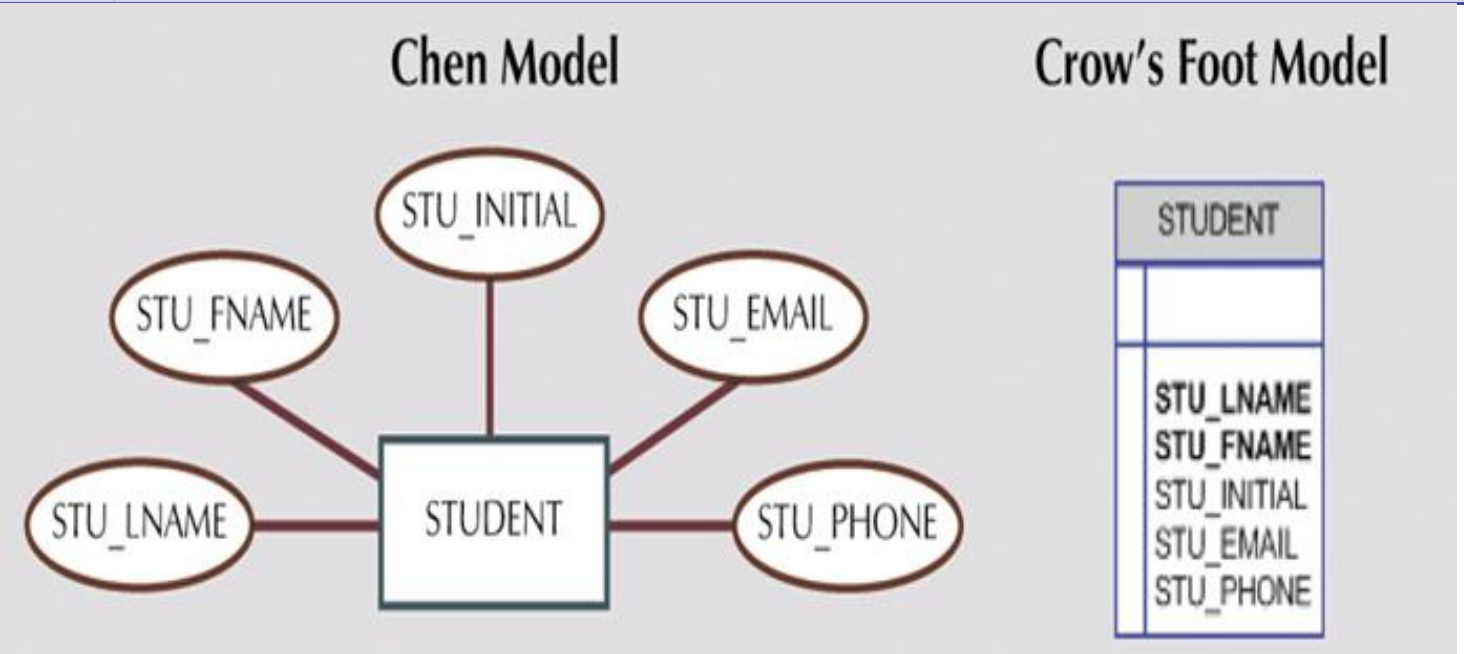
MOVIE_STAR

APU_MODULE

BOOK

# Notation of Attribute in ERD

| | |
|---|---|
| Attribute | In Chen model, depicted in an **Oval** shape, labelled with the name of the attribute<br><br>The oval is attached by a straight line to its entity type<br><br>The attribute name should be capitalised.<br><br>In Crow's Foot model, attributes are written in **attribute box** below entity rectangle |



# Example

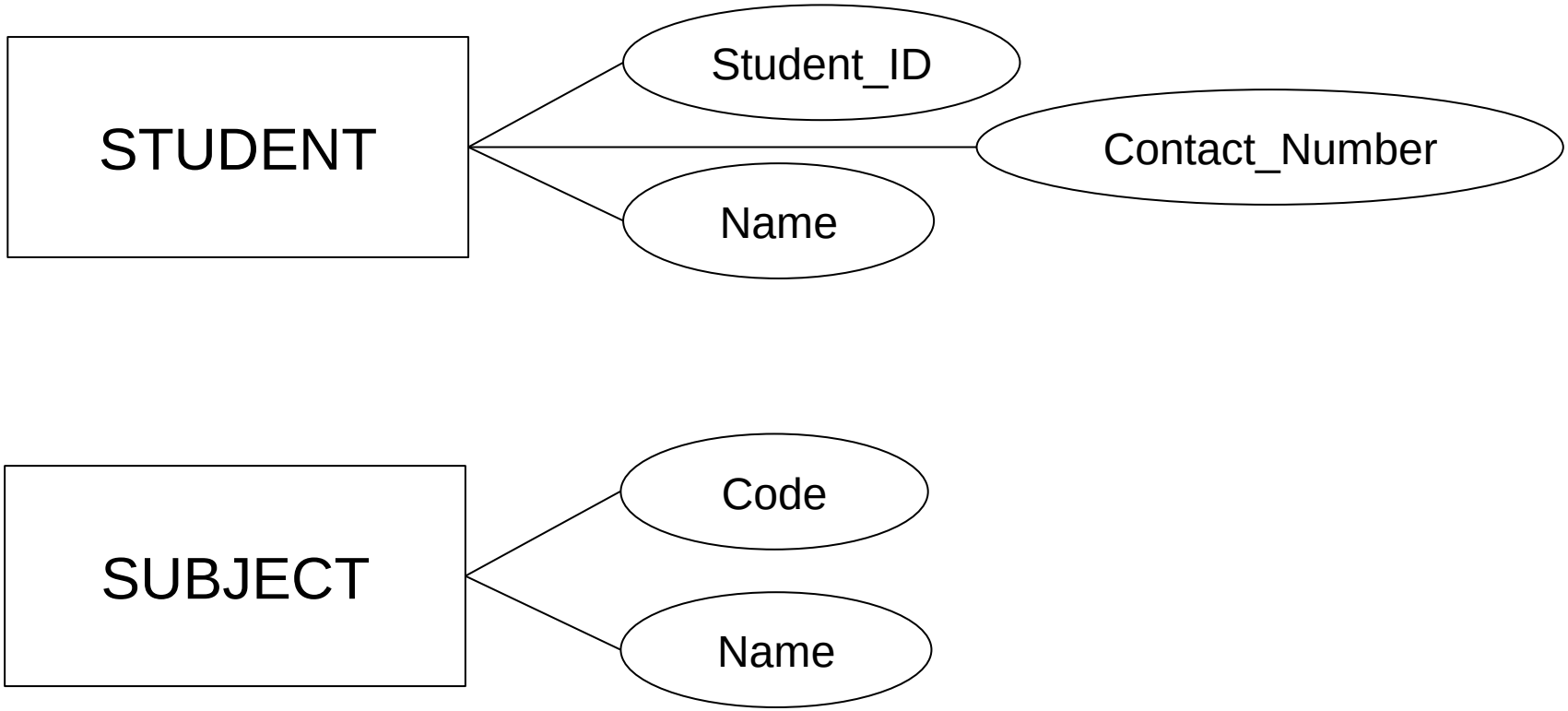
**Draw an Entity based on APU's database requirements**

APU has the following data needs:

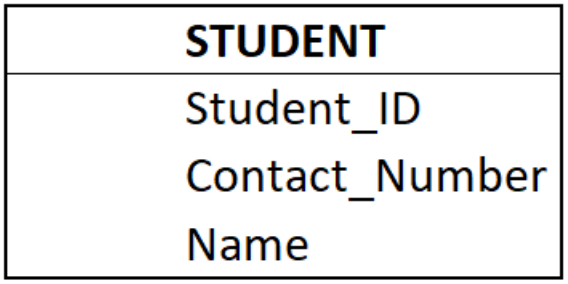
a. For each student, keep track of the student's student id, name and contact number.

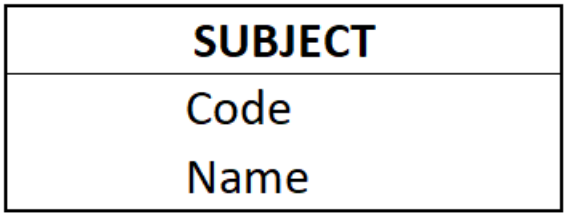
b. For each subject, keep track of the subject's subject code and name

# Draw Entity - Chen



# Draw Entity - Crow's Foot



# Types of Attribute



- Simple Attribute
- Composite Attribute
- Multi-Valued Attribute

# Simple Attribute

A ASIA PACIFIC UNIVERSITY OF TECHNOLOGY & INNOVATION

- A **simple attribute** is an attribute that **cannot be subdivided into more basic component attributes.**
- Example:

  Age (1, 2, 3,… )
  Gender (Male, Female)
  Marital Status (Single, Married,… )

# Multivalued Attribute

- A **multivalued attribute** is an attribute that can have **many values**.
- Example:

| Attribute | Value |
|---|---|
| Name | Will Smith |
| Gender | Male |
| Birthdate | 25/09/1968 |
| Phone Number | 011-XXXXXXX,<br>012-XXXXXXX |

Single-valued Attributes

Multivalued Attribute

# Composite Attribute

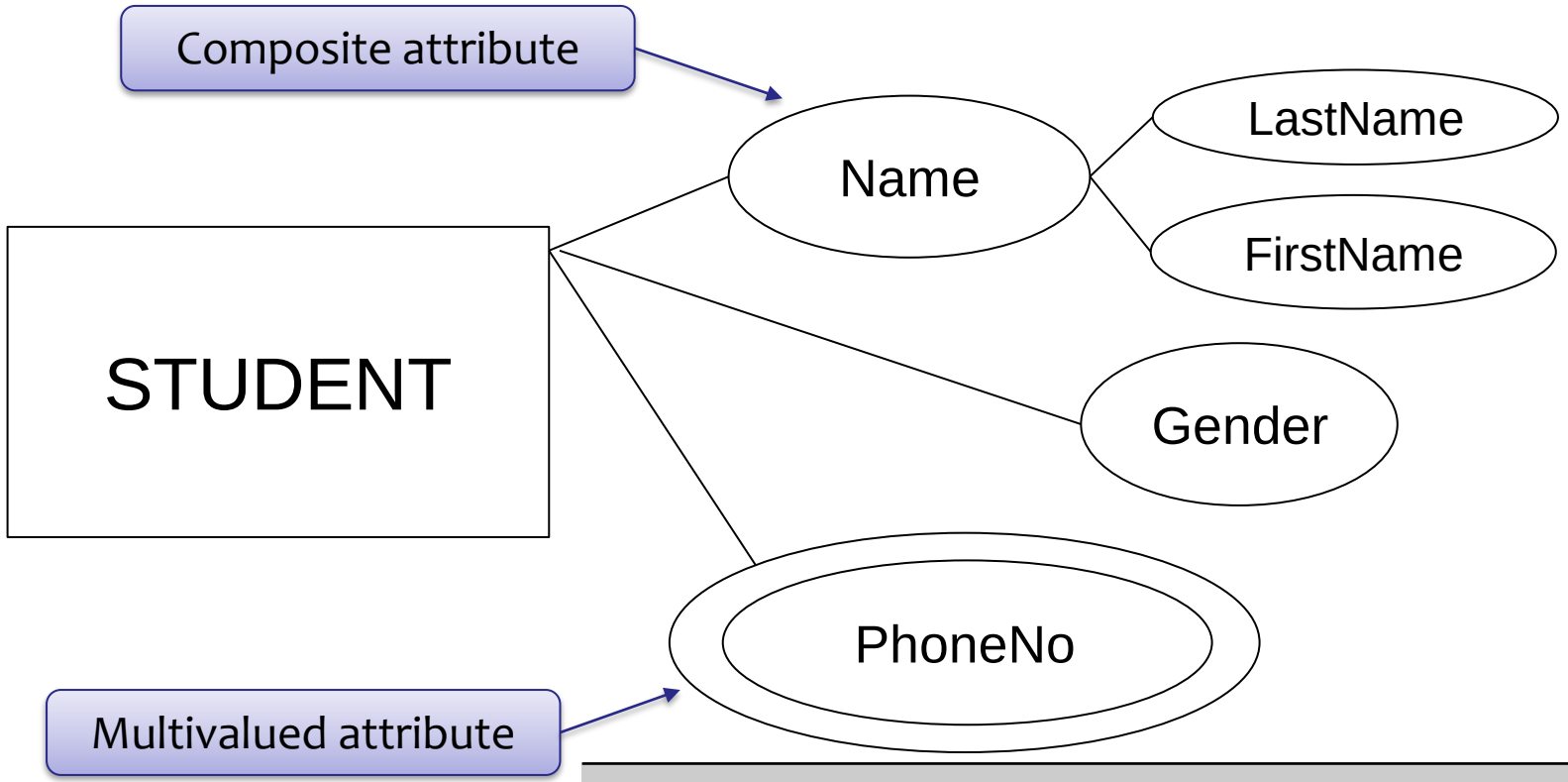
- A **composite attribute** is an attribute that can be **subdivided into more basic component attributes.**
- Example:

Name = First Name + Last Name
John Smith = John + Smith

Address = House/Building No + Road + Area + Postcode + City + State

APU = Jalan Teknologi 5, Taman Teknologi Malaysia, 57000 Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur

# Composite & Multivalued Attribute - Chen




| | |
|---|---|
| Composite attribute | Component attributes are attached by straight lines to the oval representing the composite attribute |
| Multivalued attribute | Double ovals as oppose to single oval |

# Composite & Multivalued – Crow's Foot

| STUDENT |
| --- |
| Lastname |
| Firstname |
| Gender |
| House Phone No |
| Mobile Phone No |

Break composite and multi-valued attributes into attributes with single value

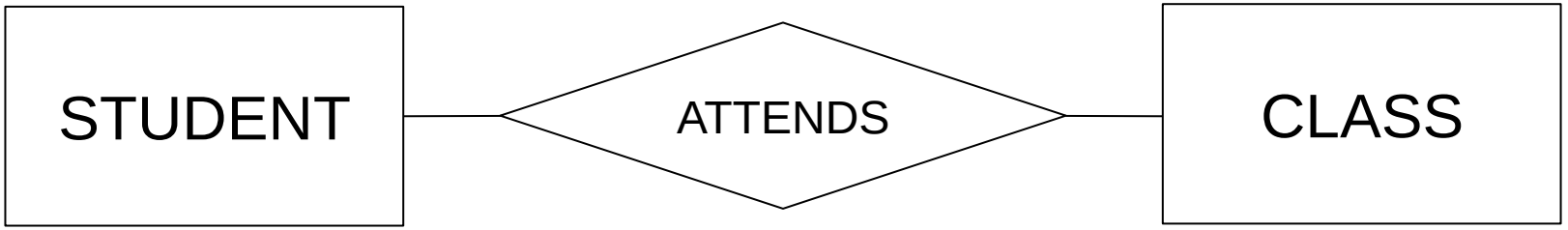
# Relationship

- A relationship is a meaning for the association between entities

| Entity Type | Relationship | Entity Type |
|---|---|---|
| STUDENT | **attends** | CLASS |
| LECTURER | **teaches** | CLASS |
| PILOT | **flies** | AIRPLANE |

# Relationship Notation in ERD

| | |
|---|---|
| Relationship | In Chen notation, **Diamond-shaped box** labelled with the name of the relationship<br><br>The box is attached to the participating entity types with straight lines<br><br>The relationship name should be in uppercase<br><br>In Crow's foot notation, straight line depict relationship |



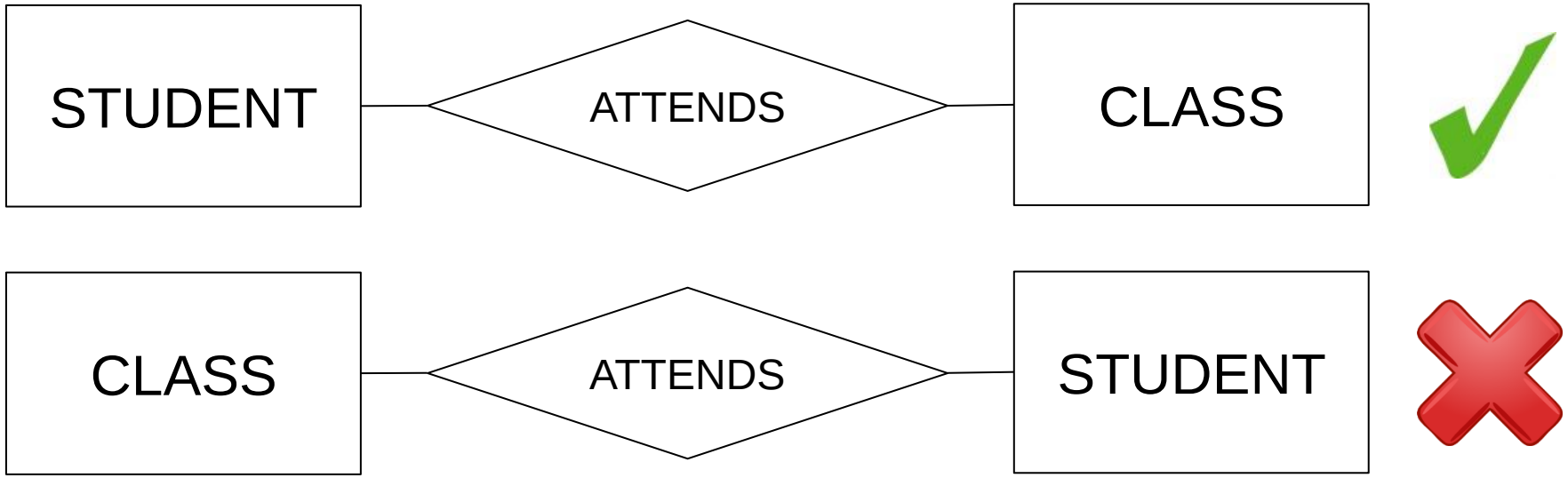

Read from left-to-right: Student attends class

Read from right-to-left: Class attended by student

Relationship is to be read from both directions

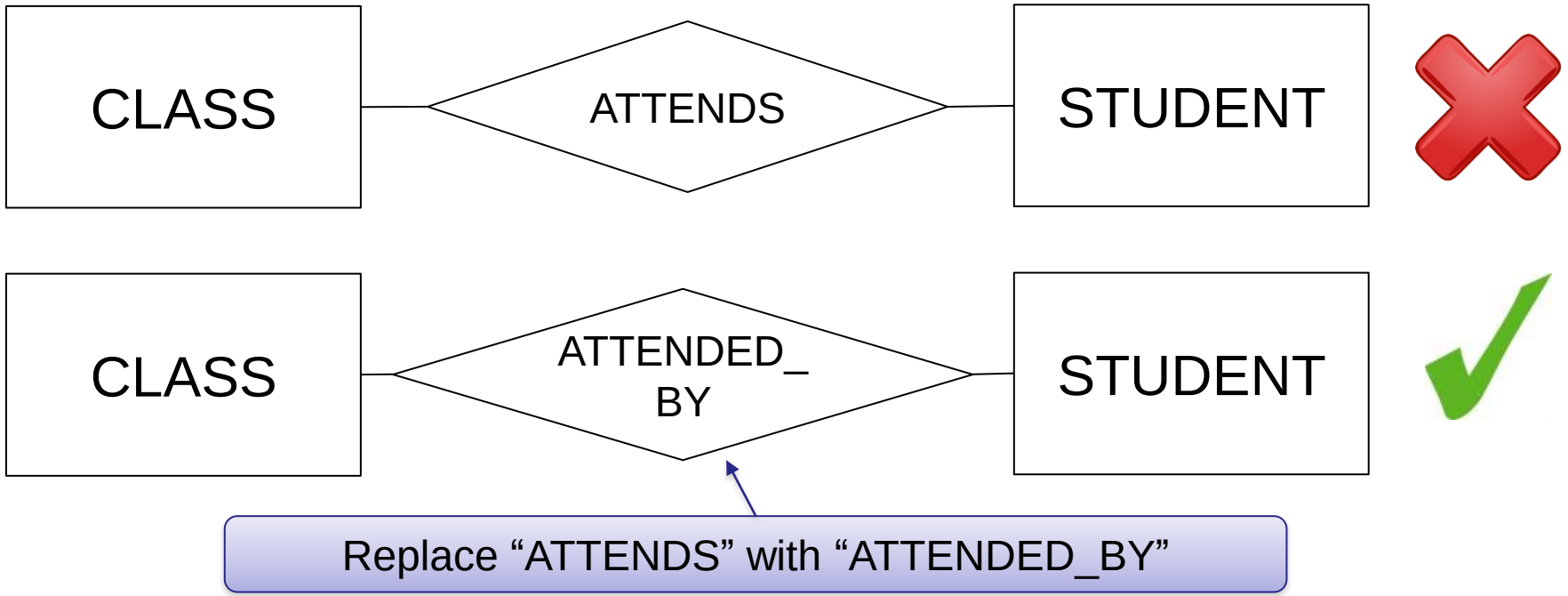
# Relationship Drawing Tips

- For the purpose of readability, always draw a relationship as if it is read from left-to-right.



# Relationship Drawing Tips

- For the purpose of readability, always draw a relationship as if it is read from left-to-right.



# Relationship Cardinality



- Cardinality expresses the **number of entities instances** that can be in a relationship.
- The cardinality of a relationship can be expressed as:

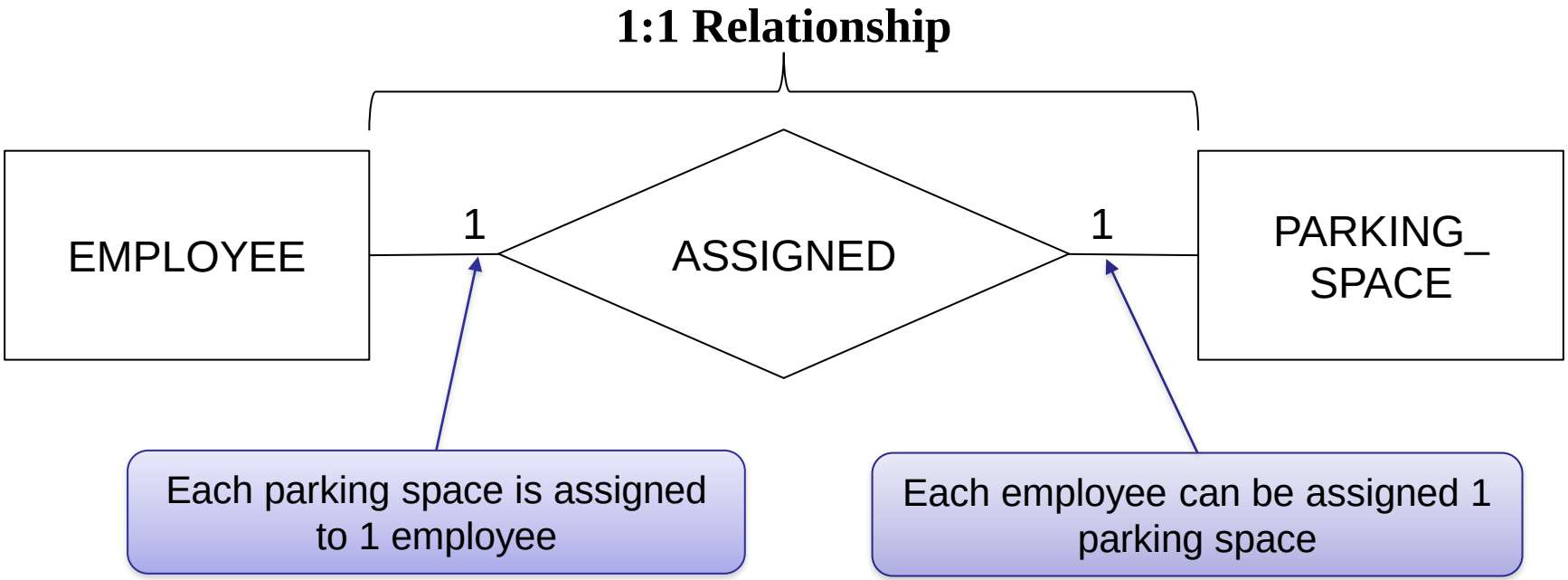
One-to-One

One-to-Many

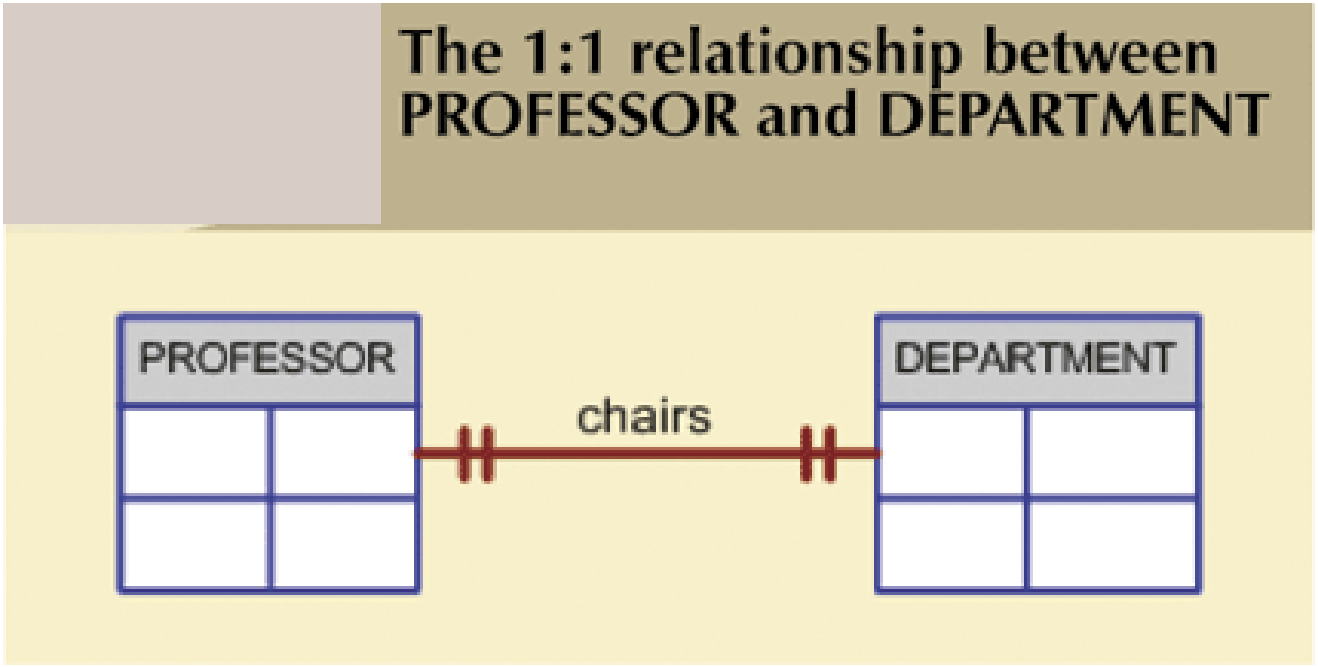
Many-to-Many

# One-to-One
# - Chen Notation

- If an entity in a relationship will have at most one instance of the related entity, this is known as one-to-one (1:1) relationship

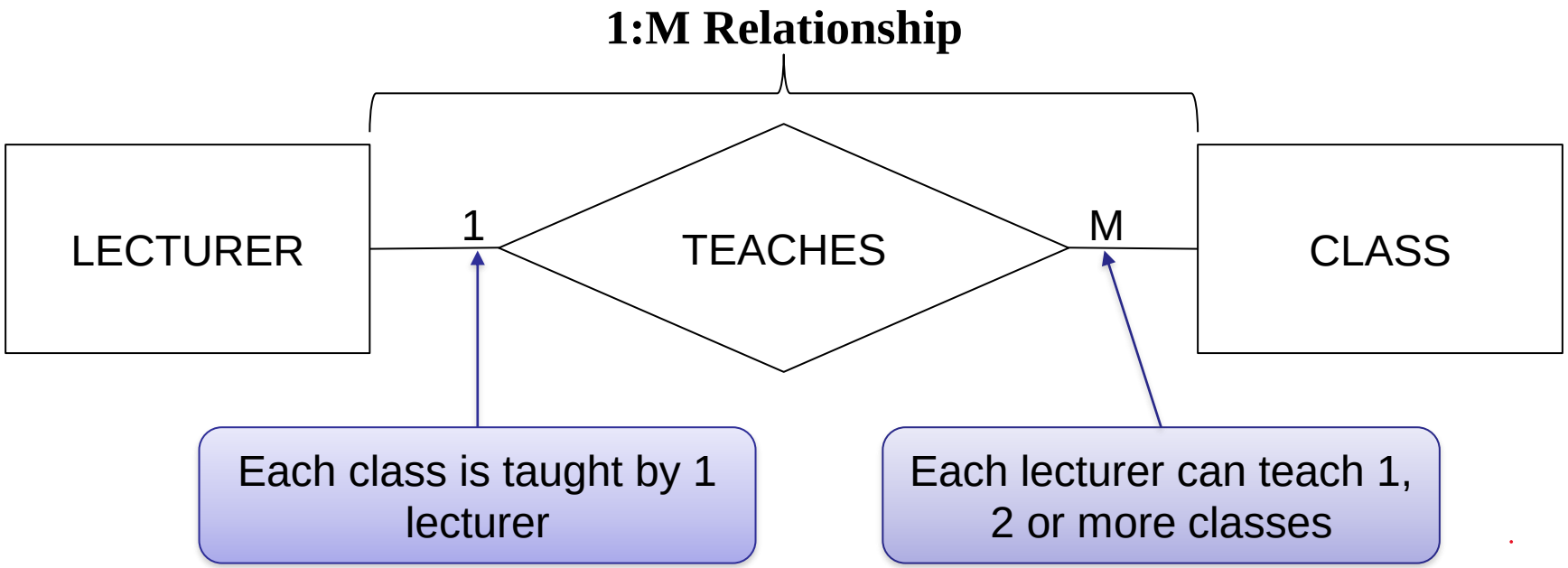


# One-to-One
# - Crow's Foot Notation

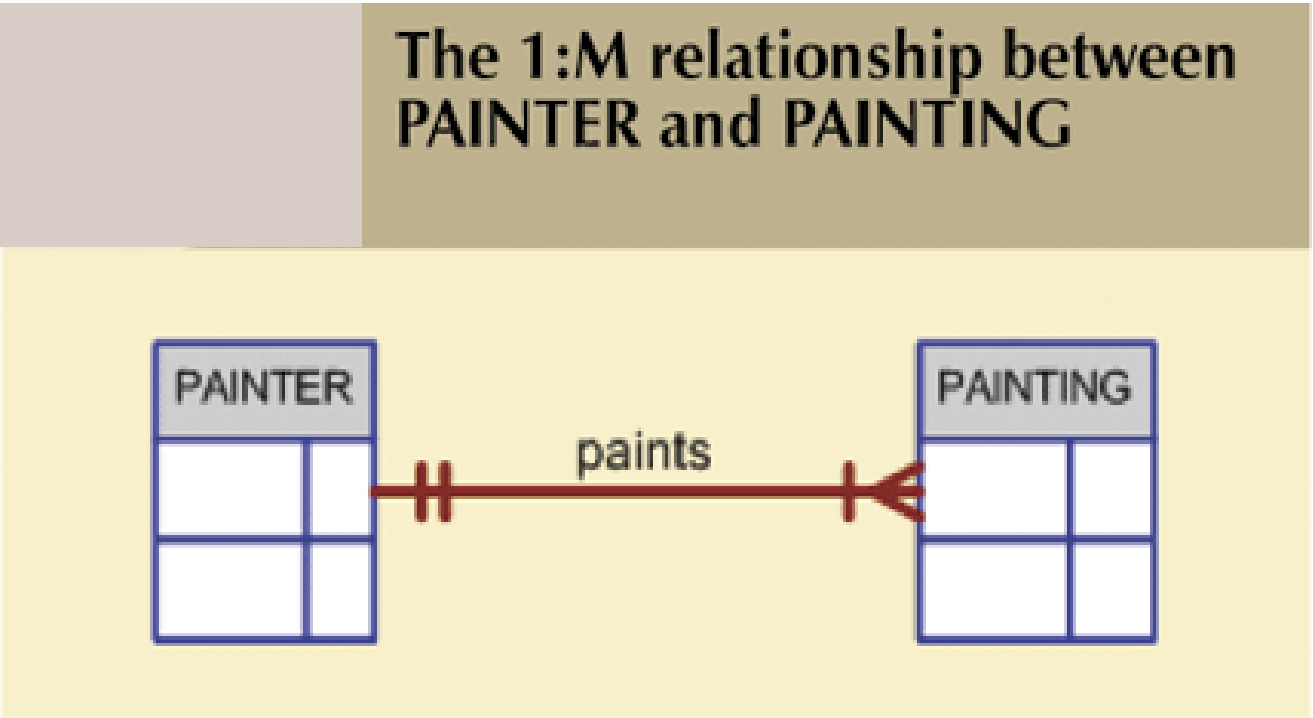


# One-to-Many

- If an entity A has many instances of entity B, but the B entity can have a maximum of only one instance of entity A, then we have one-to-many (1:M) relationship

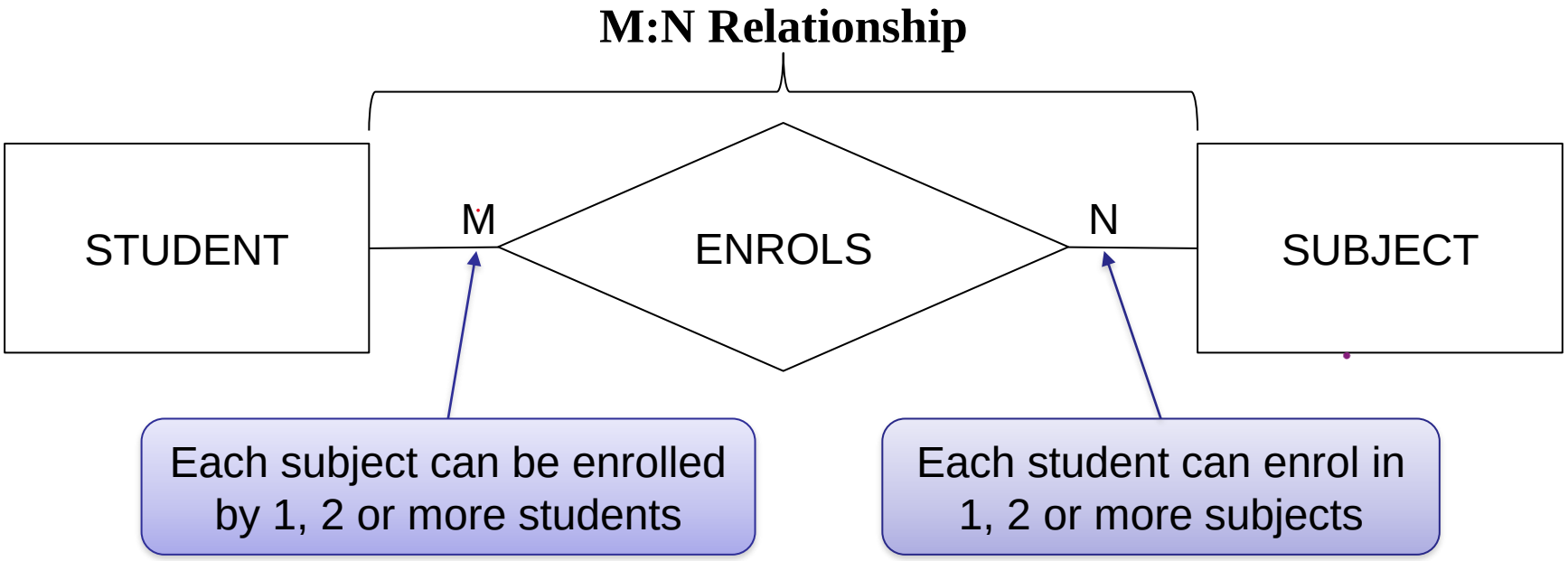


# One-to-Many


The 1:M relationship between PAINTER and PAINTING
PAINTER
paints
PAINTING

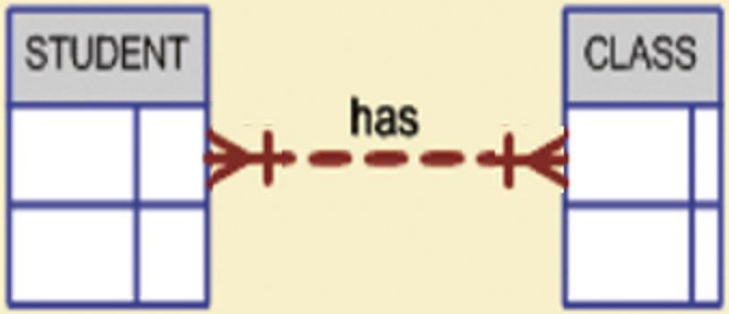
# Many-to-Many

- If entities on both sides of the relationship can have many instances of the other entity, then we have a many-to-many (M:N) relationship



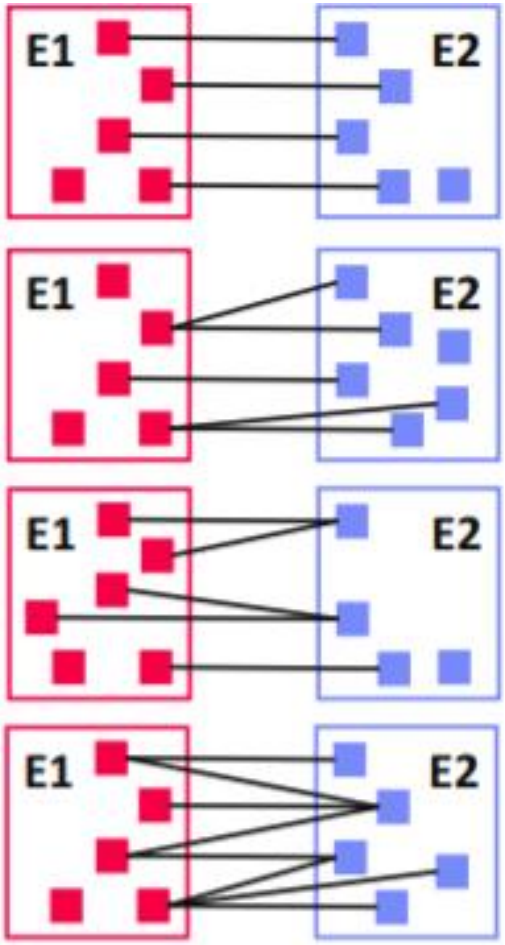
# Many-to-Many

The ERD's M:N relationship between STUDENT and CLASS



# Cardinality Summarized

- **1:1 (one-to-one)** ⟷
  - Each e1 relates to at most one e2
  - Each e2 relates to at most one e1
- **1:N (one-to-many)** ⟵
  - Each e1 relates to many e2 (0,1,…N)
  - Each e2 relates to at most one e1
- **N:1 (many-to-one)** ⟶
  - Symmetric to 1:N
- **N:M (many-to-many)**
  - Each e1 relates to many e2 (0,1,…M)
  - Each e2 related to many e1 (0,1,…N)

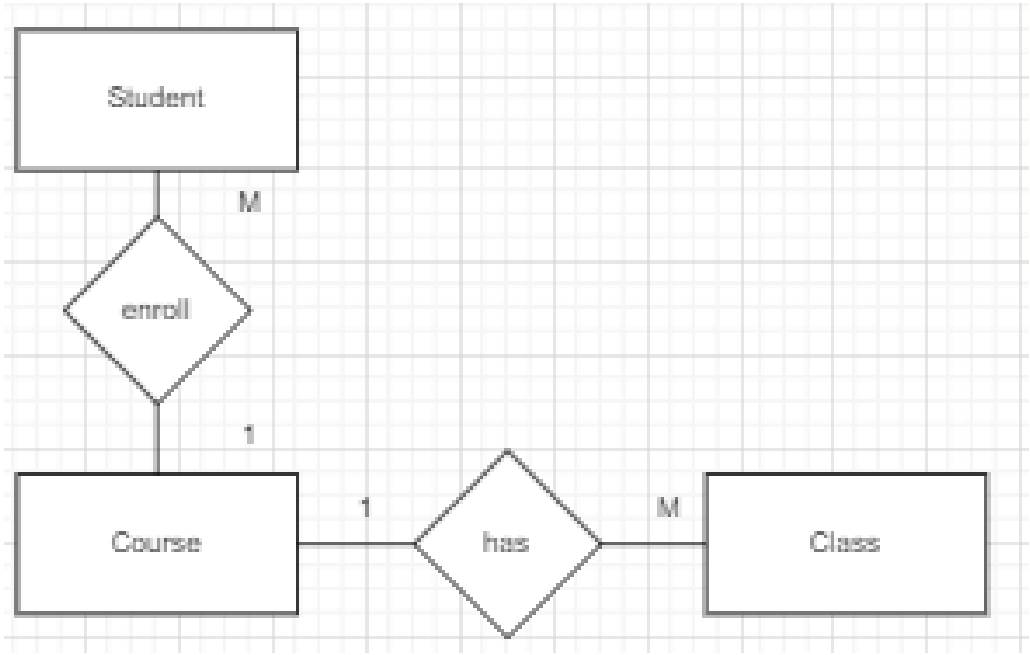


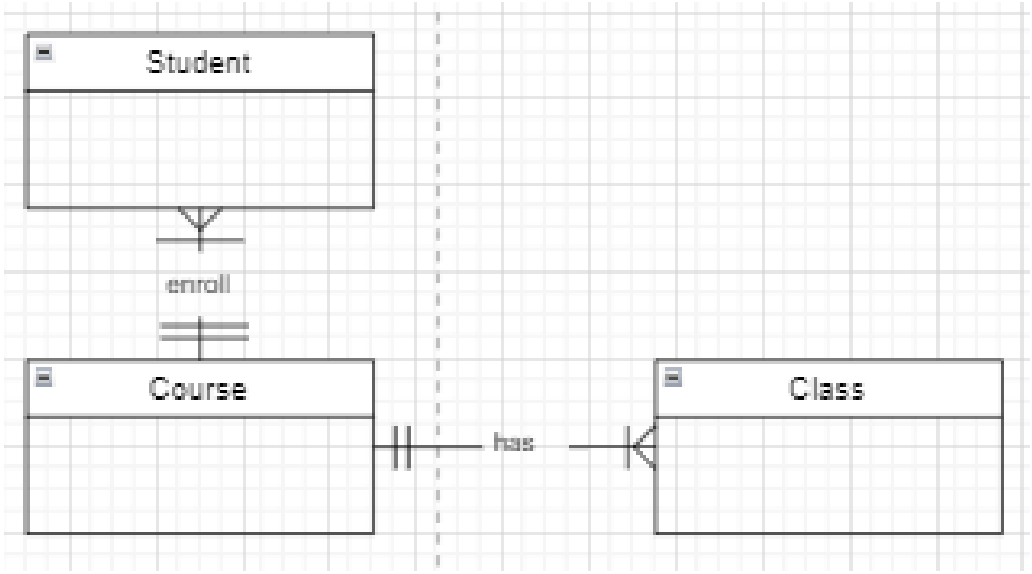
# Activity

Using Chen's notation and Crow's Foot notation, draw entity relationship model for the following scenario.

A Course has one or many classes; each class belongs to one course.
A student can enroll in only one course; each course can be enrolled by many students.

Chen



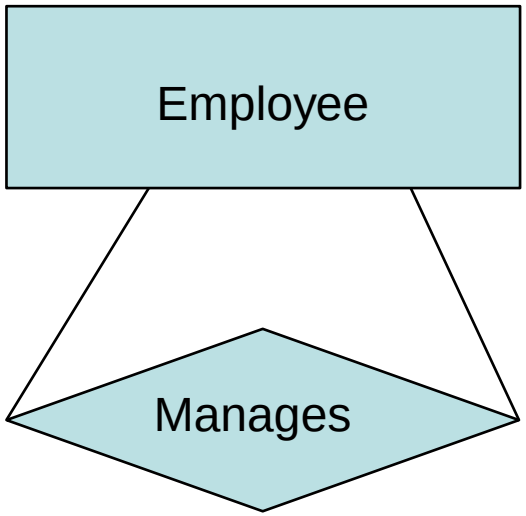




Crow's Foot

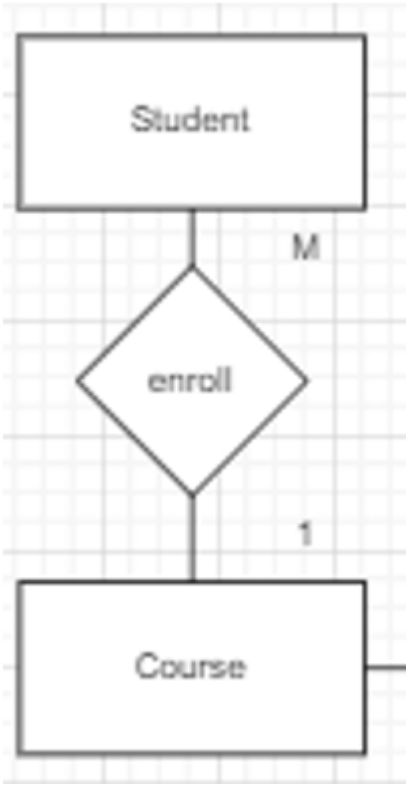
# Relationship Degree

- Indicates number of associated entities in a relationship
- Unary relationship (recursive)
  - Association is maintained within single entity
- Binary relationship
  - Two entities are associated
- Ternary relationship
  - Three entities are associated
- Quaternary relationship
  - Four entities are associated

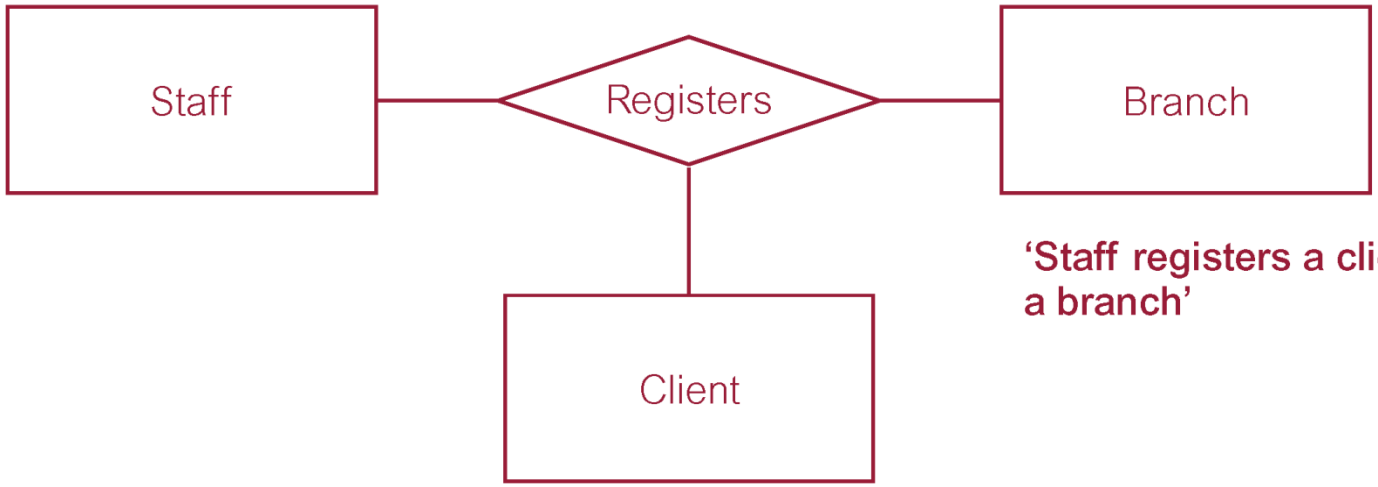
# Unary Relationship



# Binary Relationships

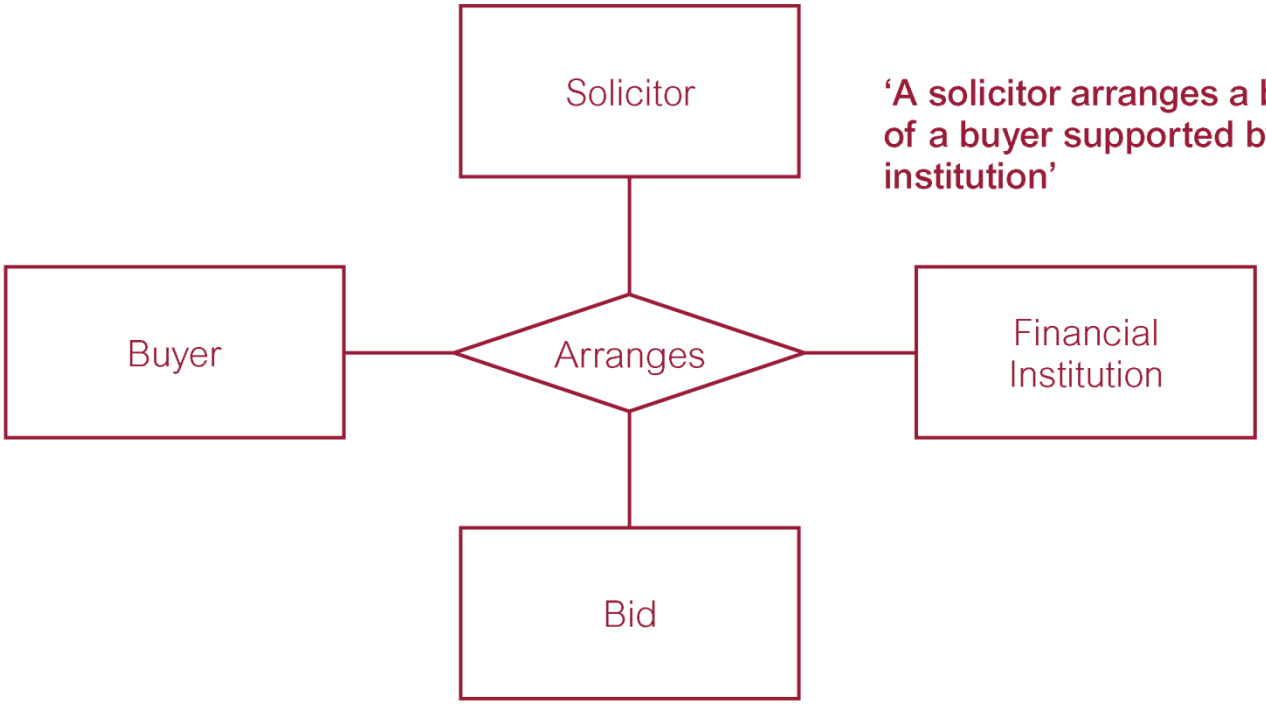


# Ternary Relationship called *Registers*




'Staff registers a client at a branch'

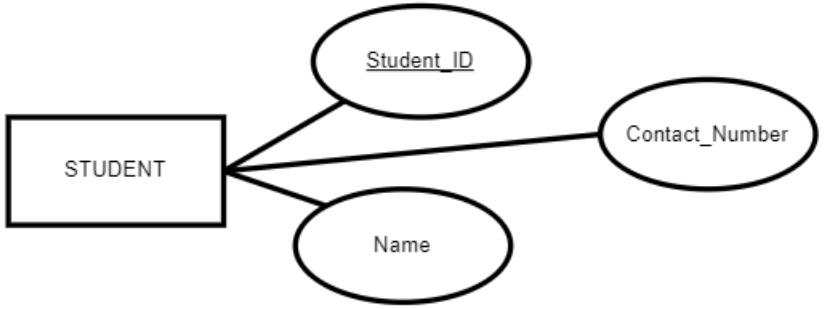
# Quaternary Relationship called *Arranges*




**'A solicitor arranges a bid on behalf of a buyer supported by a financial institution'**

# Primary Key in ERD

| | |
|---|---|
| Primary Key | In Chen notation, **Underline** the attribute name.<br>Straight line<br>In Crow's foot notation, **labelling** is sufficient. |



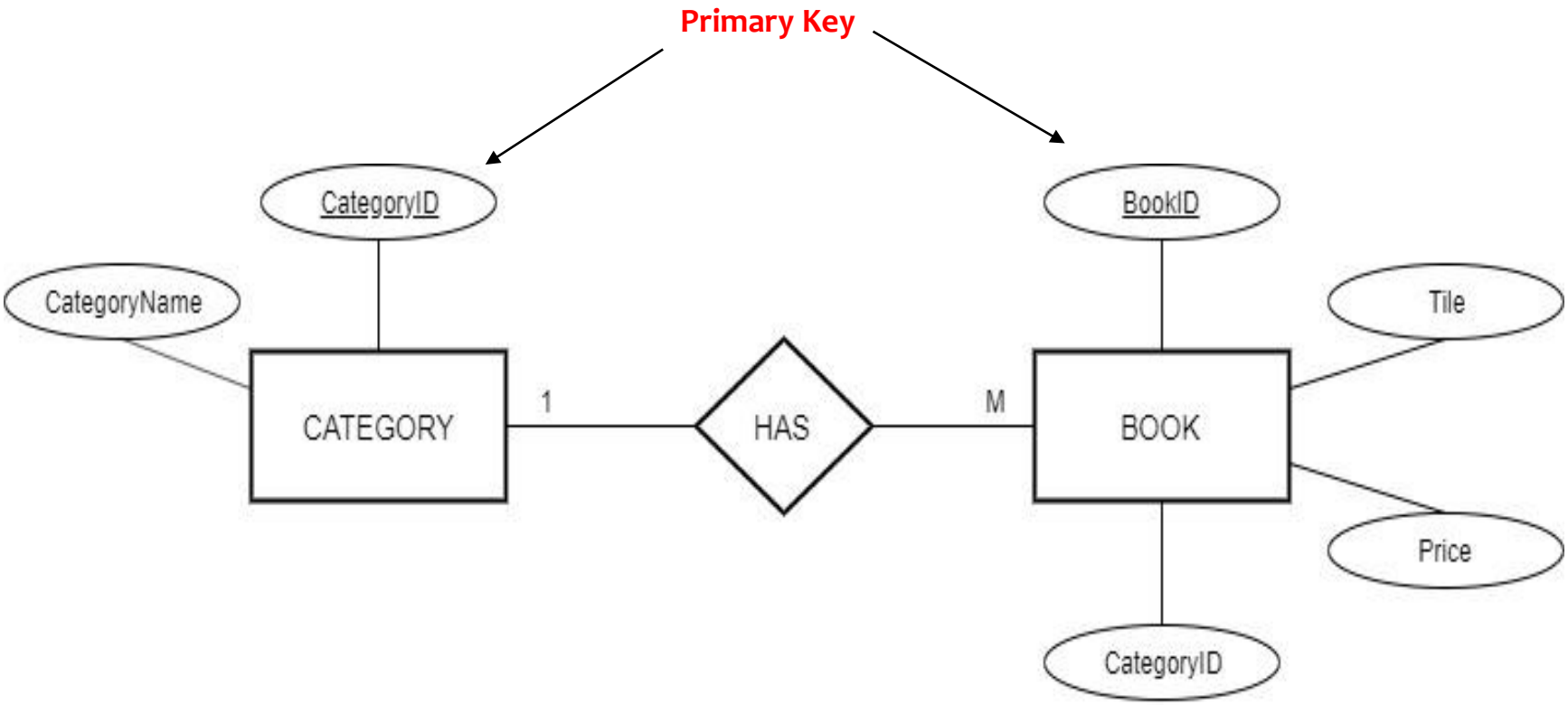

Chen

| STUDENT | |
|---|---|
| PK | Student_ID |
| | Contact_Number |
| | Name |

Crow's Foot

A primary key is **an attribute** or **set of attributes** that can be used to **uniquely identify an entity instance**

# Notation of Primary Key in ERD

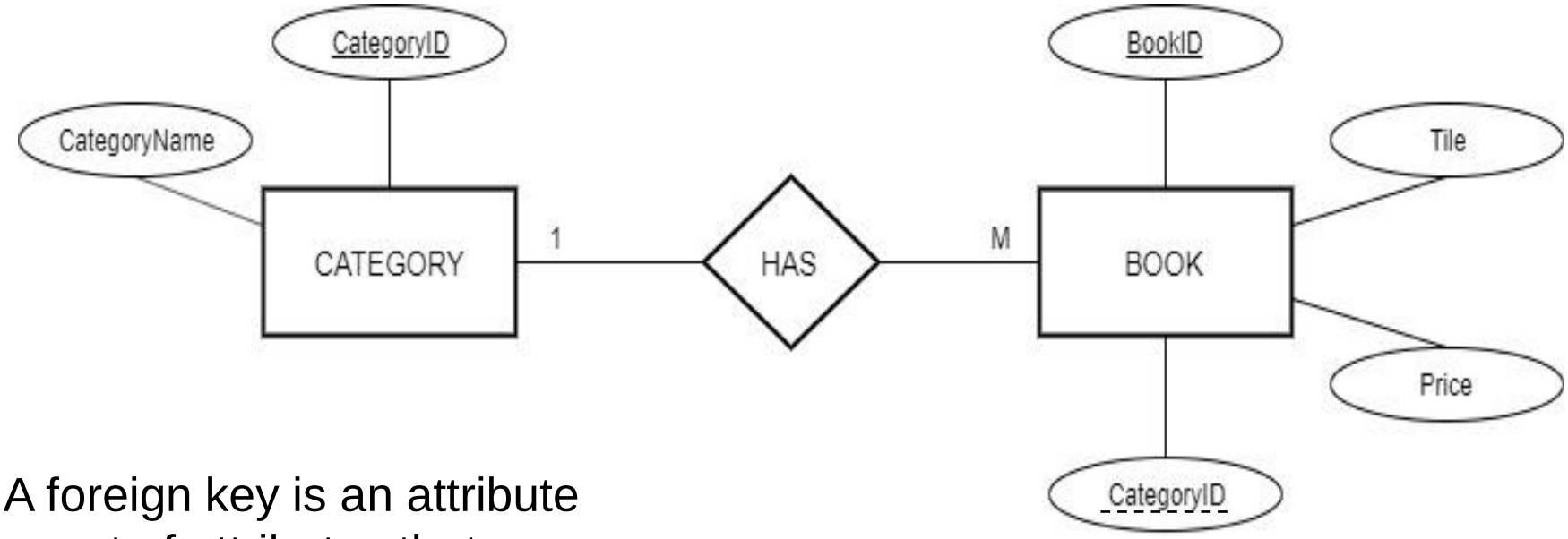


# Composite Primary Keys

A·P·U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

- Primary keys ideally composed of only single attribute
- Possible to use a composite key
  - Primary key composed of more than one attribute

# Foreign Key in ERD



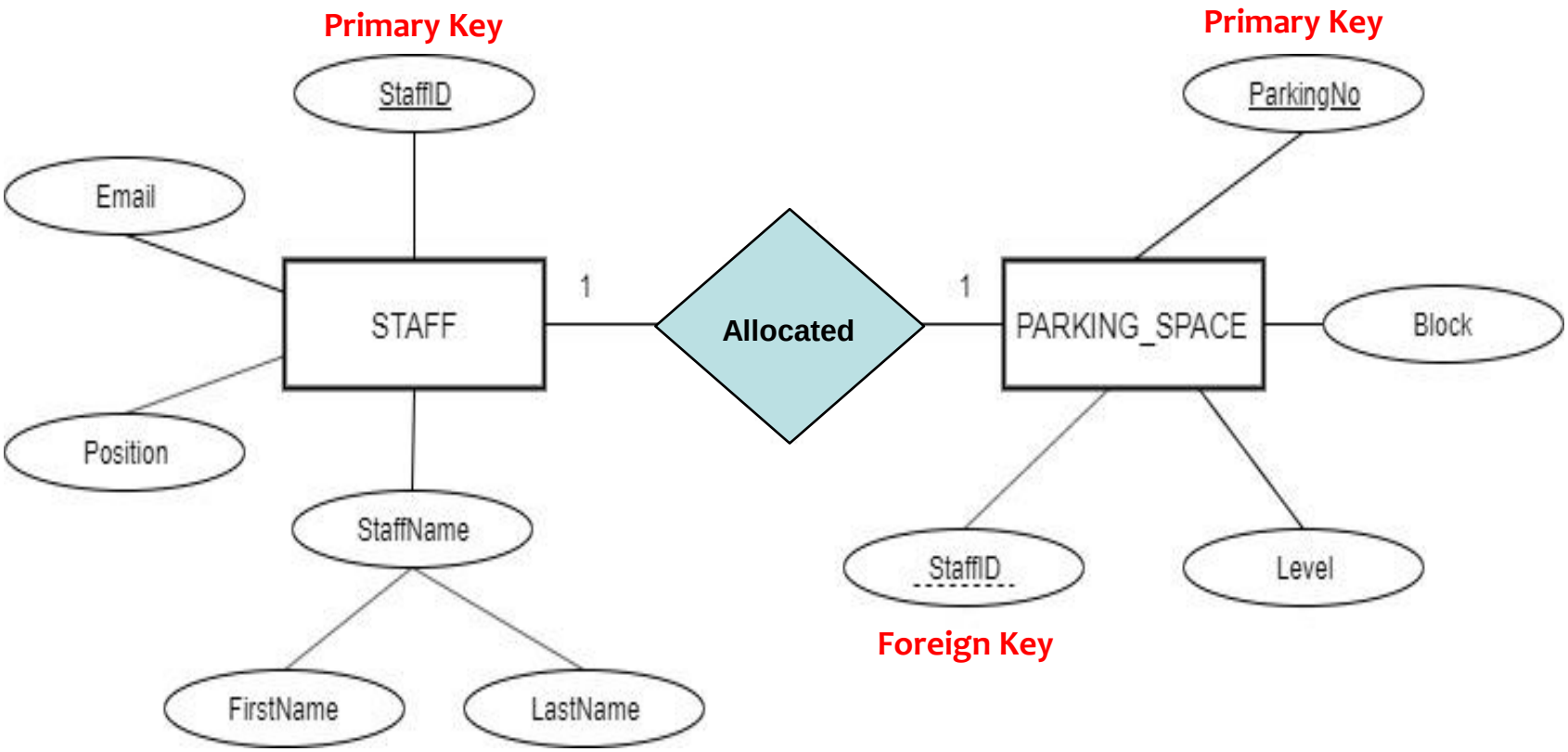

A foreign key is an attribute or set of attributes that can be **used to link two entities together**.

**Foreign Key**

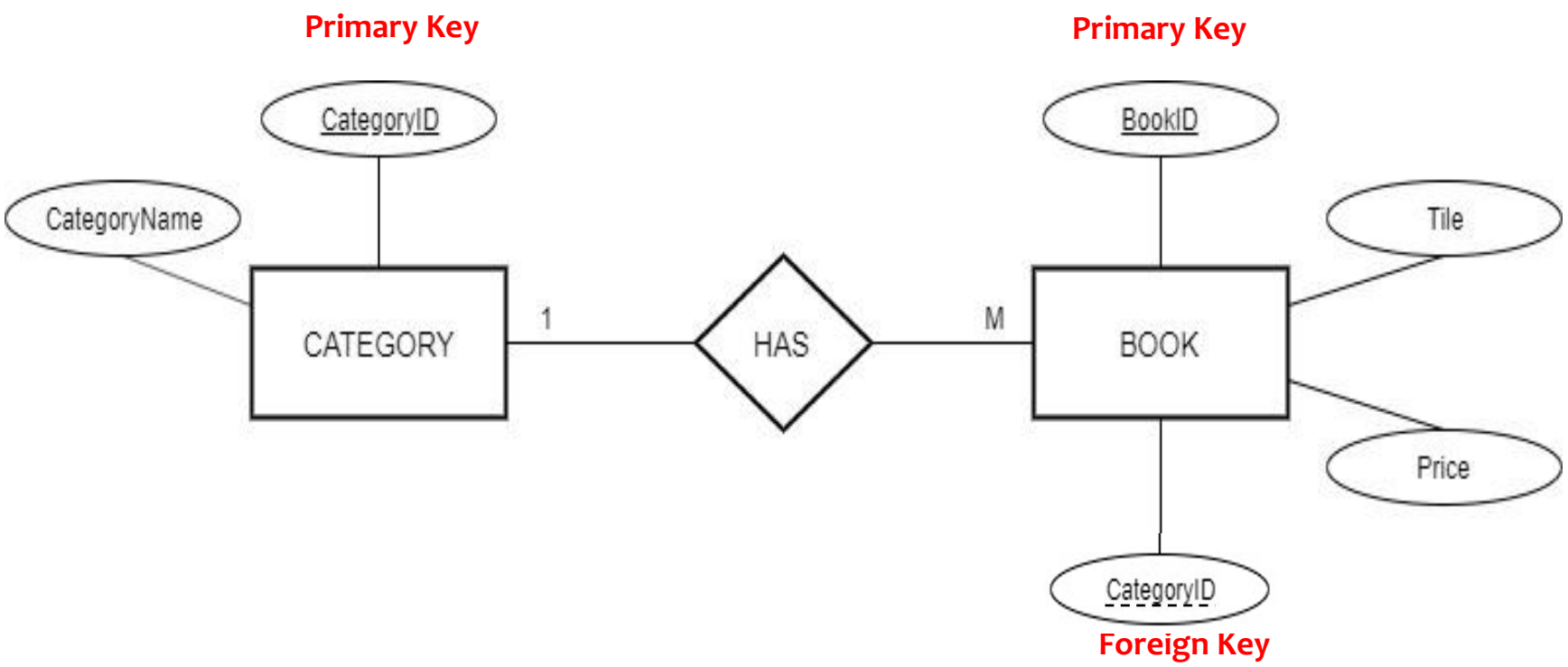
# Foreign Key Placement in ERD

- 1:1 relationship: Place the foreign key at either of the two entities.
- 1:M: Place the foreign key at entity that is on the MANY side of the relationship.
- M:N: Place the foreign keys on the ***relationship/composite entity.***

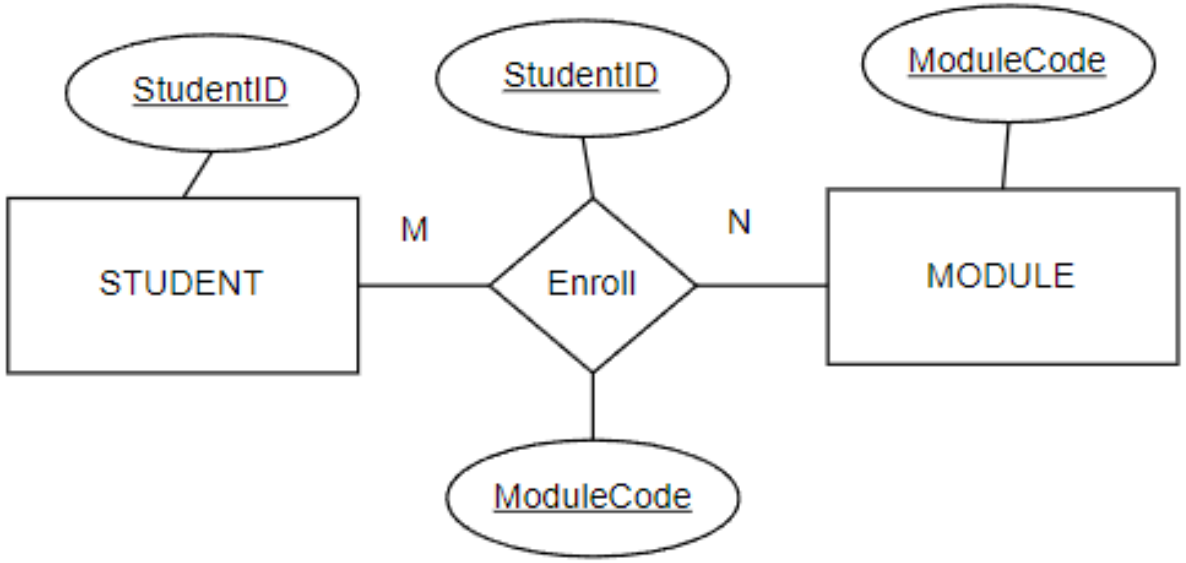
# Foreign Key Placement in ERD (1:1)



# Foreign Key Placement in ERD (1:M)



# Foreign Key Placement in ERD (M:N)

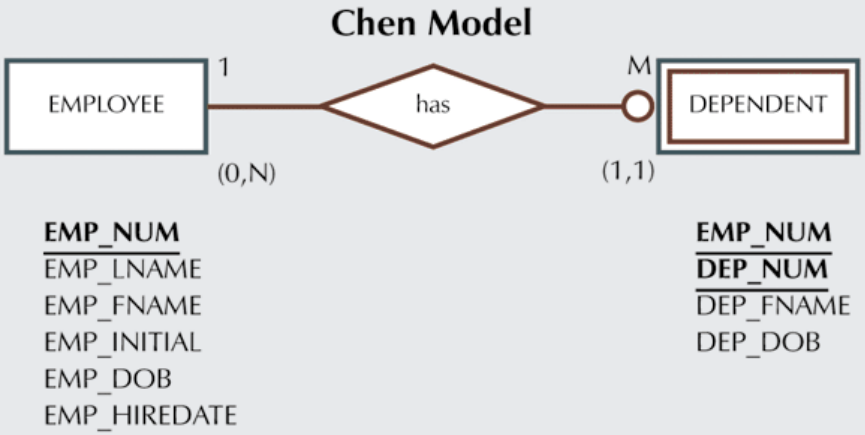


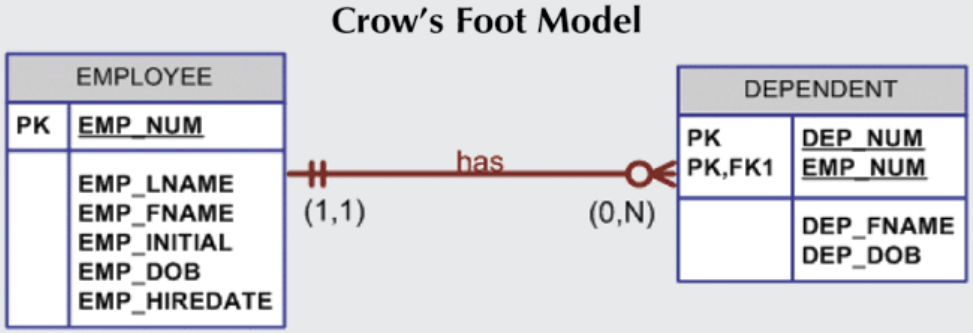
# Weak Entities

- Weak entity meets two conditions
  - Existence-dependent
    - Cannot exist without entity with which it has a relationship
  - Has primary key that is partially or totally derived from parent entity in relationship
- Database designer usually determines whether an entity can be described as weak based on business rules

# Weak Entities

**A weak entity in an ERD**



# Summary of Main Teaching Points

A·P·U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

- Entity
- Attributes
- Relationship

# Question and Answer Session

A·P·U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

Q & A