Database basics

First section

Lecturer : Shaimaa Sayed

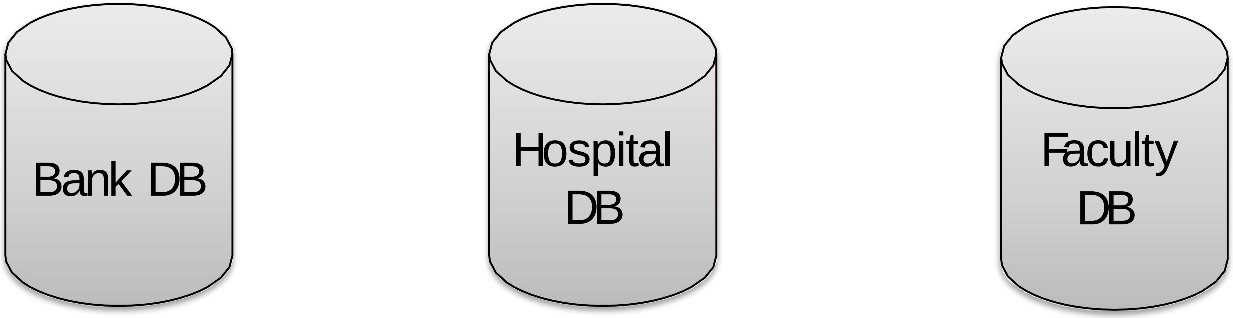
Part 1

- What is database (DB) ?
- Database Users
- Database Management Systems (DBMS)

What is database (DB)?

- A **database** is a collection of related data.

most of us encounter several activities every day that involve some interaction with a database. For example, if we go to the bank to deposit or withdraw funds, if we make a hotel or airline reservation.

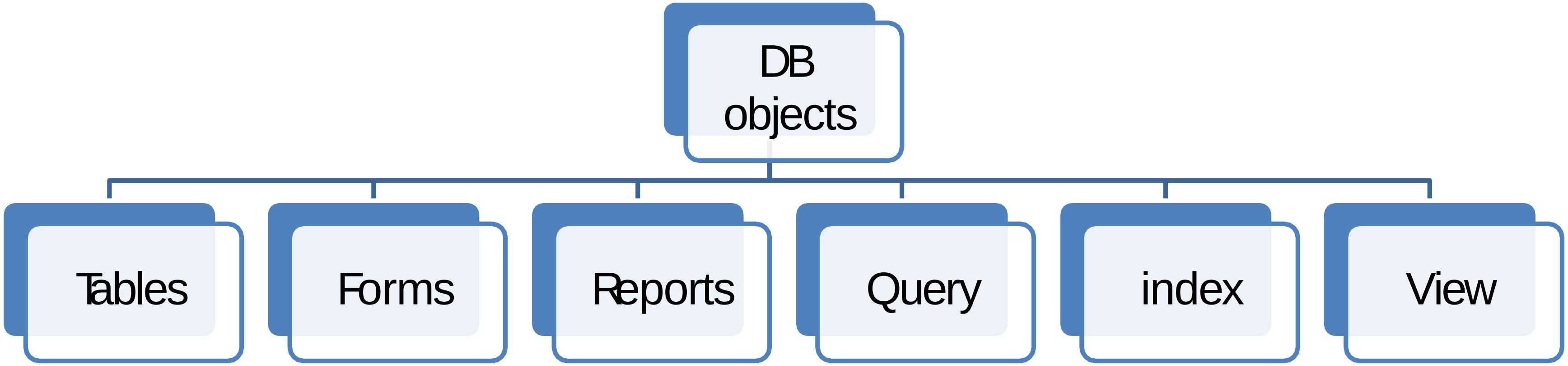


Bank DB

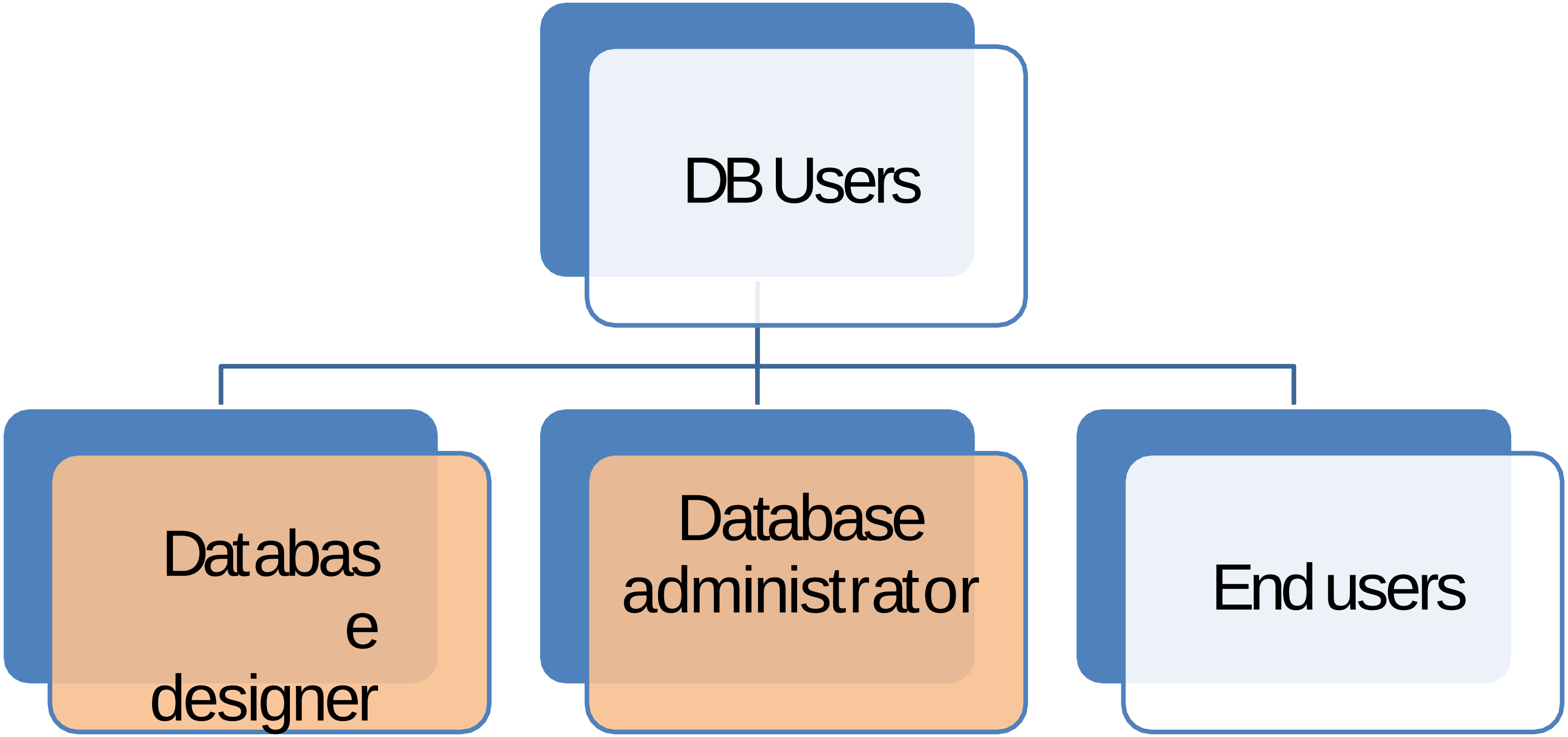
Hospital DB

Faculty DB

Database Objects



Database users


DB Users

Database designer

Database administrator

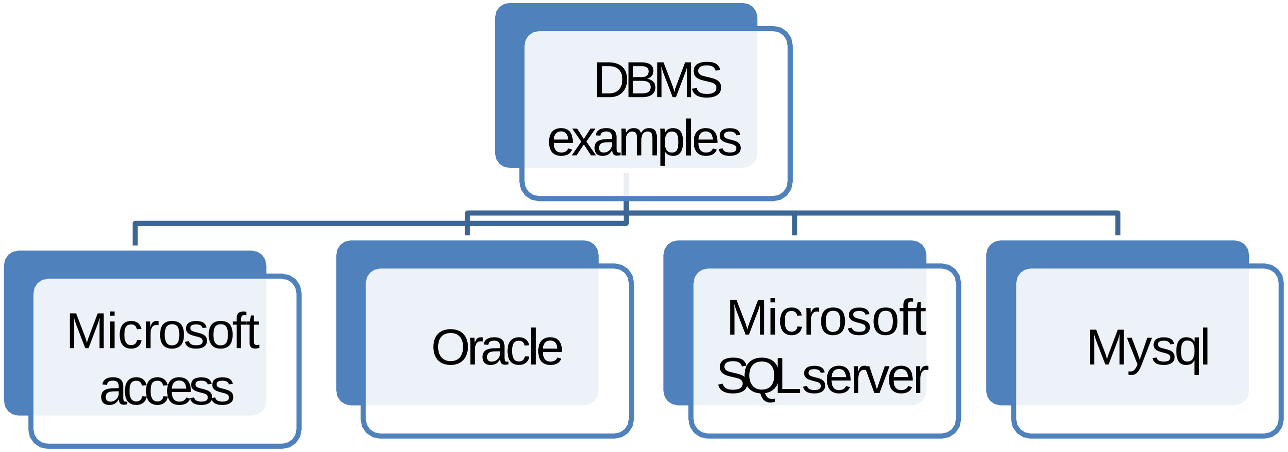
End users

Database users

- •**Database administrators :** They create users access and apply limitation to maintain isolation and force security. Administrators also look after DBMS resources like system license, software application and tools required and other hardware related maintenance.
- •**Data base designers :** This is the group of people who actually works on designing part of database. The actual database is started with requirement analysis followed by a good designing process. They people keep a close watch on what data should be kept and in what format. They identify and design the whole set of entities, relations, constraints and views.
- •**End Users**: This group contains the persons who actually take advantage of database system.

Database management systems (DBMS)

- DBMS is a computer software application that interacts with the user
- A general-purpose DBMS is designed to allow (1- data manipulation , 2- data control)



Part 2

Steps to built database

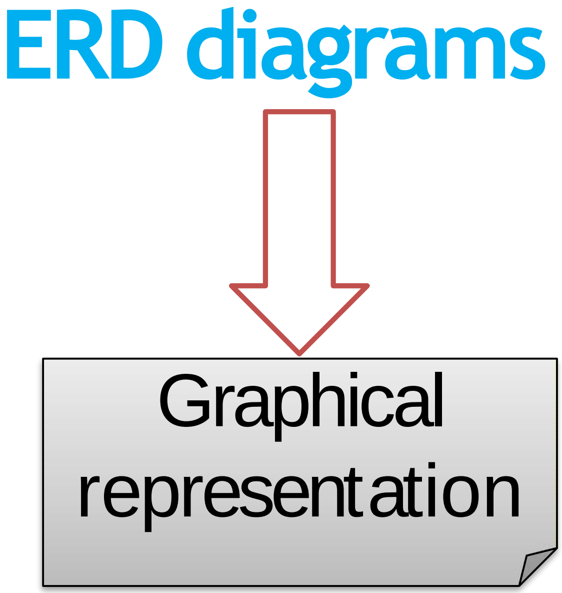
Step 1

- Define your target !!

For example: create database for faculty organization

Step 2

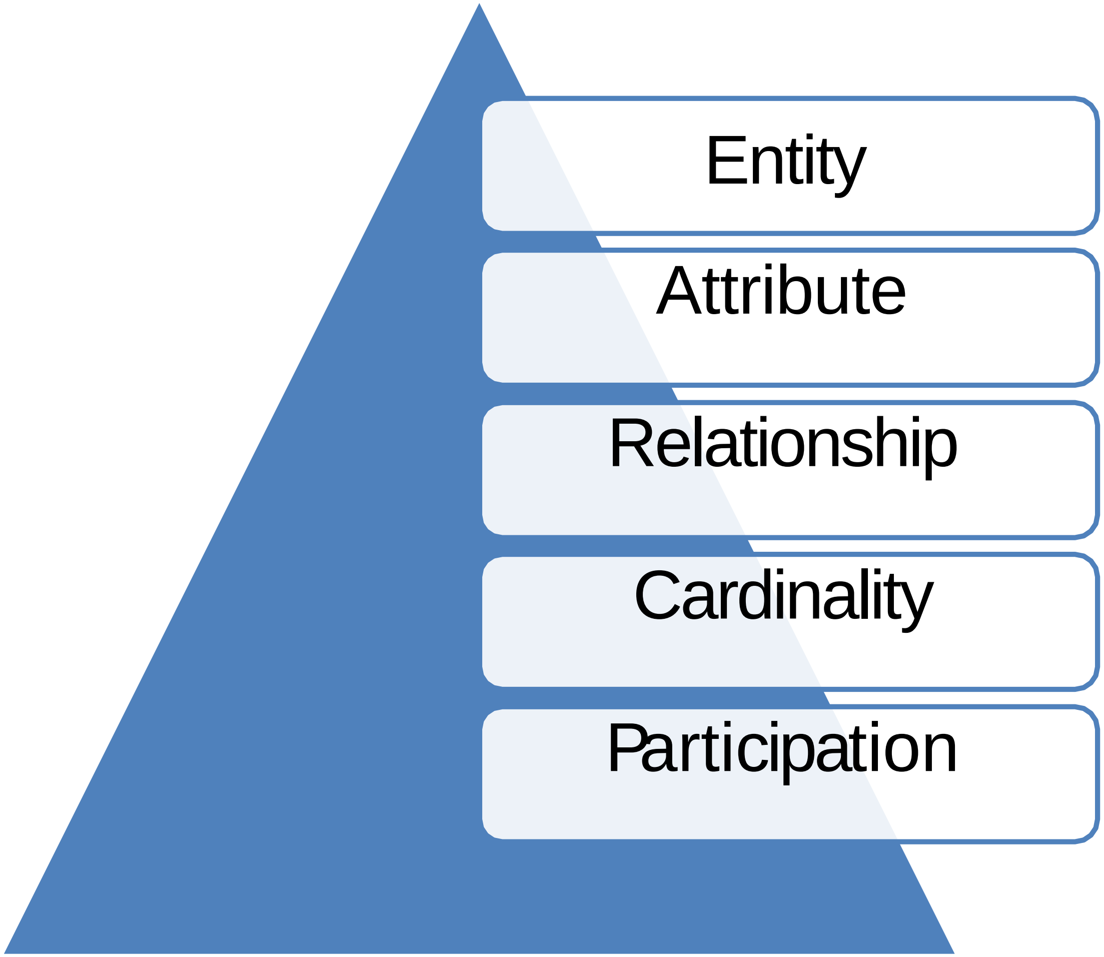
- Database design using Entity Relationship data model (ERD)



ERD diagrams

Graphical representation

Steps of ERD



Entity

Attribute

Relationship

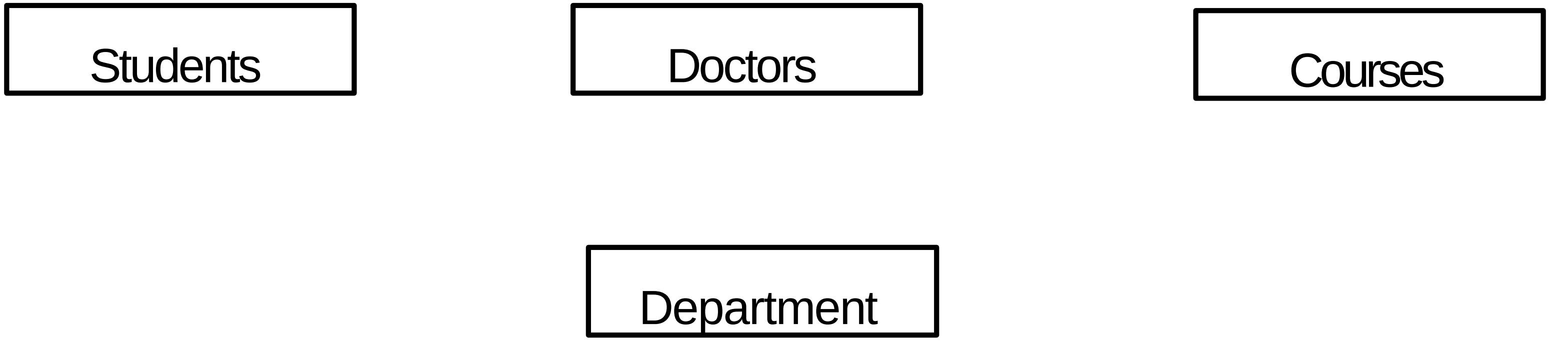
Cardinality

Participation

Entity Relationship data model (ERD) for faculty organization

- Entity

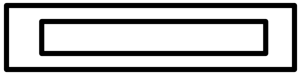
Entity examples :



Entity types

1-strong entity

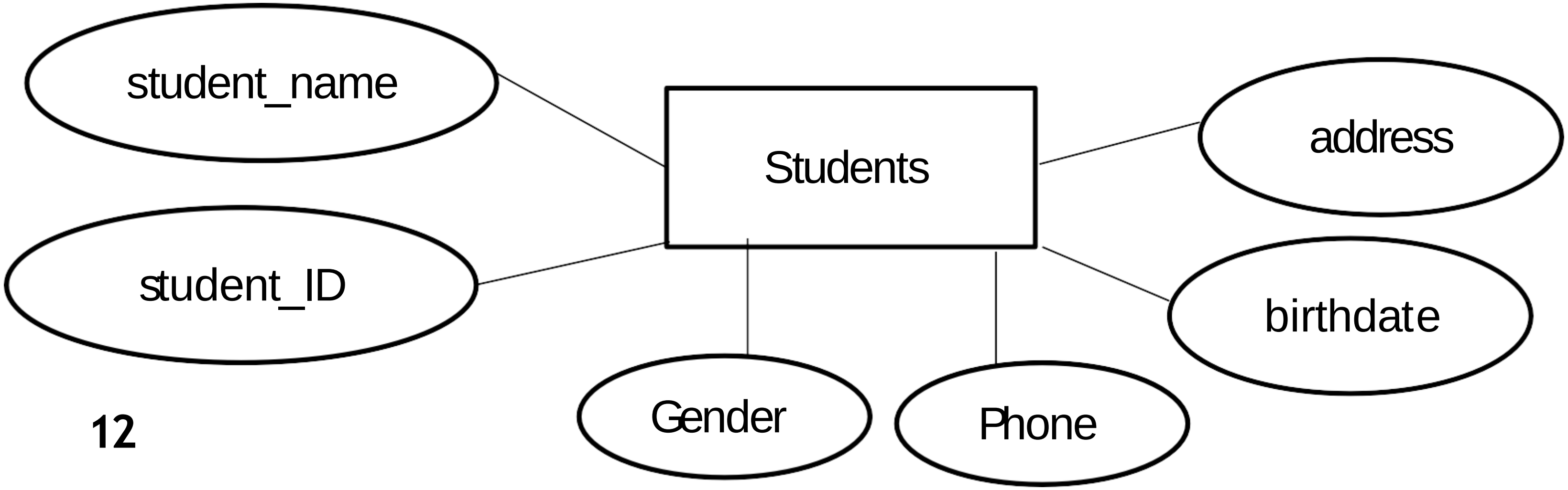
Every entity considered strong

2-weak entity 

Entity based on another entity

Attributes

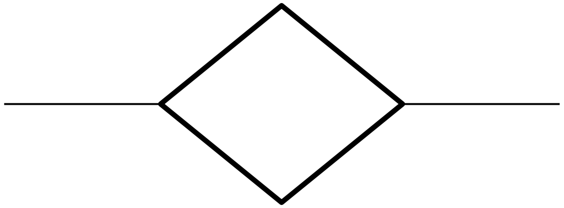
- Attribute For ⬭ examples : student_name,address,birthdate,phone, gender



Attributes types

- Simple attribute
- Composite attribute
- Derived attribute =calculate attribute
- Multi-valued attribute
- Complex attribute

Relationship and its rules



- Relationships

Note : Select verb to representrelationship

rules of relationship

1-degree of relationship

* Unary

* binary

* ternary

Cardinality of relationship

Types of relationships:

1- one to one

2- one to many or many to one

3-many to many

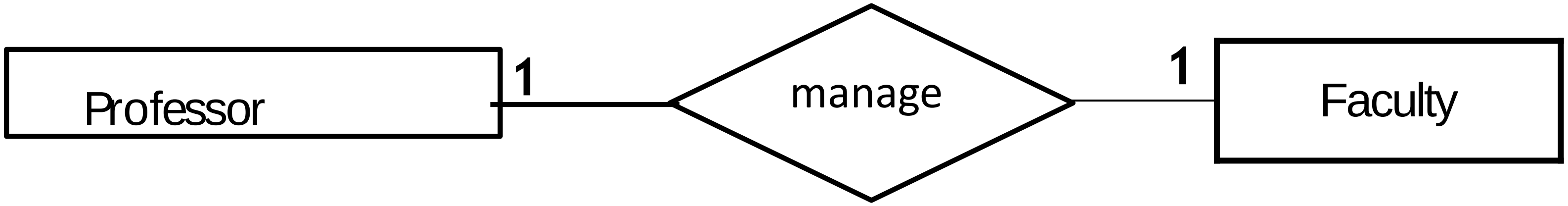
Participation of relationship

- Partial
- Total or full participation

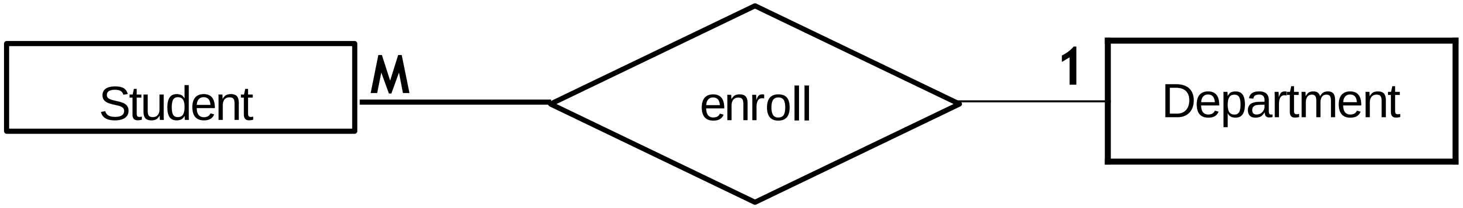
Entity Relationship data model (ERD) for faculty organization

- relationships examples :

1- one to one

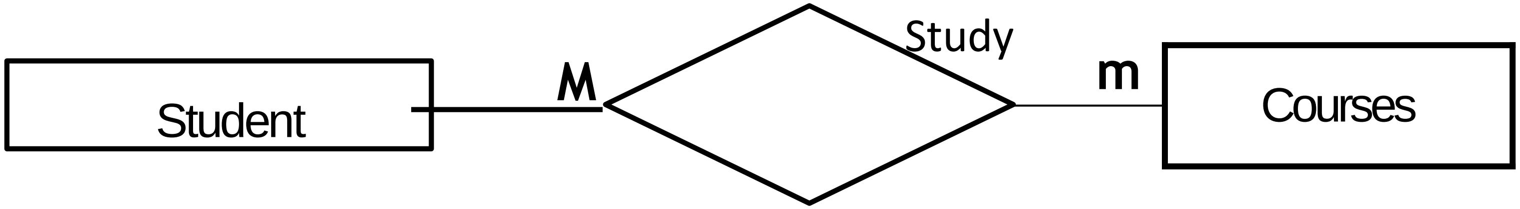


2- one to many



Entity Relationship data model (ERD) for faculty organization

3- many to many



Summary of notation for ER diagrams

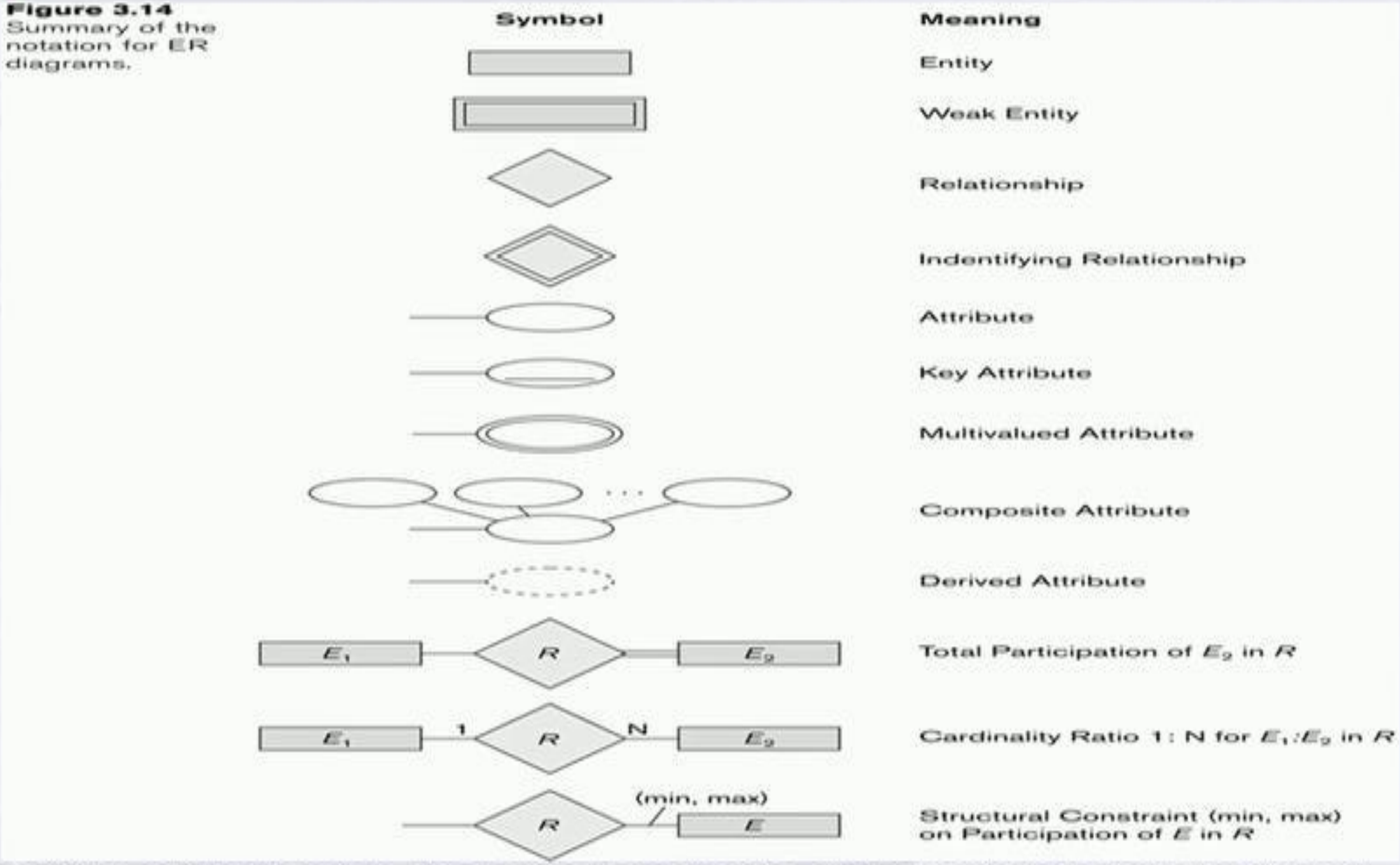


Figure 3.14
Summary of the notation for ER diagrams.

Meaning

- Entity
- Weak Entity
- Relationship
- Indentifying Relationship
- Attribute
- Key Attribute
- Multivalued Attribute
- Composite Attribute
- Derived Attribute
- Total Participation of E_2 in R
- Cardinality Ratio 1: N for E_1:E_2 in R
- Structural Constraint (min, max) on Participation of E in R

Identifying relationship is links strong entities to weak entities and represented with double line diamond

different types of Keys

- Candidate key
- Primary key
- Foreign key
- Composite key

Types of constraints

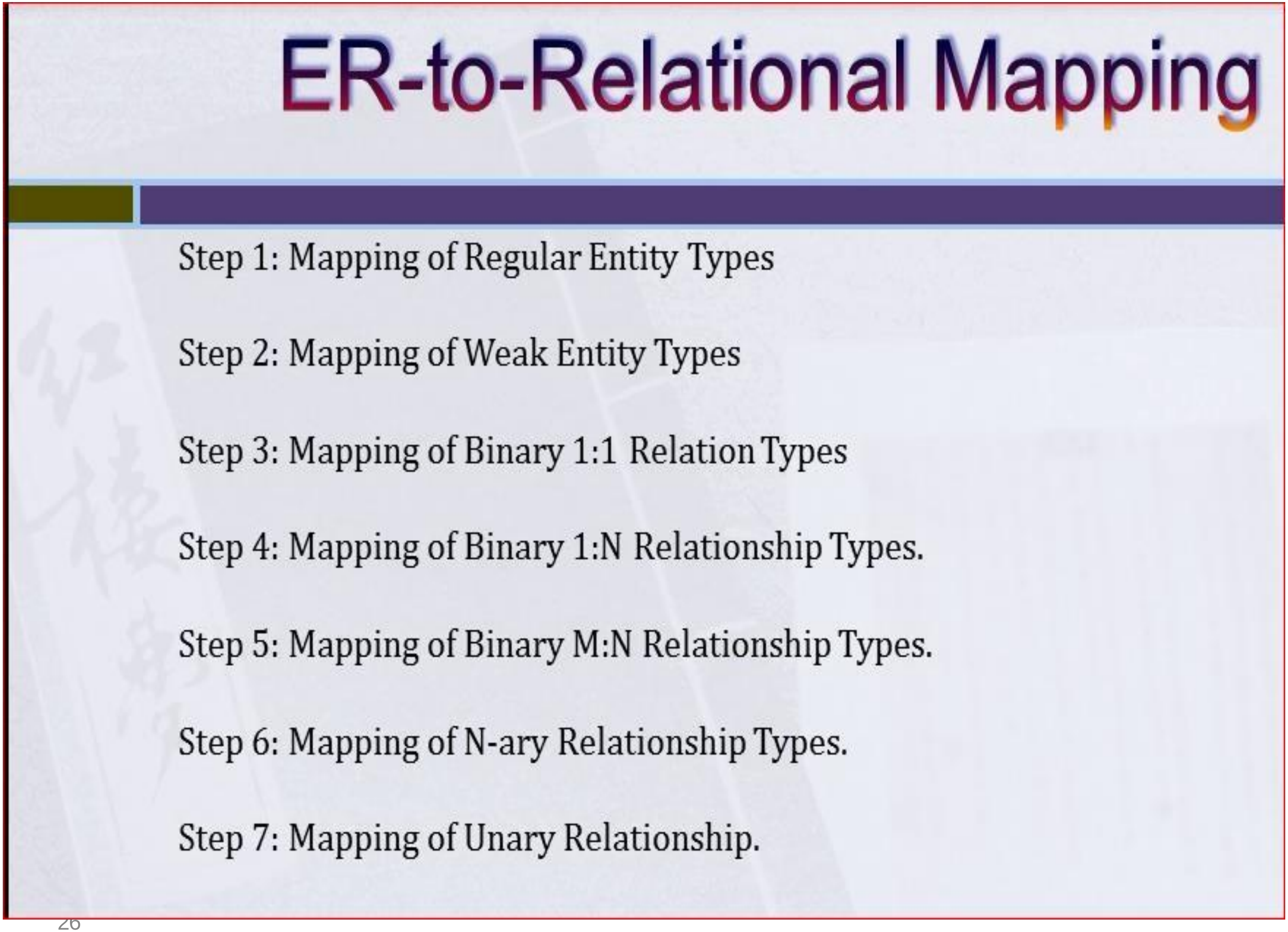
- NOT NULL Constraint: Ensures that a column cannot have NULL value.
- DEFAULT Constraint: Provides a default value for a column when none is specified.
- UNIQUE Constraint: Ensures that all values in a column are different.
- PRIMARY Key: Uniquely identified each rows/records in a database table.
- FOREIGN Key: Uniquely identified a row/record in any other database table.
- CHECK Constraint: The CHECK constraint ensures that all values in a column satisfy certain conditions.
- INDEX: Use to create and retrieve data from the database very quickly.

Part 3

Mapping
Convert ERD into database schema

Relational database definitions

- 1- table =instance=relation=entity
- 2- attribute =columns=field
- 3-row=tuple =record
- 4- cell=value=domain
- 5-Database=set of tables
- 6-Database integrity



ER-to-Relational Mapping

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relation Types

Step 4: Mapping of Binary 1:N Relationship Types.

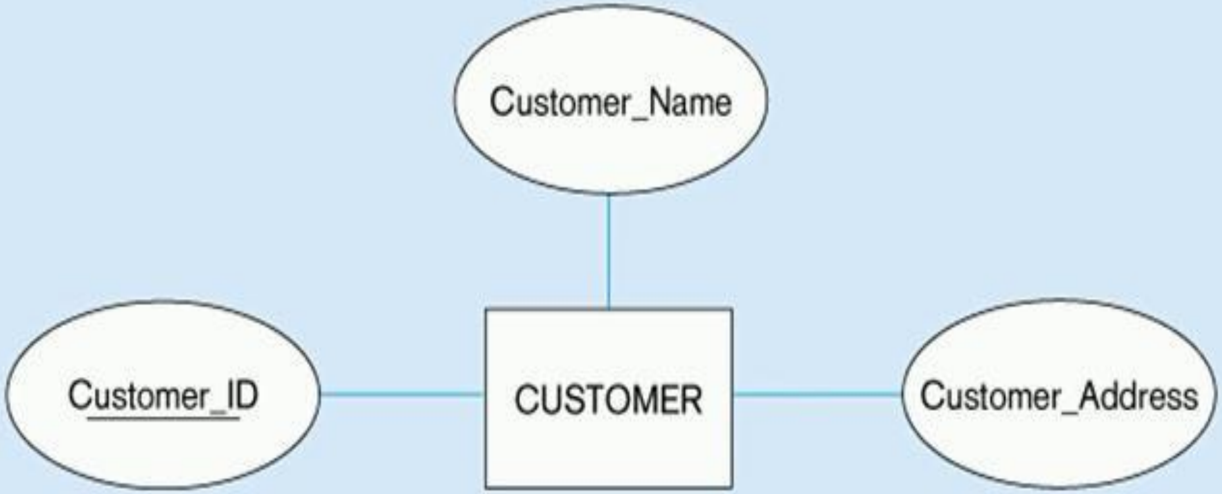
Step 5: Mapping of Binary M:N Relationship Types.

Step 6: Mapping of N-ary Relationship Types.

Step 7: Mapping of Unary Relationship.

Mapping Regular entity

(a) CUSTOMER entity type with simple attributes



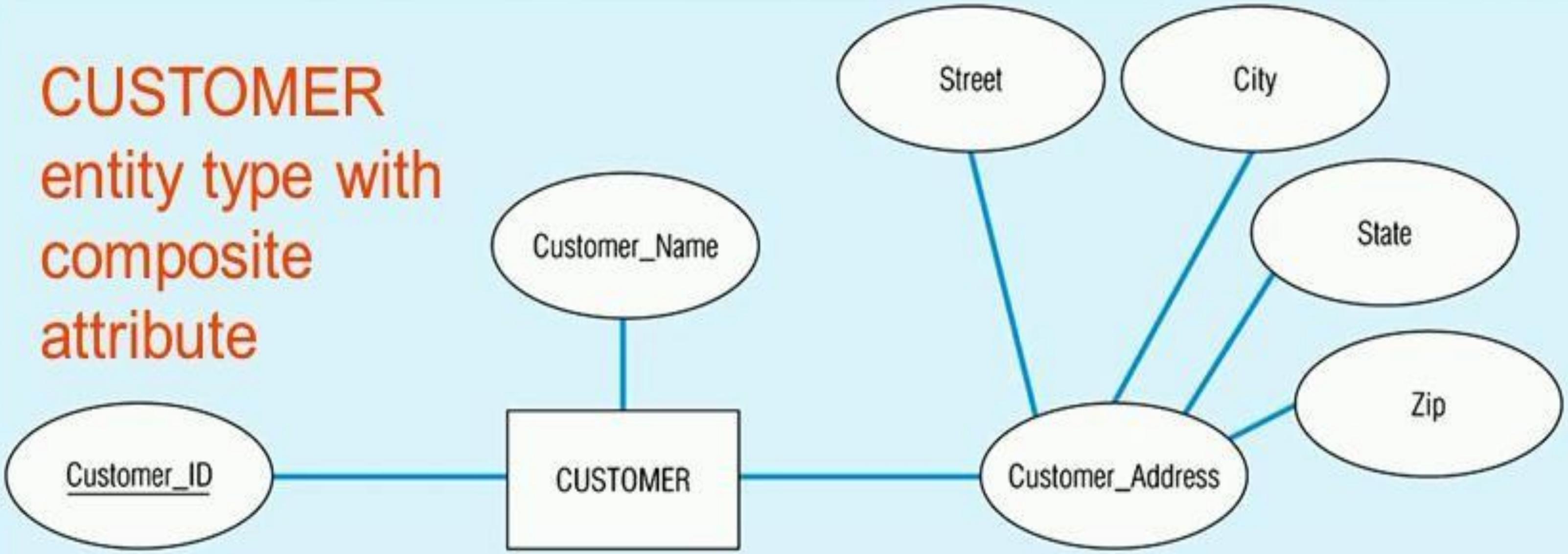
(b) CUSTOMER relation

CUSTOMER

Customer_ID	Customer_Name	Customer_Address

Mapping Composite attribute

CUSTOMER entity type with composite attribute

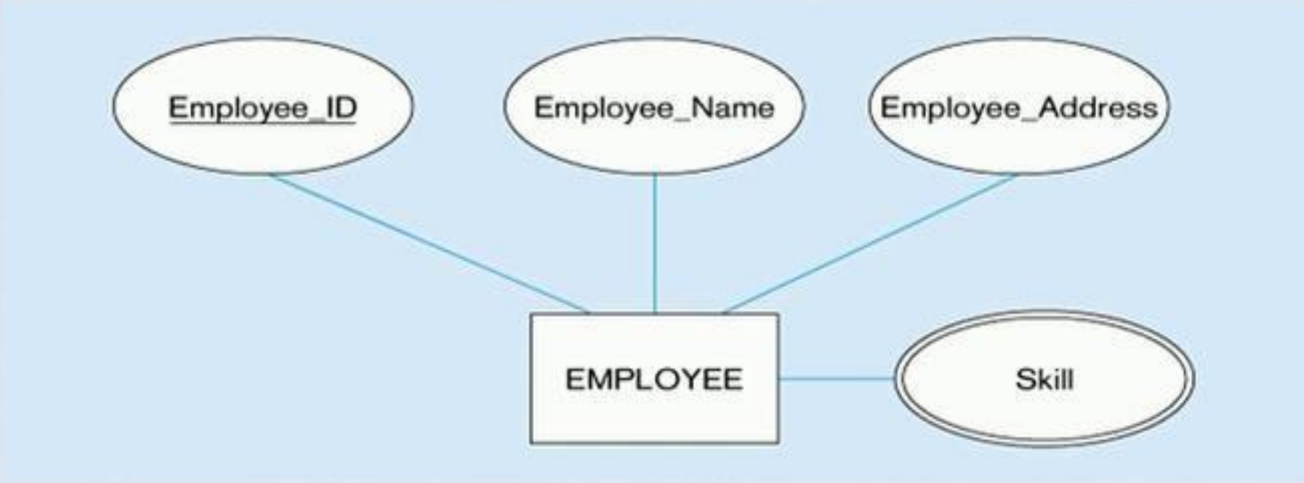


CUSTOMER relation with address detail

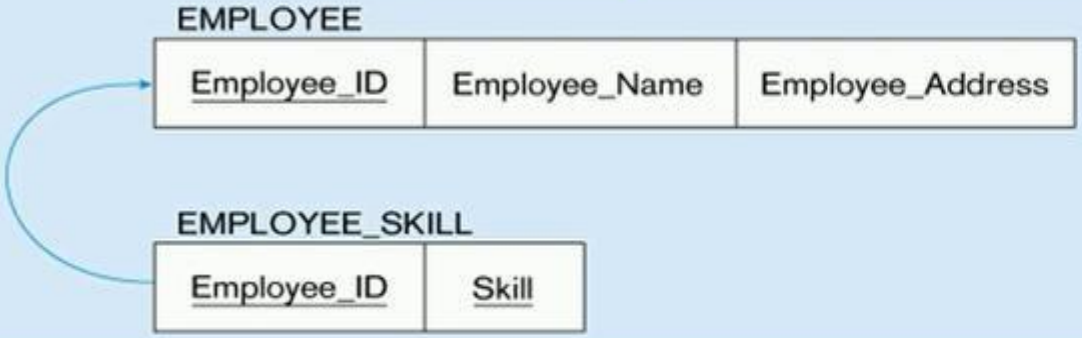
CUSTOMER

Customer_ID	Customer_Name	Street	City	State	Zip

Mapping Multivalued Attribute



Multivalued attribute becomes a separate relation with foreign key

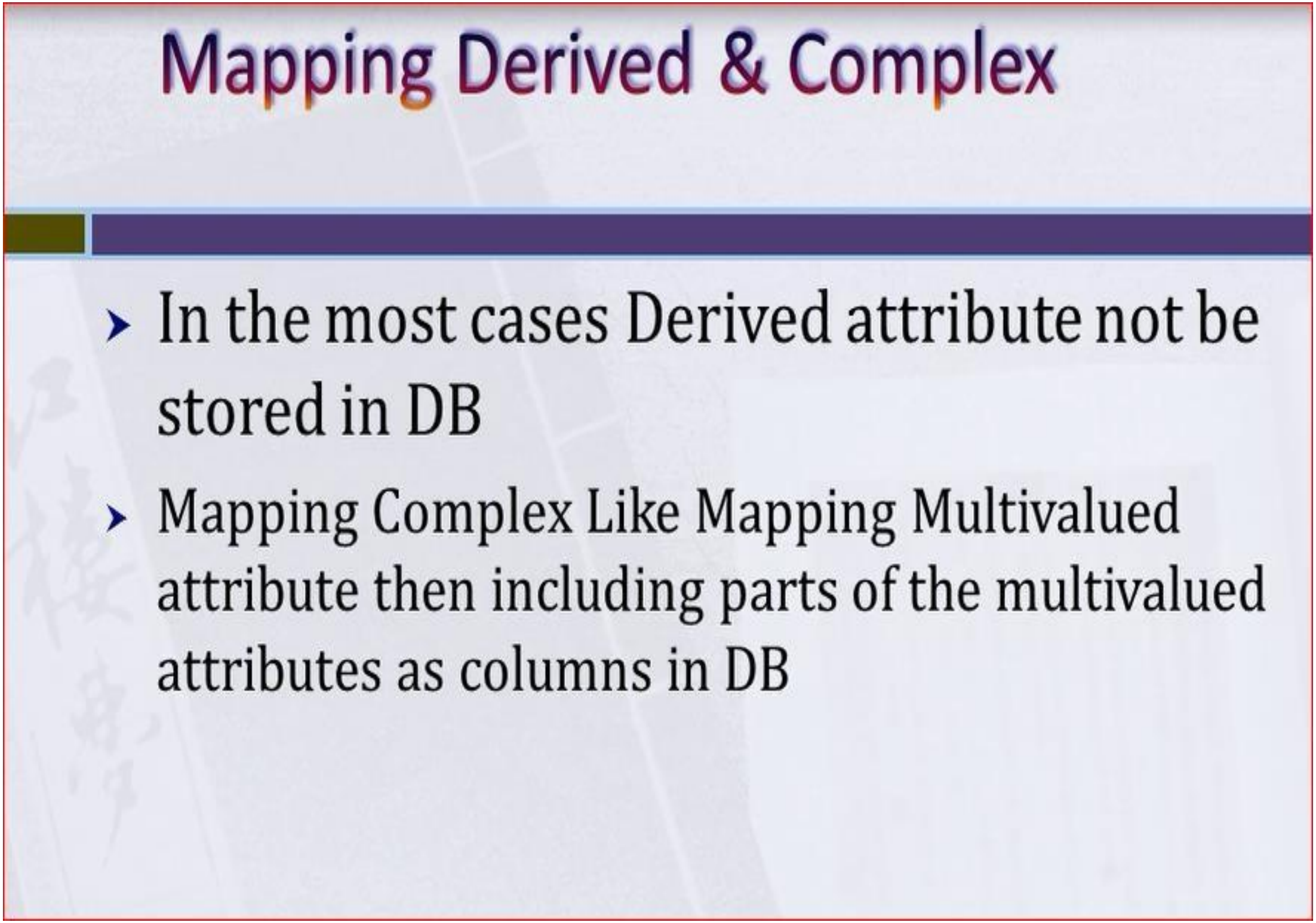


EMPLOYEE

Employee_ID	Employee_Name	Employee_Address

EMPLOYEE_SKILL

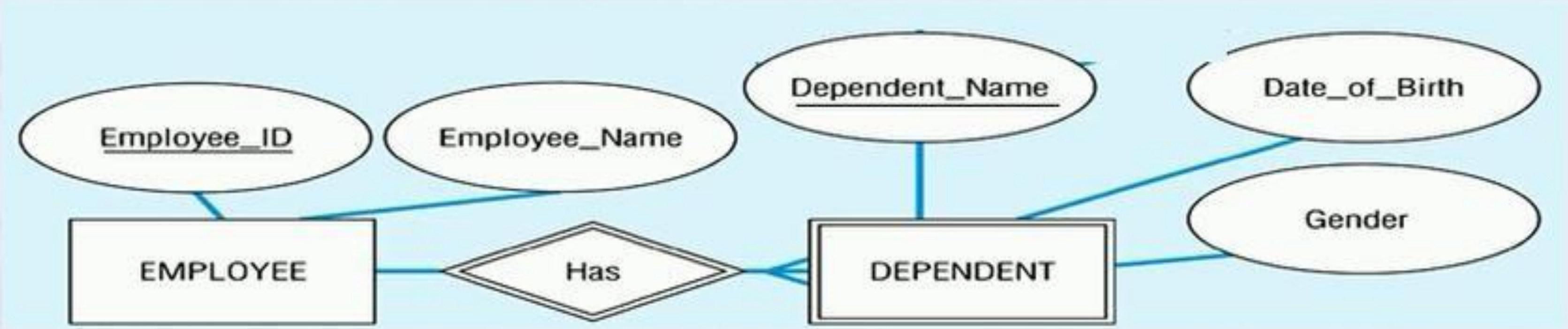
Employee_ID	Skill

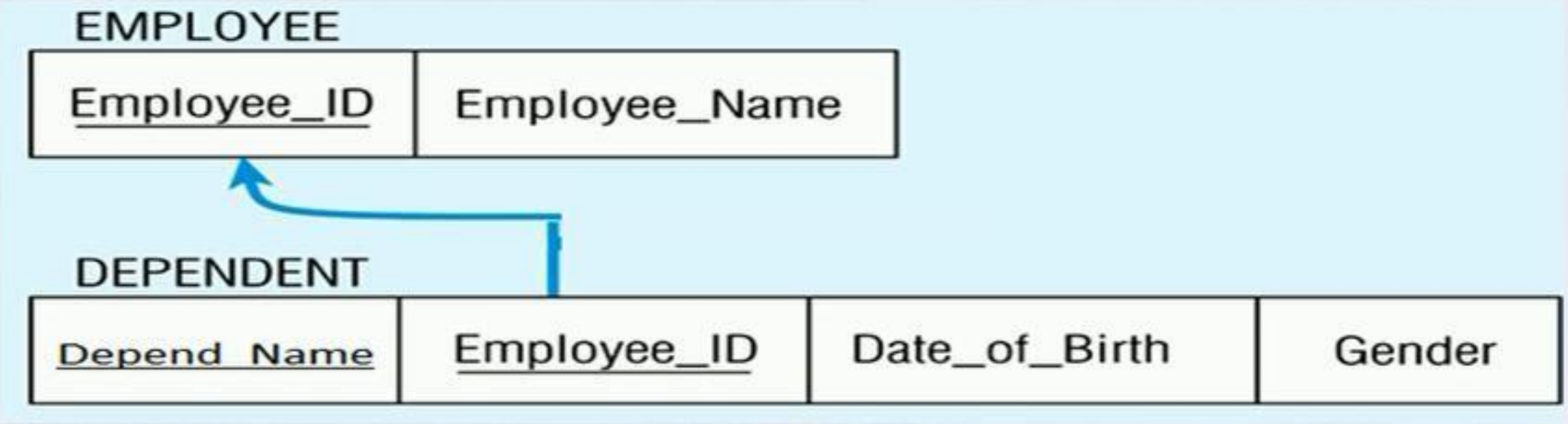


Mapping Derived & Complex

- In the most cases Derived attribute not be stored in DB
- Mapping Complex Like Mapping Multivalued attribute then including parts of the multivalued attributes as columns in DB

Mapping Weak entity





EMPLOYEE

Employee_ID	Employee_Name

DEPENDENT

Depend_Name	Employee_ID	Date_of_Birth	Gender

Composite primary key

Any question??