Department of Software Engineering
Mehran University of Engineering and Technology, Jamshoro

Course: SW222 – Database Management & Administration			
Instructor	Ms Shafiya Qadeer	**Practical/Lab No.**	02
Date	20-21/01/2021	**CLOs**	2
Signature		**Assessment Score**	2 Marks

Topic	**To become familiar with Table creation and population of table**
Objectives	- TO BECOME FAMILIAR DESIGNING A DATABASE

Lab Discussion: Theoretical concepts and Procedural steps

DATABASE DESIGN:

- It is the process of creating data models of database.
- data model: it determines the logical structure of a database.
- Designing is always done before the implementation of the database.
- It focuses on data requirements/needs.
- It is also called database modeling

PHASES FOR DESIGNING THE DATABASE

- There are three levels of data modeling/designing.
    1. Conceptual Data Model
    2. Logical Data Model
    3. Physical Data Model

CONCEPTUAL MODEL

- It describes the data in less detail and does not contain any implementation details(not physically implemented).
- It is a graphical representation of the data requirements.
- Transforms data requirements to conceptual model.
- Conceptual data model includes ERDs.

ENTITY RELATIONSHIP DIAGRAM(ERD)

- An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, places, concepts or events within that system.
- An entity relationship diagram (ERD) shows the relationships of entities stored in a database.
- ER Diagrams are most often used to design relational databases.

COMPONENTS OF ER DIAGRAM

1. Entity
2. Attributes
3. Relationship

1. ENTITY

- An entity can be a real-world object that can be easily identifiable.
- A definable thing such as a person, object, place or event.
- Examples: a customer, student, car or product etc.
- Entities are represented by rectangles.

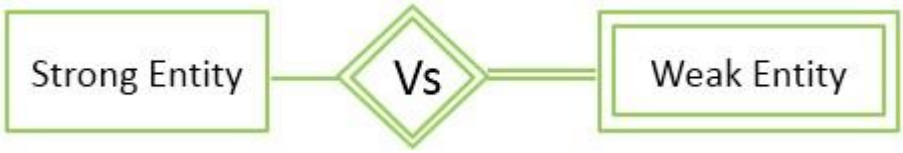
Student Entity

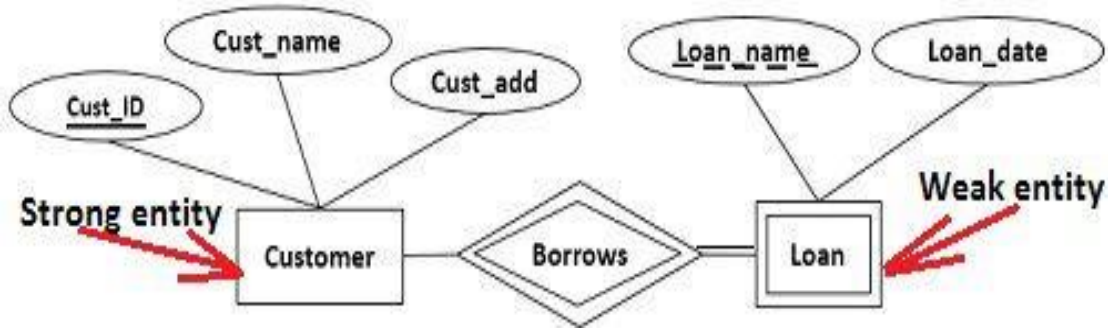
Teacher Entity

TYPES OF ENTITIES

1. **<u>WEAK ENTITY:</u>**
   - Weak entity is an entity that depends on another entity. Weak entities are also called dependent entities, it is represented by a double rectangle.
   - The weak entity has a partial discriminator key.

2. **STRONG ENTITY**
   - If an entity type has a key attribute specified then it is a strong entity type.
   - The Strong Entity is the one whose existence does not depend on the existence of any other entity. It is denoted by a single rectangle.



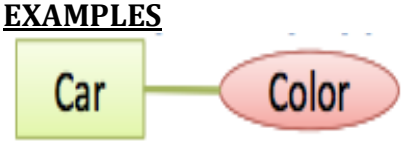


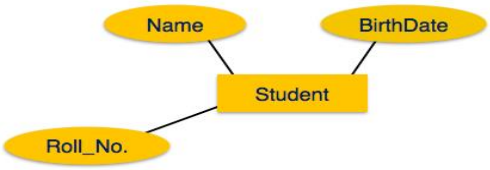
Observing the ER-diagram above, for each loan, there should be at least one borrower otherwise that loan would not be listed in Loan entity set. But even if a customer does not borrow any loan it would be listed in Customer entity set. So we can conclude that a customer entity does not depend on a loan entity.

2. ATTRIBUTES

- Attributes are the properties or characteristics of entities.
- Attributes are represented by means of ellipses.
- Every ellipses represents one attribute and is directly connected to its entity (rectangle).
- E.g. color of a car, balance of an account, price of a product, name of a student etc.

EXAMPLES





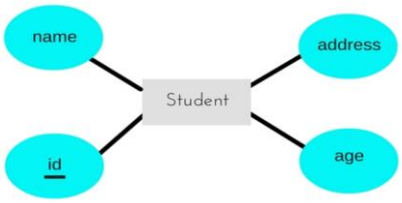
Types of Attributes

1. Simple/ Single Valued attribute: Simple attributes are atomic/single values, which cannot be divided further. For example, student's age etc.

2. Multi-valued attribute: Multi-value attributes may contain more than one values. For example, a person can have more than one phone number, email_address, etc. Multivalued attributes are depicted by double ellipse.

3. Derived attribute: Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average_salary in a department should not be saved directly in the database, instead it can be derived.
Derived attributes are depicted by dashed ellipse.

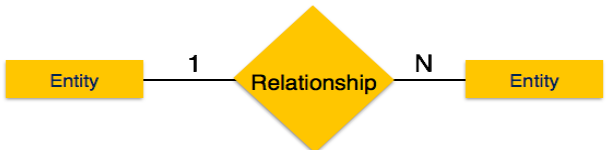
4. Key Attribute:
Key attribute represents the main characterstic/attribute of an Entity. It is used to represent Primary key. Ellipse with underlying lines represent Key Attribute



RELATIONSHIP

- There can be relationships between entities, which also can have attributes.
- Relationships are represented by diamond-shaped box.
- Name of the relationship is written inside the diamond-box.
- All the entities (rectangles) participating in a relationship, are connected to it by a line.
- For example, an employee works_at a department, a student enrolls in a course.

REPRESENTATION:



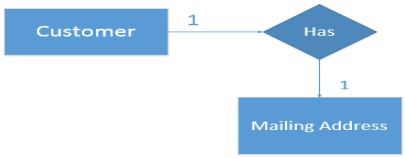
E.g. Person owns Car

TYPES OF RELATIONSHIP(3 TYPES)

1. One-to-one: (1:1)

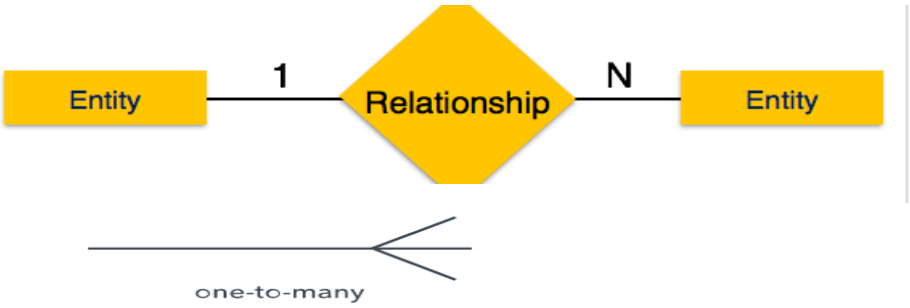
One element of entity A may only be linked to one element of entity B. For example, A university has 1 VC. Or customer will have one mailing address

EXAMPLE:



2. One-to-many (1:N)

- An element of A may be linked to many elements of B.
- one entity is associated with many number of same entity.
- For example, a single customer might place an order for multiple products.



3. Many to many (M:N) :

multiple elements of A linked to multiple elements of B.

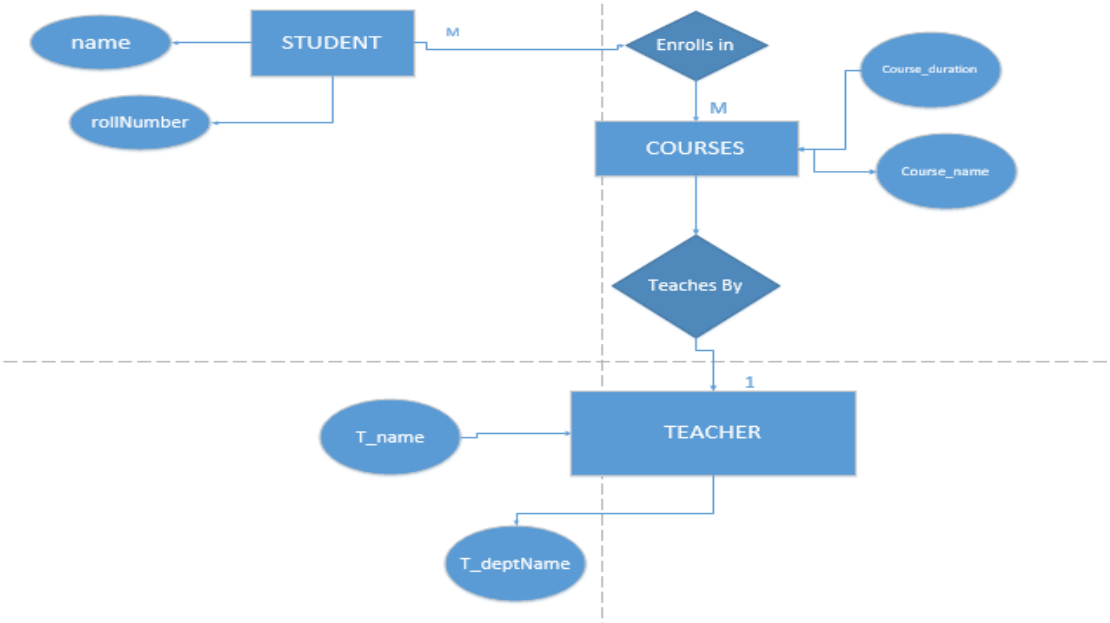
eg: Multiple students enrolls in multiple courses.

ERD CASE STUDY (Enrollment of Student in Courses)

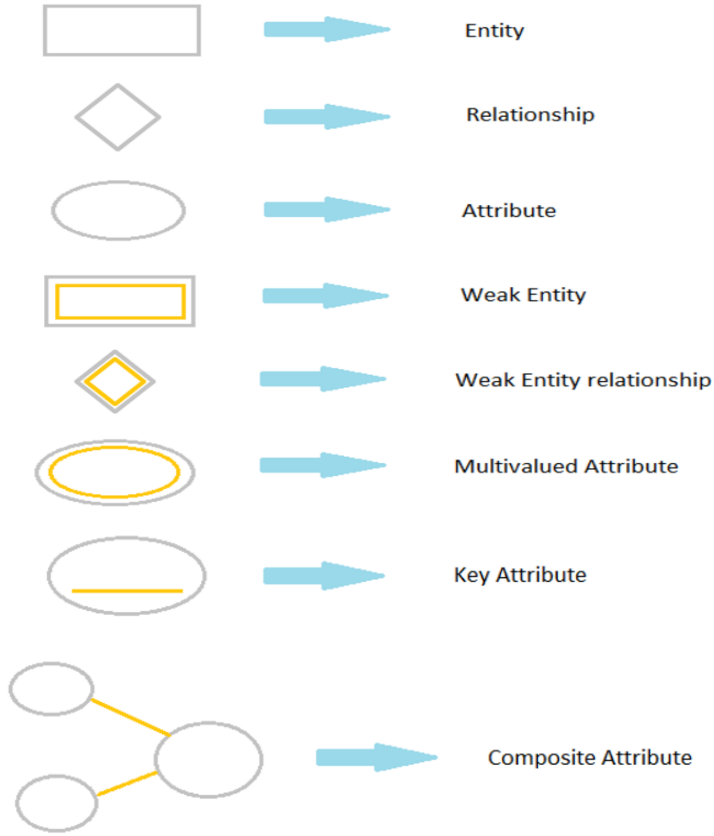
Consider the Following requirement List:

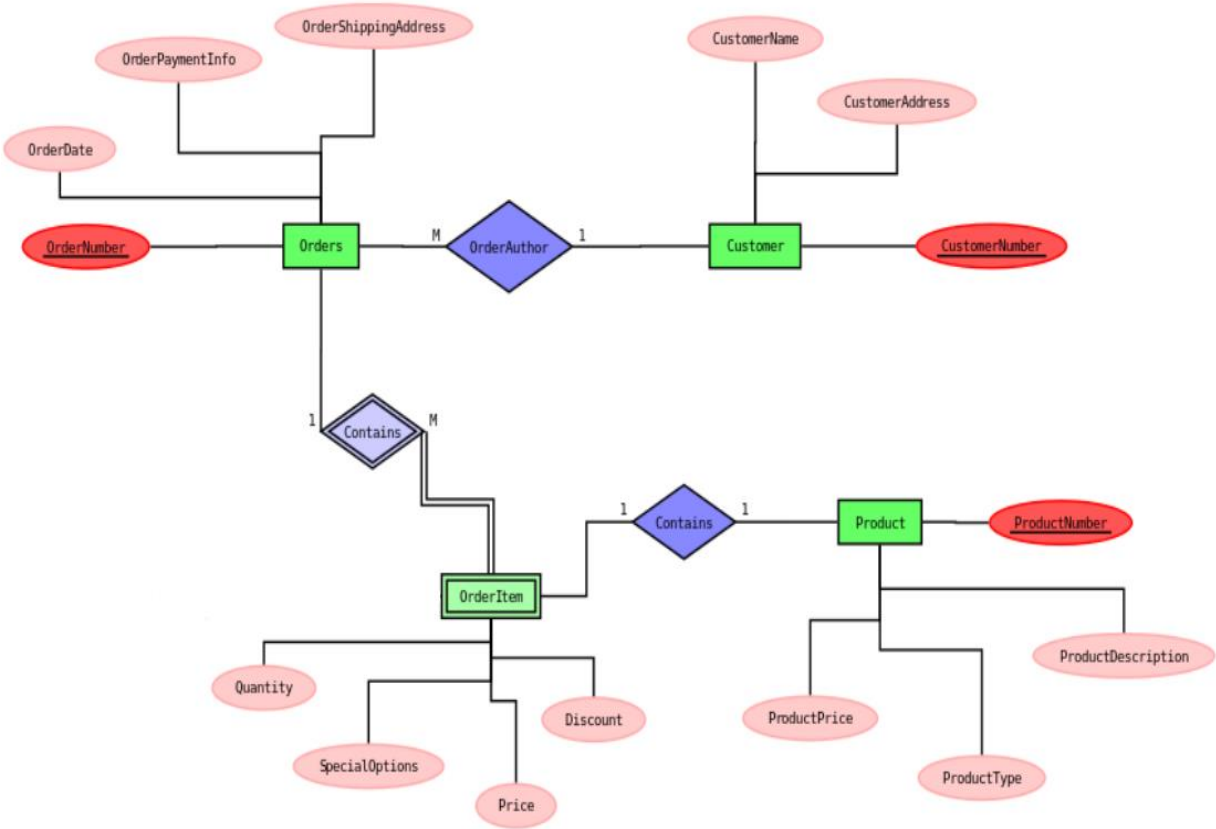
1. A student can be enrolled in multiple courses.
2. Every student has a unique roll number and name.
3. Course has a unique courseID , name of the course and course duration.
4. A teacher can teach multiple courses.
5. Teacher has a unique ID , name and department name.

ERD EXAMPLE



Symbols and Notations



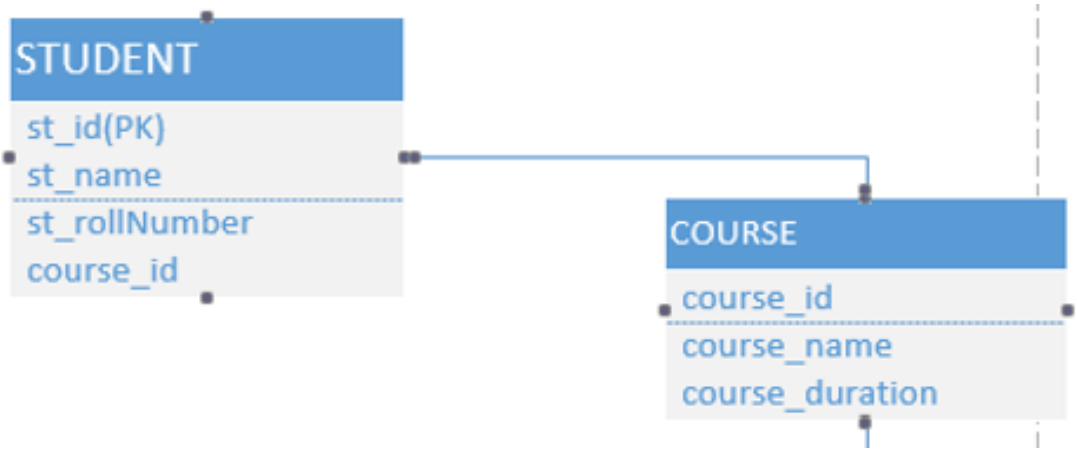


LOGICAL DATA MODEL

A logical data model describes the data in as much detail as possible, without regard to how they will be physically implemented in the database.

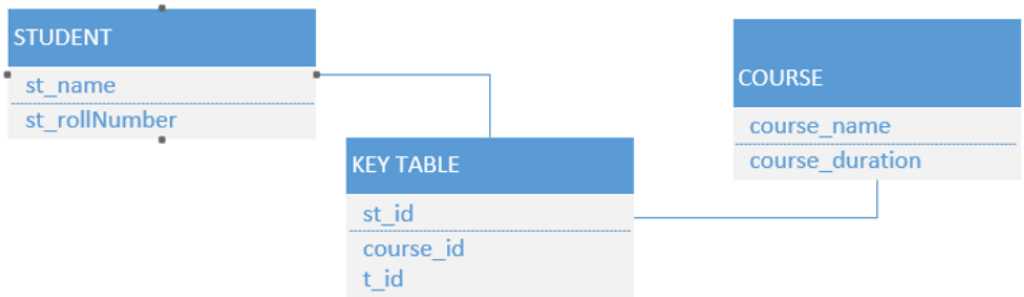
- The steps for designing the logical data model are as follows:
- Specify primary keys for all entities.
- Find the relationships between different entities.
- Find all attributes for each entity.
- Resolve many-to-many relationships.
- Normalization.

EXAMPLE



Resolve many-to-many relationships

- To resolve a many-to-many relationship, A new entity comes between the original entities, and this new entity is referred to as an intersection entity.
- Intersection Entity contains the ID's (the integer values) and is referred as “Fact Table”.



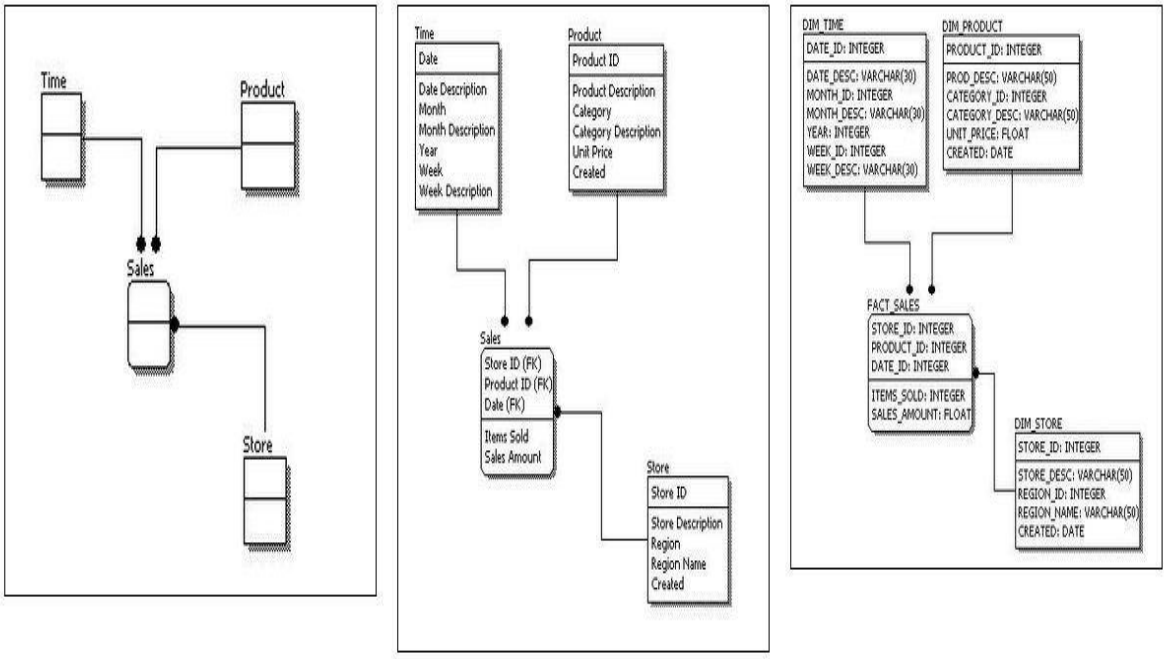
Physical Data Model

- Physical data model represents how the model will be built in the database.
- A physical database model shows all table structures, including column name, column data type, primary key and foreign key, and relationships between tables.

Steps for physical data model design

1. Convert entities into tables.
2. Convert relationships into foreign keys.
3. Convert attributes into columns.

Conceptual Model Design Logical Model Design Physical Model Design



Lab Tasks

The Flight Database stores details about an airline's flights and seat bookings. Consider the Following requirement List:

- The airline has one or more airplanes.
- An airplane has a model number, a unique registration number, and the capacity.
- An airplane flight has a unique flight number, a departure airport, a destination airport, a departure date and time and an arrival date and time.
- Each flight is carried out by a single airplane.
- A passenger has given names, a surname and a unique email address.
- A passenger can book one or more seats on a flight.

- **For the above mentioned case study**
- **1. Identify the Entities.**
- **2. Specify the attributes for each of the entity.**
- **3. Specify the relationship among entities.**
- **Draw the Conceptual Model, , Logical Model and Physical Model.**