EXPERIMENT NO: 4

Problem Statement:

Design any database with at least 3 entities and relationships between them. Draw suitable ER/EER diagram for the system.

Objective:

To design a relational database with at least 3 entities and relationships between them, and to create a suitable ER/EER diagram for the system.

Prerequisites:

1. Basic understanding of database concepts.
2. Access to a database design tool or software.

Equipment and Software:

1. Database design tool (e.g., draw.io, Lucidchart, ERDPlus, etc.).

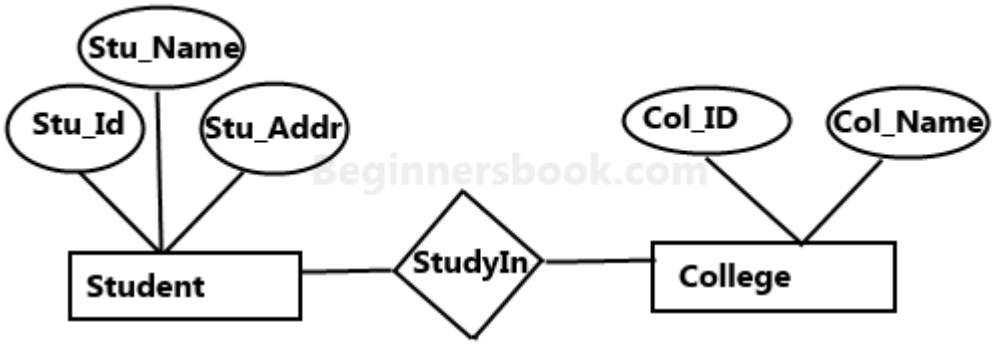
Theory:

An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

What is an Entity Relationship Diagram (ER Diagram)?

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Lets have a look at a simple ER diagram to understand this concept.

A simple ER Diagram:



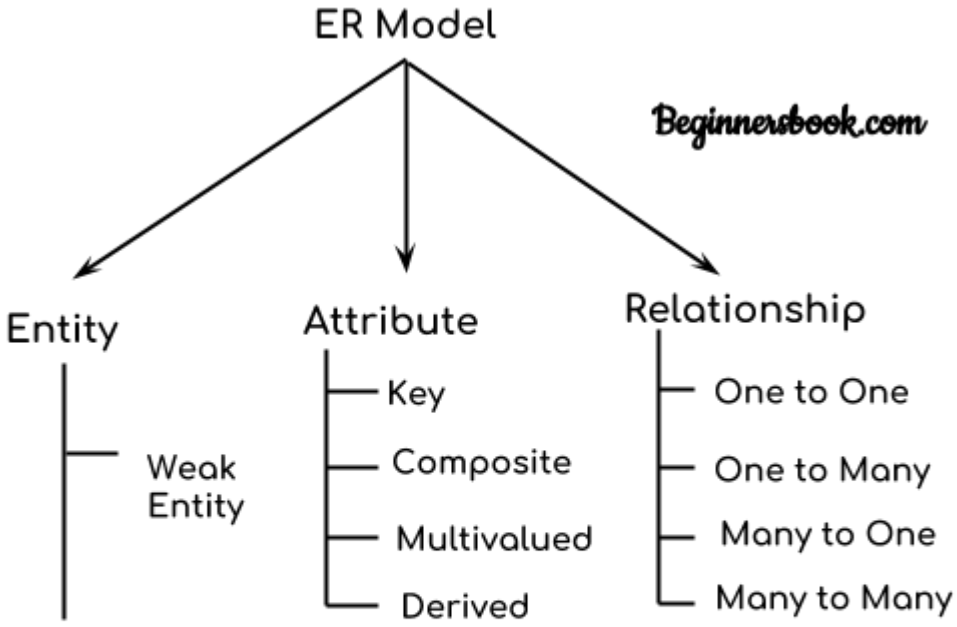
Sample E-R Diagram

In the following diagram we have two entities Student and College and their relationship. The relationship between Student and College is many to one as a college can have many students however a student cannot study in multiple colleges at the same time. Student entity has attributes such as Stu_Id, Stu_Name & Stu_Addr and College entity has attributes such as Col_ID & Col_Name.

Here are the geometric shapes and their meaning in an E-R Diagram. We will discuss these terms in detail in the next section(Components of a ER Diagram) of this guide so don't worry too much about these terms now, just go through them once.

Rectangle: Represents Entity sets.
Ellipses: Attributes
Diamonds: Relationship Set
Lines: They link attributes to Entity Sets and Entity sets to Relationship Set
Double Ellipses: Multivalued Attributes
Dashed Ellipses: Derived Attributes
Double Rectangles: Weak Entity Sets
Double Lines: Total participation of an entity in a relationship set

Components of a ER Diagram

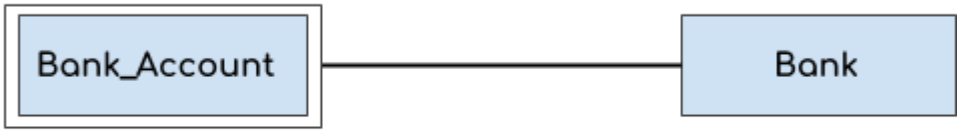


Components of ER Diagram

As shown in the above diagram, an ER diagram has three main components:
1. Entity
2. Attribute
3. Relationship

1. Entity

An entity is an object or component of data. An entity is represented as rectangle in an ER diagram.
For example: In the following ER diagram we have two entities Student and College and these two entities have many to one relationship as many students study in a single college. We will read more about relationships later, for now focus on entities.



Weak Entity:

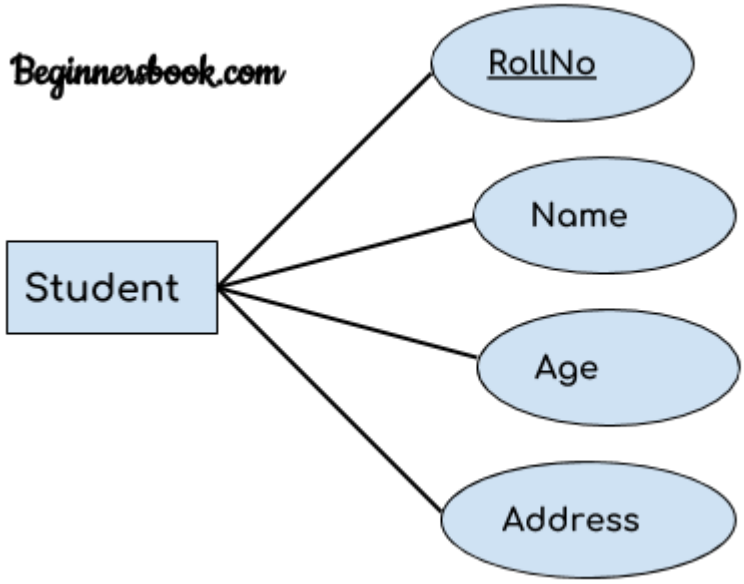
An entity that cannot be uniquely identified by its own attributes and relies on the relationship with other entity is called weak entity. The weak entity is represented by a double rectangle. For example – a bank account cannot be uniquely identified without knowing the bank to which the account belongs, so bank account is a weak entity.

2. Attribute

An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram. There are four types of attributes:

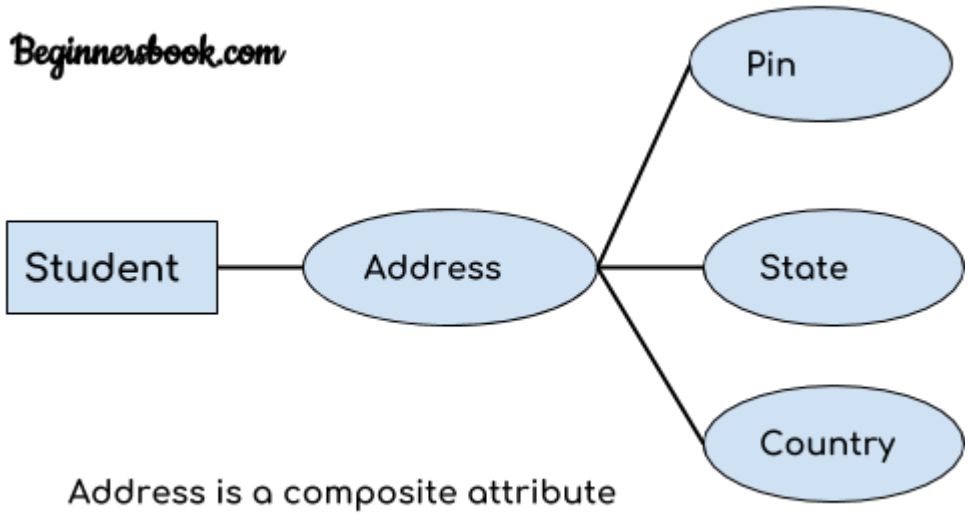
1. Key attribute
2. Composite attribute
3. Multivalued attribute
4. Derived attribute

1. Key attribute:



A key attribute can uniquely identify an entity from an entity set. For example, student roll number can uniquely identify a student from a set of students. Key attribute is represented by oval same as other attributes however the **text of key attribute is underlined**.

2. Composite attribute:



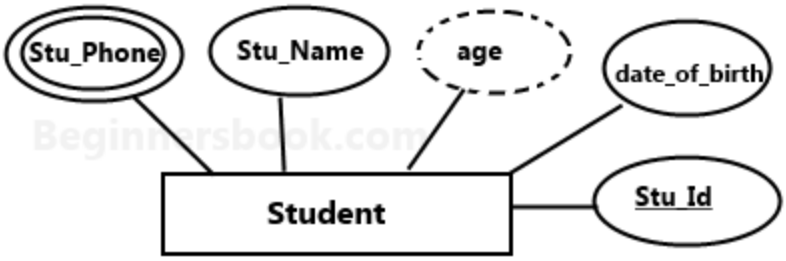
An attribute that is a combination of other attributes is known as composite attribute. For example, In student entity, the student address is a composite attribute as an address is composed of other attributes such as pin code, state, country.

3. Multivalued attribute:

An attribute that can hold multiple values is known as multivalued attribute. It is represented with **double ovals** in an ER Diagram. For example – A person can have more than one phone numbers so the phone number attribute is multivalued.

4. Derived attribute:

A derived attribute is one whose value is dynamic and derived from another attribute. It is represented by **dashed oval** in an ER Diagram. For example – Person age is a derived attribute as it changes over time and can be derived from another attribute (Date of birth).

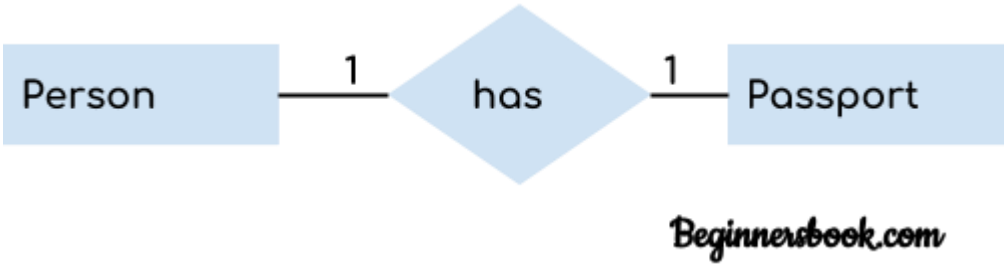


E-R diagram with multivalued and derived attributes:

3. Relationship

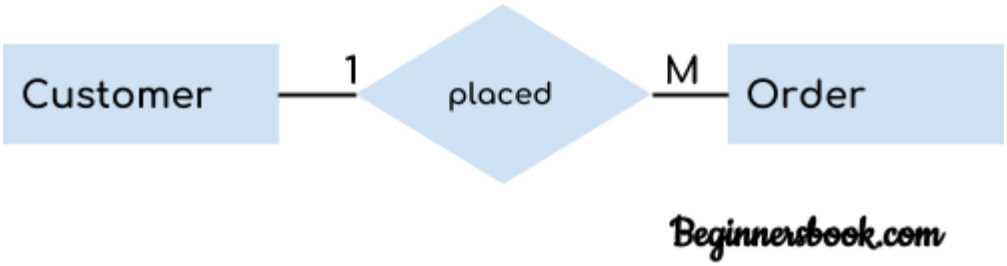
A relationship is represented by diamond shape in ER diagram, it shows the relationship among entities. There are four types of relationships:
1. One to One
2. One to Many
3. Many to One
4. Many to Many

1. One to One Relationship



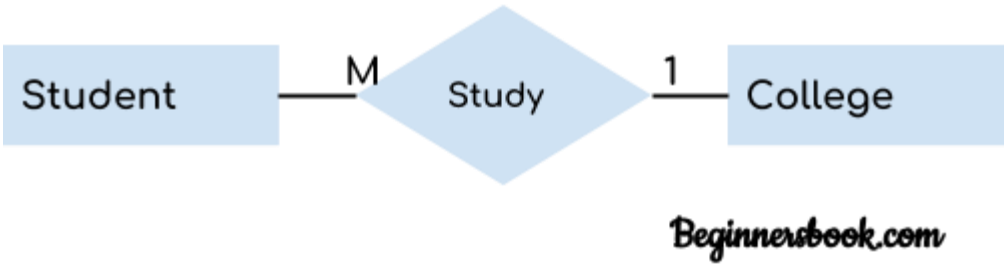
When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship. For example, a person has only one passport and a passport is given to one person.

2. One to Many Relationship



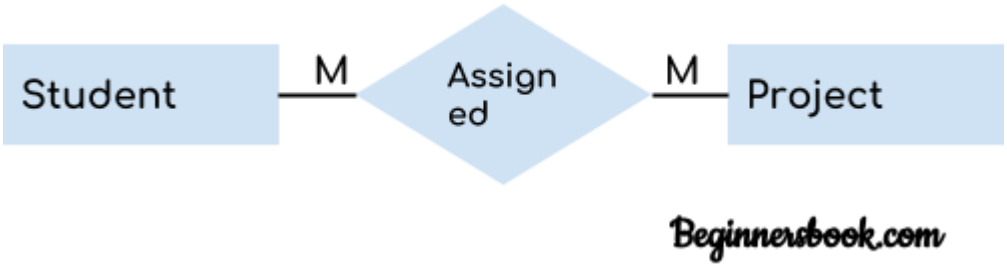
When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationship. For example – a customer can place many orders but a order cannot be placed by many customers.

3. Many to One Relationship



When more than one instances of an entity is associated with a single instance of another entity then it is called many to one relationship. For example – many students can study in a single college but a student cannot study in many colleges at the same time.

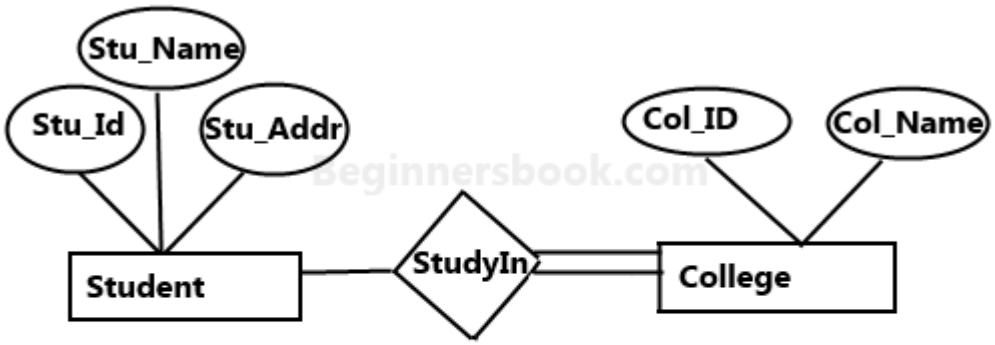
4. Many to Many Relationship



When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship. For example, a can be assigned to many projects and a project can be assigned to many students.

Total Participation of an Entity set

Total participation of an entity set represents that each entity in entity set must have at least one relationship in a relationship set. It is also called **mandatory participation**. **For example:** In the following diagram each college must have at-least one associated Student. Total participation is represented using a **double line** between the entity set and relationship set.



E-R Digram with total participation of College entity set in StudyIn relationship Set - This indicates that each college must have atleast one associated Student.

Let's design a simple database with three entities and relationships between them. The entities are: **Customer**, **Order**, and **Product**. The relationships are such that a **Customer** can place many **Orders**, and each **Order** can contain multiple **Products**. Here's the corresponding Entity-Relationship (ER) diagram:

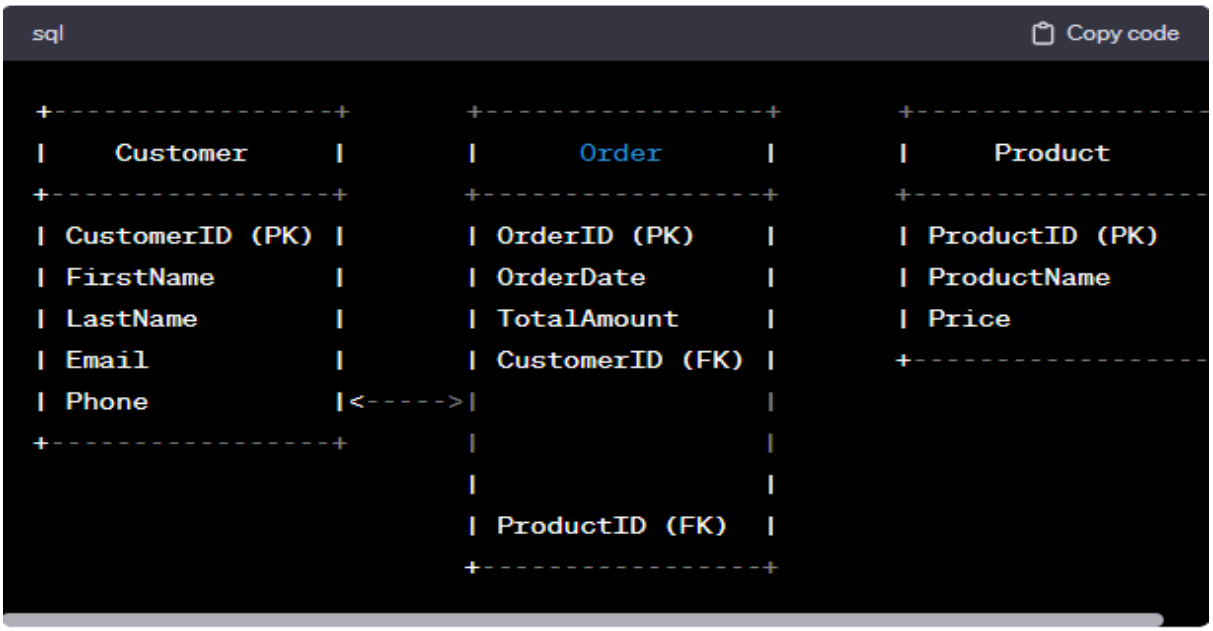
Entities:

1. **Customer:**
    - Attributes: CustomerID (Primary Key), FirstName, LastName, Email, Phone
2. **Order:**
    - Attributes: OrderID (Primary Key), OrderDate, TotalAmount
    - Foreign Key: CustomerID (references Customer)
3. **Product:**
    - Attributes: ProductID (Primary Key), ProductName, Price

Relationships:

1. **Customer-Order Relationship:**
    - One-to-Many relationship from Customer to Order (One customer can place many orders, but each order belongs to one customer).
    - Foreign Key: CustomerID in Order entity.
2. **Order-Product Relationship:**
    - One-to-Many relationship from Order to Product (One order can have multiple products, but each product belongs to one order).
    - Foreign Key: OrderID in Product entity.

ER Diagram:



In this ER diagram:

- Each box represents an entity.
- The ovals represent attributes of each entity.
- The lines connecting entities represent relationships.
- The diamond shapes represent the cardinality of the relationships.

Cardinality Notation:

- **1** on one side of the line represents "One."
- **M** (or **N**) on the other side represents "Many."

Example:

- A **Customer** can place **0** or **M** (many) **Orders**.
- An **Order** must be placed by **1** (one) **Customer**.
- An **Order** can contain **1** or **M** (many) **Products**.
- A **Product** must belong to **1** (one) **Order**.

Conclusion:

This design allows you to represent the relationships between customers, orders, and products in a structured and normalized way. Feel free to adapt this design based on your specific requirements and constraints.