Name: **Megh Dave**

Roll no: **D061**

Branch: **CSEDS**

SAP ID: **60009230176**

# AY 2023-24 (Semester -III)

**Experiments No: 1 (ER Model)**

**Aim:** To study Entity-Relationship Model

## Problem Definition:

Define a study system scenario. Specify all the entity sets, their respective attributes and relationship sets among the entities for the same. Show generalization, specialization, aggregation. Specify the cardinality and participation in the relationship sets.

## Theory:

1. **What is an Entity?**

In databases, an \*\*entity\*\* is a real-world object or concept that can be uniquely identified and is represented by a table. Each entity has attributes that describe its characteristics, and each instance of an entity is stored as a row in the table.

An example of an entity in a database is \*\*Customer\*\*.

- Entity: Customer

- Attributes:

- CustomerID (Primary Key)

- FirstName

- LastName

- Email

- PhoneNumber

Each row in the "Customer" table would represent a different customer, and the `CustomerID` uniquely identifies each customer.

1. **What is an attribute? Explain different types of attributes.**

An attribute in a database is a property or characteristic of an entity that provides more information about that entity. In simpler terms, attributes describe the details of an entity. For example, if "Student" is an entity, attributes could include Student ID, Name, Age, and

Enrollment Date. Different types of attributes include:

1. **Simple Attribute:** Indivisible and atomic, representing a single value.

2. **Composite Attribute:** Can be broken down into smaller sub-attributes.

3. **Single-Valued Attribute:** Holds only one value for each entity.

4. **Multi-Valued Attribute:** Can hold multiple values for each entity.

5. **Derived Attribute:** Computed from other attributes.

6. **Stored Attribute:** Physically stored in the database.

7. **Key Attribute:** Uniquely identifies an entity.

8. **Non-Key Attribute:** Provides additional information but doesn't uniquely identify an entity.

1. **What is relationship set?**

A relationship set in a database is a fundamental concept that represents the collection of all relationships of a particular type between two or more entities within an Entity-Relationship (ER) model. It defines how entities are associated with one another and captures the interaction or connection between them.

1. **What are different cardinality constraints? Specify various notations for the same.**

Cardinality constraints in a database specify the number of instances of one entity that can or must be associated with an instance of another entity through a relationship. These constraints define the rules for the relationships between entities and are critical in designing the structure of a database.

Types of Cardinality Constraints:

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

Description: Each entity in the first entity set is associated with at most one entity in the second set, and vice versa.

Example: A person has one passport, and each passport is assigned to one person.

Notation: Represented as "1:1" in ER diagrams.

1. **One-to-Many (1 : M) or Many-to-One (M:1):**

Description: In a one-to-many relationship, one entity from the first set can be associated with multiple entities from the second set, but each entity in the second set is associated with at most one entity from the first set.

Example: A professor teaches many courses, but each course is taught by only one professor.

Notation: Represented as "1 " (one-to-many) or "M:1" (many-to-one) in ER diagrams.

1. **Many-to-Many (M : N ):**

Description: Entities in the first set can be associated with multiple entities in the second set,

and vice versa.

Example: Students can enroll in many courses, and each course can have many students. o Notation: Represented as "M " in ER diagrams

1. **Explain participation in the relationship sets.**

Participation in relationship sets refers to the extent to which entities from an entity set are involved in a relationship within a database. It determines whether all or only some entities of an entity set must participate in a relationship set. Total Participation: When an entity set has total participation in a relationship set, it indicates that every entity within the set is required to be related to at least one entity in the related set. For example, in a database of a university, every

"Course" might need to be taught by a "Professor," meaning the "Course" entity set has total participation in the "Teaches" relationship set. Partial Participation: On the other hand, partial participation means that some entities in the entity set may not be involved in the relationship at all. For instance, in a company, not every "Employee" might be assigned to a "Project," so the participation of "Employee" in the "Works\_On" relationship set is partial.

1. **What is generalization and specialization?**

Generalization is the process of extracting common characteristics from multiple entities and combining them into a generalized, higher-level entity, often referred to as a "superclass" or "parent" entity. It is essentially a bottom-up approach where multiple specialized entities are generalized into a broader category. Suppose there are entities "Car" and "Truck," each with attributes like "Engine Type" and "Wheels." These can be generalized into a higher-level entity called "Vehicle," which includes common attributes like "Engine Type" and "Wheels," while specific attributes (like "Cargo Capacity" for "Truck") remain in their respective specialized entities. Specialization is the reverse process of generalization, where a generalized entity is divided into more specialized, lower-level entities, known as "subclasses" or "child" entities. It is a top-down approach where a broad entity is specialized into multiple entities that have distinct characteristics. Consider a generalized entity "Employee." This entity can be specialized into "Manager," "Engineer," and "Salesperson," each with additional attributes that are specific to their role. For instance, "Manager" might have an attribute "Team Size," while "Salesperson" might have "Sales Target

1. **Explain the concept of aggregation.**

Aggregation is a concept in database design, particularly in the Entity Relationship (ER) model, used to simplify complex ER diagrams by abstracting relationships between entities into higher-level entities. It allows for a more manageable and organized representation of relationships, especially when dealing with complex data structures.

## ER model:

Draw ER model for the system.