Experiment No.1
Design an EntityRelationship (ER) / Extended Entity-Relationship (EER) Model.
Date of Performance:
Date of Submission:

Aim :- Identify the case study and detailed statement of the problem. Design an EntityRelationship (ER) / Extended Entity-Relationship (EER) Model.

Objective :- To identify and explore a real world problem, and to design an Entity Relationship (ER) / Extended Entity-Relationship (EER) Model.

Theory:

1. **Entity:**

- An entity is a real-world object or concept that exists independently and has distinguishable attributes.
- In a database context, an entity represents a table, and each row in that table represents a unique instance of that entity.
- For example, in a university database, entities could include Student, Course, Professor, Department, etc.
- Each entity has a set of attributes that describe its properties.

2. **Attributes:**

- Attributes are the properties or characteristics that describe an entity.
- They represent the data we want to store about each instance of an entity.
- For example, attributes of a Student entity might include StudentID, Name, Age, GPA, etc.
- Attributes can be categorized as simple (atomic) attributes, which cannot be divided further, or composite attributes, which are made up of smaller sub-parts.

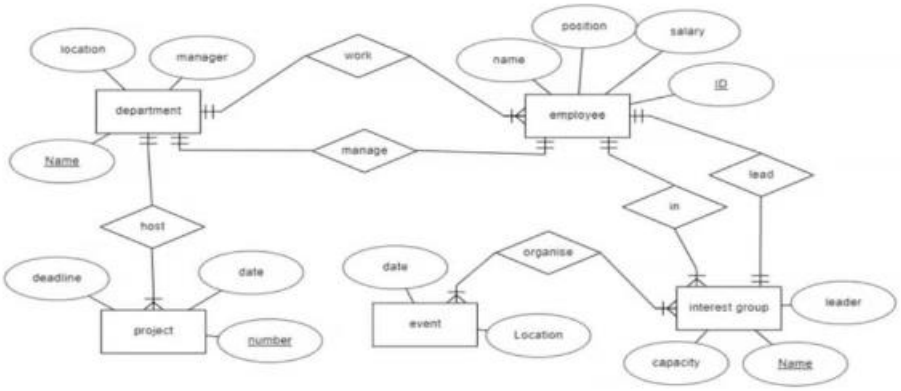
3. **Relationships:**

- Relationships describe how entities are related to each other or how they interact.
- They represent the associations between entities.
- Relationships are depicted as lines connecting related entities in the ER diagram.
- Each relationship has a degree, indicating the number of entities involved. It could be unary (involving one entity), binary (involving two entities), or ternary (involving three entities).
- Relationships also have cardinality, which defines the number of instances of one entity that can be associated with the number of instances of another entity through the relationship.

4. **Cardinality:**

- Cardinality specifies the number of instances of one entity that are related to the number of instances of another entity through a relationship.
- It defines the maximum and minimum number of occurrences of one entity that can be associated with the occurrences of another entity.
- Common cardinality constraints include:
    I. One-to-One (1:1): Each instance of one entity is associated with exactly one instance of another entity, and vice versa.
    II. One-to-Many (1:N): Each instance of one entity is associated with zero or more instances of another entity, but each instance of the second entity is associated with exactly one instance of the first entity.
    III. Many-to-One (N:1): The reverse of One-to-Many; many instances of one entity are associated with one instance of another entity.
    IV. Many-to-Many (N:N): Many instances of one entity can be associated with many instances of another entity.

Implementation:



Conclusion:

1. Define Entity, Attributes(also types) and Relationship between entities
2. Write ER/EER diagram notations

In conclusion, when designing a database schema, it's essential to understand the concepts of entities, attributes, and relationships:

1. Entity:
  - An entity is a real-world object or concept that exists independently and can be uniquely identified. In a database context, entities are represented as tables.
  - Example: In a university database, entities could include students, courses, professors, and departments.

2. Attributes:
  - Attributes are properties or characteristics that describe an entity. Each attribute has a name and a data type.
  - Examples of attribute types:
    - Integer: Represents whole numbers (e.g., student ID).
    - Varchar: Represents variable-length character strings (e.g., student name).
    - Date: Represents dates (e.g., enrollment date).
  - Attributes define the columns in a database table.

3. Relationships Between Entities:
  - Relationships define connections or associations between entities. They describe how entities are related to each other.
  - Common relationship types:
    - One-to-One (1:1): A single instance of an entity is associated with only one instance of another entity.
    - One-to-Many (1:N): A single instance of an entity can be associated with multiple instances of another entity.
    - Many-to-Many (N:M): Multiple instances of an entity can be associated with multiple instances of another entity.
  - Relationships are represented using foreign keys in database tables.

As for ER/EER diagram notations:

1. Entity:
  - Represented as rectangles with the entity name inside.
  - Example:
    ```
    Student
    ```

2. Attribute:
  - Represented as ovals connected to their respective entities.
  - Includes attribute name and type.
  - Example:
    ```
    Student
    -------
    student_id: int
    name: varchar
    enrollment_date: date
    ```

3. Primary Key:
  - Underlined attribute represents the primary key of an entity.

- Example:
  ```
  Student
  -------
  student_id (PK): int
  ```

4. Relationship:
   - Represented as lines connecting related entities.
   - Relationship types (1:1, 1:N, N:M) are indicated near the lines.
   - Example:
     ```
     Student ----(1:N)---- Enrollment
     ```

5. Cardinality:
   - Indicated near the entities to specify the cardinality of the relationship (e.g., 1, N).
   - Example:
     ```
     Student (1) ----(N)---- Enrollment
     ```

These notations help visualize the structure of the database schema, including entities, attributes, and their relationships, facilitating effective communication and understanding during the database design process.