Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

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:

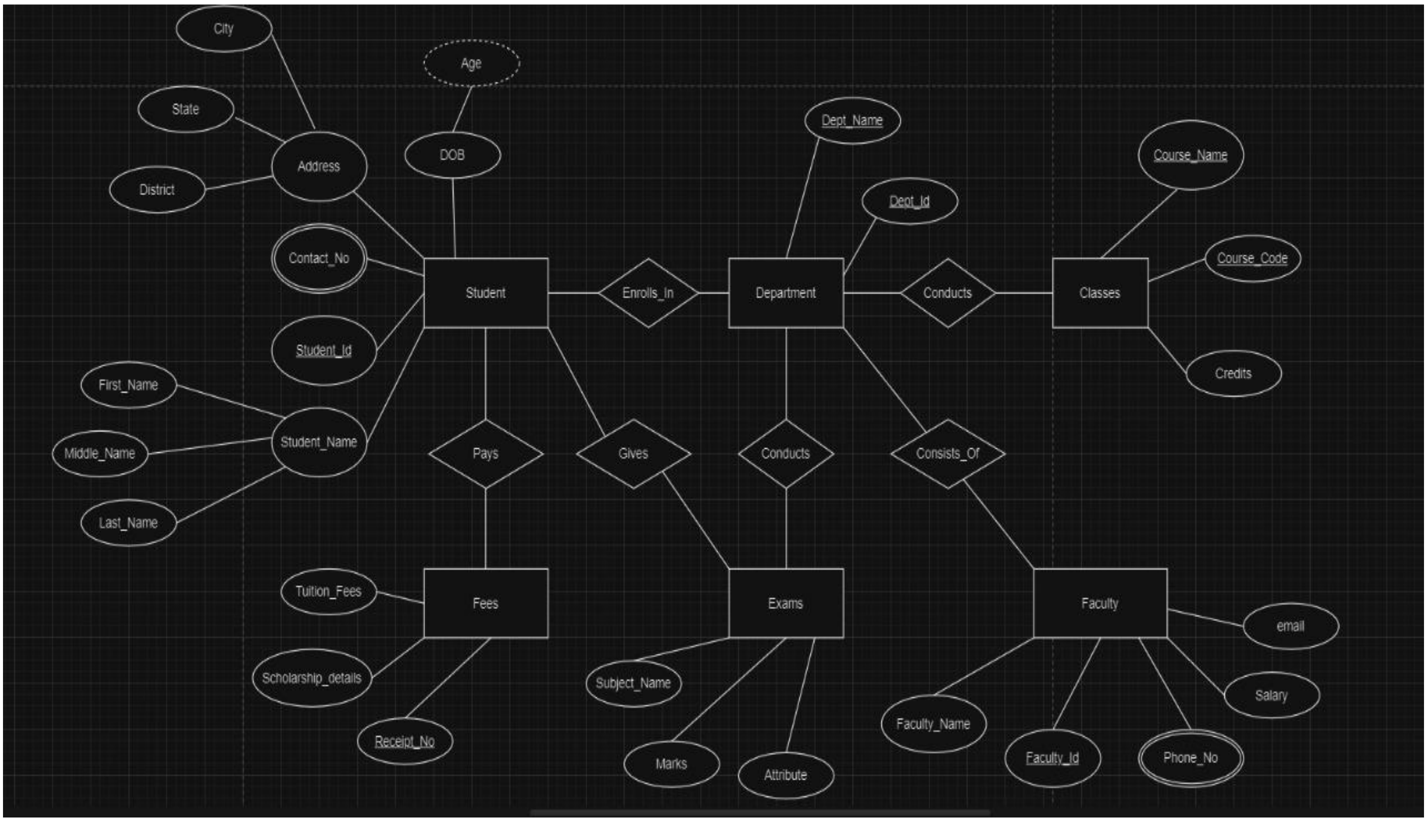
I. One-to-One (1:1): Each instance of one entity is associated with exactly one instance of another entity, and vice versa.

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**

In the context of databases, an entity is a real-world object or concept that is distinguishable from other objects. Attributes are the properties or characteristics that describe an entity, and they can be of different types such as strings, numbers, dates, etc. Relationships define how entities are related to each other. There are several types of relationships, including one-to-one, one-to-many, and many-to-many.

For example, consider a database for a library. Here, a book could be an entity with attributes like title, author, and publication date. A library member could be another entity with attributes such as name, address, and membership ID

2.**Write ER/EER diagram notations**

Here are the notations for an Entity-Relationship (ER) or Enhanced Entity-Relationship (EER) diagram:

Entities: Represented by rectangles. Example: Book, Author, Library Member.

Attributes: Represented by ovals connected to their respective entity. Example: Book entity with attributes Title, ISBN, Publication Date.

Primary Key: Underline the attribute that uniquely identifies each entity instance. Example: Underline BookID in the Book entity.

Relationships: Represented by diamond shapes connected by lines to the related entities. Example: Relationship between Book and Library Member for "borrowed by".

Cardinality: Indicates the number of instances of one entity that can be associated with each instance of another entity. Represented by symbols like 1, M, or N. Example: Book to Library Member relationship with 1 on the Book side and M on the Library Member side indicating "one-to-many".

Optionality: Indicates whether the existence of an entity depends on its relationship with another entity. Represented by a small oval or a line at the end of the relationship line. Example: Optional relationship between Book and Library Member.

Weak Entity: Represented by double rectangles. Example: Book Copy entity, which depends on the Book entity for its existence.

Derived Attribute: Represented by a dashed oval. Example: Total Copies attribute in the Book entity, which can be derived from the count of Book Copy entities.

These notations help in visualizing the structure of a database and the relationships between its entities.