



Vidyavardhini's College of Engineering and Technology, Vasai

Department of Artificial Intelligence & Data Science

Experiment No.1
Design an EntityRelationship (ER) / Extended Entity-Relationship (EER) Model.
Student Name:-Bhagyashri Kaleni Sutar
Roll No:- 75
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.



The diagram is an Entity-Relationship (E-R) model for a Railway Reservation System. It features the following entities and attributes:

- Login** (Entity): Attributes include Login_id, Login_role_id, Login_username, and Login_password.
- User** (Entity): Attributes include User_id, User_name, User_mobile, User_email, and User_address.
- Role** (Entity): Attributes include role_id, role_name, and role_desc.
- Permission** (Entity): Attributes include per_id, per_name, per_role, and per_module.
- Train** (Entity): Attributes include tn_id, tn_name, tn_class, and tn_desc.
- Payment** (Entity): Attributes include pay_id, pay_desc, pay_date, pay_amt, and pay_status.
- Customer** (Entity): Attributes include cus_id, cus_name, cus_mobile, cus_email, and cus_address.
- Booking** (Entity): Attributes include book_id, book_type, and book_desc.

Relationships and Cardinalities:

- Role** is associated with **User** and **Permission** via a central diamond relationship labeled **Role**.
- User** is associated with **Train** via a diamond relationship labeled **Manage**.
- Train** is associated with **Payment** via a diamond relationship labeled **Role**.
- Payment** is associated with **Booking** via a diamond relationship labeled **Role**.
- Customer** is associated with **Payment** via a diamond relationship labeled **Role**.

Cardinalities are indicated by numbers at the relationship points: 1:1 for User-Train, 1:1 for Train-Payment, 1:1 for Payment-Booking, and 1:1 for Customer-Payment.



Conclusion

1. Define Entity, Attributes(also types) and Relationship between entities

Ans:- **1. Entity**

An **Entity** is any object, person, thing, or concept about which data is stored. It is something that can have attributes (properties) and can be distinctly identified in the system.

Example: In a university database, entities could include Student ,Professor
Course ,subject.

2. Attributes

Attributes are the properties or characteristics of an entity that provide more details about it.

a. Types of Attributes:

- i. **Simple Attribute:** Cannot be divided further. Example:age or Name.
- ii. **Composite Attribute:** An attribute that can be subdivided into smaller subparts. Example:full Name, which can be divided into first Name,Middle name,and Last name.
- iii. **Derived Attribute:** An attribute whose value can be derived from other attributes. Example:Age can be derived from date of Birth
- iv. **Multi-valued Attribute:** An attribute that can hold multiple values. Example:Phone Numbers for a person who may have multiple contact numbers.
- v. **Key Attribute:** An attribute that uniquely identifies an entity in an entity set. Example:Student Id in student in entity.



3. Relationship

A Relationship defines how entities are associated with one another. It describes the interactions or associations between entities.

- **Types of Relationships:**

- **One-to-One (1:1):** A single instance of one entity is associated with a single instance of another entity. Example: A **Student** has one **Student ID**.
- **One-to-Many (1:N):** A single instance of one entity is associated with multiple instances of another entity. Example: A **Professor** teaches multiple **Courses**, but each **Course** is taught by only one **Professor**.
- **Many-to-One (N:1):** Multiple instances of one entity are associated with a single instance of another entity. This is the reverse of the One-to-Many relationship.
- **Many-to-Many (M:N):** Multiple instances of one entity are associated with multiple instances of another entity. Example: **Students** enroll in multiple **Courses**, and each **Course** has multiple **Students**.

- **Cardinality of Relationships:** The number of instances of one entity that can or must be associated with the number of instances of another entity.

Example (University Database):

- **Entities:** Student, Course, Professors
- **Attributes:**
 - **Student:** Student ID (Key Attribute), Name, Age, Email



- Course: Course Code (Key Attribute), Course Name, Credits
- Professor: Professor ID (Key Attribute), Name, Department
- **Relationships:**
 - Enroll: A **Many-to-Many** relationship between Student and Course (Students can enroll in multiple courses, and courses can have multiple students).
 - Teaches: A **One-to-Many** relationship between Professor and Course (One professor can teach many courses, but each course is taught by one professor).

2. Write ER/EER diagram notations

Ans:- When designing ER or EER diagrams, specific symbols and notations are used to represent entities, attributes, relationships, and constraints. Below are the most commonly used notations for **ER** and **EER** diagrams:

1. Entity Notations:-

a. Entity:

- **Rectangle:** Represents an entity set (e.g., **Student**, **Course**).
- The name of the entity is written inside the rectangle

b. Weak Entity (in EER diagrams):

- **Double Rectangle:** Represents a weak entity set that depends on another entity for its identification.

c. Identifying Relationship (for weak entities):



- **Double Diamond:** Represents the relationship between a weak entity and its owner entity

2. Attribute Notations:

a. Simple Attribute:

- **Oval:** Represents an attribute of an entity. Simple attributes are represented by a single oval

b. Composite Attribute:

- **Double Oval:** Represents an attribute that can be divided into smaller parts

c. Derived Attribute:

- **Dashed Oval:** Represents an attribute whose value is derived from other attributes

d. Multi-valued Attribute:

- **Double Oval:** Represents an attribute that can have multiple values.

3. Relationship Notations:

a. Relationship:

- **Diamond:** Represents a relationship between two or more entities. The name of the relationship is written inside the diamond



b.Cardinality of Relationships:

- **1:1** (One-to-One): Draw a line between the entities with "1" on both sides of the relationship.
- **1:N** (One-to-Many): Draw a line with "1" on one side and "N" (or "M") on the other.
- **M:N** (Many-to-Many): Draw a line between the entities with "M" and "N" on each side of the relationship.

4. Key Attribute Notations:

a.Key Attribute:

- **Underlined Oval:** Key attributes are underlined to represent that they uniquely identify the entity.

5. Participation Constraints:

1.Total Participation (every instance of an entity must be involved in a relationship):

- **Represented by a double line connecting the entity to the relationship.**

2.Partial Participation (only some instances of the entity are involved in the relationship):

- **Represented by a single line connecting the entity to the relationship**



Vidyavardhini's College of Engineering and Technology, Vasai

Department of Artificial Intelligence & Data Science