Deccan Education Society's (DES)
Pune University, Pune
School of Engineering and Technology
Department of Computer Engineering and Technology
Program: B. Tech in Computer Science and Engineering

Academic Year: 2024-25	**Year:** Second Year	**Semester:** II
PRN No.: 102412079	**Name:Ratnajeet Patil**	
Subject: Database Management System		
Assignment No.: 1		
Date: 19/02/2025		

Lab Assignment: 01

Title: Design and Implementation of an ER/EER Diagram: Consider any innovative application like Hackathon Problem Statements and develop a conceptual design using ER/EER modelling tools such as ERD Plus, LucidChart, Erwin or any open-source tool. Try to show attributes types, constraints and relationships.
Finally, transform the ER/EER diagram into relational tables and normalize the relational data model to ensure efficiency and consistency.

Theory:

What is ER/EER?

ER (Entity-Relationship) Model:

- A conceptual data model that represents real-world entities, their attributes, and relationships.
- Used in database design to define data structure before implementation.
- Depicts entities as rectangles, relationships as diamonds, and attributes as ovals.

EER (Enhanced Entity-Relationship) Model:

- An extension of the ER model that includes additional concepts like generalization, specialization, and aggregation.
- Useful for representing more complex relationships in a database.

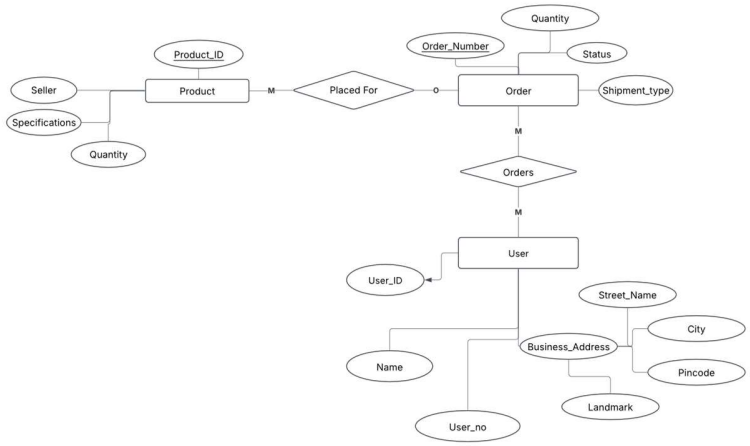
Basic Notations used in ER

- **Entity – Represented as a rectangle.**
- **Attributes – Represented as ovals (can be simple, composite, derived, or multi-valued).**
- **Primary Key – Underlined attribute inside an oval.**
- **Relationship – Represented as a diamond, connects entities.**
- **Cardinality – Indicates the number of instances involved (e.g., 1:1, 1:M, M:N).**
- **Weak Entity – Represented as a double rectangle, dependent on a strong entity.**

Steps followed to convert ER/EER into Tables

- **Convert Entities into Tables:**
  - **Each entity becomes a table.**
  - **Attributes become columns.**
  - **The primary key is assigned.**
- **Convert Relationships into Tables:**
  - **For 1:1 relationships, foreign key can be used in one of the tables.**
  - **For 1:M relationships, the foreign key is placed in the "many" side.**
  - **For M:N relationships, create a separate table with foreign keys from both entities.**
- **Convert Weak Entities:**
  - **Create a table including the foreign key of its owner entity as part of the primary key.**
- **Convert Multi-Valued Attributes:**
  - **Create a separate table linking it to the original entity.**

Designed ER/EER Diagram:



Converted Tables List:

1. Product Table

Column Name	Data Type	Constraints
Product_ID	INT	PRIMARY KEY
Seller	VARCHAR(255)	NOT NULL
Specifications	TEXT	NULLABLE
Quantity	INT	NOT NULL

2. Order Table

Column Name	Data Type	Constraints
Order_Number	INT	PRIMARY KEY
Quantity	INT	NOT NULL
Status	VARCHAR(50)	NOT NULL
Shipment_Type	VARCHAR(50)	NOT NULL

3. User Table

Column Name	Data Type	Constraints
User_ID	INT	PRIMARY KEY
Name	VARCHAR(255)	NOT NULL
User_no	VARCHAR(15)	UNIQUE, NOT NULL

What is Normalization?

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

The inventor of the relational model Edgar Codd proposed the theory of normalization of data with the introduction of the First Normal Form, and he continued to extend theory with Second and Third Normal Form. Later he joined Raymond F. Boyce to develop the theory of Boyce-Codd Normal Form.

Database Normal Forms:

- 1NF (First Normal Form)
- 2NF (Second Normal Form)
- 3NF (Third Normal Form)
- BCNF (Boyce-Codd Normal Form)
- 4NF (Fourth Normal Form)
- 5NF (Fifth Normal Form)

First Normal Form (1NF):

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single(atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

Second Normal Form (2NF):

For a table to be in the Second Normal Form,

1. It should be in the First Normal form.
2. And, it should not have Partial Dependency.

Third Normal Form (3NF):

A table is said to be in the Third Normal Form when,

1. It is in the Second Normal form.
2. And, it doesn't have Transitive Dependency.

FAQs:

1. What are different types of Attributes?

   Attributes are characteristics or properties of an entity. Different types of attributes in an ER model include:

   - **Simple (Atomic) Attribute**: Cannot be divided further (e.g., Name, Age).
   - **Composite Attribute**: Can be divided into smaller sub-parts (e.g., Name → First Name, Last Name).
   - **Derived Attribute**: Derived from other attributes (e.g., Age from Date of Birth).
   - **Multivalued Attribute**: Can have multiple values for a single entity (e.g., Phone Numbers).
   - **Stored Attribute**: Directly stored in the database (e.g., Date of Birth).
   - **Key Attribute**: Uniquely identifies an entity (e.g., Roll Number in a Student entity).

2. Differentiate specialization and generalization.

Feature	Specialization	Generalization
Definition	Process of creating sub-entities from a parent entity	Process of combining multiple entities into a single general entity
Approach	Top-down	Bottom-up
Example	Employee → {Engineer, Manager, Clerk}	Car, Truck → Vehicle
Use Case	Used to differentiate entities based on some criteria	Used to reduce redundancy by grouping similar entities

3. What is aggregation? How it is used in ER/EER?

Aggregation is an abstraction in which a relationship itself is treated as an entity. It is used when a relationship involves another relationship.

Usage in ER/EER Model:

- Used when a relationship needs to be linked with another entity.
- Helps represent higher-level abstractions in a database.

Example:
Consider a scenario where:

- **Entities**: Project, Supplier, Employee
- **Relationships**:
  - A supplier supplies a project (Supplies relationship).
  - An employee manages the Supplies relationship.

4. Explain total and partial participation with examples.

Feature	Total Participation	Partial Participation
Definition	Every entity in the set must participate in the relationship	Some entities in the set may not participate
Representation	Double-line in ER diagram	Single-line in ER diagram
Example	In a university, every student must have an ID card (Total Participation)	A professor may or may not be assigned to a research project (Partial Participation)

5. What do you mean by mapping cardinality?

Mapping Cardinality (also called Cardinality Ratio) defines the number of entities in one set that can be associated with entities in another set.
Types of Mapping Cardinality:

- One-to-One (1:1) → A person has one passport.
- One-to-Many (1:M) → A teacher teaches multiple students.
- Many-to-One (M:1) → Many students belong to one department.
- Many-to-Many (M:N) → Students enroll in multiple courses, and courses have multiple students.