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:

I. 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:

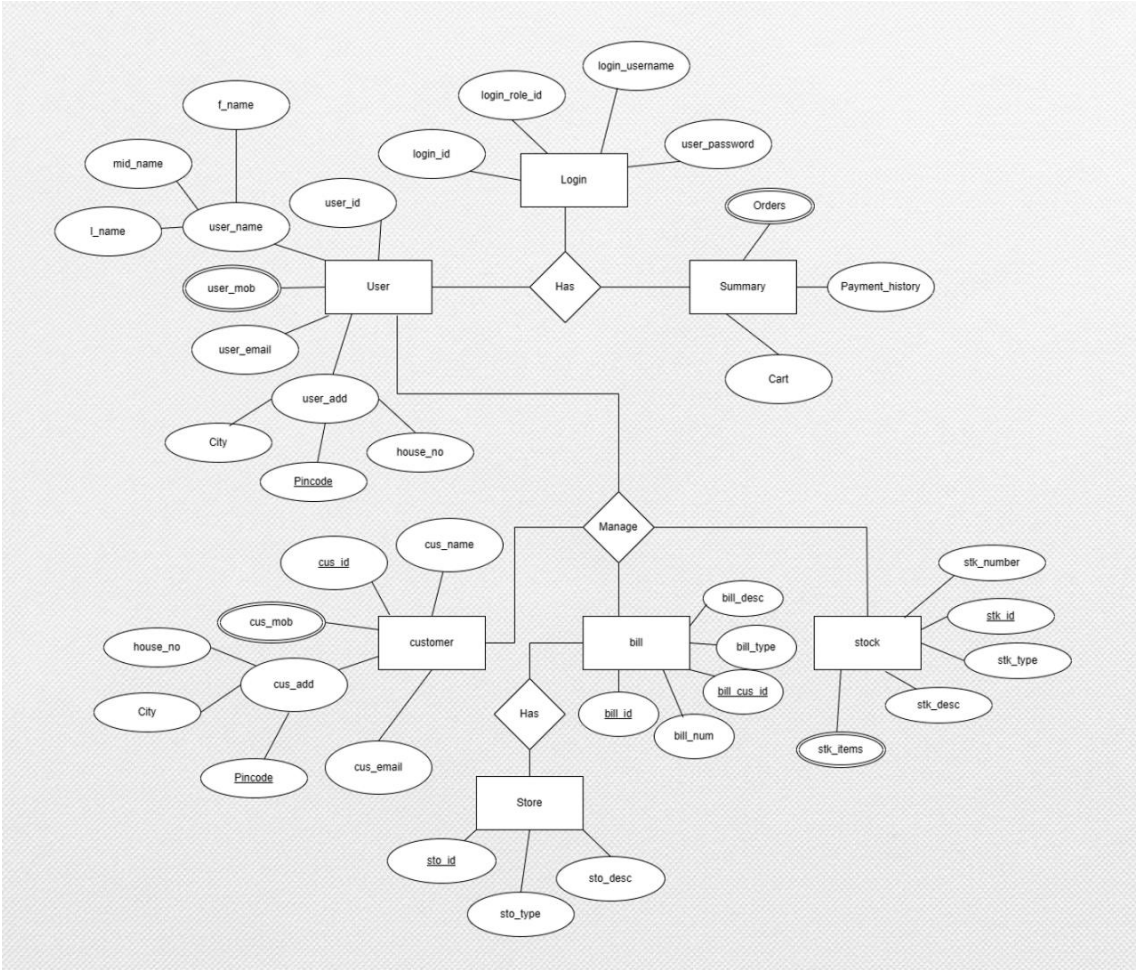
I. One-to-One (l : l): Each instance of one entity is associated with exactly one instance of another entity, and vice versa.

Il. One-to-Many (I: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.

111. 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. Entity:An entity is a real-world object or concept that has an independent existence and can be uniquely identified. In a database context, entities are typically represented as tables. For example, in a university database, entities could include Student, Course, and Professor.

2. Attributes: Attributes are the properties or characteristics of entities. They describe the entity's characteristics or features. Each entity has attributes that define its properties. For example, a Student entity might have attributes like Student ID, Name, and Date of Birth. Attributes also have types, such as integer, string, date, etc.

3. Relationship:A relationship represents an association between entities. It describes how entities are related to each other. Relationships can be one-to-one, one-to-many, or many-to-many. For example, in a university database, there could be a relationship between Student and

Course entities representing the fact that a student can enroll in multiple courses, and a course can have multiple students enrolled.

Now, regarding Entity-Relationship (ER) or Enhanced Entity-Relationship (EER) diagram notations:

Entities:Entities are represented as rectangles. The name of the entity is written inside the rectangle.

Attributes: Attributes are represented as ovals and are connected to their respective entity by a line. The attribute name is written inside the oval, and its data type can be specified if needed.

Primary Key: The primary key attribute(s) of an entity are underlined.

Relationships: Relationships are represented as diamond shapes connecting related entities. The name of the relationship is written inside the diamond. Lines connect the diamond to the related entities, indicating the cardinality and optionality of the relationship.

Cardinality: Cardinality indicates the number of instances of one entity that can be associated with the number of instances of another entity. Common cardinality notations include "1" for one, "M" for many, and "0..1" for optional.

Optionality: Optionality indicates whether the presence of an entity instance is required or optional in a relationship. It's often denoted by placing a circle or a line at the end of the line connecting the relationship to the entity. A circle indicates the entity is optional, while a line indicates it's mandatory.

Recursive Relationships: In case of recursive relationships (where an entity is related to itself), a self-referencing line is drawn between the entity and itself, with appropriate cardinality and optionality indicators.

This notation helps in visually representing the database structure, including entities, attributes, and their relationships.