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:

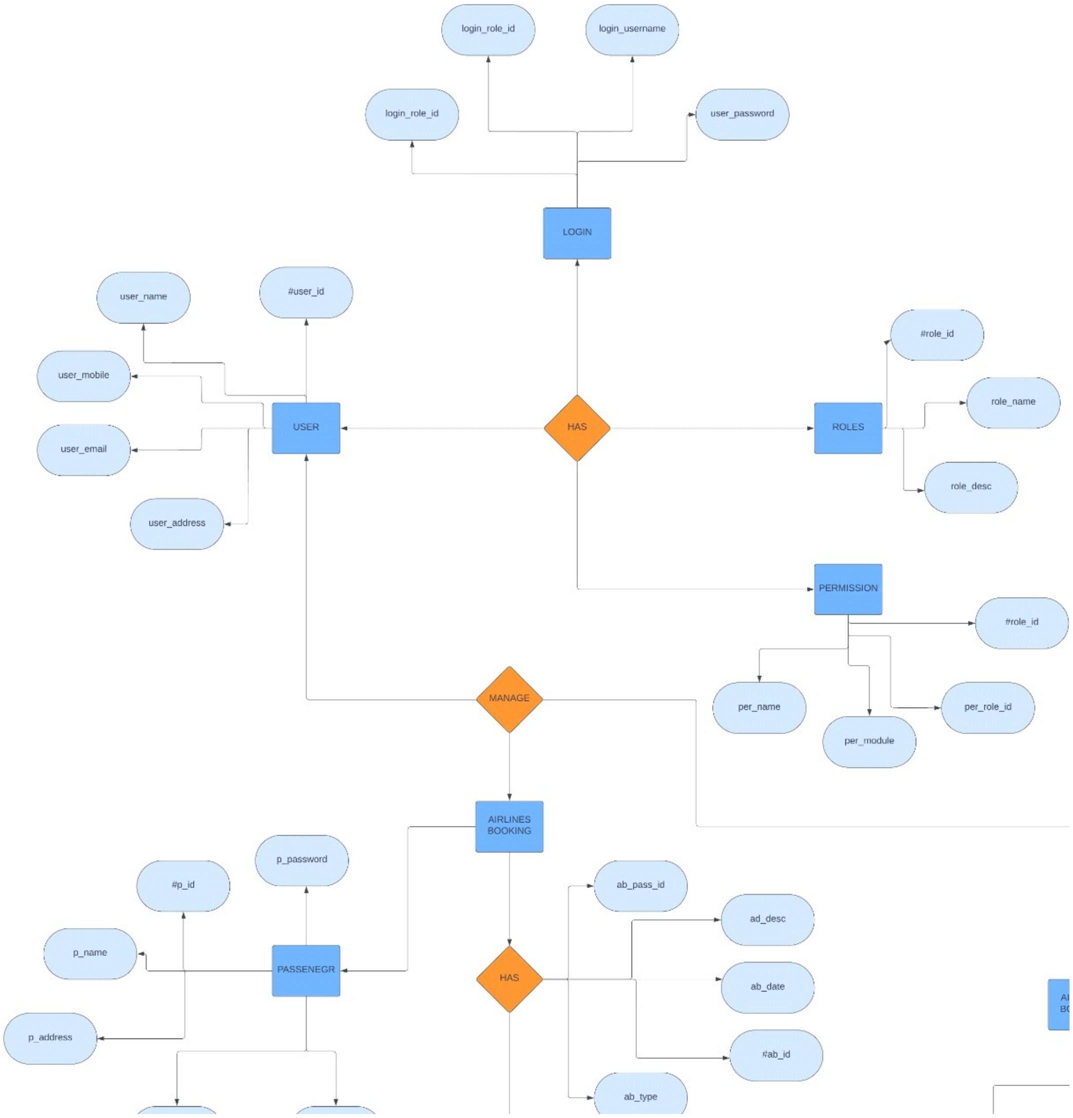
- **Entity:**
- An entity is a real-world object or concept that exists independently and has distinguishable attributes.
- In a database context, an entity represents a table, and each row in that table represents a unique instance of that entity.
- For example, in a university database, entities could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

- **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.

- 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.

- 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:
  - One-to-One (1:1): Each instance of one entity is associated with exactly one instance of another entity, and vice versa.
  - 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.
  - Many-to-One (N:1): The reverse of One-to-Many; many instances of one entity are associated with one instance of another entity.
  - Many-to-Many (N:N): Many instances of one entity can be associated with many instances of another entity.

Implementation



Conclusion

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

Ans. Entity:

An entity represents a real-world object or concept that has a distinct identity, such as a person, place, thing, event, or concept.
In a database, an entity is typically mapped to a table.
Example entities: Employee, Customer, Product, Order.

Attributes:
Attributes are properties or characteristics that describe an entity. They represent the various pieces of information we want to store about each instance of an entity.
Attributes have types, which define the kind of data they can hold (e.g., integer, string, date).
Example attributes:
For an Employee entity: EmployeeID (integer), FirstName (string), LastName (string), DateOfBirth (date), Department (string).
For a Product entity: ProductID (integer), Name (string), Price (float), QuantityInStock (integer).

Relationship:
Relationships define how entities are connected or associated with each other.
Relationships can be one-to-one, one-to-many, or many-to-many.
They are typically depicted graphically using lines connecting entities in entity-relationship diagrams (ER diagrams).

- Write ER/EER diagram notations

Ans. Entity:

Represented by a rectangle.

Entity name is written inside the rectangle.

Attribute:

Represented by an oval or ellipse.

Attribute name is written inside the oval.

Primary Key:

Underline the attribute(s) that uniquely identify each instance of the entity.

Relationship:

Represented by a diamond shape connecting two entities.

Relationship type (e.g., one-to-one, one-to-many, many-to-many) is written inside the diamond.

Cardinality:

Indicates the number of instances of an entity that can be associated with another entity.

Typically shown near the ends of the relationship lines.

Notation: "1" for one instance, "N" for many instances, "0..1" for zero or one instance, "0..N" for zero to many instances.

Foreign Key:

Represented by dashed underline or simply with the attribute name followed by "(FK)".

Indicates an attribute that is a primary key in another entity (usually to establish a relationship).