**Section 1: Basics of ERD (Recall Questions)**

1. **What is an Entity-Relationship Diagram (ERD)?**

An **Entity-Relationship Diagram (ERD)** is a visual representation of data and the relationships between different entities in a database. It is used in database design to model how data is structured and related to one another. ERDs help in understanding the logical structure of a database before implementation.

**2. What are entities in an ERD?**

Entities in an ERD represent real-world objects or concepts that have distinct identities and can be stored in a database. An entity can be a **physical object (e.g., a student, an employee, a product)** or an **abstract concept (e.g., an order, an event, a course)**.

**Types of Entities:**

* **Strong Entity:** Exists independently in a database (e.g., Student, Employee).
* **Weak Entity:** Depends on a strong entity for identification (e.g., Order Item in an order).

**3. What are attributes? List different types.**

Attributes define the characteristics or properties of an entity. Each attribute holds specific data about an entity.

* **Types of Attributes:**
  1. **Simple (Atomic) Attribute:** Cannot be divided further (e.g., Name, Age, ID).
  2. **Composite Attribute:** Can be broken down into smaller components (e.g., Name → First Name, Last Name).
  3. **Derived Attribute:** Computed from other attributes (e.g., Age derived from Date of Birth).
  4. **Multivalued Attribute:** Can have multiple values for a single entity (e.g., Phone Numbers of an Employee).
  5. **Key Attribute:** Uniquely identifies an entity (e.g., Student ID, Employee ID).

**4. What is the difference between a weak entity and a strong entity?**

| **Feature** | **Strong Entity** | **Weak Entity** |
| --- | --- | --- |
| **Definition** | Exists independently in the database | Depends on a strong entity for identification |
| **Primary Key** | Has its own primary key | No primary key; uses a foreign key from a strong entity |
| **Representation in ERD** | Represented by a **single rectangle** | Represented by a **double rectangle** |
| **Example** | Student, Employee, Product | Order Item (depends on Order), Dependent (depends on Employee) |

**5. What is a composite attribute? Give an example.**

A **composite attribute** is an attribute that can be divided into smaller sub-attributes.

* **Example:**
  + **Address** can be broken down into **Street, City, State, and Zip Code**.
  + **Name** can be divided into **First Name, Middle Name, Last Name**.

In an ERD, a composite attribute is represented as an oval with its sub-attributes branching from it.

**6. What is a derived attribute? Give an example.**

A **derived attribute** is an attribute that is not stored in the database directly but is derived from other stored attributes.

* **Example:**
  + **Age** can be derived from **Date of Birth**.
  + **Total Price** can be derived from **Quantity × Unit Price**.

In an ERD, a derived attribute is represented by a **dashed oval**.

**7. What does cardinality represent in an ERD?**

Cardinality defines the **number of instances** of one entity that can be associated with an instance of another entity. It describes **the relationship constraints** between entities.

* **Types of Cardinality:**
  + **One-to-One (1:1):** One entity is related to only one other entity (e.g., One Person has one Passport).
  + **One-to-Many (1:M):** One entity is related to multiple other entities (e.g., One Teacher teaches many Students).
  + **Many-to-Many (M:N):** Many entities are related to many other entities (e.g., Students enroll in multiple Courses, and Courses have multiple Students).

In an ERD, **cardinality is shown using symbols (e.g., 1, M, N) on the connecting lines between entities**.

**8. What are relationship types in ERD?**

A **relationship type** represents how two entities are related to each other in an ERD.

* **Types of Relationships:**
  1. **One-to-One (1:1):** Each entity instance is associated with only one instance of another entity (e.g., A person has one passport).
  2. **One-to-Many (1:M):** One entity instance is related to multiple instances of another entity (e.g., A department has many employees).
  3. **Many-to-Many (M:N):** Many instances of one entity are related to many instances of another entity (e.g., Students enroll in multiple courses, and courses have multiple students).

Relationships are represented by **diamonds** in an ERD, with connecting lines showing entity participation.

**9. What symbol is used to represent entities in an ERD?**

Entities are represented by **rectangles** in an ERD.

* **Strong entities** are shown using a **single rectangle**.
* **Weak entities** are shown using a **double rectangle**.

**10. What is a primary key and how is it shown in an ERD?**

A **primary key** is a unique identifier for each record in an entity. It ensures that no two rows have the same value in this column.

* **Example:**
  + In a **Student** entity, Student\_ID is a primary key because it uniquely identifies each student.
  + In an **Employee** entity, Employee\_ID is the primary key.

In an ERD, **a primary key is underlined** to distinguish it from other attributes.

**Section 2: Explain ERD Components**

**11. Explain the difference between a 1:1, 1:M, and M:M relationship.**

In an ERD, relationships define how entities are associated with each other. There are three main types:

| Relationship Type | Description | Example | ERD Representation |
| --- | --- | --- | --- |
| One-to-One (1:1) | Each entity in Set A is related to at most one entity in Set B, and vice versa. | A Person has one Passport. | A straight line between entities with "1" on both ends. |
| One-to-Many (1:M) | One entity in Set A can be related to multiple entities in Set B, but each entity in Set B is related to only one entity in Set A. | A Teacher teaches multiple Students, but each student has only one teacher. | A line with "1" on one end and "M" on the other. |
| Many-to-Many (M:N) | Many entities in Set A can be related to many entities in Set B. | Students enroll in multiple Courses, and each course has multiple students. | A line with "M" on both ends (usually handled with a junction table). |

**12. How do you represent a multi-valued attribute in an ERD?**

A **multi-valued attribute** is an attribute that can have multiple values for a single entity instance.

* **Representation in ERD:**
* It is represented by a **double oval** connected to the entity.
* Example: An **Employee** can have multiple **Phone Numbers**.

**Example:**

Employee

|-- (Emp\_ID) - Primary Key

|-- Name

|-- Salary

|-- (Multi-valued Attribute)

Here, {Phone\_Number} is a multi-valued attribute, meaning an employee can have multiple phone numbers.

**13. Why do we use a junction table (bridge table) in M:M relationships?**

A **junction table** (or **bridge table**) is used to resolve a **Many-to-Many (M:N)** relationship by breaking it down into two **One-to-Many (1:M)** relationships.

**Why is it needed?**

* Databases do not directly support M:M relationships because they cause redundancy and inconsistency.
* The junction table ensures data integrity and eliminates duplication.

**How does it work?**

* A new table is created, storing the Primary Keys of both related entities as Foreign Keys.
* This junction table creates two 1:M relationships with the original entities.

**Example:**

* Students and Courses have an M:N relationship (A student can enroll in multiple courses, and a course can have multiple students).
* We create a junction table called Student\_Course:

Student\_Course

|-- Student\_ID (FK)

|-- Course\_ID (FK)

Now, the **Many-to-Many** relationship is converted into two **One-to-Many** relationships:

1. **Student → Student\_Course (1:M)**
2. **Course → Student\_Course (1:M)**

**14. Explain how a foreign key appears in an ERD.**

A foreign key (FK) is an attribute in one entity that refers to the primary key (PK) of another entity. It establishes relationships between tables.

**Representation in ERD:**

* A dashed line or normal line connects the foreign key to the primary key of the related entity.
* The foreign key attribute is not underlined (only primary keys are underlined).

**Example:**

* A Department has multiple Employees, so Dept\_ID in the Employee table is a foreign key that references Dept\_ID in the Department table.

|  |  |
| --- | --- |
| Department  |-- (Dept\_ID) - Primary Key  |-- Dept\_Name | Employee  |-- (Emp\_ID) - Primary Key  |-- Name  |-- Salary  |-- Dept\_ID (Foreign Key referencing Department) |

In the ERD, Dept\_ID in Employee is a foreign key that links to Dept\_ID in Department.

**15. What happens if we do not normalize an ERD properly?**

Normalization is the process of organizing a database to reduce redundancy and improve integrity. If an ERD is not normalized properly, several issues arise:

1. **Data Redundancy:**
   * The same data is stored in multiple places, leading to storage waste and update anomalies.
   * Example: If an employee's department is stored in both Employee and Project tables, changing it in one place but not the other leads to inconsistencies.
2. **Insertion Anomaly:**
   * If an ERD is not properly designed, inserting a new record may require irrelevant or unnecessary data.
   * Example: If a Student table contains a Course Name, adding a new course means adding a dummy student, which is illogical.
3. **Update Anomaly:**
   * If the same data appears in multiple places, updating it requires modifying multiple rows, increasing the risk of errors.
   * Example: Changing a customer’s address in one table but not in another leads to inconsistent data.
4. **Deletion Anomaly:**
   * Deleting data may unintentionally remove important information.
   * Example: If an Order entity stores both Customer Details and Order Items, deleting an order might remove a customer’s information, even if they placed multiple orders..

**Solution:**

* Use Normalization Forms (1NF, 2NF, 3NF, BCNF) to eliminate redundancies and ensure a well-structured ERD.

**16. What is the role of a primary key in entity identification?**

A **primary key** is a unique identifier for each record in an entity. It plays a critical role in entity identification by ensuring that each instance (or row) of an entity can be uniquely distinguished from others.

* **Role in Entity Identification:**
  + It ensures that each record in a table is uniquely identifiable.
  + It acts as the reference point for foreign key relationships, linking the entity to other entities in the database.
  + It guarantees data integrity by ensuring no two records have the same value for the primary key.
  + It improves performance by making it easier to index and search for records.

**Example:**  
In a Customer entity, Customer\_ID might be the primary key, uniquely identifying each customer. No two customers can have the same Customer\_ID. This uniqueness allows other tables (e.g., Orders) to reference this Customer\_ID through a foreign key.

**17. What are the best practices for designing an ERD?**

Designing an effective ERD requires careful planning to ensure clarity, efficiency, and scalability. Here are some **best practices** for designing an ERD:

1. **Identify Entities and Relationships First:**
   * Identify the main entities of the system and their relationships before diving into attributes.
   * Ensure that the **relationships** between entities are clear and logical.
2. **Use Correct Symbols:**
   * Use the correct symbols for entities, attributes, relationships, primary keys, etc. This makes the ERD easy to understand.
     + **Entities**: Rectangles.
     + **Attributes**: Ovals.
     + **Relationships**: Diamonds.
     + **Primary Key**: Underlined attribute.
     + **Foreign Key**: Dashed line or normal line connecting to the related entity.
3. **Normalize the Data Model:**
   * Follow normalization principles to reduce data redundancy and ensure data integrity.
   * This avoids issues like update anomalies, insertion anomalies, and deletion anomalies.
4. **Show Cardinality Clearly:**
   * Use proper cardinality notation (1:1, 1:M, M:N) to show the number of instances of one entity that can relate to the instances of another.
   * This ensures the relationships are clearly defined.
5. **Represent Weak Entities Properly:**
   * Use double rectangles for weak entities and include the primary key of the strong entity they depend on.

**18. How does a weak entity depend on a strong entity?**

A weak entity is an entity that cannot be uniquely identified by its own attributes alone. It depends on a strong entity (also called a parent entity) for identification. The weak entity uses a combination of its own partial key and the primary key of the strong entity to create a unique identifier.

**Dependence on Strong Entity:**

* The weak entity cannot exist without the strong entity. The strong entity provides the context or identity for the weak entity.
* The weak entity relies on the strong entity's primary key, and this primary key becomes part of the weak entity's composite primary key.
* In ERD diagrams, a weak entity is depicted with a double rectangle, and the relationship between the weak entity and the strong entity is shown with a double diamond.

**Example:**

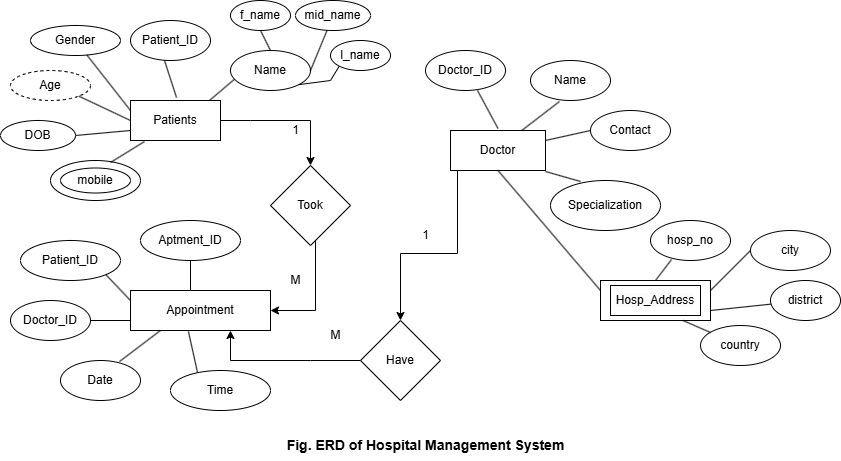
**Employee (strong entity) and Dependent (weak entity):**  
- An Employee can have multiple Dependents (e.g., spouse, children), but a dependent cannot exist without being related to an employee.  
- In the ERD, the Dependent entity will use the Employee\_ID (primary key of the Employee entity) to form a composite key for uniquely identifying the dependent.

|  |  |
| --- | --- |
| Employee  |-- Emp\_ID (PK)  |-- Name | Dependent  |-- Dependent\_ID (partial key)  |-- Emp\_ID (FK, part of PK) |

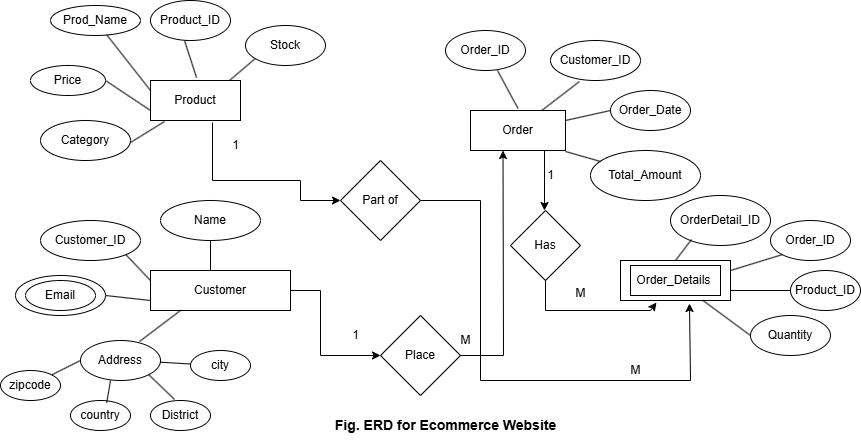
In this case, the **Dependent** cannot exist without the **Employee**, and the **Employee\_ID** acts as part of the **Dependent’s primary key**.

**Section 3: Design-Based Questions**

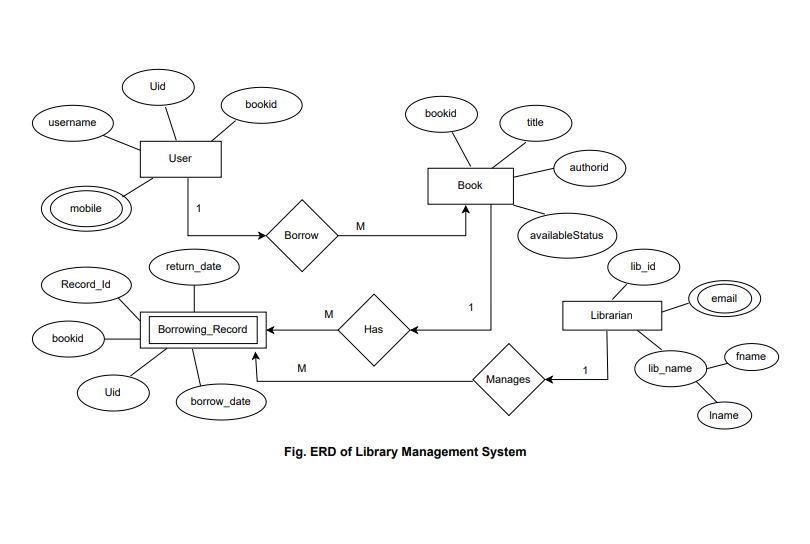
**19. ERD for a Hospital Management System**

****

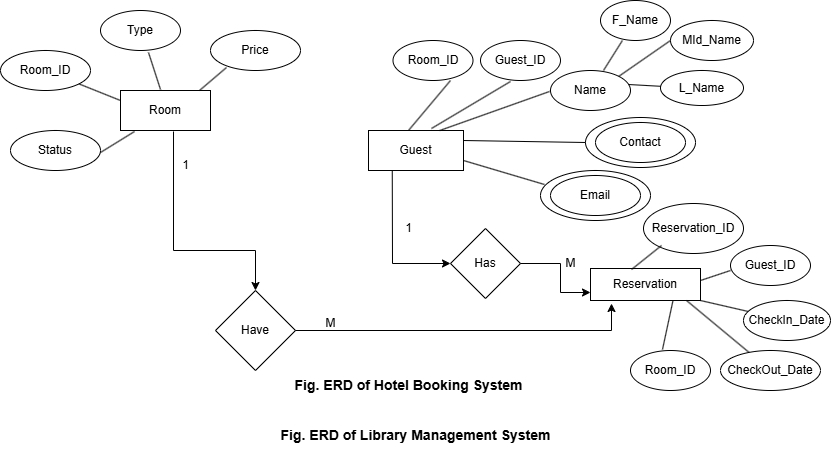
**20. ERD for an E-commerce Website**

****

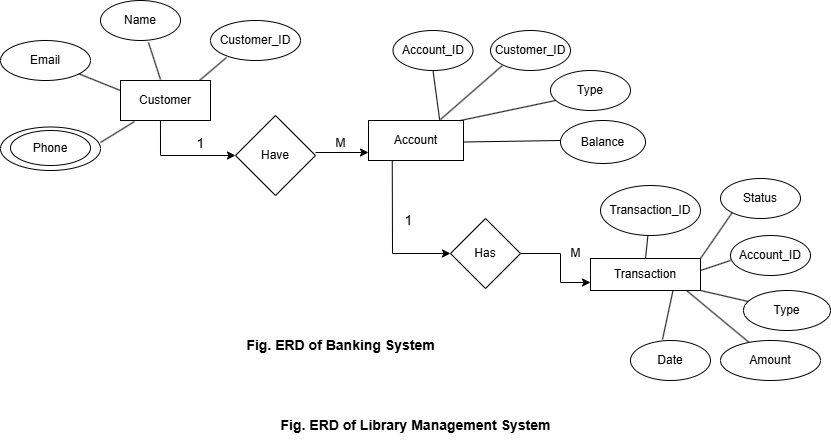
**21. ERD for a Library Management System**

****

**22. ERD for a Hotel Booking System**

****

**23. ERD for a Banking System**

****