**1. Define Database and Database Management System (DBMS)**

* **Database**: A database is an organized collection of structured information or data, typically stored electronically in a computer system. Databases are often managed using a Database Management System (DBMS), which provides tools for storing, modifying, and extracting data. The structure of a database is usually designed using tables, schemas, views, and indexes.
* **Database Management System (DBMS)**: A DBMS is software that provides an interface for users to interact with databases. It helps in managing, storing, retrieving, and manipulating data efficiently. It ensures data integrity, security, and consistency while allowing multiple users to access the data concurrently. Examples of DBMS include MySQL, PostgreSQL, Oracle, and Microsoft SQL Server.

**2. Describe Primary Key**

A **Primary Key** is a unique identifier for a record in a relational database table. It ensures that each record can be uniquely identified and helps in maintaining data integrity.

* A primary key column cannot contain **NULL** values.
* Every table can have only **one primary key**, which may consist of a single or multiple columns (composite key).
* It guarantees that the data in that column(s) will be unique across the table.

**3. E-R Diagram for Library Management System**

Here is a simplified outline of an Entity-Relationship (E-R) diagram for a Library Management System:

Entities:

* **Book**: Stores information about books.
  + Attributes: Book\_ID (PK), Title, Author, ISBN, Publication\_Year
* **Member**: Stores information about library members.
  + Attributes: Member\_ID (PK), Name, Address, Email, Phone
* **Loan**: Represents the loaning of books to members.
  + Attributes: Loan\_ID (PK), Member\_ID (FK), Book\_ID (FK), Issue\_Date, Return\_Date

Relationships:

* A **Member** can borrow multiple **Books** (1 to Many between Member and Loan).
* A **Book** can be borrowed multiple times by different **Members** (1 to Many between Book and Loan).
* The **Loan** entity connects **Member** and **Book**.

**E-R Diagram:**

scss

Copy code

+------------+ +-----------+ +---------+

| Member | | Loan | | Book |

+------------+ +-----------+ +---------+

| Member\_ID (PK) |----<| Loan\_ID (PK) | | Book\_ID (PK) |

| Name | | Member\_ID (FK) | | Title |

| Address | | Book\_ID (FK) | | Author |

| Email | | Issue\_Date | | ISBN |

| Phone | | Return\_Date | | Publication\_Year|

+------------+ +-----------+ +---------+

**4. Normalize the Following Table of EMP to 3NF**

The given table is:

scss

Copy code

EMP(empno, ename, mgr, job, deptno, loc, dname)

**Step 1: First Normal Form (1NF)**  
To achieve 1NF, we need to ensure all attributes contain atomic values and each record is unique.

* Remove repeating groups and ensure each column contains atomic values.
* **EMP** already satisfies 1NF, as each attribute contains atomic values (single values per field).

**Step 2: Second Normal Form (2NF)**  
To achieve 2NF, the table must first be in 1NF, and there should be no partial dependencies (i.e., non-prime attributes should depend on the entire primary key).

* **empno** is the primary key for the **EMP** table.
* However, the **dname** and **loc** depend only on **deptno**, not the entire primary key.
* Hence, we decompose the table to remove partial dependencies.

Tables in 2NF:

1. **EMP**:

scss

Copy code

empno (PK), ename, mgr, job, deptno (FK)

1. **DEPT**:

scss

Copy code

deptno (PK), dname, loc

**Step 3: Third Normal Form (3NF)**  
To achieve 3NF, the table must first be in 2NF, and there should be no transitive dependencies (i.e., non-prime attributes should not depend on other non-prime attributes).

* In **EMP**, **mgr** depends on **empno**, but **mgr** is a reference to **empno**, which is already in the **EMP** table.
* This is a transitive dependency, so we need to remove it.

Decompose into the following tables:

1. **EMP**:

scss

Copy code

empno (PK), ename, job, deptno (FK), mgr (FK)

1. **DEPT**:

scss

Copy code

deptno (PK), dname, loc

1. **MANAGER**:

scss

Copy code

mgr (PK), empno (FK)

* Here, **mgr** is now treated as a foreign key referring to **empno**, so no transitive dependencies remain.

**Final Schema in 3NF:**

1. **EMP**:

scss

Copy code

empno (PK), ename, job, deptno (FK), mgr (FK)

1. **DEPT**:

scss

Copy code

deptno (PK), dname, loc

1. **MANAGER**:

scss

Copy code

mgr (PK), empno (FK)