1. What do you understand By Database

Ans.A database is an organized collection of data that is stored and accessed electronically.

It is designed to efficiently store, retrieve, manage, and update information.

2.What is Normalization?

Ans.Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity.

It involves dividing a large table into smaller, related tables and defining relationships between them.

3. What is Difference between DBMS and RDBMS?

Ans.The main difference between DBMS (Database Management System) and RDBMS (Relational Database Management System) lies in how they store and manage data.

 **DBMS**: A software system that allows users to create, store, manage, and retrieve data from databases.

 **RDBMS**: A type of DBMS that organizes data into **tables** (i.e., rows and columns) and maintains **relationships** between the data using keys.

4. What is MF Cod Rule of RDBMS Systems?

Ans. There is no known rule called **"MF Cod Rule"** in RDBMS. However, it's likely you're referring to **Dr. E. F. Codd’s Rules**, also known as **Codd's 12 Rules**, which define what a **true Relational Database Management System (RDBMS)** should support.

5. What do you understand By Data Redundancy?

Ans. **Data Redundancy** means **storing the same piece of data in multiple places** within a database or system.

6. What is DDL Interpreter?

Ans. A **DDL Interpreter** is a **component of a database management system (DBMS)** that **reads and processes Data Definition Language (DDL) statements**.

7. What is DML Compiler in SQL?

Ans. A **DML Compiler** is a component of a **Database Management System (DBMS)** that **translates Data Manipulation Language (DML) statements into low-level instructions** that the **query processor** and **storage engine** can execute.

8. What is SQL Key Constraints writing an Example of SQL Key Constraints?

Ans. **SQL Key Constraints** are rules applied to **columns in a table** to enforce **uniqueness**, **relationships**, and **data integrity** in a relational database.

| **Constraint** | **Description** |
| --- | --- |
| **PRIMARY KEY** | Uniquely identifies each row in a table. Cannot be NULL. |
| **FOREIGN KEY** | Links one table to another. Ensures referential integrity. |
| **UNIQUE** | Ensures all values in a column (or set of columns) are unique. |
| **NOT NULL** | Ensures that a column cannot have NULL values. |
| **CHECK** | Validates data based on a condition (e.g., salary > 0). *(Not a key but often grouped)* |

9. What is save Point? How to create a save Point write a Query?

Ans. A **SAVEPOINT** is a **temporary marker within a transaction** that allows you to **roll back to a specific point** without undoing the entire transaction.

| **Command** | **Purpose** |
| --- | --- |
| SAVEPOINT | Creates a rollback point |
| ROLLBACK TO | Rolls back to a specific savepoint |
| RELEASE SAVEPOINT | Removes a savepoint (optional) |
| COMMIT | Finalizes the transaction |
| ROLLBACK | Cancels the whole transaction |

10. .What is trigger and how to create a Trigger in SQL?

Ans. A **Trigger** is a special kind of **stored procedure** that **automatically runs** (or “fires”) in response to certain events on a table or view.

| **Keyword** | **Meaning** |
| --- | --- |
| BEFORE / AFTER | When the trigger runs (relative to the event) |
| NEW | Refers to the **new row** being inserted or updated |
| OLD | Refers to the **old row** being updated or deleted |
| FOR EACH ROW | Trigger runs **once per affected row** |