**1. What do you understand By Database:**

A database is an organized collection of structured information, typically stored electronically in a computer system. It is designed to efficiently manage, store, and retrieve data according to the needs of users and applications. Databases serve as repositories for various types of data, ranging from simple text and numbers to multimedia files and complex structures. They provide mechanisms for data manipulation, such as querying, updating, and deleting, as well as ensuring data integrity, security, and concurrency control. Databases are widely used in various fields, including business, science, education, and government, to store and manage large volumes of information effectively.

**2. What is Normalization:**

Normalization is the process of organizing a database schema to reduce redundancy and dependency by breaking down large tables into smaller ones and defining relationships between them. The goal of normalization is to design a database schema that minimizes data duplication and anomalies, ensuring data integrity and efficiency in data retrieval and manipulation operations. There are several normal forms, each representing a higher level of normalization, with the most common ones being First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), and Boyce-Codd Normal Form (BCNF).

**3. What is the Difference between DBMS and RDBMS:**

DBMS (Database Management System) is a software system that provides an interface for managing databases, whereas RDBMS (Relational Database Management System) is a type of DBMS that stores data in the form of tables with rows and columns, and enforces the relational model principles, including integrity constraints and referential integrity. In other words, RDBMS is a subset of DBMS that specifically deals with relational databases. While all RDBMS are DBMS, not all DBMS are RDBMS.

**4. What is the ACID Rule of RDBMS Systems:**

The ACID (Atomicity, Consistency, Isolation, Durability) properties are a set of principles that ensure reliability and consistency in database transactions. In the context of RDBMS systems, the ACID rule ensures that transactions are processed reliably and consistently, even in the presence of system failures or concurrent access by multiple users.

- Atomicity ensures that transactions are treated as indivisible units, either fully completed or fully aborted.

- Consistency ensures that transactions preserve the integrity of the database, maintaining valid relationships and constraints.

- Isolation ensures that concurrent transactions do not interfere with each other, providing a level of isolation and concurrency control.

- Durability ensures that committed transactions are permanently saved and not lost, even in the event of system crashes or failures.

**5. What do you understand By Data Redundancy:**

Data redundancy refers to the duplication of data within a database or across multiple databases. It occurs when the same piece of data is stored multiple times in different locations or tables within a database. Data redundancy can lead to several issues, including increased storage space, data inconsistency, and update anomalies. Normalization techniques are often employed to minimize data redundancy and improve database efficiency.

**6. What is DDL Interpreter:**

DDL (Data Definition Language) Interpreter is a component of a database management system responsible for processing and executing Data Definition Language commands. DDL commands are used to define and modify the structure and schema of a database, including creating and altering tables, views, indexes, and constraints. The DDL interpreter translates DDL commands into low-level instructions that interact with the database's storage engine to implement the specified changes to the database schema.

**7. What is DML Compiler in SQL:**

DML (Data Manipulation Language) Compiler is a component of a database management system responsible for processing and executing Data Manipulation Language commands. DML commands are used to retrieve, insert, update, and delete data from a database. The DML compiler translates DML commands written in SQL (Structured Query Language) into low-level instructions that interact with the database's storage engine to perform the specified data manipulation operations.

**8. What is SQL Key Constraints writing an Example of SQL Key Constraints:**

SQL Key Constraints are rules defined on columns in a table to enforce data integrity and maintain relationships between tables. There are several types of key constraints in SQL, including Primary Key, Foreign Key, Unique Key, and Check Constraint. Here's an example of SQL Key Constraints:

```sql

CREATE TABLE employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES departments(department\_id)

);

```

In this example:

- `employee\_id` column is defined as the Primary Key, ensuring uniqueness and acting as the identifier for each employee.

- `department\_id` column is defined as a Foreign Key, establishing a relationship between the `employees` table and the `departments` table.

**9. What is Save Point and How to create a Save Point write a Query:**

A savepoint is a point within a transaction where you can mark a specific point to which you can later roll back the transaction if needed. It allows you to divide a transaction into smaller segments and rollback to a specific point without rolling back the entire transaction. Here's how you can create a savepoint in SQL:

```

SAVEPOINT my\_savepoint;

```

This SQL command creates a savepoint named `my\_savepoint` within the current transaction.

**10. What is a Trigger and How to create a Trigger in SQL:**

A trigger is a database object that automatically executes a specified set of actions in response to certain database events, such as INSERT, UPDATE, or DELETE operations on a table. Triggers are used to enforce data integrity, implement business rules, or log changes to the database. Here's how you can create a trigger in SQL:

```

CREATE TRIGGER trigger\_name

AFTER INSERT ON table\_name

FOR EACH ROW

BEGIN

-- Trigger action statements

END;

```

In this example:

- `trigger\_name` is the name of the trigger.

- `AFTER INSERT ON table\_name` specifies the event that triggers the execution of the trigger.

- `FOR EACH ROW` indicates that the trigger should be executed for each row affected by the triggering event.

- The `BEGIN` and `END` keywords enclose the statements to be executed when the trigger is fired.