Database Question – Answer

Q-1) What do you understand By Database?

Ans:

- A database is an organized collection of data that allows for efficient storage, retrieval, and management of information.
- It is designed to handle large amounts of data and can be used to store various types of information, such as text, numbers, images, or other formats.
- Databases are managed using Database Management Systems (DBMS), which provide tools to create, update, and query the data.

Q-2) What is Normalization?

Ans:

- Normalization is a process in database design used to organize data to reduce redundancy and improve data integrity.
- It involves dividing a database into multiple related tables and defining relationships between them.
- The main goal is to eliminate duplicate data and ensure that data dependencies are logical, leading to more efficient and consistent data storage.
- There are different levels, called normal forms, that represent the extent to which a database is normalized.

Q-3) What is Difference between DBMS and RDBMS?

Ans:

DBMS (Database Management System)	RDMS (Relational Database Management System)
Data is stored in files or flat tables.	Data is stored in tabular form with rows and columns.
Does not enforce relationships between data.	Enforces relationships using primary and foreign keys.
Suitable for small-scale applications.	Suitable for large-scale applications with complex data relationships.
Does not support normalization effectively.	Supports normalization, reducing redundancy.
Examples: File system, XML	Examples: MySQL, Oracle, SQL Server

Q-4) What is MF Cod Rule of RDBMS Systems?

Ans:

- E.F. Codd's Rules define the guidelines for a system to qualify as a Relational Database Management System (RDBMS).
- There are 12 rules (numbered from 0 to 12), which outline the features and principles required for a database system to be considered truly relational.
- These rules collectively ensure that the system supports data integrity, consistency, and efficient data handling in a relational model.

Q-5) What do you understand By Data Redundancy?

Ans:

- Data redundancy refers to the unnecessary duplication of data within a database or data storage system.
- It occurs when the same piece of data is stored in multiple places, leading to increased storage space and potential inconsistencies when the data is updated.
- Redundancy can make the database less efficient and more difficult to maintain.
- Normalization techniques are often used to minimize data redundancy by organizing data into related tables and eliminating duplicate entries.

Q-6) What is DDL Interpreter?

Ans:

- A DDL Interpreter is a component of a Database Management System (DBMS) that processes Data Definition Language (DDL) commands.
- DDL commands are used to define and manage the structure of database objects, such as tables, indexes, views, and schemas.
- The DDL Interpreter interprets these commands and converts them into a set of operations that modify the database's metadata.
- Examples of DDL commands include CREATE, ALTER, DROP, and TRUNCATE, which help in creating or modifying database structures.
- The DDL Interpreter ensures that these changes are properly executed in the database system.

Q-7) What is DML Compiler in SQL?

Ans:

- A DML Compiler is a component of a Database Management System (DBMS) responsible for processing Data Manipulation Language (DML) commands.
- DML commands are used to manage and manipulate the data stored in a database. This includes operations such as:
  - **SELECT:** Retrieving data from the database.
  - **INSERT:** Adding new records to a table.
  - **UPDATE:** Modifying existing records.
  - **DELETE:** Removing records from a table.
- The DML Compiler translates these high-level DML statements into a lower-level representation, often in the form of query execution plans.
- These plans optimize how the database engine retrieves or modifies the data, ensuring efficient processing and execution of the DML commands.

Q-8) What is SQL Key Constraints writing an Example of SQL Key Constraints.

Ans:

- **SQL key constraints** are rules applied to columns in a database table to ensure data integrity. The main types include:
  1. **Primary Key**: Uniquely identifies each record in a table and cannot contain NULL values.
  2. **Foreign Key**: Establishes a link between two tables, referencing the primary key of another table to maintain referential integrity.
  3. **Unique Key**: Ensures that all values in a column are unique, allowing NULL values.
  4. **Check Constraint**: Validates that all values in a column meet a specified condition.
- **Example:**

```
CREATE TABLE Employees (
  EmployeeID INT PRIMARY KEY,
  Email VARCHAR(100) UNIQUE,
  Salary DECIMAL(10, 2) CHECK (Salary > 0),
```

```
    FOREIGN KEY (DepartmentID) REFERENCES
Departments(DepartmentID)
);
```

Q-9) What is save Point? How to create a save Point write a Query?

Ans:

- A **SAVEPOINT** is a marker within a database transaction that allows for partial rollback to a specific point, helping manage complex transactions and maintain data integrity.
- **How to create:**

```
BEGIN;

INSERT INTO Employees (EmployeeID, FirstName, LastName) VALUES (1,
'John', 'Doe');
SAVEPOINT before_update;

UPDATE Employees SET LastName = 'Smith' WHERE EmployeeID = 1;

-- Rollback to the SAVEPOINT if needed
ROLLBACK TO SAVEPOINT before_update;

COMMIT;
```

Q-10) What is trigger and how to create a Trigger in SQL?

Ans:

- A **trigger** is a stored procedure that automatically executes in response to specific events on a table, such as **INSERT**, **UPDATE**, or **DELETE**. Triggers are useful for enforcing business rules and maintaining data integrity.
- **How to create:**

```
CREATE TRIGGER log_employee_insert
AFTER INSERT ON Employees
FOR EACH ROW
BEGIN
   INSERT INTO EmployeeAudit (EmployeeID, ChangeTime, Action)
   VALUES (NEW.EmployeeID, NOW(), 'INSERT');
END;
```