1). What do you understand By Database

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system(DBMS) Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

2). What is Normalization?

- Normalization is a process of decomposing the relations into relations with fewer attributes.
- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
- Normalization divides the larger table into smaller and links them using relationships.
- The normal form is used to reduce redundancy from the database table.

3). What is Difference between DBMS and RDBMS?

DBMS:

- It manages data using various data models, such as hierarchical, network, or object-oriented. The relationships between data are not well-defined, and it may not support a tabular structure.
- It may not enforce data integrity constraints, such as unique keys, foreign keys, and referential integrity, leaving the responsibility to the application or user.
- It may or may not support normalization, a process to organize data to reduce redundancy and dependency.
- It might have its query language, specific to the data model it supports.
- It may not be as scalable or performant as RDBMS in handling large datasets and complex queries efficiently.
- Ex. File-based systems, like Microsoft Access or FoxPro.

RDBMS:

- It specifically manages data using a relational model, where data is organized into tables with rows and columns. The relationships between tables are well-defined, and it enforces a tabular structure.
- It enforces data integrity constraints to maintain the accuracy and reliability of data. This includes enforcing primary keys, unique constraints, and relationships between tables.
- It typically supports normalization, following specific rules to organize data efficiently and minimize redundancy.
- It uses SQL (Structured Query Language) as the standard query language for interacting with the database, providing a standardized way to manage and manipulate relational data.
- It is designed to handle large datasets and complex queries, with optimization techniques like indexing, query optimization, and transaction management.
- Ex. MySQL, PostgreSQL, Oracle, Microsoft SQL Server.

4. What is MF Cod Rule of RDBMS Systems?

Rule 1: The Information Rule - All information, whether it is user information or metadata, that is stored in a database must be entered as a value in a cell of a table. It is said that everything within the database is organized in a table layout.

Rule 2: The Guaranteed Access Rule - Each data element is guaranteed to be accessible logically with a combination of the table name, primary key (row value), and attribute name (column value).

Rule 3: Systematic Treatment of NULL Values - Every Null value in a database must be given a systematic and uniform treatment.

Rule 4: Active Online Catalog Rule - The database catalog, which contains metadata about the database, must be stored and accessed using the same relational database management system.

Rule 5: The Comprehensive Data Sublanguage Rule - A crucial component of any efficient database system is its ability to offer an easily understandable data manipulation language (DML) that facilitates defining, querying, and modifying information within the database.

Rule 6: The View Updating Rule - All views that are theoretically updatable must also be updatable by the system.

Rule 7: High-level Insert, Update, and Delete - A successful database system must possess the feature of facilitating high-level insertions, updates, and deletions that can grant users the ability to conduct these operations with ease through a single query.

Rule 8: Physical Data Independence - Application programs and activities should remain unaffected when changes are made to the physical storage structures or methods.

Rule 9: Logical Data Independence - Application programs and activities should remain unaffected when changes are made to the logical structure of the data, such as adding or modifying tables.

Rule 10: Integrity Independence - Integrity constraints should be specified separately from application programs and stored in the catalog. They should be automatically enforced by the database system.

Rule 11: Distribution Independence - The distribution of data across multiple locations should be invisible to users, and the database system should handle the distribution transparently.

Rule 12: Non-Subversion Rule - If the interface of the system is providing access to low-level records, then the interface must not be able to damage the system and bypass security and integrity constraints.

5. What do you understand By Data Redundancy?

Data redundancy occurs when the same piece of data exists in multiple places, whereas data inconsistency is when the same data exists in different formats in multiple tables. Unfortunately, data redundancy can cause data inconsistency, which can provide a company with unreliable and/or meaningless information.

6. What is DDL Interpreter?

It interprets the DDL (Data Definition Language) Instructions and stores the record in a data dictionary (in a table containing meta-data)

7. What is DML Compiler in SQL?

DML Compiler: Translates DML statements in a query language within low level instructions understandable through the query evaluation engine. Attempts to transforms users request within an equivalent and well-organized from for executing the query understandable through Data Manager, Interprets DDL statements and records them within a set of tables containing Meta data in a form that can be used through other elements of a DBMS.

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

SQL key constraints are rules applied to columns in a relational database table to ensure the integrity and consistency of the data stored in that table. Keys are used to establish relationships between tables and to enforce uniqueness and referential integrity.

some common types of key constraints in SQL:

Primary Key Constraint:
A primary key is a column or a set of columns that uniquely identifies each row in a table.
It must have unique values, and it cannot contain NULL values.
Each table can have only one primary key.

Example

```
CREATE TABLE Students (
    StudentID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50)
);
```

Unique Constraint:
A unique constraint ensures that all values in a column (or a set of columns) are unique. Unlike the primary key, a unique constraint allows NULL values, but if a value is present, it must be unique.

Example

```
CREATE TABLE Employees (
    EmployeeID INT UNIQUE,
    FirstName VARCHAR(50),
    LastName VARCHAR(50)
);
```

Foreign Key Constraint:

A foreign key is a column or a set of columns in a table that refers to the primary key of another table. It establishes a relationship between the two tables, enforcing referential integrity. The values in the foreign key column must match the values in the referenced primary key column.

Example

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

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

savepoint is a mechanism that allows you to set a point within a transaction to which you can later roll back if needed. Savepoints provide a way to create nested transactions and selectively roll back to a specific point without affecting the entire transaction.

```
/*
-- SQL Query

-- Start a transaction
BEGIN TRANSACTION;

-- Make some changes
INSERT INTO your_table (column1, column2) VALUES ('value1', 'value2');

-- Create a savepoint
SAVEPOINT my_savepoint;

-- Make more changes
UPDATE your_table SET column1 = 'new_value' WHERE some_condition;

-- If needed, roll back to the savepoint
-- ROLLBACK TO my_savepoint;
```

-- Continue with the transaction or commit
-- COMMIT;

-- Roll back the entire transaction if needed
-- ROLLBACK;
*/

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

A trigger in SQL is a set of instructions or a set of actions that are automatically executed ("triggered") in response to specific events on a particular table or view in a database. These events can include INSERT, UPDATE, DELETE, or a combination of these operations. Triggers are often used to enforce business rules, perform auditing, or automate certain actions when data modifications occur.

Trigger Syntax

```
CREATE [ OR REPLACE ] TRIGGER trigger_name
{ BEFORE | AFTER | INSTEAD OF } { event [ OR ... ] }
ON table_name
[ REFERENCING { OLD | NEW } { TABLE | ROW | STATEMENT } AS
transition_relation_name ]
[ FOR EACH { ROW | STATEMENT } ]
[ WHEN (condition) ]
EXECUTE FUNCTION function_name (arguments);
```

CREATE TRIGGER: Defines a new trigger.

trigger_name: The name of the trigger.

BEFORE | AFTER | INSTEAD OF: Specifies when the trigger should be executed.

event: The event that triggers the execution (e.g., INSERT, UPDATE, DELETE).

table_name: The table on which the trigger is defined.

REFERENCING: Specifies how the OLD and NEW values are referenced in the trigger.

FOR EACH: Specifies whether the trigger should be executed for each row affected by the triggering event or once per statement.

WHEN: An optional condition that, if specified, determines whether the trigger should execute based on the given condition.

EXECUTE FUNCTION: Specifies the function or procedure to be executed when the trigger fires.