1.What do you understand By Database

• A database is an organized collection of data stored and accessed electronically. It provides a way to store, organize, and retrieve large amount of data efficiently.

2. What is Normalization?

• Normalization is the process of organizing data in a database to minimize redundancy and dependency. It involves breaking down a table into smaller tables and establishing relationships between them.

3.What is Difference between DBMS and RDBMS?

• DBMS:

▪ Data stored is in the file format.

▪ Individual access of data elements.

▪ No connection between data.

▪ No support for distributed database.

▪ Data stored is a small quantity.

▪ DBMS supports a single user.

▪ The software and hardware requirements are low.

▪ Example: XML, Microsoft Access.

• RDBMS:

▪ Data stored is in table format.

▪ Multiple data elements are accessible together.

▪ Data in the form of a table are linked together.

▪ Support distributed database.

▪ Data is stored in a large amount.

▪ RDBMS supports multiple users.

▪ The software and hardware requirements are higher.

▪ Example: Oracle, SQL Server.

4. What is MF Cod Rule of RDBMS Systems?

• Cod’s Rules in RDBMS:

1. Information Rule: All information is represented in ta bles.
2. .Guaranteed Access Rule: All data is accessible by tabl e name, primary key, and column name.
3. Systematic Treatment of Null Values: Nulls are handl ed consistently.
4. Dynamic Online Catalog: Metadata is stored in table s and accessible via SQL.
5. . Comprehensive Data Sublanguage Rule: The system supports a single language for data definition, manipulation , and transaction management.
6. View Updating Rule: Any view can be updated.
7. Highlevel Insert, Update, and Delete: Operations are supported by set-based, high-level language.
8. Physical Data Independence: Changes in physical storage do not affect the logical structure.
9. Logical Data Independence: Changes in logical structure do not affect applications.
10. Integrity Independence: Integrity constraints are stored in the catalog and independent of application programs.

11. Distribution Independence: Distribution of data sh ould be transparent to users.

12. NonSubversion Rule: The integrity of the database can't be subverted by bypassing the constraints usin g a lower-level language.

5. What do you understand By Data Redundancy?

• Data redundancy refers to the unnecessary duplication of data within a database or data storage system.

6. What is DDL Interpreter?

• A DDL Interpreter is a component of a database management system that interprets and executes DDL commands. These commands are used to define and manage all the structures in a database, such as tables, indexes, and constraints. DDL commands include statements like CREATE, ALTER, and DROP.

7. What is DML Compiler in SQL?

• A DML Compiler is a component of a database mana gement system that processes DML

commands. Thes e commands are used to manipulate data within the database, such as inserting, updating, deleting, and q uerying data. DML commands include SELECT, INSERT , UPDATE, and DELETE.

8. What is SQL Key Constraints writing an Example of SQL Key Constraints SQL Key Constraints are rules applied to table columns to enforce database integrity. They ensure the accuracy and reliability of the data within the database. Here ar e the main types of key constraints:

1. Primary Key: Uniquely identifies each record in a table.
2. Foreign Key: Ensures referential integrity by linking to a primary key in another table.
3. Unique Key: Ensures all values in a column or a set of columns are unique across the table.
4. . Check Constraint: Ensures the values in a column s atisfy a specific condition.

• Example: -- Create a table with a primary key CREATE TABLE Customers (

customer\_id INT PRIMARY KEY,

name VARCHAR(50),

city VARCHAR(50)

);

-- Create another table with a foreign key

CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

order\_date DATE,

customer\_id INT,

FOREIGN KEY (customer\_id) REFERENCES Customers (customer\_id)

);

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

• A savepoint in SQL is a point within a transaction to which you can later roll back. Savepoints allow you to selectively undo parts of a transaction without affect ing the entire transaction.

Syntax:

1. Create Savepoint:

SAVEPOINT savepoint\_name;

1. Rollback to Savepoint :-

ROLLBACK TO SAVEPOINT savepoint\_name;

3. Release Savepoint (Optional):

RELEASE SAVEPOINT savepoint\_name;

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

• A trigger in SQL is a stored procedure that is automat ically executed in response to certain events on a par ticular table or view. These events can be INSERT, UP DATE, or DELETE operations.

Syntax:

CREATE TRIGGER [schema\_name.]trigger\_name

ON table\_name

AFTER {[INSERT], [UPDATE], [DELETE]}

AS

BEGIN

-- SQL statements

END;