**Module 4 (Database)**

**(1) What is RDBMS?**

* RDBMS is a common type of database that stores data in table format with each table consisting of raw and column, so it can be used in relation to other stored databases.

**(2) What is SQL(Structured Query Language)?**

* SQL is a computer language for storing , manipulating and retrieving data stored in relational databases.

**(3) Write SQL Commands**.

* **DQL** – Data Query Language (Select)

**DML** – Data Manipulation Language (Insert, Update, Delete)

**DCL** – Data Control Language (Grant, Revoke)

**DDL** – Data Definition Language (Create, Alter, Drop)

**(4).What is Join?**

“A join statement is used to combine data of rows from two or more tables based on a common field between them.”

**(5).Write types of Joins.**

* **There are four types of joins:**

**1) INNER JOIN:** returns rows when there is a match in both tables.

**2) LEFT JOIN:** returns all rows from the left table, even if there are no matches in the right table.

**3) RIGHT JOIN:** returns all rows from the right table, even if there are no matches in the left table.

**4) IN:** returns rows when there is a match in one of the tables.

**(6) How many constraint? and describe itself.**

* SQL Constraints are used to specify rules for the data in table.

**1) NOT NULL**: Ensure that column cannot have null value.

**2) UNIQUE**: Ensures that values in column are different.

**3) PRIMARY KEY**: A combination of NOT NULL and UNIQUE. Uniquely identifies each row in table.

**4) FOREIGN KEY**: Uniquely identifies a row/record in another table.

**5) CHECK**: Ensures that all values in a column satisfy a specific condition.

**6) DEFAULT**: Sets a default value for column when no value is specified.

**7) INDEX**: Used to create and retrieve data from the database very quickly.

**(7) Difference between RDBMS vs DBMS.**

|  |  |
| --- | --- |
| **DBMS** | **RDBMS** |
| DBMS stores data as a file. | Data is stored in the form of tables. |
| DBMS supports single user only. | RDBMS supports multiple user. |
| Low software and hardware needs. | High hardware and software needs. |
| Data elements needs to access individually. | Multiple data elements can be accessed at the same time using SQL query. |
| Example of DBMS are a file system, XML, Windows Registry … | Example of RDBMS is MySQL, Oracle, SQL Server… |

**(8).What is an SQL Alias?**

SQL Alias is a temporary name that is used to give a table or a column a different name for the duration of a query.

Aliases is a commonly used to make column names or table names more readable , concise , or to improve query quality , especially when working with complex queries or joins.

**(9).Write a Query to create the table in SQL.**

CREATE DATABASE Testing;

USE Testing;

CREATE TABLE Student

(

SID INT(3) PRIMARY KEY,

SName VARCHAR(25),

City VARCHAR(25)

)

**(10).Write a Query to insert data into table.**

INSERT INTO Student VALUES(01,”Namrata”,”Ahmedabad”)

INSERT INTO Student VALUES(02,”Ami”,”Delhi”)

**(11).Write a Query to update data into table with validations.**

UPDATE Student SET City=”Pune” WHERE SName=”Namrata”

**(12). Write a Query to delete data into table with validations**

DELETE FROM Student WHERE SName=”Namrata”

**(13).Write a Query to Insert new column in existing table.**

ALTER TABLE Student ADD Age INT(2);

**(14).Write a Query to drop the table and database.**

DROP TABLE Student;

DROP DATABASE Testing;

**(15).Write a Query to find max and min value from table**

SELECT MAX(Age) FROM Student;

SELECT MIN(Age) FROM Student;

**(16).Create two tables named Seller and Product apply foreign key in product table . Fetch data from both table using different joins**

CREATE TABLE Seller (

Seller\_Id INT(3) PRIMARY KEY,

Seller\_Name VARCHAR(25),

Seller\_Info VARCHAR(25)

);

CREATE TABLE Product (

Product\_Id INT(3) PRIMARY KEY,

Product\_Name VARCHAR(50),

Price INT(5) ,

Seller\_Id INT(3),

FOREIGN KEY (Seller\_Id) REFERENCES Seller(Seller\_Id)

);

**INNER JOIN**

SELECT Product. Product\_Id , Product. Product\_Name , Product. Price , Selller.Seller\_Name

FROM Product

INNER JOIN Seller ON Product. Seller\_Id = Seller. Seller\_Id;

**LEFT JOIN**

SELECT Product. Product\_Id , Product. Product\_Name , Product. Price , Selller.Seller\_Name

FROM Product

LEFT JOIN Seller ON Product. Seller\_Id = Seller. Seller\_Id;

**RIGHT JOIN**

SELECT Product. Product\_Id , Product. Product\_Name , Product. Price , Selller.Seller\_Name

FROM Product

RIGHT JOIN Seller ON Product. Seller\_Id = Seller. Seller\_Id;

**FULL OUTER JOIN**

SELECT Product. Product\_Id , Product. Product\_Name , Product. Price , Selller.Seller\_Name

FROM Product

FULL OUTER JOIN Seller ON Product. Seller\_Id = Seller. Seller\_Id;