MODULE: 5 (Database) Basics of Database

1. What do you understand By Database?

It is a organized collection of data or a type of data store based on the use of a database management system, the software that interacts with ends users, applications, and the database itself to capture and analyse the data.

1. What is Normalization?

Normalization is the process of organizing data in a database. It include creating table and establishing relationship between those table according to rules designing both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

1. What is Difference between DBMS and RDBMS?

|  |  |
| --- | --- |
| **DBMS** | **RDMS** |
| * Data is stored in a database management system as a file * Data stored are either navigational or hierarchical format * Only Single user is supported * Software and hardware requirement are minimal * Normalization is not Supported | * Tables are used to store information * As it is tabular, column names as header and associated data as rows * May use numerous people * Higher hardware and software are required * Normalization is Supported |

1. What is MF Cod Rule of RDBMS Systems?

The MF Cod Rule of RDBMS Systems states that for a system to qualify as an RDBMS, it must be able to manage database entirely through the relational capabilities. Rule 0 of the MF Cod Rules states that the system must qualify as relational, as a database, and as a management system. For a system to qualify as an RDBMS, that system must use its relational facilities exclusively to manage the database.

1. What do you understand By Data Redundancy?

Data redundancy refers to the situation where the same pieces of data are stored in multiple places within a database or data storage system. This can happen intentionally or accidentally. Redundancy can be useful for data recovery in case of corruption or loss. In computer memory and storage, data redundancy allows for error correction

1. What is DDL Interpreter?

Data Definition Language (DDL) is used to create and modify the structure of [objects](https://www.techtarget.com/searchapparchitecture/definition/object) in a [database](https://www.techtarget.com/searchdatamanagement/definition/database) using predefined [commands](https://www.techtarget.com/searchwindowsserver/definition/command) and a specific syntax. DDL is a standardized language with commands to define the storage groups (stogroups), different structures and objects in a database. DDL statements create, modify and remove database objects, such as tables, indexes and stogroups.

1. What is DML Compiler in SQL?

The SQL commands that deal with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. It is the component of the SQL statement that controls access to data and to the database. Basically, DCL statements are grouped with DML statements

1. 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 uniqueness of the data. Here are some common SQL key constraints:-

* Primary Key Constraint:-

A primary key is uniquely identifies each record in a table. A table can have only one primary key, which can be composed of one or more columes .

* Foreign Key Constraint:-

A foreign key is a field (or collection of fields) in one table that uniquely identifies a row of another table. It establishes a relationship between the tables.

* Unique Constraint:-

A unique constraint ensures that all the values in a column are unique across the table.

* Not Null Constraint:-

A not null constraint ensures that a column cannot have a NULL value.

* Check Constraint:-

A check constraint ensures that all values in a column satisfy a specific condition.

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

A savepoint in SQL is a point within a transaction that allows you to roll back a portion of the transaction without affecting the entire transaction. Savepoints are useful for complex transactions where you may want to undo certain operations without discarding all the changes made during the transaction.

To create a savepoint in SQL, you use the SAVEPOINT command followed by the name of the SAVEPOINT.

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

A trigger is a special type of stored procedure that automatically runs when an event occurs in the database server.

creating trigger after inserting the record.

DELIMITER $$

create TRIGGER tri\_candidate AFTER INSERT on candidate

for EACH ROW

BEGIN

insert into test(id, name, action\_performed)

VALUES(new.id, new.cname, 'Record inserted');

End

MODULE: 5 (Database) SQL Queries

|  |  |
| --- | --- |
| **Q-1** | Create Table Name : Student and Exam |
|  | CREATE table Student ( roll\_no int PRIMARY key, sname varchar(20), sbranch varchar(20));  C:\Users\ndp87\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2024-06-26 (1).png |

|  |  |
| --- | --- |
| **Q-2** | Create table given below: Employee and IncentiveTable |
|  | C:\Users\ndp87\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2024-06-26 (6).png  **3.** Get First\_Name from employee table using Tom name “Employee Name”  SELECT First\_name AS employee\_name FROM employee WHERE First\_name='tom';    **4.** Get FIRST\_NAME, Joining Date, and Salary from employee table.  Select First\_name,Joining\_data,salary from employee;    **5.** Get all employee details from the employee table order by First\_Name ascending and Salary descending?  SELECT \* FROM employee ORDER by First\_name and Salary DESC;    **6.** Get employee details from employee table whose first name contains ‘J’.  SELECT \* FROM employee WHERE First\_name like 'j%';    **7.** Get department wise maximum salary from employee table order by.  SELECT Dept,COUNT(emp\_id) FROM employee GROUP by Dept HAVING MAX(Salary);    **8.** salaryascending?  SELECT \* from employee ORDER by Salary ;    **9.** Select first\_name, incentive amount from employee and incentives table forthose employees who have incentives and incentive amount greater than3000.  SELECT First\_name,Incentive\_amount from employee,incentivetable WHERE Salary=3000;  **10**.Create After Insert trigger on Employee table which insert records in viewtable  DELIMITER $$  CREATE TRIGGER after\_inster after INSERT on employee for EACH ROW  BEGIN  INSERT into viewtable(emp\_id,first\_name,last\_name,salary,joining\_date,dept);  VALUES(New.emp\_id,New.first\_name,New.last\_name,New.salary,New.joining\_date,New.dept)  END |

|  |  |
| --- | --- |
| **Q-11** | Create table given below: Salesperson and Customer. |
|  | CREATE table Salesperson  (  Sno int PRIMARY key,  sname varchar(30),  city varchar(30),  comm varchar(30)  );  INSERT into salesperson VALUES(1001,'Pee;','London',.12);  INSERT into salesperson VALUES(1002,'Serres','San Jose',.13);  INSERT into salesperson VALUES(1004,'Motika','London',.11);  INSERT into salesperson VALUES(1007,'Rafkix','Barcelona',.15);  INSERT into salesperson VALUES(1003,'Axelrod','New York',.1);    CREATE TABLE Customer  (  CNM int PRIMARY key,  Cname varchar(30),  city varchar(30),  Rating int,  Sno int,  FOREIGN key (Sno) REFERENCES salesperson(Sno)    );  INSERT INTO customer VALUES(201,'Hoffman','London',100,1001);  INSERT INTO customer VALUES(202,'Giovance','Roe',200,1003);  INSERT INTO customer VALUES(203,'Liu','San Jose',300,1002);  INSERT INTO customer VALUES(204,'Grass','Barcelona',100,1002);  INSERT INTO customer VALUES(205,'Clemens','London',300,1007);  INSERT INTO customer VALUES(206,'Pereia','Roe',100,1004);    **12**. Retrieve the below data from above table  **13**.All orders for more than $1000.  SELECT  o.OrderID, o. CustomerID, o.OrderAmount, o.OrderDate, c.CName AS CustomerName, c. City AS CustomerCity,  5. SName AS SalespersonName,  FROM  Orders o  JOIN  Customer c ON o. CustomerID = c. CNo  JOIN  Salesperson s ON c.SNo = 5. SNo  WHERE  o.OrderAmount > 1000;    **14.** Names and cities of all salespeople in London with commission above 0.12.  SELECT \* FROM salesperson WHERE city='london' and comm>=.12;    **15**.All salespeople either in Barcelona or in London  SELECT \* FROM `salesperson` WHERE city in ('Barcelona','London');    **16.** All salespeople with commission between 0.10 and 0.12. (Boundary valuesshould be excluded)  SELECT \* FROM salesperson WHERE comm BETWEEN .10 AND .12;    **17**. All customers excluding those with rating <= 100 unless they are located in Rome.  SELECT \* FROM customer WHERE Rating>=100 or city='Rome'  **18.** Write a SQL statement that displays all the information about allsalespeople  CREATE table salesperson  (  sales\_id int,  name varchar(30),  city varchar(30),  Comm float  );  INSERT into salespeople VALUES(5001,'James Hoog','New York',0.15);  INSERT into salespeople VALUES(5002,'Nail Knite','Paris',0.13);  INSERT into salespeople VALUES(5005,'Pit Alex','Londin',0.11);  INSERT into salespeople VALUES(5006,'Mc Lyon','Paris',0.14);  INSERT into salespeople VALUES(5007,'Paul Adam','Rome',0.13);  INSERT into salespeople VALUES(5003,'Lauson Hen','San Jose',0.12);    **19.** From the following table, write a SQL query to find orders that aredelivered by a salesperson with ID. 5001. Return ord\_no, ord\_date, purch\_amt.  CREATE table orders  (  ord\_no int,  purch\_amt float,  ord\_date date,  cust\_id int,  salesman\_id int  );  INSERT INTO orders VALUES(70001,150.5,'2012-10-05',3005,5002);  INSERT INTO orders VALUES(70009,270.65,'2012-09-10',3001,5005);  INSERT INTO orders VALUES(70002,65.26,'2012-10-05',3002,5001);  INSERT INTO orders VALUES(70004,110.5,'2012-08-17',3009,5003);  INSERT INTO orders VALUES(70007,948.5,'2012-09-10',3005,5002);  INSERT INTO orders VALUES(70005,2400.6,'2012-07-27',3007,5001);  INSERT INTO orders VALUES(70008,5760,'2012-09-10',3002,5001);  INSERT INTO orders VALUES(70010,1983.43,'2012-10-10',3004,5006);  INSERT INTO orders VALUES(70003,2480.4,'2012-10-10',3009,5003);  INSERT INTO orders VALUES(70012,250.45,'2012-06-27',3008,5002);  INSERT INTO orders VALUES(70011,75.29,'2012-08-17',3003,5007);  INSERT INTO orders VALUES(70013,3045.6,'2012-04-25',3002,5001);    **Query:**  SELECT ord\_no,ord\_date,purch\_amt from orders WHERE salesman\_id=5001;    **20.** From the following table, write a SQL query to select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro\_id, pro\_name, pro\_price, and pro\_com.  CREATE TABLE item\_mast  (  pro\_id int,  pro\_name varchar(30),  pro\_price float,  pro\_com int  );  INSERT INTO item\_mast VALUES(101,'Mother Board',3200.00,15);  INSERT INTO item\_mast VALUES(102,'Key Board',450.00,16);  INSERT INTO item\_mast VALUES(103,'ZIP Drive',250.00,14);  INSERT INTO item\_mast VALUES(104,'Speaker',550.00,16);  INSERT INTO item\_mast VALUES(105,'Monitor',5000.00,11);  INSERT INTO item\_mast VALUES(106,'DVD drive',900.00,12);  INSERT INTO item\_mast VALUES(107,'CD drive',800.00,12);  INSERT INTO item\_mast VALUES(108,'Printer',2600.00,13);  INSERT INTO item\_mast VALUES(109,'Refill catridge',350.00,13);  INSERT INTO item\_mast VALUES(110,'Mouse',250.00,12);    **Query:**  SELECT pro\_id,pro\_name,pro\_price,pro\_com from item\_mast WHERE pro\_price BETWEEN 200 and 600;    **21.** From the following table, write a SQL query to calculate the averageprice for a manufacturer code of 16. Return avg.  SELECT AVG(pro\_id) as avg\_price\_is from item\_mast WHERE pro\_com=16;    **22.** From the following table, write a SQL query to display the pro\_nameas 'Item Name' and pro\_priceas 'Price in Rs.'  SELECT pro\_name as “Item Name”, pro\_price as “Price in Rs.” from item\_mast;    **23.** From the following table, write a SQL query to find the items whose prices are higher than or equal to $250. Order the result by product price indescending, then product name in ascending. Return pro\_name and pro\_price.  SELECT pro\_name,pro\_price FROM item\_mast WHERE pro\_price>=250 ORDER BY pro\_price DESC,pro\_name ASC;    **24.** From the following table, write a SQL query to calculate average price of the items for each company. Return average price and companycode.  SELECT pro\_com,AVG(pro\_price) as avg\_price\_is FROM item\_mast GROUP by pro\_com; |