MODULE: 5

SE – Database.

❖ Topics Covered Basics of Database

1. What do you understand By Database?

Ans.

Database is a collection of organized or arranged data that can be easily accessed, updated/ modified or controlled. Information within the data is easily placed into rows and columns, or tables.

2. What is Normalization?

Ans.

Normalization is the process of minimizing **redundancy** from a relation or set of relations. Redundancy in relation may cause insertion, deletion, and update anomalies. So, it helps to minimize redundancies in relations.

3. What is Difference between DBMS and RDBMS?

Ans.

DBMS:

- Data stored is in the file format.
- Individual access to data elements.

- Data stored is a small quantity.
- DBMS supports a single user.
- The software and hardware requirements are low.

  Example of dbms :- Oracle, SQL Server.

RDMS:

- Data stored is in table format.
- Multiple data elements are accessible together.
- Data is stored in a large amount.
- RDBMS supports multiple users.
- The software and hardware requirements are higher.

  Example of rdms :- XML, Microsoft Access.

4. What is MF Cod Rule of RDBMS Systems?

Ans.

- **Codd's Rules** are a set of principles defined by Dr. Edgar F. Codd, the father of the relational database model.These rules outline the essential characteristics and requirements for a database management system (DBMS) to be considered a true relational database management system (RDBMS).

- **12 Codd's rule in RDBMS:** The Foundation Rule, Information Rule, Guaranteed Access Rule, Systematic Treatement of Null Values, Active Online Cataloh, Comprehensive Data Sublanguage, view updating,

Relational Level Operation, Physical Data independence, Logical Data independence, integrity independence, Distribution independence,Non-Subversion.

5. What do you understand By Data Redundancy?

Ans.

- 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 a DDL Interpreter?

Ans.

- DDL commands are used to define and manage database structures such as tables and indexes. Common DDL commands include `CREATE`, `ALTER`, `DROP`, and `TRUNCATE`.

- ‘`CREATE`’ TABLE will create a new table in the database.
- ‘`ALTER`’ TABLE will modify the structure of an existing table.
- ‘`DROP`’ TABLE will remove a table from the database.
- ‘`TRUNCATE`’ TABLE will remove all rows from a table, quickly clearing its contents without logging individual row deletions.

7. What is DML Compiler in SQL?

Ans.

- DML commands are used to interact with and modify the data within a database. Common DML commands include : insert,update,delete,call,explaincall
- `INSERT:` Adds new data into the database.
  `UPDATE:` Modifies existing data in the database.
  `DELETE:` Removes data from the database.

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

Ans.

SQL key constraints are rules that you apply to database columns to ensure the integrity and accuracy of the data. These constraints help control what kind of data can be stored in a column and how it relates to other data within the database.

NOT NULL - Ensures that a column cannot have a NULL value.

Example:

- CREATE TABLE Employees (EmployeeID INT PRIMARY KEY, Name VARCHAR (100) NOT NULL, Position VARCHAR (100));

UNIQUE - Ensures that all values in a column are different.

Example:

- CREATE TABLE Teachers (TeacherID INT PRIMARY KEY, Email VARCHAR (100) UNIQUE, Name VARCHAR (100));

PRIMARY KEY - A combination of NOT NULL and UNIQUE. Uniquely identifies each row in a table.

Example:

- CREATE TABLE Students (StudentID INT PRIMARY KEY, Name VARCHAR (100), Age INT);

FOREIGN KEY - Prevents actions that would destroy links between tables.

Example:

- CREATE TABLE Enrollments (EnrollmentID INT PRIMARY KEY, StudentID INT, CourseID INT, FOREIGN KEY (StudentID) REFERENCES Students(StudentID));

CHECK - Ensures that the values in a column satisfies a specific condition .

Example :

- CREATE TABLE Products (ProductID INT PRIMARY KEY, ProductName VARCHAR(100), Price DECIMAL(10, 2), CHECK (Price > 0));

DEFAULT - Sets a default value for a column if no value is specified.

Example :

- CREATE TABLE Orders (OrderID INT PRIMARY KEY, OrderDate DATE DEFAULT GETDATE(), Quantity INT DEFAULT 1);

CREATE INDEX - Used to create and retrieve data from the database very quickly.

Example :

- CREATE INDEX idx_LastName ON Employees(LastName);

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

Ans.

- A savepoint is like a bookmark for your database changes. You can set a savepoint before making some changes. If something goes wrong, you can go back to that savepoint and undo all the changes made after that point. It's a way to protect your database from accidental errors.

- SAVEPOINT first_savepoit;

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

Ans.

- A trigger is a special type of stored procedure that automatically runs when an event occurs in the database server.
- DML triggers run when a user tries to modify data through a data manipulation language (DML) event.
- DML events are INSERT, UPDATE, or DELETE statements on a table or view.

- Create table

1.

```
create table candidate
(
        id int,
        cname varchar (30)
);
```

2.

```
create table test
(
        id int,
        name varchar (30),
        date_time timestamp,
        action_performed text
);
```

3. creating trigger after inserting the record.

```
DELIMITER $$
```

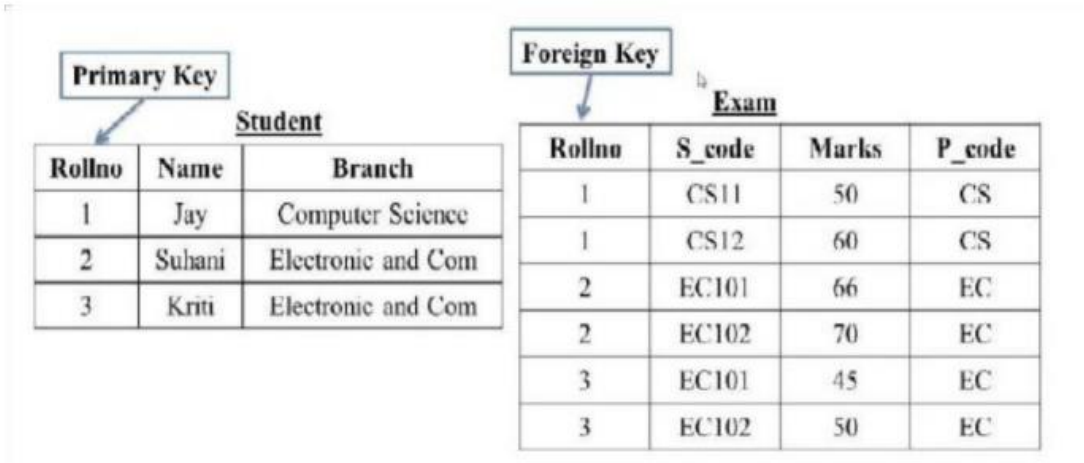
```
create TRIGGER tri_candidate AFTER INSERT on candidate
for EACH ROW

BEGIN
        insert into test (id, name, action_performed)
        VALUES(new.id, new.cname, 'Record inserted');
end
```

4. Now insert records into candidates and see the history table "test".

❖ SQL Queries

1. Create Table Name : Student and Exam



Student

Rollno	Name	Branch
1	Jay	Computer Science
2	Suhani	Electronic and Com
3	Kriti	Electronic and Com

Exam

Rollno	S_code	Marks	P_code
1	CS11	50	CS
1	CS12	60	CS
2	EC101	66	EC
2	EC102	70	EC
3	EC101	45	EC
3	EC102	50	EC

Ans.

- CREATE DATABASE qus1;

- CREATE TABLE Student

  (

  RollNo int PRIMARY KEY,

  Name varchar (20),

  Branch varchar (30)

```
);
```

- INSERT INTO student VALUES (1, 'Jay', 'Computer Science');

  INSERT INTO student VALUES (2, 'Suhani', 'Electronic and Com');

  INSERT INTO student VALUES (3, 'Kriti', 'Electronic and Com');

- CREATE TABLE Exam
  (
  RollNo int,
  S_code varchar (20),
  Marks varchar (20),
  P_code varchar (20),

  FOREIGN KEY(RollNo) REFERENCES student (RollNo)
  );

- INSERT INTO exam VALUES (1,'CS11', 50,'CS');
  INSERT INTO exam VALUES (1,'CS12', 60,'CS');

  INSERT INTO exam VALUES (2,'EC101', 66,'EC');

  INSERT INTO exam VALUES (2,'EC102', 70,'EC');

  INSERT INTO exam VALUES (3,'EC101', 45,'EC');

  INSERT INTO exam VALUES (3,'EC102', 50,'EC');

2. Create table given below: Employee and IncentiveTable

Employee_id	First_name	Last_name	Salary	Joining_date	Department
1	John	Abraham	1000000	01-JAN-13 12.00.00 AM	Banking
2	Michael	Clarke	800000	01-JAN-13 12.00.00 AM	Insurance
3	Roy	Thomas	700000	01-FEB-13 12.00.00 AM	Banking
4	Tom	Jose	600000	01-FEB-13 12.00.00 AM	Insurance
5	Jerry	Pinto	650000	01-FEB-13 12.00.00 AM	Insurance
6	Philip	Mathew	750000	01-JAN-13 12.00.00 AM	Services
7	TestName1	123	650000	01-JAN-13 12.00.00 AM	Services
8	TestName2	Lname%	600000	01-FEB-13 12.00.00 AM	Insurance

Name: Employee

Table Name: Incentive

Employee_ref_id	Incentive_date	Incentive_amount
1	01-FEB-13	5000
2	01-FEB-13	3000
3	01-FEB-13	4000
1	01-JAN-13	4500
2	01-JAN-13	3500

Ans.

- CREATE DATABASE qus2;

- CREATE TABLE Employee (

  Employee_id INT PRIMARY KEY,

  First_name VARCHAR (20),

  Last_name VARCHAR (20),

  Salary INT,

  Joining_date DATETIME,

  Department VARCHAR (20)

);

- INSERT INTO employee VALUES (1, 'John', 'Abraham', 1000000, '2013-01-01 12:00:00 AM', 'Banking');

  INSERT INTO employee VALUES (2, 'Michael', 'clarke', 800000, '2013-01-01 12:00:00 AM', 'Insurance');

  INSERT INTO employee VALUES (3, 'Roy', 'Thomas', 700000, '2013-02-01 12:00:00 AM', 'Banking');

  INSERT INTO employee VALUES (4, 'Tom', 'Jose', 600000, '2013-02-01 12:00:00 AM', 'Insurance');

  INSERT INTO employee VALUES (5, 'Jerry', 'Pinto', 650000, '2013-02-01 12:00:00 AM', 'Insurance');
  INSERT INTO employee VALUES (6, 'Philip', 'Mathew', 750000, '2013-01-01 12:00:00 AM', 'Services');

INSERT INTO employee VALUES (7, 'TestName1', '123', 650000, '2013-01-01 12:00:00 AM', 'Services');

INSERT INTO employee VALUES (8, 'TestName2', 'Lname%', 600000, '2013-02-01 12:00:00 AM', 'Insurance');

- CREATE TABLE Incentive(
  Employee_ref_id INT,
  Incentive_date DATE,
  Incentive_amount INT,

  FOREIGN KEY (Employee_ref_id) REFERENCES employee(Employee_id)

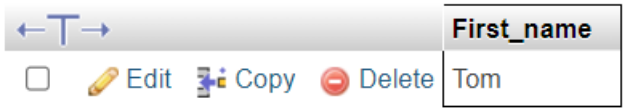
  );

- INSERT INTO incentive VALUES (1, '2013-02-01', 5000);
  INSERT INTO incentive VALUES (2, '2013-02-01', 3000);
  INSERT INTO incentive VALUES (3, '2013-02-01', 4000);
  INSERT INTO incentive VALUES (1, '2013-01-01', 4500);
  INSERT INTO incentive VALUES (2, '2013-01-01', 350);

3. Get First_Name from employee table using Tom name “Employee Name”.

Ans.

- SELECT First_name FROM Employee WHERE First_name = 'Tom';



	First_name
Edit Copy Delete	Tom

4. Get FIRST_NAME, Joining Date, and Salary from employee table.

Ans.

- SELECT First_name,Joining_date,Salary FROM employee;

	First_name	Joining_date	Salary
Edit Copy Delete	John	2013-01-01 12:00:00	1000000
Edit Copy Delete	Michael	2013-01-01 12:00:00	800000
Edit Copy Delete	Roy	2013-02-01 12:00:00	700000
Edit Copy Delete	Tom	2013-02-01 12:00:00	600000
Edit Copy Delete	Jerry	2013-02-01 12:00:00	650000
Edit Copy Delete	Philip	2013-01-01 12:00:00	750000
Edit Copy Delete	TestName1	2013-01-01 12:00:00	650000
Edit Copy Delete	TestName2	2013-02-01 12:00:00	600000

5. Get all employee details from the employee table order by First_Name Ascending and Salary descending?

Ans.

- SELECT * FROM Employee ORDER BY First_name ASC, Salary DESC;

	Employee_id	First_name 1	Last_name	Salary 2	Joining_date	Department
Edit Copy Delete	5	Jerry	Pinto	650000	2013-02-01 12:00:00	Insurance
Edit Copy Delete	1	John	Abraham	1000000	2013-01-01 12:00:00	Banking
Edit Copy Delete	2	Michael	clarke	800000	2013-01-01 12:00:00	Insurance
Edit Copy Delete	6	Philip	Mathew	750000	2013-01-01 12:00:00	Services
Edit Copy Delete	3	Roy	Thomas	700000	2013-02-01 12:00:00	Banking
Edit Copy Delete	7	TestName1	123	650000	2013-01-01 12:00:00	Services
Edit Copy Delete	8	TestName2	Lname%	600000	2013-02-01 12:00:00	Insurance
Edit Copy Delete	4	Tom	Jose	600000	2013-02-01 12:00:00	Insurance

6. Get employee details from employee table whose first name contains ‘J’.

Ans.

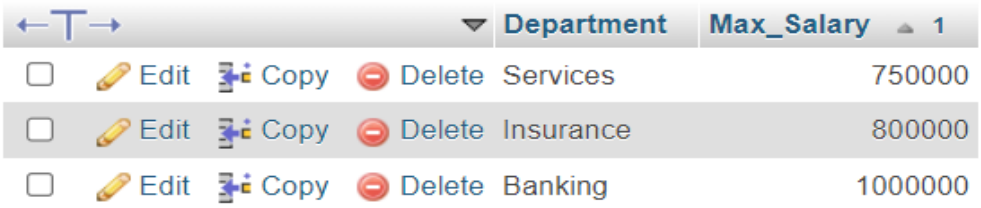
- SELECT * FROM employee WHERE First_name LIKE 'j%';

	Employee_id	First_name	Last_name	Salary	Joining_date	Department
Edit Copy Delete	1	John	Abraham	1000000	2013-01-01 12:00:00	Banking
Edit Copy Delete	5	Jerry	Pinto	650000	2013-02-01 12:00:00	Insurance

7. Get department wise maximum salary from employee table order by salary ascending?

Ans.

- SELECT Department, MAX(Salary) AS Max_Salary FROM employee GROUP BY Department ORDER BY Max_Salary ASC;



	Department	Max_Salary ▲ 1
Edit Copy Delete	Services	750000
Edit Copy Delete	Insurance	800000
Edit Copy Delete	Banking	1000000

9. Select first_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000.

Ans.

- SELECT e.First_name, i.Incentive_amount FROM Employee e JOIN Incentive i ON e.Employee_id = i.Employee_ref_id WHERE i.Incentive_amount > 3000;

First_name	Incentive_amount
John	5000
Roy	4000
John	4500
Michael	3500

10. Create After Insert trigger on Employee table which insert records in viewtable.

Ans.

- CREATE TABLE ViewHistory (

  Employee_id INT,

  First_name VARCHAR (20),

  Last_name VARCHAR (20),

  Salary INT,

  Joining_date DATETIME,

  Department VARCHAR (20),

  date_time timestamp,

action_performed text

);

- DELIMITER $$
  CREATE TRIGGER AfterEmployeeInsert AFTER INSERT ON Employee FOR EACH ROW
  BEGIN

```
INSERT INTO ViewHistory (Employee_id, First_name,
Last_name, Salary, Joining_date,
Department ,action_performed)
VALUES (NEW.Employee_id, NEW.First_name,
NEW.Last_name, NEW.Salary, NEW.Joining_date,
NEW.Department,'Record inserted');
END;
```

- INSERT INTO Employee VALUES (9, 'Alice', 'Williams', 900000, '2024-08-08 09:00:00', 'IT');

Employee_id	First_name	Last_name	Salary	Joining_date	Department	date_time	action_performed
9	Alice	Williams	900000	2024-08-08 09:00:00	IT	2024-08-08 22:30:10	Record inserted

11. Create table given below: Salesperson and Customer

TABLE-1

TABLE NAME- SALSEPERSON

(PK)SNo	SNAME	CITY	COMM
1001	Peel	London	.12
1002	Serres	San Jose	.13
1004	Motika	London	.11
1007	Rafkin	Barcelona	.15
1003	Axelrod	New York	.1

TABLE-2

TABLE NAME- CUSTOMER

(PK)CNM.	CNAME	CITY	RATING	(FK)SNo
201	Hoffman	London	100	1001
202	Giovanne	Roe	200	1003
203	Liu	San Jose	300	1002
204	Grass	Barcelona	100	1002
206	Clemens	London	300	1007
207	Pereira	Roe	100	1004

Ans.

- CREATE DATABASE qus3;
- CREATE TABLE Salesperson (
  SNo INT PRIMARY KEY,
  SName VARCHAR (50),
  City VARCHAR (50),
  Comm DECIMAL(5, 2)
  );

- INSERT INTO Salesperson VALUES (1001, 'Peel', 'London', 0.12);

  INSERT INTO Salesperson VALUES (1002, 'Serres', 'San Jose', 0.13);

  INSERT INTO Salesperson VALUES (1004, 'Motika', 'London', 0.11);

  INSERT INTO Salesperson VALUES (1007, 'Rafkin', 'Barcelona', 0.15);

  INSERT INTO Salesperson VALUES (1003, 'Axelrod', 'New York', 0.10);

- CREATE TABLE Customer (
  CNo INT PRIMARY KEY,
  CName VARCHAR (50),
  City VARCHAR (50),

```
    Rating INT, SNo INT,
    FOREIGN KEY (SNo) REFERENCES Salesperson (SNo)
);
```

- INSERT INTO Customer VALUES(201, 'Hoffman', 'London', 100,1001);

  INSERT INTO Customer VALUES(202, 'Giovanne', 'Roe', 200, 1003);

  INSERT INTO Customer VALUES(203, 'Liu', 'San Jose', 300, 1002);

  INSERT INTO Customer VALUES(204, 'Grass', 'Barcelona', 200, 1007);

  INSERT INTO Customer VALUES(206, 'Clemens', 'London', 300, 1001);

  INSERT INTO Customer VALUES(207, 'Pereira', 'Roe', 100, 1004);

13. All orders with a rating greater than 100.

Ans.

- SELECT * FROM customer WHERE rating > 100;

	CNo	CName	City	Rating	SNo
Edit Copy Delete	202	Giovanne	Roe	200	1003
Edit Copy Delete	203	Liu	San Jose	300	1002
Edit Copy Delete	204	Grass	Barcelona	200	1007
Edit Copy Delete	206	Clemens	London	300	1001

14. Names and cities of all salespeople in London with commission above 0.12.

Ans.

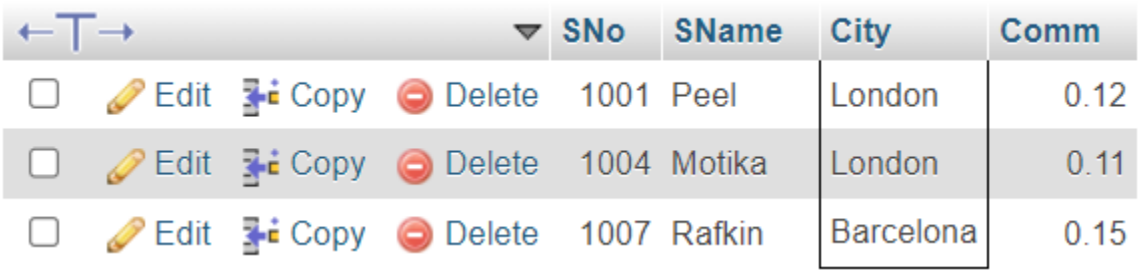
- SELECT sname,City FROM salesperson WHERE City = 'london' AND comm > 0.12;

sname	City

15. All salespeople either in Barcelona or in London.

Ans.

- SELECT * FROM salesperson WHERE City IN ('Barcelona' ,'London');

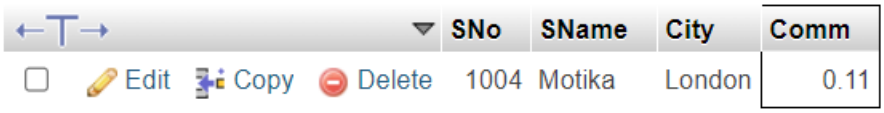


	SNo	SName	City	Comm
Edit Copy Delete	1001	Peel	London	0.12
Edit Copy Delete	1004	Motika	London	0.11
Edit Copy Delete	1007	Rafkin	Barcelona	0.15

16. All salespeople with commission between 0.10 and 0.12. (Boundary valuesshould be excluded).

Ans.

- _SELECT * FROM Salesperson WHERE Comm > 0.10 AND Comm < 0.12;



	SNo	SName	City	Comm
Edit Copy Delete	1004	Motika	London	0.11

17. All customers excluding those with rating <= 100 unless they are located inRome.

Ans.

- SELECT * FROM Customer WHERE (Rating > 100) OR (City = 'Roe' AND Rating <= 100);

	CNo	CName	City	Rating	SNo
Edit Copy Delete	202	Giovanne	Roe	200	1003
Edit Copy Delete	203	Liu	San Jose	300	1002
Edit Copy Delete	204	Grass	Barcelona	200	1007
Edit Copy Delete	206	Clemens	London	300	1001
Edit Copy Delete	207	Pereira	Roe	100	1004

18. Write a SQL statement that displays all the information about all salespeople

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

- CREATE DATABASE qus4;

- CREATE TABLE salesman (
  Salesman_id INT,

Name varchar (50),
City varchar (50),
Commission INT
);

- INSERT INTO salesman VALUES (5001,'James Hoog','New York', 0.15);
  INSERT INTO salesman VALUES (5002,'Nail Knite', 'Paris', 0.13);
  INSERT INTO salesman VALUES (5005,'Pit Alex', 'London', 0.11);
  INSERT INTO salesman VALUES (5006,'Mc Lyon', 'Paris', 0.14);
  INSERT INTO salesman VALUES (5007,'Paul Adam', 'Rome', 0.13);
  INSERT INTO salesman VALUES (5003,'Lauson Hen', 'San Jose', 0.12);

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

19.From the following table, write a SQL query to find orders that are delivered by a salesperson with ID. 5001. Return ord_no, ord_date, purch_amt.

Sample table: orders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Ans.

- CREATE DATABASE qus5;
-
- CREATE TABLE orders(
  ord_no INT,
  purch_amt DECIMAL(10, 2),
  ord_date DATE,
  customer_id INT,
  salesman_id INT

  );

- INSERT INTO orders VALUES (70001, 150.50, '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, 110.50, '2012-08-17',
3009, 5003);
INSERT INTO orders VALUES (70007, 948.50, '2012-09-10',
3005, 5002);
INSERT INTO orders VALUES (70005, 2400.60, '2012-07-27',
3007, 5001);
INSERT INTO orders VALUES (70008, 5760.00, '2012-09-10',
3002, 5001);
INSERT INTO orders VALUES (70010, 1983.43, '2012-10-10',
3004, 5006);
INSERT INTO orders VALUES (70003, 2480.40, '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, 3045.60, '2012-04-25',
3002, 5001);
```

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.50	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.50	2012-08-17	3009	5003
70007	948.50	2012-09-10	3005	5002
70005	2400.60	2012-07-27	3007	5001
70008	5760.00	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.40	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.60	2012-04-25	3002	5001

- SELECT ord_no, ord_date, purch_amt FROM orders WHERE salesman_id = 5001;

ord_no	ord_date	purch_amt
70002	2012-10-05	65.26
70005	2012-07-27	2400.60
70008	2012-09-10	5760.00
70013	2012-04-25	3045.60

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.

Sample table: item_mast

PRO_ID	PRO_NAME	PRO_PRICE	PRO_COM
101	Mother Board	3200.00	15
102	Key Board	450.00	16
103	ZIP drive	250.00	14
104	Speaker	550.00	16
105	Monitor	5000.00	11
106	DVD drive	900.00	12
107	CD drive	800.00	12
108	Printer	2600.00	13
109	Refill cartridge	350.00	13
110	Mouse	250.00	12

Ans.

- CREATE DATABASE qus6;

- CREATE TABLE item_mast(
  pro_id INT PRIMARY KEY,
  pro_name VARCHAR (50),
  pro_price DECIMAL (10, 2),
  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 cartridge', 350.00, 13);
INSERT INTO item_mast VALUES (110, 'Mouse', 250.00, 12);

	pro_id	pro_name	pro_price	pro_com
Edit Copy Delete	101	Mother Board	3200.00	15
Edit Copy Delete	102	Key Board	450.00	16
Edit Copy Delete	103	ZIP drive	250.00	14
Edit Copy Delete	104	Speaker	550.00	16
Edit Copy Delete	105	Monitor	5000.00	11
Edit Copy Delete	106	DVD drive	900.00	12
Edit Copy Delete	107	CD drive	800.00	12
Edit Copy Delete	108	Printer	2600.00	13
Edit Copy Delete	109	Refill cartridge	350.00	13
Edit Copy Delete	110	Mouse	250.00	12

- SELECT pro_id,pro_name,pro_price,pro_com FROM item_mast WHERE PRO_PRICE BETWEEN 200 AND 600;

	pro_id	pro_name	pro_price	pro_com
Edit Copy Delete	102	Key Board	450.00	16
Edit Copy Delete	103	ZIP drive	250.00	14
Edit Copy Delete	104	Speaker	550.00	16
Edit Copy Delete	109	Refill cartridge	350.00	13
Edit Copy Delete	110	Mouse	250.00	12

21.From the following table, write a SQL query to calculate the averageprice for a manufacturer code of 16. Return avg.

Ans.

- SELECT AVG(PRO_PRICE) avg_price FROM item_mast WHERE PRO_COM = 16;

avg_price
500.000000

22.From the following table, write a SQL query to display the pro_nameas 'Item Name' and pro_priceas 'Price in Rs.'

Ans

- SELECT pro_name AS "Item_Name",pro_price AS "Price in RS." FROM item_mast ;

	Item_Name	Price in RS.
Edit Copy Delete	Mother Board	3200.00
Edit Copy Delete	Key Board	450.00
Edit Copy Delete	ZIP drive	250.00
Edit Copy Delete	Speaker	550.00
Edit Copy Delete	Monitor	5000.00
Edit Copy Delete	DVD drive	900.00
Edit Copy Delete	CD drive	800.00
Edit Copy Delete	Printer	2600.00
Edit Copy Delete	Refill cartridge	350.00
Edit Copy Delete	Mouse	250.00

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 in descending, then product name in ascending. Return pro_name and pro_price.

Ans.

- SELECT pro_name , pro_price FROM item_mast WHERE pro_price >= 250 ORDER BY pro_price DESC, pro_name ASC;

	pro_name 2	pro_price 1
Edit Copy Delete	Monitor	5000.00
Edit Copy Delete	Mother Board	3200.00
Edit Copy Delete	Printer	2600.00
Edit Copy Delete	DVD drive	900.00
Edit Copy Delete	CD drive	800.00
Edit Copy Delete	Speaker	550.00
Edit Copy Delete	Key Board	450.00
Edit Copy Delete	Refill cartridge	350.00
Edit Copy Delete	Mouse	250.00
Edit Copy Delete	ZIP drive	250.00

24.From the following table, write a SQL query to calculate average price ofthe items for each company. Return average price and companycode.

Ans.

- SELECT AVG(PRO_PRICE) AS average_price , PRO_COM AS companycode FROM item_mast GROUP BY PRO_COM;

←T→				average_price	companycode
☐	Edit	Copy	Delete	5000.000000	11
☐	Edit	Copy	Delete	650.000000	12
☐	Edit	Copy	Delete	1475.000000	13
☐	Edit	Copy	Delete	250.000000	14
☐	Edit	Copy	Delete	3200.000000	15
☐	Edit	Copy	Delete	500.000000	16