SQL Queries

Que – 1). Create Table Name : Student and Exam

Ans – 1) Create Database =

```
CREATE DATABASE student_db;
```

- create Student table =

```
CREATE TABLE student( Rollno int PRIMARY KEY,
                      Name varchar(20) NOT NULL,
                      Branch varchar(20) NOT NULL );
```

- insert data in student table =

```
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 using foreign key =

```
CREATE TABLE Exam ( Rollno int NOT NULL,
                    S_code varchar(10),
                    Marks int, P_code varchar(3),
                    FOREIGN KEY (Rollno) REFERENCES student(Rollno) );
```

- insert data into Exam table =

```
INSERT INTO Exam (Rollno, S_code, Marks, P_code) VALUES
        (1, 'CS11', 50, 'CS'), (1, 'CS12', 60, 'CS'),
        (2, 'EC101', 66, 'EC'), (2, 'EC102', 70, 'EC'),
        (3, 'EC101', 45, 'EC'), (3, 'EC102', 50, 'EC');
```

Student Table =

←T→				Rollno	Name	Branch
☐	Edit	Copy	Delete	1	Jay	Computer Science
☐	Edit	Copy	Delete	2	Suhani	Electronic and Com
☐	Edit	Copy	Delete	3	Kriti	Electronic and Com

Exam Table =

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

Que – 2). Create table given below: Employee and IncentiveTable

Ans – 2). create database name Employee_db =

```
CREATE DATABASE Employee_db;
```

- create table name Employee =

```
CREATE TABLE employee ( Employee_id int, First_name varchar(20),
                        Last_name varchar(20), Salary int,
                       Joining_date datetime,Department varchar(20));
```

- Insert value =

```
INSERT INTO employee VALUES (1, 'John', 'Abraham', 1000000, '2013-01-13
12.00.00', 'Banking');
```

INSERT INTO employee VALUES (2, 'Michael', 'Clarke', 800000, '2013-01-13 12.00.00', 'Insurance');

INSERT INTO employee VALUES (3, 'Roy', 'Thomas', 700000, '2013-01-13 12.00.00', 'Banking’);

INSERT INTO employee VALUES (4, 'Torm', 'Jose', 600000, '2013-02-13 12.00.00', 'Insurance');

INSERT INTO employee VALUES (5, 'Jerry', 'Pinto', 650000, '2013-02-01 12.00.00', 'Insurance');

INSERT INTO employee VALUES (6, 'Philip', 'Mathew', 750000, '2013-01-01 12.00.00', 'Services');

INSERT INTO employee VALUES (7, 'TestName1', '123', 650000, '2013-01-01 12.00.00', 'Services');

INSERT INTO employee VALUES (8, 'TestName2', 'Lname%', 650000, '2013-02-01 12.00.00', 'Insurance');

Step-4) Create the Incentive Table:

```
CREATE TABLE Incentive (
        Employee_ref_id int,
        Incentive_date date,
        Incentive_amount int,
        FOREIGN KEY(Employee_ref_id) REFERENCE employee(Employee_id)
);
```

Step-5) Insert Data into Incentive table =

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

(1, '2013-01-01', 4500),

(2, '2013-01-01', 3500);

➔ Employee Table

Employee_id	First_name	Last_name	Salary	Joining_date	Department
1	John	Abraham	1000000	2013-01-01 12:00:00	Banking
2	Michael	Clarke	800000	2013-01-13 12:00:00	Insurance
3	Roy	Thomas	700000	2013-02-13 12:00:00	Banking
4	Torn	Jose	600000	2013-02-13 12:00:00	Insurance
5	Jerry	Pinto	650000	2013-02-01 12:00:00	Insurance
6	Philip	Mathew	750000	2013-01-01 12:00:00	Services
7	TestName1	123	650000	2013-01-01 12:00:00	Services
8	TestName2	Lname%	650000	2013-02-01 12:00:00	Insurance

➔ Incentive Table

Employee_ref_id	Incentive_date	Incentive_amount
1	2013-02-01	5000
2	2013-02-01	3000
3	2013-02-01	4000
1	2013-01-01	4500
2	2013-01-01	3500

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

Ans – 3) SELECT First_name FROM employee WHERE First_name LIKE '%Tom%';

First_name
Tom

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

Ans – 4) SELECT First_name, Joining_date, Salary FROM employee;

First_name	Joining_date	Salary
John	2013-01-01 12:00:00	1000000
Michael	2013-01-13 12:00:00	800000
Roy	2013-02-13 12:00:00	700000
Tom	2013-02-13 12:00:00	600000
Jerry	2013-02-01 12:00:00	650000
Philip	2013-01-01 12:00:00	750000
TestName1	2013-01-01 12:00:00	650000
TestName2	2013-02-01 12:00:00	650000

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

Ans – 5) SELECT * FROM employee ORDER by First_name ASC, Salary DESC;

Employee_id	First_name ▲ 1	Last_name	Salary ▼ 2	Joining_date	Department
5	Jerry	Pinto	650000	2013-02-01 12:00:00	Insurance
1	John	Abraham	1000000	2013-01-01 12:00:00	Banking
2	Michael	Clarke	800000	2013-01-13 12:00:00	Insurance
6	Philip	Mathew	750000	2013-01-01 12:00:00	Services
3	Roy	Thomas	700000	2013-02-13 12:00:00	Banking
7	TestName1	123	650000	2013-01-01 12:00:00	Services
8	TestName2	Lname%	650000	2013-02-01 12:00:00	Insurance
4	Tom	Jose	600000	2013-02-13 12:00:00	Insurance

Que - 6). Get employee details from employee table whose first name contains 'J'

Ans – 6)

SELECT * FROM Employee WHERE First_name LIKE '%J%';

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

Que – 7). Get department wise maximum salary from employee table order by

Ans – 7)

SELECT Department , MAX(Salary) AS MaxSalary FROM employee GROUP by Department ORDER by Salary DESC;

Department	Max Salary
Banking	1000000
Insurance	800000
Services	750000

Que – 8). salaryascending?

Ans – 8)

Query -->

```
SELECT * FROM `employee` ORDER by Salary;
```

Output -->

Employee_id	First_name	Last_name	Salary ▲ 1	Joining_date	Department
4	Tom	Jose	600000	2013-02-13 12:00:00	Insurance
5	Jerry	Pinto	650000	2013-02-01 12:00:00	Insurance
7	TestName1	123	650000	2013-01-01 12:00:00	Services
8	TestName2	Lname%	650000	2013-02-01 12:00:00	Insurance
3	Roy	Thomas	700000	2013-02-13 12:00:00	Banking
6	Philip	Mathew	750000	2013-01-01 12:00:00	Services
2	Michael	Clarke	800000	2013-01-13 12:00:00	Insurance
1	John	Abraham	1000000	2013-01-01 12:00:00	Banking

Que – 9). Select first_name, incentive amount from employee and incentives table forthose employees who have incentives and incentive amount greater than 3000

Ans – 9)

```
SELECT e.first_name, i.Incentive_amount
    FROM employee
    INNER JOIN incentive i ON e.employee_id = i.Employee_ref_id
    WHERE i.incentive_amount > 3000;
```

first_name	Incentive_amount
John	4500
John	5000
Michael	3500
Roy	4000

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

---

Ans – 10) Create Database :

```
CREATE DATABASE TRIGGER_db;
```

➔ Create table Employee :

```
CREATE TABLE Employee( id int, Ename varchar(20), Salary int);
```

➔ Create table Viewtable :

```
CREATE TABLE Viewtable( id int, name varchar(20),
                        Salary int,  action_perform text);
```

➔ Create Trigger :

```
DELIMITER $$

CREATE TRIGGER viewtable AFTER INSERT on employee for EACH
ROW

BEGIN

    INSERT INTO viewtable(id, name, Salary, action_perform)

    VALUES(new.id, new.Ename, new.Salary, 'Record Inserted');

END
```

➔ Insert Record in Employee Table :

```
INSERT INTO employee VALUES (1, 'Devarsh', 50000);
INSERT INTO employee VALUES (2, 'Yash', 100000);
INSERT INTO employee VALUES (3, 'Dev', 150000);
```

INSERT INTO employee VALUES (4, 'Ayush', 200000);

➔ Viewtable Data :

Employee Table :

id	Ename	Salary
1	Devarsh	50000
2	Yash	100000
3	Dev	150000
4	Ayush	200000

Viewtable :

id	name	Salary	action_perform
1	Devarsh	50000	Record Inserted
2	Yash	100000	Record Inserted
3	Dev	150000	Record Inserted
4	Ayush	200000	Record Inserted

Que – 11).Create table given below: Salesperson and Customer

Ans – 11) Create Database :

CREATE DATABASE bussiness_db;

→ Create Table Salesperson :

CREATE TABLE Salesperson (

SNo int, SNAME varchar(10), CITY varchar(10),

COMM int, PRIMARY KEY(SNo));

→ Create Table Customer :

CREATE TABLE CUSTOMER (

CNM int, CNAME VARCHAR(15), CITY VARCHAR(15),

RATING INT, SNo int, PRIMARY KEY (CNM),

FOREIGN KEY (SNo) REFERENCES salesperson (SNo));

→ Insert Data in Salesman table :

```
INSERT INTO salesperson VALUES (1001, 'Peel', 'London', 12);
INSERT INTO salesperson VALUES (1002, 'Serres', 'San Jose', 13);
INSERT INTO salesperson VALUES (1004, 'Motika', 'London', 11);
INSERT INTO salesperson VALUES (1007, 'Rafkin', 'Barcelona', 15);
INSERT INTO salesperson VALUES (1003, 'Axelord', 'New York', 1);
```

--> Insert Data into Customer Table :

```
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', 100, 1002);
INSERT INTO customer VALUES (206, 'Clemens', 'London', 300, 1007);
INSERT INTO customer VALUES (207, 'Pereia', 'Roe', 100, 1004);
```

→ Salesperson Table :

	SNo	SNAME	CITY	COMM
Edit Copy Delete	1001	Peel	London	12
Edit Copy Delete	1002	Serres	San Jose	13
Edit Copy Delete	1004	Motika	London	11
Edit Copy Delete	1007	Rafkin	Barcelona	15
Edit Copy Delete	1003	Axelord	New York	1

→ Customer Table :

	CNM	CNAME	CITY	RATING	SNo
Edit Copy Delete	201	Hoffman	London	100	1001
Edit Copy Delete	202	Giovanne	Roe	200	1003
Edit Copy Delete	203	Liu	San Jose	300	1002
Edit Copy Delete	204	Grass	Barcelona	100	1002
Edit Copy Delete	206	Clemens	London	300	1007
Edit Copy Delete	207	Pereia	Roe	100	1004

Que – 13). All orders for more than $100.

Ans – 13)

```
SELECT * FROM `customer` WHERE RATING > 100;
```

	CNM	CNAME	CITY	RATING	SNo
Edit Copy Delete	202	Giovanne	Roe	200	1003
Edit Copy Delete	203	Liu	San Jose	300	1002
Edit Copy Delete	206	Clemens	London	300	1007

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

Ans – 14)

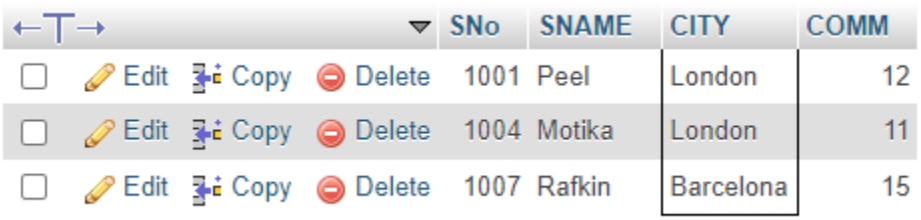
```
SELECT * FROM `salesperson` WHERE CITY = 'London' AND COMM >= 12;
```

	SNo	SNAME	CITY	COMM
Edit Copy Delete	1001	Peel	London	12

Que – 15).All salespeople either in Barcelona or in London.

Ans – 15)

```
SELECT * FROM `salesperson` WHERE CITY='Barcelona' OR CITY='London';
```

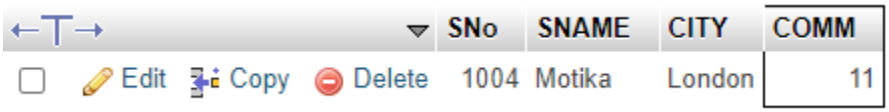


	SNo	SNAME	CITY	COMM
Edit Copy Delete	1001	Peel	London	12
Edit Copy Delete	1004	Motika	London	11
Edit Copy Delete	1007	Rafkin	Barcelona	15

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

Ans – 16)

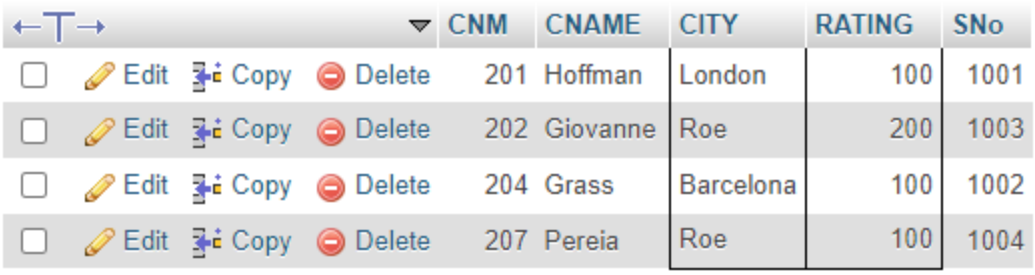
```
SELECT * FROM `salesperson` WHERE COMM > 10 and COMM < 12;
```



	SNo	SNAME	CITY	COMM
Edit Copy Delete	1004	Motika	London	11

Que – 17) . All customers excluding those with rating <= 100 unless they are located in Rome

Ans – 17) SELECT * FROM `customer` WHERE CITY = 'Roe' or RATING <=100;



	CNM	CNAME	CITY	RATING	SNo
Edit Copy Delete	201	Hoffman	London	100	1001
Edit Copy Delete	202	Giovanne	Roe	200	1003
Edit Copy Delete	204	Grass	Barcelona	100	1002
Edit Copy Delete	207	Pereia	Roe	100	1004

Que – 18).Write a SQL statement that displays all the information about all

Salespeople

Ans – 18)

```
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 TABLE Salespeople (
        salesman_id int,
        name varchar(20),
        city varchar(20),
        commision int);
INSERT INTO salespeople VALUES
        ( 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);
```

SELECT * FROM `salespeople`;

salesman_id	name	city	commision
5001	James Hoog	New York	0
5002	Nail Knite	Paris	0
5005	Pit Alex	London	0
5006	Mc Lyon	Paris	0
5007	Paul Adam	Rome	0
5003	Lauson Hen	San Jose	0

Que – 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.

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 – 19)

```
CREATE TABLE orders (
    ord_no INT PRIMARY KEY,
    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),
    (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', 3009, 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
Edit Copy Delete	70002	2012-10-05	65.26
Edit Copy Delete	70005	2012-07-27	2400.60
Edit Copy Delete	70008	2012-09-10	5760.00
Edit Copy Delete	70013	2012-04-25	3045.60

Que – 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.

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 – 20)
```
CREATE TABLE Products (
        PRO_ID INT PRIMARY KEY,
        PRO_NAME VARCHAR(50),
        PRO_PRICE DECIMAL(10, 2),
        PRO_COM INT
);
INSERT INTO Products VALUES
        (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);

SELECT * FROM `products` 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

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

Ans – 21)

CREATE TABLE Products (

PRO_ID INT PRIMARY KEY,

PRO_NAME VARCHAR(50),

PRO_PRICE DECIMAL(10, 2),

PRO_COM INT

);

INSERT INTO Products VALUES

(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);

SELECT AVG(PRO_PRICE) AS avg_price FROM product WHERE PRO_COM = 16;

avg_price
500.000000

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

***Sample table*: item_mast**

Ans – 22)SELECT PRO_NAME AS 'Item_Names', PRO_PRICE AS 'Price in RS' FROM product;

	Item_Names	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

Que - 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 – 23)SELECT PRO_NAME , PRO_PRICE FROM product 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

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

Ans – 24) SELECT AVG(PRO_PRICE) AS average_price, PRO_COM AS companycode FROM product GROUP by PRO_COM;

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