MOD-5 SQL Queries

1.Create Table Name: Student and Exam

- ### Create Student Table Queries

```
1  CREATE TABLE Student
2  (
3      RollNo int PRIMARY KEY,
4      Name Varchar(20),
5      Branch varchar(30)
6      );
```

- ### Create Exam Table Queries

```
1  CREATE TABLE Exam
2  (
3      RollNo int,
4      S_code varchar(10),
5      marks int,
6      P_Code varchar(10),
7      FOREIGN KEY (RollNo) REFERENCES student(RollNo)
8      );
```

- ### Student Insert Query

```
1  INSERT into student VALUES(1,'Jay','Computer Scicnce');
2  INSERT into student VALUES(2,'Suhani','Electronic and Com');
3  INSERT into student VALUES(3,'Kriti','Electronic and Com');
```

- ### Exam Insert Query

```
1  INSERT INTO exam VALUES(1,'CS11',50,'CS');
2  INSERT INTO exam VALUES(1,'CS12',60,'CS');
3  INSERT INTO exam VALUES(2,'EC101',66,'EC');
4  INSERT INTO exam VALUES(2,'EC102',70,'EC');
5  INSERT INTO exam VALUES(3,'EC101',45,'EC');
6  INSERT INTO exam VALUES(3,'EC102',50,'EC');
```

- ### Student Table

RollNo	Name	Branch
1	Jay	Computer Scicnce
2	Suhani	Electronic and Com
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

2. Create table given below: Employee and Incentive Table

- Employee Table

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

- Incentive Table

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

- Create Employee Table Queries

```
1  CREATE table Employee
2  (
3     Employee_id int,
4      First_name varchar(20),
5      Last_name varchar(20),
6      Salary int,
7      Joining_date varchar(40),
8      Department varchar(30),
9      PRIMARY KEY(Employee_id)
10      );
```

- Create Incentive Table Queries

```
1  CREATE TABLE Incentive
2  (
3      Employee_id int,
4      Incentive_date varchar(20),
5      Incentive_amount int,
6
7      FOREIGN KEY(Employee_id)REFERENCES employee(Employee_id)
8      );
```

MOD-5 SQL Queries

- ## Employee Insert Query

```
1 INSERT INTO employee VALUES(1,'John','Abraham',1000000,'01-Jan-13 12.00.00 AM','Banking');
2 INSERT INTO employee VALUES(2,'Michael','Clarke',800000,'01-Jan-13 12.00.00 AM','Insurance');
3 INSERT INTO employee VALUES(3,'Roy','Thomas',700000,'01-Feb-13 12.00.00 AM','Banking');
4 INSERT INTO employee VALUES(4,'Tom','Jose',600000,'01-Feb-13 12.00.00 AM','Insurance');
5 INSERT INTO employee VALUES(5,'Jerry','Pinto',650000,'01-Feb-13 12.00.00 AM','Insurance');
6 INSERT INTO employee VALUES(6,'Philip','Mathew',750000,'01-Jan-13 12.00.00 AM','Services');
7 INSERT INTO employee VALUES(7,'TestName1',123,650000,'01-Jan-13 12.00.00 AM','Services');
8 INSERT INTO employee VALUES(8,'TestName2','Lname%',600000,'01-Feb-13 12.00.00 AM','Insurance');
```

- ## Incentive Insert Query

```
1 INSERT INTO incentive VALUES(1,'01-Feb-13',5000);
2 INSERT INTO incentive VALUES(2,'01-Feb-13',3000);
3 INSERT INTO incentive VALUES(3,'01-Feb-13',4000);
4 INSERT INTO incentive VALUES(1,'01-Jan-13',4500);
5 INSERT INTO incentive VALUES(2,'01-Jan-13',3500);
```

- ## Employee Table

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

- Incentive Table

Employee_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

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

Ans:

```
1  SELECT First_name FROM employee  WHERE First_name='Tom';
```

First_name
Tom

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

Ans:

```
1  SELECT First_name,Joining_date,Salary FROM employee;
```

First_name	Joining_date	Salary
John	01-Jan-13 12.00.00 AM	1000000
Michael	01-Jan-13 12.00.00 AM	800000
Roy	01-Feb-13 12.00.00 AM	700000
Tom	01-Feb-13 12.00.00 AM	600000
Jerry	01-Feb-13 12.00.00 AM	650000
Philip	01-Jan-13 12.00.00 AM	750000
TestName1	01-Jan-13 12.00.00 AM	650000
TestName2	01-Feb-13 12.00.00 AM	600000

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

Ans:

```
1 SELECT * FROM employee ORDER BY First_Name,Salary DESC;
```

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

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

Ans:

```
1 SELECT * FROM `employee` WHERE First_name LIKE'j%';
```

Employee_id	First_name	Last_name	Salary	Joining_date	Department
1	John	Abraham	1000000	01-Jan-13 12.00.00 AM	Banking
5	Jerry	Pinto	650000	01-Feb-13 12.00.00 AM	Insurance

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

Ans:

```
1 SELECT department ,MAX(Salary) as 'maximum salary' FROM employee GROUP BY Department ORDER BY Salary;
```

department	maximum salary
Services	750000
Insurance	800000
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:

```
1  SELECT e.first_name, i.incentive_amount from employee e INNER JOIN incentive i on e.Employee_id=i.Employee_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 view table.

Ans:

- Create View_table Table Queries

```
1  CREATE TABLE view_table
2  (
3      Employee_id INT,
4      First_name VARCHAR(20),
5      Last_name VARCHAR(20),
6      Salary INT,
7      Joining_date VARCHAR(20),
8      Department VARCHAR(20),
9      date_time TIMESTAMP,
10     task TEXT
11     );
```

• Trigger Queries

```
 1  DELIMITER //
 2
 3  CREATE TRIGGER tri_emp
 4  AFTER INSERT ON employee
 5  FOR EACH ROW
 6  BEGIN
 7      INSERT INTO view_table (Employee_id, First_name, Last_name, Salary, Joining_date, Department, task)
 8      VALUES (NEW.Employee_id, NEW.First_name, NEW.Last_name, NEW.Salary, NEW.Joining_date, NEW.department, 'Inserted
    Successfully');
 9  END; //
10
11  DELIMITER ;
```

Employee_id	First_name	Last_name	Salary	Joining_date	Department	date_time	task
10	John	Doe	75000	2012-01-10	banking	2024-10-19 11:52:24	Inserted Successfully

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

- ## Create Salse person Table Queries

```
1  CREATE TABLE salesperson
2  (
3      SNo int,
4      SName Varchar(20),
5      City varchar(20),
6      Comm float,
7      PRIMARY KEY(SNo)
8      );
```

- ## Create Customer Table Queries

```
1  CREATE TABLE Customer
2  (
3      CNm int,
4      CName varchar(20),
5      City varchar(20),
6      Rating INT,
7      SNo int,
8      PRIMARY KEY(CNm),
9      FOREIGN KEY(SNo) REFERENCES salesperson(SNo)
10
11     );
```

- ## Salse person Insert Query

```
1  INSERT INTO salesperson VALUES(1001,'Peel','London',0.12);
2  INSERT INTO salesperson VALUES(1002,'Serres','San Jose',0.13);
3  INSERT INTO salesperson VALUES(1004,'Motika','London',0.11);
4  INSERT INTO salesperson VALUES(1007,'Rafkin','Barcelona',0.15);
5  INSERT INTO salesperson VALUES(1003,'Axelrod','New York',0.1);
```

- ## Customer Insert Query

```
1  INSERT INTO customer VALUES(201,'Hoffman','London',100,1001);
2  INSERT INTO customer VALUES(202,'Giovanne','Roe',200,1003);
3  INSERT INTO customer VALUES(203,'Liu','San Jose',300,1002);
4  INSERT INTO customer VALUES(204,'Grass','Baecelona',100,1002);
5  INSERT INTO customer VALUES(206,'Clemens','London',300,1007);
6  INSERT INTO customer VALUES(207,'Pereira','Roe',100,1004);
```

- Salse person Table

SNo	SName	City	Comm
1001	Peel	London	0.12
1002	Serres	San Jose	0.13
1003	Axelrod	New york	0.1
1004	Motika	London	0.11
1007	Rafkin	Barcelona	0.15

- Customer Table

CNm	CName	City	Rating	SNo
201	Hoffman	London	100	1001
202	Giovanne	Roe	200	1003
203	Liu	San Jose	300	1002
204	Grass	Baecelona	100	1002
206	Clemens	London	300	1007
207	Pereira	Roe	100	1004

13.All Customer name whose rating is more than 100

Ans:

```
1  SELECT * FROM `customer` WHERE Rating>100;
```

CNm	CName	City	Rating	SNo
202	Giovanne	Roe	200	1003
203	Liu	San Jose	300	1002
206	Clemens	London	300	1007

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;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

SName	city

15.All salespeople either in Barcelona or in London

```
1 SELECT SName FROM salesperson WHERE city IN ('Barcelona', 'London');
```

SName
Peel
Motika
Rafkin

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

Ans:

```
1 SELECT SName FROM salesperson WHERE comm>0.10 AND comm<0.12;
```

SName
Peel
Axelrod
Motika

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

Ans:

```
1  SELECT CName FROM customer WHERE (rating>100 OR city='Rome');
```

CName
Giovanne
Liu
Clemens

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

- Create sales people Table Queries

```
1  CREATE TABLE sales_people
2  (
3      salesman_id int,
4      name varchar(20),
5      City varchar(20),
6      Commission float,
7      PRIMARY KEY(salesman_id)
8      );
```

- Insert Sales people Table Queries

```
1  INSERT INTO sales_people VALUES(5001,'James Hoog',New York,0.15);
2  INSERT INTO sales_people VALUES(5002,'Nail Knite',Paris,0.13);
3  INSERT INTO sales_people VALUES(5005,'Pit Alex',London,0.11);
4  INSERT INTO sales_people VALUES(5006,'Mc Lyon',Paris,0.14);
5  INSERT INTO sales_people VALUES(5007,'Paul Adam',Rome,0.13);
6  INSERT INTO sales_people VALUES(5003,'Lauson Hen',San Jose,0.12);
```

- Sales people Table

Ans:

```
1  SELECT * FROM sales_people;
```

Salesman_id	Name	City	Commission
5001	James Hoog	New Yoark	0.15
5002	Nail Knite	Paris	0.13
5003	Lauson Hen	San Jose	0.12
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13

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
```

- Create Orders Table Queries

```
1  CREATE TABLE orders
2  (
3    ord_no int,
4      purch_amt float,
5      ord_date date,
6      customer_id int,
7      salesman_id int,
8      PRIMARY KEY(ord_no),
9      FOREIGN key(Salesman_id) REFERENCES sales_people(Salesman_id)
10     );
```

- Insert Orders Table Queries

```
1   INSERT INTO orders VALUES (70001,150.5,'2012-10-5',3005, 5002);
2   INSERT INTO orders VALUES (70009,270.65,'2012-09-10',3001, 5005);
3   INSERT INTO orders VALUES (70002,65.26,'2012-10-05',3002, 5001);
4   INSERT INTO orders VALUES (70004,110.5,'2012-08-17',3009, 5003);
5   INSERT INTO orders VALUES (70007,948.5,'2012-09-10',3005, 5002);
6   INSERT INTO orders VALUES (70005,2400.6,'2012-07-27',3007, 5001);
7   INSERT INTO orders VALUES (70008,5760,'2012-09-10',3002, 5001);
8   INSERT INTO orders VALUES (70010,1983.43,'2012-10-10',3004, 5006);
9   INSERT INTO orders VALUES (70003,2480.4,'2012-10-10',3009, 5003);
10  INSERT INTO orders VALUES (70012,250.45,'2012-06-27',3008, 5002);
11  INSERT INTO orders VALUES (70011,75.29,'2012-08-17',3003, 5007);
12  INSERT INTO orders VALUES (70013,3045.6,'2012-04-25',3002, 5001);
```

- Orders Table

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

Ans:

```
1 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	2400.6
70008	2012-09-10	5760
70013	2012-04-25	3045.6

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
```

- Create item_mast Table Queries

```
1  CREATE table item_mast
2  (
3        pro_id int,
4      pro_name varchar(30),
5      pro_price int,
6      pro_comm int
7       );
```

- Insert item_mast Table Queries

```
1  INSERT INTO item_mast VALUES(101,'Mother Board',3200.00,15);
2  INSERT INTO item_mast VALUES(102,'Key Board',450.00,16);
3  INSERT INTO item_mast VALUES(103,'Zip driver',250.00,14);
4  INSERT INTO item_mast VALUES(104,'Speaker',550.00,16);
5  INSERT INTO item_mast VALUES(105,'Monitor',5000.00,11);
6  INSERT INTO item_mast VALUES(106,'DVD drive',900.00,12);
7  INSERT INTO item_mast VALUES(107,'CD drive',800.00,12);
8  INSERT INTO item_mast VALUES(108,'Printer',2600.00,13);
9  INSERT INTO item_mast VALUES(109,'Refill cartridge',350.00,13);
10 INSERT INTO item_mast VALUES(110,'Mouse',250.00,12);
```

- item_mast Table

pro_id	pro_name	pro_price	pro_comm
101	Mother Board	3200	15
102	Key Board	450	16
103	Zip driver	250	14
104	Speaker	550	16
105	Monitor	5000	11
106	DVD drive	900	12
107	CD drive	800	12
108	Printer	2600	13
109	Refill cartridge	350	13
110	Mouse	250	12

20 Ans:

```
1  SELECT pro_id, pro_name, pro_price, pro_comm FROM item_mast WHERE pro_price BETWEEN 200 AND 600;
```

pro_id	pro_name	pro_price	pro_comm
102	Key Board	450	16
103	Zip driver	250	14
104	Speaker	550	16
109	Refill cartridge	350	13
110	Mouse	250	12

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

Ans:

```
1 SELECT AVG(pro_price) AS average FROM item_mast WHERE pro_comm=16;
```

average
500

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

Ans:

```
1 SELECT pro_name AS "Item Name", pro_price AS "Price in Rs" FROM item_mast;
```

Item Name	Price in Rs
Mother Board	3200
Key Board	450
Zip driver	250
Speaker	550
Monitor	5000
DVD drive	900
CD drive	800
Printer	2600
Refill cartridge	350
Mouse	250

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:

```
1 SELECT pro_name, pro_price FROM item_mast WHERE pro_price >= 250
2 ORDER BY pro_name,pro_price DESC;
```

pro_name ▲ 1	pro_price ▼ 2
CD drive	800
DVD drive	900
Key Board	450
Monitor	5000
Mother Board	3200
Mouse	250
Printer	2600
Refill cartridge	350
Speaker	550
Zip driver	250

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

Ans:

```
1 SELECT pro_comm AS "Company code", AVG(pro_price) AS "Average Price"
2 FROM item_mast GROUP BY pro_comm;
```

MOD-5 SQL Queries

Company code	Average Price
11	5000
12	650
13	1475
14	250
15	3200
16	500