#### **Module 5 – Introduction To Dbms**

#### Q1: Create Table Name: Student and Exam

```
CREATE TABLE `students` (
  `Rollno` int(11) PRIMARY key NOT NULL AUTO_INCREMENT,
  `Name` varchar(25) DEFAULT NULL,
  `Branch` varchar(30) DEFAULT NULL
)
```

CREATE TABLE marks ( rollno INT, s\_code VARCHAR(10) NOT NULL, marks INT NOT NULL, p\_code VARCHAR(10) NOT NULL, FOREIGN KEY (rollno) REFERENCES students(rollno) );



```
1 SELECT s.rollno,s.name,m.p_code,m.marks,m.s_code
2 FROM students s
3 JOIN marks m
4 ON s.rollno = m.rollno;
```

rollno	name	p_code	marks	s_code
1	jay	P1	85	CS11
1	jay	P2	90	CS12
2	suhani	P1	78	EC101
2	suhani	P2	82	EC102
3	kriti	P1	88	EC101
3	kriti	P2	79	EC102

#### Q2: Create table given below: Employee and IncentiveTable

```
INSERT INTO Employees (F_name, L_name, salary, joining_date, department) VALUES
('John', 'abraham', 10000, '1-jan-13 12.00.00AM', 'banking'),
('michel', 'clark', 75000, '1-jan-13 12.00.00AM', 'insurance'),
('roy', 'thomas', 80000, '1-feb-13 12.00.00AM', 'banking'),
('tom', 'josh', 67000, '1-feb-13 12.00.00AM', 'insurance'),
('jerry', 'pinto', 15000, '1-feb-13 12.00.00AM', 'insurance'),
('philip', 'mathew', 25000, '1-jan-13 12.00.00AM', 'service'),
('testname1', '123', 85000, '1-jan-13 12.00.00AM', 'service'),
('testname2', 'Lname%', 57000, '1-feb-13 12.00.00AM', 'insurance');
```

```
1 INSERT INTO Incentives (employee_ref_id, incentive_date, incentive_amount) VALUES
2 (1, '2013-02-01', 5000.00),
3 (2, '2013-02-01', 3000.00),
4 (3, '2013-02-01', 4000.00),
5 (1, '2013-01-01', 4500.00),
6 (2, '2013-01-01', 3500.00);
```

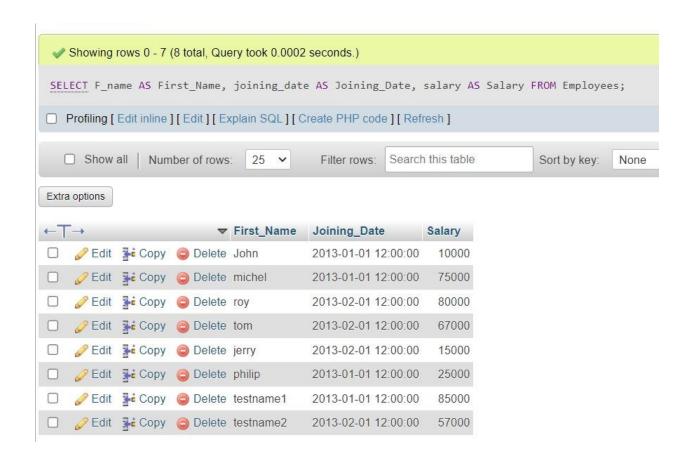
# Q3: Get First\_Name from employee table using Tom name "Employee Name".

```
1 SELECT F_name
2 FROM Employees
3 WHERE F_name = 'tom';
4
```



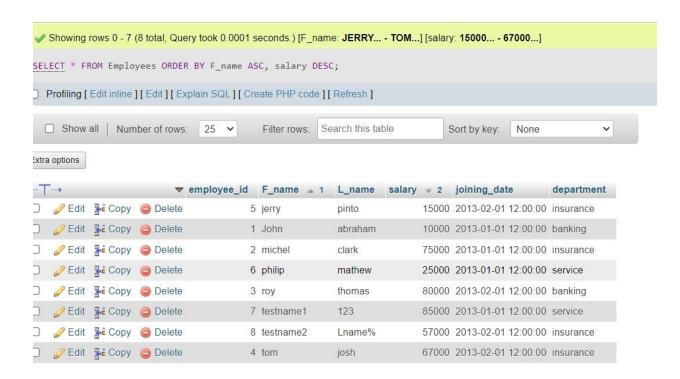
## Q4: Get FIRST\_NAME, Joining Date, and Salary from employee table

```
1 SELECT F_name AS First_Name,
2          joining_date AS Joining_Date,
3          salary AS Salary
4 FROM Employees;
5 |
```

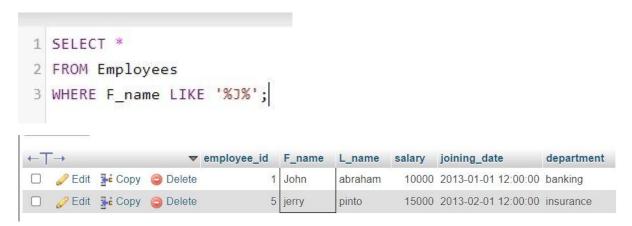


## Q5: Get all employee details from the employee table order by First\_Name Ascending and Salary descending?

```
1 SELECT *
2 FROM Employees
3 ORDER BY F_name ASC, salary DESC;
4
```



## Q6: Get employee details from employee table whose first name contains 'J'.



### Q7,8: Get department wise maximum salary from employee table order by 8. salary ascending?

```
SELECT department, MAX(salary) AS max_salary
FROM Employees
GROUP BY department
ORDER BY max_salary ASC;
```



# Q9: Select First\_Name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000

```
1 SELECT e.F_name AS first_name, i.incentive_amount
2 FROM Employees e
3 JOIN Incentives i ON e.employee_id = i.employee_ref_id
4 WHERE i.incentive_amount > 3000;
```

first_name	incentive_amount
John	5000
roy	4000
John	4500
michel	3500

### Q10: Create After Insert trigger on Employee table which insert records in viewtable

```
1 CREATE TABLE viewtable (
2   employee_id INT,
3   F_name VARCHAR(50),
4   L_name VARCHAR(50),
5   salary DECIMAL(10, 2),
6   joining_date DATE,
7   department VARCHAR(50)
8 );
```

```
DELIMITER //

CREATE TRIGGER after_employee_insertes

AFTER INSERT ON Employees

FOR EACH ROW

BEGIN

INSERT INTO viewtable (employee_id, F_name, L_name, salary, joining_date, department)

VALUES (NEW.employee_id, NEW.F_name, NEW.L_name, NEW.salary, NEW.joining_date, NEW.department);

END;

//

DELIMITER;
```

```
1 INSERT INTO Employees (F_name, L_name, salary, joining_date, department)
2 VALUES ('John', 'Doe', 60000, '2024-01-15', 'Engineering');
3
```

employee_id	F_name	L_name	salary	joining_date	department
12	John	Doe	60000.00	2024-01-15	Engineering
12	John	Doe	60000.00	2024-01-15	Engineering

#### Q11,12: Create table given below: Salesperson and Customer

```
1 CREATE TABLE Salesperson (
2 SNo INT PRIMARY KEY,
3 SName VARCHAR(50),
4 City VARCHAR(50),
5 Comm DECIMAL(4, 2)
6 );
```



```
INSERT INTO Salesperson (SNo, SName, City, Comm) VALUES
(1001, 'Peel', 'London', 0.12),
(1002, 'Serres', 'San Jose', 0.13),
(1004, 'Motika', 'London', 0.11),
(1007, 'Rafkin', 'Barcelona', 0.15),
(1003, 'Axelrod', 'New York', 0.10);
```

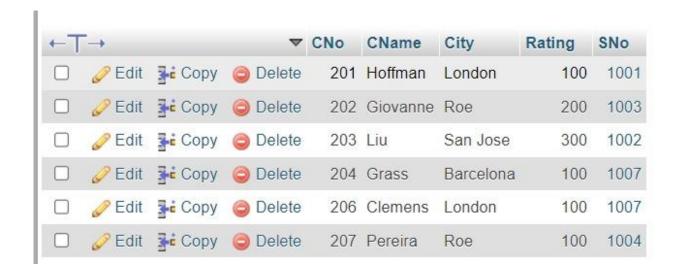
←7	$\rightarrow$		~	SNo	SName	City	Comm
	Edit	<b>3-</b> € Сору	Delete	1001	Peel	London	0.12
	Edit	<b>≩</b> сору	Delete	1002	Serres	San Jose	0.13
	Edit	<b>3-</b> € Сору	Delete	1003	Axelrod	New York	0.10
	Edit	<b>3-</b> € Сору	Delete	1004	Motika	London	0.11
	Edit	<b>3-</b> € Сору	Delete	1007	Rafkin	Barcelona	0.15

```
CREATE TABLE Customer (
CNo INT PRIMARY KEY,
CName VARCHAR(50),
City VARCHAR(50),
Rating INT,
SNo INT,
FOREIGN KEY (SNo) REFERENCES Salesperson(SNo)

);
```

#	Name	Туре	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	SNo 🔑	int(11)			No	None			Change	Drop	More
2	SName	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
3	City	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
4	Comm	decimal(4,2)			Yes	NULL			Change	Drop	More

```
1 INSERT INTO Customer (CNo, CName, City, Rating, SNo) VALUES
2 (201, 'Hoffman', 'London', 100, 1001),
3 (202, 'Giovanne', 'Roe', 200, 1003),
4 (203, 'Liu', 'San Jose', 300, 1002),
5 (204, 'Grass', 'Barcelona', 100, 1007),
6 (206, 'Clemens', 'London', 100, 1007),
7 (207, 'Pereira', 'Roe', 100, 1004);
```



#### Q13: All Customer name whose rating is more than 100.



## Q14: Names and cities of all salespeople in London with commission above 0.12

```
SELECT SName, City
FROM Salesperson
WHERE City = 'London' AND Comm > 0.12;

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

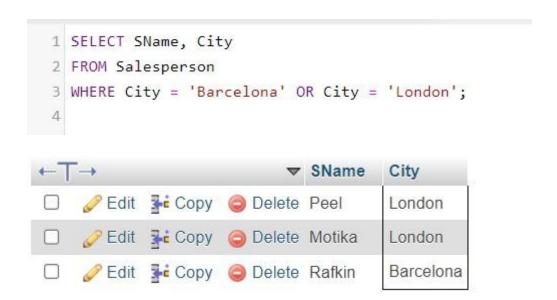
SELECT SName, City FROM Salesperson WHERE City = 'London' AND Comm > 0.12;

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

SName City

Query results operations
```

#### Q15: All salespeople either in Barcelona or in London



## Q16: All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded).



## Q17: All customers excluding those with rating <= 100 unless they are located in Rome

```
SELECT CName, City, Rating
FROM Customer
WHERE Rating > 100 OR City = 'Rome';
```



## Q18: Write a SQL statement that displays all the information about all salespeople

```
1 SELECT *
2 FROM Salesperson;
3
```

<b>←</b> T	<b>−</b> →		~	SNo	SName	City	Comm
	Edit	<b>3</b> - € Copy	Delete	1001	Peel	London	0.12
	Edit	<b>≩</b> copy	Delete	1002	Serres	San Jose	0.13
	Edit	<b>≩</b> сору	Delete	1003	Axelrod	New York	0.10
	Edit	<b>≩</b> copy	Delete	1004	Motika	London	0.11
	@ Edit	<b>≩</b> Copy	Delete	1007	Rafkin	Barcelona	0.15

# Q19: 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

```
CREATE TABLE Salesmen (
salesman_id INT PRIMARY KEY,
name VARCHAR(50),
city VARCHAR(50),
commission DECIMAL(4,2)

);
```



```
8 INSERT INTO Salesmen (salesman_id, name, city, commission) VALUES 9 (5001, 'James Hoog', 'New York', 0.15), 10 (5002, 'Nail Knite', 'Paris', 0.13), 11 (5005, 'Pit Alex', 'London', 0.11), 12 (5006, 'Mc Lyon', 'Paris', 0.14), 13 (5007, 'Paul Adam', 'Rome', 0.13), 14 (5003, 'Lauson Hen', 'San Jose', 0.12);
```

```
▼ salesman_id name
                                                 city
                                                        commission

    Ø Edit 
    ♣ Copy 
    Ø Delete

                                  5001 James Hoog New York
                                                                0.15
5002 Nail Knite
                                                Paris
                                                                0.13
Edit  Copy  Delete
                                  5003 Lauson Hen San Jose
                                                                0.12
☐ Ø Edit ♣ Copy 	 Delete
                                  5005 Pit Alex
                                                London
                                                                0.11
Edit  Copy  Delete
                                  5006 Mc Lyon
                                                Paris
                                                                0.14
Edit  Copy  Delete
                                  5007 Paul Adam
                                                                0.13
                                                 Rome
```

```
1 CREATE TABLE Orders (
2    ord_no INT PRIMARY KEY,
3    purch_amt DECIMAL(10,2),
4    ord_date DATE,
5    customer_id INT,
6    salesman_id INT,
7    FOREIGN KEY (salesman_id) REFERENCES Salesmen(salesman_id)
8 );
```

```
1 INSERT INTO Orders (ord_no, purch_amt, ord_date, customer_id, salesman_id) VALUES
2 (70001, 150.50, '2012-10-05', 3005, 5002),
3 (70009, 270.65, '2012-09-10', 3001, 5005),
4 (70002, 65.26, '2012-10-05', 3002, 5001),
5 (70004, 110.50, '2012-08-17', 3009, 5003),
6 (70007, 948.50, '2012-09-10', 3005, 5002),
7 (70005, 2400.60, '2012-07-27', 3007, 5001),
8 (70008, 5760.00, '2012-09-10', 3002, 5001),
9 (70010, 1983.43, '2012-10-10', 3004, 5006),
10 (70003, 2480.40, '2012-10-10', 3009, 5003),
11 (70012, 250.45, '2012-06-27', 3008, 5002),
12 (70011, 75.29, '2012-08-17', 3003, 5007),
13 (70013, 3045.60, '2012-04-25', 3002, 5001);
```

←T	- <u>→</u>		~	ord_no	purch_amt	ord_date	customer_id	salesman_id
	Edit	<b>≩</b> Copy	Delete	70001	150.50	2012-10-05	3005	5002
	Edit	<b>≩</b> copy	Delete	70002	65.26	2012-10-05	3002	5001
	Edit	<b>≩</b> Copy	Delete	70003	2480.40	2012-10-10	3009	5003
	Ø Edit	<b>≩</b> Copy	Delete	70004	110.50	2012-08-17	3009	5003
	Edit	<b>≩</b> copy	Delete	70005	2400.60	2012-07-27	3007	5001
	Edit	<b>3</b> с Сору	Delete	70007	948.50	2012-09-10	3005	5002
	Ø Edit	<b>≩</b> Copy	Delete	70008	5760.00	2012-09-10	3002	5001
	Ø Edit	<b>3</b> сору	Delete	70009	270.65	2012-09-10	3001	5005
	Edit	<b>≩</b> сору	Delete	70010	1983.43	2012-10-10	3004	5006
	Ø Edit	<b>≩</b> сору	Delete	70011	75.29	2012-08-17	3003	5007
	🧷 Edit	<b>≩</b> сору	Delete	70012	250.45	2012-06-27	3008	5002
		<b>3</b> € Сору	Delete	70013	3045.60	2012-04-25	3002	5001

```
1 SELECT ord_no, ord_date, purch_amt
2 FROM Orders
3 WHERE salesman_id = 5001;
```

←Τ	<b>−</b> →		▽	ord_no	ord_date	purch_amt
	Edit	<b>≩</b> Copy	Delete	70002	2012-10-05	65.26
	Edit	<b>≩</b> сору	Delete	70005	2012-07-27	2400.60
	Edit	<b>≩</b> Copy	Delete	70008	2012-09-10	5760.00
	Edit	<b>≩</b> Copy	Delete	70013	2012-04-25	3045.60

Q20: 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.

```
1 CREATE TABLE item_mast (
2 PRO_ID INT PRIMARY KEY,
3 PRO_NAME VARCHAR(50),
4 PRO_PRICE DECIMAL(10, 2),
5 PRO_COM INT
6 );
```

```
INSERT INTO item_mast (PRO_ID, PRO_NAME, PRO_PRICE, PRO_COM) 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);
)
```

<b>←</b> T	_ →		~	PRO_ID	PRO_NAME	PRO_PRICE	PRO_COM
	Edit	<b>≩</b> € Copy	Delete	101	Mother Board	3200.00	15
	Edit	<b>3</b> € Copy	Delete	102	Key Board	450.00	16
	Edit	<b>≩</b> Copy	Delete	103	ZIP drive	250.00	14
	Edit	<b>3</b> € Copy	Delete	104	Speaker	550.00	16
	Edit	<b>≩</b> Copy	Delete	105	Monitor	5000.00	11
	Edit	<b>≩</b> с Сору	Delete	106	DVD drive	900.00	12
	Edit	<b>≩</b> Copy	Delete	107	CD drive	800.00	12
	Edit	<b>≩-ċ</b> Copy	Delete	108	Printer	2600.00	13
	Edit	<b>3</b> € Copy	Delete	109	Refill cartridge	350.00	13
	Edit	<b>≩-</b> сору	Delete	110	Mouse	250.00	12

```
1 SELECT PRO_ID, PRO_NAME, PRO_PRICE, PRO_COM
2 FROM item_mast
3 WHERE PRO_PRICE BETWEEN 200 AND 600;
```

<b>←</b> T	<b>-</b> →		~	PRO_ID	PRO_NAME	PRO_PRICE	PRO_COM
	Edit	<b>3 ≟</b> Copy	Delete	102	Key Board	450.00	16
	Edit	<b>≩</b> сору	Delete	103	ZIP drive	250.00	14
	Edit	<b>≩</b> Copy	Delete	104	Speaker	550.00	16
	Edit	<b>≩</b> сору	Delete	109	Refill cartridge	350.00	13
	Edit	<b>3</b> - сору	Delete	110	Mouse	250.00	12

## Q21: From the following table, write a SQL query to calculate the average price for a manufacturer code of 16. Return avg.

```
1 SELECT AVG(PRO_PRICE) AS avg_price
2 FROM item_mast
3 WHERE PRO_COM = 16;
4

avg_price
500.000000
```

## Q22: From the following table, write a SQL query to display the pro\_name as 'Item Name' and pro\_priceas 'Price in Rs.'

```
1 SELECT PRO_NAME AS 'Item Name', PRO_PRICE AS 'Price in Rs.'
2 FROM item_mast;
```

←T	<b>→</b>		~	Item Name	Price in Rs.
	Edit	₫ Сору	Delete	Mother Board	3200.00
		<b>З</b> сору	Delete	Key Board	450.00
	Edit	<b>Copy</b>	Delete	ZIP drive	250.00
	Edit	<b>≩</b> € Copy	Delete	Speaker	550.00
	Edit	<b>3</b> € Copy	Delete	Monitor	5000.00
	Edit	<b>≩</b> copy	Delete	DVD drive	900.00
	Edit	<b>≩</b> € Copy	Delete	CD drive	800.00
	Edit	<b>3</b> € Сору	Delete	Printer	2600.00
	Edit	<b>≩</b> сору	Delete	Refill cartridge	350.00
	Edit	<b>З</b> Сору	Delete	Mouse	250.00

Q23: 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.

```
1 SELECT PRO_NAME, PRO_PRICE
2 FROM item_mast
3 WHERE PRO_PRICE >= 250
4 ORDER BY PRO_PRICE DESC, PRO_NAME ASC;
- |
```

←Τ			~	PRO_NAME a 2	PRO_PRICE > 1
	Edit	<b>≩</b> copy	Delete	Monitor	5000.00
	Edit	<b>3</b> - сору	Delete	Mother Board	3200.00
	Edit	<b>≩</b> Copy	Delete	Printer	2600.00
	Edit	<b>3-</b> € Сору	Delete	DVD drive	900.00
	Edit	<b>≩</b> Copy	Delete	CD drive	800.00
	Edit	<b>≩</b> сору	Delete	Speaker	550.00
	Edit	<b>3-</b> € Сору	Delete	Key Board	450.00
	Edit	<b>3-</b> € Сору	Delete	Refill cartridge	350.00
	Edit	<b>3 c</b> Copy	Delete	Mouse	250.00
	Ø Edit	<b>≩</b> сору	Delete	ZIP drive	250.00

# Q24: From the following table, write a SQL query to calculate average price of the items for each company. Return average price and company code.

```
1 SELECT PRO_COM AS company_code, AVG(PRO_PRICE) AS average_price
 2 FROM item mast
 3 GROUP BY PRO COM;
                ▼ company_code average_price
5000.000000
☐ Ø Edit ♣ Copy ⑤ Delete
                             650.000000
1475.000000
                         13
250.000000
3200.000000
☐ Ø Edit ♣ Copy 	 Delete
                             500.000000
```