

# **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?***

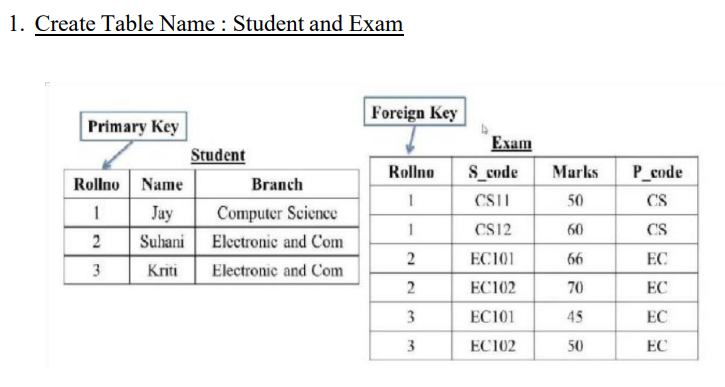
***Ans.***

* 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**



***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');

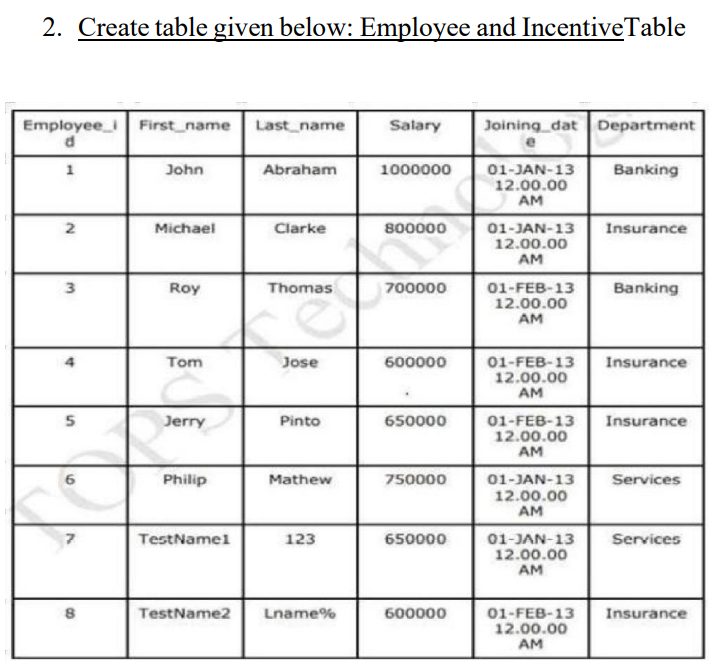
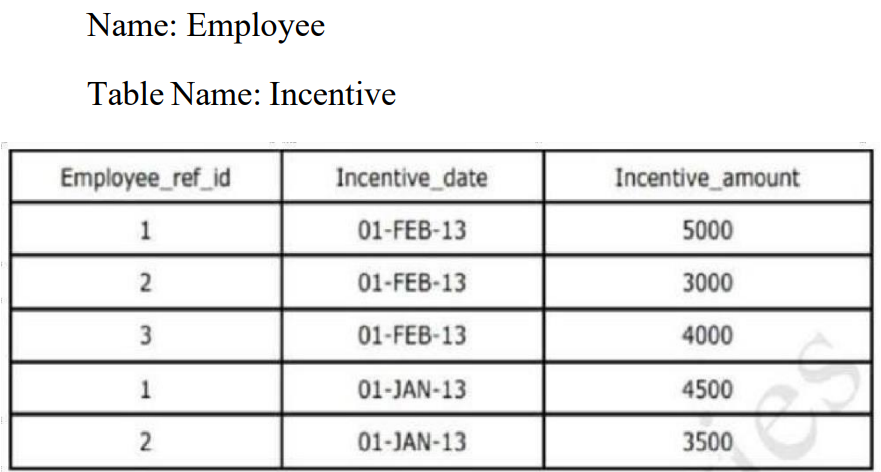
INSERT INTO exam VALUES (1,'CS12', 60,'CS');

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');

***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);

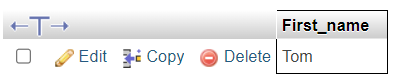
INSERT INTO incentive VALUES (1, '2013-01-01', 4500);

INSERT INTO incentive VALUES (2, '2013-01-01', 350);

***3. Get First\_Name from employee table using Tom name “Employee Name”.***

***Ans.***

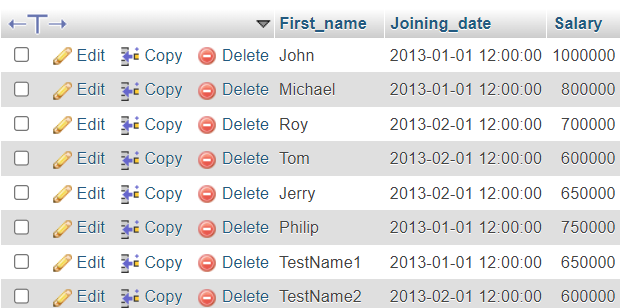
* SELECT First\_name FROM Employee WHERE First\_name = 'Tom';



***4. Get FIRST\_NAME, Joining Date, and Salary from employee table.***

***Ans.***

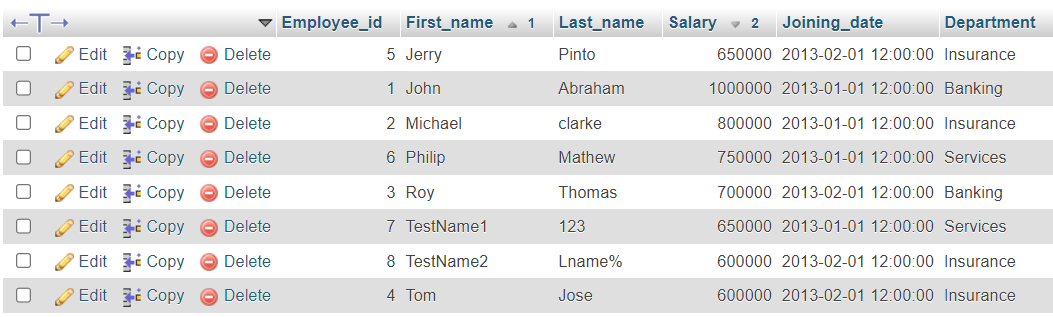
* SELECT First\_name,Joining\_date,Salary FROM employee;



***5. Get all employee details from the employee table order by First\_Name Ascending and Salary descending?***

***Ans.***

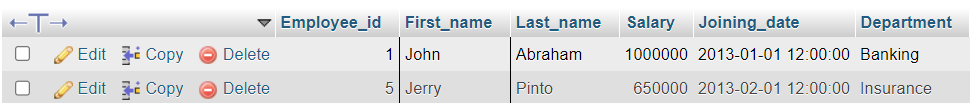
* [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM Employee ORDER BY First\_name ASC, Salary DESC;



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

***Ans.***

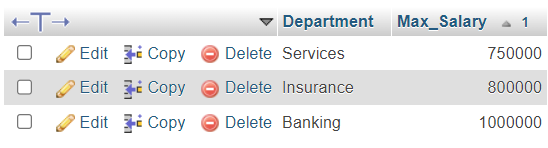
* SELECT \* FROM employee WHERE First\_name LIKE 'j%';



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

***Ans.***

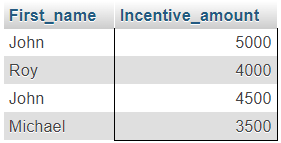
* SELECT Department, MAX(Salary) AS Max\_Salary FROM employee GROUP BY Department ORDER BY Max\_Salary ASC;



***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;



***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

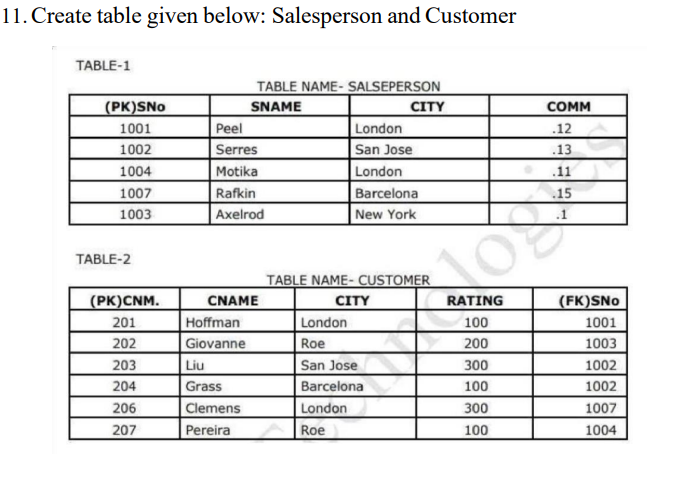
INSERT INTO ViewHistory (Employee\_id, First\_name, Last\_name, Salary, Joining\_date, Department ,action\_performed)

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');





***Ans.***

* [CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-database.html) [DATABASE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-database.html) 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);

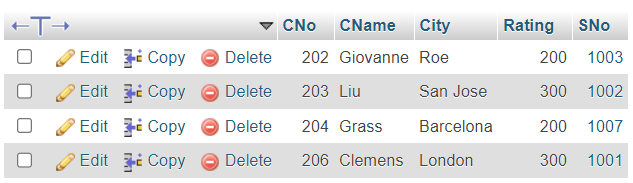
INSERT INTO Customer VALUES(206, 'Clemens', 'London', 300, 1001);

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

***13. All orders with a rating greater than 100.***

***Ans.***

* [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM customer WHERE rating > 100;



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

***Ans.***

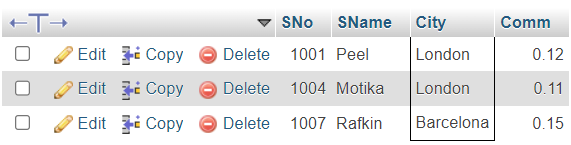
* SELECT sname,City FROM salesperson WHERE City = 'london' AND comm > 0.12;



***15. All salespeople either in Barcelona or in London.***

***Ans.***

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



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

***Ans.***

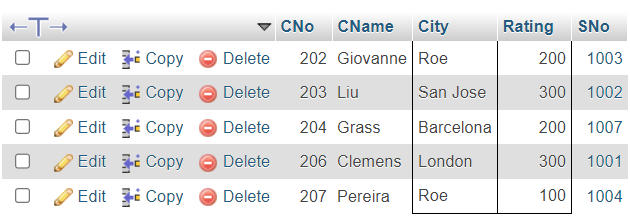
* SELECT \* FROM Salesperson WHERE Comm > 0.10 AND Comm < 0.12;

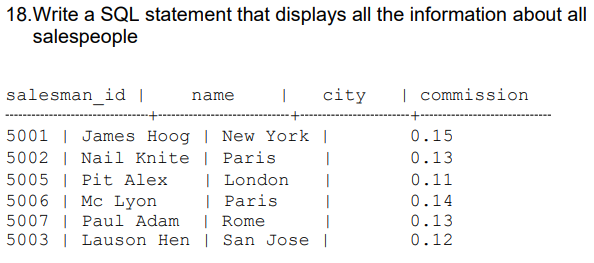


***17. All customers excluding those with rating <= 100 unless they are located inRome.***

***Ans.***

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





* CREATE DATABASE qus4;
* CREATE TABLE [salesman](http://localhost/phpmyadmin/index.php?route=/sql&db=qus4&table=salesman) (

Salesman\_id INT,

Name varchar (50),

City varchar (50),

Commission INT

);

* INSERT INTO salesman VALUES (5001,'James Hoog','New York', 0.15);

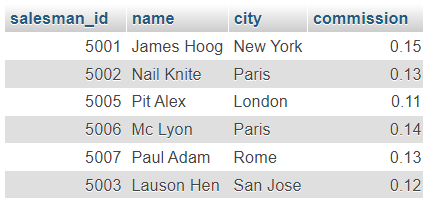
INSERT INTO salesman VALUES (5002,'Nail Knite', 'Paris', 0.13);

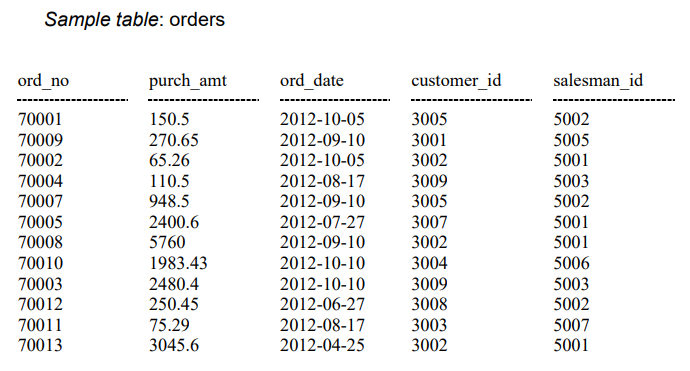
INSERT INTO salesman VALUES (5005,'Pit Alex', 'London', 0.11);

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);



***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.***

***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);

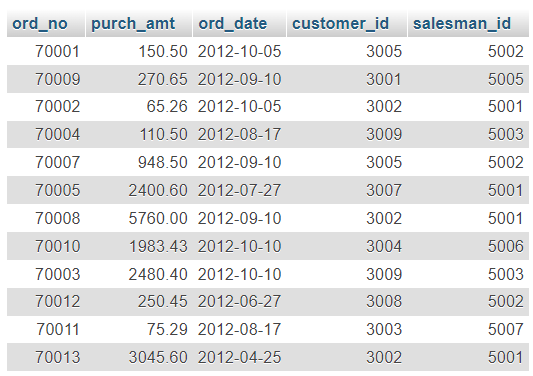
INSERT INTO orders VALUES (70010, 1983.43, '2012-10-10', 3004, 5006);

INSERT INTO orders VALUES (70003, 2480.40, '2012-10-10', 3009, 5003);

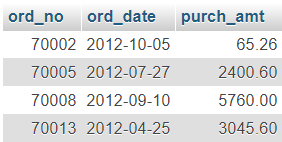
INSERT INTO orders VALUES (70012, 250.45, '2012-06-27', 3008, 5002);

INSERT INTO orders VALUES (70011, 75.29, '2012-08-17', 3003, 5007);

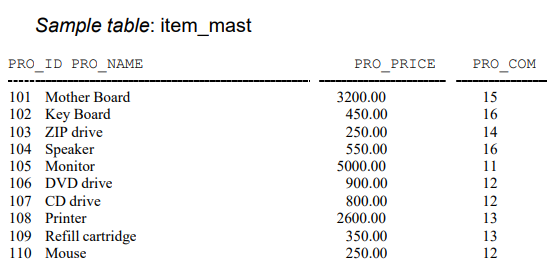
INSERT INTO orders VALUES (70013, 3045.60, '2012-04-25', 3002, 5001);



* 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.***



***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);

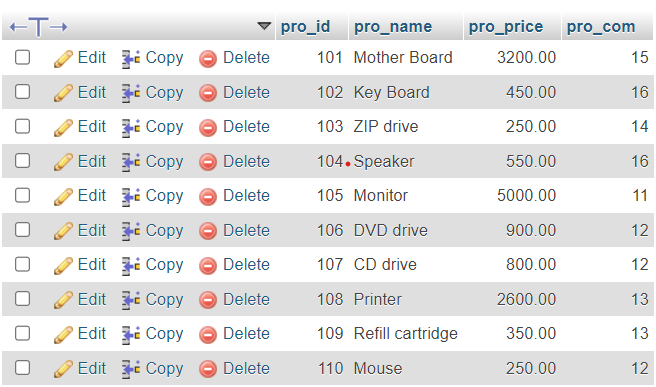
INSERT INTO item\_mast VALUES (106, 'DVD drive', 900.00, 12);

INSERT INTO item\_mast VALUES (107, 'CD drive', 800.00, 12);

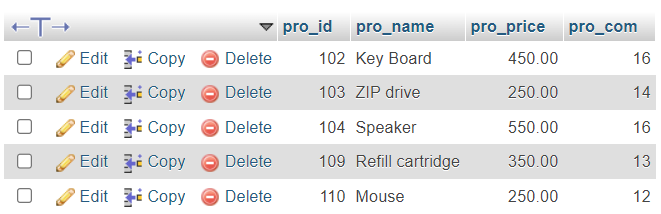
INSERT INTO item\_mast VALUES (108, 'Printer', 2600.00, 13);

INSERT INTO item\_mast VALUES (109, 'Refill cartridge', 350.00, 13);

INSERT INTO item\_mast VALUES (110, 'Mouse', 250.00, 12);



* 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.***

***Ans.***

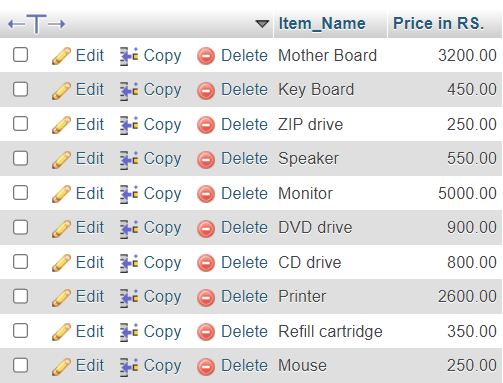
* SELECT AVG(PRO\_PRICE) avg\_price FROM item\_mast WHERE PRO\_COM = 16;



***22.From the following table, write a SQL query to display the pro\_nameas 'Item Name' and pro\_priceas 'Price in Rs.'***

***Ans***

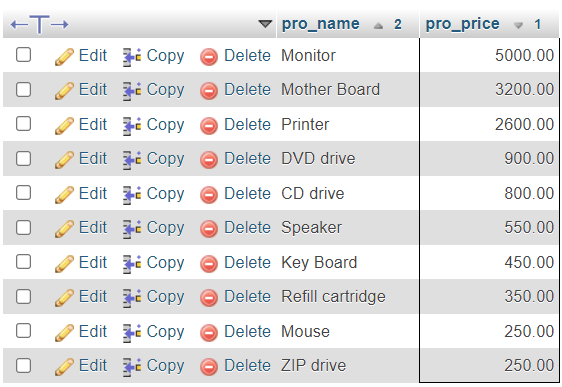
* SELECT pro\_name AS "Item\_Name",pro\_price AS "Price in RS." FROM item\_mast ;



***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 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;



***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;

