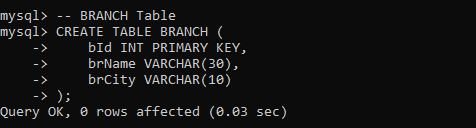
**Shahuraj Bhoite 258077**

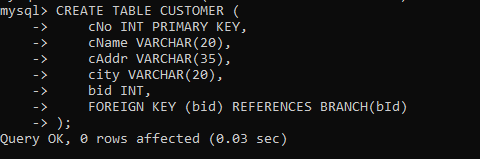
**Lab -4**

### ****Tables****

1. **BRANCH CREATE TABLE BRANCH (bId INT PRIMARY KEY,brName VARCHAR(30),brCity VARCHAR(10));**

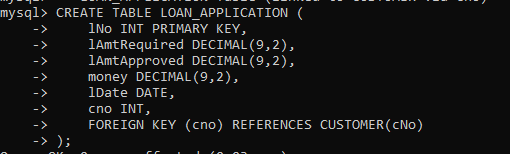
****

1. **CUSTOMER CREATE TABLE CUSTOMER (cNo INT PRIMARY KEY,cName VARCHAR(20),cAddr VARCHAR(35),city VARCHAR(20),bid INT,FOREIGN KEY (bid) REFERENCES BRANCH(bId));**

****

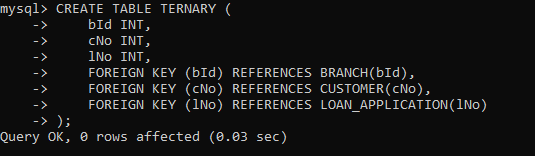
1. **LOAN\_APPLICATION**

**CREATE TABLE LOAN\_APPLICATION (lNo INT PRIMARY KEY,lAmtRequired DECIMAL(9,2),lAmtApproved DECIMAL(9,2),money DECIMAL(9,2),lDate DATE,cno INT,FOREIGN KEY (cno) REFERENCES CUSTOMER(cNo));**

****

1. **Ternary Table**

**CREATE TABLE TERNARY (bId INT, cNo INT,lNo INT,FOREIGN KEY (bId) REFERENCES BRANCH(bId),FOREIGN KEY (cNo) REFERENCES CUSTOMER(cNo),FOREIGN KEY (lNo) REFERENCES LOAN\_APPLICATION(lNo));**

****

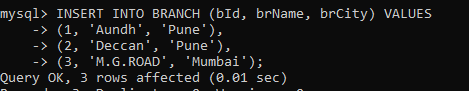
1. **In branch table:**

**INSERT INTO BRANCH (bId, brName, brCity) VALUES**

**(1, 'Aundh', 'Pune'),**

**(2, 'Deccan', 'Pune'),**

**(3, 'M.G.ROAD', 'Mumbai');**

****

1. **In customer table:**

**INSERT INTO CUSTOMER (cNo, cName, cAddr, city, bid) VALUES**

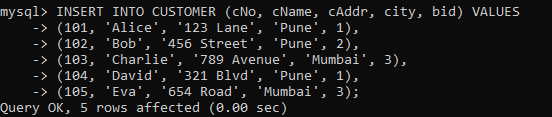
**(101, 'Alice', '123 Lane', 'Pune', 1),**

**(102, 'Bob', '456 Street', 'Pune', 2),**

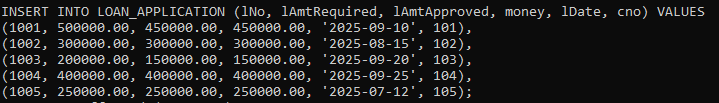
**(103, 'Charlie', '789 Avenue', 'Mumbai', 3),**

**(104, 'David', '321 Blvd', 'Pune', 1),**

**(105, 'Eva', '654 Road', 'Mumbai', 3);**

****

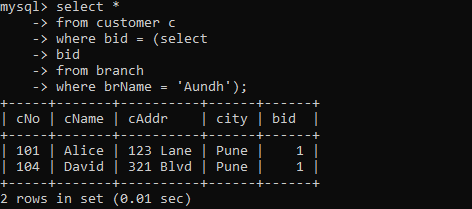
1. **In loan application table:**

****

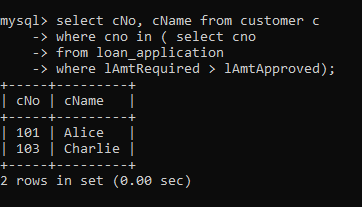
1. **Insert into ternary table:**

****

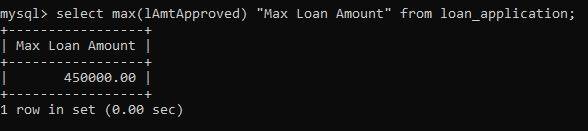
1. **Find the names of the customers for the “Aundh” branch.**

****

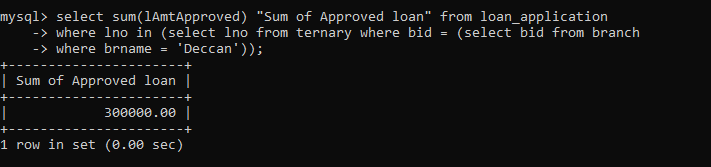
1. **List the names of the customers who have received loan less than their requirement.**

****

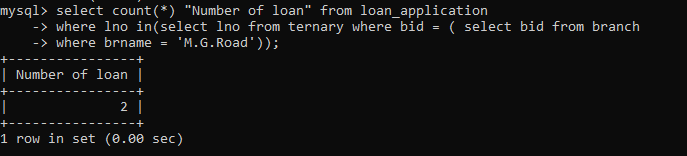
1. **Find the maximum loan amount approved.**

****

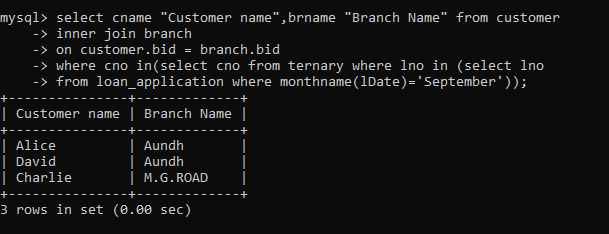
1. **Find out the total loan amount sanctioned by “Deccan” branch.**

****

1. **Count the number of loan applications received by “M.G.ROAD” branch.**

****

1. **List the names of the customer along with the branch names who have applied for loan in the month of September.**

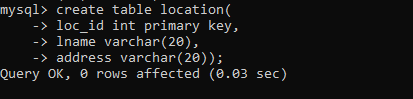
****

**LAB 6**

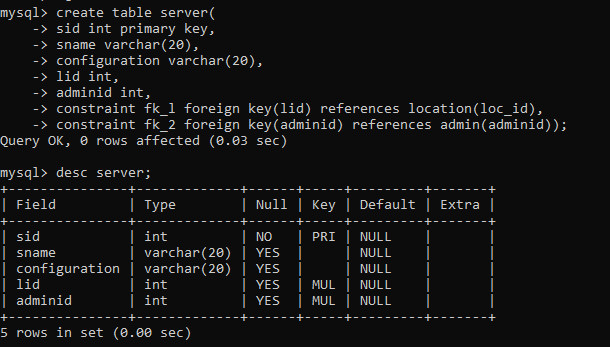
**Assignment -1**

**Create a table**

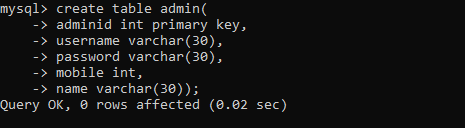
**Location(loc\_id, lname,address)**

****

**Server (sid,sname,configuration,lid, adminid)**

****

**Admin(adminid,username,password,mobile,name)**

****

**INSERT INTO location (loc\_id, lname, address) VALUES**

**(1, 'New York DC', '123 5th Ave'),**

**(2, 'San Francisco DC', '456 Market St'),**

**(3, 'Chicago DC', '789 Lake Shore Dr'),**

**(4, 'Dallas DC', '1011 Elm St');**

**INSERT INTO admin (adminid, username, password, mobile, name) VALUES**

**(1, 'rohit123', 'pass1', 9876543210, 'rohit'),**

**(2, 'namrata456', 'pass2', 9876501234, 'namrata'),**

**(3, 'john789', 'pass3', 9876512345, 'john'),**

**(4, 'alice001', 'pass4', 9876523456, 'alice');**

**INSERT INTO server (sid, sname, configuration, lid, adminid) VALUES**

**(101, 'Server\_A', '8GB RAM, 500GB SSD', 1, 1),**

**(102, 'Server\_B', '16GB RAM, 1TB SSD', 2, 2),**

**(103, 'Server\_C', '32GB RAM, 2TB SSD', 3, 1),**

**(104, 'Server\_D', '64GB RAM, 4TB SSD', NULL, 2 ),**

**(105, 'Server\_E', '16GB RAM, 1TB SSD', 4, NULL);**

**1. Display all server-name, location of the server**

**SELECT s.sname, l.lname**

**FROM server s**

**JOIN location l ON s.lid = l.loc\_id;**

**2. Display all server-name, admin-name, admin mobile**

**SELECT s.sname, a.name, a.mobile**

**FROM server s**

**JOIN admin a ON s.adminid = a.adminid;**

**3. Display all servers which are managed by admin Rohit**

**SELECT s.\***

**FROM server s**

**JOIN admin a ON s.adminid = a.adminid**

**WHERE a.name = 'rohit';**

**4. Display all servers and the locations, for which admin is Rohit**

**SELECT s.sname, l.lname, l.address**

**FROM server s**

**JOIN admin a ON s.adminid = a.adminid**

**JOIN location l ON s.lid = l.loc\_id**

**WHERE a.name = 'rohit';**

**5. Display all admins for whom no server is assigned**

**SELECT a.\***

**FROM admin a**

**LEFT JOIN server s ON a.adminid = s.adminid**

**WHERE s.sid IS NULL;**

**6. Display all servers and admin details, and also display admins for whom no server is assigned**

**SELECT s.sid, s.sname, a.adminid, a.name, a.mobile**

**FROM admin a**

**LEFT JOIN server s ON a.adminid = s.adminid;**

**7. Display servers, admins and location of the server**

**SELECT s.sname, a.name AS admin\_name, l.lname AS location\_name**

**FROM server s**

**JOIN admin a ON s.adminid = a.adminid**

**JOIN location l ON s.lid = l.loc\_id;**

**8. Display all admins for whom no server is assigned, also display locations at which no server is placed**

**SELECT 'Admin' AS type, a.adminid, a.name AS value**

**FROM admin a**

**LEFT JOIN server s ON a.adminid = s.adminid**

**WHERE s.sid IS NULL**

**UNION**

**SELECT 'Location' AS type, l.loc\_id, l.lname AS value**

**FROM location l**

**LEFT JOIN server s ON l.loc\_id = s.lid**

**WHERE s.sid IS NULL;**

**9. Display all servers for which no location is assigned**

**SELECT s.\***

**FROM server s**

**LEFT JOIN location l ON s.lid = l.loc\_id**

**WHERE s.lid IS NULL;**

**10. Display all servers, for which no location is assigned, whose admin is Namrata**

**SELECT s.\***

**FROM server s**

**JOIN admin a ON s.adminid = a.adminid**

**WHERE s.lid IS NULL AND a.name = 'namrata';**

**Lab 6**

**Assignment -2**

**CREATE TABLE Worker (**

**first\_name VARCHAR(50),**

**last\_name VARCHAR(50),**

**department VARCHAR(50),**

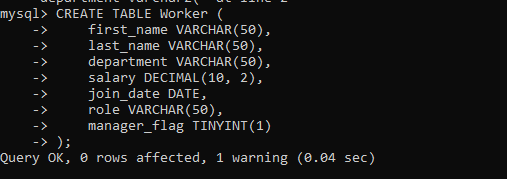
**salary DECIMAL(10, 2),**

**join\_date DATE,**

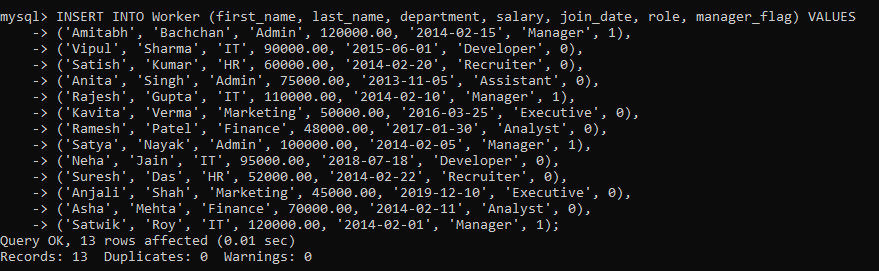
**role VARCHAR(50),**

**manager\_flag TINYINT(1)**

**);**

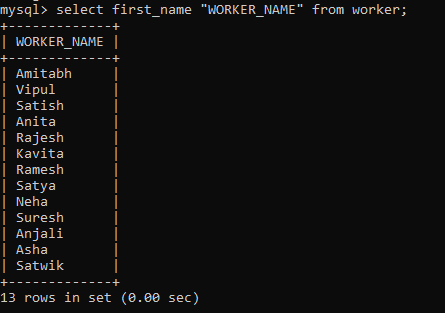
****

**INSERT multiple entries in worker table**

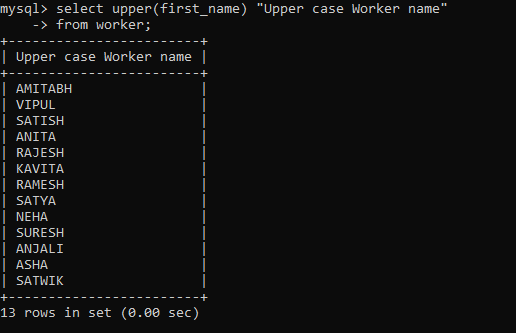
****

1. **Write an SQL query to fetch “FIRST\_NAME” from Worker table using the alias name as . select first\_name "WORKER\_NAME" from worker;**

**select first\_name "WORKER\_NAME" from worker;**

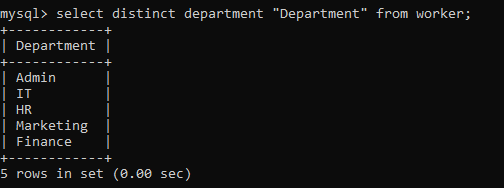
****

**2. Write an SQL query to fetch “FIRST\_NAME” from Worker table in upper case.**

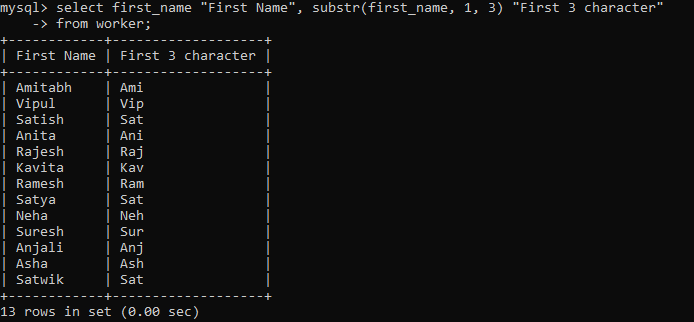
****

**3. Write an SQL query to fetch unique values of DEPARTMENT from Worker table.**

**select distinct department "Department" from worker;**

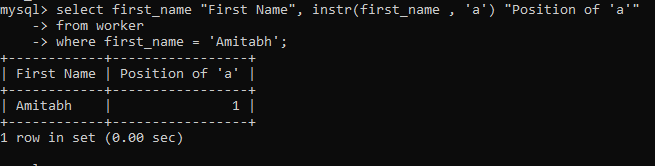
****

**4. Write an SQL query to print the first three characters of FIRST\_NAME from Worker table.**

**select first\_name "First Name", substr(first\_name, 1, 3) "First 3 character" from worker;**

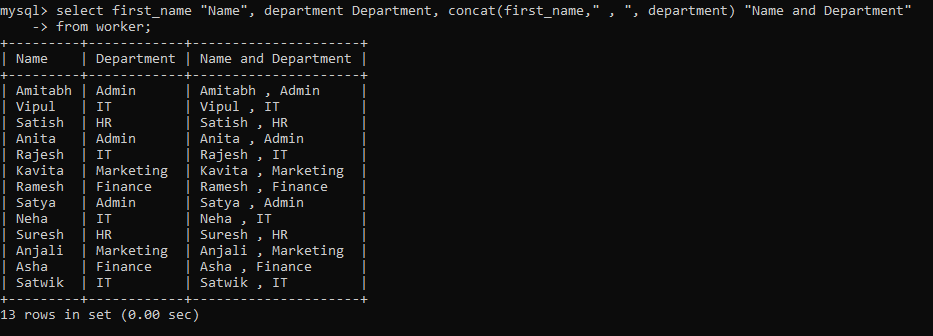
**5. Write an SQL query to find the position of the alphabet (‘a’) in the first name column ‘Amitabh’ from Worker table.**

**select first\_name "First Name", instr(first\_name , 'a') "Position of 'a'" from worker where first\_name = 'Amitabh';**

****

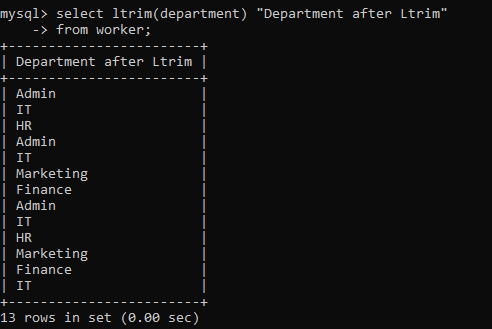
**6. Write an SQL query to print the FIRST\_NAME , departmentname from Worker table separated by white space.**

**select first\_name "Name", department Department, concat(first\_name," , ", department) "Name and Department" from worker;**

****

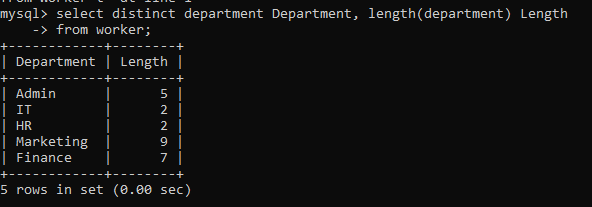
**7. Write an SQL query to print the DEPARTMENT from Worker table after removing white spaces from the left side.**

**select ltrim(department) "Department after Ltrim" from worker;**

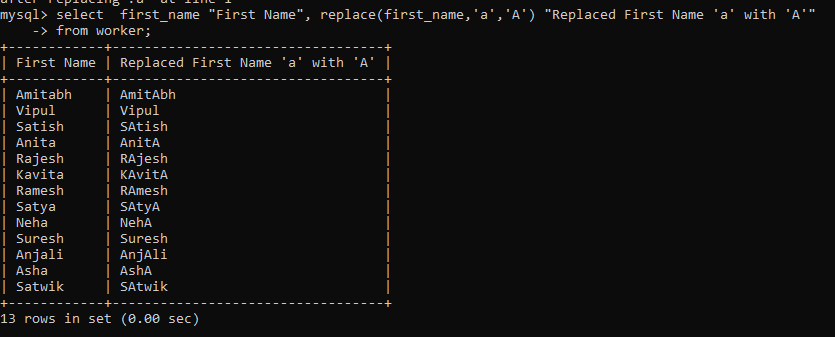
****

**8. Write an SQL query that fetches the unique values of DEPARTMENT from Worker table and prints its length.**

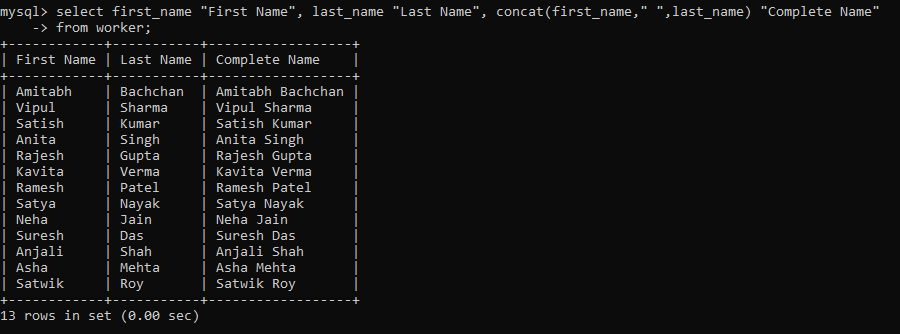
**select distinct department Department, length(department) Length from worker;**

****

**9. Write an SQL query to print the FIRST\_NAME from Worker table after replacing ‘a’ with ‘A’.**

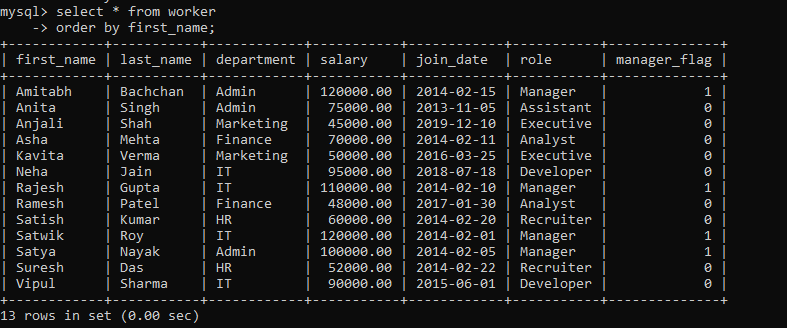
**select first\_name "First Name", replace(first\_name,'a','A') "Replaced First Name 'a' with 'A'" from worker;**

**10. Write an SQL query to print the FIRST\_NAME and LAST\_NAME from Worker table into a single column COMPLETE\_NAME. A space char should separate them**

**select first\_name "First Name", last\_name "Last Name", concat(first\_name," ",last\_name) "Complete Name" from worker;**

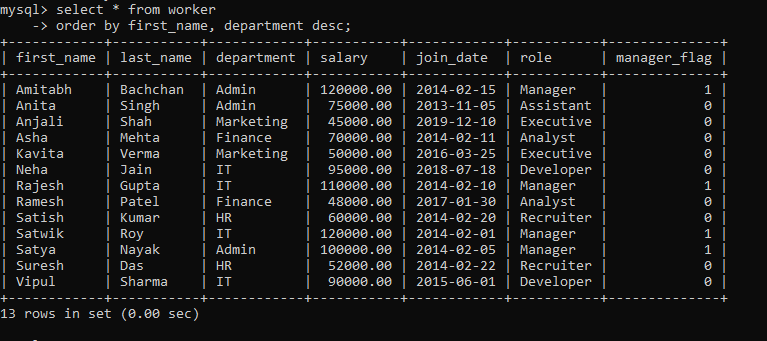
**11. Write an SQL query to print all Worker details from the Worker table order by FIRST\_NAME Ascending.**

**select \* from worker order by first\_name;**

****

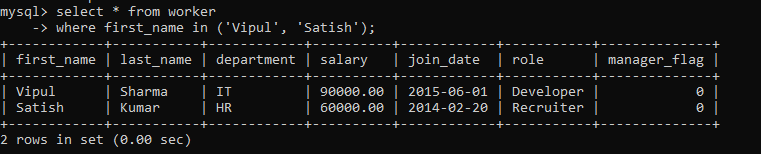
**12. Write an SQL query to print all Worker details from the Worker table order by FIRST\_NAME Ascending and DEPARTMENT Descending.**

**select \* from worker order by first\_name, department desc;**

****

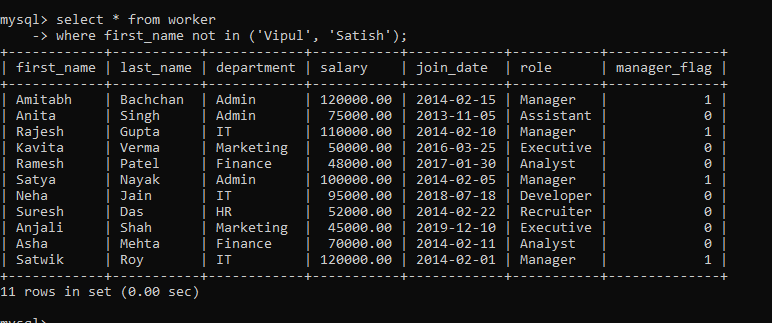
**13. Write an SQL query to print details for Workers with the first name as “Vipul” and “Satish” from Worker table**

**select \* from worker where first\_name in ('Vipul', 'Satish');**

****

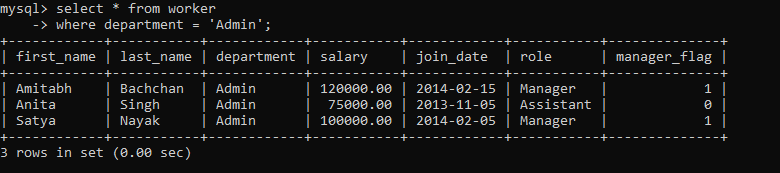
**14. Write an SQL query to print details of workers excluding first names, “Vipul” and “Satish” from Worker table.**

**select \* from worker where first\_name not in ('Vipul', 'Satish');**

****

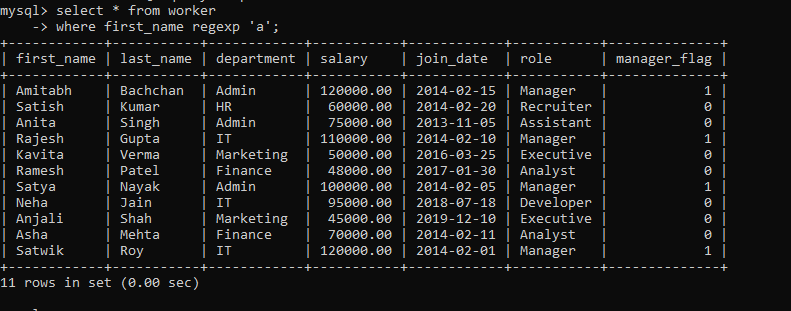
**15. Write an SQL query to print details of Workers with DEPARTMENT name as “Admin”.**

**select \* from worker where department = 'Admin';**

****

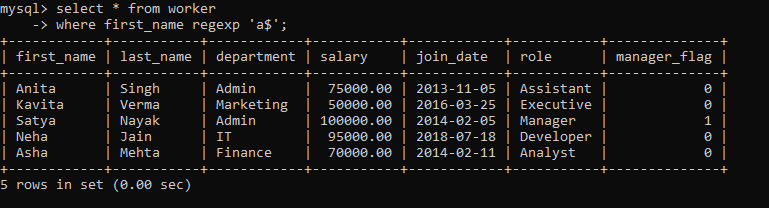
**16. Write an SQL query to print details of the Workers whose FIRST\_NAME contains ‘a’.**

**select \* from worker where first\_name regexp 'a';**

****

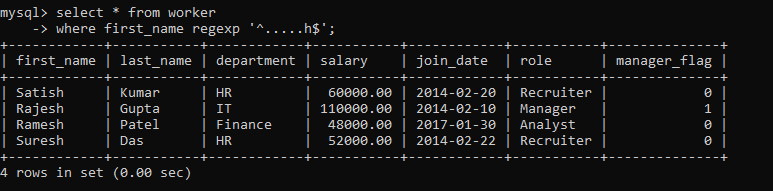
**17. Write an SQL query to print details of the Workers whose FIRST\_NAME ends with ‘a’**

**select \* from worker where first\_name regexp 'a$';**

****

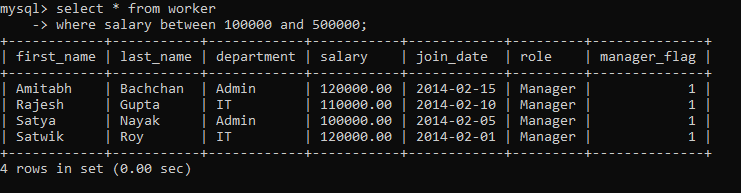
**18. Write an SQL query to print details of the Workers whose FIRST\_NAME ends with ‘h’ and contains six alphabets.**

**select \* from worker where first\_name regexp '^.....h$';**

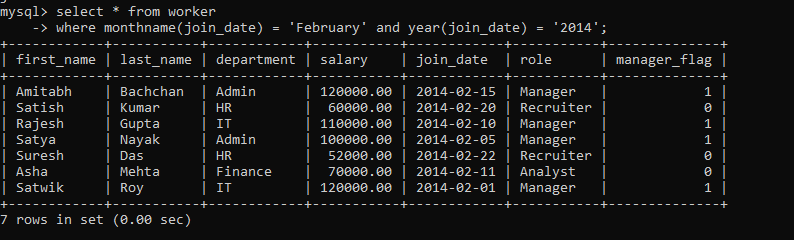
****

**19. Write an SQL query to print details of the Workers whose SALARY lies between 100000 and 500000.**

**select \* from worker where salary between 100000 and 500000;**

****

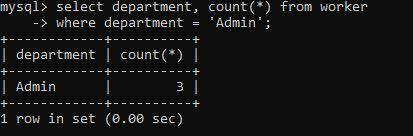
**20. Write an SQL query to print details of the Workers who have joined in Feb’2014.**

**select \* from worker where monthname(join\_date) = 'February' and year(join\_date) = '2014';**

**21. Write an SQL query to fetch the count of employees working in the department ‘Admin’.**

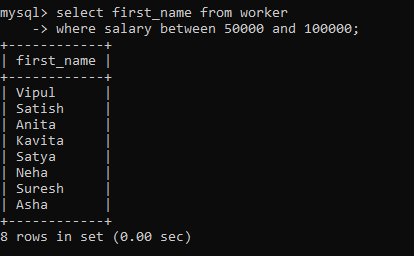
**select department, count(\*) from worker**

**where department = 'Admin';**

****

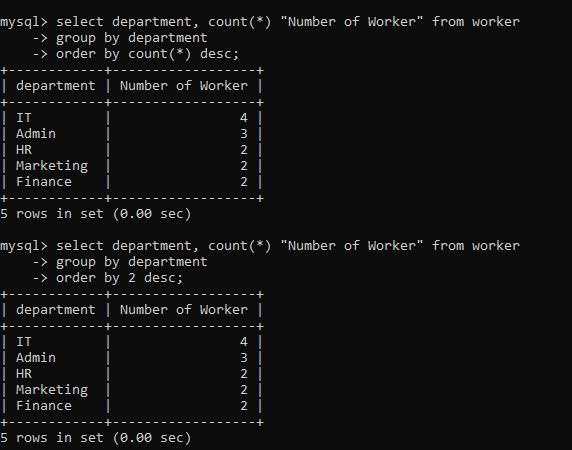
**22. Write an SQL query to fetch worker names with salaries >= 50000 and <= 100000.**

**select first\_name from worker where salary between 50000 and 100000;**

****

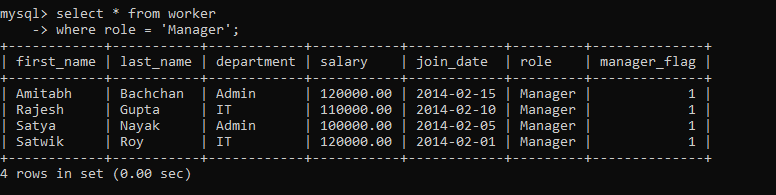
**23. Write an SQL query to fetch the no. of workers for each department in the descending order.**

**select department, count(\*) "Number of Worker" from worker group by department order by 2 desc;**

****

**24. Write an SQL query to print details of the Workers who are also Managers**

**select \* from worker where role = 'Manager';**

****

**25. Write an SQL query to fetch duplicate records having matching data in some fields of a table.**

**select count(\*) "Duplicate", first\_name,department from worker group by first\_name, department having count(\*) > 1;**

****

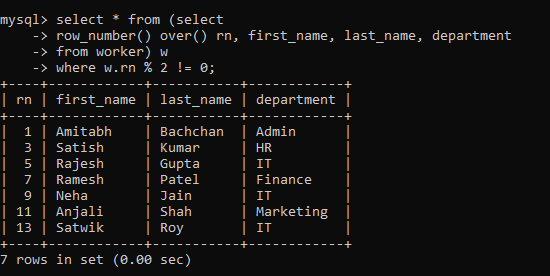
**26. Write an SQL query to show only odd rows from a table**

**select \* from (select**

**row\_number() over() rn, first\_name, last\_name, department**

**from worker) w**

**where w.rn % 2 != 0;**

****

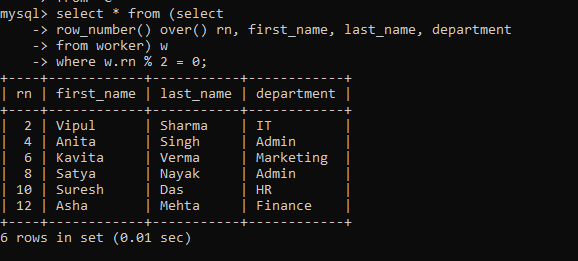
**27. Write an SQL query to show only even rows from a table**

**select \* from (select**

**row\_number() over() rn, first\_name, last\_name, department**

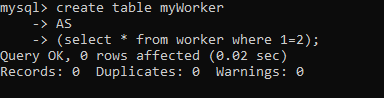
**from worker) w**

**where w.rn % 2 = 0;**

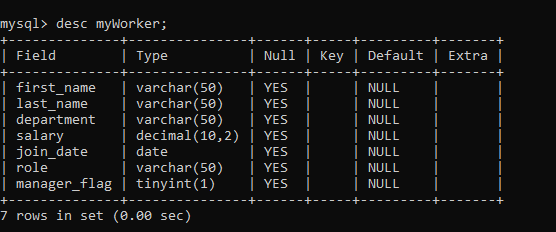
****

**28. Write an SQL query to clone a new table from another table.**

**create table myWorker AS (select \* from worker where 1=2);**

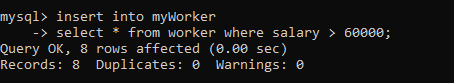
****

**desc myWorker;**

****

**insert into myWorker**

**select \* from worker where salary > 60000;**

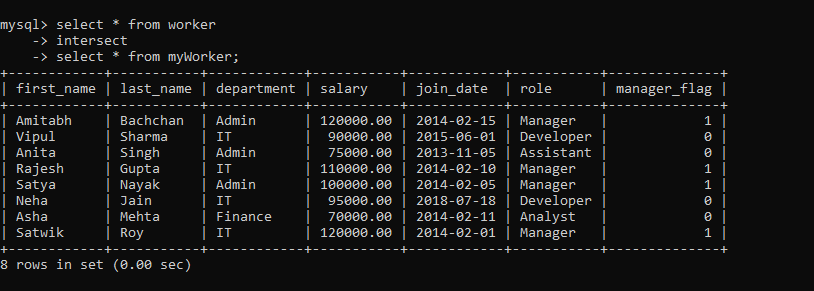
****

**29. Write an SQL query to fetch intersecting records of two tables.**

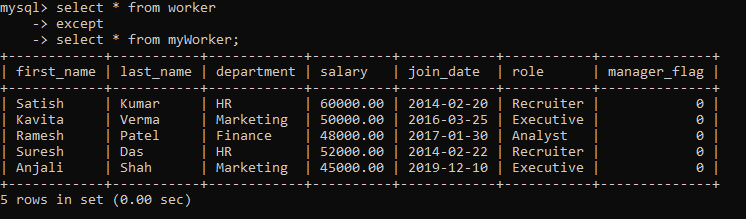
**select \* from worker**

**-> intersect**

**-> select \* from myWorker;**

****

**30. Write an SQL query to show records from one table that another table does not have**

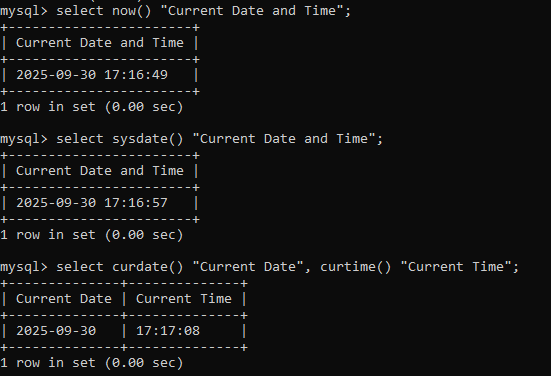
****

**31. Write an SQL query to show the current date and time.**

**select now() "Current Date and Time";**

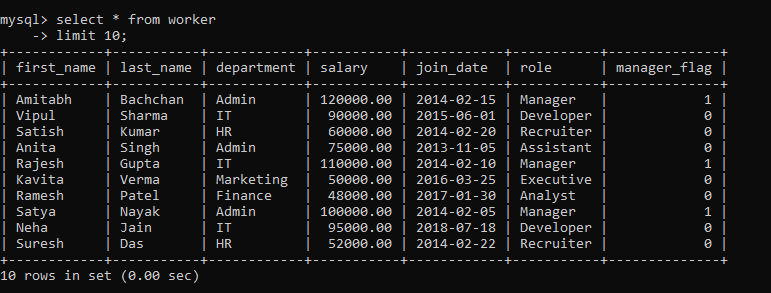
**select sysdate() "Current Date and Time";**

**select curdate() "Current Date", curtime() "Current Time";**

****

**32. Write an SQL query to show the top n (say 10) records of a table.**

**select \* from worker limit 10;**

****

**delimiter //**

**create procedure topRecord(in num int)**

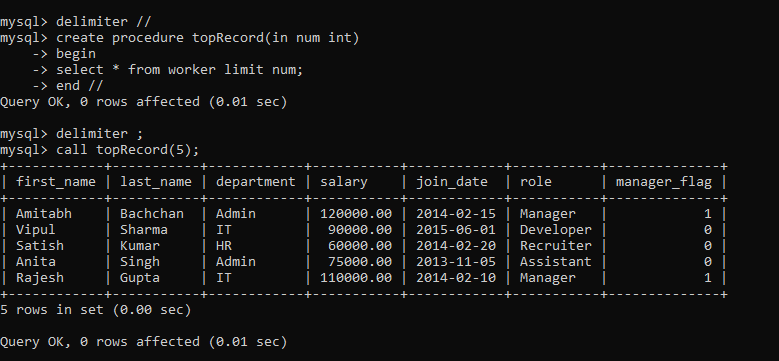
**begin**

**select \* from worker limit num;**

**end //**

**delimiter ;**

**call topRecord(5);**

****

**33. Write an SQL query to determine the nth (say n=5) highest salary from a table.**

**delimiter //**

**create procedure topSalary(in num int)**

**begin**

**select \***

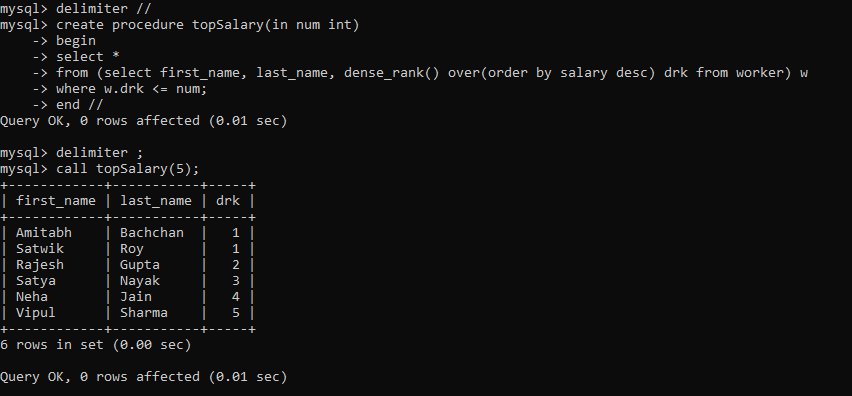
**from (select first\_name, last\_name, dense\_rank() over(order by salary desc) drk from worker) w**

**where w.drk <= num;**

**end //**

**delimiter ;**

**call topSalary(5);**

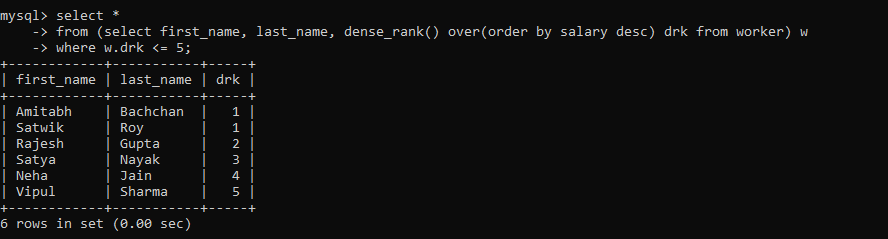
****

**34. Write an SQL query to determine the 5th highest salary without using TOP or limit method.**

**select \***

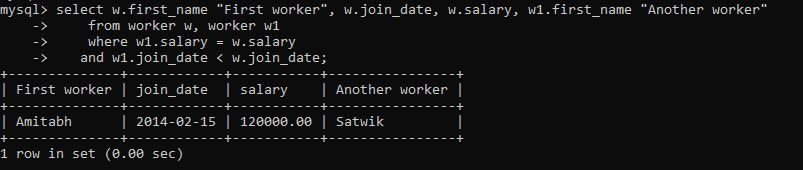
**from (select first\_name, last\_name, dense\_rank() over(order by salary desc) drk from worker) w**

**where w.drk <= 5;**

****

**35. Write an SQL query to fetch the list of employees with the same salary**

**select w.first\_name "First worker", w.join\_date, w.salary, w1.first\_name "Another worker" from worker w, worker w1 where w1.salary = w.salary and w1.join\_date < w.join\_date;**

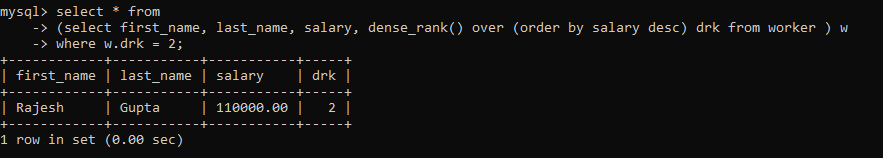
****

**36. Write an SQL query to show the second highest salary from a table.**

**select \* from**

**-> (select first\_name, last\_name, salary, dense\_rank() over (order by salary desc) drk from worker ) w**

**-> where w.drk = 2;**

****

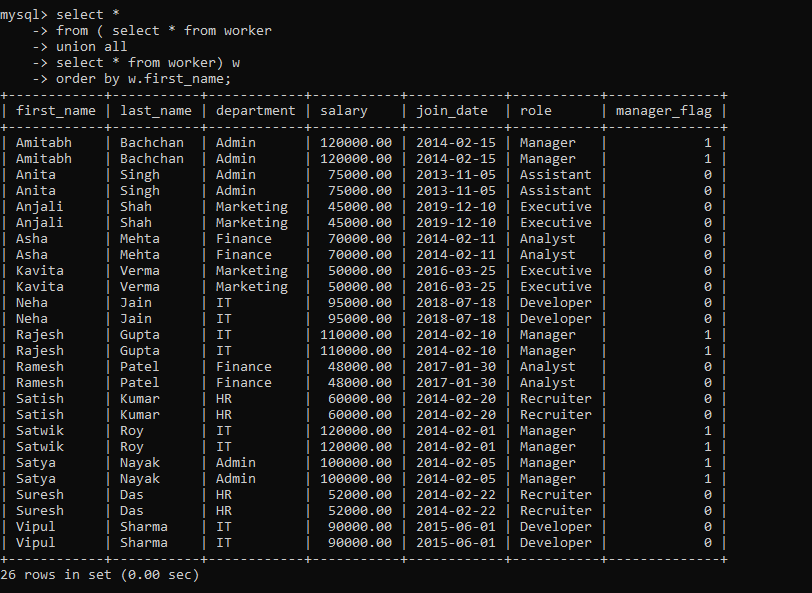
**37. Write an SQL query to show one row twice in results from a table.**

**select \***

**-> from ( select \* from worker**

**-> union all**

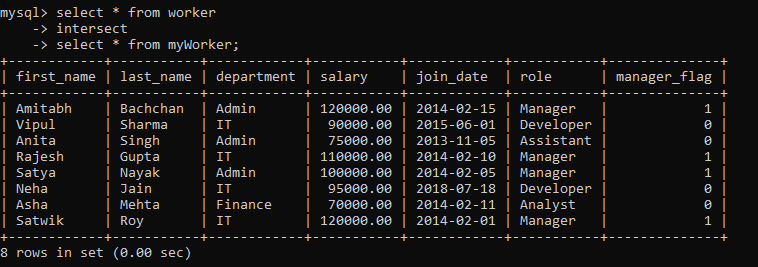
**-> select \* from worker) w**

** -> order by w.first\_name;**

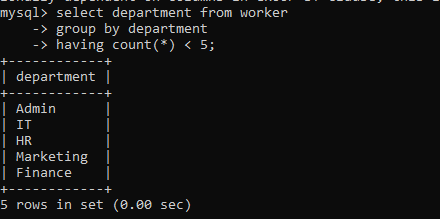
**38. Write an SQL query to fetch intersecting records of two tables.**

**select \* from worker**

**-> intersect**

**-> select \* from myWorker;**

**40. Write an SQL query to fetch the departments that have less than five people in it**

****

**DAY 8**

1. **create all given tables**

**Create the 'vehicle' table**

**CREATE TABLE vehicle (**

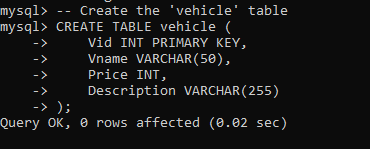
**Vid INT PRIMARY KEY,**

**Vname VARCHAR(50),**

**Price INT,**

**Description VARCHAR(255)**

**);**

****

**Create the 'customer' table**

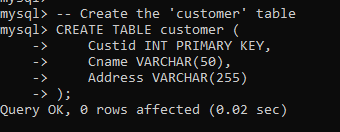
**CREATE TABLE customer (**

**Custid INT PRIMARY KEY,**

**Cname VARCHAR(50),**

**Address VARCHAR(255)**

**);**

****

**Create the 'salesman' table**

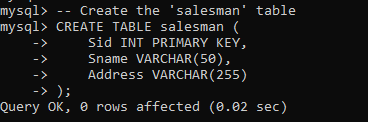
**CREATE TABLE salesman (**

**Sid INT PRIMARY KEY,**

**Sname VARCHAR(50),**

**Address VARCHAR(255)**

**);**

****

**Create the 'cust\_vehicle' table (many-to-many relationship between customers and vehicles)**

**CREATE TABLE cust\_vehicle (**

**Custid INT,**

**Vid INT,**

**Sid INT,**

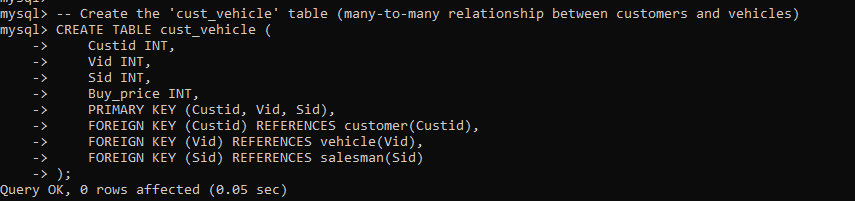
**Buy\_price INT,**

**PRIMARY KEY (Custid, Vid, Sid),**

**FOREIGN KEY (Custid) REFERENCES customer(Custid),**

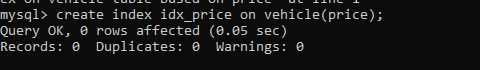
**FOREIGN KEY (Vid) REFERENCES vehicle(Vid),**

**FOREIGN KEY (Sid) REFERENCES salesman(Sid)**

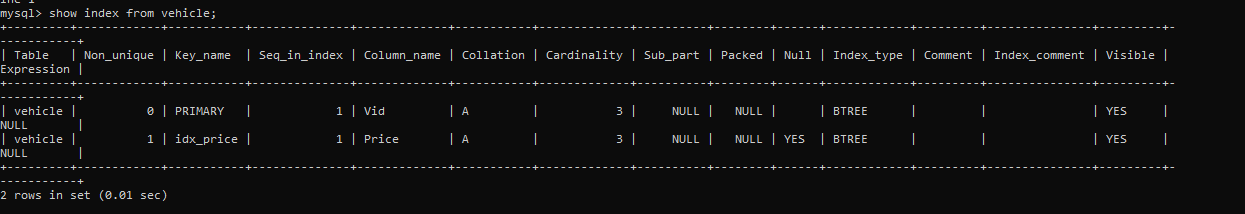
**);**

1. **create index on vehicle table based on price**

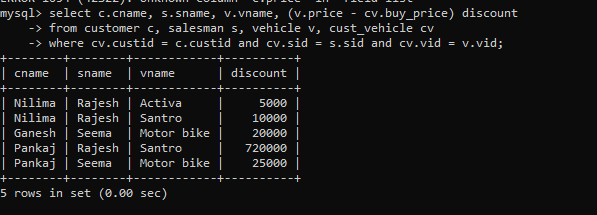
**create index idx\_price on vehicle(price);**

****

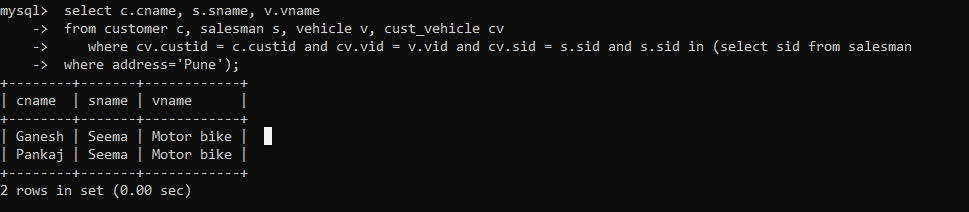
**show index from vehicle;**

****

1. **find all customer name,vehicle name, salesman name, discount earn by all customer**

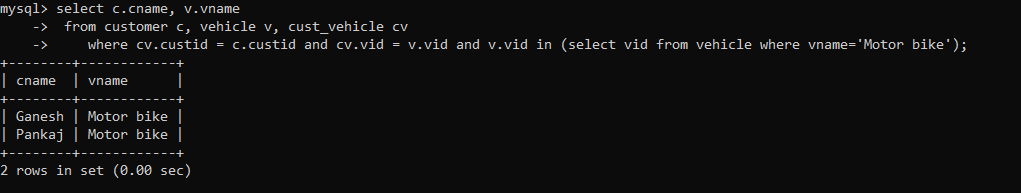
****

1. **find all customer name,vehicle name,salesman name for all salesman who stays in pune**

****

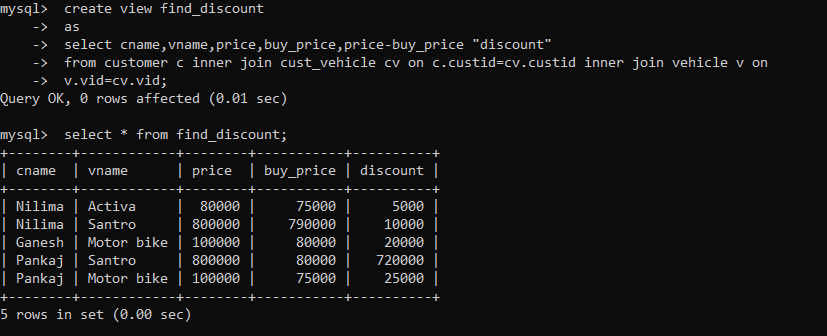
1. **find how many customers bought motor bike**

**select c.cname, v.vname**

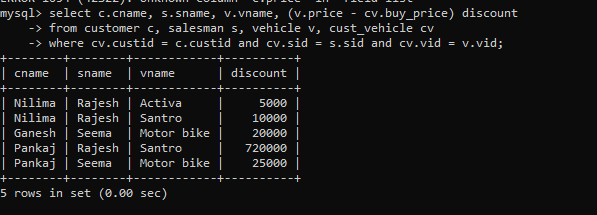
****

1. **create a view find\_discount which displays output**

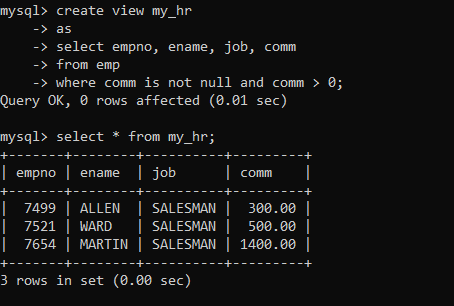
**-------to create view**

****

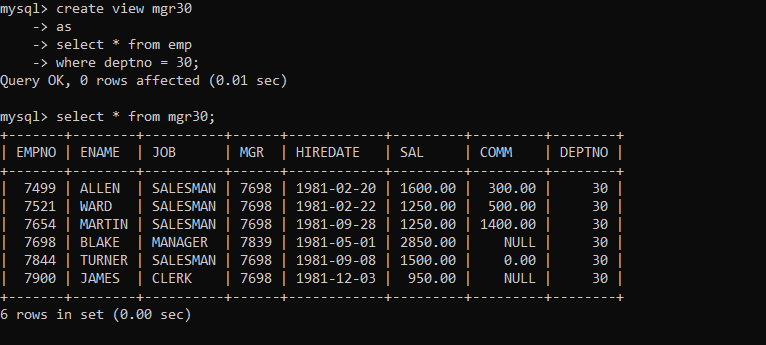
1. **find all customer name, vehicle name, salesman name, discount earn by all customer**

****

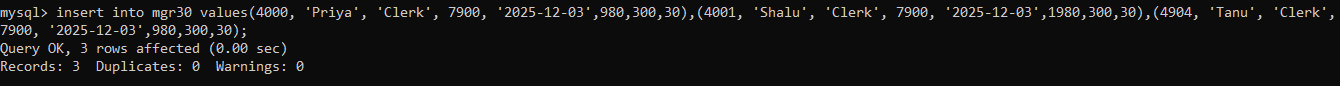
1. **create view my\_hr to display empno,ename,job,comm for all employees who earn commission**

****

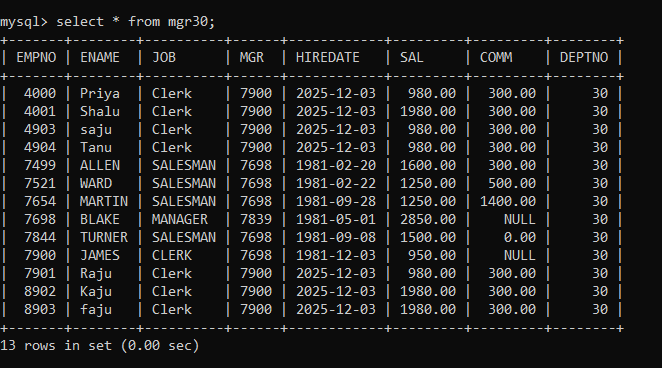
1. **create view mgr30 to display all employees from department 30**

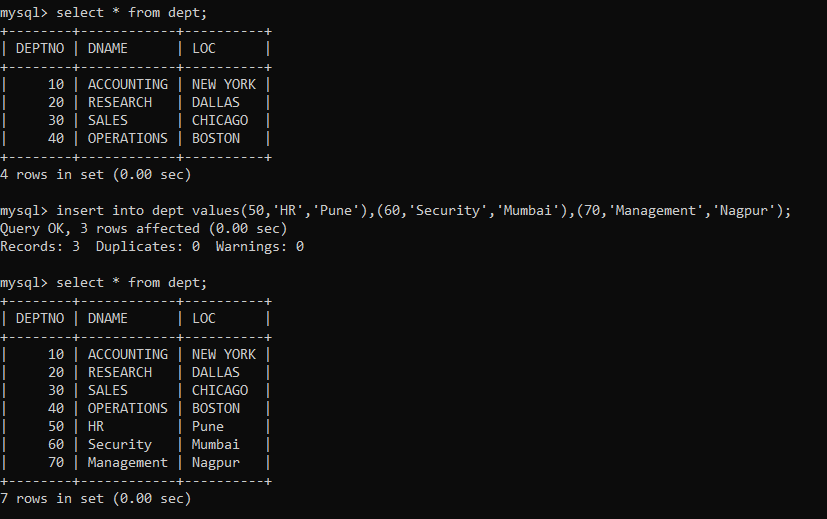
****

1. **insert 3 employees in view mgr30 check whether insertion is possible**

****

**select \* from mgr30;**

****

1. **insert 3 records in dept and display all records from dept**
2. **use rollback command check what happens**

**rollback;**

****

1. **do the following**

**set auto commint on**

**insert row in emp with empno 100**

**insert row in emp with empno 101**

**insert row in emp with empno 102**

**add savepoint A**

**insert row in emp with empno 103**

**insert row in emp with empno 104**

**insert row in emp with empno 105**

**add savepoint B**

**delete emp with empno 100**

**delete emp with emp no 104**

**rollback upto svaepoint B**

**check what all records will appear in employee table**

**rollback upto A check**

**what all records will appear in employee table commit all changes check what all records will appear in employee table check whether you can roll back the contents.**

1. **create a procedure getMin(deptno,minsal) to find minimum salary of given table**

**delimiter //**

**create procedure getMin(in pdno int, out pminsal double(9,2))**

**begin**

**select min(sal) into pminsal**

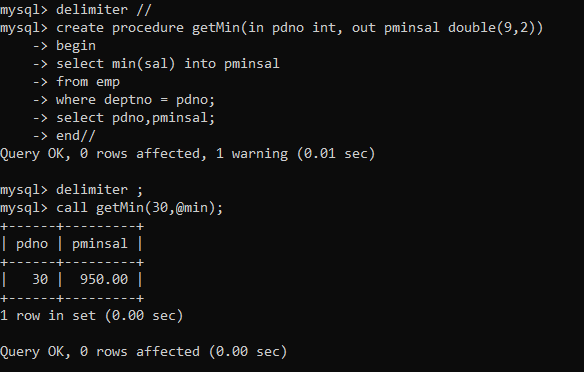
**from emp**

**where deptno = pdno;**

**select pdno,pminsal;**

**end//**

**delimiter ;**

****