**SQL Assignment**

Q-1 write a SQL query to find customers who are either from the city 'NewYork' or who do not have a grade greater than 100. Return customer\_id, cust\_name, city, grade, and salesman\_id.

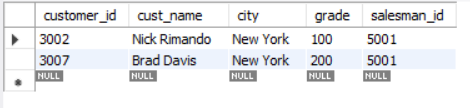
Input:-

select customer\_id,cust\_name,city,grade,salesman\_id

from customer

where city ='New York' or grade<100;

Output:-



Q-2 write a SQL query to find all the customers in ‘New York’ city who have agradevalue above 100. Return customer\_id, cust\_name, city, grade, and salesman\_id.

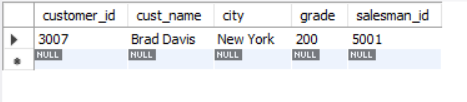
Input:-

select customer\_id,cust\_name,city,grade,salesman\_id

from customer

where city ='New York' and grade>100;

Output:-



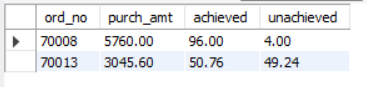
Q-3 Write a SQL query that displays order number, purchase amount, and the achieved and unachieved percentage (%) for those orders that exceed 50%of the target value of 6000.

Input:-

select ord\_no,purch\_amt,round((purch\_amt / 6000)\* 100,2) as achieved,round(100 - (purch\_amt / 6000)\* 100,2) as unachieved

from orders

where purch\_amt > 3000;

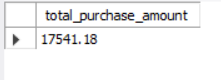
Output:-

Q-4 write a SQL query to calculate the total purchase amount of all orders. Returntotal purchase amount.

Input:-

select sum(purch\_amt) as total\_purchase\_amount

from orders;

Output:-

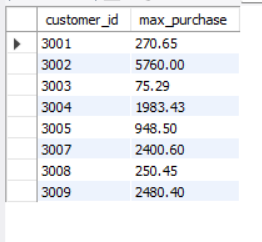
Q-5 write a SQL query to find the highest purchase amount ordered by each customer. Return customer ID, maximum purchase amount.

Input:-

select customer\_id, max(purch\_amt) as max\_purchase

from orders

group by customer\_id;

Output:-

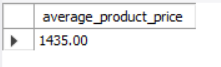
Q-6 write a SQL query to calculate the average product price. Return average product price.

Input:-

select \* from item\_mast;

select round(avg(pro\_price),2) as average\_product\_price

from item\_mast;

Output:-

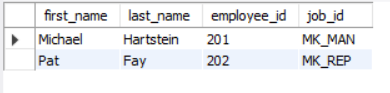
Q-7 write a SQL query to find those employees whose department is located at ‘Toronto’. Return first name, last name, employee ID, job ID.

Input:-

select first\_name,last\_name,employee\_id,job\_id

from emp\_details\_view

where city = 'Toronto';

Output:-

Q-8 write a SQL query to find those employees whose salary is lower than that of employees whose job title is "MK\_MAN". Exclude employees of the Jobtitle‘MK\_MAN’. Return employee ID, first name, last name, job ID.

Input:-

with min\_salary as (

select salary

from emp\_details\_view

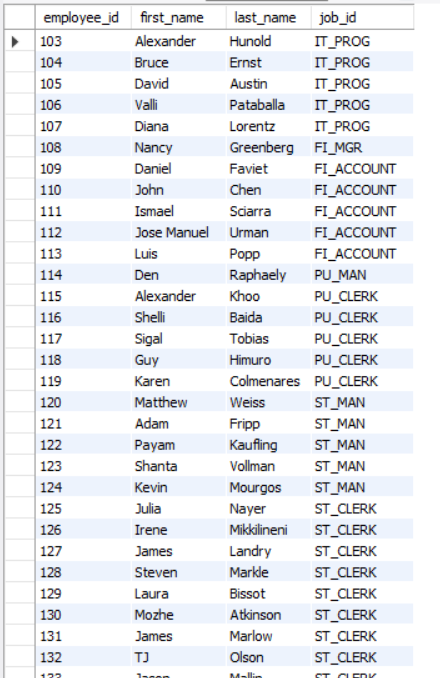
where job\_id='MK\_MAN'

)

select employee\_id, first\_name, last\_name, job\_id

from emp\_details\_view

where salary<(select salary from min\_salary)and job\_id<>'MK\_MAN';

Output:-

Q-9 write a SQL query to find all those employees who work in department ID80or40. Return first name, last name, department number and department name.

Input:-

select first\_name, last\_name, department\_id,department\_name

from emp\_details\_view

where department\_id = 80 or department\_id = 40;

Output:-

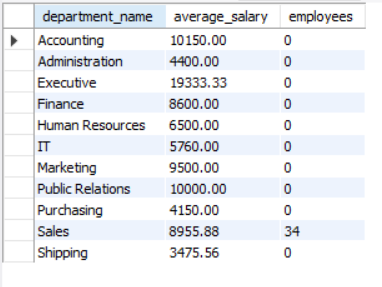
Q-10 write a SQL query to calculate the average salary, the number of employees receiving commissions in that department. Return department name, averagesalary and number of employees.

Input:-

Select department\_name,round(avg(salary),2)as average\_salary,count(commission\_pct) as employees

from emp\_details\_view

group by department\_name;

Output:-

Q-11 write a SQL query to find out which employees have the same designationas theemployee whose ID is 169. Return first name, last name, department IDandjobID.

Input:-

with emp as (

select job\_id

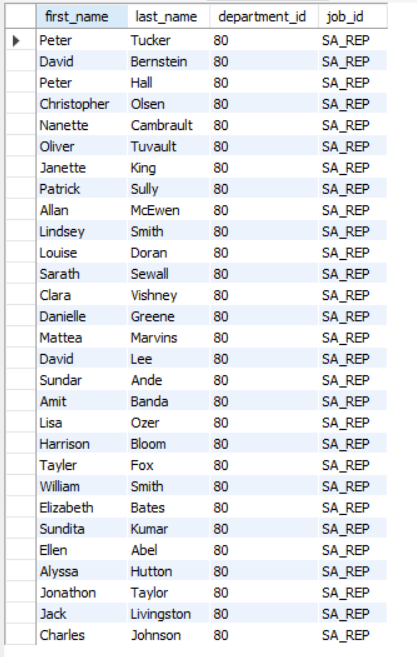
from emp\_details\_view

where employee\_id=169)

select first\_name, last\_name, department\_id,job\_id

from emp\_details\_view

where job\_id = (select job\_id from emp);

Output:-

Q-12 write a SQL query to find those employees who earn more than the averagesalary. Return employee ID, first name, last name.

Input:-

with avgs as (

select avg(salary) as avrage

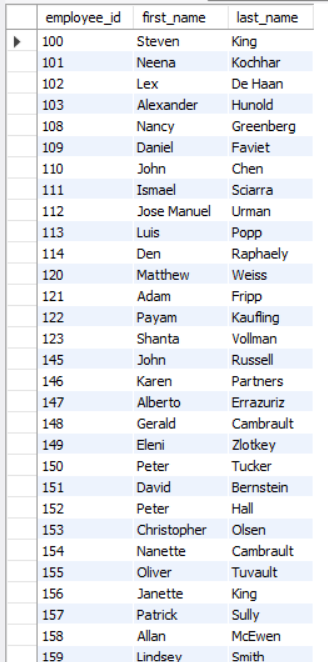
from emp\_details\_view

)

select employee\_ID, first\_name, last\_name

from emp\_details\_view

where salary >(select avrage from avgs);

Output:-

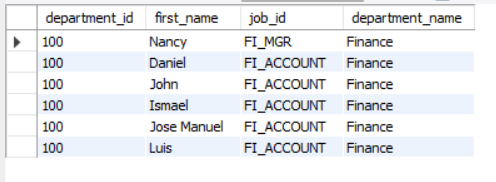
Q-13 write a SQL query to find all those employees who work in the Finance department. Return department ID, name (first), job ID and department name.

Input:-

select department\_id,first\_name, job\_id,department\_name

from emp\_details\_view

where department\_name = 'Finance';

Output:-

Q-14 From the following table, write a SQL query to find the employees who earn lessthan the employee of ID 182. Return first name, last name and salary.

Input:-

with employe as (

select salary

from emp\_details\_view

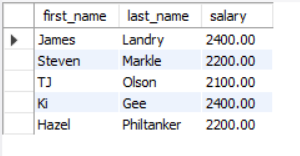
where employee\_id =182

)

select first\_name, last\_name,salary

from emp\_details\_view

where salary<(select salary from employe);

Output:-

Q-15 Create a stored procedure CountEmployees By Dept that returns the number of employees in each department.

Input:-

delimiter //

create procedure emp\_details()

begin

select department\_name,count(\*) as employees

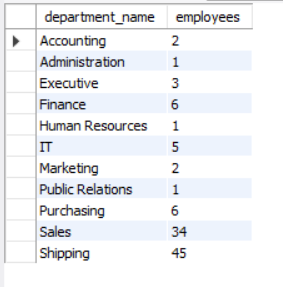
from emp\_details\_view

group by department\_name;

end;

//

call emp\_details();

Output:-

Q-16 Create a stored procedure AddNewEmployee that adds a new employee tothedatabase.

Input:-

delimiter //

create procedure add\_employee(

in emp\_id INT (11) UNSIGNED,

in fname VARCHAR(20),

in lname VARCHAR(25),

in em VARCHAR(25),

in pnumber VARCHAR(20),

in hdate DATE,

in jid VARCHAR(10),

in sal DECIMAL(8, 2),

in comm DECIMAL(2, 2),

in mid INT (11) UNSIGNED,

in did INT (11) UNSIGNED

)

begin

insert into employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,commission\_pct,manager\_id,department\_id)

values(emp\_id,fname,lname,em,pnumber,hdate,jid,sal,comm,mid,did);

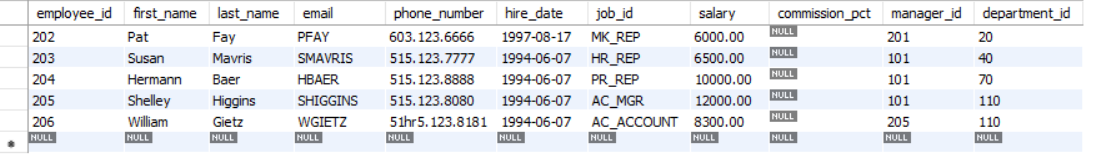
end;

//

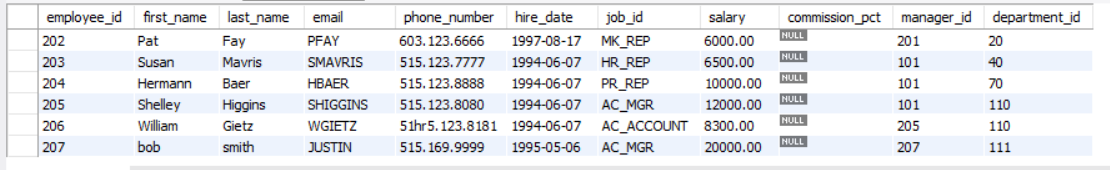
delimiter;

call add\_employee(207,'bob','smith','JUSTIN','515.169.9999','1995-05-06','AC\_MGR',20000.00,null,207,111);

select \* from employees;

Output:-

Before



After

Q-17 Create a stored procedure DeleteEmployeesByDept that removes all employeesfrom a specific department

Input:-

SET SQL\_SAFE\_UPDATES = 0;

delimiter //

create procedure delete\_emp(

in deptid int

)

begin

delete from employees

where department\_id = deptid;

end;

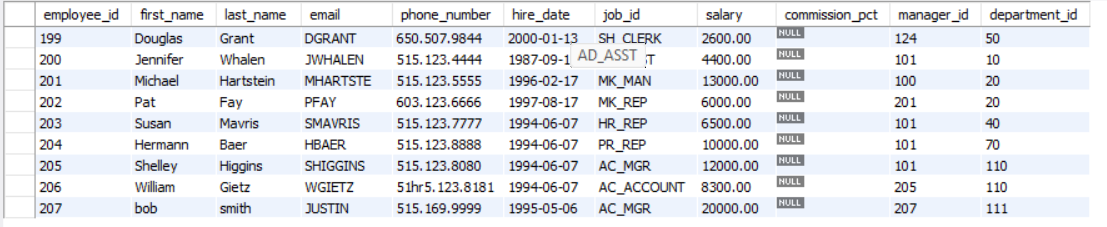
//

delimiter;

call delete\_emp(30);

select \* from employees;

Output:-



Q-18 Create a stored procedure GetTopPaidEmployees that retrieves the highest-paidemployee in each department.

Input:-

delimiter //

create procedure high\_paid\_emp()

begin

with emp as (

select first\_name,last\_name,salary,department\_name,

rank() over(partition by department\_name order by salary desc) as ranks

from emp\_details\_view)

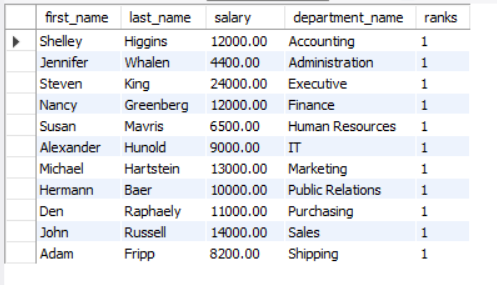
select \*

from emp

where ranks=1;

end//

delimiter;

Output:-

Q-19 Create a stored procedure PromoteEmployee that increases an employee’s salaryand changes their job role.

Input:-

delimiter //

create procedure promoteemployee (

in empid int,

in newjobid varchar(10),

in newsalary decimal(10,2)

)

begin

-- update salary and job id

update employees

set

salary = newsalary,

job\_id = newjobid

where

employee\_id = empid;

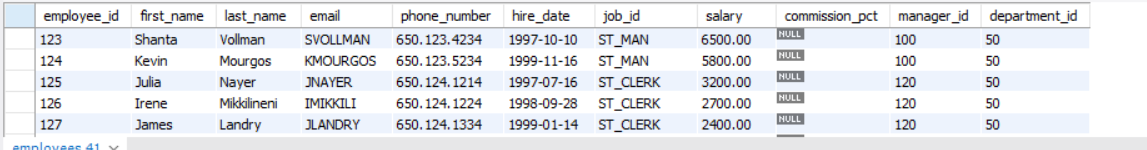
end;

//

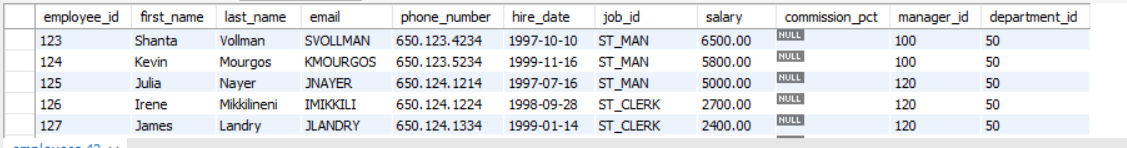
delimiter ;

select \* from employees;

call promoteemployee(125,'ST\_MAN',5000.00);

Output:-

Before



After

Q-20 Create a stored procedure AssignManagerToDepartment that assigns a newmanager to all employees in a specific department.

Input:-

select \* from departments;

delimiter //

create procedure assign (

in mid int,

in deptid int

)

begin

update departments

set manager\_id = mid

where department\_id = deptid;

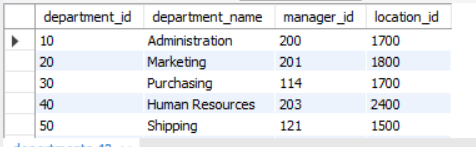
end;

//

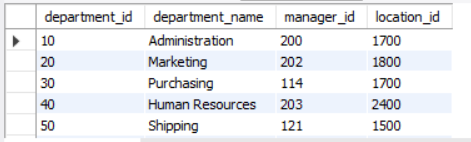
delimiter ;

call assign(202,20);

select \* from departments;

Output:-

Before



After