NAME: Rohit. M. Somani

LMS LOGIN ID:BFL2022067

SQL QUERIES ASSIGNMENT

Q1. **Write an SQL query to fetch the EmpId and FullName of all the employees working under Manager with id – '101'.**

```
postgres=# create table EmployeeDetails(Empid int primary key not null,FullName varchar(20),ManagerId int not null,city varchar(20));
CREATE TABLE
postgres=# select * from EmployeeDetails;
 empid | fullname | managerid | city
-------+----------+-----------+------
(0 rows)


postgres=# insert into EmployeeDetails(Empid,FullName,ManagerId,city) values (1,'Rohit',101,'Kolkata'),(2,'Raj',102,'Delhi'),(3,'Raghav',101,'Mumbai');
INSERT 0 3
postgres=# select * from EmployeeDetails;
 empid | fullname | managerid |  city
-------+----------+-----------+---------
     1 | Rohit    |       101 | Kolkata
     2 | Raj      |       102 | Delhi
     3 | Raghav   |       101 | Mumbai
(3 rows)


postgres=# create table EmployeeSalary(Empid int primary key not null,project varchar(2) not null,Salary int not null, variable int);
CREATE TABLE
postgres=# insert into EmployeeSalary(Empid,project,salary,variable) values (1,'P1',15000,5000),(2,'P2',10000,1000),(3,'P1',12000,0);
INSERT 0 3
postgres=# select Empid,FullName from EmployeeDetails where ManagerId=101;
 empid | fullname
-------+----------
     1 | Rohit
     3 | Raghav
(2 rows)
```

Q2**Write an SQL query to fetch the count of employees working in project 'P1'.**

```
postgres=# select count(Empid) from EmployeeSalary where project='P1';
 count
-------
     2
(1 row)
```

Q3 **Write an SQL query to find the maximum, minimum, and average salary of the employees.**

```
postgres=# select max(salary),min(salary),avg(salary) from EmployeeSalary;
  max  |  min  |          avg
-------+-------+-----------------------
 15000 | 10000 | 12333.3333333333333333
(1 row)
```

Q4 **Write an SQL query to find the employee id whose salary lies in the range of 10000 and 15000.**

```
postgres=# select Empid,salary from EmployeeSalary where salary between 10000 and 15000;
 empid | salary
-------+--------
     1 |  15000
     2 |  10000
     3 |  12000
(3 rows)
```

Q5 Write an SQL query to display the total salary of each employee adding the Salary with Variable value.

```
postgres=# select salary+variable as total_salary from EmployeeSalary;
 total_salary
--------------
        20000
        11000
        12000
(3 rows)
```

Q6 Write an SQL query to fetch the EmpIds that are present in both the tables – 'EmployeeDetails' and 'EmployeeSalary.

```
postgres=# select Empid from EmployeeDetails where Empid in(select Empid from EmployeeSalary);
 empid
-------
     1
     2
     3
(3 rows)
```

Q7 Write an SQL query to upper case the name of the employee and lower case the city values.

```
postgres=# select upper(FullName),lower(city) from EmployeeDetails;
 upper  |  lower
--------+---------
 ROHIT  | kolkata
 RAJ    | delhi
 RAGHAV | mumbai
(3 rows)
```

Q8 Write an SQL query to fetch project-wise count of employees sorted by project's count in descending order.

```
postgres=#  select project,count(Empid) from EmployeeSalary group by project order by count(Empid) desc;
 project | count
---------+-------
 P1      |     2
 P2      |     1
(2 rows)
```

Q9 Write an SQL query to fetch only odd rows from the table.

```
postgres=#  select * from EmployeeDetails where Mod(Empid,2)!=0;
 empid | fullname | managerid |  city
-------+----------+-----------+---------
     1 | Rohit    |       101 | Kolkata
     3 | Raghav   |       101 | Mumbai
```

Q10 Write SQL query to find the 3rd highest salary from a table without using the TOP/limit keyword.

```
postgres=# select max(salary) from EmployeeSalary where salary not in(select max(salary) from EmployeeSalary where salary not in(select max(salary) from EmployeeSalary));
  max
-------
 15000
(1 row)
```