SQL Assignment Task 1

1. create employee(emp_id,employee_name,department_name,location,salary), department(dept_id,department_name), locations (location_id, location_name) tables with relevant attributes. create primary key on each table and foreign keys (location->department, department->employee)

   create table employee(emp_id number, employee_name char,department_id number,location_id number,salary number,primary key(emp_id));

   create table department(dept_id number,department_name char,primary key(dept_id));

   create table locations(location_id number,location_name char,primary key(location_id));

   alter table employee add foreign key (department_id) references department(dept_id);

   alter table employee add foreign key (location_id) references locations(location_id);

2. insert 20 employee's data, 4 departments data, 2 locations data.

```
insert into department values(1,'PVSS-CE');
insert into department values(2,'PT');
insert into department values(3,'UI');
insert into department values(4,'PVD');

insert into locations values(1,'Noida');
insert into locations values(2,'Delhi');

insert into employee values(1,'A',1,1,6000);
insert into employee values(2,'B',2,2,8000);
insert into employee values(3,'C',3,1,11000);
insert into employee values(4,'D',4,2,15000);
insert into employee values(5,'E',1,1,4000);
insert into employee values(6,'F',2,2,9000);
insert into employee values(7,'G',3,1,12000);
insert into employee values(8,'H',4,2,20000);
insert into employee values(9,'I',1,1,2000);
insert into employee values(10,'J',2,2,7000);
insert into employee values(11,'K',3,1,13000);
insert into employee values(12,'L',4,2,18000);
insert into employee values(13,'M',4,1,19000);
insert into employee values(14,'N',4,2,17000);
insert into employee values(15,'O',4,1,16000);
insert into employee values(16,'P',4,2,14000);
insert into employee values(17,'Q',1,2,5000);
insert into employee values(18,'R',2,1,10000);
insert into employee values(19,'S',1,2,3000);
insert into employee values(20,'T',1,2,1000);
```

3. display all employees' names and their department names

   SELECT E.EMPLOYEE_NAME, D.DEPARTMENT_NAME FROM EMPLOYEE E JOIN DEPARTMENT D ON E.DEPARTMENT_ID = D.DEPT_ID;

4. display all location_name, department_name, employee_name, salary for all matching rows from 3 tables

   SELECT L.LOCATION_NAME, D.DEPARTMENT_NAME, E.EMPLOYEE_NAME, E.SALARY FROM EMPLOYEE E JOIN DEPARTMENT D ON E.DEPARTMENT_ID = D.DEPT_ID JOIN LOCATIONS L ON E.LOCATION_ID = L.LOCATION_ID;

5. select maximum salary earned from each department

   SELECT MAX(E.SALARY), D.DEPARTMENT_NAME FROM EMPLOYEE E JOIN DEPARTMENT D ON E.DEPARTMENT_ID = D.DEPT_ID GROUP BY D.DEPARTMENT_NAME;

6. select 2nd highest salary from each department.

   select max(salary), d.department_name from employee e join department d on e.department_id = d.dept_id where 1 = (select count (distinct salary) from employee m where m.salary > e.salary and e.department_id = m.department_id group by department_id) group by d.department_name;

7. select location_name, department_name, average_salary(of each location

   select avg(salary),location_name, department_name FROM EMPLOYEE E JOIN DEPARTMENT D ON E.DEPARTMENT_ID = D.DEPT_ID JOIN LOCATIONS L ON E.LOCATION_ID = L.LOCATION_ID group by location_name, department_name;

8. Show departments with no of employees

   SELECT COUNT(emp_id), department_name FROM employee e join department d on e.department_id = d.dept_id group by department_name;

9. Show locations with no of department where no of department is 2

   SELECT location_name,count(distinct employee.department_id) FROM employee JOIN locations ON employee.location_id = locations.location_id GROUP BY locations.location_name HAVING count(distinct employee.department_id) = 2;