DBMS assignment 3
Name: Rishika Varma K
Roll No: CS18B045

1. Query to find the maximum number of software accesses among the employees in each department of each branch along with names of the department and branch.

   ```
   select Department_Name, Branch_Name, max( softCount) as max_softwareAccesses
   from employee e,
   ( select Employee_Id, count(*) as softCount
   from has_access a, employee e
   where e.Job_Id = a.Job_Id
   group by Employee_Id) accesses
   where e.Employee_Id = accesses.Employee_Id
   group by Department_Name, Branch_Name;
   ```

2. Query for the employee Id, last name, first name, and number of projects that an employee who is a project lead as well as a supervisor works in. List those who work in atleast 2 projects.

   ```
   select e.Employee_Id, Last_Name, First_Name, count( w.Project_Id ) as NoOfProjects
   from employee e, works_in w
   where exists( select * from  project_ where e.Employee_Id = Project_lead) and
   exists( select * from reports_to where e.Employee_Id = Supervisor )
   and w.Employee_Id = e.Employee_Id
   group by e.Employee_Id, Last_Name, First_Name
   having count( w.Project_Id ) > 1;
   ```

3. Query to arrange all the project names in ascending order of the maximum work experience in that company among the employees working in the project.

   ```
   select Project_Name
   from project_ p,
   ( select Project_Id, min( e.Joining_Date) as minVal
   from works_in w, employee e
   where w.Employee_Id = e.Employee_Id
   group by Project_Id ) yearMin
   where p.Project_Id = yearMin.Project_Id
   order by minVal desc;
   ```

4. Query for the names (first and last names only) of all the pairs of employees for whom at least one of their first, middle, or last names match along with the common name. (if 2

employees have multiple names common then each of the common name is recorded as a separate entry)

```
select e1.First_Name, e1.Last_Name, e2.First_Name, e2.Last_Name, e1.First_Name as
commonName
from employee e1, employee e2
where e1.Employee_Id < e2.Employee_Id and e1.First_Name = e2.First_Name
union
select e1.First_Name, e1.Last_Name, e2.First_Name, e2.Last_Name, e1.Last_Name as
commonName
from employee e1, employee e2
where e1.Employee_Id < e2.Employee_Id and e1.Last_Name = e2.Last_Name
union
select e1.First_Name, e1.Last_Name, e2.First_Name, e2.Last_Name, m.Middle_Name
as commonName
from employee e1, employee e2, middle_names m
where e1.Employee_Id < e2.Employee_Id and
m.Middle_Name = ANY ( select m1.Middle_Name
from middle_names m1, middle_names m2
where m1.Employee_Id= e1.employee_Id and m2.Employee_Id = e2.Employee_Id and
m1.Middle_Name = m2.Middle_Name );
```

5. Query to find the names (last and first names) and emloyee Ids of all the employees who have listed more than 1 dependant along with their joining date and number of projects they are involved in.

```
        select Last_Name, First_Name, e.Employee_Id, Joining_Date, count(w.Project_Id) as
NoOfProjects
        from employee e, works_in w,
        ( select Employee_Id, count(*) as cnt
        from dependants
        group by Employee_Id ) DepCount
        where e.Employee_Id = DepCount.Employee_Id and cnt>1
        and w.Employee_Id = e.Employee_Id
        group by Last_Name, First_Name, Employee_Id, Joining_Date;
```

6. Find the average salary of all the employees who studied in the same branch and are working in the same job title. Arrange in ascending order.

```
select Job_Name, e.Job_Id, Branch, avg( Salary )
from employee e, Job_Title j, education_entries p
where e.Job_Id = j.Job_Id and e.Employee_Id = p.Employee_Id
```

```
group by Job_Name, e.Job_Id, Branch
order by avg( Salary ) asc;
```