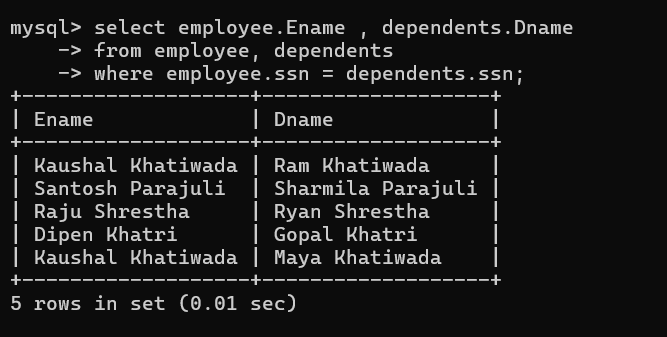
**Prepare Lab Sheet of MYSQL Statements for following. Use the Company Database in Lab-1 and Lab-2.**

1. Select the names of employees and their dependents without using JOIN.

***select*** *employee.Ename , dependents.Dname*

***from*** *employee, dependents*

***where*** *employee.ssn = dependents.ssn;*



Using foreign key reference.

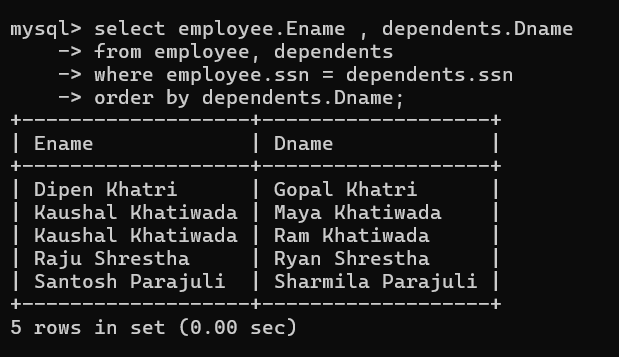
1. Select the names of employees and their dependents without using INNER JOIN and order the result based on dependents name.

***select*** *employee.Ename , dependents.Dname*

***from*** *employee, dependents*

***where*** *employee.ssn = dependents.ssn*

***order by*** *dependents.Dname;*



1. Use JOIN between Employee, Project and Works\_on and retrieve the name of employees and the projects on which they work.

***select*** *employee.Ename, project.Pname*

***from*** *employee*

***join*** *works\_on* ***on*** *employee.SSN = works\_on.ESSN*

***join*** *project* ***on*** *works\_on.Pno = project.Pnumber;*

A screenshot of a computer program

Description automatically generated

JOIN is used to combine rows from two or more tables based on a related column between them.

1. Use Inner join between Employee and PF table with the join condition, Employee.SSN=PF.SSN and Employee.Salary>PF.Amount

***select*** *\**

***from*** *employee*

***inner******join*** *pf* ***on*** *employee.SSN = pf.SSN* ***and*** *employee.Salary>pf.Amount;*

A screenshot of a computer screen

Description automatically generated

**INNER JOIN** returns records that have matching values in both tables

1. Write a query to show the results of Left and Right Join between Office and Project.

**LEFT JOIN**

***select*** *\**

***from*** *office*

***left******join*** *project* ***on*** *office.Onumber = project.Onumber;*

**RIGHT JOIN**

***select*** *\**

***from*** *office*

***right******join*** *project* ***on*** *office.Onumber = project.Onumber;*

A screenshot of a computer screen

Description automatically generated

**LEFT JOIN** returns all the rows from the left side table and only matching rows from the right-side table.

For no matching row on right side, it will contain *NULL*.

**RIGHT JOIN** returns all the rows from the right side and only matching rows from the left side table.

For no matching row on left side, it will contain *NULL.*

1. Write a query to show the results of Cross Join between Employee and PF tables.

***select*** *\**

***from*** *employee*

***cross******join*** *pf;*

A screenshot of a computer screen

Description automatically generated

.

.

.

.

.

A black and white screen with white text

Description automatically generated

**CROSS JOIN** performs Cartesian Product of two tables. Every row of the first table is joined with every row of the second table. All possible combinations of rows are displayed.

1. Show results of using natural join between Employee and PF.

**select** \*

**from** employee

**natural** **join** pf;

A screenshot of a computer screen

Description automatically generated

**NATURAL JOIN** performs the join operation on the base of common columns in the tables. i.e. SSN.

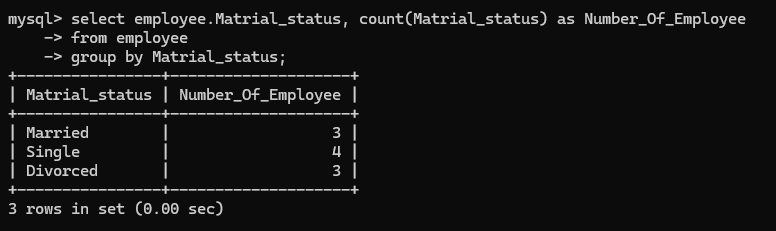
At least one common attribute is required between the tables.

1. Find the number of employees and status in each status of “Married”, “Single”, “Divorced”. Use the COUNT function with the GROUP BY clause with status.

***select*** *employee.Matrial\_status,* ***count****(Matrial\_status)* ***as*** *Number\_Of\_Employee*

***from*** *employee*

***group******by*** *Matrial\_status;*



**COUNT** returns the number of rows that matches a specified criterion.

1. Find the number of employees and status in each status of “Married” OR “Single”. Use the COUNT function with the GROUP BY clause with status and Having clause with status = “Married” OR “Single”

***select*** *employee.Matrial\_status,* ***count****(Matrial\_status)* ***as*** *Number\_Of\_Employee*

***from*** *employee*

***group******by*** *Matrial\_status*

***having*** *Matrial\_status="Married"* ***or*** *Matrial\_status="Single";*

A screenshot of a computer program

Description automatically generated

1. Using sub query, select the name and location of projects whose Onumber is in the Onumber of the offices located in country Nepal and India.

***select*** *Pname, Proj\_location*

***from*** *project*

***where*** *Onumber* ***in*** *(*

***select*** *Onumber*

***from*** *office*

***where*** *country* ***in*** *('Nepal', 'India')*

*);*

A screen shot of a computer

Description automatically generated

Subquery is a query within another query. Subqueries are embedded in the WHERE clause with IN operator.