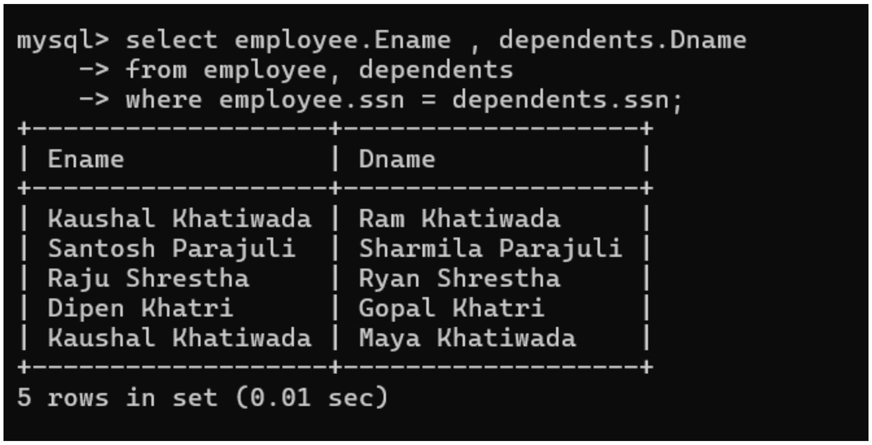
Lab 3: JOIN Operations, Group by & Having Clause and Sub queries

**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;*

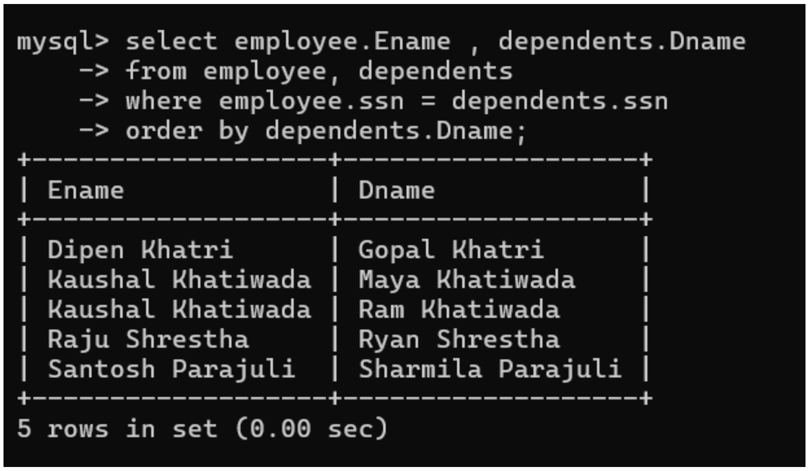


```
mysql> select employee.Ename , dependents.Dname
    -> from employee, dependents
    -> where employee.ssn = dependents.ssn;
+-------------------+-------------------+
| Ename             | Dname             |
+-------------------+-------------------+
| Kaushal Khatiwada | Ram Khatiwada     |
| Santosh Parajuli  | Sharmila Parajuli |
| Raju Shrestha     | Ryan Shrestha     |
| Dipen Khatri      | Gopal Khatri      |
| Kaushal Khatiwada | Maya Khatiwada    |
+-------------------+-------------------+
5 rows in set (0.01 sec)
```

   Using foreign key reference.

2. 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;*

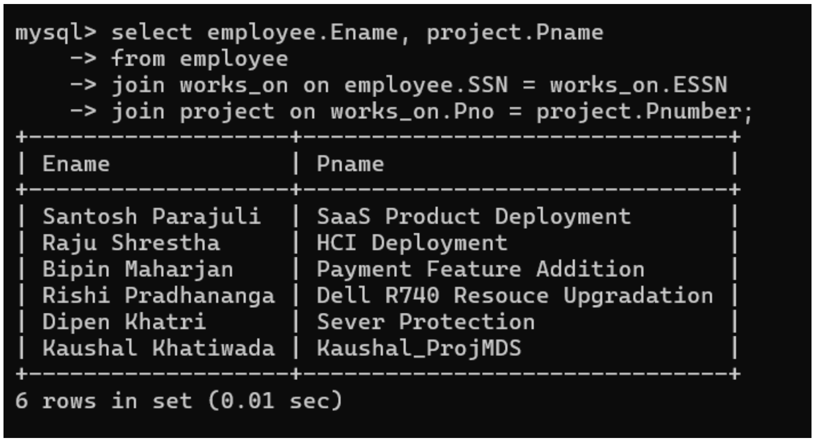


```
mysql> select employee.Ename , dependents.Dname
    -> from employee, dependents
    -> where employee.ssn = dependents.ssn
    -> order by dependents.Dname;
+-------------------+-------------------+
| Ename             | Dname             |
+-------------------+-------------------+
| Dipen Khatri      | Gopal Khatri      |
| Kaushal Khatiwada | Maya Khatiwada    |
| Kaushal Khatiwada | Ram Khatiwada     |
| Raju Shrestha     | Ryan Shrestha     |
| Santosh Parajuli  | Sharmila Parajuli |
+-------------------+-------------------+
5 rows in set (0.00 sec)
```

Lab 3: JOIN Operations, Group by & Having Clause and Sub queries

3. 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;*

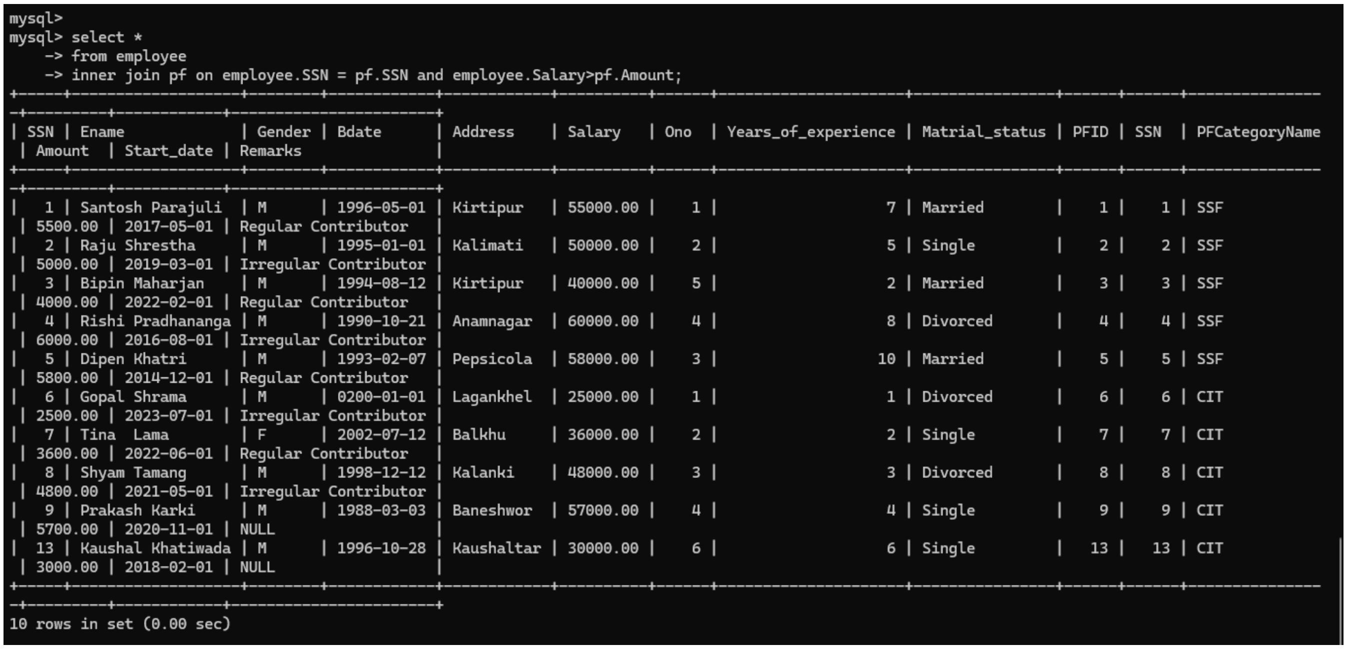


```
mysql> select employee.Ename, project.Pname
    -> from employee
    -> join works_on on employee.SSN = works_on.ESSN
    -> join project on works_on.Pno = project.Pnumber;
+-------------------+-------------------------------+
| Ename             | Pname                         |
+-------------------+-------------------------------+
| Santosh Parajuli  | SaaS Product Deployment       |
| Raju Shrestha     | HCI Deployment                |
| Bipin Maharjan    | Payment Feature Addition      |
| Rishi Pradhananga | Dell R740 Resouce Upgradation |
| Dipen Khatri      | Sever Protection              |
| Kaushal Khatiwada | Kaushal_ProjMDS               |
+-------------------+-------------------------------+
6 rows in set (0.01 sec)
```

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

4. 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;*



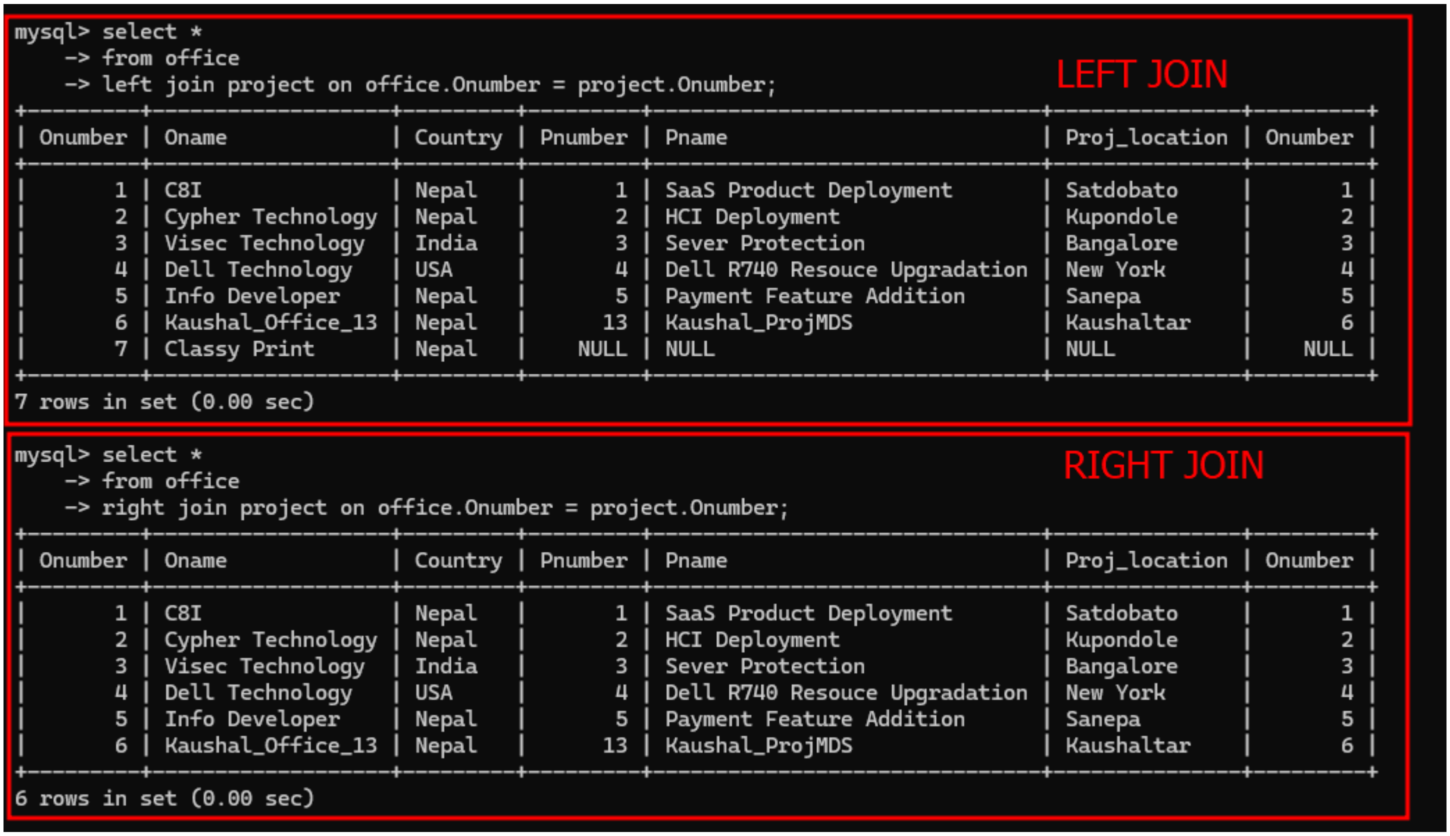
```
mysql>
mysql> select *
    -> from employee
    -> inner join pf on employee.SSN = pf.SSN and employee.Salary>pf.Amount;
+-----+-------------------+--------+------------+------------+----------+------+---------------------+----------------+------+------+---------------
-+---------+------------+-----------------------+
| SSN | Ename             | Gender | Bdate      | Address    | Salary   | Ono  | Years_of_experience | Matrial_status | PFID | SSN  | PFCategoryName
 | Amount  | Start_date | Remarks               |
+-----+-------------------+--------+------------+------------+----------+------+---------------------+----------------+------+------+---------------
-+---------+------------+-----------------------+
|   1 | Santosh Parajuli  | M      | 1996-05-01 | Kirtipur   | 55000.00 |    1 |                   7 | Married        |    1 |    1 | SSF
 | 5500.00 | 2017-05-01 | Regular Contributor   |
|   2 | Raju Shrestha     | M      | 1995-01-01 | Kalimati   | 50000.00 |    2 |                   5 | Single         |    2 |    2 | SSF
 | 5000.00 | 2019-03-01 | Irregular Contributor |
|   3 | Bipin Maharjan    | M      | 1994-08-12 | Kirtipur   | 40000.00 |    5 |                   2 | Married        |    3 |    3 | SSF
 | 4000.00 | 2022-02-01 | Regular Contributor   |
|   4 | Rishi Pradhananga | M      | 1990-10-21 | Anamnagar  | 60000.00 |    4 |                   8 | Divorced       |    4 |    4 | SSF
 | 6000.00 | 2016-08-01 | Irregular Contributor |
|   5 | Dipen Khatri      | M      | 1993-02-07 | Pepsicola  | 58000.00 |    3 |                  10 | Married        |    5 |    5 | SSF
 | 5800.00 | 2014-12-01 | Regular Contributor   |
|   6 | Gopal Shrama      | M      | 0200-01-01 | Lagankhel  | 25000.00 |    1 |                   1 | Divorced       |    6 |    6 | CIT
 | 2500.00 | 2023-07-01 | Irregular Contributor |
|   7 | Tina  Lama        | F      | 2002-07-12 | Balkhu     | 36000.00 |    2 |                   2 | Single         |    7 |    7 | CIT
 | 3600.00 | 2022-06-01 | Regular Contributor   |
|   8 | Shyam Tamang      | M      | 1998-12-12 | Kalanki    | 48000.00 |    3 |                   3 | Divorced       |    8 |    8 | CIT
 | 4800.00 | 2021-05-01 | Irregular Contributor |
|   9 | Prakash Karki     | M      | 1988-03-03 | Baneshwor  | 57000.00 |    4 |                   4 | Single         |    9 |    9 | CIT
 | 5700.00 | 2020-11-01 | NULL                  |
|  13 | Kaushal Khatiwada | M      | 1996-10-28 | Kaushaltar | 30000.00 |    6 |                   6 | Single         |   13 |   13 | CIT
 | 3000.00 | 2018-02-01 | NULL                  |
+-----+-------------------+--------+------------+------------+----------+------+---------------------+----------------+------+------+---------------
-+---------+------------+-----------------------+
10 rows in set (0.00 sec)
```

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

5. 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;*



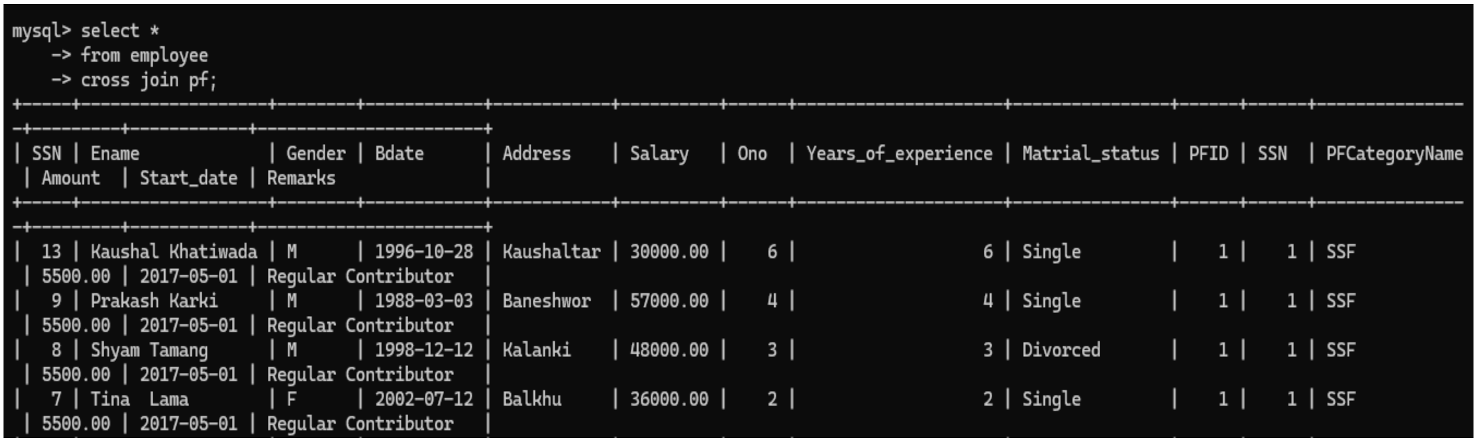
```
mysql> select *
    -> from office
    -> left join project on office.Onumber = project.Onumber;
+---------+--------------------+---------+---------+--------------------------------+--------------+---------+
| Onumber | Oname              | Country | Pnumber | Pname                          | Proj_location | Onumber |
+---------+--------------------+---------+---------+--------------------------------+--------------+---------+
|       1 | C8I                | Nepal   |       1 | SaaS Product Deployment        | Satdobato    |       1 |
|       2 | Cypher Technology  | Nepal   |       2 | HCI Deployment                 | Kupondole    |       2 |
|       3 | Visec Technology   | India   |       3 | Sever Protection               | Bangalore    |       3 |
|       4 | Dell Technology    | USA     |       4 | Dell R740 Resouce Upgradation  | New York     |       4 |
|       5 | Info Developer     | Nepal   |       5 | Payment Feature Addition       | Sanepa       |       5 |
|       6 | Kaushal_Office_13  | Nepal   |      13 | Kaushal_ProjMDS                | Kaushaltar   |       6 |
|       7 | Classy Print       | Nepal   |    NULL | NULL                           | NULL         |    NULL |
+---------+--------------------+---------+---------+--------------------------------+--------------+---------+
7 rows in set (0.00 sec)

mysql> select *
    -> from office
    -> right join project on office.Onumber = project.Onumber;
+---------+--------------------+---------+---------+--------------------------------+--------------+---------+
| Onumber | Oname              | Country | Pnumber | Pname                          | Proj_location | Onumber |
+---------+--------------------+---------+---------+--------------------------------+--------------+---------+
|       1 | C8I                | Nepal   |       1 | SaaS Product Deployment        | Satdobato    |       1 |
|       2 | Cypher Technology  | Nepal   |       2 | HCI Deployment                 | Kupondole    |       2 |
|       3 | Visec Technology   | India   |       3 | Sever Protection               | Bangalore    |       3 |
|       4 | Dell Technology    | USA     |       4 | Dell R740 Resouce Upgradation  | New York     |       4 |
|       5 | Info Developer     | Nepal   |       5 | Payment Feature Addition       | Sanepa       |       5 |
|       6 | Kaushal_Office_13  | Nepal   |      13 | Kaushal_ProjMDS                | Kaushaltar   |       6 |
+---------+--------------------+---------+---------+--------------------------------+--------------+---------+
6 rows in set (0.00 sec)
```

**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.*

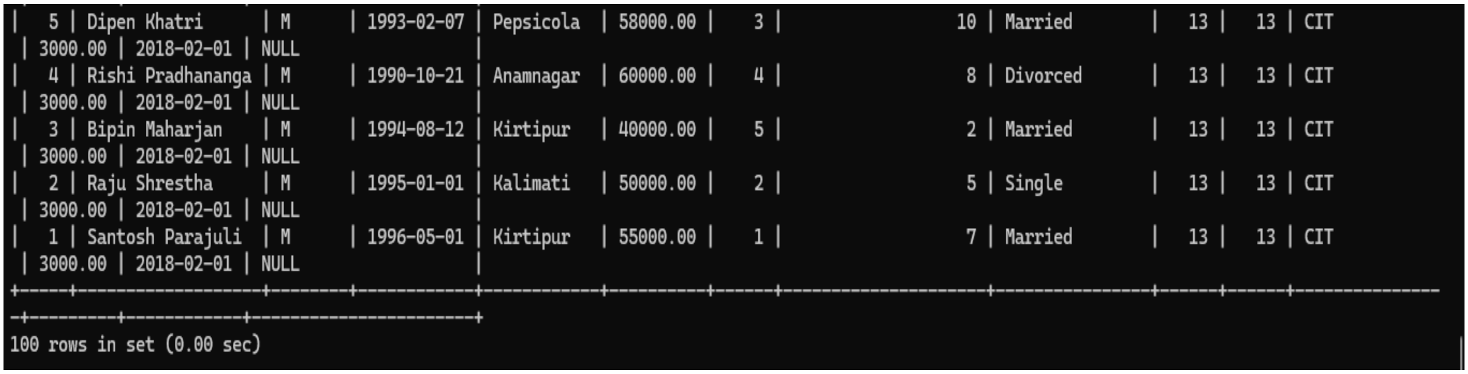
6. Write a query to show the results of Cross Join between Employee and PF tables.

   ***select*** ***
   ***from*** *employee*
   ***cross join*** *pf;*



```
mysql> select *
    -> from employee
    -> cross join pf;
+-----+-------------------+--------+------------+------------+----------+------+---------------------+----------------+------+------+---------------
-+---------+------------+----------------------+
| SSN | Ename             | Gender | Bdate      | Address    | Salary   | Ono  | Years_of_experience | Matrial_status | PFID | SSN  | PFCategoryName
 | Amount  | Start_date | Remarks              |
+-----+-------------------+--------+------------+------------+----------+------+---------------------+----------------+------+------+---------------
-+---------+------------+----------------------+
|  13 | Kaushal Khatiwada | M      | 1996-10-28 | Kaushaltar | 30000.00 |    6 |                   6 | Single         |    1 |    1 | SSF
 | 5500.00 | 2017-05-01 | Regular Contributor  |
|   9 | Prakash Karki     | M      | 1988-03-03 | Baneshwor  | 57000.00 |    4 |                   4 | Single         |    1 |    1 | SSF
 | 5500.00 | 2017-05-01 | Regular Contributor  |
|   8 | Shyam Tamang      | M      | 1998-12-12 | Kalanki    | 48000.00 |    3 |                   3 | Divorced       |    1 |    1 | SSF
 | 5500.00 | 2017-05-01 | Regular Contributor  |
|   7 | Tina  Lama        | F      | 2002-07-12 | Balkhu     | 36000.00 |    2 |                   2 | Single         |    1 |    1 | SSF
 | 5500.00 | 2017-05-01 | Regular Contributor  |
```

.
.
.
.
.

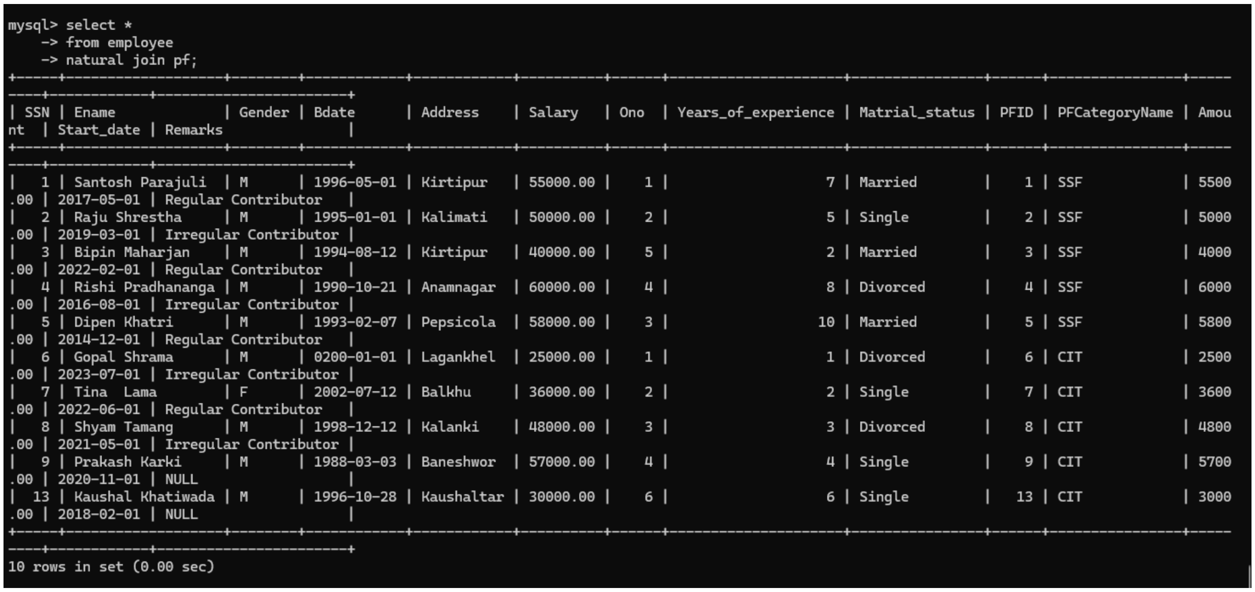


```
|   5 | Dipen Khatri      | M      | 1993-02-07 | Pepsicola  | 58000.00 |    3 |                  10 | Married        |   13 |   13 | CIT
 | 3000.00 | 2018-02-01 | NULL                 |
|   4 | Rishi Pradhananga | M      | 1990-10-21 | Anamnagar  | 60000.00 |    4 |                   8 | Divorced       |   13 |   13 | CIT
 | 3000.00 | 2018-02-01 | NULL                 |
|   3 | Bipin Maharjan    | M      | 1994-08-12 | Kirtipur   | 40000.00 |    5 |                   2 | Married        |   13 |   13 | CIT
 | 3000.00 | 2018-02-01 | NULL                 |
|   2 | Raju Shrestha     | M      | 1995-01-01 | Kalimati   | 50000.00 |    2 |                   5 | Single         |   13 |   13 | CIT
 | 3000.00 | 2018-02-01 | NULL                 |
|   1 | Santosh Parajuli  | M      | 1996-05-01 | Kirtipur   | 55000.00 |    1 |                   7 | Married        |   13 |   13 | CIT
 | 3000.00 | 2018-02-01 | NULL                 |
+-----+-------------------+--------+------------+------------+----------+------+---------------------+----------------+------+------+---------------
-+---------+------------+----------------------+
100 rows in set (0.00 sec)
```

**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.

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

    **select** *
    **from** employee
    **natural join** pf;

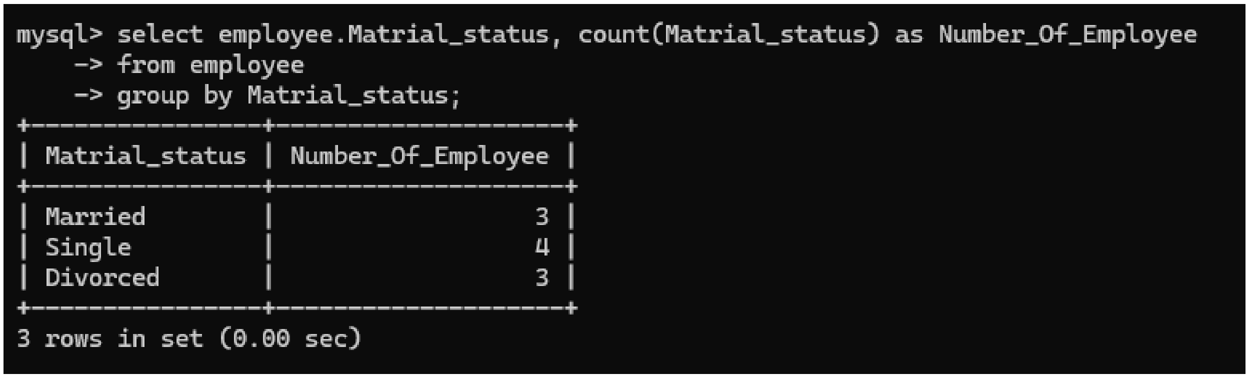


```
mysql> select *
    -> from employee
    -> natural join pf;
+-----+-------------------+--------+------------+------------+----------+------+---------------------+----------------+------+----------------+-----
----+------------+-----------------------+
| SSN | Ename             | Gender | Bdate      | Address    | Salary   | Ono  | Years_of_experience | Matrial_status | PFID | PFCategoryName | Amou
nt  | Start_date | Remarks               |
+-----+-------------------+--------+------------+------------+----------+------+---------------------+----------------+------+----------------+-----
----+------------+-----------------------+
|   1 | Santosh Parajuli  | M      | 1996-05-01 | Kirtipur   | 55000.00 |    1 |                   7 | Married        |    1 | SSF            | 5500
.00 | 2017-05-01 | Regular Contributor   |
|   2 | Raju Shrestha     | M      | 1995-01-01 | Kalimati   | 50000.00 |    2 |                   5 | Single         |    2 | SSF            | 5000
.00 | 2019-03-01 | Irregular Contributor |
|   3 | Bipin Maharjan    | M      | 1994-08-12 | Kirtipur   | 40000.00 |    5 |                   2 | Married        |    3 | SSF            | 4000
.00 | 2022-02-01 | Regular Contributor   |
|   4 | Rishi Pradhananga | M      | 1990-10-21 | Anamnagar  | 60000.00 |    4 |                   8 | Divorced       |    4 | SSF            | 6000
.00 | 2016-08-01 | Irregular Contributor |
|   5 | Dipen Khatri      | M      | 1993-02-07 | Pepsicola  | 58000.00 |    3 |                  10 | Married        |    5 | SSF            | 5800
.00 | 2014-12-01 | Regular Contributor   |
|   6 | Gopal Shrama      | M      | 0200-01-01 | Lagankhel  | 25000.00 |    1 |                   1 | Divorced       |    6 | CIT            | 2500
.00 | 2023-07-01 | Irregular Contributor |
|   7 | Tina  Lama        | F      | 2002-07-12 | Balkhu     | 36000.00 |    2 |                   2 | Single         |    7 | CIT            | 3600
.00 | 2022-06-01 | Regular Contributor   |
|   8 | Shyam Tamang      | M      | 1998-12-12 | Kalanki    | 48000.00 |    3 |                   3 | Divorced       |    8 | CIT            | 4800
.00 | 2021-05-01 | Irregular Contributor |
|   9 | Prakash Karki     | M      | 1988-03-03 | Baneshwor  | 57000.00 |    4 |                   4 | Single         |    9 | CIT            | 5700
.00 | 2020-11-01 | NULL                  |
|  13 | Kaushal Khatiwada | M      | 1996-10-28 | Kaushaltar | 30000.00 |    6 |                   6 | Single         |   13 | CIT            | 3000
.00 | 2018-02-01 | NULL                  |
+-----+-------------------+--------+------------+------------+----------+------+---------------------+----------------+------+----------------+-----
----+------------+-----------------------+
10 rows in set (0.00 sec)
```

**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.

8. 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;*

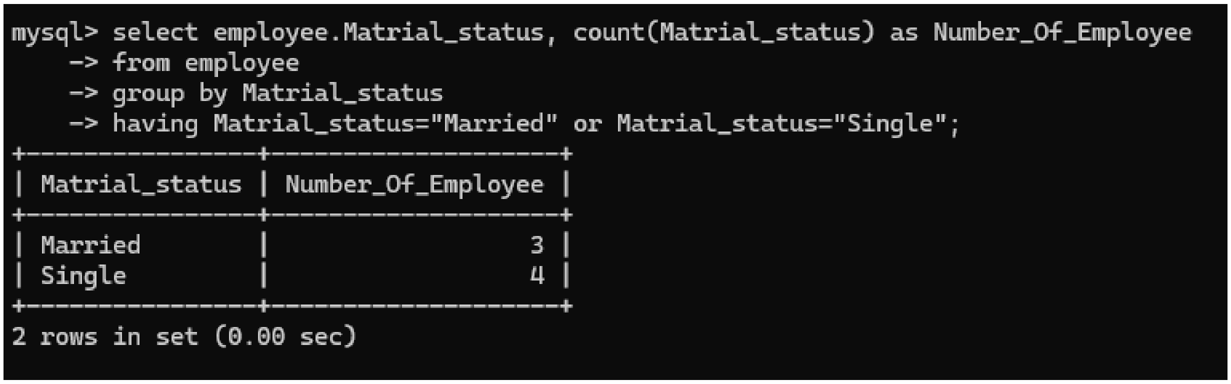


```
mysql> select employee.Matrial_status, count(Matrial_status) as Number_Of_Employee
    -> from employee
    -> group by Matrial_status;
+----------------+--------------------+
| Matrial_status | Number_Of_Employee |
+----------------+--------------------+
| Married        |                  3 |
| Single         |                  4 |
| Divorced       |                  3 |
+----------------+--------------------+
3 rows in set (0.00 sec)
```

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

9. 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”

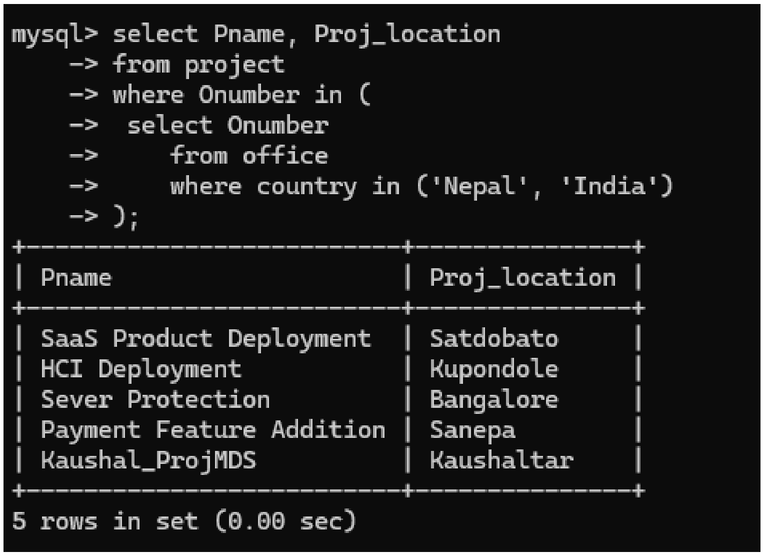
```
select employee.Matrial_status, count(Matrial_status) as Number_Of_Employee
from employee
group by Matrial_status
having Matrial_status="Married" or Matrial_status="Single";
```



```
mysql> 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";
+----------------+--------------------+
| Matrial_status | Number_Of_Employee |
+----------------+--------------------+
| Married        |                  3 |
| Single         |                  4 |
+----------------+--------------------+
2 rows in set (0.00 sec)
```

10. 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')
);
```



```
mysql> select Pname, Proj_location
    -> from project
    -> where Onumber in (
    ->  select Onumber
    ->     from office
    ->     where country in ('Nepal', 'India')
    -> );
+--------------------------+---------------+
| Pname                    | Proj_location |
+--------------------------+---------------+
| SaaS Product Deployment  | Satdobato     |
| HCI Deployment           | Kupondole     |
| Sever Protection         | Bangalore     |
| Payment Feature Addition | Sanepa        |
| Kaushal_ProjMDS          | Kaushaltar    |
+--------------------------+---------------+
5 rows in set (0.00 sec)
```

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

Lab 3: JOIN Operations, Group by & Having Clause and Sub queries