

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

```
# Lab3
SELECT e.Ename, d.Dname FROM EMPLOYEE e, Dependents d
WHERE e.SSN = d.SSN;
```

EMPLOYEE(+) 1 x

SELECT e.Ename, d.Dname FROM EMPLOYEE e, Dependents d WHERE e.SSN = d.SSN Enter a SQL expression to filter results (use Ctrl+Sp

	ABC Ename	ABC Dname	
1	Prabhat	Kabir	
2	Suman	Subin	
3	Anish	Anisha	
4	Gaurav	Gauri	
5	Ashmita	Alisha	
6	Kabita	Atul	
7	Kusum	Kaushal	
8	Pratigya	Jenisa	
9	Abin	Asbina	

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

```
SELECT e.Ename, d.Dname FROM EMPLOYEE e, Dependents d
WHERE e.SSN = d.SSN
ORDER BY d.Dname;
```

EMPLOYEE(+) 1 x

SELECT e.Ename, d.Dname FROM EMPLOYEE e, Dependents d WHERE e.SSN = d.SSN Enter a SQL expression to filter results (use Ctrl+Space

	ABC Ename	ABC Dname	
1	Ashmita	Alisha	
2	Anish	Anisha	
3	Abin	Asbina	
4	Kabita	Atul	
5	Gaurav	Gauri	
6	Pratigya	Jenisa	
7	Prabhat	Kabir	
8	Kusum	Kaushal	
9	Suman	Subin	

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

```

SELECT e.Ename, p.Pname FROM EMPLOYEE e JOIN Works_on w
ON e.SSN = w.ESSN
JOIN Project p
ON w.Pno = p.Pnumber;

```

	Ename	Pname
1	Abin	Abin_ProjML
2	Anish	Anish_ProjPandas
3	Ashmita	Ashmita_ProjMDS
4	Gaurav	Gaurav_ProjPython
5	Kabita	Kabita_ProjDBMS
6	Kusum	Kusum_ProjPython
7	Prabhat	Prabhat_ProjDBMS
8	Pratigya	Pratigya_ProjPandas
9	Suman	Suman_ProjML

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 e INNER JOIN PF p
WHERE e.SSN = p.SSN AND
e.Salary > p.Amount;

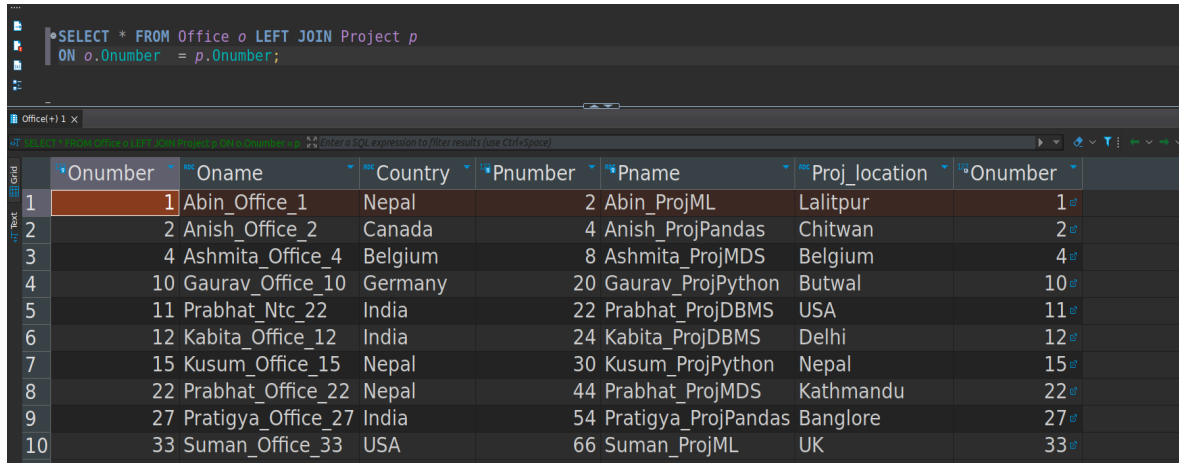
```

	SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_experience	Marital_Status	PFID	SSN	PFCategoryName	Amount	Start_date	Remarks
1	6	Ashmita	F	2053-12-05	Brussels	95,000	4		5 Divorced	111	6	Employee_Provident_Fund	4,500	2079-12-05	Regular_Monthly_Contribution
2	7	Kabita	F	2054-12-15	Delhi	80,000	12		4 Single	222	7	Pension_Fund	5,500	2078-12-15	Monthly_Contribution_For_Pension
3	8	Kusum	F	2052-07-06	Sunwal	75,000	15		2 Single	333	8	Retirement_Fund	4,700	2076-07-06	Contribution_For_Retirement
4	9	Pratigya	F	2057-11-15	Bangalore	80,000	27		3 Single	444	9	Children_Savings_Fund	5,800	2080-11-15	Contribution_For_Children_Education
5	10	Abin	M	2051-07-09	Sunwal	75,000	1		2 Single	555	10	General_Provident_Fund	2,500	2074-07-09	Normal_Monthly_Contribution
6	2	Gaurav	M	2056-07-24	MacheGaun	30,000	10		1 Married	1,111	2	Employee_Provident_Fund	3,000	2078-12-05	[NULL]
7	2	Gaurav	M	2056-07-24	MacheGaun	30,000	10		1 Married	2,222	2	Pension_Fund	6,000	2077-12-15	[NULL]
8	3	Anish	M	2053-12-05	Budhanilkandha	95,000	2		3 Married	3,333	3	Retirement_Fund	4,000	2075-07-06	[NULL]
9	4	Prabhat	M	2054-12-15	Nawalparasi	80,000	11		2 Divorced	4,444	4	Children_Savings_Fund	5,000	2079-11-15	[NULL]
10	5	Suman	M	2052-07-06	Sunwal	75,000	33		3 Divorced	5,555	5	General_Provident_Fund	3,500	2073-07-09	[NULL]

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

### Left Join:

```
SELECT * FROM Office o LEFT JOIN Project p
ON o.Onumber = p.Onumber;
```

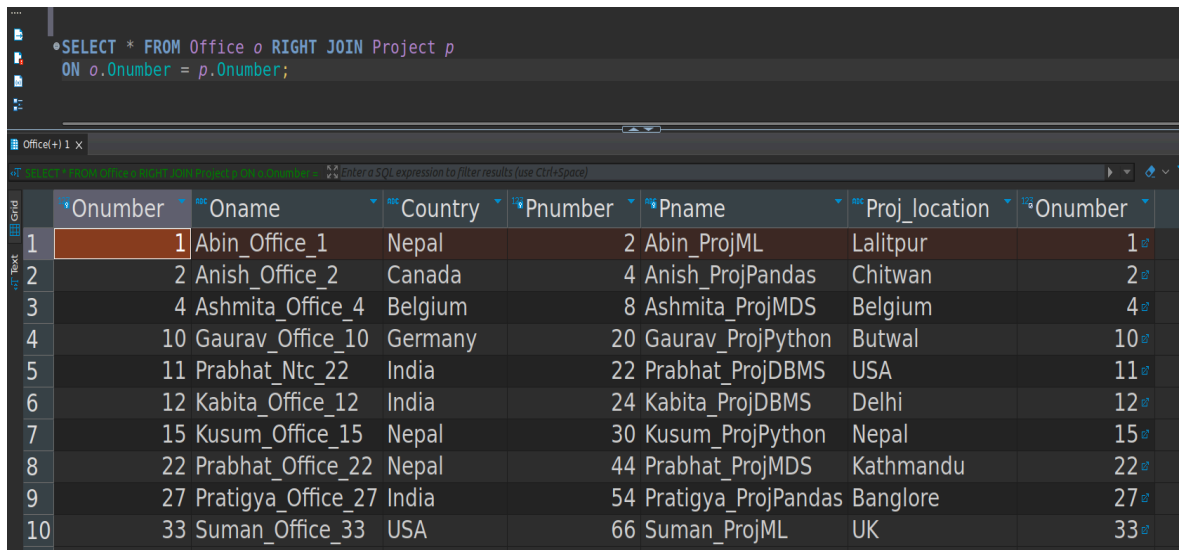


The screenshot shows a SQL query window with the query: `SELECT * FROM Office o LEFT JOIN Project p ON o.Onumber = p.Onumber;`. Below the query window is a grid displaying the results of the query. The grid has 8 columns: `Onumber`, `Oname`, `Country`, `Pnumber`, `Pname`, `Proj_location`, `Onumber`, and `Onumber`. The results show 10 rows of data, where the first 10 rows are from the Office table and the last 10 rows are from the Project table. The first 10 rows have `Onumber` values from 1 to 10, and the last 10 rows have `Onumber` values from 1 to 10. The `Onumber` column in the last 10 rows is highlighted in orange.

	Onumber	Oname	Country	Pnumber	Pname	Proj_location	Onumber	Onumber
1	1	Abin_Office_1	Nepal	2	Abin_ProjML	Lalitpur	1	1
2	2	Anish_Office_2	Canada	4	Anish_ProjPandas	Chitwan	2	2
3	4	Ashmita_Office_4	Belgium	8	Ashmita_ProjMDS	Belgium	4	4
4	10	Gaurav_Office_10	Germany	20	Gaurav_ProjPython	Butwal	10	10
5	11	Prabhat_Ntc_22	India	22	Prabhat_ProjDBMS	USA	11	11
6	12	Kabita_Office_12	India	24	Kabita_ProjDBMS	Delhi	12	12
7	15	Kusum_Office_15	Nepal	30	Kusum_ProjPython	Nepal	15	15
8	22	Prabhat_Office_22	Nepal	44	Prabhat_ProjMDS	Kathmandu	22	22
9	27	Pratigya_Office_27	India	54	Pratigya_ProjPandas	Banglore	27	27
10	33	Suman_Office_33	USA	66	Suman_ProjML	UK	33	33

### Right Join:

```
SELECT * FROM Office o RIGHT JOIN Project p
ON o.Onumber = p.Onumber;
```

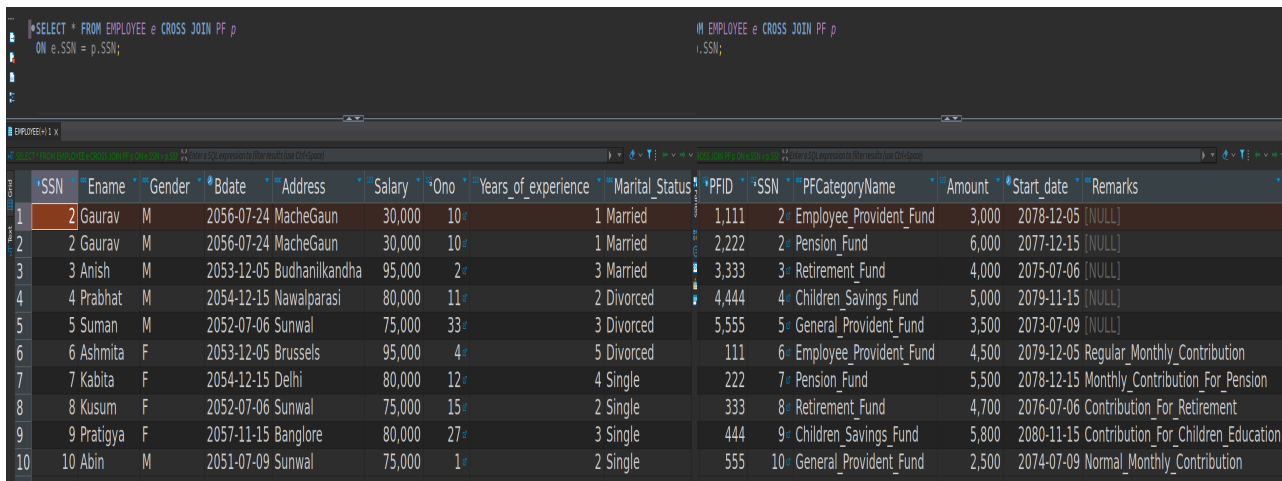


The screenshot shows a SQL query window with the query: `SELECT * FROM Office o RIGHT JOIN Project p ON o.Onumber = p.Onumber;`. Below the query window is a grid displaying the results of the query. The grid has 8 columns: `Onumber`, `Oname`, `Country`, `Pnumber`, `Pname`, `Proj_location`, `Onumber`, and `Onumber`. The results show 10 rows of data, where the first 10 rows are from the Project table and the last 10 rows are from the Office table. The first 10 rows have `Onumber` values from 1 to 10, and the last 10 rows have `Onumber` values from 1 to 10. The `Onumber` column in the last 10 rows is highlighted in orange.

	Onumber	Oname	Country	Pnumber	Pname	Proj_location	Onumber	Onumber
1	1	Abin_Office_1	Nepal	2	Abin_ProjML	Lalitpur	1	1
2	2	Anish_Office_2	Canada	4	Anish_ProjPandas	Chitwan	2	2
3	4	Ashmita_Office_4	Belgium	8	Ashmita_ProjMDS	Belgium	4	4
4	10	Gaurav_Office_10	Germany	20	Gaurav_ProjPython	Butwal	10	10
5	11	Prabhat_Ntc_22	India	22	Prabhat_ProjDBMS	USA	11	11
6	12	Kabita_Office_12	India	24	Kabita_ProjDBMS	Delhi	12	12
7	15	Kusum_Office_15	Nepal	30	Kusum_ProjPython	Nepal	15	15
8	22	Prabhat_Office_22	Nepal	44	Prabhat_ProjMDS	Kathmandu	22	22
9	27	Pratigya_Office_27	India	54	Pratigya_ProjPandas	Banglore	27	27
10	33	Suman_Office_33	USA	66	Suman_ProjML	UK	33	33

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

```
SELECT * FROM EMPLOYEE e CROSS JOIN PF p
ON e.SSN = p.SSN;
```



The screenshot shows a SQL query execution in a database client. The query is: `SELECT * FROM EMPLOYEE e CROSS JOIN PF p ON e.SSN = p.SSN;`. The results are displayed in a table with 10 rows and 15 columns. The columns are: SSN, Ename, Gender, Bdate, Address, Salary, Ono, Years\_of\_experience, Marital\_Status, PFID, SSN, PFCategoryName, Amount, Start\_date, and Remarks. The results show a cross join between the EMPLOYEE and PF tables, resulting in 10 rows for each employee, one for each PF record.

	SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_experience	Marital_Status	PFID	SSN	PFCategoryName	Amount	Start_date	Remarks
1	2	Gaurav	M	2056-07-24	MacheGaun	30,000	10		1 Married	1,111	2	Employee_Provident_Fund	3,000	2078-12-05	(NULL)
2	2	Gaurav	M	2056-07-24	MacheGaun	30,000	10		1 Married	2,222	2	Pension_Fund	6,000	2077-12-15	(NULL)
3	3	Anish	M	2053-12-05	Budhanilkandha	95,000	2		3 Married	3,333	3	Retirement_Fund	4,000	2075-07-06	(NULL)
4	4	Prabhat	M	2054-12-15	Nawalparasi	80,000	11		2 Divorced	4,444	4	Children_Savings_Fund	5,000	2079-11-15	(NULL)
5	5	Suman	M	2052-07-06	Sunwal	75,000	33		3 Divorced	5,555	5	General_Provident_Fund	3,500	2073-07-09	(NULL)
6	6	Ashmita	F	2053-12-05	Brussels	95,000	4		5 Divorced	111	6	Employee_Provident_Fund	4,500	2079-12-05	Regular_Monthly_Contribution
7	7	Kabita	F	2054-12-15	Delhi	80,000	12		4 Single	222	7	Pension_Fund	5,500	2078-12-15	Monthly_Contribution_For_Pension
8	8	Kusum	F	2052-07-06	Sunwal	75,000	15		2 Single	333	8	Retirement_Fund	4,700	2076-07-06	Contribution_For_Retirement
9	9	Pratigya	F	2057-11-15	Banglore	80,000	27		3 Single	444	9	Children_Savings_Fund	5,800	2080-11-15	Contribution_For_Children_Education
10	10	Abin	M	2051-07-09	Sunwal	75,000	1		2 Single	555	10	General_Provident_Fund	2,500	2074-07-09	Normal_Monthly_Contribution

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

```
SELECT * FROM EMPLOYEE e NATURAL JOIN PF;
```

SELECT \* FROM EMPLOYEE e NATURAL JOIN PF;

SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_experience	Marital_Status	PFID	PFCategoryName	Amount	Start_date	Remarks
2	Gaurav	M	2056-07-24	MacheGaun	30,000	10	1	Married	1,111	Employee_Provident_Fund	3,000	2078-12-05	[NULL]
2	Gaurav	M	2056-07-24	MacheGaun	30,000	10	1	Married	2,222	Pension_Fund	6,000	2077-12-15	[NULL]
3	Anish	M	2053-12-05	Budhanilkandha	95,000	2	3	Married	3,333	Retirement_Fund	4,000	2075-07-06	[NULL]
4	Prabhat	M	2054-12-15	Nawalparasi	80,000	11	2	Divorced	4,444	Children_Savings_Fund	5,000	2079-11-15	[NULL]
5	Suman	M	2052-07-06	Sunwal	75,000	33	3	Divorced	5,555	General_Provident_Fund	3,500	2073-07-09	[NULL]
6	Ashmita	F	2053-12-05	Brussels	95,000	4	5	Divorced	111	Employee_Provident_Fund	4,500	2079-12-05	Regular_Monthly_Contribution
7	Kabita	F	2054-12-15	Delhi	80,000	12	4	Single	222	Pension_Fund	5,500	2078-12-15	Monthly_Contribution_For_Pension
8	Kusum	F	2052-07-06	Sunwal	75,000	15	2	Single	333	Retirement_Fund	4,700	2076-07-06	Contribution_For_Retirement
9	Pratigya	F	2057-11-15	Bangalore	80,000	27	3	Single	444	Children_Savings_Fund	5,800	2080-11-15	Contribution_For_Children_Education
10	Abin	M	2051-07-09	Sunwal	75,000	1	2	Single	555	General_Provident_Fund	2,500	2074-07-09	Normal_Monthly_Contribution

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

```
SELECT COUNT(*) as number_of_employees, Marital_Status FROM EMPLOYEE
GROUP BY Marital_Status;
```

	number_of_employees	Marital_Status
1	2	Married
2	3	Divorced
3	4	Single

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 COUNT(*) as number_of_employees, Marital_Status FROM EMPLOYEE e
GROUP BY Marital_Status HAVING
Marital_Status IN ('Married', 'Single');
```

	number_of_employees	Marital_Status
1	2	Married
2	4	Single

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

```
SELECT Pname, Proj_location
FROM Project p
WHERE p.Onumber IN (SELECT o.Onumber FROM Office o WHERE o.Country IN ('Nepal', 'India'));
```

Project 1 x

SELECT Pname, Proj\_location FROM Project p WHERE p.Onumber IN (SELECT o.Onumber FROM Office o WHERE o.Country IN ('Nepal', 'India'))

	Pname	Proj_location
1	Abin_ProjML	Lalitpur
2	Prabhat_ProjDBMS	USA
3	Kabita_ProjDBMS	Delhi
4	Kusum_ProjPython	Nepal
5	Prabhat_ProjMDS	Kathmandu
6	Pratigya_ProjPandas	Bangalore