- **Joins**

1. Combine information FROM student and result table using cross join or Cartesian product.

```
SELECT *
FROM Student
CROSS JOIN Result
OR
SELECT *
FROM Student, Result
```

2. Display Rno, Name, Branch and SPI of CE branch's student only.

```
SELECT
        Student.Rno,
        Student.Name,
        Student.Branch,
        Result.SPI
FROM Student
LEFT OUTER JOIN Result
ON Student.Rno = Result.Rno
WHERE Student.Branch='CE'
```

3. Display Rno, Name, Branch and SPI of other than EC branch's student only.

```
SELECT
        Student.Rno,
        Student.Name,
        Student.Branch,
        Result.SPI
FROM Student
LEFT OUTER JOIN Result
ON Student.Rno = Result.Rno
WHERE Student.Branch <> 'EC'
```

4. Display the average result of each branch.

```
SELECT
        Student.Branch,
        AVG(Result.SPI) As Avg_Spi
FROM Student
INNER JOIN Result
ON Student.Rno = Result.Rno
GROUP BY Student.Branch
```

5. Display the average result of each branch and sort them in ascending ORDER BY SPI.

```
SELECT
        Student.Branch,
        AVG(Result.SPI) As Avg_Spi
```

```
FROM Student
INNER JOIN Result
ON Student.Rno = Result.Rno
GROUP BY Student.Branch
ORDER BY AVG(Result.SPI)
```

6. Display average result of CE and ME branch.

```
SELECT
        Student.Branch,
        AVG(Result.SPI) As Avg_Spi
FROM Student
INNER JOIN Result
ON Student.Rno = Result.Rno
AND (Student.Branch IN ('CE','ME’))
GROUP BY Student.Branch
```

7. Perform the left outer join on Student and Result tables.

```
SELECT
        Student.Rno,
        Student.Name,
        Student.Branch,
        Result.SPI
FROM Student
LEFT OUTER JOIN Result
ON Student.Rno = Result.Rno
```

8. Perform the right outer join on Student and Result tables.

```
SELECT
        Student.Rno,
        Student.Name,
        Student.Branch,
        Result.SPI
FROM Student
RIGHT OUTER JOIN Result
ON Student.Rno = Result.Rno
```

9. Perform the full outer join on Student and Result tables.

```
SELECT
        Student.Rno,
        Student.Name,
        Student.Branch,
        Result.SPI
FROM Student
FULL OUTER JOIN Result
```

```
ON Student.Rno = Result.Rno
```

10. Retrieve the names of employees along with their manager’s name FROM the Employee table.

```
SELECT
        E.Name As EmployeeName,
        M.Name As ManagerName
FROM Employee E
LEFT OUTER JOIN Employee M
ON E.ManagerNo = M.EmployeeNo
```

---

1. Display all the villages of Rajkot city.

```
FROM V.VillageName
FROM City C
FULL OUTER JOIN Village V
ON C.CityID = V.CityID
WHERE C.CityName='Rajkot'
```

2. Display city along with their villages & pin code.

```
SELECT
        C.CityName,
        C.Pincode,
        V.VillageName
FROM City C
RIGHT OUTER JOIN Village V
ON C.CityID = V.CityID
```

3. Display the city having more than one village.

```
SELECT
        City.CityName,
        COUNT(Village.CityID) AS NoOfVillages
FROM City
LEFT OUTER JOIN Village
ON City.CityID=Village.CityID
GROUP BY City.CityName
HAVING COUNT(Village.CityID)>1
```

4. Display the city having no village.

```
SELECT
        City.CityName
FROM City
LEFT OUTER JOIN Village
ON City.CityID=Village.CityID
```

```
GROUP BY City.CityName
HAVING COUNT(Village.CityID) = 0
```

5. Count the total number of villages in each city.

```
SELECT
        City.CityName,
        COUNT(Village.VillageID) As TotalVillages
FROM City
LEFT OUTER JOIN Village
ON City.CityID=Village.CityID
GROUP BY City.CityName
```

6. Count the number of cities having more than one village.

```
SELECT COUNT(*)
FROM
(
        SELECT City.[CityName],
                COUNT(Village.VillageID) AS TOTAL
        FROM City
        LEFT OUTER JOIN VillageID
        ON City.CityID = VillageID.CityID
        GROUP BY City. [CityName]
) AS T
WHERE TOTAL > 1
```