Working with Joins

```sql
/*
Joins
        - Inner Join (Join) / Natural Join
  - Left Join / Left Outer Join  : selects all the records from left table
  - Right Join / Right Outer Join : selects all the records from right table
  - Full Outer Join** : union combination of left and right
  - Self Join**
*/
/*
Q1. Display all students that are living in given city.
projection: student
criteria: address
*/
select *
from student s join address a ON s.address_id = a.id
where a.city='mumbai';

/*
Q2. Display number of students that are living in each state.
projection: students
criteria: address
*/
select a.state, count(s.id) as number_of_students
from student s join address a ON s.address_id=a.id
group by a.state
order by number_of_students desc;

/*
Q3. Display courses that belong to given department
projection:courses
```

```
criteria:department
*/
select d.name, c.*
from course c join department d on c.department_id = d.id
where d.name = 'IT';

/*
Q4. Display number of courses for each department.
projection: courses
criteria: department
*/
select d.name, count(distinct c.id)
from course c
        join department d ON c.department_id = d.id
group by d.name;

/*
Q5. Display students that have enrolled in given course.
projection: student
criteria: course
*/
select s.*, c.name as course_name
from student s
        join student_course sc on s.id = sc.student_id
        join course c on sc.course_id = c.id
where c.name='java programming';

/*
Q6. Display students associated with given department.
 projection: student
 criteria: department
```

```sql
*/
select distinct s.*
from student s
        JOIN student_course sc ON s.id= sc.student_id
        JOIn course c ON sc.course_id = c.id
        JOIN department d ON c.department_id = d.id
where d.name = 'dev';

/*
Q7. Display number of students associated with each department.
display records having more than 1 student in Ascending order
project: students
criteria: department
*/
select d.name, count(distinct s.id) as number_of_students
from student s
   JOIN student_course sc ON s.id = sc.student_id
   JOIN course c ON sc.course_id = c.id
   JOIN department d ON c.department_id = d.id
group by d.name
having number_of_students > 1
order by number_of_students ASC;

/*
Q8. Display students that have enrolled before given date
projection: student
criteria: student_course
*/
select distinct s.name
from student s JOIN student_course sc ON s.id=sc.student_id
where sc.date_of_enrollment <'2024-02-05';
```

```sql
/*
Q9. Display courses for which the discount of more than 5% is given.
projection: course
criteria: student_course
*/
select distinct c.name
from course c
        JOIN  student_course sc ON c.id = sc.course_id
where sc.discount >'5';
/*
Q10. Display avg discount given for each course
projection: student_course
criteria: course
*/
select c.name,avg(sc.discount)
from course c join student_course sc on c.id=sc.course_id
group by c.name;

/*
Q11. Display avg discount given to each student
projection: student_course
criteria: student
*/
select s.name,avg(sc.discount)
from student s join student_course sc on s.id = sc.student_id
group by s.name;

/* Display all students along with number of courses enrolled */

-- display students who have enrolled in atleast 1 course.
```

```
select s.name, count(sc.course_id)
from student s left join student_course sc ON s.id = sc.student_id
group by s.name;

-- left join
select *
from product p left join vendor v ON p.vendor_id = v.id;
/*
+----+----------------+-----------+------+-----------------+---------+
| id | name           | vendor_id | id   | name            | city    |
+----+----------------+-----------+------+-----------------+---------+
|  1 | Samsumg mobile |         1 |    1 | ABC Electronics | mumbai  |
|  2 | apple mobile   |         2 |    2 | XYZ electronics | chennai |
|  3 | oppo mobile    |      NULL | NULL | NULL            | NULL    |
+----+----------------+-----------+------+-----------------+---------+
*/

select *
from product p right join vendor v ON p.vendor_id = v.id;
/*
+------+----------------+-----------+----+-----------------+---------+
| id   | name           | vendor_id | id | name            | city    |
+------+----------------+-----------+----+-----------------+---------+
|    1 | Samsumg mobile |         1 |  1 | ABC Electronics | mumbai  |
|    2 | apple mobile   |         2 |  2 | XYZ electronics | chennai |
| NULL | NULL           |      NULL |  3 | PQR Electronics | pune    |
+------+----------------+-----------+----+-----------------+---------+
*/
-- full join
(select *
from product p left join vendor v ON p.vendor_id = v.id )
```

```
union
(select *
from product p right join vendor v ON p.vendor_id = v.id);

/*
+------+----------------+-----------+------+-----------------+---------+
| id   | name           | vendor_id | id   | name            | city    |
+------+----------------+-----------+------+-----------------+---------+
|    1 | Samsumg mobile |         1 |    1 | ABC Electronics | mumbai  |
|    2 | apple mobile   |         2 |    2 | XYZ electronics | chennai |
|    3 | oppo mobile    |      NULL | NULL | NULL            | NULL    |
| NULL | NULL           |      NULL |    3 | PQR Electronics | pune    |
+------+----------------+-----------+------+-----------------+---------+
*/
```