11 Set Operators - Subqueries - Joins

1. Set Operators

Set operators in SQL are used to combine the results of two or more queries. Common set operators include:

- **UNION**: Combines the result sets of two queries and removes duplicates
- **UNION ALL**: Combines the result sets of two queries without removing duplicates
- **INTERSECT**: Returns only the rows that are present in both result sets
- **EXCEPT** (or **MINUS** in some databases): Returns the rows from the first query that are not present in the second query

```
SELECT student_id, student_name FROM Students_A
UNION
SELECT student_id, student_name FROM Students_B;
```

2. Nested Queries (Subqueries)

- A nested query, or subquery, is a query within another SQL query
- Subqueries can be used in SELECT, INSERT, UPDATE, or DELETE statements and are often used to filter results based on the result of another query

```
-- Find students who have scored above average in the exam
SELECT student_name
FROM Students
WHERE marks >
    (SELECT AVG(marks) FROM Students);
```

3. Joins

Joins are used to retrieve data from two or more tables based on a related column between them. Common types of joins include:

- **INNER JOIN**: Returns only the rows that have matching values in both tables.
- **LEFT JOIN (or LEFT OUTER JOIN)**: Returns all rows from the left table, and the matched rows from the right table. If there's no match, NULLs are returned.
- **RIGHT JOIN (or RIGHT OUTER JOIN)**: Returns all rows from the right table, and the matched rows from the left table. If there's no match, NULLs are returned.

- **FULL JOIN (or FULL OUTER JOIN)**: Returns rows when there is a match in one of the tables. This means it returns all rows from the left table and right table and fills in NULLs for non-matching rows.
- **CROSS JOIN**: Returns the Cartesian product of the two tables, i.e., every row of the first table is combined with every row of the second table.

Example:

```
-- Find the names of students and the courses they are enrolled in

SELECT Students.student_name, Courses.course_name
FROM Students
INNER JOIN Enrollments ON Students.student_id = Enrollments.student_id
INNER JOIN Courses ON Enrollments.course_id = Courses.course_id;
```