

SQL Joins and Set Operations

Types of SQL Joins

1. Inner Join

Description: Returns records that have matching values in both tables.

Syntax:

```
SELECT columns  
FROM table1  
INNER JOIN table2  
ON table1.common_column = table2.common_column;
```

2. Left (Outer) Join

Description: Returns all records from the left table, and the matched records from the right table. If no match is found, NULL values are returned for columns from the right table.

Syntax:

```
SELECT columns  
FROM table1  
LEFT JOIN table2  
ON table1.common_column = table2.common_column;
```

3. Right (Outer) Join

Description: Returns all records from the right table, and the matched records from the left table. If no match is found, NULL values are returned for columns from the left table.

Syntax:

```
SELECT columns  
FROM table1  
RIGHT JOIN table2  
ON table1.common_column = table2.common_column;
```

4. Full (Outer) Join

Description: Returns all records when there is a match in either left or right table. If no match is found, NULL values are returned for columns from the table without a match.

Syntax:

```
SELECT columns  
FROM table1  
FULL OUTER JOIN table2  
ON table1.common_column = table2.common_column;
```

5. Cross Join

Description: Returns the Cartesian product of both tables, i.e., every row in the first table is paired with all rows in the second table.

Syntax:

```
SELECT columns  
FROM table1  
CROSS JOIN table2;
```

6. Self Join

Description: A join in which a table is joined with itself.

Syntax:

```
SELECT a.columns, b.columns  
FROM table a, table b  
WHERE condition;
```

Set Operations

1. UNION

Description: Combines the result set of two or more SELECT statements. It removes duplicate rows between the various SELECT statements.

Syntax:

```
SELECT columns FROM table1  
UNION  
SELECT columns FROM table2;
```

2. UNION ALL

Description: Similar to UNION, but it includes duplicate rows.

Syntax:

```
SELECT columns FROM table1  
UNION ALL  
SELECT columns FROM table2;
```

3. INTERSECT

Description: Returns only the rows that are common to both SELECT statements.

Syntax:

```
SELECT columns FROM table1  
INTERSECT  
SELECT columns FROM table2;
```

4. EXCEPT (or MINUS)

Description: Returns the rows from the first SELECT statement that are not present in the second SELECT statement.

Syntax:

```
SELECT columns FROM table1  
EXCEPT  
SELECT columns FROM table2;
```

Note: The MINUS keyword is used in some databases (like Oracle), while EXCEPT is used in others (like SQL Server).