Let’s go **in-depth on SQL Set Operators** with clear explanations and examples. Set operators are used to **combine results from two or more queries**.

**1. UNION – Combine Results, Remove Duplicates**

* **Purpose:** Combines result sets of two queries and **removes duplicates**.
* **Rules:**
  1. Same number of columns in both queries.
  2. Corresponding columns have compatible data types.

**Syntax:**

SELECT column1, column2 FROM table1

UNION

SELECT column1, column2 FROM table2;

**Example:**

SELECT Name FROM Employees

UNION

SELECT Name FROM Customers;

* **Output:** List of all distinct names from Employees and Customers.

**2. UNION ALL – Combine Results, Keep Duplicates**

* **Purpose:** Like UNION, but **includes duplicates**.
* **Syntax:**

SELECT column1 FROM table1

UNION ALL

SELECT column1 FROM table2;

**Example:**

SELECT Name FROM Employees

UNION ALL

SELECT Name FROM Customers;

* **Output:** List of all names, including repeated names.

**3. INTERSECT – Common Rows Only**

* **Purpose:** Returns only rows **present in both queries**.
* **Syntax:**

SELECT column1 FROM table1

INTERSECT

SELECT column1 FROM table2;

**Example:**

SELECT Name FROM Employees

INTERSECT

SELECT Name FROM Customers;

* **Output:** Names that exist in both Employees and Customers.

**Note:** Supported in Oracle, SQL Server, PostgreSQL, but **not MySQL** (use INNER JOIN as workaround).

**4. EXCEPT / MINUS – Difference Between Sets**

* **Purpose:** Returns rows from the first query **that are not in the second query**.

**Syntax (SQL Server / PostgreSQL):**

SELECT column1 FROM table1

EXCEPT

SELECT column1 FROM table2;

**Syntax (Oracle):**

SELECT column1 FROM table1

MINUS

SELECT column1 FROM table2;

**Example:**

-- Employees who are NOT customers

SELECT Name FROM Employees

EXCEPT

SELECT Name FROM Customers;

**Output:** List of employee names not found in Customers table.

**5. Key Points for Set Operators**

| **Operator** | **Removes Duplicates** | **Notes** |
| --- | --- | --- |
| UNION | Yes | Standard for combining queries |
| UNION ALL | No | Faster than UNION |
| INTERSECT | Yes | Returns common rows only |
| EXCEPT/MINUS | Yes | Returns rows only in first query |

**Rules for all set operators:**

1. Queries must have **same number of columns**.
2. Data types of corresponding columns must be **compatible**.
3. Order of columns matters.

**6. Example with Multiple Operators**

-- Employees or Customers, but not both

SELECT Name FROM Employees

UNION

SELECT Name FROM Customers

EXCEPT

SELECT Name FROM Employees

INTERSECT

SELECT Name FROM Customers;

* Combines results in multiple steps using UNION, INTERSECT, and EXCEPT.

**7. ORDER BY with Set Operators**

* **Important:** ORDER BY is applied **after all set operations**, usually at the end.

SELECT Name FROM Employees

UNION

SELECT Name FROM Customers

ORDER BY Name;

**✅ Summary Table**

| **Operator** | **Description** | **Duplicate Rows** | **Example Use Case** |
| --- | --- | --- | --- |
| UNION | Combine and remove duplicates | No | Merge two tables' lists |
| UNION ALL | Combine including duplicates | Yes | Merge lists, keep counts |
| INTERSECT | Only common rows | No | Find overlapping data |
| EXCEPT/MINUS | Rows in first query but not second | No | Find non-matching data |