UNION OPERATOR

SQL UNION operator combines result sets of two or more SELECT statements into one results set.

UNION Operator in SQL

The **UNION operator in SQL** is used to combine the result set of multiple SELECT statements and return one result set.

There are some rules for using the SQL UNION operator.

Rules for SQL UNION

- Each table used within UNION must have the same number of columns.
- The columns must have the same data types.
- The columns in each table must be in the same order.

Syntax

The Syntax of the SQL UNION operator is:

```
SELECT columnnames FROM table1
UNION
SELECT columnnames FROM table2;
```

UNION operator provides unique values by default. To find duplicate values, use UNION ALL.

UNION ALL Syntax

The UNION ALL syntax is:

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

Note: SQL UNION and UNION ALL difference is that UNION operator removes duplicate rows from results set and UNION ALL operator retains all rows, including duplicate.

SQL UNION Example

Let’s look at an example of UNION operator in SQL to understand it better.

Let’s create two tables "Emp1" and "Emp2".

Emp1 Table

Write the following SQL query to create Emp1 table.

```
CREATE TABLE Emp1
(
    EmpID INT PRIMARY KEY,
    Name VARCHAR(50),
    Country VARCHAR(50),
    Age int(2),
    mob int(10)
);
-- Insert some sample data into the Customers table
INSERT INTO Emp1 (EmpID, Name,Country, Age, mob)
VALUES (1, 'Shubham',  'India','23','738479734'),
       (2, 'Aman ',  'Australia','21','436789555'),
       (3, 'Naveen', 'Sri lanka','24','34873847'),
       (4, 'Aditya',  'Austria','21','328440934'),
       (5, 'Nishant', 'Spain','22','73248679');
SELECT * FROM Emp1;
```

Output

EmpID	Name	Country	Age	mob
1	Shubham	India	23	738479734
2	Aman	Australia	21	436789555
3	Naveen	Sri lanka	24	34873847
4	Aditya	Austria	21	328440934
5	Nishant	Spain	22	73248679

Emp1 Table

Emp2 Table

Write the following SQL query to create Emp2 table.

```
CREATE TABLE Emp2
(
    EmpID INT PRIMARY KEY,
    Name VARCHAR(50),
    Country VARCHAR(50),
    Age int(2),
    mob int(10)
);
-- Insert some sample data into the Customers table
INSERT INTO Emp2 (EmpID, Name,Country, Age, mob)
VALUES (1, 'Tommy',  'England','23','738985734'),
       (2, 'Allen',  'France','21','43678055'),
       (3, 'Nancy', 'India','24','34873847'),
       (4, 'Adi',  'Ireland','21','320254934'),
```

```
        (5, 'Sandy', 'Spain','22','70248679');
SELECT * FROM Emp2;
```

Output

EmpID	Name	Country	Age	mob
1	Tommy	England	23	738985734
2	Allen	France	21	43678055
3	Nancy	India	24	34873847
4	Adi	Ireland	21	320254934
5	Sandy	Spain	22	70248679

Emp2 Table

SQL UNION Operator Example

In this example, we will find the cities (only unique values) from both the “Table1” and the “Table2” tables:

Query

```
SELECT Country FROM Emp1
UNION
SELECT Country FROM Emp2
ORDER BY Country;
```

Output

Country
Australia
Austria
England
France
India
Ireland
Spain
Sri lanka

Output

SQL UNION ALL Example

In the below example, we will find the cities (duplicate values also) from both the “Emp1” and the “Emp2” tables:

Query

```
SELECT Country FROM Emp1
UNION ALL
SELECT Country FROM Emp2
ORDER BY Country;
```

Output

Country
Australia
Austria
England
France
India
Ireland
Spain
Sri lanka

Output

SQL UNION ALL With WHERE

You can use the **WHERE clause** with UNION ALL in SQL. The WHERE clause is used to filter records and is added after each SELECT statement.

SQL UNION ALL with WHERE Example

The following SQL statement returns the cities (duplicate values also) from both the “Geeks1” and the “Geeks2” tables:

Query

```
SELECT Country, Name FROM Emp1
WHERE Name='Aditya'
UNION ALL
SELECT Country, Name FROM Emp2
WHERE Country='Ireland'
ORDER BY Country;
```

Country	Name
Austria	Aditya
Ireland	Adi

Output

Important Points About SQL UNION Operator

- The SQL UNION operator combines the result sets of two or more SELECT queries.
- UNION returns unique rows, eliminating duplicate entries from the result set.
- UNION ALL includes all rows, including duplicate rows.
- Columns in the result set must be in the same order and have the same data types.
- UNION is useful for aggregating data from multiple tables or applying different filters to data from the same table.