

SQL UNION and UNION ALL

The `UNION` operator in SQL is used to **combine the result sets of two or more `SELECT` statements**. It removes duplicates by default.

If you want to include all rows including duplicates, use `UNION ALL`.

Example Scenario

You already have a `users` table for active users. Now, we'll create an `admin_users` table to store users who are administrators or have special roles. We will then combine the names from both tables using `UNION`.

Step 1: Create the `admin_users` Table

```
CREATE TABLE admin_users (  
  id INT PRIMARY KEY,  
  name VARCHAR(100),  
  email VARCHAR(100),  
  gender ENUM('Male', 'Female', 'Other'),  
  date_of_birth DATE,  
  salary INT  
);
```

Step 2: Insert Sample Data into `admin_users`

```
INSERT INTO admin_users (id, name, email, gender, date_of_birth, salary) VALUES
(101, 'Anil Kumar', 'anil@example.com', 'Male', '1985-04-12', 60000),
(102, 'Pooja Sharma', 'pooja@example.com', 'Female', '1992-09-20', 58000),
(103, 'Rakesh Yadav', 'rakesh@example.com', 'Male', '1989-11-05', 54000),
(104, 'Fatima Begum', 'fatima@example.com', 'Female', '1990-06-30', 62000);
```

Step 3: Use `UNION` to Combine Data

Let's combine the active and admin user names.

```
SELECT name FROM users
UNION
SELECT name FROM admin_users;
```

This returns a **single list of unique names** from both tables.

`UNION ALL` Example

If you want to keep duplicate names (if any), use `UNION ALL` .

```
SELECT name FROM users
UNION ALL
SELECT name FROM admin_users;
```

Using More Than One Column

You can also select multiple columns as long as both `SELECT` queries return the same number of columns with compatible types.

```
SELECT name, salary FROM users
UNION
SELECT name, salary FROM admin_users;
```

Adding separate roles

```
SELECT name, 'User' AS role FROM users
UNION
SELECT name, 'Admin' AS role FROM admin_users;
```

Using Order By with UNION

```
SELECT name FROM users
UNION
SELECT name FROM admin_users
ORDER BY name;
```

Rules of UNION

1. The number of columns and their data types must match in all `SELECT` statements.
 2. `UNION` removes duplicates by default.
 3. `UNION ALL` keeps duplicates.
-

When to Use UNION

- When you have **two similar tables** (like current and archived data).
 - When you need to **combine filtered results** (e.g., high-salary users from two sources).
 - When performing **cross-category reporting**.
-

Summary

Operator	Behavior
UNION	Combines results, removes duplicates
UNION ALL	Combines results, keeps duplicates