#56daysofSQL

Day 15
SET Operators - (Part 1)
UNION and UNION ALL

Aditya Shenoy

#56daysofSQL

UNION Operator

- MySQL **UNION** operator allows you to combine two or more result sets of queries into a single result set.
- To combine result set of two or more queries using the **UNION** operator, these are the basic rules that you must follow:
  - **First**, the number and the orders of columns that appear in all SELECT statements must be the same.
  - **Second**, the data types of columns must be the same or compatible

Aditya Shenoy

#56daysofSQL

General Syntax :

```
SELECT column_list
UNION [DISTINCT | ALL]
SELECT column_list
UNION [DISTINCT | ALL]
SELECT column_list
...
```

By default, the **UNION** operator removes duplicate rows even if you don't specify the **DISTINCT** operator explicitly.

Use Case Example :

Query :

```
SELECT
    firstName,
    lastName
FROM
    employees
UNION
SELECT
    contactFirstName,
    contactLastName
FROM
    customers;
```

Output :

	firstName	lastName
	Jean	King
	Peter	Ferguson
	Janine	Labrune
	Jonas	Bergulfsen
	Susan	Nelson
	Zbyszek	Piestrzeniew
	Roland	Keitel
	Julie	Murphy

#56daysofSQL

UNION ALL

- If you use **UNION ALL** explicitly, the duplicate rows, if available, remain in the result.
- Because **UNION ALL** does not need to handle duplicates, it performs faster than **UNION DISTINCT**.

Aditya Shenoy

#56daysofSQL

UNION vs UNION ALL :

Creating Sample Tables :

```sql
CREATE TABLE t1 (
    id INT PRIMARY KEY
);

CREATE TABLE t2 (
    id INT PRIMARY KEY
);

INSERT INTO t1 VALUES (1),(2),(3);
INSERT INTO t2 VALUES (2),(3),(4);
```

Query (UNION) :

```sql
SELECT id
FROM t1
UNION
SELECT id
FROM t2;
```

Query (UNION ALL) :

```sql
SELECT id
FROM t1
UNION ALL
SELECT id
FROM t2;
```

Output

UNION :

id
1
2
3
4

Returns the **DISTINCT** values from both tables.

UNION ALL :

id
1
2
3
2
3
4

Also returns the **duplicate** rows, if available.