NOT EQUAL OPERATORS

SQL NOT EQUAL Operator is a comparison operator used to check whether two expressions are equal. This operator is represented by "**!=**" or "**<>**".

NOT EQUAL Operator in SQL

NOT EQUAL Operator in SQL is used to compare two values and return if they are not equal.

This operator returns boolean values. If given expressions are equal, the operator returns **false** otherwise **true**. If any one expression is NULL, it will return NULL.

It performs type conversion when expressions are of different data types, for example, 5!= “Five”.

We use the NOT EQUAL operator to display our table without some exceptional values.

For example, Let’s, consider a table "**Students**". For this table, we have, "**id**", "**name**", and "**marks**" as its columns. Now we want to display all those rows that have marks not equal to “100”. In this kind of situation, the NOT EQUAL operator can be used.

> **Note**: <> and != perform the same operation i.e. check inequality. The only difference between <> and != is that <> follows the **ISO standard** but != does not. So it is recommended to use <> for NOT EQUAL Operator.>

Syntax

The SQL NOT EQUAL Operator syntax is:

```
SELECT * FROM table_name
WHERE column_name != value;
```

NOT EQUAL Operator Examples

Let’s look at some examples of the NOT EQUAL Operator in SQL, and understand its working.

First, we will create a demo SQL database and table on which we will use the NOT EQUAL operator.

Demo SQL Database

In this tutorial on the NOT EQUAL operator, we will use the below table for examples and see the NOT EQUAL operator in SQL query.

user_id	name	contest_score	rank	coding_streak
vish3001	Vishu	100	1	150

neeraj119	Neeraj	99	2	125
ayush105	Aayush	98	3	110
sumit85	Sumit	99	2	100
harsh05	Harsh	98	3	95

To create this table in your system, write the following SQL queries in your DBMS;

```
CREATE TABLE coder(
    user_id varchar(100) PRIMARY KEY,
    name varchar(100),
    contest_score int,
    rank int,
    coding_streak int
 );
INSERT INTO coder(user_id,name,contest_score,rank,coding_streak)
VALUES('vish3001','Vishu',100,01,150);
INSERT INTO coder(user_id,name,contest_score,rank,coding_streak)
VALUES('neeraj119','Neeraj',99,02,125);
INSERT INTO coder(user_id,name,contest_score,rank,coding_streak)
VALUES('ayush105','Aayush',98,03,110);
INSERT INTO coder(user_id,name,contest_score,rank,coding_streak)
VALUES('sumit85','Sumit',99,02,100);
INSERT INTO coder(user_id,name,contest_score,rank,coding_streak)
VALUES('harsh05','Harsh',98,03,95);
```

SQL NOT EQUAL operator For String Example

In this example, we will display all those rows which do not have a name equal to 'Harsh'. We will use NOT EQUAL with **WHERE clause** in this case.

Query

```
SELECT * FROM coder WHERE name!='Harsh';
```

Output

No description has been provided for this image

SQL NOT operator For String

In the above image, we can see we have all those rows displayed which do not have their name equal to 'Harsh'.

Note: The NOT EQUAL comparison is **case-sensitive for strings**. Meaning "name" and "NAME" are two different strings for NOT EQUAL operator.

SQL NOT EQUAL Operator with Multiple Condition

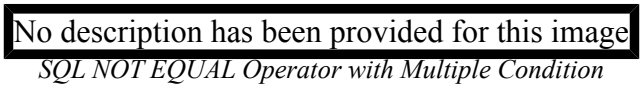
Example

In this example, we will display all those rows which do not have their contest score as 98 and rank as 3 and the coding streak should be greater than or equal to 100. Using AND or OR operator you can use the SQL NOT operator for multiple values.

Query

```
SELECT * FROM coder
WHERE contest_score != 98 AND rank != 3
AND coding_streak >= 100;
```

Output



SQL NOT EQUAL Operator with Multiple Condition

In the above image, we can observe that all those rows are displayed which have followed all three conditions.

SQL NOT EQUAL Operator with GROUP BY Clause

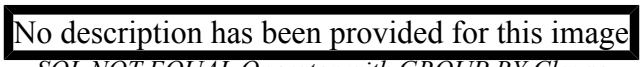
Example

In this example, we will display all those ranks with their count that do not have their contest score as 100 using **GROUP BY** clause.

Query

```
SELECT rank, COUNT(*) AS count_score
FROM coder
WHERE contest_score <> 100
GROUP BY rank;
```

Output



SQL NOT EQUAL Operator with GROUP BY Clause

In the above image, we can see ranks 2, and 3 have a count of 2 and, 2 respectively.

Important Points About SQL NOT EQUAL Operator

- SQL NOT EQUAL Operator is a comparison operator denoted as != or <>. It returns boolean values i.e. True or False.
- It returns False when the compared expressions are equal otherwise it returns True.
- We use this operator with the WHERE clause.
- We can use this operator for integers and strings-based logical reasoning. It is case-sensitive for string comparisons.
- We can put multiple conditions using the AND or OR operator.