SQLite Select Distinct

Summary: in this tutorial, you will learn how to use the SQLite `SELECT DISTINCT` clause to remove duplicate rows in the result set.

Introduction to SQLite SELECT DISTINCT clause

The `DISTINCT` clause is an optional clause of the `SELECT` (https://www.sqlitetutorial.net/sqlite-select/) statement. The `DISTINCT` clause allows you to remove the duplicate rows in the result set.

The following statement illustrates the syntax of the `DISTINCT` clause:

```
SELECT DISTINCT select_list
FROM table;
```

In this syntax:

- First, the `DISTINCT` clause must appear immediately after the `SELECT` keyword.
- Second, you place a column or a list of columns after the `DISTINCT` keyword. If you use one column, SQLite uses values in that column to evaluate the duplicate. In case you use multiple columns, SQLite uses the combination of values in these columns to evaluate the duplicate.

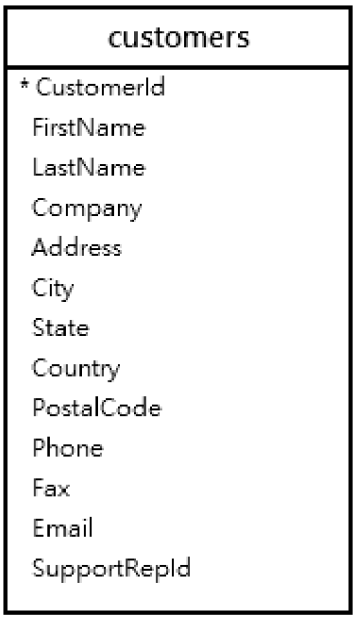
SQLite considers `NULL` values as duplicates. If you use the `DISTINCT` clause with a column that has `NULL` values, SQLite will keep one row of a `NULL` value.

> In database theory, if a column contains `NULL` values, it means that we do not have the information about that column of particular records or the information is not applicable.

For example, if a customer has a phone number with a `NULL` value, it means we don't have information about the phone number of the customer at the time of recording customer information or the customer may not have a phone number at all.

SQLite SELECT DISTINCT examples

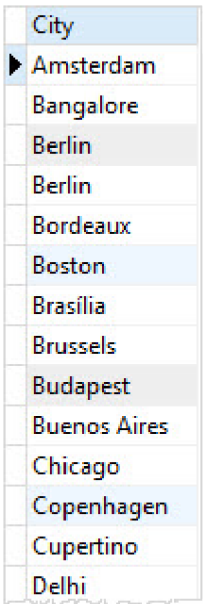
We will use the `customers` table in the sample database (https://www.sqlitetutorial.net/sqlite-sample-database/) for demonstration.



Suppose you want to know the cities where the customers locate, you can use the `SELECT` statement to get data from the `city` column of the `customers` table as follows:

```
SELECT city
FROM customers
ORDER BY city;
```

Try It ›



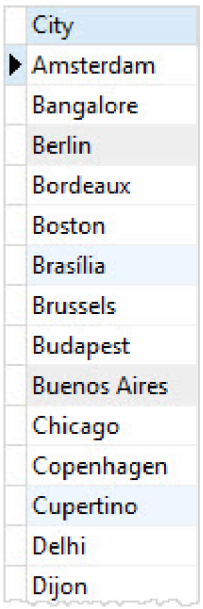
(https://www.sqlitetutorial.net/wp-content/uploads/2015/11/SQLite-without-DISTINCT.jpg)

It returns 59 rows. There are few duplicate rows such as `Berlin` `London` , and `Mountain View` To remove these duplicate rows, you use the `DISTINCT` clause as follows:

```
SELECT DISTINCT city
FROM customers
ORDER BY city;
```

Try It ›

City
Amsterdam
Bangalore
Berlin
Bordeaux
Boston
Brasília
Brussels
Budapest
Buenos Aires
Chicago
Copenhagen
Cupertino
Delhi
Dijon


(https://www.sqlitetutorial.net/wp-content/uploads/2015/11/SQLite-DISTINCT-example.jpg)

It returns 53 rows because the `DISTINCT` clause has removed 6 duplicate rows.

SQLite SELECT DISTINCT on multiple columns

The following statement finds cities and countries of all customers.

```
SELECT
        city,
        country
FROM
        customers
ORDER BY
        country;
```

Try It ›

City	Country
Buenos Aires	Argentina
Sidney	Australia
Vienne	Austria
Brussels	Belgium
São José dos Campos	Brazil
São Paulo	Brazil
São Paulo	Brazil
Rio de Janeiro	Brazil
Brasília	Brazil
Montréal	Canada
Edmonton	Canada
Vancouver	Canada
Toronto	Canada

(https://www.sqlitetutorial.net/wp-content/uploads/2015/11/SQLite-DISTINCT-multiple-columns.jpg)

The result set contains duplicate city and country e.g., Sao Paulo in Brazil as shown in the screenshot above.

To remove duplicate the city and country, you apply the `DISTINCT` clause to both city and country columns as shown in the following query:

```
SELECT  DISTINCT
        city,
        country
FROM
        customers
ORDER BY
        country;
```

Here is the partial output:

City	Country
Buenos Aires	Argentina
Sidney	Australia
Vienne	Austria
Brussels	Belgium
Brasília	Brazil
Rio de Janeiro	Brazil
São José dos Campos	Brazil
São Paulo	Brazil
Edmonton	Canada
Halifax	Canada
Montréal	Canada
Ottawa	Canada
Toronto	Canada

(https://www.sqlitetutorial.net/wp-content/uploads/2015/11/SQLite-SELECT-DISTINCT-multiple-columns.jpg)

As mentioned earlier, SQLite uses the combination of city and country to evaluate the duplicate.

SQLite SELECT DISTINCT with NULL example

This statement returns the names of companies of customers from the `customers` table.

```
SELECT company
FROM customers;
```

Try It ›

It returns 59 rows with many `NULL` values.

Now, if you apply the `DISTINCT` clause to the statement, it will keep only one row with a `NULL` value.

See the following statement:

```
SELECT DISTINCT company
FROM customers;
```

Try It ›

Company
(Null)
Apple Inc.
Banco do Brasil S.A.
Embraer - Empresa
Google Inc.
JetBrains s.r.o.
Microsoft Corporati
Riotur
Rogers Canada
Telus
Woodstock Discos

(https://www.sqlitetutorial.net/wp-content/uploads/2015/11/SQLite-SELECT-with-DISTINCT-NULL-values.jpg)

The statement returns 11 rows with one `NULL` value.

Note that if you select a list of columns from a table and want to get a unique combination of some columns, you can use the `GROUP BY` (https://www.sqlitetutorial.net/sqlite-group-by/) clause.

In this tutorial, you have learned how to remove duplicate rows from a result set using SQLite `SELECT DISTINCT` clause.