SQLite MAX

Summary: in this tutorial, you will learn how to use SQLite `MAX` function to get the maximum value of all values in a group.

Introduction to SQLite MAX function

The SQLite `MAX` function is an aggregate function (https://www.sqlitetutorial.net/sqlite-aggregate-functions/) that returns the maximum value of all values in a group. You can use the `MAX` function to accomplish a lot of things.

For example, you can use the `MAX` function to find the most expensive products, find the biggest item in its group, etc.

The following illustrates the basic syntax of the `MAX` function.

```
MAX([ALL|DISTINCT] expression);
```

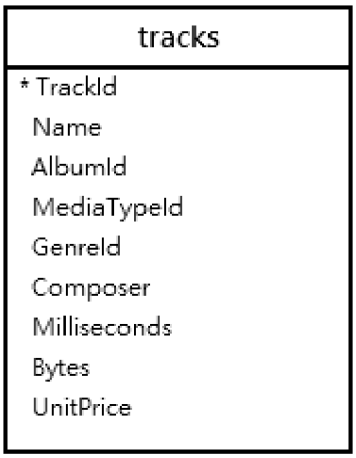
The `expression` can be a column of a table or an expression that consists of operands, which are the columns, and operators like +, *, etc.

There are some important notes about `MAX` function:

- First, the `MAX` function ignores `NULL` values.
- Second, unlike the `COUNT` (https://www.sqlitetutorial.net/sqlite-count-function/) function, the `DISTINCT` clause is not relevant to the `MAX` function.
- Third, because a column can store mixed types of data e.g., integer, real, text, blob, and `NULL` in SQLite, when comparing values to find the maximum value, the `MAX` function uses the rules mentioned in the data types tutorial (https://www.sqlitetutorial.net/sqlite-data-types/) .

The SQLite MAX function examples

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



To get the largest track in bytes, you apply the `MAX` function to the `bytes` column as the following statement:

```
SELECT MAX(bytes) FROM tracks;
```

Try It ›

max(bytes)
1059546140

SQLite MAX function in the subquery example

To get the complete information of the biggest track, you use the subquery (https://www.sqlitetutorial.net/sqlite-subquery/) as follows:

```
SELECT
        TrackId,
        Name,
        Bytes
FROM
        tracks
WHERE
        Bytes = (SELECT MAX(Bytes) FROM tracks);
```

Try It ›

TrackId	Name	Bytes
3224	Through a Looking Glass	1059546140

First, the inner query returns highest bytes of all tracks using the `MAX` function. Then, the outer query gets the largest track whose size equals the largest size returned by the subquery.

SQLite MAX function and GROUP BY clause example

You can find the largest track in each album using the `MAX` function with the GROUP BY (https://www.sqlitetutorial.net/sqlite-group-by/) clause.

First, the `GROUP BY` clause groups the tracks into groups based on albums. Then, the `MAX` function returns the largest tracks for each group.

See the following query:

```
SELECT
        AlbumId,
        MAX(bytes)
FROM
        tracks
GROUP BY
        AlbumId;
```

Try It ›

AlbumId	MAX(bytes)
1	11170334
2	5510424
3	6290521
4	12066294
5	12374569
6	16008629
7	12575396

SQLite MAX function and HAVING clause

You can combine the `MAX` function with the HAVING clause (https://www.sqlitetutorial.net/sqlite-having/) to filter the groups based on their largest values.

For example, to find the albums and their largest track where the sizes of the largest tracks are greater than 6 MB (about ~ 6000000), you use the following statement:

```
SELECT
        albumid,
        max(bytes)
FROM
        tracks
GROUP BY
        albumid
HAVING MAX(bytes) > 6000000;
```

Try It ›

AlbumId	max(bytes)
1	11170334
3	6290521
4	12066294
5	12374569
6	16008629
7	12575396
8	12089673
9	14375310

In this tutorial, you have learned how to use the SQLite `MAX` function to find the maximum value in a group of values.