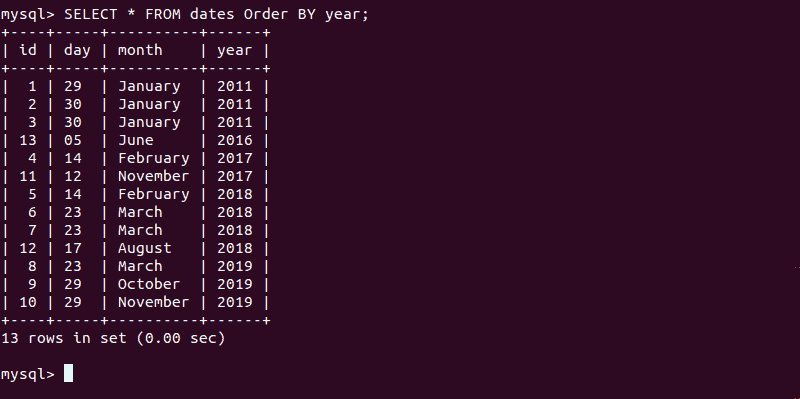
**Display the Contents of the Dates Table**

To see a display of all the dates you entered, ordered by year, type:

SELECT \* FROM dates ORDER BY year;



The output should show a list of dates in the appropriate order.

**Display Duplicate Rows**

To find out whether there are duplicate rows in the test database, use the command:

SELECT

day, COUNT(day),

month, COUNT(month),

year, COUNT(year)

FROM

dates

GROUP BY

day,

month,

year

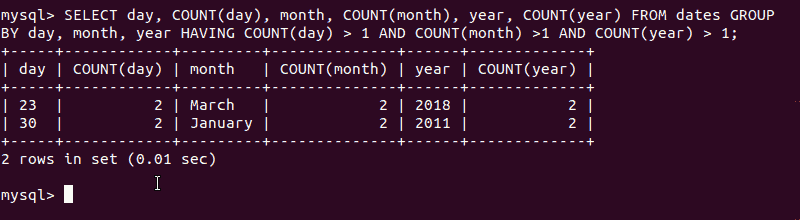
HAVING

COUNT(day) > 1

AND COUNT(month) > 1

AND COUNT(year) > 1;

The system will display any values that are duplicates. In this case, you should see:



delete t1 FROM dates t1

INNER JOIN dates t2

WHERE

t1.id < t2.id AND

t1.day = t2.day AND

t1.month = t2.month AND

t1.year = t2.year;

**Option 2: Remove Duplicate Rows Using an Intermediate Table**

You can create an **intermediate table** and use it to remove duplicate rows. This is done by transferring only the unique rows to the newly created table and deleting the original one (with the remaining duplicate rows).

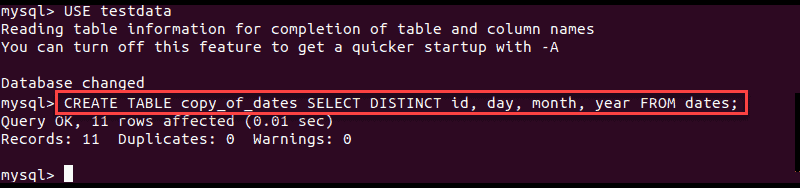
To do so follow the instructions below.

1. Create an intermediate table that has the same structure as the source table and transfer the unique rows found in the source:

CREATE TABLE [copy\_of\_source] SELECT DISTINCT [columns] FROM [source\_table];

For instance, to create a copy of the structure of the sample table **dates** the command is:

CREATE TABLE copy\_of\_dates SELECT DISTINCT id, day, month, year FROM dates;



2. With that done, you can [delete the source table with the drop command](https://phoenixnap.com/kb/mysql-drop-table) and rename the new one:

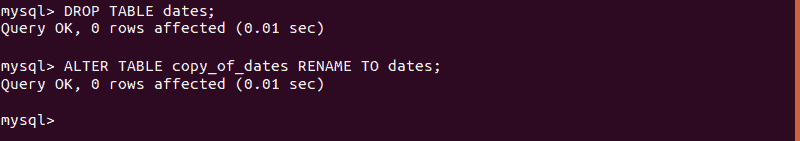
DROP TABLE [source\_table];

ALTER TABLE [copy\_of\_source] RENAME TO [source\_table];

For example:

DROP TABLE dates;

ALTER TABLE copy\_of\_dates RENAME TO dates;



### Option 3: Remove Duplicate Rows Using ROW\_NUMBER()

**Important:** This method is only available for **MySQL version 8.02** and later. [Check MySQL version](https://phoenixnap.com/kb/how-to-check-mysql-version) before attempting this method.

Another way to delete duplicate rows is with the **ROW\_NUMBER()** function.

SELECT \*. ROW\_NUMBER () Over (PARTITION BY [column] ORDER BY [column]) as [row\_number\_name];

Therefore, the command for our sample table would be:

SELECT \*. ROW\_NUMBER () Over (PARTITION BY id ORDER BY id) as row\_number;

The results include a **row\_number**column. The data is partitioned by **id**and within each partition there are unique row numbers. Unique values are labeled with row number **1**, while duplicates are **2**, **3**, and so on.

Therefore, to remove duplicate rows, you need to delete everything except the ones marked with 1. This is done by running a **DELETE** query with the **row\_number** as the filter.

To delete duplicate rows run:

DELETE FROM [table\_name] WHERE row\_number > 1;

In our example dates table, the command would be:

DELETE FROM dates WHERE row\_number > 1;

The output will tell you how many rows have been affected, that is, how many duplicate rows have been deleted.

You can verify there are no duplicate rows by running:

SELECT \* FROM [table\_name];