Find Duplicate Rows

by Sophia

WHAT'S COVERED

This tutorial explores finding duplicate rows of data in a table in three parts:

1. Introduction
2. Finding Duplicates
3. Duplicates for Counting

1. Introduction

Finding duplicate data in a table can be quite useful, as it can help us identify potential issues or matches. A duplicate row is one that refers to the same thing or person as a whole other row. However, it is important to note that not all duplicate rows will have completely identical information, as it will depend on what columns of data we want to search on. For example, we may have a large employee table that stores the employee's social security number that uniquely identifies the employee in the US. We may want to use this to ensure that the employee is only listed once. If we had customers, we could search for duplicate accounts with an email address, as traditionally an email address should only belong to a single customer for most eCommerce sites. If we had multiple records for a single customer, it would be difficult to get a full order history for the customer as they would have several rows that we would have to compare against.

2. Finding Duplicates

The structure of the query to find duplicates looks like the following:

```
SELECT <columnlist>
FROM <tablename>
GROUP BY <columnlist>
HAVING COUNT(*) > 1;
```

Note that the column list in the SELECT clause should match the column list in the GROUP BY clause. In addition, we could add a COUNT(*) in the SELECT clause so we can identify how many duplicates there are of the same criteria that we grouped by.

For example, we may want to verify that all of our customers have a unique phone number. We could do so like this:

```
SELECT phone, COUNT(*)
FROM customer
GROUP BY phone
HAVING COUNT(*) > 1;
```

We can verify that there are no customers that meet this criterion:

Query Results

Query ran successfully. 0 rows to display.

Perhaps our organization has a criterion that only allows a customer to have a single order. This query could be used to check which customers have more than one order. If they do, how many orders is listed:

```
SELECT customer_id, COUNT(*)
FROM invoice
GROUP BY customer_id
HAVING COUNT(*) > 1;
```

Query Results

Row count: 59

customer_id	count
29	7
54	7
4	7
34	7
51	7
52	7
10	7
35	7
45	7
6	7
39	7
36	7
31	7
50	7
14	7
22	7
59	6

We are quickly and easily able to identify those types of scenarios.

3. Duplicates for Counting

Note, too, that the finding of duplicates can be useful to identify rows of data that may occur more than once with the appropriate count. For example, we may want to identify the support_rep_id if they have served more than one customer:

```
SELECT support_rep_id, COUNT(*)
FROM customer
GROUP BY support_rep_id
HAVING COUNT(*) > 1;
```

Query Results

Row count: 3

support_rep_id	count
4	20
3	21
5	18

Or perhaps we want a list of state/country of those that have more than one customer, for marketing purposes:

```
SELECT state, country, COUNT(*)
FROM customer
GROUP BY state, country
HAVING COUNT(*) > 1;
```

Query Results

Row count: 9

state	country	count
SP	Brazil	3
	Czech Republic	2
	United Kingdom	3
	Germany	4
	Portugal	2
	India	2
CA	USA	3
	France	5
ON	Canada	2

Video Transcription

[MUSIC PLAYING] When it comes to finding duplicate information of data within a table, we want to identify which columns we want to query upon, being that there are certain types of data that we want to just look at a single column, or there might be two or three columns or even the entire column set within a table. What you want to do is identify it as part of the query, and then what it should be grouped by. Both of those should be exactly the same.

You don't have to have the count unless you want to display the number of times it actually repeats. So for example, in this case here, we have a couple of them that have duplicates. But there's only two of each. We can certainly make a couple of change in this case here to be able to see if there's more. But the key will be the having of the count of the number of rows that's greater than 1. That will ensure that we're checking for counts that have duplicates.

If we want to search for more than one, we can check for 2, 3, and so forth. So let's make a quick modification here, where we're only an account based on the country within. So you'll see here in this case, we'll be able to see that there's multiple different countries. There's many more repeats based on that same information.

[MUSIC PLAYING]

TRY IT

Your turn! Open the SQL tool by clicking on the LAUNCH DATABASE button below. Then enter in one of the examples above and see how it works. Next, try your own choices for which columns you want the query to provide.

SUMMARY

Finding duplicate rows in a table can be very useful for counting purposes.

Source: Authored by Vincent Tran