HAVING to Filter On Aggregates

by Sophia

WHAT'S COVERED

This tutorial explores using the HAVING clause to place a filter condition on groups and aggregates in two parts:

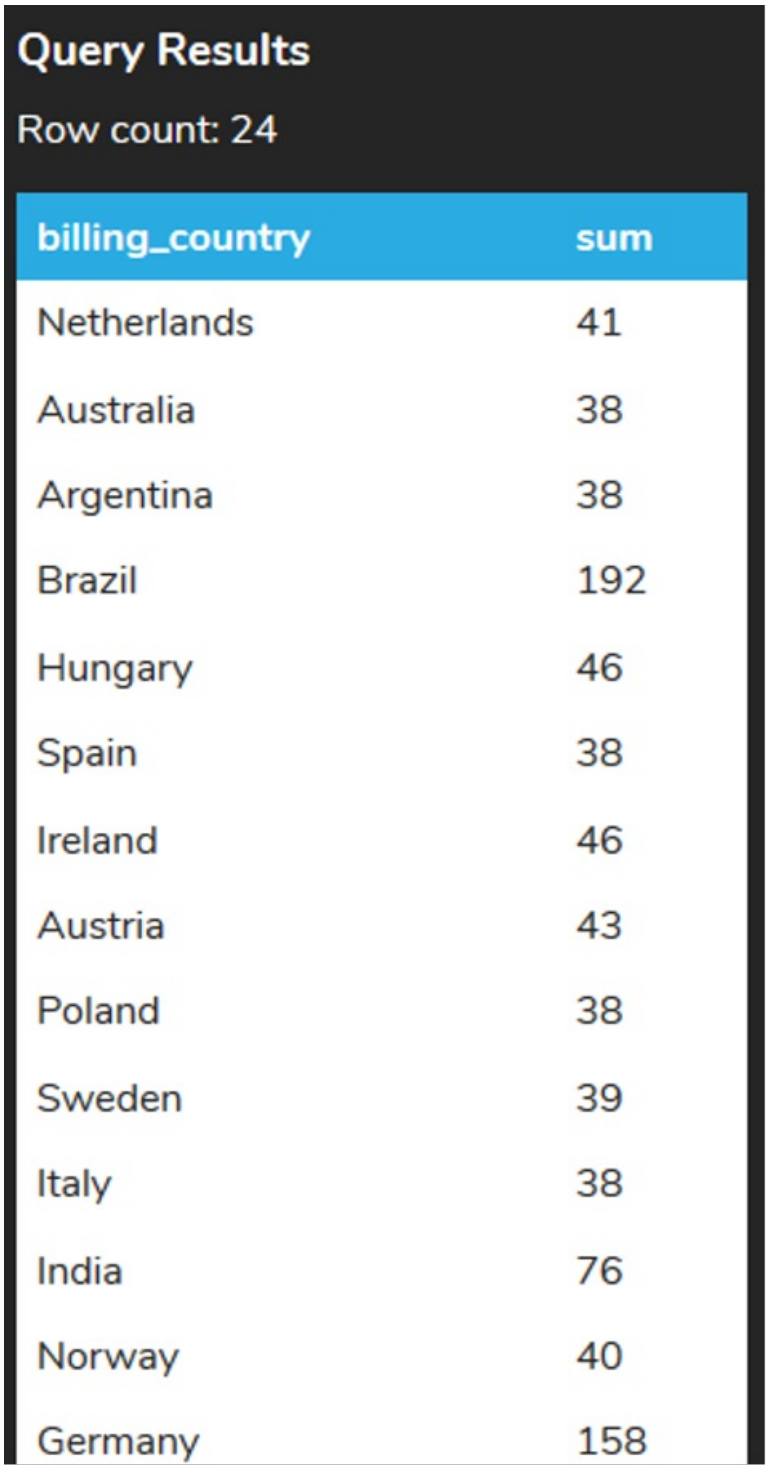
1. Using the HAVING Clause
2. More Aggregate Uses

1. Using the HAVING Clause

The HAVING clause allows us to specify a search condition for a group or an aggregate. It is used with the GROUP BY clause that divides rows into groups. It functions similarly to the WHERE clause, which you are already familiar with. The WHERE clause filters individual rows based on a specified condition. The HAVING clause filters groups of rows based on a set of conditions. They are similar, but separate from one another.

Let's take a look at the invoice table to find the SUM of the total that is grouped by country:

```
SELECT billing_country, SUM(total)
FROM invoice
GROUP BY billing_country;
```



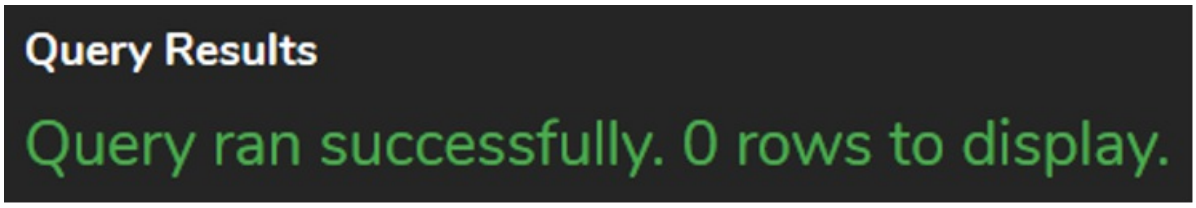
Query Results

Row count: 24

billing_country	sum
Netherlands	41
Australia	38
Argentina	38
Brazil	192
Hungary	46
Spain	38
Ireland	46
Austria	43
Poland	38
Sweden	39
Italy	38
India	76
Norway	40
Germany	158

We could not use the WHERE clause in this case, as the WHERE clause only looks at individual rows and not groups of rows. If we wanted to list only the countries that had the sum of the total greater than 50, we would add the HAVING clause to compare the aggregate function after the GROUP BY clause. Again, using the WHERE clause would not work:

```
SELECT billing_country, SUM(total)
FROM invoice
WHERE total > 50
GROUP BY billing_country;
```



This would look at individual rows that had a total > 50 instead of the groups. Instead, our query should look like this:

```
SELECT billing_country, SUM(total)
FROM invoice
GROUP BY billing_country
HAVING SUM(total) > 50;
```

Query Results

Row count: 9

billing_country	sum
Germany	158
France	197
United Kingdom	114
Czech Republic	91
India	76
Brazil	192
USA	528
Portugal	78
Canada	307

2. More Aggregate Uses

Note, too, that we don't have to have the same aggregate functions listed in the SELECT clause and HAVING clause. For example, if we wanted to show the number of invoices having the total amount by country > 50, we can change the SUM to COUNT in the SELECT clause:

```
SELECT billing_country, COUNT(total)
```

```
FROM invoice
GROUP BY billing_country
HAVING SUM(total) > 50;
```

Query Results

Row count: 9

billing_country	count
Germany	28
France	35
United Kingdom	21
Czech Republic	14
India	13
Brazil	35
USA	91
Portugal	14
Canada	56

We could also sort the results using the aggregate function as well:

```
SELECT billing_country, COUNT(total)
FROM invoice
GROUP BY billing_country
HAVING SUM(total) > 50
ORDER BY SUM(total);
```

Query Results

Row count: 9

billing_country	count
India	13
Portugal	14
Czech Republic	14
United Kingdom	21
Germany	28
Brazil	35
France	35
Canada	56
USA	91

We can use a variety of aggregate functions like the COUNT function to help filter. For example, we may want to find the countries and the number of customers that have a count greater than five:

```
SELECT country, COUNT(*)
FROM customer
GROUP BY country
HAVING COUNT(*) > 5;
```

Query Results

Row count: 2

country	count
USA	13
Canada	8

We could have multiple conditions to filter based on aggregate values by using the AND and OR operators in the HAVING clause. We may want to filter the group further to ensure that the number of customers also

checks for the number of customers that have a fax count greater than two. Note that as a reminder, counting a specific column only counts the non-null values, so if there is no fax, the row isn't counted – unlike using COUNT(*) where it counts the number of rows.

```
SELECT country, COUNT(*),COUNT(fax)
FROM customer
GROUP BY country
HAVING COUNT(*) > 5 AND COUNT(fax) > 2;
```

Query Results

Row count: 1

country	count	count
USA	13	4

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

The HAVING clause allows us to add a search condition for a group or aggregate from a GROUP BY clause.

Source: Authored by Vincent Tran