ANY and ALL Operators

by Sophia

WHAT'S COVERED

This lesson explores using ANY and ALL operators to compare values with a list of values returned by a subquery. Specifically, this lesson will cover:

1. Introduction

The **ALL** and **ANY** operators allow us to query data by comparing a value with a list of values that are returned by a subquery. This is an important distinction for the ANY and ALL operators, as they are focused on the lists from a subquery.

The syntax of the operator looks like the following:

```
<columnname> <operator> [ANY/ALL] (subquery);
```

The ANY operator is less restrictive than the ALL when it comes to comparisons. The ANY operator returns true if any of the values in the subquery meet the condition; otherwise, it returns false. The ALL operator returns true if ALL of the values in the subquery meet the condition; otherwise, it returns false.

TERMS TO KNOW

ANY
An operator that returns true if any of the values in the subquery meet its condition.

ALL
An operator that returns true only if all of the values in the subquery meet the condition.

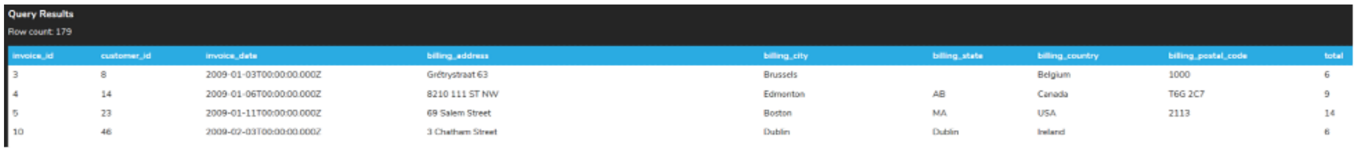
2. Subquery Example

Let's take a look at an example where we're needing to compare the average invoice totals per country:

```
SELECT AVG(total)
FROM invoice
GROUP BY billing_country;
```

We will use that average by country as our subquery. If we wanted to find invoices that have a value higher than the average of any of the totals from the country, we would use the following:

```
SELECT *
FROM invoice
WHERE total > ANY(
SELECT AVG(total)
FROM invoice
GROUP BY billing_country);
```

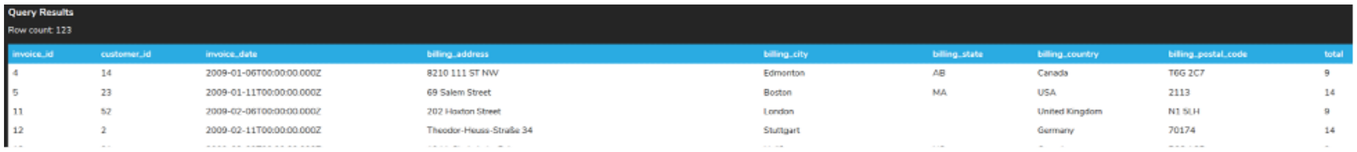

Query Results

Row count: 179

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
3	8	2009-01-03T00:00:00.000Z	Grétrystraat 63	Brussels		Belgium	1000	6
4	14	2009-01-06T00:00:00.000Z	8210 111 ST NW	Edmonton	AB	Canada	T6G 2C7	9
5	23	2009-01-11T00:00:00.000Z	69 Salem Street	Boston	MA	USA	2113	14
10	46	2009-02-03T00:00:00.000Z	3 Chatham Street	Dublin	Dublin	Ireland		6

To find invoices that have a value higher than all of the averages from all countries, use the following:

```
SELECT *
FROM invoice
WHERE total > ALL(
SELECT AVG(total)
FROM invoice
GROUP BY billing_country);
```


Query Results

Row count: 123

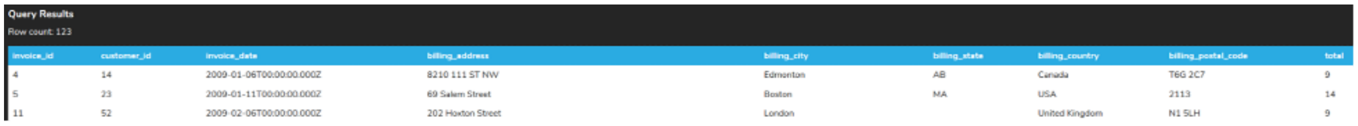
invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
4	14	2009-01-06T00:00:00.000Z	8210 111 ST NW	Edmonton	AB	Canada	T6G 2C7	9
5	23	2009-01-11T00:00:00.000Z	69 Salem Street	Boston	MA	USA	2113	14
11	52	2009-02-06T00:00:00.000Z	202 Hoxton Street	London		United Kingdom	N1 5LH	9
12	2	2009-02-11T00:00:00.000Z	Theodor-Heuss-Straße 34	Stuttgart		Germany	70174	14

Notice the count difference between the two queries. The first query (with the ANY operator) is less restrictive.

3. Potential Error

With the ANY and ALL operators, the subquery must return a single column to compare. If we return more than one column in the subquery, only the first result set is returned:

```
SELECT *
FROM invoice
WHERE (total,total) >= ALL(
SELECT AVG(total),max(total)
FROM invoice
GROUP BY billing_country);
```



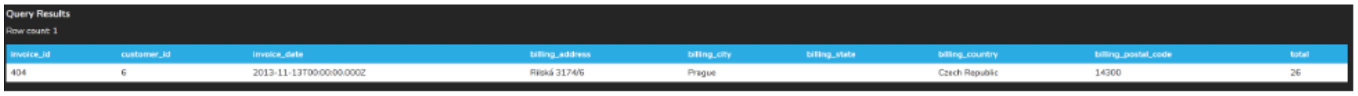
Query Results
Row count: 123

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
4	14	2009-01-06T00:00:00.000Z	8210 111 ST NW	Edmonton	AB	Canada	T6G 2C7	9
5	23	2009-01-11T00:00:00.000Z	69 Salem Street	Boston	MA	USA	2113	14
11	52	2009-02-06T00:00:00.000Z	202 Hoxton Street	London		United Kingdom	N1 5LH	9

Consider if we just compared the total to the max of the totals:

```
SELECT *
FROM invoice
WHERE total >= ALL(
SELECT max(total)
FROM invoice
GROUP BY billing_country);
```

One record would be returned:



Query Results
Row count: 1

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
404	6	2013-11-13T00:00:00.000Z	Rilská 3174/6	Prague		Czech Republic	14300	26

4. Operators for Comparisons

You may notice that the query above uses >= instead of >. We can compare ANY and ALL using a variety of operators, as shown in the following table.

Operator	The Expression Evaluates to True if...
>ALL	A value in the main query is greater than the largest value returned by the subquery.
>=ALL	A value in the main query is greater than or equal to the largest value returned by the subquery.
<ALL	A value in the main query is less than the smallest value returned by the subquery.
<=ALL	A value in the main query is less than or equal to the smallest value returned by the subquery.
=ALL	Every value in the main query is equal to every value returned by the subquery.
!=ALL	A value in the main query is not equal to every value returned by the subquery. This is equivalent to the <>ALL operator.
>ANY	A value in the main query is greater than the largest value returned by the subquery.

>=ANY	A value in the main query is greater than or equal to the smallest value returned by the subquery.
<ANY	A value in the main query is less than the largest value returned by the subquery.
<=ANY	A value in the main query is less than or equal to the largest value returned by the subquery.
=ANY	A value in the main query is equal to any value returned by the subquery. This is equivalent to the IN operator.

WATCH

TRY IT

Your turn! Open the SQL tool by clicking on the LAUNCH DATABASE button below. Then, enter 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

In this lesson, you learned that SQL's ANY and ALL operators make complex comparisons easier and more efficient by offering a concise and flexible way to compare values. They compare a value with a set of values returned by a subquery. These operators enable you to compare multiple values from a subquery without specifying each value explicitly. You saw some **subquery examples** of the correct syntax for using these operators and looked at some **potential errors**. Finally, you reviewed a table **comparing the different operators** used with ANY and ALL and what results they generate.

TERMS TO KNOW

Any and All
The ANY and ALL operators compare a value with a set of values returned by a subquery. These operators are particularly useful if you want to compare multiple values from a subquery without specifying each value explicitly.