aws re/start

Organizing Data

Database Fundamentals

Welcome to Organizing Data.

What you will learn

At the core of the lesson

You will learn how to do the following:

- Use the ORDER BY clause to sort data by a specific column in either ascending or descending order.
- Use the GROUP BY and HAVING clauses to group data and filter groups.

Key terms:

- Sorting
- ORDER BY clause
- GROUP BY clause
- HAVING clause

aws re/start

In this module, you will learn how to do the following:

- Use the ORDER BY clause to sort data by a specific column in either ascending or descending order.
- Use the GROUP BY and HAVING clauses to group data and filter groups.

Key terms include the following:

- Sorting
- ORDER BY clause
- GROUP BY clause
- HAVING clause

Organizing data by using SQL

- Sorting is the practice of organizing the sequence of the data returned by a query so that the data can be analyzed effectively.
- Structured query language (SQL) statements use the ORDER BY clause to sort query output in a specified order.
- Query output can be sorted in either ascending or descending order.
- SQL statements use the GROUP BY clause to combine query output into groups.
- SQL statements use the HAVING clause to apply filter conditions to aggregated group data.

aws re/start

- You can use the SQL ORDER BY clause to return query results in a specific sort order of your choosing.
- You can use the GROUP BY clause to aggregate query results into smaller result sets.
- You can use the HAVING clause to limit the results of a query that includes aggregated data.

Sorting and ORDER BY keyword

You'll begin with some examples of how to sort query output.

Query output with no sorting

Country Table

Column	Type
Code	Character
Name	Character
Continent	Character
Region	Character
SurfaceArea	Float
IndepYear	Integer
Population	Integer
LifeExpectancy	Float
GNP	Float
GNPOld	Float
LocalName	Character
Capital	Integer
Code2	Character

Query

```
SELECT name, continent, surfacearea
FROM country
WHERE surfacearea >= 5000000;
```

Query results are returned in no specific order.

Output

```
name               continent      surfacearea
---------------    ------------   ------------
Antarctica         Antarctica      13120000.00
Brazil             South America    8547403.00
Australia          Oceania          7741220.00
Canada             North America    9970610.00
United States      North America    9363520.00
Russian Federation Europe          17075400.00
China              Asia             9572900.00
```


aws re/start

This query does not include any SQL clauses for sorting. Therefore, the data is returned in no specific order. Some SQL databases return the rows in the order in which they were originally loaded into the table.

You can add an ORDER BY clause to the SQL statement if you want the query output to be returned in a specific order.

Query output sorted in ascending order

Query

```
SELECT name, continent, surfacearea
FROM country
WHERE surfacearea >= 5000000
ORDER BY surfacearea ASC;
```

ASC keyword with ORDER BY

Output

```
name               continent       surfacearea
---------------    ------------    -----------
Australia          Oceania          7741220.00
Brazil             South America    8547403.00
United States      North America    9363520.00
China              Asia             9572900.00
Canada             North America    9970610.00
Antarctica         Antarctica      13120000.00
Russian Federation Europe          17075400.00
```

Lowest value

Arranged in ascending order

Highest value

aws re/start

This query includes an ORDER BY clause, which sorts the query results in a specific order. In this case, it orders by the values of the surfacearea column from smallest to largest. Ordering data from smallest to largest value is referred to as ascending order.

Ascending order sorts are specified by including the ASC keyword, which follows the ORDER BY clause. Because ascending order sorts are the default sort order for an ORDER BY clause, omitting the ASC keyword will still provide the same results.

Query output sorted in descending order

Query

```
SELECT name, continent, surfacearea
FROM country
WHERE surfacearea >= 5000000
ORDER BY surfacearea DESC;
```

DESC keyword with ORDER BY

Output

```
name               continent       surfacearea
---------------    ------------    -----------
Russian Federation Europe          17075400.00
Antarctica         Antarctica      13120000.00
Canada             North America    9970610.00
China              Asia             9572900.00
United States      North America    9363520.00
Brazil             South America    8547403.00
Australia          Oceania          7741220.00
```

Highest value

Arranged in descending order

Lowest value

aws re/start

This query also includes a SQL clause with an ORDER BY clause. This time the query results are sorted by the values of the surfacearea column from largest to smallest. Ordering data from largest to smallest values is referred to as descending order.

Descending sorts are specified by including the DESC keyword, which follows the ORDER BY clause.

Query output by using multiple sort operations

Query

```
SELECT name, continent, surfacearea
FROM country
WHERE surfacearea >= 5000000
ORDER BY continent ASC, surfacearea DESC;
```

The ORDER BY clause specifies two sort operations.

Output

name	continent	surfacearea
Antarctica	Antarctica	13120000.00
China	Asia	9572900.00
Russian Federation	Europe	17075400.00
Canada	North America	9970610.00
United States	North America	9363520.00
Australia	Oceania	7741220.00
Brazil	South America	8547403.00

Arranged in ascending order of continent and descending order of surfacearea within continents

aws re/start

An ORDER BY clause can contain more than one sort operation. This example first sorts the query results by continent name in ascending order. This sort is called the primary sort order because this sort is listed first in the ORDER BY clause. Next, the results of the primary sort are sorted again in descending order of surfacearea. This sort is called a secondary sort because this sort is listed second in the ORDER BY clause.

ORDER BY clauses are not limited to two sort operations. The available columns in the table or tables that are being queried drive the list of columns by which you can sort.

Query output by using implicit columns in sort operations

Query

```
SELECT name, continent, surfacearea
FROM country
WHERE surfacearea >= 5000000
ORDER BY 2 ASC, 3 DESC;
```

Implicit column placeholders can be used with the ORDER BY clause.

Output

name	continent	surfacearea
Antarctica	Antarctica	13120000.00
China	Asia	9572900.00
Russian Federation	Europe	17075400.00
Canada	North America	9970610.00
United States	North America	9363520.00
Australia	Oceania	7741220.00
Brazil	South America	8547403.00

Instead of spelling out entire column names in an ORDER BY clause, you can also use implicit column placeholders.

This query has continent and surfacearea columns as the second and third columns in the SELECT clause.

You can use those implicit numeric placeholder values in the ORDER BY clause in place of the actual column names themselves. In this example, the number 2 in the ORDER BY clause represents the continent column from the SELECT clause. The number 3 represents the surfacearea column from the SELECT clause.

You can choose the method that you use to specify the sort columns based on your personal preferences. Alternatively, a company's established coding standards and practices might formally define this method.

Grouping and filtering data

Next, you look at how data can be grouped and filtered.

Grouping data in query output

```
SELECT continent, name
FROM country
WHERE (continent = 'South America'
AND population > 12000000)
OR continent = 'Antarctica'
ORDER BY 1, 2;
```

What if you want to return the number of countries per continent instead of listing each country by name?

```
continent       name
------------    ----------------------------------------
Antarctica      Antarctica
Antarctica      Bouvet Island
Antarctica      French Southern territories
Antarctica      Heard Island and McDonald Islands
Antarctica      South Georgia and the South Sandwich Islands
South America   Argentina
South America   Brazil
South America   Chile
South America   Colombia
South America   Ecuador
South America   Peru
South America   Venezuela
```

```
continent     countries
------------ ---------
Antarctica            5
South America         7
```


aws re/start

This query returns the continent and name of each country that has one of the following characteristics:

- The country is in South America and has a population of more than 12 million people.
- The country is on the continent of Antarctica.

The results are ordered by country name within continent.

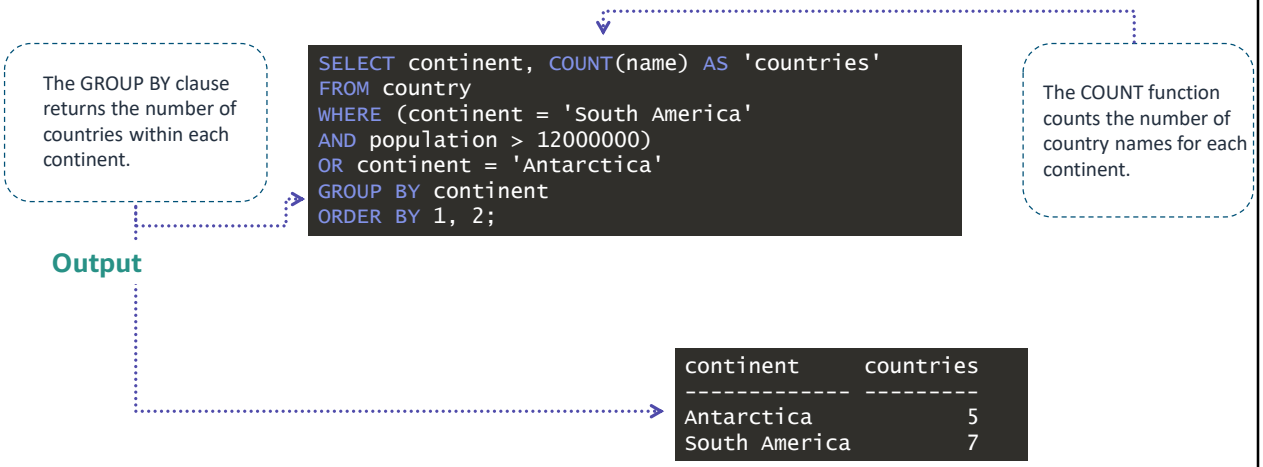
However, not all queries require such detailed results. In some situations, data that is grouped together and summarized is more useful for analysis.

For example, what if you were interested in only the total number of countries per continent and not the individual names of each country?

You can use the GROUP BY clause to summarize data in this way.

Grouping data in query output by using GROUP BY

Query



Output

continent	countries
Antarctica	5
South America	7

aws re/start

You can use the GROUP BY clause in a SQL statement to group data items of the same value together.

The GROUP BY clause is typically used in conjunction with SELECT statements that include SQL aggregation functions such as COUNT, MAX , MIN, SUM, and AVG.

The GROUP BY clause groups the query results together by using the specified aggregation function.

In this example, the GROUP BY clause groups together all the data items that have the same value for continent name. Then, it counts how many country names are included in that group.

Using GROUP BY items with filter conditions

SQL statement WHERE clauses are evaluated before the GROUP BY clause.

The HAVING clause is used to filter query results after applying the GROUP BY clause.

The HAVING clause will include the same column used in the aggregation function of the SELECT clause.

aws re/start

The WHERE clause is evaluated before the HAVING clause in a SQL statement.

Adding the HAVING clause as filter condition

Query

The column included in the aggregation function is also used in the HAVING clause to filter the query results.

```
SELECT continent, COUNT(name) AS 'countries'
FROM country
WHERE (continent = 'South America'
AND population > 12000000)
OR continent = 'Antarctica'
GROUP BY continent
HAVING COUNT(name) > 5
ORDER BY 1, 2;
```

Output

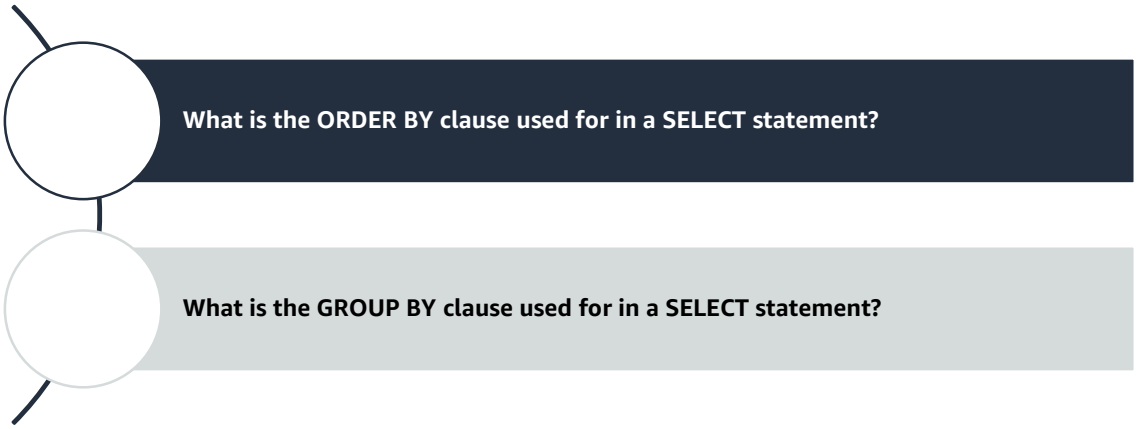
```
continent     countries
------------- ---------
South America         7
```


aws re/start

The HAVING clause in a SQL statement is used with the GROUP BY clause to add a filter condition based on the aggregated value.

In this example, the GROUP BY clause returns two rows. The HAVING clause then further filters those rows to return only the continents that have more than five countries in them.

Checkpoint questions



aws re/start

1. What is the ORDER BY clause used for in a SELECT statement?

   The ORDER BY clause is used to sort the rows in a query result set in a certain order.

2. What is the GROUP BY clause used for in a SELECT statement?

   The GROUP BY clause combines rows into groups based on matching values.

Key takeaways

- Sorting is the practice of organizing the sequence of the data returned by a query so that the data can be analyzed effectively.
- Use the ORDER BY clause to sort data in a specific column in ascending or descending order by using the keyword ASC or DESC.
- Use the GROUP BY clause to combine query output into groups.
- Use the HAVING clause to filter query results by using grouped data.

aws re/start

This lesson includes the following takeaways:

- Sorting is the practice of organizing data into a structured format so that it can be analyzed effectively.
- Use the ORDER BY clause to sort data in a specific column in ascending or descending order by using the keyword ASC or DESC.
- Use the GROUP BY clause to combine query output into groups.
- Use the HAVING clause to filter query results by using grouped data.

Thank you

Thank you for completing this module.