Numeric commands

When looking at a long list of numbers, you may want to know the sum of those numbers. Or maybe you want to know how many numbers are in the list, what the average of the numbers is, or what the highest and lowest numbers are. For example, if you're looking at your monthly expenses, you may want to know the total amount that you've spent or the most expensive item that you purchased.

Like Excel, SQL has commands that can be applied to numeric values; these commands perform common calculations for you. In this lesson, you'll learn about seven common SQL numeric commands**: SUM, AVG, MIN, MAX, COUNT, ABS, and ROUND**. As you'll see, these commands are paired with **SELECT** to calculate and then return a value.

SUM

The **SUM** command in SQL is used in a SELECT statement to return the sum of a numeric column. The SUM command follows this structure:

```
1  SELECT SUM(quantity)
2  FROM purchase_items;
```

Now, say that you want to see how many items were returned. Go ahead and use the same query as above, but add the WHERE clause to specify this condition. Your query should look like this:

```
1  SELECT SUM(quantity)
2  FROM purchase_items
3  WHERE state = 'Returned';
```

AVG

The **AVG** command is used in a SELECT statement to return the average of a numeric column. The AVG command follows this structure:

```
1  SELECT AVG(column_name)
2  FROM table_name
3  WHERE condition;
```

Remember, the WHERE clause is always optional; it's only included if you want to filter on a condition.

MIN

The **MIN** command is used in a SELECT statement to return the lowest value in a numeric column. This is the structure of the MIN command:

```
1  SELECT MIN(column_name)
2  FROM table_name
3  WHERE condition;
```

MAX

The **MAX** command is used in a SELECT statement to return the highest value in a numeric column. This is the structure of the MAX command:

```
1  SELECT MAX(column_name)
2  FROM table_name
3  WHERE condition;
```

COUNT

The **COUNT** command is used in a SELECT statement to return the total number of values in a column. The COUNT command can be applied to numeric or string data types. This is the structure of the COUNT command:

```
1  SELECT COUNT(column_name)
2  FROM table_name
3  WHERE condition;
```

ABS

The **ABS** command is paired with SELECT to return the absolute value of a number. The absolute value is useful when you're finding the difference between two numbers and want to identify the amount of change, but you don't care if it is an increase or decrease. The structure of this command is straightforward, as shown below.

```
1  SELECT ABS(value);
```

ROUND

The **ROUND** command is paired with SELECT to return a rounded numeric value. The structure of this command is similar to ABS, as shown below.

```
1  SELECT ROUND(value, decimal_places_to_round);
```