

- Video: Queries within queries

6 min
- Video: Using subqueries to aggregate data

7 min
- Reading: SQL functions and subqueries: A functional friendship

10 min
- Ungraded Plugin: Subqueries

30 min
- Video: Justin: Where data analysis takes you

2 min
- Practice Quiz: Test your knowledge on working with subqueries

3 questions

SQL functions and subqueries: A functional friendship

In this reading, you will learn about SQL functions and how they are sometimes used with subqueries. **SQL functions** are tools built into SQL to make it possible to perform calculations. A **subquery** (also called an inner or nested query) is a query within another query.

How do SQL functions, function?

SQL functions are what help make data aggregation possible. (As a reminder, data aggregation is the process of gathering data from multiple sources in order to combine it into a single, summarized collection.) So, how do SQL functions work? Going back to W3Schools, let's review some of these functions to get a better understanding of how to run these queries:

- **SQL HAVING** [↗](#): This is an overview of the HAVING clause, including what it is and a tutorial on how and when it works.
- **SQL CASE** [↗](#): Explore the usage of the CASE statement and examples of how it works.
- **SQL IF** [↗](#): This is a tutorial of the IF function and offers examples that you can practice with.
- **SQL COUNT** [↗](#): The COUNT function is just as important as all the rest, and this tutorial offers multiple examples to review.

Subqueries - the cherry on top

Think of a query as a cake. A cake can have multiple layers contained within it and even layers within those layers. Each of these layers are our subqueries, and when you put all of the layers together, you get a cake (query). Usually, you will find subqueries nested in the SELECT, FROM, and/or WHERE clauses. There is no general syntax for subqueries, but the syntax for a basic subquery is as follows:

```
SELECT account_table.*
FROM (
    SELECT *
    FROM transaction.sf_model_feature_2014_01
    WHERE day_of_week = 'Friday'
) account_table
WHERE account_table.availability = 'YES'
```

You will find that, within the first SELECT clause is another SELECT clause. The second SELECT clause marks the start of the subquery in this statement. There are many different ways in which you can make use of subqueries, and resources referenced will provide additional guidance as you learn. But first, let's recap the subquery rules.

There are a few rules that subqueries must follow:

- Subqueries must be enclosed within parentheses
- A subquery can have only one column specified in the SELECT clause. But if you want a subquery to compare multiple columns, those columns must be selected in the main query.
- Subqueries that return more than one row can only be used with multiple value operators, such as the IN operator which allows you to specify multiple values in a WHERE clause.
- A subquery can't be nested in a SET command. The SET command is used with UPDATE to specify which columns (and values) are to be updated in a table.

Additional resources

The following resources offer more guidance into subqueries and their usage:

- **SQL subqueries:** [↗](#)This detailed introduction includes the definition of a subquery, its purpose in SQL, when and how to use it, and what the results will be
- **Writing subqueries in SQL** [↗](#): Explore the basics of subqueries in this interactive tutorial, including examples and practice problems that you can work through

As you continue to learn more about using SQL, functions, and subqueries, you will realize how much time you can truly save when memorizing these tips and tricks.

