

Work with temporary tables

Temporary tables are exactly what they sound like—temporary tables in a SQL database that aren't stored permanently. In this reading, you will learn the methods to create temporary tables using SQL commands; you will also learn a few best practices to follow when working with temporary tables.

A quick refresher on what you have already learned about temporary tables

- They are automatically deleted from the database when you end your SQL session.
- They can be used as a holding area for storing values if you are making a series of calculations. This is sometimes referred to as **pre-processing** of the data.
- They can collect the results of multiple, separate queries. This is sometimes referred to as data **staging**. Staging is useful if you need to perform a query on the collected data or merge the collected data.
- They can store a filtered subset of the database. You don't need to select and filter the data each time you want to work with it. In addition, using fewer SQL commands helps to keep your data clean.

It is important to point out that each database has its own unique set of commands to create and manage temporary tables. We have been working with BigQuery, so we will focus on the commands that work in that environment. The rest of this reading will go over the ways to create temporary tables, primarily in BigQuery.

[illegible]

Temporary table creation in BigQuery

Temporary tables can be created using different clauses. In BigQuery, the **WITH** clause can be used to create