**MODULE 3**

Knowing a variety of ways to clean data can make a data analyst’s job much easier. In this part of the course, you’ll use SQL to clean data from databases. In particular, you’ll explore how SQL queries and functions can be used to clean and transform your data before an analysis.

### 

### 

### **Learning Objectives**

* Describe how SQL can be used to clean large datasets
* Compare spreadsheet data-cleaning functions to those associated with SQL in databases.
* Develop basic SQL queries for use with databases.
* Apply basic SQL functions for use in cleaning string variables in a database.
* Apply basic SQL functions for transforming data variables

**SQL FOR SPARKLING CLEAN DATA**

[**USE SQL TO CLEAN DATA**](https://www.coursera.org/learn/process-data/lecture/w6EFG/use-sql-to-clean-data)

Now that we know the difference between cleaning dirty data and some general data cleaning techniques, let's focus on data cleaning using SQL. Coming up we'll learn about the different data cleaning functions in spreadsheets and SQL and how SQL can be used to clean large data sets. I'll also show you how to develop some basic search queries for databases and how to apply basic SQL functions for transforming data and cleaning strings. Cleaning your data is the last step in the data analysis process before you can move on to the actual analysis, and SQL has a lot of great tools that can help you do that.

But before we start cleaning databases, we'll take a closer look at SQL and when to use it

**[SALLY: FOR THE LOVE OF SQL](https://www.coursera.org/learn/process-data/lecture/XWc4S/sally-for-the-love-of-sql)**

[Advertising agencies get money from their clients to advertise their brand. These agencies use our products, use certain Google platforms, advertising platforms, and I help them with how to best use those platforms, different strategies they can use to be best in class.](https://www.coursera.org/learn/process-data/lecture/XWc4S/sally-for-the-love-of-sql)

[A lot of the folks at the advertising agencies have reports that they have to send out to their clients.](https://www.coursera.org/learn/process-data/lecture/XWc4S/sally-for-the-love-of-sql)

[These reports take a lot of time to create and visualize, and so what I do is I help the practitioners and the analytics teams use a particular product that enables them to create those reports much faster and much easier.](https://www.coursera.org/learn/process-data/lecture/XWc4S/sally-for-the-love-of-sql)

[**If you're going to start off as a data analyst, it opens tons of doors because everybody is tracking data, is using data, needs to use data, regardless of industry**.](https://www.coursera.org/learn/process-data/lecture/XWc4S/sally-for-the-love-of-sql)

[Anywhere from health care, to advertising, to e-commerce, to entertainment, anything and everything, everybody uses data, so everybody needs you as a data analyst.](https://www.coursera.org/learn/process-data/lecture/XWc4S/sally-for-the-love-of-sql)

[**SQL makes our lives easier when we're analyzing lots of different data.**](https://www.coursera.org/learn/process-data/lecture/XWc4S/sally-for-the-love-of-sql)

[It's only somewhat recently that the SQL programs that we use now can give us instant results for analyzing millions or billions of data. Years ago, maybe about five years ago or so, even though we could still analyze those millions of rows, we would end up having to wait fifteen minutes, thirty minutes for the queries to run. But **now it's instantaneous**, and so that's really exciting, and we can do so much more with that power. **SQL** has helped a lot in my career because it's one of those **fundamental things you have to know as a data analyst**.](https://www.coursera.org/learn/process-data/lecture/XWc4S/sally-for-the-love-of-sql)

[Back in the day, not everyone knew SQL, so **knowing SQL** was definitely a competitive advantage. Nowadays, I would say more people, maybe most people know it. **It's a core skill and highly sought after by everybody**. So, knowing SQL, becoming a data analyst makes you quite popular with recruiters, so I think that's really fun.](https://www.coursera.org/learn/process-data/lecture/XWc4S/sally-for-the-love-of-sql)

[I taught myself SQL, so my knowledge about SQL is something I hold near and dear, close to my heart since it's something that almost I've made for myself, and I feel so much satisfaction from it. So that's why I really like SQL. One of the fun things about SQL and another reason why I really enjoy using it is because when you type something in that query, and you just hit Control, Shift, Enter, or once you've run the query, you get the results almost instantly, depending on the platform you use. But it's fascinating to see if you think conceptually how much analysis the computer is doing for you based on that little bit of command code or a little bit of code you wrote, and it's just so powerful if you think about what's happening behind the scenes. So I think that's fun to look at. **We live in a world of big data**, and it keeps getting bigger.](https://www.coursera.org/learn/process-data/lecture/XWc4S/sally-for-the-love-of-sql)

[The computing power is also increasing exponentially. With all the data that we can track, the more and more we can track that data, the more and more we need data analysts. Our career prospects are basically skyrocketing. I'm Sally, I'm a measurement and analytical lead at Google.](https://www.coursera.org/learn/process-data/lecture/XWc4S/sally-for-the-love-of-sql)

**[UNDERSTAND SQL CAPABILITIES](https://www.coursera.org/learn/process-data/lecture/fDPil/understand-sql-capabilities)**

[We've talked about SQL a lot already. You've seen some databases and some basic functions in SQL, and you've even seen how SQL can be used to process data. But now let's actually define SQL. SQL is a structured query language that analysts use to work with databases. Data analysts usually use SQL to deal with large datasets because it can handle huge amounts of data. And I mean trillions of rows. That's a lot of rows to wrap your head around. So let me give you an idea about how much data that really is.](https://www.coursera.org/learn/process-data/lecture/fDPil/understand-sql-capabilities)

[Imagine a data set that contains the names of all 8 billion people in the world. It would take the average person 101 years to read all 8 billion names. SQL can process this in seconds. Personally, I think that's pretty cool. Other tools like spreadsheets might take a really long time to process that much data,](https://www.coursera.org/learn/process-data/lecture/fDPil/understand-sql-capabilities) **[which is one of the main reasons data analysts choose to use SQL, when dealing with big datasets. Let me give you a short history on SQL. Development on SQL actually began in the early 70s.](https://www.coursera.org/learn/process-data/lecture/fDPil/understand-sql-capabilities)**

[In 1970, **Edgar F.Codd developed the theory about relational databases**. You might remember learning about **relational databases** a while back. This is a **database that contains a series of tables that can be connected to form relationships**.](https://www.coursera.org/learn/process-data/lecture/fDPil/understand-sql-capabilities)

[At the time IBM was using a relational database management system called System R. Well, IBM computer scientists were trying to figure out a way to manipulate and retrieve data from IBM System R.](https://www.coursera.org/learn/process-data/lecture/fDPil/understand-sql-capabilities)

[Their first query language was hard to use. **So they quickly moved on to the next version, SQL. In 1979**, after extensive testing SQL, now just spelled S-Q-L, was released publicly. By 1986, SQL had become the standard language for relational database communication, and it still is. **This is another reason why data analysts choose SQL**. It's a well-known standard within the community.](https://www.coursera.org/learn/process-data/lecture/fDPil/understand-sql-capabilities)

[The first time I used SQL to pull data from a real database was for my first job as a data analyst. I didn't have any background knowledge about SQL before that. I only found out about it because it was a requirement for that job.](https://www.coursera.org/learn/process-data/lecture/fDPil/understand-sql-capabilities)

[The recruiter for that position gave me a week to learn it. So I went online and researched it and ended up teaching myself SQL. They actually gave me a written test as part of the job application process. I had to write SQL queries and functions on a whiteboard. But I've been using SQL ever since. And I really like it.](https://www.coursera.org/learn/process-data/lecture/fDPil/understand-sql-capabilities)

[**And just like I learned SQL on my own, I wanted to remind you that you can figure things out yourself too.**](https://www.coursera.org/learn/process-data/lecture/fDPil/understand-sql-capabilities)

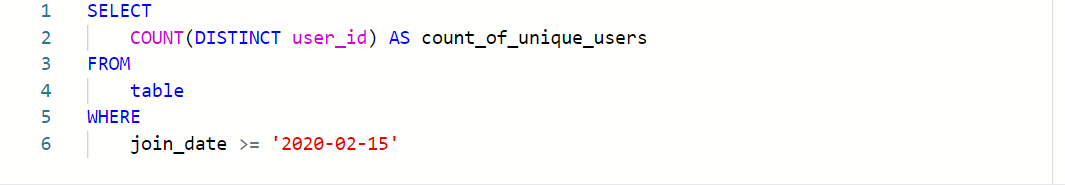
[There's tons of **great online resources for learning**. So don't let one job requirement stand in your way without doing some research first. Now that we know a little more about why analysts choose to work with SQL when they're handling a lot of data and a little bit about the history of SQL, we'll move on and learn some practical applications for it. Coming up next, we'll check out some of the tools we learned in spreadsheets and figure out if any of those apply to working in SQL.](https://www.coursera.org/learn/process-data/lecture/fDPil/understand-sql-capabilities)

[**HOW A JUNIOR DATA ANALYST USES SQL**](https://www.coursera.org/learn/process-data/supplement/GzgMb/how-a-junior-data-analyst-uses-sql)

**Spreadsheets functions and formulas or SQL queries?**

Before they can address this question, this data analyst needs to choose what tool to use. First, they have to think about where the data lives. If it is stored in a database, then SQL is the best tool for the job. But if it is stored in a spreadsheet, then they will have to perform their analysis in that spreadsheet. In that scenario, they could create a pivot table of the data and then apply specific formulas and filters to their data until they were given the number of users that joined after February 15th. It isn’t a really complicated process, but it would involve a lot of steps.

In this case, the data is stored in a database, so they will have to work with SQL. And this data analyst knows they could get the same results with a single SQL query:



Spreadsheets and SQL both have their advantages and disadvantages:

| **Features of Spreadsheets** | **Features of SQL Databases** |
| --- | --- |
| Smaller data sets | Larger datasets |
| Enter data manually | Access tables across a database |
| Create graphs and visualizations in the same program | Prepare data for further analysis in another software |
| Built-in spell check and other useful functions | Fast and powerful functionality |
| Best when working solo on a project | Great for collaborative work and tracking queries run by all users |

## **Key takeaways**

When it comes down to it, where the data lives will decide which tool you use. If you are working with data that is already in a spreadsheet, that is most likely where you will perform your analysis. And if you are working with data stored in a database, SQL will be the best tool for you to use for your analysis. You will learn more about SQL coming up, so that you will be ready to tackle any business problem with the best tool possible.

**[SPREADSHEETS VERSUS SQL](https://www.coursera.org/learn/process-data/lecture/Y15Mz/spreadsheets-versus-sql)**

[Hey there. So far we've learned about both spreadsheets and SQL. While there's lots of differences between spreadsheets and SQL, you'll find some similarities too.](https://www.coursera.org/learn/process-data/lecture/Y15Mz/spreadsheets-versus-sql)

[Let's check out what spreadsheets and SQL have in common and how they're different. Spreadsheets and SQL actually have a lot in common.](https://www.coursera.org/learn/process-data/lecture/Y15Mz/spreadsheets-versus-sql)

[Specifically, there's tools you can use in both spreadsheets and SQL to achieve similar results. We've already learned about some tools for cleaning data in spreadsheets, which means you already know some tools that you can use in SQL. For example, you can still perform arithmetic, use formulas and join data when you're using SQL, so we'll build on the skills we've learned in spreadsheets and use them to do even more complex work in SQL. Here's an example of what I mean by more complex work. If we were working with health data for a hospital, we'd need to be able to access and process a lot of data. We might need demographic data, like patients' names, birthdays, and addresses, information about their insurance or past visits, public health data or even user generated data to add to their patient records. All of this data is being stored in different places, maybe even in different formats, and each location might have millions of rows and hundreds of related tables. This is way too much data to input manually, even for just one hospital. That's where SQL comes in handy. Instead of having to look at each individual data source and record it in our spreadsheet, we can use SQL to pull all this information from different locations in our database.](https://www.coursera.org/learn/process-data/lecture/Y15Mz/spreadsheets-versus-sql)

[Now, let's say we want to find something specific in all this data, like how many patients with a certain diagnosis came in today. In a spreadsheet we can use the COUNTIF function to find that out, or we can combine the COUNT and WHERE queries in SQL to find out how many rows match our search criteria. This will give us similar results, but works with a much larger and more complex set of data.](https://www.coursera.org/learn/process-data/lecture/Y15Mz/spreadsheets-versus-sql)

[Next, let's talk about how spreadsheets and SQL are different.](https://www.coursera.org/learn/process-data/lecture/Y15Mz/spreadsheets-versus-sql)

[First, it's important to understand that spreadsheets and SQL are different things. Spreadsheets are generated with a program like Excel or Google Sheets. These programs are designed to execute certain built-in functions. SQL on the other hand is a language that can be used to interact with database programs, like Oracle MySQL or Microsoft SQL Server.](https://www.coursera.org/learn/process-data/lecture/Y15Mz/spreadsheets-versus-sql)

[The differences between the two are mostly in how they're used. If a data analyst was given data in the form of a spreadsheet they'll probably do their data cleaning and analysis within that spreadsheet, but if they're working with a large data set with more than a million rows or multiple files within a database, it's easier, faster and more repeatable to use SQL.](https://www.coursera.org/learn/process-data/lecture/Y15Mz/spreadsheets-versus-sql)

[SQL can access and use a lot more data because it can pull information from different sources in the database automatically, unlike spreadsheets which only have access to the data you input. This also means that data is stored in multiple places. A data analyst might use spreadsheets stored locally on their hard drive or their personal cloud when they're working alone, but if they're on a larger team with multiple analysts who need to access and use data stored across a database, SQL might be a more useful tool. Because of these differences, spreadsheets and SQL are used for different things.](https://www.coursera.org/learn/process-data/lecture/Y15Mz/spreadsheets-versus-sql)

[As you already know, spreadsheets are good for smaller data sets and when you're working independently. Plus, spreadsheets have built-in functionalities, like spell check that can be really handy. SQL is great for working with larger data sets, even trillions of rows of data. Because SQL has been the standard language for communicating with databases for so long, it can be adapted and used for multiple database programs. SQL also records changes in queries, which makes it easy to track changes across your team if you're working collaboratively. Next, we'll learn more queries and functions in SQL that will give you some new tools to work with. You might even learn how to use spreadsheet tools in brand new ways. See you next time.](https://www.coursera.org/learn/process-data/lecture/Y15Mz/spreadsheets-versus-sql)

**[SQL DIALECTS AND THEIR USES](https://www.coursera.org/learn/process-data/supplement/gTnKd/sql-dialects-and-their-uses)**

[In this reading, you will learn more about SQL dialects and some of their different uses. As a quick refresher, **Structured Query Language**, or SQL, is a language used to talk to databases. Learning SQL can be a lot like learning a new language—including the fact that languages usually have different dialects within them. Some database products have their own variant of SQL, and these different varieties of SQL dialects are what help you communicate with each database product.](https://www.coursera.org/learn/process-data/supplement/gTnKd/sql-dialects-and-their-uses)

[These dialects will be different from company to company and might change over time if the company moves to another database system. So, a lot of analysts start with Standard SQL and then adjust the dialect they use based on what database they are working with. Standard SQL works with a majority of databases and requires a small number of syntax changes to adapt to other dialects.](https://www.coursera.org/learn/process-data/supplement/gTnKd/sql-dialects-and-their-uses)

[As a junior data analyst, it is important to know that there are slight differences between dialects. But by mastering Standard SQL, which is the dialect you will be working with in this program, you will be prepared to use SQL in any database.](https://www.coursera.org/learn/process-data/supplement/gTnKd/sql-dialects-and-their-uses)

## **[More information](https://www.coursera.org/learn/process-data/supplement/gTnKd/sql-dialects-and-their-uses)**

[You may not need to know every SQL dialect, but it is useful to know that these different dialects exist. If you are interested in learning more about SQL dialects and when they are used, you can check out these resources for more information:](https://www.coursera.org/learn/process-data/supplement/gTnKd/sql-dialects-and-their-uses)

* [LearnSQL’s blog,](https://www.coursera.org/learn/process-data/supplement/gTnKd/sql-dialects-and-their-uses) [**What Is a SQL Dialect, and Which One Should You Learn?**](https://learnsql.com/blog/what-sql-dialect-to-learn/)
* [Software Testing Help’s article,](https://www.coursera.org/learn/process-data/supplement/gTnKd/sql-dialects-and-their-uses) [**Differences Between SQL Vs MySQL vs SQL Server**](https://www.softwaretestinghelp.com/sql-vs-mysql-vs-sql-server/)

[Datacamp’s blog,](https://www.coursera.org/learn/process-data/supplement/gTnKd/sql-dialects-and-their-uses) [**SQL Server, PostgreSQL, MySQL... what's the difference? Where do I start?**](https://www.datacamp.com/community/blog/sql-differences)

* [Note that there is an error in this blog article. The comparison table incorrectly states that SQlite uses subqueries instead of window functions. Refer to the](https://www.coursera.org/learn/process-data/supplement/gTnKd/sql-dialects-and-their-uses) [**SQLite Window Functions**](https://sqlite.org/windowfunctions.html)[documentation for proper clarification.](https://www.coursera.org/learn/process-data/supplement/gTnKd/sql-dialects-and-their-uses)
* [SQL Tutorial’s tutorial,](https://www.coursera.org/learn/process-data/supplement/gTnKd/sql-dialects-and-their-uses) [**What is SQL**](https://www.sqltutorial.org/what-is-sql/)

**[REVIEW: SET UP YOUR BIGQUERY ACCOUNT](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)**

[**Note:** This reading is also in Courses 3 and 5 of this program. If you’re taking the courses in order, you may either review it or move on to the next new course item,](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account) [Hands-On Activity: Processing time with SQL](https://www.coursera.org/learn/process-data/quiz/9fIb9/hands-on-activity-processing-time-with-sql)[. If you haven’t taken Courses 3 or 5 you should complete this reading before proceeding to the next course item.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

[As you’ve been learning, BigQuery is a database you can use to access, explore, and analyze data from many sources. Now, you’ll begin using BigQuery, which will help you gain SQL knowledge by typing out commands and troubleshooting errors. This reading will guide you through the process of setting up your very own BigQuery account.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

[**Note:** Working with BigQuery is not a requirement of this program. Additional resources for other SQL database platforms are also provided at the end of this reading if you choose to use them instead.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

## 

## **[BigQuery account options](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)**

[BigQuery offers a variety of account tiers to cater to various user needs and has two free-of-charge entry points, a sandbox account and a free-of-charge trial account. These options allow you to explore the program before selecting the best choice to suit your needs. A sandbox account allows you to practice writing queries and to explore public datasets free of charge, but it has](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account) [quotas and limits](https://cloud.google.com/bigquery/quotas)[, as well as some additional](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account) [restrictions](https://cloud.google.com/bigquery/docs/sandbox#limits)[. If you prefer to use BigQuery with the standard limits, you can set up a free-of-charge trial account instead. The free-of-charge trial is a trial period prior to paying for a subscription. In this instance, there is no automatic charge, but you will be asked for payment information when you create the account.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

[This reading provides instructions for setting up either account type. An effective first step is to begin with a sandbox account and switch to a free-of-charge trial account when needed to run the SQL presented upcoming courses.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

### **[Sandbox account](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)**

[The sandbox account is available at no cost, and anyone with a Google account can use it. However, it does have some limitations. For instance, you are limited to a maximum of 12 projects at a time. This means that, to create a 13th project, you'll need to delete one of your existing 12 projects. Additionally, the sandbox account doesn't support all operations you’ll do in this program. For example, there are limits on the amount of data you can process and you can’t insert new records into a database or update the values of existing records. However, a sandbox account is perfect for most program activities, including all of the activities in this course. Additionally, you can convert your sandbox account into a free-of-charge trial account at any time.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

**[Set up your sandbox account](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)**

[To set up a sandbox account:](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

1. [Visit the](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account) [BigQuery sandbox documentation](https://cloud.google.com/bigquery/docs/sandbox#limits) [page.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)
2. [Log in to your preferred Google account by selecting the profile icon in the BigQuery menu bar.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)
3. [Select the **Go to BigQuery** button on the documentation page.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)
4. [You'll be prompted to select your country and read the terms of service agreement.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)
5. [This will bring you to the **SQL Workspace**, where you'll be conducting upcoming activities. By default, BigQuery creates a project for you.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

[After you set up your account, the name of the project will be in the banner in your BigQuery console.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

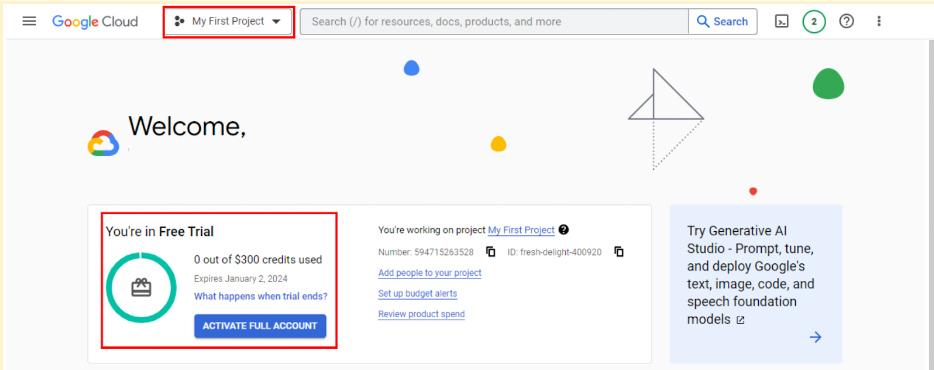
### **[Free-of-charge trial](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)**

[If you wish to explore more of BigQuery's capabilities with fewer limitations, consider the Google Cloud Free Trial. It provides you with $300 in credit for Google Cloud usage during the first 90 days. If you're primarily using BigQuery for SQL queries, you're unlikely to come close to this spending limit. After you've used up the $300 credit or after 90 days, your free trial will expire, and you will only be able to use this account if you pay to do so. Google won't automatically charge your payment method when the trial ends. However, you'll need to set up a payment option with Google Cloud. This means that you’ll need to enter your financial information. Rest assured, it won't charge you unless you consciously opt to upgrade to a paid account. If you're uncomfortable providing payment information, don't worry; you can use the BigQuery sandbox account instead.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

**[Set up your free-of-charge trial](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)**

1. [Go to the](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account) [BigQuery](https://cloud.google.com/bigquery) [page.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)
2. [Select **Try BigQuery free**.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)
3. [Log in using your Google email, or create an account free of charge if you don't have one.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account) [Click here](https://cloud.google.com/bigquery?utm_source=google&utm_medium=cpc&utm_campaign=na-US-all-en-dr-bkws-all-all-trial-e-dr-1605212&utm_content=text-ad-none-any-DEV_c-CRE_665665924750-ADGP_Hybrid+%7C+BKWS+-+MIX+%7C+Txt_BigQuery-KWID_43700077225652770-kwd-274188433361&utm_term=KW_bigquery%20account-ST_bigquery+account&gclid=CjwKCAjwkNOpBhBEEiwAb3MvvYQXjIQ4TRnkITJoSXz7DFez4T-XKPG5IpfKmxUg2iHPEmiJBNQByhoCLVgQAvD_BwE&gclsrc=aw.ds) [to create an account.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)
4. [Select your country, a description of your organization or needs, and the checkbox to accept the terms of service, Then select **CONTINUE**.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)
5. [Enter your billing information and select **START MY FREE TRIAL**.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

[After you set up your account, your first project, titled **My First Project** will be in the banner.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

[](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

### [**Transferring between BigQuery accounts**](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

[With either a sandbox or free-of-charge trial account, you have the flexibility to upgrade to a paid account at any time. If you upgrade, all your existing projects will be retained and transferred to your new account. If you started with a free-of-charge trial, but choose not to upgrade when it ends, you can switch to a sandbox account. However, note that projects from your trial won't transfer to your sandbox. Essentially, creating a sandbox is like starting from scratch.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

## **[Get started with other databases (if not using BigQuery)](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)**

[It’s easiest to follow along with the course activities if you use BigQuery, but you may use other SQL platforms, if you prefer. If you decide to practice SQL queries on other database platforms, here are some resources to get started:](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

* [Getting Started with MySQL](https://dev.mysql.com/doc/mysql-getting-started/en/)
* [Getting Started with Microsoft SQL Server](https://docs.microsoft.com/en-us/sql/relational-databases/tutorial-getting-started-with-the-database-engine?view=sql-server-ver15)
* [Getting Started with PostgreSQL](https://www.postgresql.org/docs/10/tutorial-start.html)
* [Getting Started with SQLite](https://www.sqlite.org/quickstart.html)

## **[Key takeaways](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)**

[BigQuery offers multiple account options. Keep the following in mind when you choose an account type:](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

* [**Account tiers:** BigQuery provides various account tiers to cater to a wide range of user requirements. Whether you're starting with a sandbox account or exploring a paid account with the free-of-charge trial option, BigQuery offers flexibility to choose the option that aligns best with your needs and budget.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)
* [**Sandbox limitations:** While a sandbox account is a great starting point, it comes with some limitations, such as a cap on the number of projects and restrictions on data manipulation operations like inserting or updating records, which you will encounter later in this program. Be aware of these limitations if you choose to work through this course using a sandbox account.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)
* [**Easy setup and upgrades:** Getting started with any BigQuery account type is quick and easy. And if your needs evolve, you have the flexibility to modify your account status at any time. Additionally, projects can be retained even when transitioning between account types.](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

[Choose the right BigQuery account type to match your specific needs and adapt as your requirements change!](https://www.coursera.org/learn/process-data/supplement/GaWA7/review-set-up-your-bigquery-account)

[**REVIEW: GET STARTED WITH BIG QUERY**](https://www.coursera.org/learn/process-data/supplement/0jd6x/review-get-started-with-bigquery)

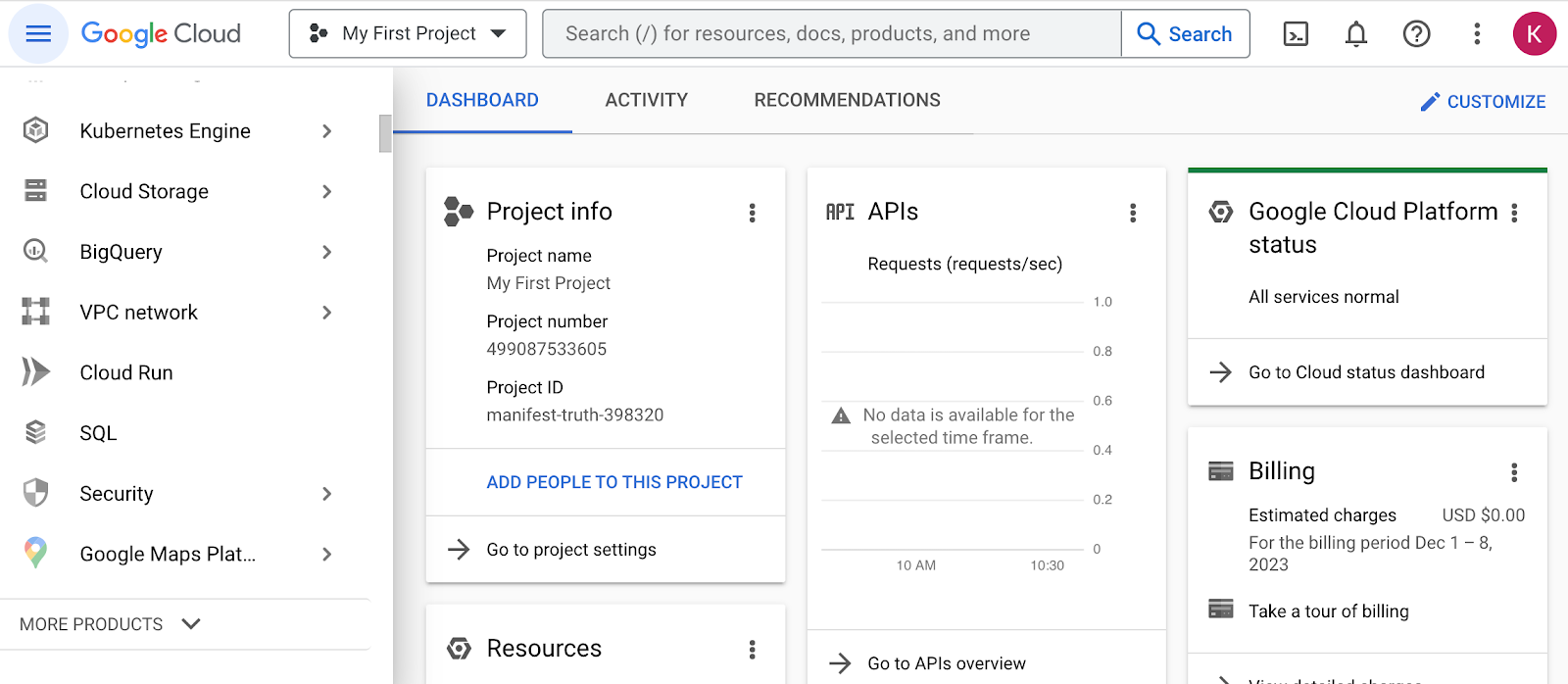
**Note:** This reading is also in Courses 3 and 5 of this program. If you’re taking the courses in order, you may either review it or move on to the next course item, [Hands-On Activity: Processing time with SQL](https://www.coursera.org/learn/process-data/quiz/9fIb9/hands-on-activity-processing-time-with-sql). If you haven’t taken Courses 3 or 5 you should complete this reading before proceeding to the next course item.

BigQuery is a data warehouse on the Google Cloud Platform used to query and filter large datasets, aggregate results, and perform complex operations. Throughout this program, you’re going to use BigQuery to practice your SQL skills and collect, prepare, and analyze data. At this point, you have set up your own account. Now, explore some of the important elements of the SQL workspace. This will prepare you for the upcoming activities in which you will use BigQuery. Note that BigQuery updates its interface frequently, so your console might be slightly different from what is described in this reading. That’s okay; use your troubleshooting skills to find what you need!

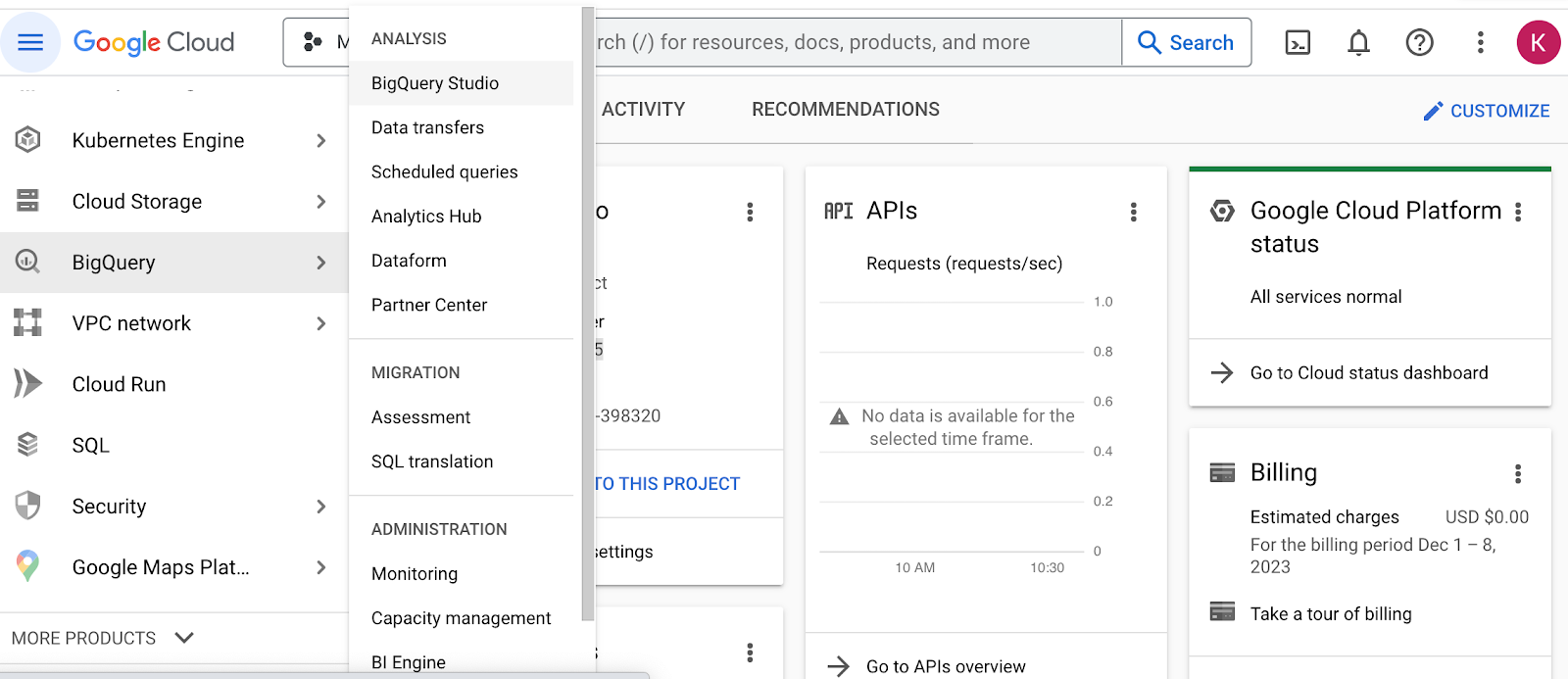
## 

## **Log in to BigQuery**

When you log in to BigQuery using the landing page, you will automatically open your project space. This is a high-level overview of your project, including the project information and the current resources being used. From here, you can check your recent activity.



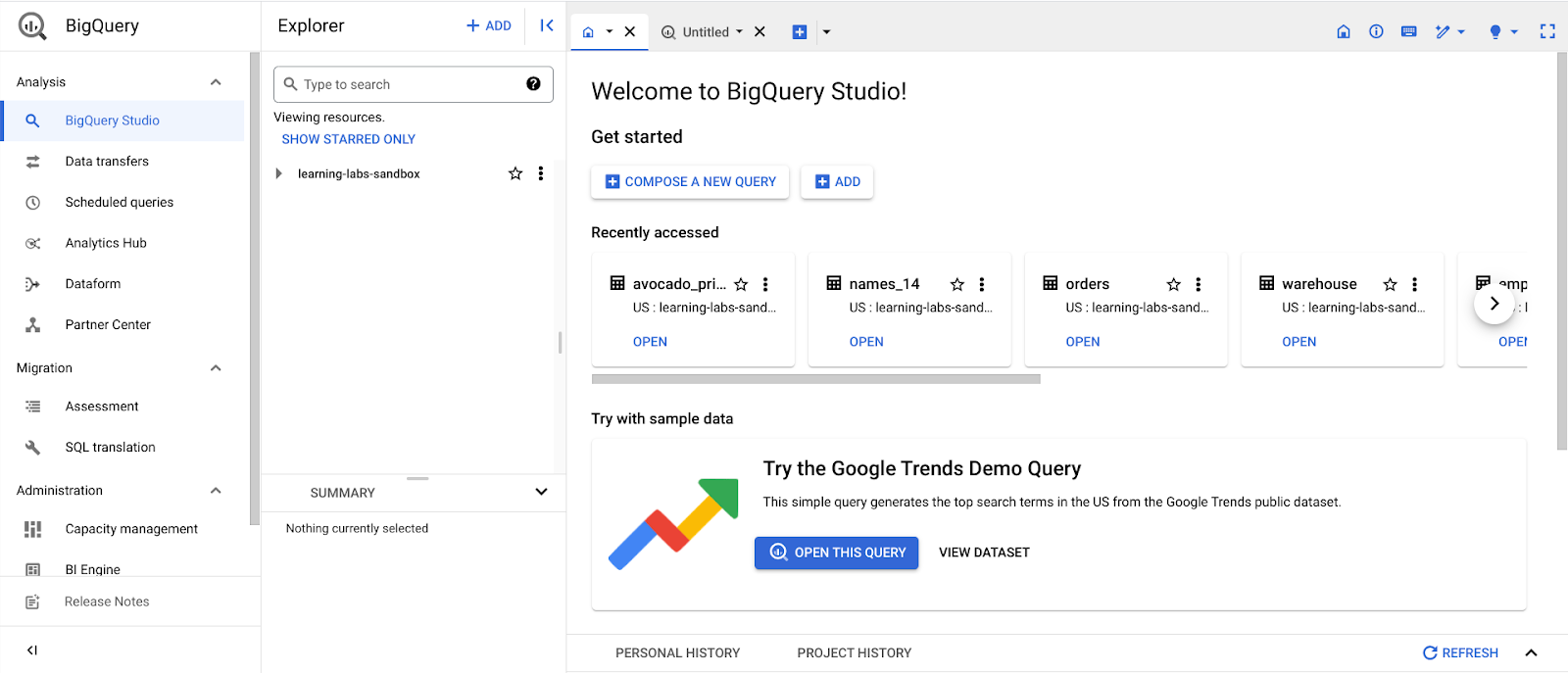
Navigate to your project’s BigQuery Studio by selecting BigQuery from the navigation menu and BigQuery Studio from the dropdown menu.



## 

## **BigQuery Studio components**

Once you have navigated to BigQuery from the project space, most of the major components of the BigQuery console will be present: the **Navigation** pane, the **Explorer** pane, and the **SQL Workspace**.

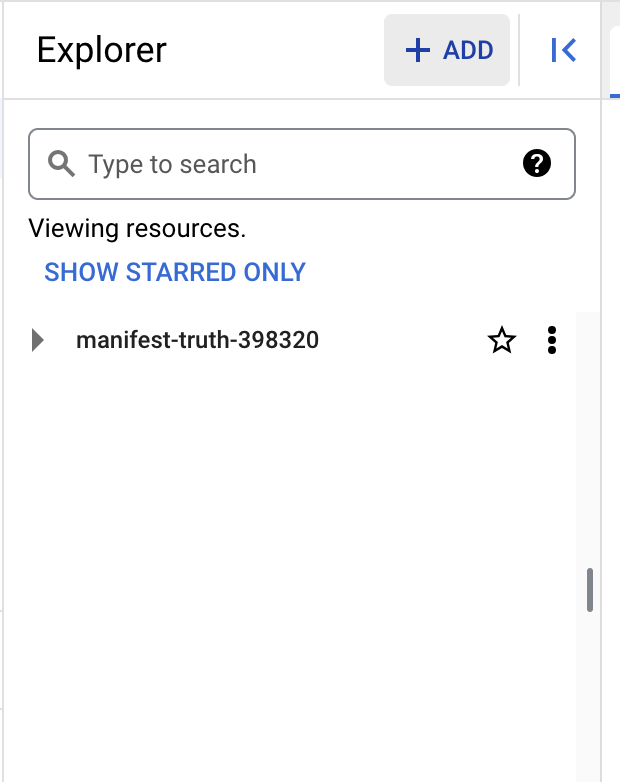


### **The Navigation pane**

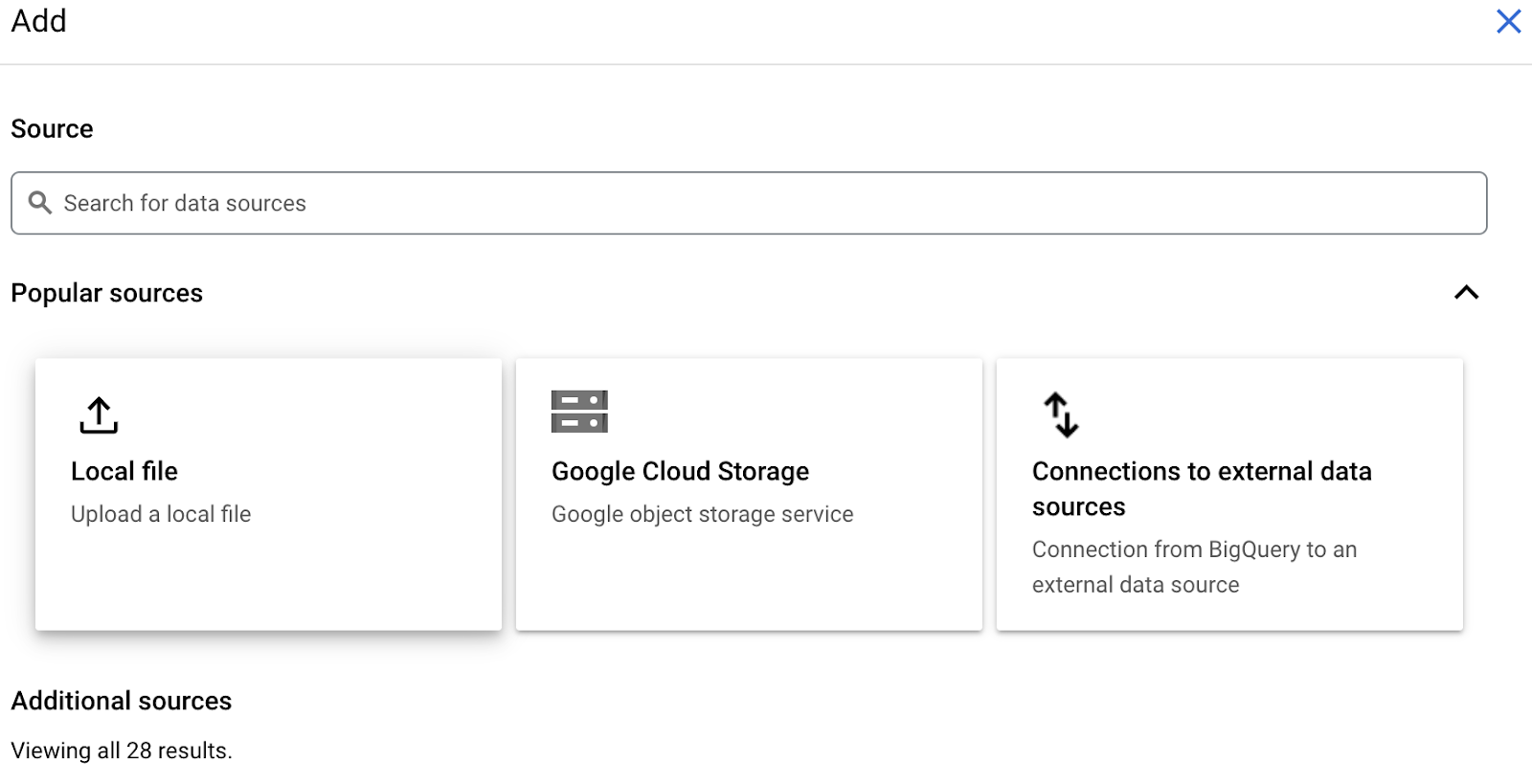
On the console page, find the **Navigation** pane. This is how you navigate from the project space to the BigQuery tool. This menu also contains a list of other Google Cloud Project (GCP) data tools. During this program, you will focus on BigQuery, but it’s useful to understand that the GCP has a collection of connected tools data professionals use every day.

### **The Explorer pane**

The **Explorer** pane lists your current projects and any starred projects you have added to your console. It’s also where you’ll find the **+ ADD** button, which you can use to add datasets.

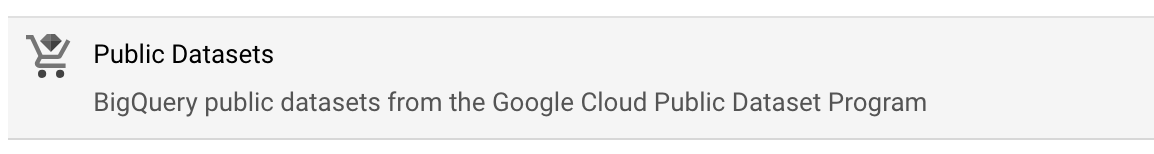


This button opens the **Add** dialog that allows you to open or import a variety of datasets.

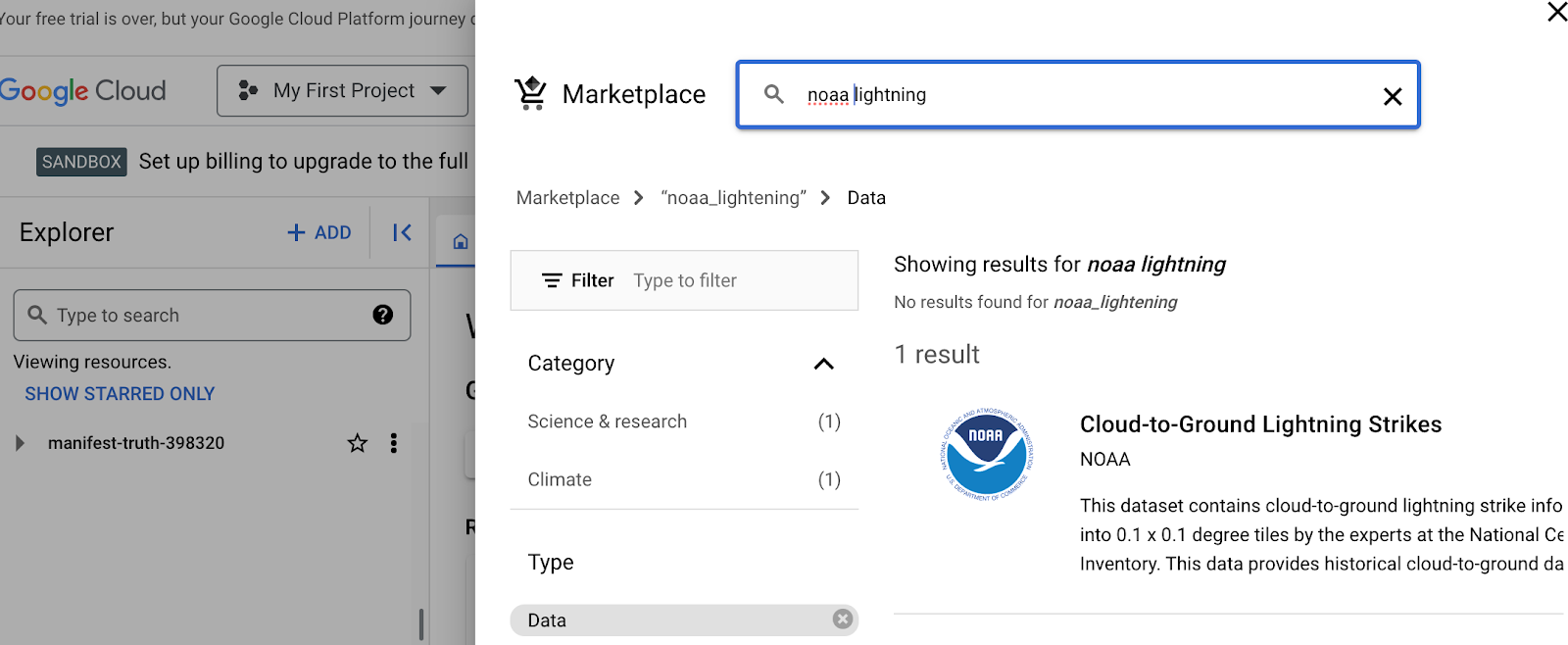


### **Add Public Datasets**

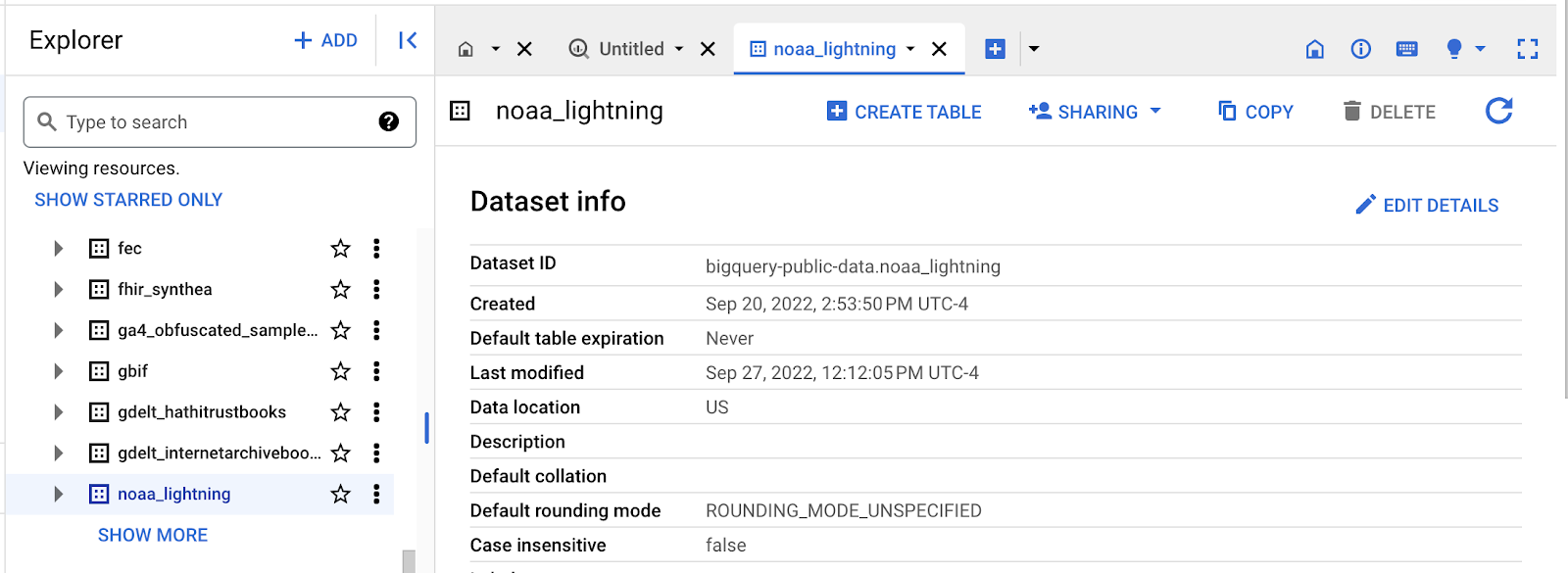
BigQuery offers a variety of public datasets from the Google Cloud Public Dataset Program. Scroll down the **Add** dialog to the **Public Datasets** option.



Select **Public Datasets**. This takes you to the **Public Datasets Marketplace**, where you can search for and select public datasets to add to your BigQuery console. For example, search for the "noaa lightning" dataset in the Marketplace search bar. When you search for this dataset, you will find NOAA’s Cloud-to-Ground Lightning Strikes data.



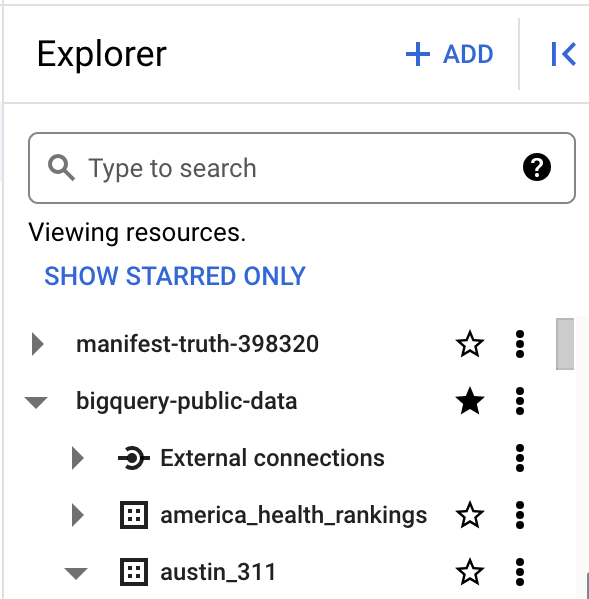
Select the dataset to read its description. Select **View dataset** to create a tab of the dataset’s information within the SQL workspace.



The Explorer Pane lists the noaa\_lightning and other public datasets.

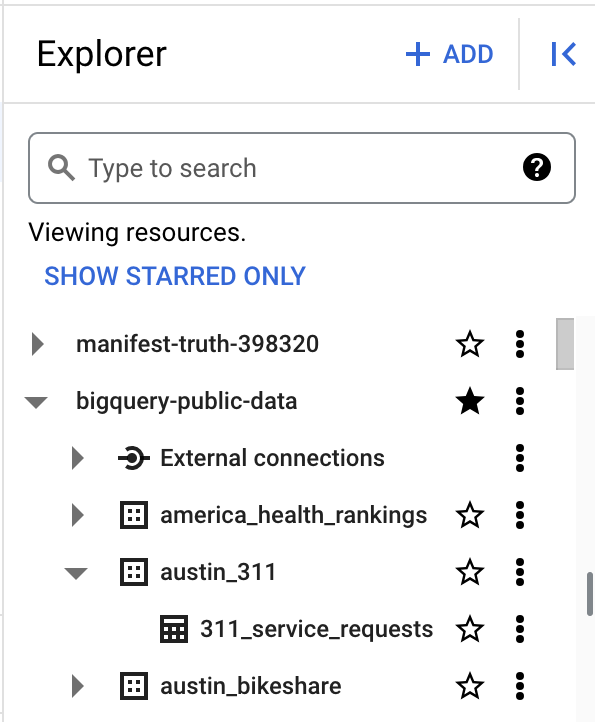
### **Star and examine Public Datasets**

You added the public **noaa\_lightning** dataset to your BigQuery Workspace, so the **Explorer** pane displays the **noaa\_lightning** dataset, along with the list of other public datasets. These datasets are nested under **bigquery-public-data**. Star **bigquery-public-data** by navigating to the top of the **Explorer** pane and selecting the star next to **bigquery-public-data**.



Starring **bigquery-public-data** will enable you to search for and add public datasets by scrolling in the **Explorer** pane or by searching for them in the **Explorer** search bar.

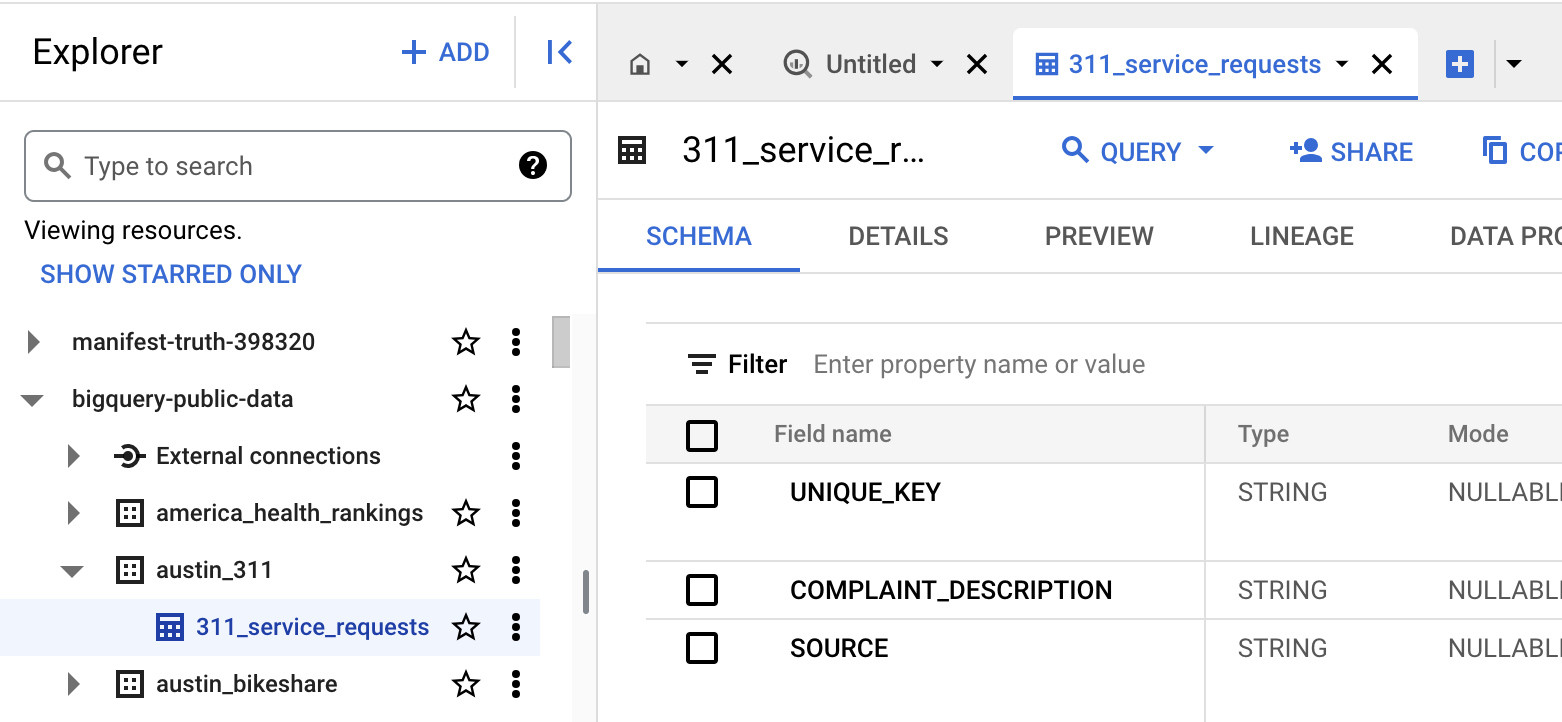
For example, you might want to select a different public dataset. If you select the second dataset, **austin\_311**, it will expand to list the table stored in it, **311\_service\_requests**.



The Explorer pane with the “bigquery-public data” and “austin\_311” datasets expanded, revealing the “311\_service\_requests” table

When you select a table, its information is displayed in the SQL Workspace. Select the **311\_service\_requests** table to examine several tabs that describe it, including:

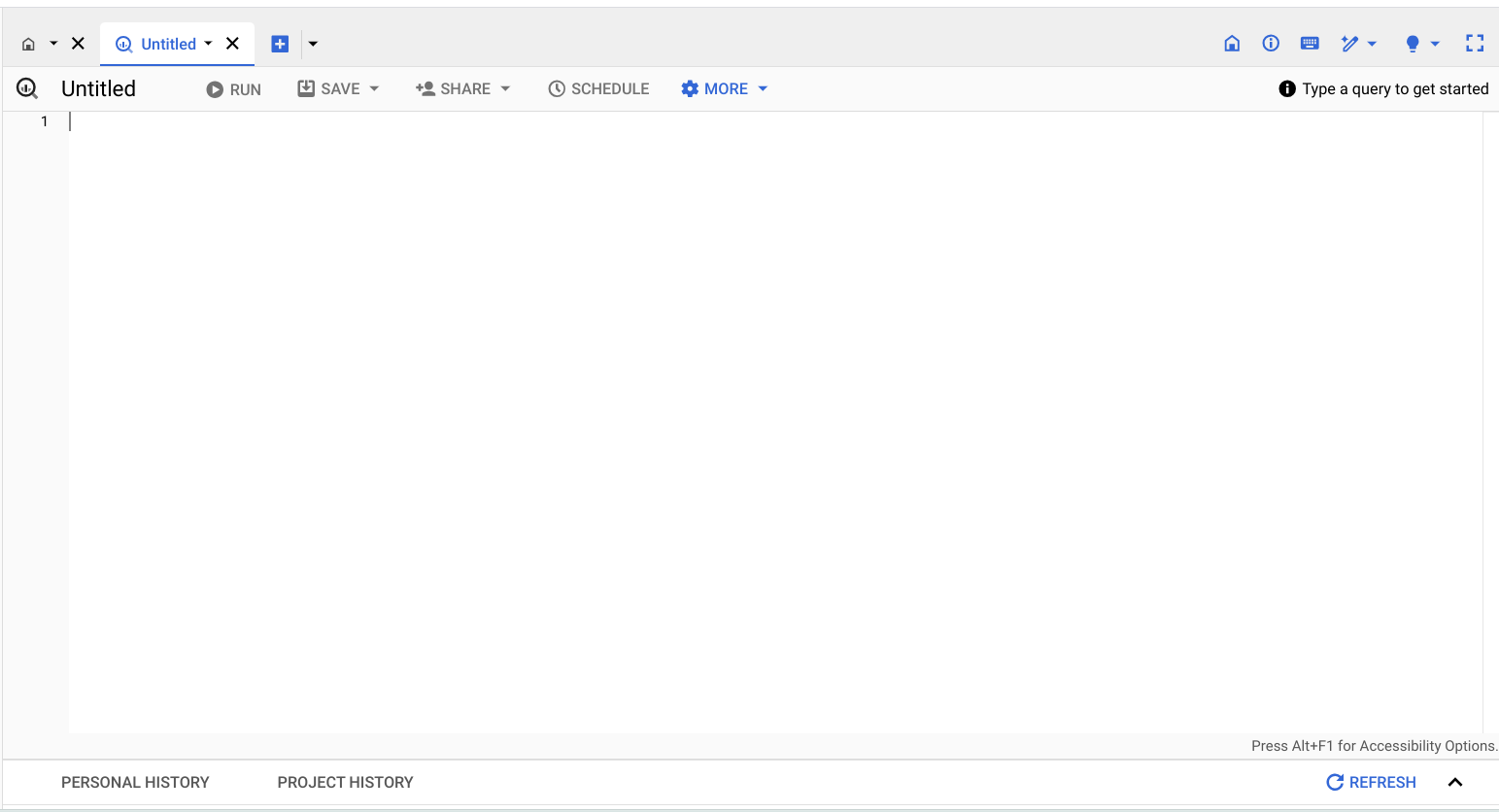
* **Schema**, which displays the column names in the dataset
* **Details**, which contains additional metadata, such as the creation date of the dataset
* **Preview**, which shows the first rows from the dataset



Additionally, you can select the **Query** button from the menu bar in the SQL Workspace to query this table.

### **The SQL Workspace**

The final menu pane in your console is the SQL Workspace. This is where you will actually write and execute queries in BigQuery.



The SQL Workspace also gives you access to your personal and project history, which stores a record of the queries you’ve run. This can be useful if you want to return to a query to run it again or use part of it in another query.

## **Upload your data**

In addition to offering access to public datasets, BigQuery also gives you the ability to upload your own data directly into your workspace. Access this feature by opening the **+ ADD** menu again or by clicking the three vertical dots next to your project’s name in the Explorer pane. This will give you the option to create your own dataset and upload your own tables. You will have the opportunity to upload your own data in an upcoming activity to practice using this feature!

## **Key takeaways**

BigQuery's SQL workspace allows you to search for public datasets, run SQL queries, and even upload your own data for analysis. Whether you're working with public datasets, running SQL queries, or uploading your own data, BigQuery’s SQL workspace offers a range of features to support all kinds of data analysis tasks. Throughout this program, you will be using BigQuery to practice your SQL skills, so being familiar with the major components of your BigQuery console will help you navigate it effectively in the future!

**[HANDS-ON ACTIVITY: PROCESSING TIME WITH SQL](https://www.coursera.org/learn/process-data/quiz/9fIb9/hands-on-activity-processing-time-with-sql)**

## 

## **Activity Overview**

****

In this activity, you’ll explore how the amount of data processed by a SQL query affects how long it takes the query to run.

By the time you complete this activity, you’ll be familiar with the different units used to measure data quantity. This will help you understand how dataset size affects the amount of time queries take to run and how valuable tools like SQL can be to data analysts.

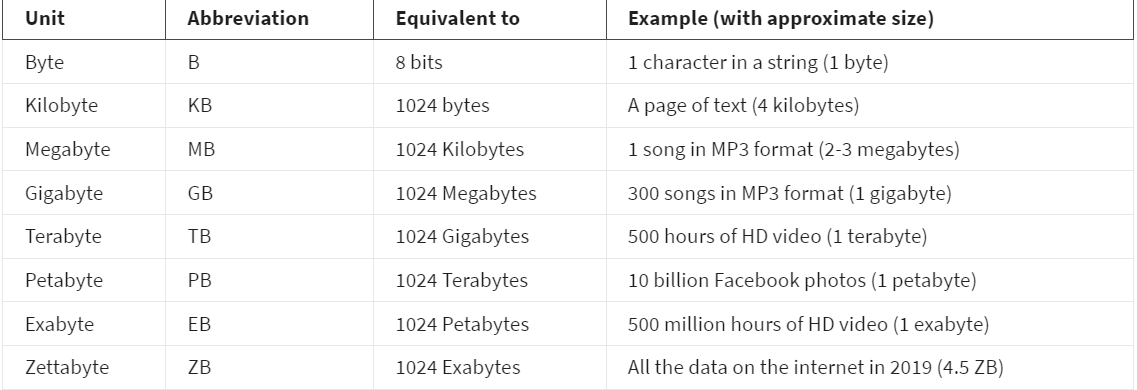
### 

### **Step-By-Step Instructions**

### **Step 1: Understand how data is measured**

All information in a computer is represented as a binary number consisting solely of 0’s and 1’s. Each 0 or 1 in a number is a bit, which is the smallest unit of storage in computers. Data is measured by the number of bits it takes to represent it. This is then described in bytes, which are equal to 8 bits.

Take a moment to examine the table below to understand each data measurement and its size relative to the others..



**Step 2: Relate to the amount of data in the world**

Now that you’ve explored data measurements, think about the amount of data in the world. It’s growing at an incredible pace largely due to the more than 5.3 billion people in the world connected to the internet (as of November 2023). Smartphones and other internet-connected devices generate a staggering amount of new data. Many experts believe that the amount of all the data on the internet will swell to 175 ZB by the end of 2025!

**Dataset size is important**

The size of the dataset you’re working with usually determines which tool—spreadsheets or SQL—is best suited for the task. Spreadsheets often start to have performance issues as dataset sizes increase beyond a few megabytes. SQL databases are much better at working with larger datasets that have billions of rows with sizes measured in gigabytes. Yet the dataset’s size still matters here: Even in SQL, it takes longer for queries to complete when they’re run on longer datasets, depending on the query’s content and the number of rows SQL has to process.

### **Step 3: Prepare to query large datasets**

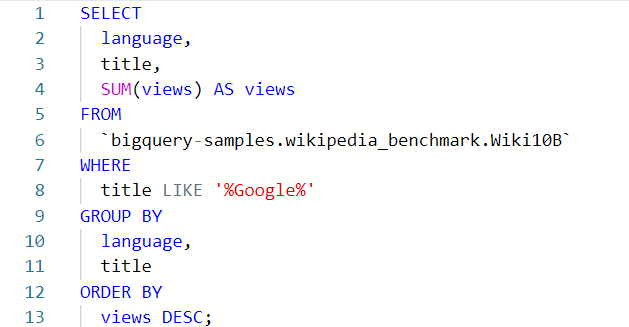
You’ll now discover for yourself how query runtimes change with dataset size by running some queries on a huge dataset—Wikipedia!

1. On the [Enable the BigQuery sandbox](https://cloud.google.com/bigquery/docs/sandbox) page, select Go to BigQuery. If you have a free trial version of BigQuery, you can use that instead.
   1. Note: BigQuery Sandbox frequently updates its user interface. The latest changes may not be reflected in the screenshots presented in this activity, but the principles remain the same. Adapting to changes in software updates is an essential skill for data analysts, and it’s helpful for you to practice troubleshooting. You can also reach out to your community of learners on the discussion forum for help.
2. The main section is the home screen from which you can access the Query Editor. You can navigate to different projects and data sets available to you using the Explorer menu.
3. Select Compose a new query so that you can work through an example query.

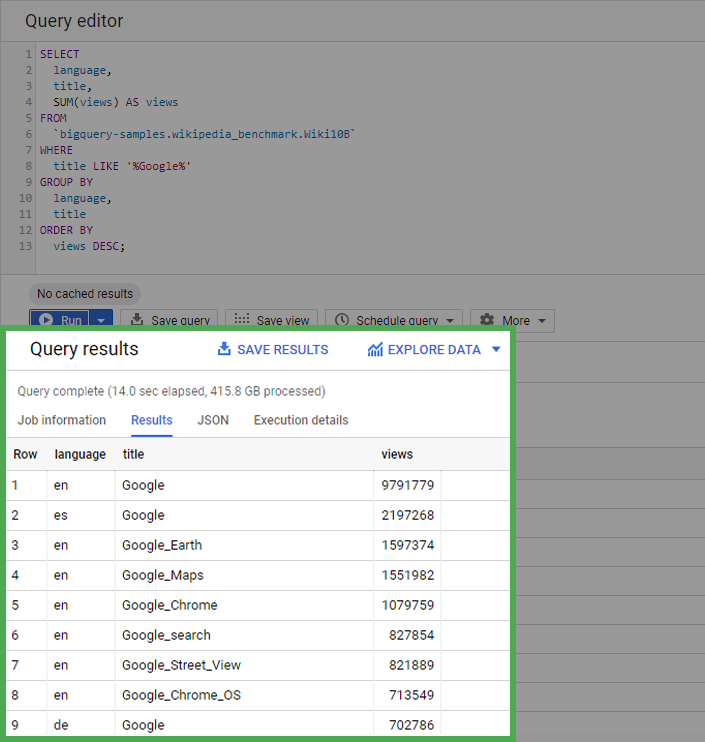
**Step 4: Run a large query**

1. Copy and paste the following query into the Query editor. Select Run to run it. The formatting improves readability, but it’s okay if it changes when copied over—it won’t affect how your code runs.

This query sorts and filters data from the dataset **bigquery-samples.wikipedia\_benchmark.Wiki10B**, which is a sample from the Wikipedia public dataset that contains 10 billion rows.



**Note:** Later in this course and program, you will learn what each part of this query means and how to use its functions in your own work.

After the query finishes, you will get a table that displays the total number of times each Wikipedia page with “Google” in the title has been viewed in each language.

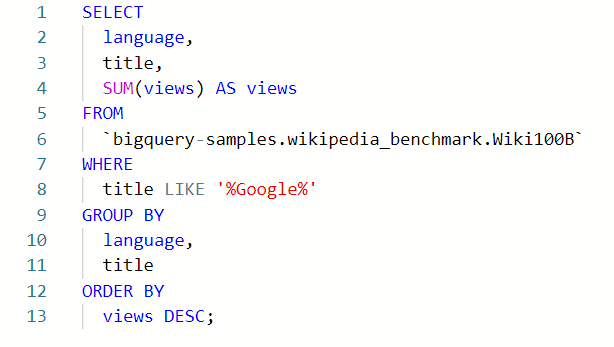
2. Note the information that BigQuery provides on the query you just ran. (Remember, many of the public databases on BigQuery are living records and, as such, are periodically updated with new data. Throughout this course (and others in this certificate program), if your results differ from those you encounter in videos or screenshots, there's a good chance it is due to a data refresh.)

You’ll find that the query processes more than 415 gigabytes of data when run—very impressive for 15 seconds! If you run the query on this dataset again, the runtime will be almost instant (as long as you haven’t changed the default caching settings). This is because BigQuery caches (stores in the background) the query results to avoid extra work if the query needs to be rerun.

### **Step 5: Run a larger query**

Now, run the same query on a 100-billion-row version of the Wikipedia dataset. Copy and paste the following query into the editor and run it:

Note: This query will only run in the free trial account, not in the sandbox version of BigQuery. If you use a sandbox account, use the results presented below.



After the query finishes, you will get a table that displays the total number of times each Wikipedia page with “Google” in the title has been viewed in each language.

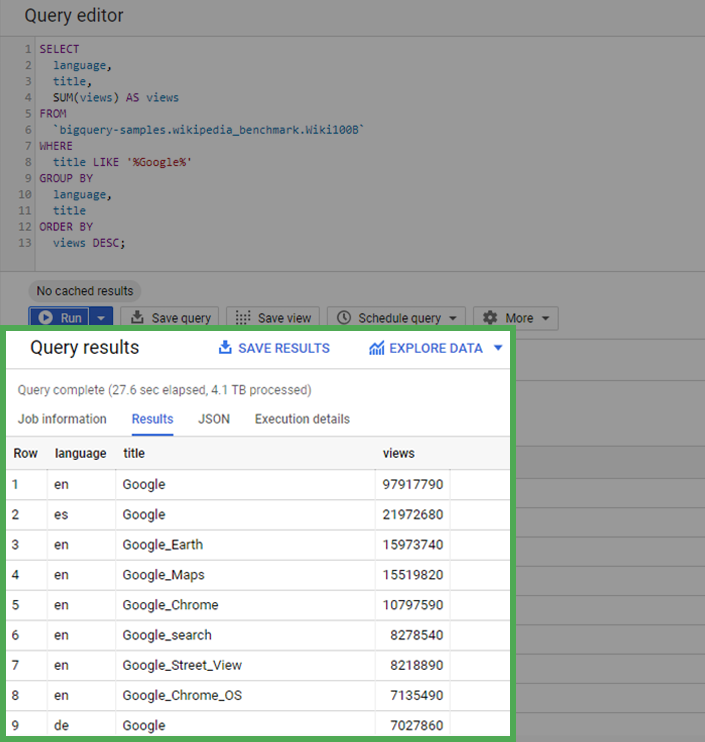


Image of the query results indicating that the query was completed in 27.6 seconds and processed 4.1 TB of data. The table returned by the query includes the rows language, title, and views.

Notice that this query takes longer to run than the first query, at least 25-30 seconds. At 100 billion rows, the query processed 4.1 terabytes of data!

**LEARN BASIC SQL QUERIES**

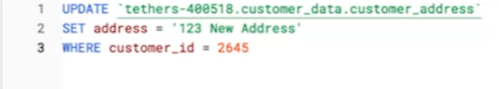
[**USE SQL TO CLEAN DATA**](https://www.coursera.org/learn/process-data/lecture/w6EFG/use-sql-to-clean-data)

**[OPTIONAL: UPLOAD THE CUSTOMER DATASET TO BIGQUERY](https://www.coursera.org/learn/process-data/supplement/kt8qL/optional-upload-the-customer-dataset-to-bigquery)**

[**WIDELY USED SQL QUERIES**](https://www.coursera.org/learn/process-data/lecture/Ktoxq/widely-used-sql-queries)

Changing customer information:

**[](https://www.coursera.org/learn/process-data/lecture/Ktoxq/widely-used-sql-queries)**

Updating customer address:  


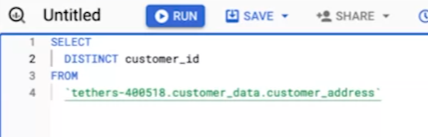
If we want to create a new table for this database, we can use the CREATE TABLE IF NOT EXISTS statement.

[**EVAN: HAVING FUN WITH SQL**](https://www.coursera.org/learn/process-data/lecture/P89Un/evan-having-fun-with-sql)

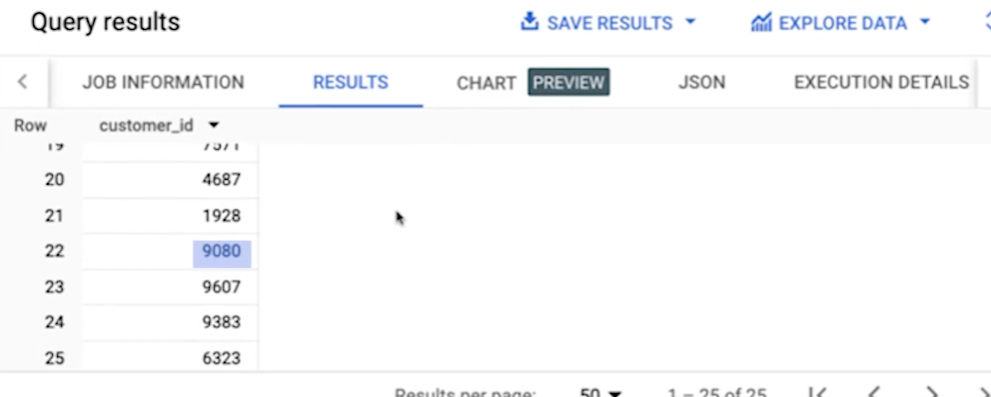
[**CLEAN STRING VARIABLES USING SQL**](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[It's so great to have you back. Now that we know some basic SQL queries and spent some time working in a database, let's apply that knowledge to something else we've been talking about; preparing and cleaning data. You already know that cleaning and completing your data before you analyze it is an important step. In this video, I'll show you some ways SQL can help you do just that. Including how to remove duplicates as well as four functions to help you clean string variables. Earlier we covered how to remove duplicates and spreadsheets using the Remove duplicates tool. In SQL, we can do the same thing by including Distinct in our Select statement.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[For example, let's say the company we work for has a special promotion for customers in Ohio. We want to get the customer IDs of customers who live in Ohio, but some customer information has been entered multiple times. We can get these customer IDs by writing, select customer\_id from customer\_data.customer\_address.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[This query will give us duplicates if they exist in the table. If customer ID 9080 shows up three times in our table, our results will have three of that customer ID, but we don't want that. We want a list of all unique customer IDs. To do that, we add Distinct to our Select statement by writing, Select Distinct customer\_ id from customer\_data.customer\_address.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[Now the customer ID 9080 will show up only once in our results.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[You might remember we talked before about text strings as a group of characters within a cell, commonly composed of letters, numbers or both. These text strings need to be clean sometimes.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql) 

[Maybe they've been entered differently in different places across your database, and now they don't match. In those cases, you'll need to clean them before you can analyze them. Here are some functions you can use in SQL to handle string variables. You might recognize some of these functions from when we talked about spreadsheets.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[Now it's time to see them work in a new way. Pull up the data set we shared right before this video, and you can follow along step by step with me during the rest of this video. The first function I want to show you is **length**, which we've encountered before. If we already know the length our string variables are supposed to be, we can use length to double check that our string variables are consistent. For some databases, this query is written as **LEN**, but it does the same thing.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[Let's say we're working with the customer\_address table from our earlier example, we can make sure that all country codes have the same length by using Length on each of these strings.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

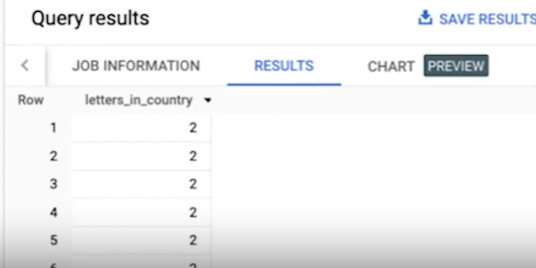
[To write our SQL query, let's first start with **Select** and **From**. We know our data comes from the customer\_address table within the customer\_data dataset. We add customer\_ data.customer \_address after the From clause.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)



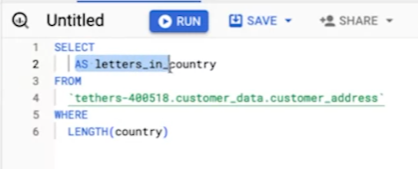
[Then under Select, we'll write Length, and then the column we want to check; country. To remind ourselves what this is, we can label this column in our results as letter\_in \_country. We add as letters\_in \_country after Length parenthesis country.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)



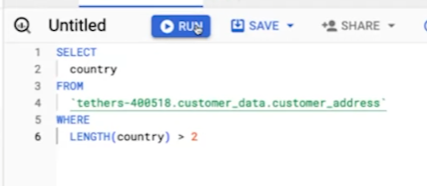
[The result we get is a list of the number of letters in each country listed for each of our customers. It seems that almost all of them are twos, which means the country field contains only two letters, but we notice one that has three.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

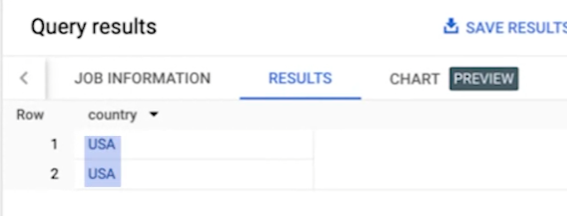


[That's not good. We want our data to be consistent. Let's check out which countries were incorrectly listed in our table. We can do that by putting the Length parenthesis, country parentheses function that we created into the Where clause, because we're telling SQL to filter the data to show only customers whose country contains more than two letters.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)



[Now we'll write, Select country from customer\_data.customer\_address, where Length parentheses, country parentheses greater than 2.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)



[When we run this query, we now get the two countries where the number of letters is greater than the 2 we expect to find. **The incorrectly listed countries show up as USA instead of US.**](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[If we created this table, then we could update our table so that this entry shows up as US instead of USA. But in this case, we didn't create this table, so we shouldn't update it.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[We still need to fix this problem, so we can pull a list of all the customers in the US, including the two that have USA instead of US.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

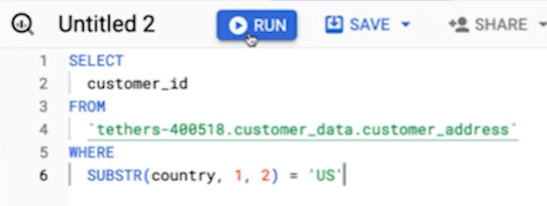
[The good news is that we can account for this error in our results by using the substring function in our SQL query. To write our SQL query, let's start by writing the **basic structure**. **Select**, **From**, **Where**. We know our data is coming from the customer\_address table from the customer\_data dataset. We type in customer\_data.customer \_ address. After from.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[Next, we tell SQL what data we want it to give us. We want all the customers in the US by their IDs, so we type into customer\_id after **SELECT**.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

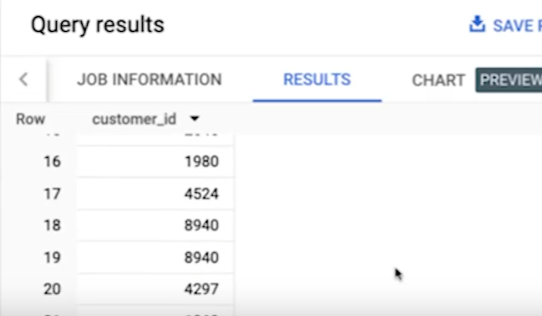
[Finally, we want SQL to filter out only American customers, so we use the **substring function** after the **where** clause.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[We're going to use the substring function to pull the first two letters of each country so that all of them are consistent and only contain two letters. To use the substring function, we first need to tell SQL, the column where we found this error country.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[Then we specify which letter to start with. We want SQL to pull the first two letters, so we're starting with the first letter, so we type in one. Then we need to tell SQL how many letters, including this first letter to pull. Since we want the first two letters, we need SQL to pull two total letters, so we type in two. This will give us the first two letters of each country. We want US only, so we'll set this function to equal US.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)



[When we run this query, we get a list of all customer IDs of customers whose country is the US, including the customers that had USA instead of US. Going through our results, it seems like we have a couple duplicates where the customer ID is shown multiple times.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)



**[Remember how we get rid of duplicates. We add distinct before customer underscore ID.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)**[](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[Now when we run this query, we have our final list of customer IDs of the customers who live in the US.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[Finally, let's check out the **trim function** which you've come across before. This is really useful if you find entries with extra spaces and need to eliminate those extra spaces for consistency.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[For example, let's check out the state column in our customer\_address table. Just like we did for the country column, we want to make sure the state column has a consistent number of letters. Let's use the **length function** again **to learn if we have any state that has more than two letters**, which is what we would expect to find in our data table.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[We start writing our SQL query by typing the **basic SQL structure** of , **select** **from where**.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[We're working with the customer\_address table in the customer\_data dataset, so we type in customer\_data, dot customer\_address after from.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)



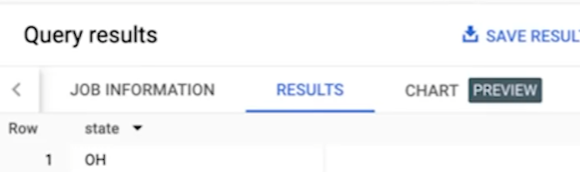
[Next, we tell SQL what we want it to pull. We want it to give us any state that has more than two letters, so we type in state after select.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)



[Finally, we want SQL to filter for states that have more than two letters. This condition is written in the **where** clause. We type in length, parentheses state, parentheses, and that it must be greater than two because we want the states that have more than two letters.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)



[We want to figure out what the incorrectly listed states look like, if we have any. When we run this query, we get one result. We have one state that has more than two letters.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)



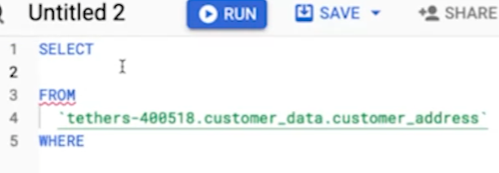
[But hold on. How can this state that seems like it has two letters, O and H for Ohio have more than two letters?](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[We know that there are more than two characters because we use the length parentheses state, parentheses greater than two statements in the where clause when filtering our results. **That means the extra characters that SQL is counting must then be a space**. There must be a space after the age. **This is where we would use the trim function.**](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[The trim function removes any spaces. Let's write a SQL query that accounts for this error.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[Let's say we want a list of all customer IDs of the customers who live in, OH, for Ohio. **We start with the basic SQL structure. Select from where**. We know the data comes from the customer\_address table, and the customer\_data dataset.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[We type in customer\_data.customer\_address after from.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)



[Next, we tell SQL what data we want. We want SQL to give us the customer IDs of customers who live in Ohio. We type in customer\_id after select. Since we know we have some duplicate customer entries, we'll go ahead and type in **distinct** before customer ID to remove any duplicate customer IDs from appearing in our results.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)



[Finally, we want SQL to give us the customer IDs of the customers who live in Ohio. We're asking SQL to **filter the data**. This belongs in the **where** clause. Here's where we'll use the trim function.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

[**To use the trim function, we tell SQL the column we want to remove spaces from, which is state in our case, and we want only Ohio customers, so we type in equals OH**.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)



[That's it. We have all customer IDs of the customers who live in Ohio, including that customer with the extra space after the H.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)



[Making sure that your string variables are complete and consistent will save you a lot of time later by avoiding errors or miscalculations. That's why we clean data in the first place. Hopefully, **functions like length, substring, and trim will give you the tools you need to start working with string variables in your own datasets**. Next up, we'll check out some other ways you can work with strings and more advanced cleaning functions. Then you'll be ready to start working in SQL on your own. See you soon.](https://www.coursera.org/learn/process-data/lecture/xVA4Z/clean-string-variables-using-sql)

**[HANDS-ON ACTIVITY: CLEAN DATA USING SQL](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)**

**[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)**

## **[Activity Overview](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)**

**[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)**

[In previous lessons, you learned about the importance of being able to clean your data where it lives. When it comes to data stored in databases, that means using SQL queries. In this activity, you will create a custom dataset and table, import a .csv file, and use SQL queries to clean automobile data.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql) [automobile\_data](https://docs.google.com/spreadsheets/d/1U3ktsROmhoCZG3Yz5xFLnjJmzm-MgBYYxwt4xcb6d3o/template/preview) from <https://archive.ics.uci.edu/dataset/10/automobile>

### 

### [Scenario](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[Review the following scenario. Then complete the step-by-step instructions.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[In this scenario, you are a data analyst working with a used car dealership startup venture. The investors want you to find out which cars are most popular with customers so they can make sure to stock accordingly.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[By the time you complete this activity, you will be able to clean data using SQL. This will enable you to process and analyze data in databases, which is a common task for data analysts.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

### 

### [Step-By-Step Instructions](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[Follow the instructions to complete each step of the activity. Then answer the questions at the end of the activity before going to the next course item.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

### 

### [Step 1: Access the template](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[To get started, download the *automobile\_data.csv* file. This is data from an external source that contains historical sales data on car prices and their features.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[Click the link to the *automobile\_data.csv* file to download it. Or you may download the .csv file directly from the attachments below.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

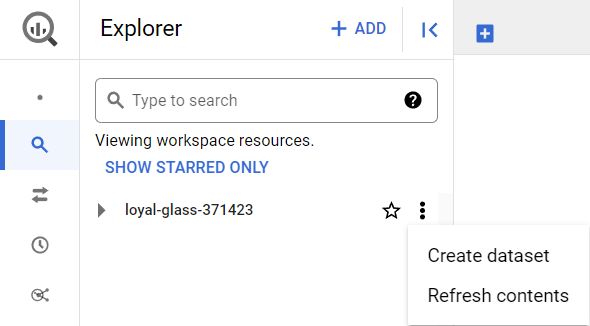
[Link to data:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql) [automobile\_data](https://docs.google.com/spreadsheets/d/1U3ktsROmhoCZG3Yz5xFLnjJmzm-MgBYYxwt4xcb6d3o/template/preview) [OR](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql) [automobile\_data CSV File](https://d3c33hcgiwev3.cloudfront.net/CHOsbSb3RYGzrG0m98WBiQ_f6e1579446f9464699fab3ea55ecb6f1_automobile_data.csv?Expires=1712361600&Signature=B6Rvd~rdJY4BfCgzpEiPBK6lDCx7VAGVZ~jYEmI9g-cPwc8KBtFsVsF9PfKqgZtiBPCgFjFNeBZN0N95cgomqb-adhvAUEFaIfbtWlDdmKhpkIJcXOo2BBxIIiE5grWXb6kxnMqoM0K71lHwyt~hbHvdpWmVECWi-9935lzjFhs_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

### 

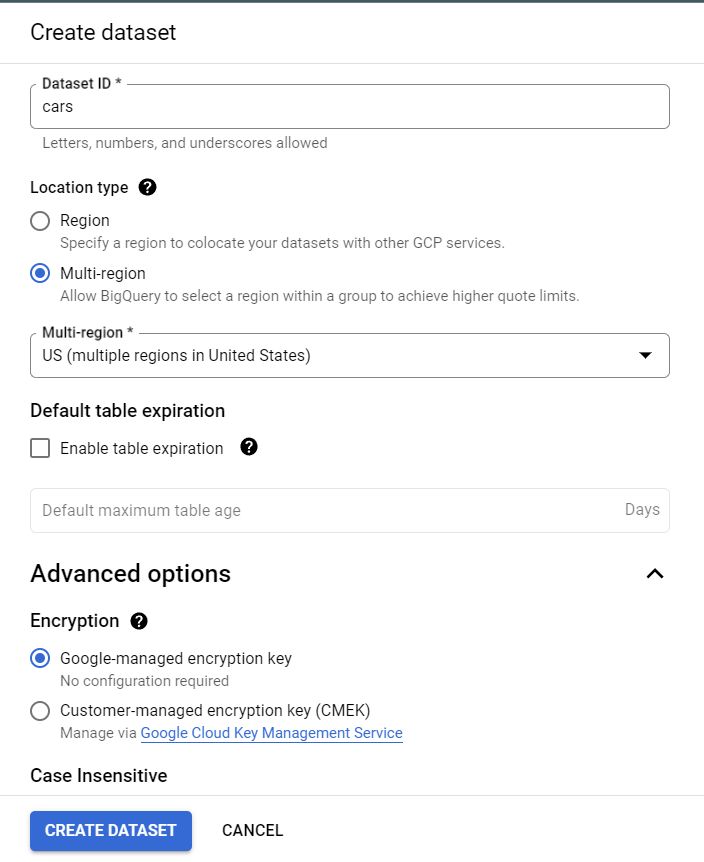
### [Step 2: Create a dataset](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[Once you’ve downloaded the **automobile\_data.csv** file, create your dataset.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

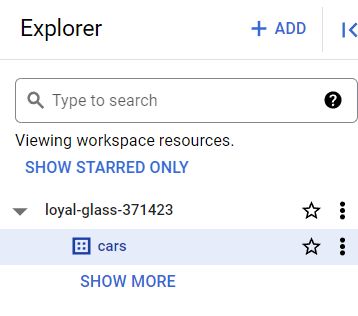
[Go to the Explorer pane in your workspace and click the three dots next to your personal project name to open the drop-down menu. From here, select Create dataset.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[From the Create dataset menu, fill out some information about the dataset. Input the Dataset ID as **cars** you can keep the Location type as Multi-region, US (multiple regions in United States), and the Encryption as Google-managed encryption key default settings. Then, click the CREATE DATASET button.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[The **cars** dataset should appear under your project in the Explorer pane as shown below.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

### 

### 

### 

### 

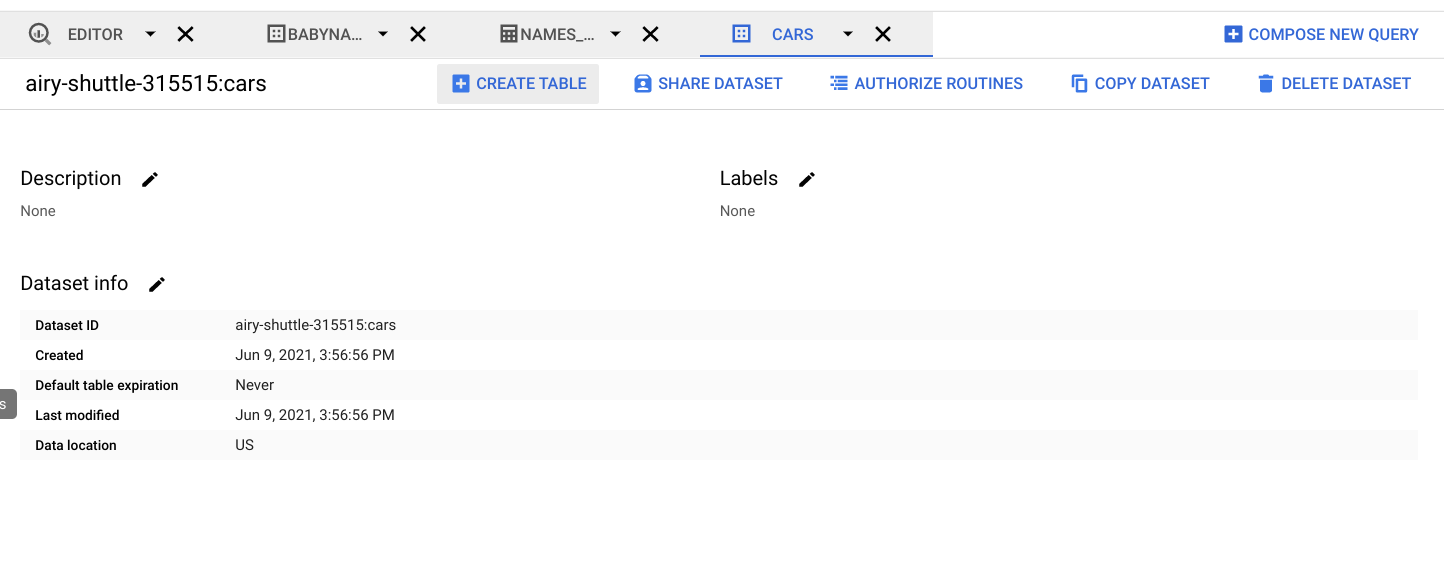
### 

### [Step 3: Create a table](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

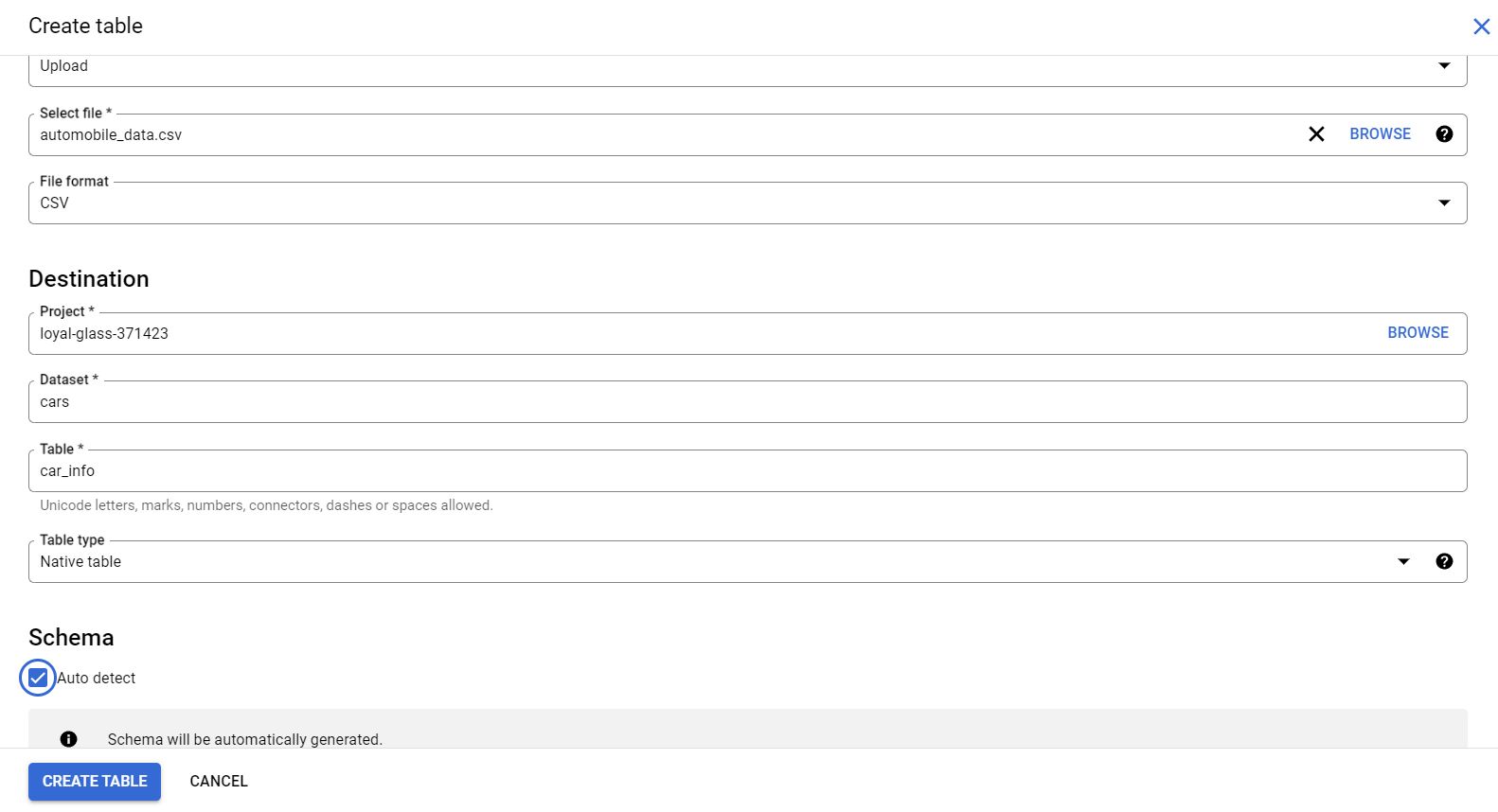
[Now that you've created a dataset. You'll create a custom table to house your data. This will enable you to use SQL queries to explore and clean data.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[After clicking on **cars** to open your newly created dataset, you will be able to add a custom table for the insertion of your downloaded data.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[From the **cars** dataset info window, click CREATE TABLE.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[Within the Create table window, upload the **automobile\_data.csv** by clicking the drop-down arrow under Source and choosing the Upload option. Click the BROWSE button and navigate to the folder where your .csv document is located, and notice the File format will automatically change to CSV. Ensure the dataset name is **cars** and name your table **car\_info**. Set the schema to Auto-detect, and finally click the Create table button.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[After creating your table, it will appear in your Explorer pane. You can click on the newly created table, **car\_info**, to explore the SCHEMA and DETAILS buttons within your data page. Once you have gotten familiar with your data, you can start querying it.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

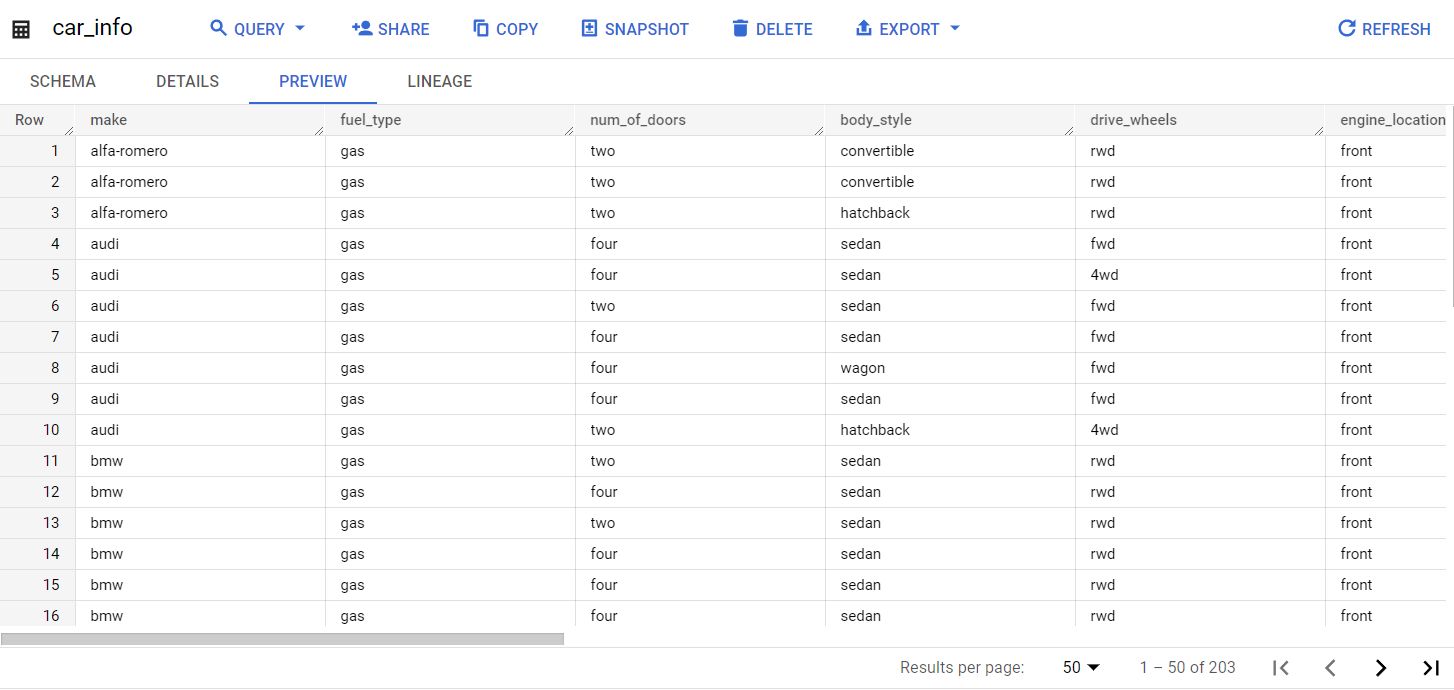
### [Step 4: Understand why you clean your data](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[Your new dataset contains historical sales data, including details such as car features and prices. You can use this data to find the top 10 most popular cars and trims. But before you can perform your analysis, you’ll need to make sure your data is clean. If you analyze dirty data, you could end up presenting the wrong list of cars to the investors. That may cause them to lose money on their car inventory investment.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

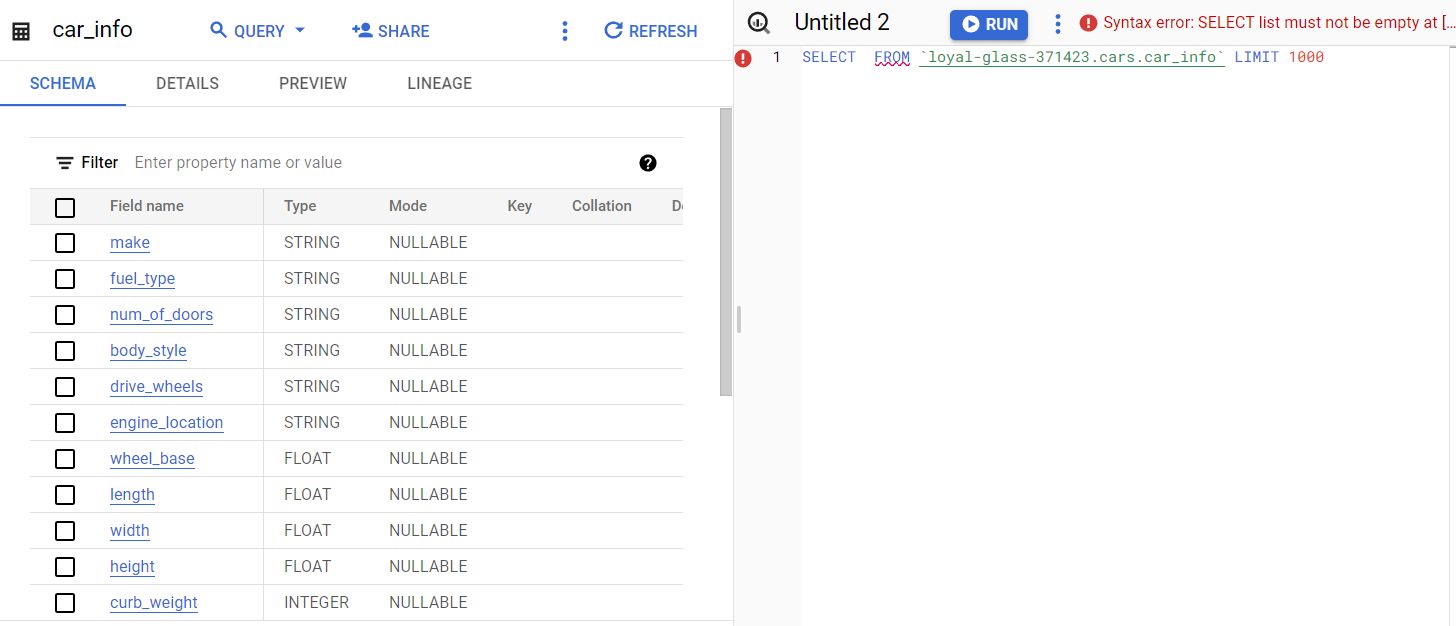
[Continue below to clean your data.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

### [Step 5: Inspect the fuel\_type column](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

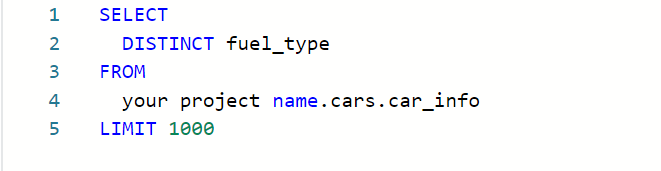
[The first thing you want to do is inspect the data in your table so you can find out if there is any specific cleaning that needs to be done. Get an initial understanding of the data table by clicking on the PREVIEW tab that sits below the car\_info toolbar.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[According to the](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql) [data’s description](https://archive.ics.uci.edu/ml/datasets/Automobile)[, the **fuel\_type** column should only have two unique string values: **diesel** and **gas**. To check and make sure that’s true, run the following query. You can generate the default query setup by clicking on the QUERY button and selecting the In split tab. This will give you a dual view of the info window and the query.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

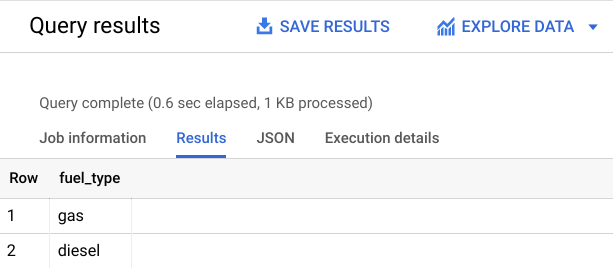
[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[Next, we can generate the first query in the workspace:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)



[NOTE: Within the **FROM** clause of the syntax above, you will need to begin the **Table ID** line with your personalized project name, period, the dataset name, period, and end with the table name. It's important to understand that the personal project name will be unique to each learner. You can also locate and copy the full **Table ID** filename by clicking on the DETAIL option tab in your **car\_info** Table info window. Once copied, paste it after the **FROM** clause and run the above query.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[This returns the following results:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

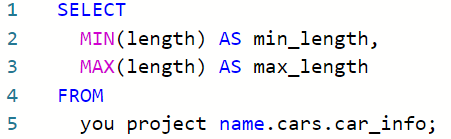
[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[This confirms that the **fuel\_type** column doesn’t have any unexpected values. Also note that the default **LIMIT 1000** is added to your query, but in this case, BigQuery is only returning two distinct fuel types.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

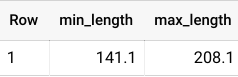
### 

### [Step 6: Inspect the length column](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[Next, you will inspect a column with numerical data. The **length** column should contain numeric measurements of the cars. So you will check that the minimum and maximum lengths in the dataset align with the](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql) [data description](https://archive.ics.uci.edu/ml/datasets/Automobile)[, which states that the lengths in this column should range from 141.1 to 208.1. Run this query to confirm:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[Your results should confirm that 141.1 and 208.1 are the minimum and maximum values respectively in this column.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

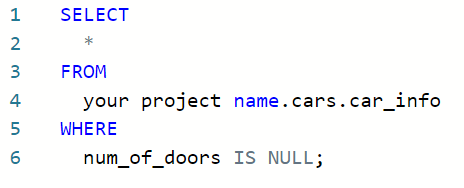
[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

### 

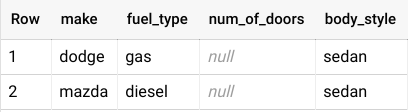
### [Step 7: Fill in missing data](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[Missing values can create errors or skew your results during analysis. You’re going to want to check your data for null or missing values. These values might appear as a blank cell or the word **null** in BigQuery.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

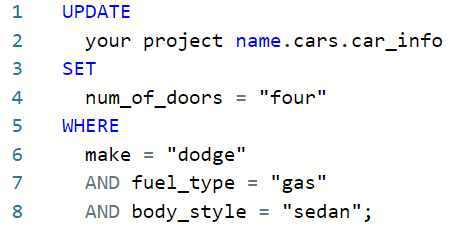
[You can check to see if the **num\_of\_doors** column contains null values using this query:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

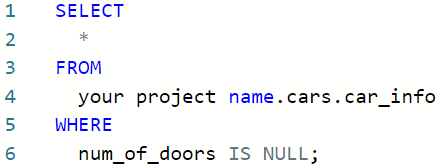
[This will select any rows with missing data for the **num\_of\_doors** column and return them in your results table. You should get two results, one Mazda and one Dodge:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[In order to fill in these missing values, you check with the sales manager, who states that all Dodge gas sedans and all Mazda diesel sedans sold had four doors. If you are using the BigQuery free trial, you can use this query to update your table so that all Dodge gas sedans have four doors:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)



[You should get a message telling you that three rows were modified in this table. To make sure, you can run the previous query again:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)



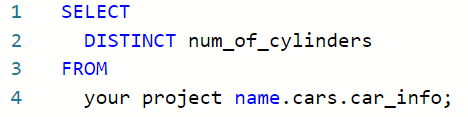
[Now, you only have one row with a null value for **num\_of\_doors**. Repeat this process to replace the null value for the Mazda.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[If you are using the BigQuery Sandbox, you can skip these **UPDATE** queries; they will not affect your ability to complete this activity.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

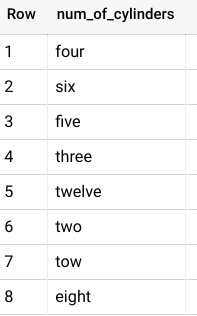
### 

### [Step 8: Identify potential errors](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

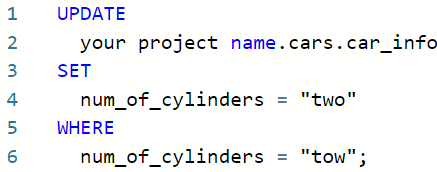
[Once you have finished ensuring that there aren’t any missing values in your data, you’ll want to check for other potential errors. You can use **SELECT DISTINCT** to check what values exist in a column. You can run this query to check the **num\_of\_cylinders** column:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

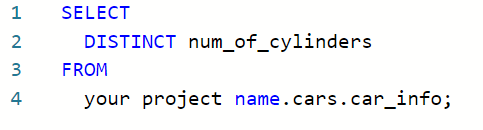
[After running this, you notice that there are one too many rows. There are two entries for two cylinders: rows 6 and 7. But the *two* in row 7 is misspelled.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

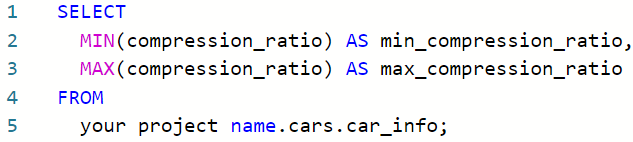
[To correct the misspelling for all rows, you can run this query if you have the BigQuery free trial:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)



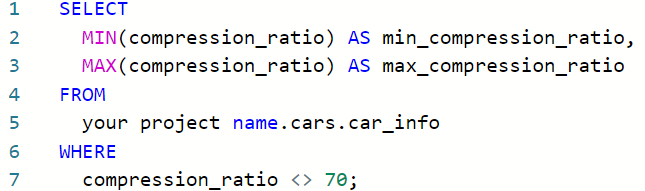
[You will get a message alerting you that one row was modified after running this statement. To check that it worked, you can run the previous query again:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)



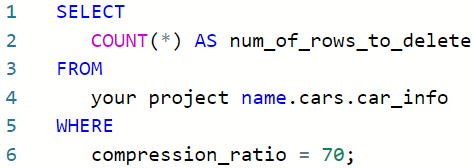
[Next, you can check the **compression\_ratio** column. According to the](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql) [data description](https://archive.ics.uci.edu/ml/datasets/Automobile)[, the **compression\_ratio** column values should range from 7 to 23. Just like when you checked the length values , you can use **MIN** and **MAX** to check if that’s correct:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)



[Notice that **this returns a maximum of 70**. But you know this is an error because the maximum value in this column should be 23, not 70. So the 70 is most likely a 7.0. Run the above query again without the row with 70 to make sure that the rest of the values fall within the expected range of 7 to 23.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[Now the highest value is 23, which aligns with the data description. So you’ll want to correct the 70 value. You check with the sales manager again, who says that this row was made in error and should be removed. Before you delete anything, you should check to see how many rows contain this erroneous value as a precaution so that you don’t end up deleting 50% of your data. If there are too many (for instance, 20% of your rows have the incorrect 70 value), then you would want to check back in with the sales manager to inquire if these should be deleted or if the 70 should be updated to another value. Use the query below to count how many rows you would be deleting:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[Turns out there is only one row with the erroneous 70 value. So you can delete that row using this query:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

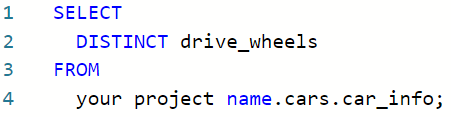


[If you are using the BigQuery sandbox, you can replace **DELETE** with **SELECT** to see which row would be deleted.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

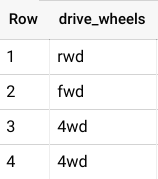
### [Step 9: Ensure consistency](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[Finally, you want to check your data for any inconsistencies that might cause errors. These inconsistencies can be tricky to spot—sometimes even something as simple as an extra space can cause a problem.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

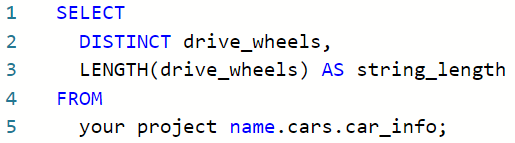
[Check the **drive\_wheels** column for inconsistencies by running a query with a **SELECT DISTINCT** statement:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

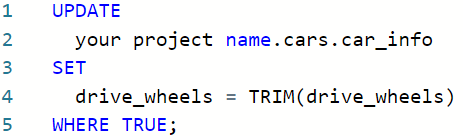
[It appears that 4wd appears twice in results. However, because you used a **SELECT DISTINCT** statement to return unique values, **this probably means there’s an extra space in one of the 4wd entries that makes it different from the other 4wd.**](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

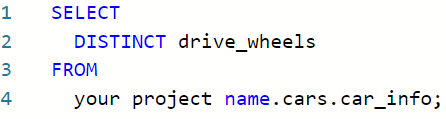
[To check if this is the case, you can use a **LENGTH** statement to determine the length of how long each of these string variables:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)



[According to these results, some instances of the 4wd string have four characters instead of the expected three (4wd has 3 characters). In that case, you can use the **TRIM** function to remove all extra spaces in the **drive\_wheels** column if you are using the BigQuery free trial:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)



[Then, you run the **SELECT DISTINCT** statement again to ensure that there are only three distinct values in the drive\_wheels column:](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)



[And now there should only be three unique values in this column! Which means your data is clean, consistent, and ready for analysis!](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

### 

### [Pro Tip: Save the activity template](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

[Be sure to save a copy of the .csv template you used to complete this activity. You can use it for further practice or to help you work through your thought processes for similar tasks in a future data analyst role.](https://www.coursera.org/learn/process-data/quiz/kU8TQ/hands-on-activity-clean-data-using-sql)

**TRANSFORMING DATA**

[**OPTIONAL: UPLOAD THE STORE TRANSACTIONS DATASET TO BIGQUERY**](https://www.coursera.org/learn/process-data/supplement/kFRhX/optional-upload-the-store-transactions-dataset-to-bigquery)

In the next video, the instructor uses a specific dataset. The instructions in this reading are provided for you to upload the same dataset in your BigQuery console so you can follow along.

You must have a BigQuery account to follow along.

## 

## **Prepare for the next video**

* First, download the .csv file from the attachment below.

[Lauren's Furniture Store Transaction Table](https://d3c33hcgiwev3.cloudfront.net/0cvJS5ocTSu-77CVZvn6kw_55078d160a924e49aca0837f03d995f1_Lauren-s-Furniture-Store-Transaction-Table.csv?Expires=1712361600&Signature=hjA69umKQm~n8WgrzJvgi7G4ldtHMvhToQx2ImCa2JrjBP5AmFTgQGVqf2-4xtoHu03bejgBBaLR~eSI-bgLeIZ7on2zlY1GUp-3Bc4u-1IK0zLHAt2V-e8PC~qmO56Y7z8RJDDa6p1JX0rjTxT6jOPKKY258u82E-WvdjQuA4E_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

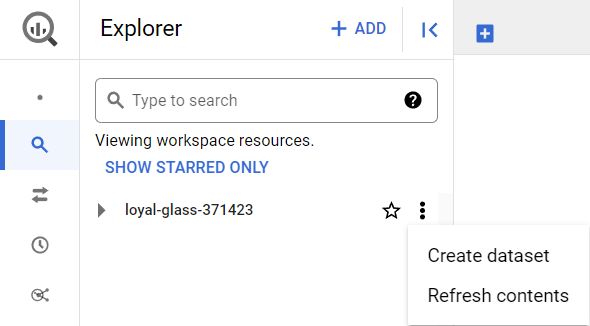
[CSV File](https://d3c33hcgiwev3.cloudfront.net/0cvJS5ocTSu-77CVZvn6kw_55078d160a924e49aca0837f03d995f1_Lauren-s-Furniture-Store-Transaction-Table.csv?Expires=1712361600&Signature=hjA69umKQm~n8WgrzJvgi7G4ldtHMvhToQx2ImCa2JrjBP5AmFTgQGVqf2-4xtoHu03bejgBBaLR~eSI-bgLeIZ7on2zlY1GUp-3Bc4u-1IK0zLHAt2V-e8PC~qmO56Y7z8RJDDa6p1JX0rjTxT6jOPKKY258u82E-WvdjQuA4E_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)

* Next, complete the steps below in your BigQuery console to upload the Store Transaction dataset.

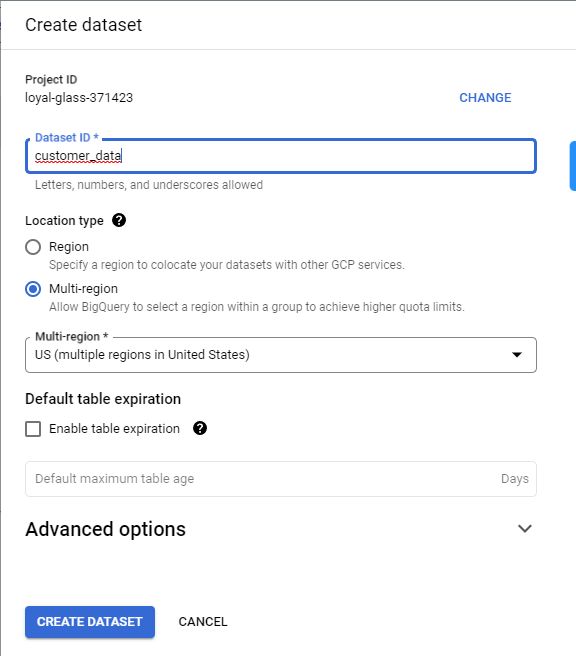
**Note:** These steps will be different from what you performed before. In previous instances, you selected the **Auto detect** check box to allow BigQuery to auto-detect the schema. This time, you will choose to create the schema by editing it as text. This method can be used when BigQuery doesn't automatically set the desired type for a particular field. In this case, you will specify **STRING** instead of **FLOAT** as the type for the purchase\_price field.

**Step 1**: Open your BigQuery console and click on the project you want to upload the data to. If you already created a **customer\_data** dataset for your project, jump to step 5; otherwise, continue with step 2.

**Step 2:** In the **Explorer** on the left, click the **Actions** icon (three vertical dots) next to your project name and select **Create dataset**.

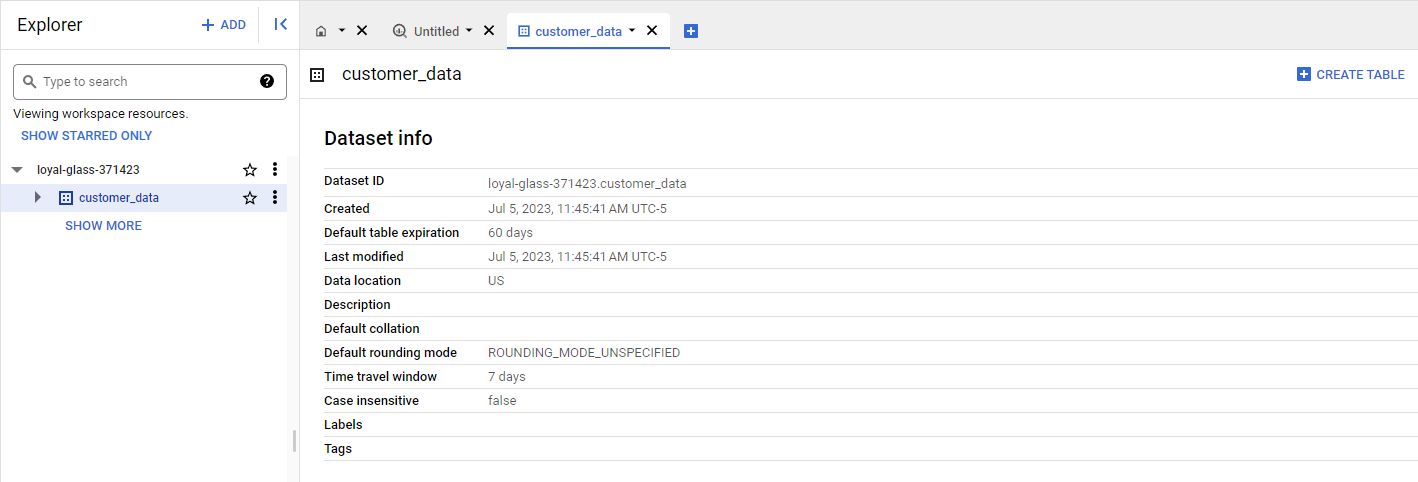


**Step 3:** In the **Create dataset** window, enter **customer\_data** for the Dataset ID. Make sure the *Location type* is set to **Multi-region, US (Multiple regions in United States),** and all the default *Advanced options* remain set to the **Google-managed encryption key** option.



**Step 4:** Click **CREATE DATASET** (blue button) to add the dataset to your project.

**Step 5:** In the **Explorer** pane, click on the expansion arrow under your project name, and then click the **customer\_data** dataset.



**Step 6:** On the far right hand side of the new **Dataset info** page, click the blue **+ CREATE TABLE** button to open the **Create table** window. Use the visual settings in the next image to complete the steps below.



**Step 7:** Under **Source**, for the **Create table from** selection, choose where the data will be coming from.

* Select **Upload**.
* Click **Browse** to select the Store Transaction Table .csv file you downloaded.
* Choose **CSV** from the file format drop-down.

**Step 8:** For Table name, enter **customer\_purchase** if you plan to follow along with the video.

**Step 9:** For Schema, click the toggle switch for **Edit as text**. This opens up a box for the text.

**Step 10:** Copy and paste the following text into the box. Be sure to include the opening and closing brackets. They are required.

[

{

"description": "date",

"mode": "NULLABLE",

"name": "date",

"type": "DATETIME"

},

{

"description": "transaction id",

"mode": "NULLABLE",

"name": "transaction\_id",

"type": "INTEGER"

},

{

"description": "customer id",

"mode": "NULLABLE",

"name": "customer\_id",

"type": "INTEGER"

},

{

"description": "product name",

"mode": "NULLABLE",

"name": "product",

"type": "STRING"

},

{

"description": "product\_code",

"mode": "NULLABLE",

"name": "product\_code",

"type": "STRING"

},

{

"description": "product color",

"mode": "NULLABLE",

"name": "product\_color",

"type": "STRING"

},

{

"description": "product price",

"mode": "NULLABLE",

"name": "product\_price",

"type": "FLOAT"

},

{

"description": "quantity purchased",

"mode": "NULLABLE",

"name": "purchase\_size",

"type": "INTEGER"

},

{

"description": "purchase price",

"mode": "NULLABLE",

"name": "purchase\_price",

"type": "STRING"

},

{

"description": "revenue",

"mode": "NULLABLE",

"name": "revenue",

"type": "FLOAT"

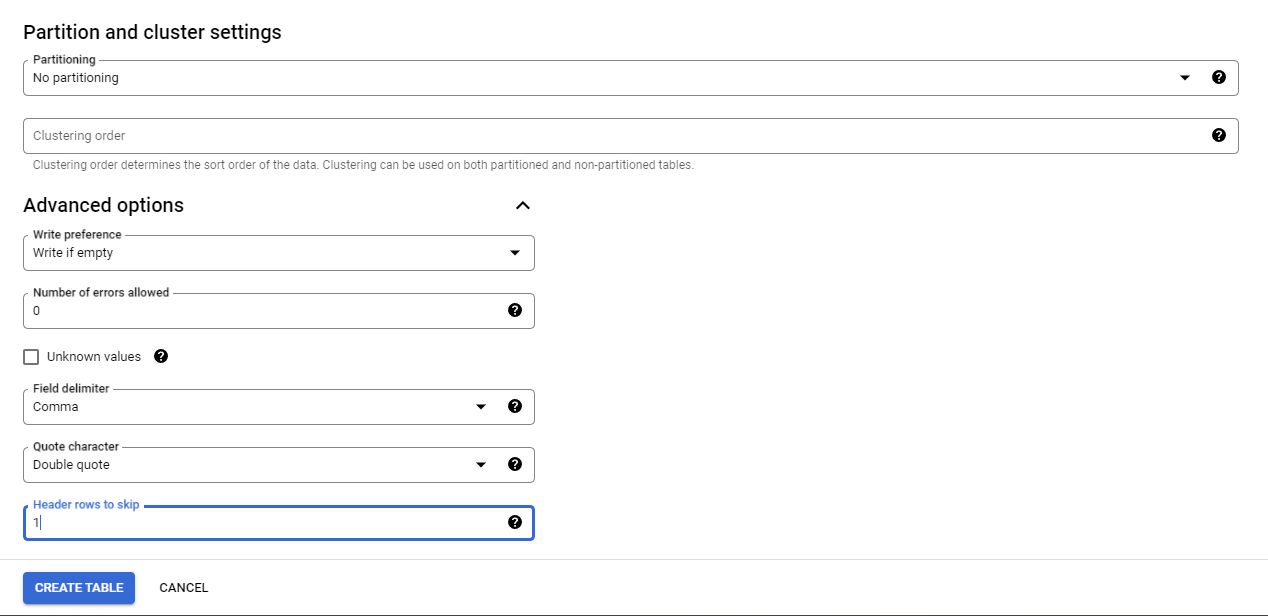
}

]

**Step 11:** Scroll down and expand the **Advanced options** section.

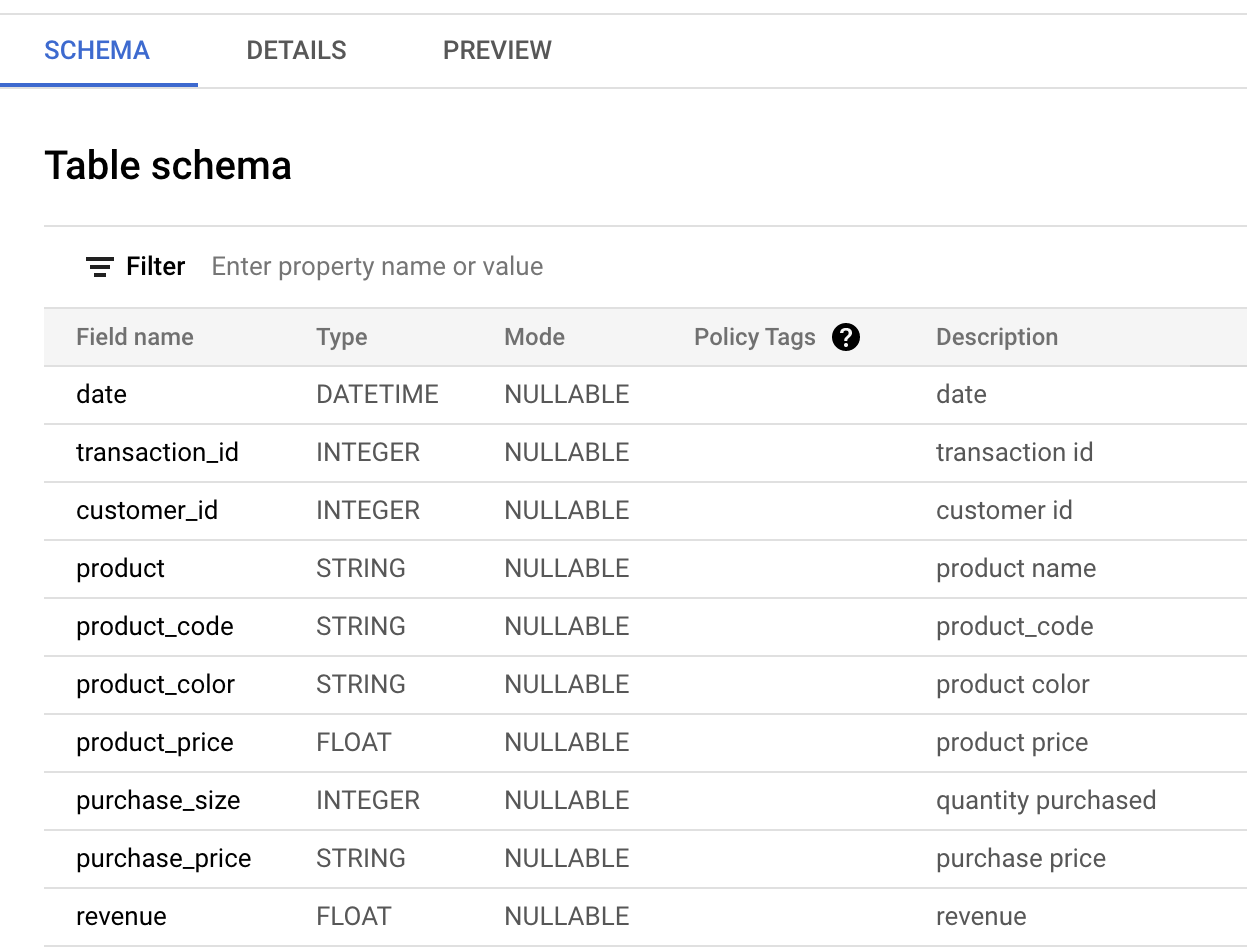
**Step 12:** For the **Header rows to skip** field, enter **1**.

**NOTE**: It is very important that you don't skip the last step, or you will receive 'parsing' errors, as BigQuery will try to apply the schema editing functions to the title row.

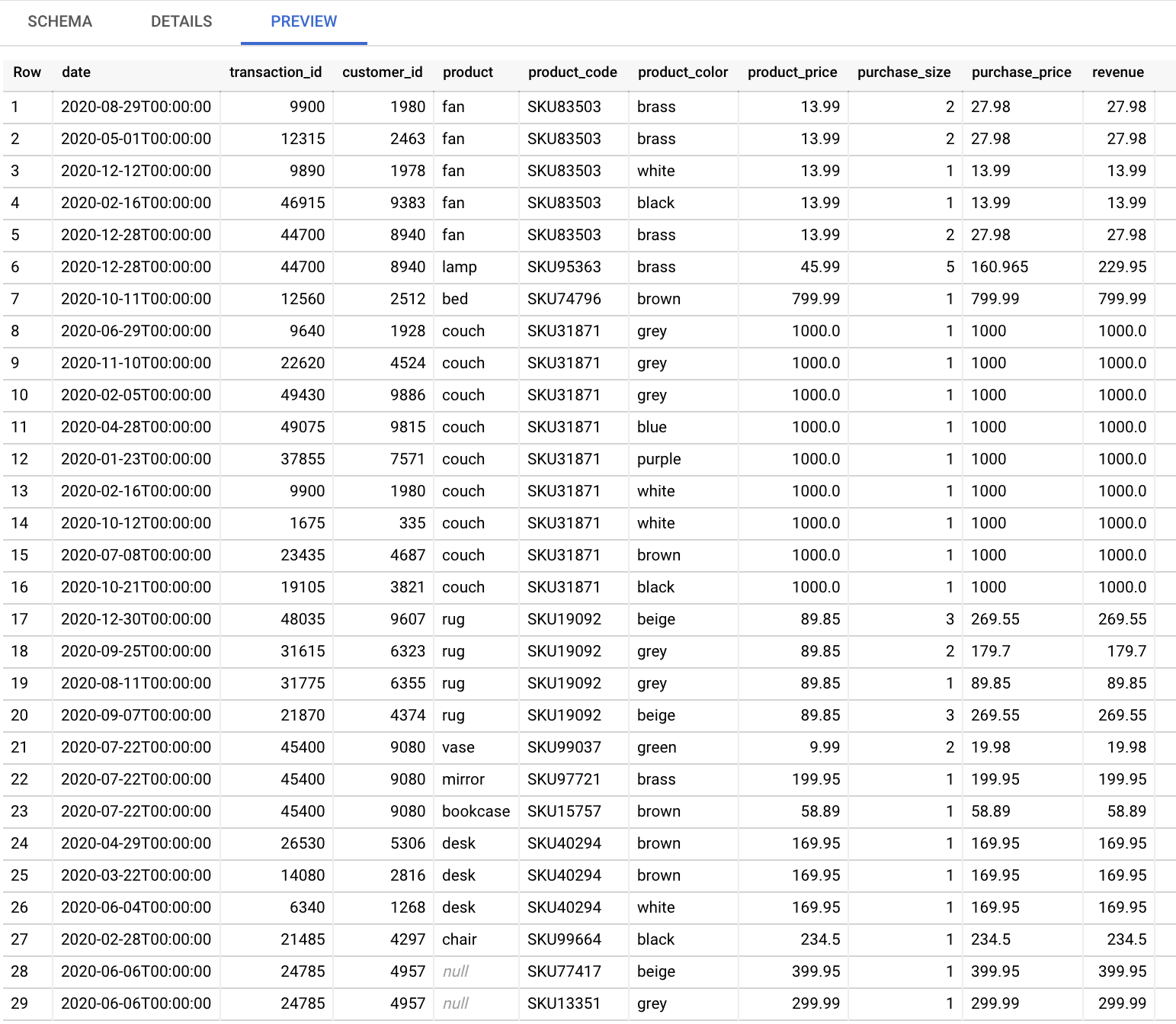


**Step 13:** Click **Create** **table** (blue button). You will now see the **customer\_purchase** table under your **customer\_data** dataset in your **Explorer** pane.

**Step 14:** Click the **customer\_purchase** table and in the **Schema** tab, confirm that the schema matches the schema shown below.



**Step 15:** Click the **Preview** tab and confirm that your data matches the data shown below.



[**ADVANCED DATA-CLEANING FUNCTIONS, PART 1**](https://www.coursera.org/learn/process-data/lecture/OlLEQ/advanced-data-cleaning-functions-part-1)

Hi there and welcome back. So far we've gone over some basic SQL queries and functions that can help you clean your data.

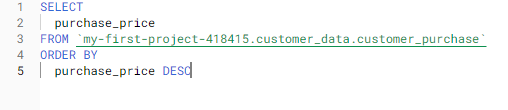
We've also checked out some ways you can deal with string variables in SQL to make your job easier. Get ready to learn more functions for dealing with strings in SQL. Trust me, these functions will be really helpful in your work as a data analyst.

In this video, we'll check out strings again and learn how to use the cast function to correctly format data. When you import data that doesn't already exist in your SQL tables, the data types from the new dataset might not have been imported correctly. This is where the CAST function comes in handy.

Basically, CAST can be used to convert anything from one data type to another. Let's check out an example. Imagine we're working with Lauren's Furniture Store. The owner has been collecting transaction data for the past year, but she just discovered that they can't actually organize their data because it hadn't been formatted correctly.

So we'll help her by converting her data to make it useful again. For example, let's say we want to sort all purchases by purchase\_price in descending order. That means we want the most expensive purchase to show up first in our results. To write the SQL query, we start with the basic SQL structure. SELECT, FROM, WHERE, we know the data is stored in the customer\_purchase table in the customer\_dataset.

So we write customer\_data.customer\_purchase after FROM. Next, we tell SQL what data to give us in the select clause. We want to see the purchase\_price data, so we type purchase\_price after SELECT. Next is the where clause. We are not filtering out any data since we want all purchase prices shown, so we can take out the where clause. Finally, to sort the purchase\_price in descending order, we type ORDER BY purchase\_price DESC at the end of our query. Let's run this query.



We see that 89.85 shows up at the top with 799.99 below it, but we know that 799.99 is a bigger number than 89.85. The database doesn't recognize that these are numbers, so it didn't sort them that way. If we go back to the customer\_purchase table and take a look at its schema, we can see what data type the database thinks purchase\_price is. It says here the database thinks purchase\_price is a string, when in fact it is a float, which is a number that contains a decimal. That is why 89.85 shows up before 799.99.

When we sort letters, we start from the first letter before moving on to the second letter.

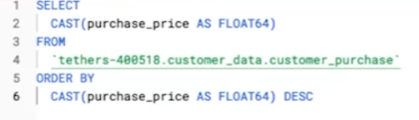
So if we want to sort the words apple and orange in descending order, we start with the first letters a and o. Since o comes after a, orange will show up first, then apple.

The database did the same with 89.85 and 799.99. It started with the first letter, which in this case was 8 and 7 respectively. Since 8 is bigger than 7, the database sorted 89.85 first and then 799.99 because the database treated these as text strings.

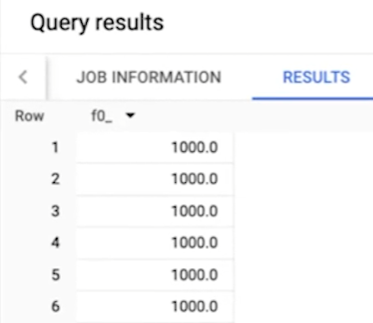
The database doesn't recognize these strings as floats because they haven't been typecast to match that data type yet. **Typecasting means converting data from one type to another**, which is what we'll do with the CAST function. We use the CAST function to replace purchase\_price with a new purchase\_price that the database recognizes as float instead of string. We start by replacing purchase\_price with CAST. Then we tell SQL the field we want to change, which is the purchase\_price field. Next is a data type we want to change purchase\_price to, which is the FLOAT data type. BigQuery stores numbers in a 64 bit system, so the FLOAT data type is referenced as float 64 in our query.

This might be slightly different in other SQL platforms, but basically the 64 and float 64 just indicates that we're casting numbers in the 64 bit system as FLOATs. We also need to sort this new field so we change purchase\_price after ORDER BY to CAST purchase\_price as FLOAT64.

This is how we use the cast function to allow SQL to recognize the purchase\_price column as FLOATs instead of text strings. Now we can sort our purchases by purchase\_price.

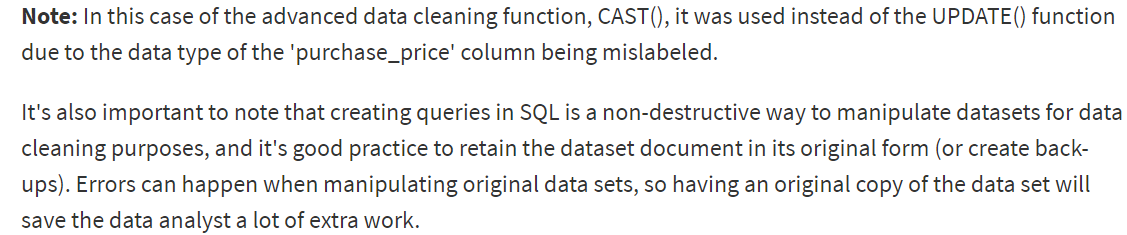


And just like that, Lauren's Furniture Store has data that can actually be used for analysis.



As a data analyst, you'll be asked to locate and organize data a lot, which is why you want to make sure you convert between data types early on.

Businesses like our Furniture Store are interested in timely sales data, and you need to be able to account for that in your analysis. **The CAST function can be used to change strings into other data types too, like date and time**. As a data analyst, you might find yourself using data from various sources. Part of your job is making sure the data from those sources is recognizable and usable in your database so that you won't run into any issues with your analysis.



And now you know how to do that. The CAST function is one great tool you can use when you're cleaning data. And coming up, we'll cover some other advanced functions that you can add to your toolbox. See you soon.

[**ADVANCED DATA-CLEANING FUNCTIONS, PART 2**](https://www.coursera.org/learn/process-data/lecture/BIh2D/advanced-data-cleaning-functions-part-2)

Hey there. Great to see you again. So far we've seen some SQL functions in action. In this video, we'll go over more uses for CAST and then learn about CONCAT and COALESCE.



Earlier we talked about the **CAST function**, which lets us **typecast text strings into floats**. I called out that the CAST function can be used to change into other data types too. Let's check out another example of how you can use CAST in your own data work.

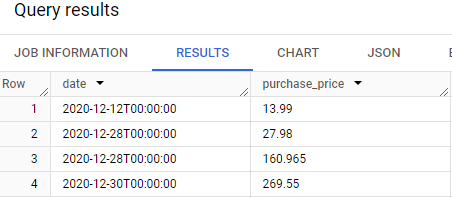
We've got the transaction data we were working with from our Lauren's furniture store example, but now we'll check out the purchase date field. The furniture store owner has asked us to look at purchases that occurred during their sales promotion period in December.

Let's write a SQL query that will pull date and purchase\_price for all purchases that occurred between December 1st, 2020 and December 31st, 2020. **We start by writing the basic SQL structure; SELECT, FROM, WHERE**. We know the data comes from the customer\_purchase table in the customer\_data data set, so we write customer\_data.customer\_purchase after FROM. Next we tell SQL what data to pull. Since we want date and purchase\_price, we add them into the SELECT statement.

Finally, we want SQL to filter for purchases that occurred in December only, so we type date BETWEEN 2020-12-01 and 2020-12-31 in the WHERE clause.

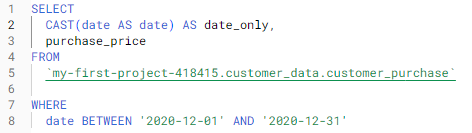


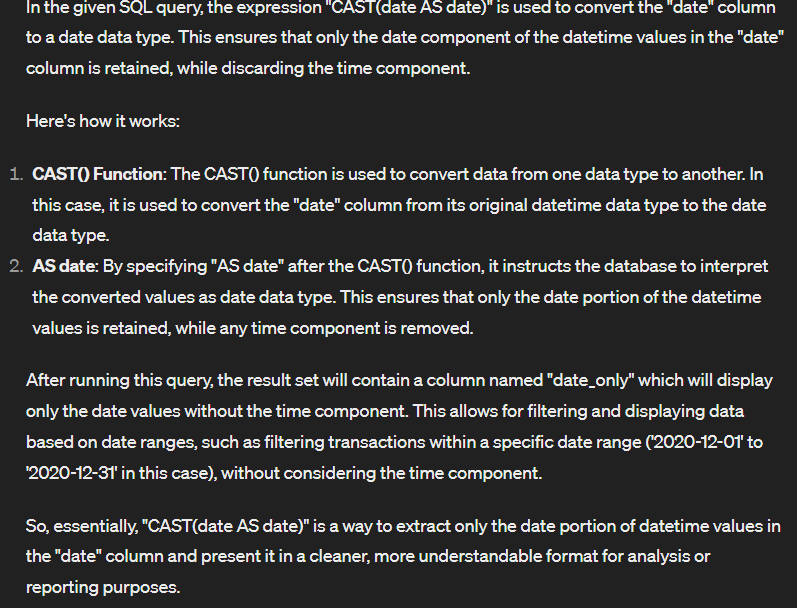
Let's run the query. Four purchases occurred in December, but the **date field looks odd**.



That's because the database recognizes the date field as date time, which consists of the date and time. Our SQL query still works correctly even if the date field is date time instead of date. But we can tell SQL to convert the date field into the date data type so we see just the date and not the time.

To do that, we use the CAST function again. We'll use the CAST function to replace the date field in our select statement with the new date field that will show the date and not the time. We can do that **by typing CAST and adding the date as the field we want to change, then we tell SQL the data type we want instead, which is the date data type.**





Now we can have cleaner results for purchases that occurred during the December sales period.

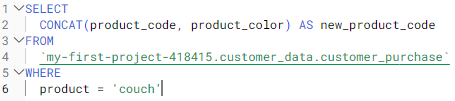
**CAST is a super useful function for cleaning and sorting data**, which is why I wanted you to see it in action one more time.

Next up, let's check out the CONCAT function. **CONCAT lets you add strings together to create new text strings that can be used as unique keys**.

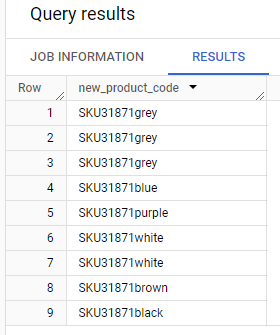
Going back to our customer\_purchase table, we see that the furniture store sells different colors of the same product. The owner wants to know if customers prefer certain colors so the owner can manage store inventory accordingly. The problem is the product\_code is the same regardless of the product color. We need to find another way to separate products by color so we can tell if customers prefer one color over the others.

We'll use CONCAT to produce a unique key that'll help us tell the products apart by color and count them more easily. Let's write our SQL query by starting with the basic structure. SELECT, FROM, WHERE. We know our data comes from the customer\_purchase table and the customer\_data data set, so we type customer\_data.customer\_purchase after FROM. Next we tell SQL what data to pull. We use the CONCAT function here to get that unique key of product and color. We type CONCAT, the first column we want, product\_code, and the other column we want, product\_color.

Finally, let's say we want to look at couches, so we filter for couches by typing product='couch' in the WHERE clause. Now we can count how many times each couch was purchased and figure out if customers preferred one color over the others.



With CONCAT, the furniture store can find out which color couches are the most popular and order more.



I've got one last advanced function to show you, COALESCE. **COALESCE can be used to return non-null values in a list**. Null values are missing values.

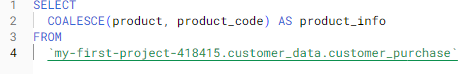
If you have a field that's optional in your table, it'll have null in that field for rows that don't have appropriate values to put there.

Let's open the customer\_purchase table so I can show you what I mean. In the customer\_purchase table, we can see a couple of rows where product information is missing. That is why we see nulls there. But for the rows where product name is null, we see that there is product\_code data that we can use instead. We'd prefer SQL to show us the product name, like bed or couch because it's easier for us to read. But if the product name doesn't exist, we can tell SQL to give us the product\_code instead. That is where the COALESCE function comes into play.

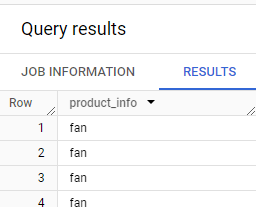
Let's say we wanted a list of all products that were sold. We want to use the product name column to understand what product was sold.

So we write our SQL query with the basic SQL structure, SELECT, FROM, WHERE. We know our data comes from customer\_purchase table and the customer\_data data set, so we type customer\_data.customer\_purchase after FROM. Next, we tell SQL the data we want. We want a list of product names. But if names aren't available, then give us the product code. Here is where we type COALESCE, then we tell which column to check first, product, and which column to check second, if the first column is null, product\_code. We'll name this new field as product\_info.

Finally, we are not filtering out any data so we can take out the WHERE clause.



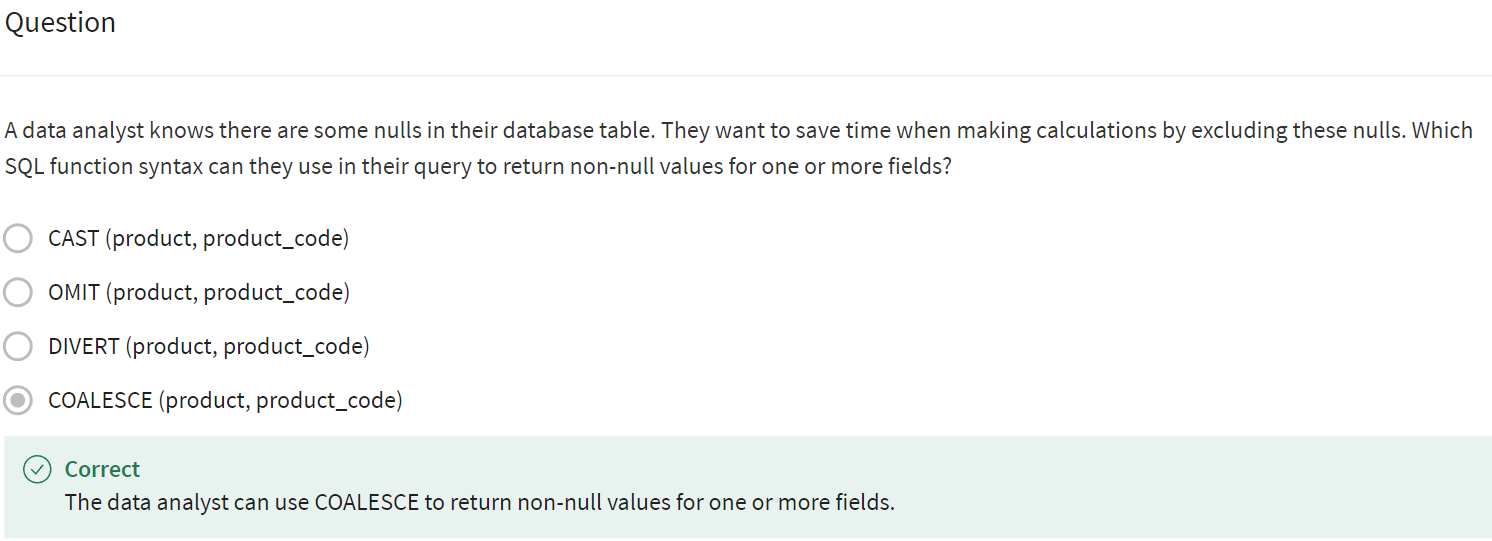
In this case if there’s a null value on the 1st column then it will be replaced with a value from the 2nd column stated in the query. This gives us product information for each purchase.



Now we have a list of all products that were sold for the owner to review.

**COALESCE can save you time when you're making calculations too, by skipping any null values and keeping your math correct**.

Those were just some of the advanced functions you can use to clean your data and get it ready for the next step in the analysis process. You'll discover more as you continue working in SQL. But that's the end of this video and this module. Great work.We've covered a lot of ground.



**You learned the different data cleaning functions and spreadsheets and SQL, and the benefits of using SQL to deal with large data sets**.

We also added some SQL formulas and functions to your toolkit, and most importantly, we got to experience some of the ways that SQL can help you get data ready for your analysis.

After this, you'll get to spend some time learning how to verify and report your cleaning results so that your data is squeaky clean and your stakeholders know it.

**Some of these concepts might seem challenging at first, but they'll become second nature to you as you progress in your career**. **It just takes time and practice**.

Speaking of practice, feel free to go back to any of these videos and re-watch or even try some of these commands on your own. Good luck and I'll see you again when you're ready.

[**DEBUG SQL CODE**](https://www.coursera.org/learn/process-data/discussionPrompt/P2zJL/debug-sql-code)

**[DISCUSSION PROMPT](https://www.coursera.org/learn/process-data/discussionPrompt/P2zJL/debug-sql-code)**

[**DATA-CLEANING WITH SQL FUNCTIONS**](https://www.coursera.org/learn/process-data/ungradedWidget/IEEEF/data-cleaning-with-sql-functions)

[**SELF-REFLECTION: CHALLENGES WITH SQL**](https://www.coursera.org/learn/process-data/quiz/wnoX9/self-reflection-challenges-with-sql)

****

## **Activity Overview**

****

Now that you have practiced writing SQL functions, you can pause for a moment and think about what you are learning. In this self-reflection, you will consider your thoughts about your experience with learning SQL and respond to brief questions.

This self-reflection will help you develop insights into your own learning and prepare you to identify your successes and difficulties with learning SQL so you can understand how to develop your skills further. As you answer questions—and come up with questions of your own—you will consider concepts, practices, and principles to help refine your understanding and reinforce your learning. You’ve done the hard work, so make sure to get the most out of it: This reflection will help your knowledge stick!

### 

### **Your SQL experience so far**

So far, you have been introduced to many different tools available in SQL. As a brief review, you learned how to complete tasks such as:

* Getting data from a table using **SELECT** statements.
* De-duplicating data using commands like **DISTINCT** and **COUNT** + **WHERE**.
* Manipulating string data with **TRIM()** and **SUBSTR**.
* Creating/dropping tables with **CREATE TABLE** and **DROP TABLE**.
* Changing data types with **CAST**.

Some of these tasks are more challenging than others, and learning all the various SQL functions takes work. But, when you practice different functions, you can master the skills needed to make SQL work the way you need it to. Take a moment to think about the parts of SQL that you’ve found most challenging.

**Question1 - Reflection**Consider everything you’ve learned about and practiced in SQL so far:

**Are there any areas of data processing with SQL that you’ve found particularly challenging?**

**Are there any data processing skills that you’d like to improve upon? If so, what are they?**

Now, write 2-3 sentences (40-60 words) in response to each of these questions. Type your response in the text box below.

**COURSE 4 MODULE 3 CHALLENGE**

## 

## **TERMS AND DEFINITIONS FOR COURSE 4, MODULE 3**

**CAST:** A SQL function that converts data from one datatype to another

**COALESCE:** A SQL function that returns non-null values in a list

**CONCAT:** A SQL function that adds strings together to create new text strings that can be used as unique keys

**DISTINCT:** A keyword that is added to a SQL SELECT statement to retrieve only non-duplicate entries

**Float:** A number that contains a decimal

**Substring:** A subset of a text string

**Typecasting:** Converting data from one type to another