1. 1 point

Activity overview

By now, you have worked with data using both spreadsheets and SQL. These tools operate very differently: In spreadsheets, you are able to observe and interact with data directly; with SQL, you interact with data through queries to the database. In this activity, you will use spreadsheets to clean your data before importing it into SQL for analysis.

In this scenario, you have been working for a national store chain as a data analyst. Management is interested in the amount of inventory being kept in storage at regional sites. Your supervisor has asked you to perform an analysis on inventory and sales data to make recommendations for changes to inventory management practices. You have been provided with three datasets containing information about inventory, products, and sales.

By the time you complete this activity, you will be able to combine tools to successfully analyze data. Switching between spreadsheets and SQL can be challenging because they're so different, but once you're more used to both tools, you'll be able to use both more easily. This is important for tackling larger and more complex projects in your career as a data analyst.

To get started, first download the three store data CSV files: inventory, products, and sales.

Click the link to each CSV file to create a copy. If you don't have a Google account, you may download the data directly from the attachments below.

Link to data: inventory, sales, and products.

OR

Download data:

Inventory CSV File

Sales CSV File

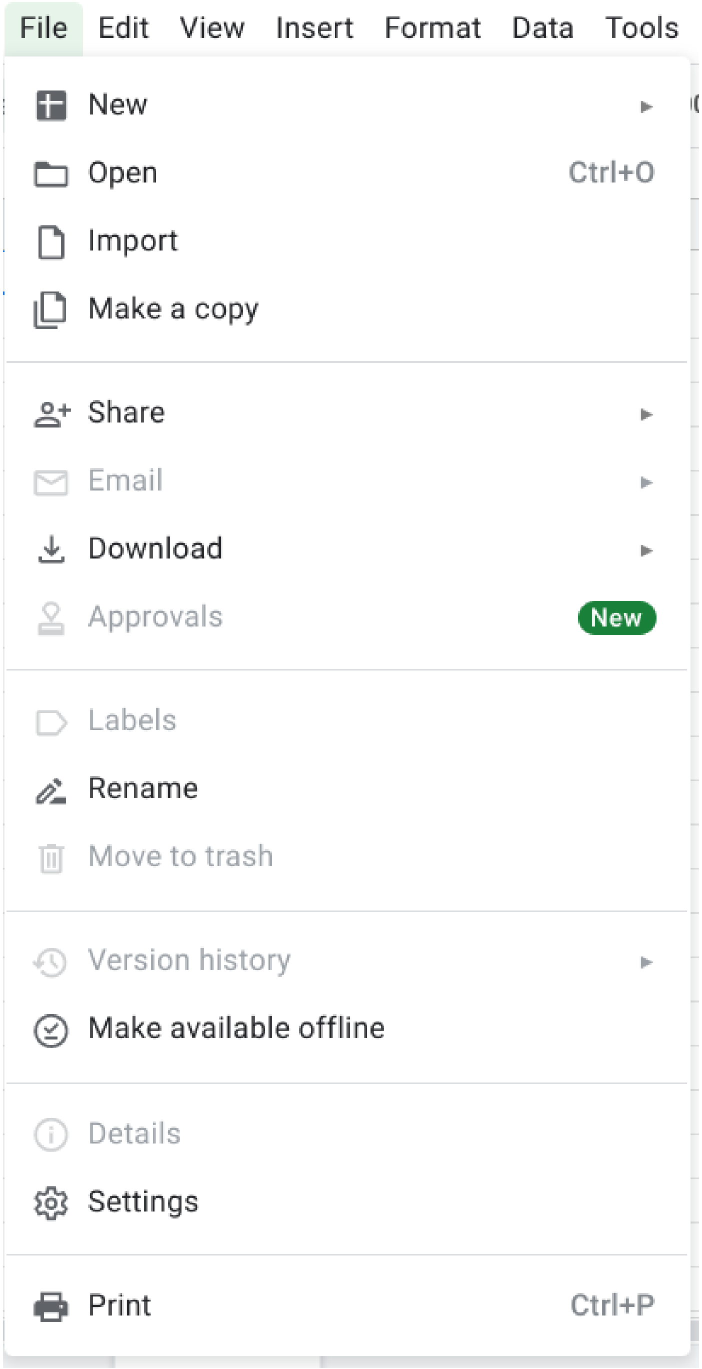
Products CSV File

Cleaning the data

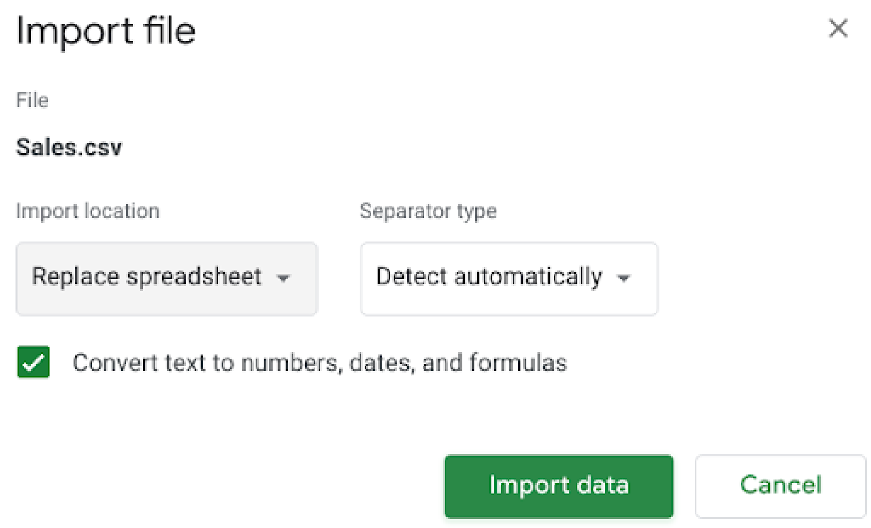
Before you upload these files to SQL, you can import them into a spreadsheet in Sheets to get comfortable with the data before you start analyzing it in BigQuery. This might not always be possible with larger datasets you encounter in the future, but you should explore as much as possible within this exercise! You can also use this step to perform some data-cleaning tasks.

Step 1: Import the data

If you're using Google Sheets, you'll first need to import the data files into your spreadsheet . Open Sheets and navigate to the File menu, then select Import from the dropdown list.



Select the first file and upload it to the spreadsheet. Choose Replace spreadsheet to insert it into the current sheet.



Import file

File

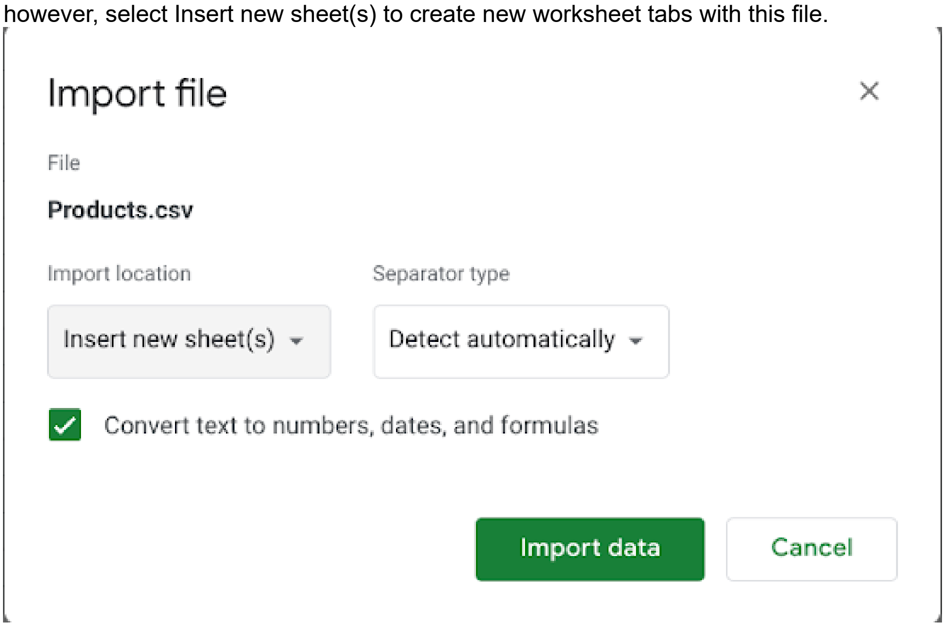
Sales.csv

Import location: Replace spreadsheet

Separator type: Detect automatically

☑ Convert text to numbers, dates, and formulas

Then return to the Import menu under the File menu and upload the next file. This time, however, select Insert new sheet(s) to create new worksheet tabs with this file.



Import file

File

Products.csv

Import location: Insert new sheet(s)

Separator type: Detect automatically

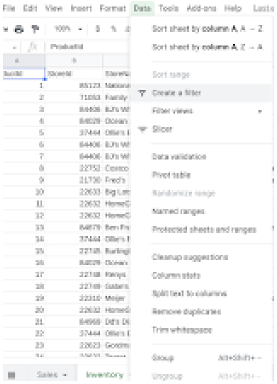
☑ Convert text to numbers, dates, and formulas

Repeat these steps until you have all three files added to your spreadsheet.

Step 2: Inspect the data

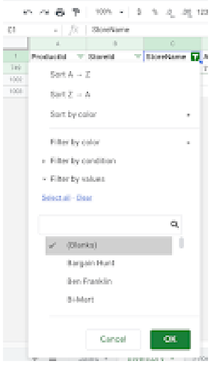
Applying filters in spreadsheets is a good way to identify any data that needs to be cleaned. You'll inspect the Inventory sheet now.

Navigate to the Inventory sheet and click any cell in the spreadsheet. Open the Data dropdown menu and select Create a filter.

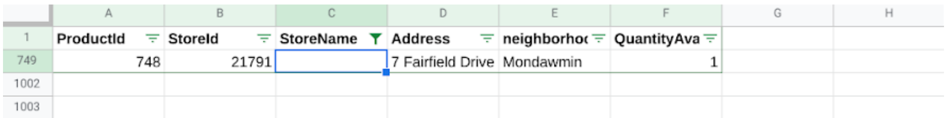


Now you can click the filter icons for each column to inspect the values. Start with the StoreID column. As you scroll through, you'll notice that there do not appear to be any blanks or incorrectly entered values. However, if you inspect the StoreName column, you'll find a blank.

Deselect all of the values except for the blank.

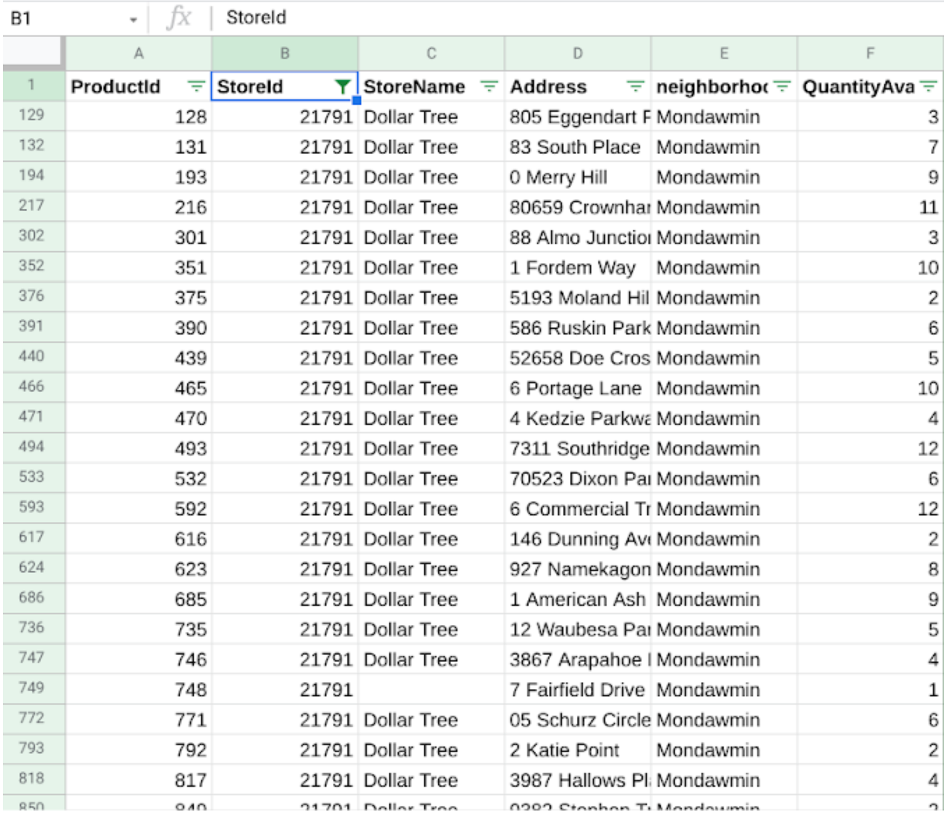


This should return one row with a missing entry under the StoreName column.



	A	B	C	D	E	F	G	H
1	ProductId	StoreId	StoreName	Address	neighborhoo	QuantityAva		
749	748	21791		7 Fairfield Drive	Mondawmin	1		
1002								
1003								

You might be able to find what the missing value is and input it correctly using the filter. Clear the Storename filter and use the StoreId column filter for other stores with the ID 21791.



B1 ▾ | fx | StoreId

	A	B	C	D	E	F
1	ProductId	StoreId	StoreName	Address	neighborhoo	QuantityAva
129	128	21791	Dollar Tree	805 Eggendart F	Mondawmin	3
132	131	21791	Dollar Tree	83 South Place	Mondawmin	7
194	193	21791	Dollar Tree	0 Merry Hill	Mondawmin	9
217	216	21791	Dollar Tree	80659 Crownhai	Mondawmin	11
302	301	21791	Dollar Tree	88 Almo Junctio	Mondawmin	3
352	351	21791	Dollar Tree	1 Fordem Way	Mondawmin	10
376	375	21791	Dollar Tree	5193 Moland Hil	Mondawmin	2
391	390	21791	Dollar Tree	586 Ruskin Park	Mondawmin	6
440	439	21791	Dollar Tree	52658 Doe Cros	Mondawmin	5
466	465	21791	Dollar Tree	6 Portage Lane	Mondawmin	10
471	470	21791	Dollar Tree	4 Kedzie Parkwa	Mondawmin	4
494	493	21791	Dollar Tree	7311 Southridge	Mondawmin	12
533	532	21791	Dollar Tree	70523 Dixon Pai	Mondawmin	6
593	592	21791	Dollar Tree	6 Commercial Tr	Mondawmin	12
617	616	21791	Dollar Tree	146 Dunning Av	Mondawmin	2
624	623	21791	Dollar Tree	927 Namekagon	Mondawmin	8
686	685	21791	Dollar Tree	1 American Ash	Mondawmin	9
736	735	21791	Dollar Tree	12 Waubesa Pai	Mondawmin	5
747	746	21791	Dollar Tree	3867 Arapahoe I	Mondawmin	4
749	748	21791		7 Fairfield Drive	Mondawmin	1
772	771	21791	Dollar Tree	05 Schurz Circle	Mondawmin	6
793	792	21791	Dollar Tree	2 Katie Point	Mondawmin	2
818	817	21791	Dollar Tree	3987 Hallows Pl	Mondawmin	4
850	849	21791	Dollar Tree	9283 Stephen Ti	Mondawmin	2

It appears that the other stores with this ID are all Dollar Tree, so it's probably safe to input that as the StoreName value in the blank cell.

Inspect the other columns in this sheet, then return to the Data menu to turn off the filters. Next, navigate to the Products sheet.

Similarly to the last sheet, you can repeat this process to inspect the Products data. Go to the Data menu and select Create filter.

Check the ProductID column. You'll find that there is a NA value in this column, despite the fact that this column should only have numeric values. In this case, you've checked in with the dataset owner, who said you can delete this row because it was input by mistake and does not belong in this dataset. Turn off the filter and move on to the next step.

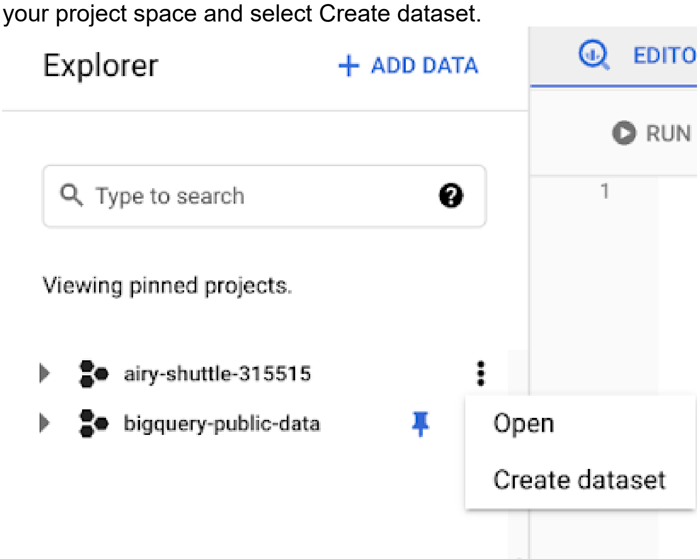
From spreadsheets to BigQuery

Now that you have checked out your data in a tool that lets you observe and interact with your data directly, it's time to transition to using SQL. With SQL, you can only observe the results of your query, which requires a different mindset than spreadsheets — but SQL is very powerful when you're working with databases and larger datasets!

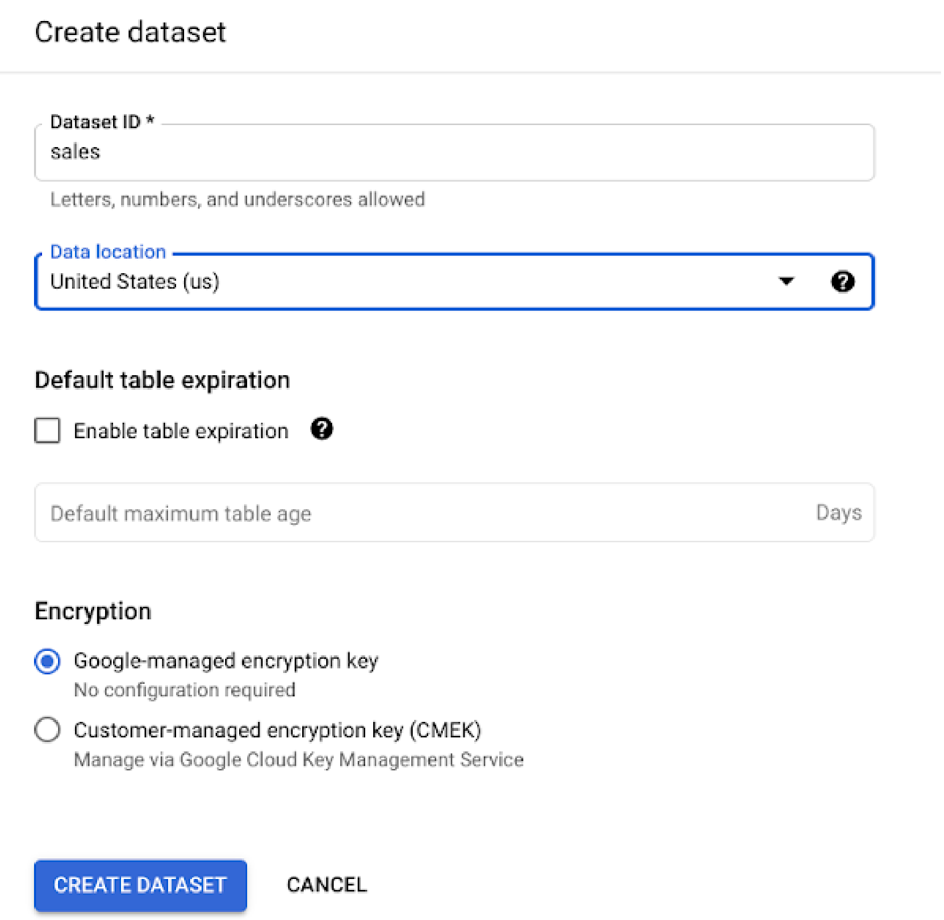
Step 1: Create a dataset and custom table

Similar to previous activities, you will need to create a dataset and custom table to house this data before you can inspect it in BigQuery.

1. From the Explorer pane in your BigQuery console, click the three vertical dots next to your project space and select Create dataset.



2. Name the new dataset *sales* and leave the other settings as their default. Then click CREATE DATASET. The new dataset should appear in your Explorer pane.



Create dataset

Dataset ID *
sales
Letters, numbers, and underscores allowed

Data location
United States (us)

Default table expiration

□ Enable table expiration

Default maximum table age — Days

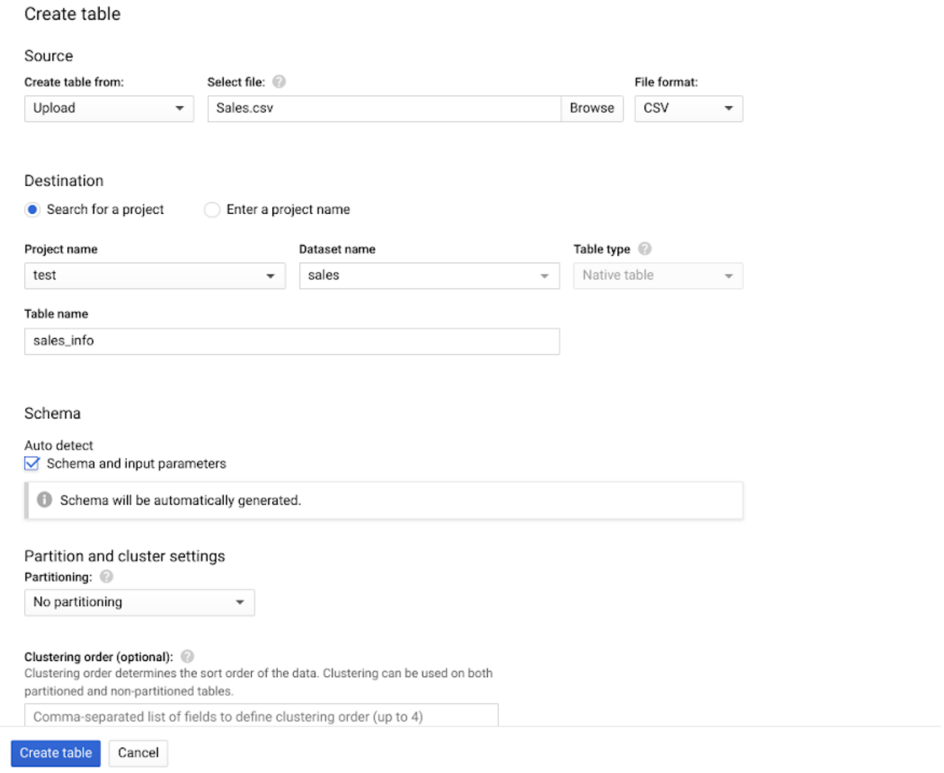
Encryption

● Google-managed encryption key
No configuration required

○ Customer-managed encryption key (CMEK)
Manage via Google Cloud Key Management Service

CREATE DATASET CANCEL

3. Open the new dataset. Click CREATE TABLE. This will open a Create table menu. Select create table from upload and import your sales data. Name the table sales_info, select Auto detect under Schema, and leave the rest of the options as default. Then select Create table.



Create table

Source

Create table from: Upload
Select file: Sales.csv Browse
File format: CSV

Destination

● Search for a project ○ Enter a project name

Project name: test
Dataset name: sales
Table type: Native table

Table name: sales_info

Schema

Auto detect
☑ Schema and input parameters
Schema will be automatically generated.

Partition and cluster settings

Partitioning: No partitioning

Clustering order (optional):
Clustering order determines the sort order of the data. Clustering can be used on both partitioned and non-partitioned tables.
Comma-separated list of fields to define clustering order (up to 4)

Create table Cancel

4. Open the new table to inspect the schema and preview your data.

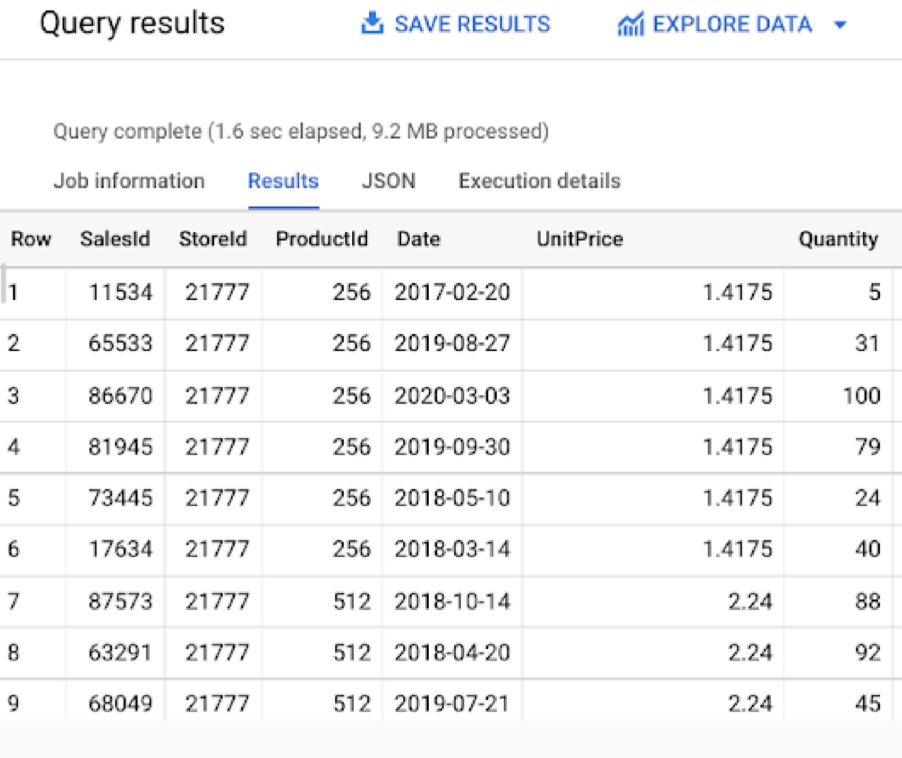
Step 2: Inspect the data

Next, you will need to inspect the data to determine how much of it will be useful for your final analysis.

1. Ensure that the import was successful by running this query:

```
SELECT
  *
FROM
  sales.sales_info
LIMIT 10;
```

Your results should appear like this:



Query complete (1.6 sec elapsed, 9.2 MB processed)

Row	SalesId	StoreId	ProductId	Date	UnitPrice	Quantity
1	11534	21777	256	2017-02-20	1.4175	5
2	65533	21777	256	2019-08-27	1.4175	31
3	86670	21777	256	2020-03-03	1.4175	100
4	81945	21777	256	2019-09-30	1.4175	79
5	73445	21777	256	2018-05-10	1.4175	24
6	17634	21777	256	2018-03-14	1.4175	40
7	87573	21777	512	2018-10-14	2.24	88
8	63291	21777	512	2018-04-20	2.24	92
9	68049	21777	512	2019-07-21	2.24	45

2. Next, inspect the data to find out how many years of sales data it includes.You can use the MIN and MAX functions to get the oldest and newest dates:

```
SELECT
  MIN(Date) AS min_date,
  MAX(Date) AS max_date
FROM
sales.sales_info;
```

Now you know what years this data covers. In this case, you'll want to group the data by month because management wants to see year-over-year changes to inventory by month.

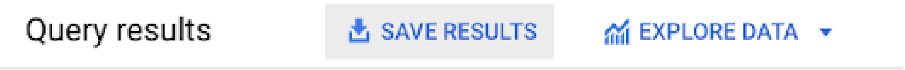
3. Click COMPOSE NEW QUERY and run the following query, which will return the total quantity sold for each ProductId grouped by the month and year it was sold:

```
SELECT
  EXTRACT(YEAR FROM date) AS Year,
  EXTRACT(MONTH FROM date) AS Month,
  ProductId,
  ROUND(MAX(UnitPrice),2) AS UnitPrice,
  SUM(Quantity) AS UnitsSold
FROM
  sales.sales_info
GROUP BY
  Year,
  Month,
  ProductId
ORDER BY
  Year,
  Month,
  ProductId;
```

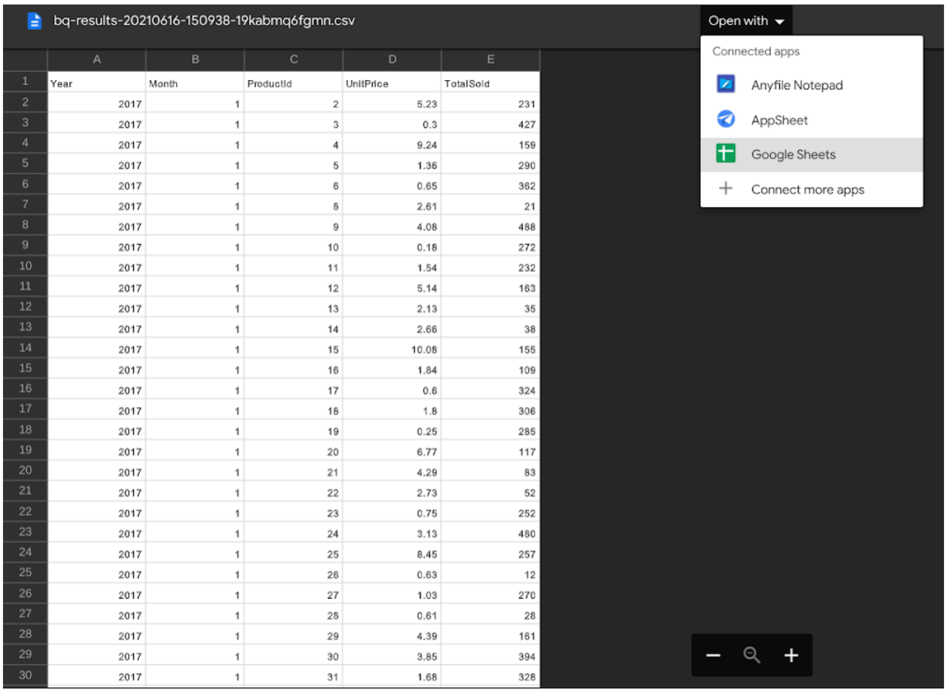
Step 3: Export results to spreadsheet

The subset of data you queried is fewer than 50,000 rows. This means it can be easily exported to a spreadsheet, if your stakeholder requests the data in this form. Or, you can use this exported spreadsheet for visualization. First, however, you'll need to save your results.

1. After running the query, click SAVE RESULTS. There will be a pop-up menu with the option to choose the file type for export. Select CSV Google Drive. Once it is downloaded, open the new CSV file in Drive.



2. Open the CSV file with Google Sheets.

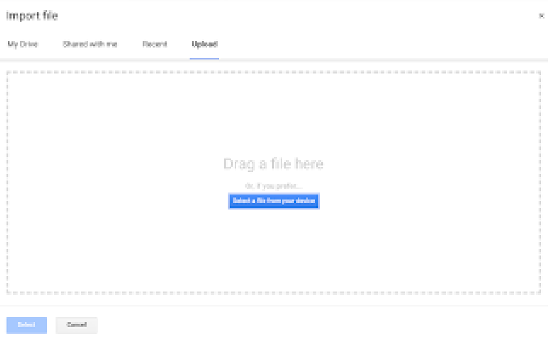

bq-results-20210616-150938-19kabmq6fgmn.csv

	A	B	C	D	E
1	Year	Month	ProductId	UnitPrice	TotalSold
2	2017	1	2	5.23	231
3	2017	1	3	0.3	427
4	2017	1	4	9.24	159
5	2017	1	5	1.36	290
6	2017	1	6	0.65	362
7	2017	1	8	2.61	21
8	2017	1	9	4.08	488
9	2017	1	10	0.18	272
10	2017	1	11	1.54	232
11	2017	1	12	5.14	163
12	2017	1	13	2.13	35
13	2017	1	14	2.66	38
14	2017	1	15	10.08	155
15	2017	1	16	1.84	109
16	2017	1	17	0.6	324
17	2017	1	18	1.8	306
18	2017	1	19	0.25	285
19	2017	1	20	6.77	117
20	2017	1	21	4.29	83
21	2017	1	22	2.73	52
22	2017	1	23	0.75	252
23	2017	1	24	3.13	480
24	2017	1	25	8.45	257
25	2017	1	26	0.63	12
26	2017	1	27	1.03	270
27	2017	1	28	0.61	28
28	2017	1	29	4.39	161
29	2017	1	30	3.85	394
30	2017	1	31	1.68	328

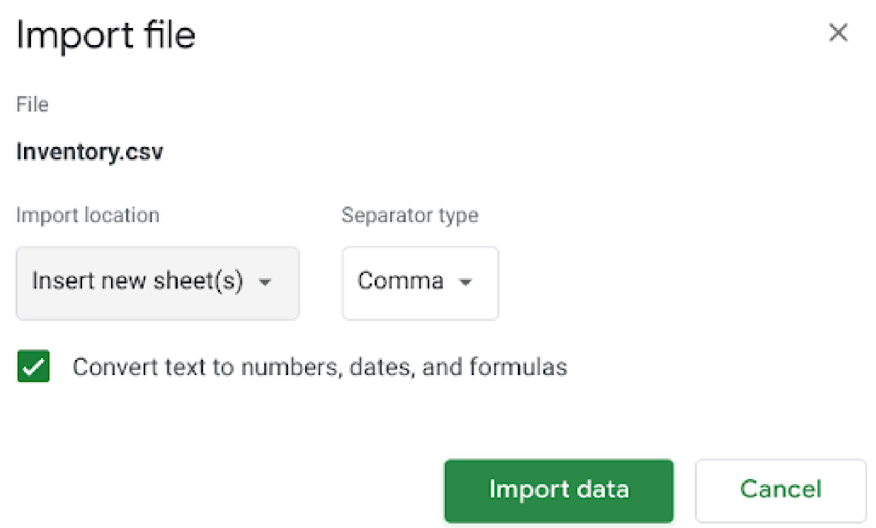
There should be about 47,000 rows. Right-click on the sheet tab and rename the sheet Sales.

3. Next, if you're using Sheets, you can open these results by selecting the File menu and clicking Import.

This will open a pop-up menu. Click Upload and select the inventory CSV file.



Select Insert new sheet(s) to add this data as a worksheet to your spreadsheet and choose Comma for Separator type.


Import file

File

Inventory.csv

Import location: Insert new sheet(s)

Separator type: Comma

☑ Convert text to numbers, dates, and formulas

4. Repeat these steps for the productsCSV file.

Confirmation and reflection

What is the earliest year included in this dataset?

○ 2017
○ 2018
○ 2019
○ 2020

2. In the text box below, write 2-3 sentences (40-60 words) in response to each of the following questions:

- Why is being able to make use of multiple analysis tools useful for some projects?
- How is working with data in spreadsheets and with SQL different? How are they similar?