

Creating a Data Warehouse Through Joins and Unions

GSP413



Google Cloud Self-Paced Labs

Overview

[BigQuery](#) is Google's fully managed, NoOps, low cost analytics database. With BigQuery you can query terabytes and terabytes of data without having any infrastructure to manage or needing a database administrator. BigQuery uses SQL and can take advantage of the pay-as-you-go model. BigQuery allows you to focus on analyzing data to find meaningful insights.

The dataset you'll use is an [ecommerce dataset](#) that has millions of Google Analytics records for the [Google Merchandise Store](#) loaded into BigQuery. You have a copy of that dataset for this lab and will explore the available fields and row for insights.

This lab focuses on how to create new reporting tables using SQL JOINS and UNIONS.

What you'll do

In this lab, you learn how to perform these tasks:

- Explore new ecommerce data on sentiment analysis
- Join together datasets and create new tables
- Append historical data with unions and table wildcards

Setup and requirements

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This Qwiklabs hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

What you need

To complete this lab, you need:

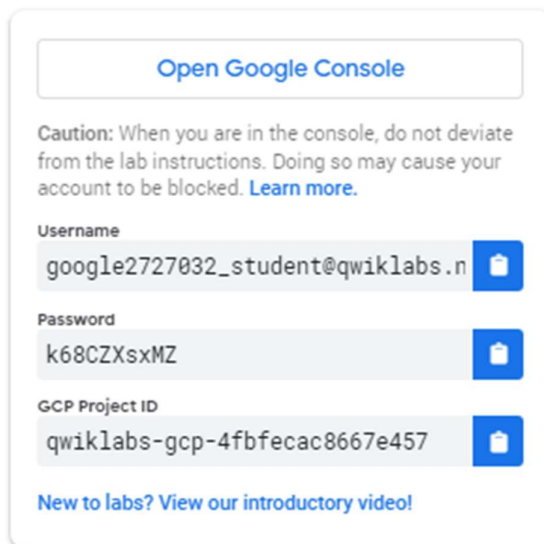
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.

Note: If you are using a Pixelbook, open an Incognito window to run this lab.

How to start your lab and sign in to the Google Cloud Console

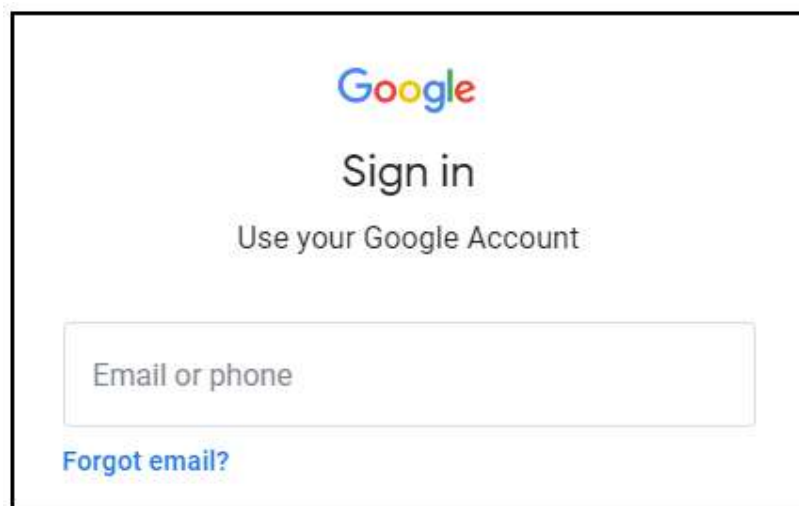
1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.



This panel contains the following information:

- Open Google Console** (button)
- Caution:** When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)
- Username:** google2727032_student@qwiklabs.n (with a copy icon)
- Password:** k68CZXsxMZ (with a copy icon)
- GCP Project ID:** qwiklabs-gcp-4fbfecac8667e457 (with a copy icon)
- [New to labs? View our introductory video!](#)

2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.



The sign-in page displays the Google logo, the text "Sign in" and "Use your Google Account". It features a text input field labeled "Email or phone" and a link for "Forgot email?" below it.

Tip: Open the tabs in separate windows, side-by-side.

If you see the **Choose an account** page, click **Use Another**



Account.

3. In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

4. Click through the subsequent pages:

- Accept the terms and conditions.
- Do not add recovery options or two-factor authentication (because this is a temporary account).
- Do not sign up for free trials.

After a few moments, the Cloud Console opens in this tab.

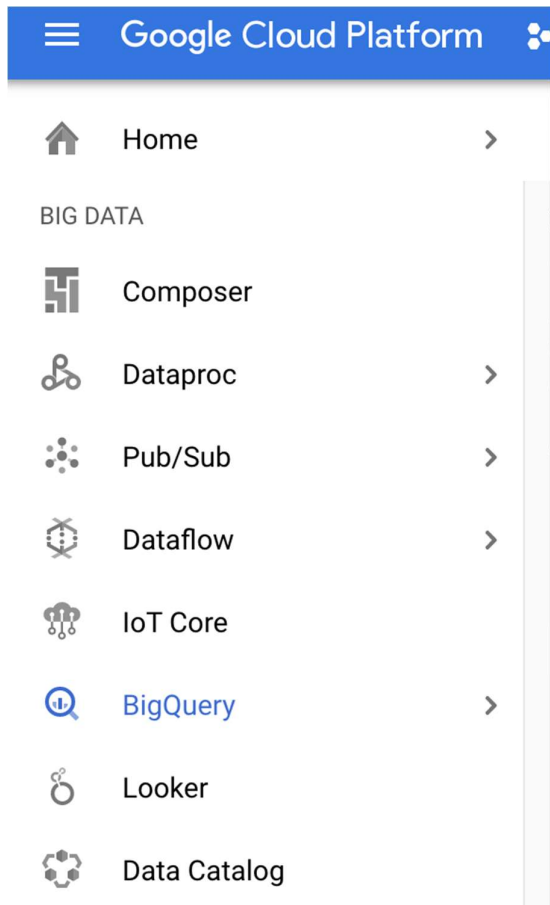
Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-left.



The BigQuery console

Open BigQuery Console

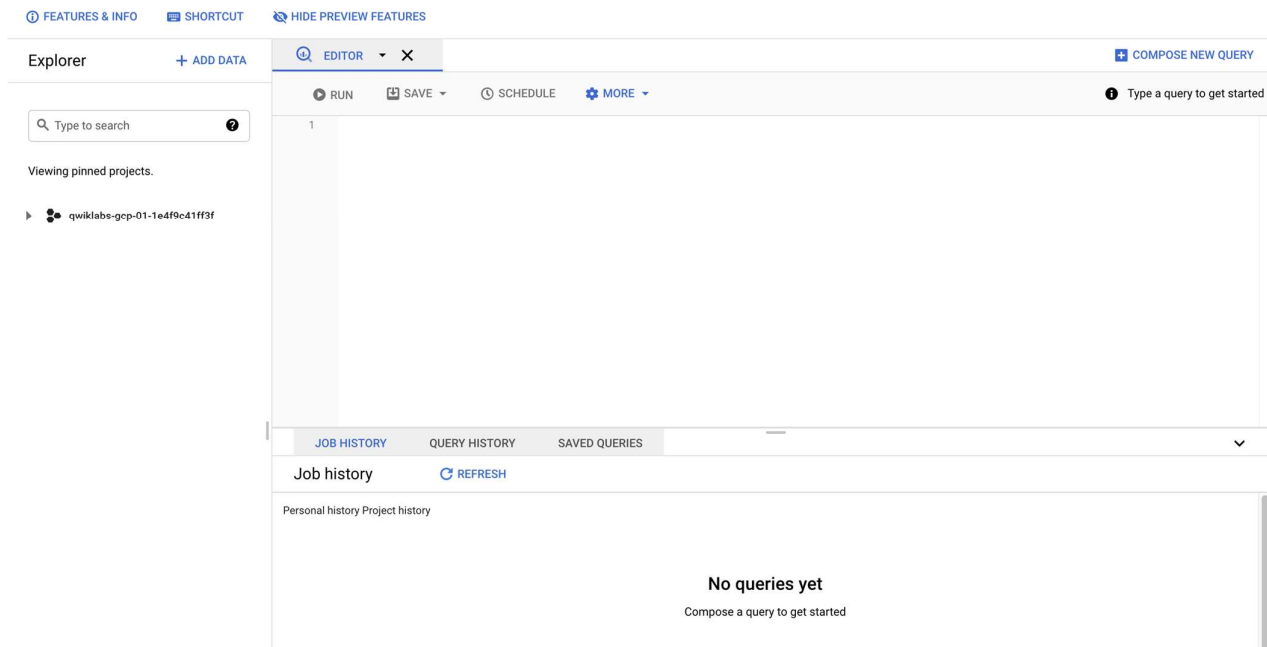
In the Google Cloud Console, select **Navigation menu** > **BigQuery**:



The **Welcome to BigQuery in the Cloud Console** message box opens. This message box provides a link to the quickstart guide and the release notes.

Click **Done**.

The BigQuery console opens.



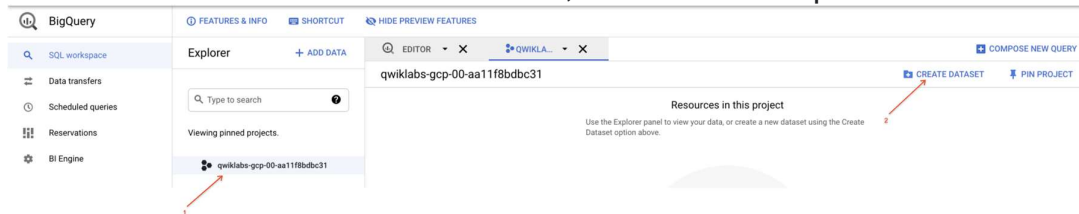
Create a new dataset to store your tables

In your BigQuery project, create a new dataset titled **ecommerce**.

1. In the left pane, click on your BigQuery project (`qwiklabs-gcp-xxxx`).
2. To the right, click **CREATE DATASET**.

The **Create dataset** dialog opens.

3. Set the *Dataset ID* to `ecommerce`, leave all other options at their default values.



Click **Create dataset**.

Click *Check my progress* to verify the objective.

Create a new dataset to store the tables

Check my progress

Scenario: Your marketing team provided you and your data science team all of the product reviews for your ecommerce website. You partner with them to create a data warehouse in BigQuery which joins together data from three sources:

- Website ecommerce data
- Product inventory stock levels and lead times
- Product review sentiment analysis

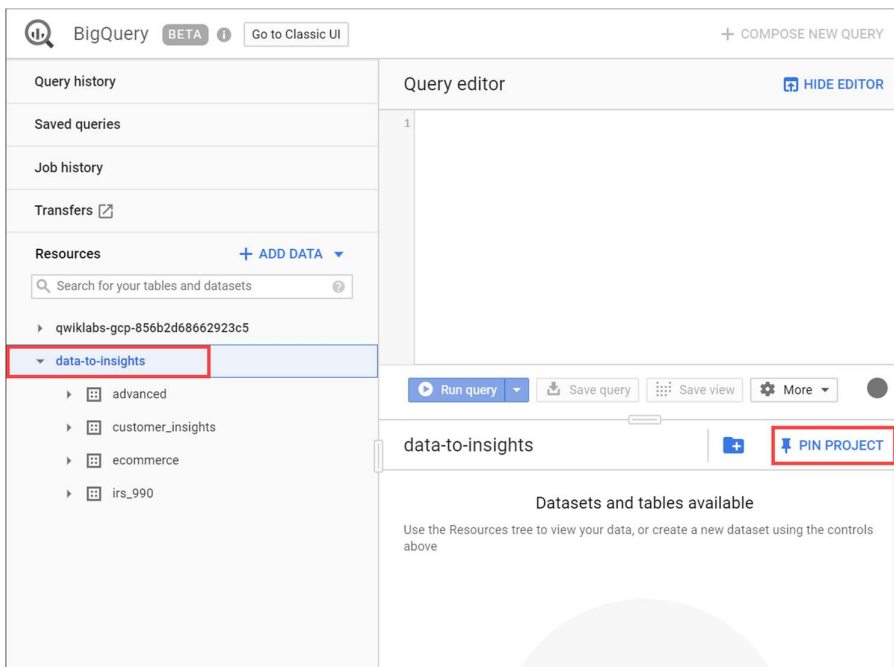
In this lab, you examine a new dataset based on product reviews.

Pin the project to your Resource Tree

The project with your marketing team's new dataset is **data-to-insights**.

BigQuery public datasets are not displayed by default in the BigQuery web UI. Since **data-to-insights** is a public dataset project, you have to pin it to your Resource Tree:

1. Click on **HIDE PREVIEW FEATURES**.
2. In a new browser window, open the public datasets project, <https://console.cloud.google.com/bigquery?p=data-to-insights&page=ecommerce>.
3. In the left pane, in the Resource section, click **data-to-insights**. In the right pane, click **Pin Project**.



4. Close this browser window.
5. Return to and refresh the first BigQuery browser window to refresh the BigQuery web UI.

The **data-to-insights** project is listed in the Resource section.

6. Click on **SHOW PREVIEW FEATURES**.

Enrich ecommerce data with Machine Learning

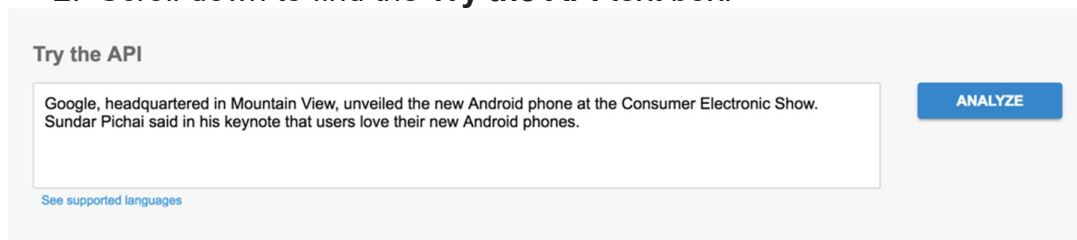
In this section, you get a feel for how Google's sentiment analysis API works to analyze the sentiment of product reviews.

Sentiment analysis attempts to determine the overall attitude (positive or negative) expressed within text. Sentiment is represented by numerical score and magnitude values.

- *Score* of the sentiment ranges between -1.0 (negative) and 1.0 (positive) and corresponds to the overall emotional leaning of the text.
- *Magnitude* indicates the overall strength of emotion (both positive and negative) within the given text, between 0.0 and $+\infty$. Unlike score, magnitude is not normalized; each expression of emotion within the text (both positive and negative) contributes to the text's magnitude (so longer text blocks may have greater magnitudes).

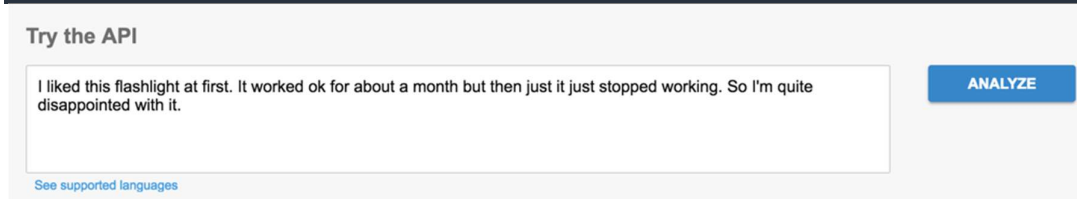
Use some fictional product reviews to checkout how the Sentiment Analysis API works.

1. Open the [Cloud Natural Language](#) page in a new browser window or tab.
2. Scroll down to find the **Try the API** text box.

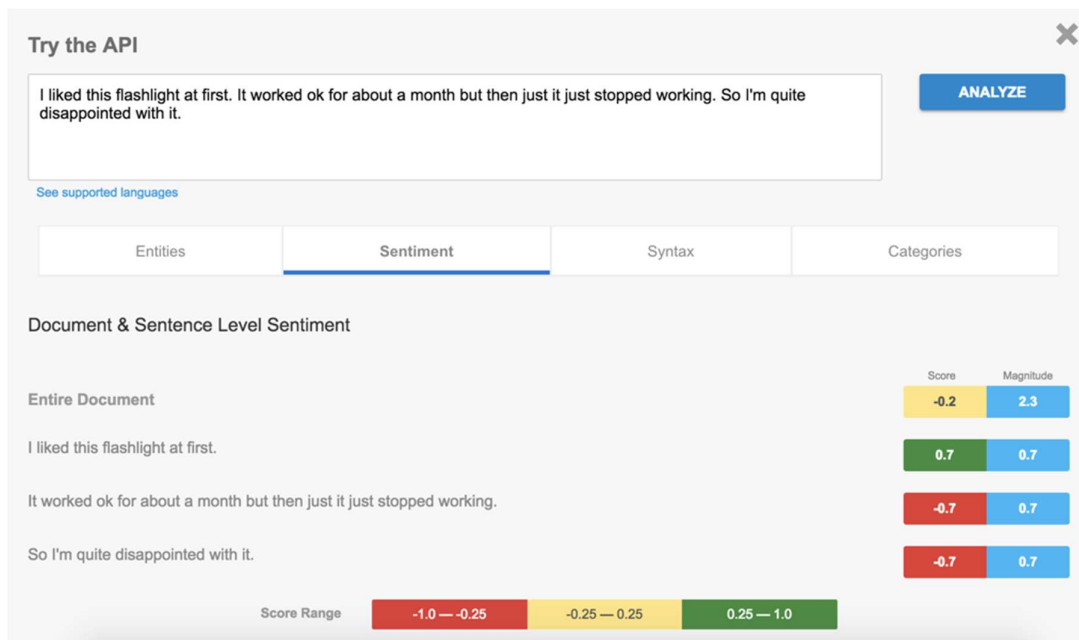


3. Replace the text inside the text box with the following text:

```
I liked this flashlight at first. It worked ok for about a month but then just it just stopped working. So I'm quite disappointed with it.
```



4. Click **Analyze**.
5. Click the **Sentiment** tab.



What are the sentiment score and magnitude values for the first sentence?

checkScore = `0.7`. Magnitude = `0.7`

closeScore = `-0.25`. Magnitude = `-1.0`

closeScore = `0.7`. Magnitude = `0.2`

Submit

Is the overall sentiment for the entire produce review positive or negative?

closeNeutral, not positive or negative

closePositive

checkNegative

Submit

Analyze the following product reviews. Keep track of Scores as you go as you'll be asked. Which of those product reviews have the most positive sentiment? The most negative sentiment? The most neutral sentiment?

Review #1:

The three dog frisbees we ordered unfortunately didn't do well with our bigger German Shepherd dogs.

Review #2:

The three dog frisbees we ordered unfortunately didn't do well with our bigger German Shepherd dogs. Firstly, they had a tough time catching them in the air since the light blue models matched the color of the sky and secondly the material they were made out of wasn't strong enough to withstand more than a couple months of use before they got chewed up.

Review #3:

Honestly I've gone through quite a few umbrellas in the past but this new red Executive Umbrella is one of the best. We ended up going with red but you have a wide variety of colors to choose from. The umbrella material was excellent and didn't degrade after heavy use (we're in Seattle!).

Review #4:

I love these sunglasses. They are sturdy, look nice, and are functional. Highly recommended!

Review #5:

I got one of the microfleece jackets as a gift and wear it most days. The material is good and not itchy.

Review #6:

I pre-ordered a few yoga blocks but my shipment kept getting delayed because of supplier delays. Not sure what's going on there but would be great to speed up shipment.

Which review is the most positive?

☐ Review 6

☐ Review 3

☐ Review 4

☐ Review 2

Which review is the most negative?

☐ Review 2

☐ Review 5

☐ Review 6

☐ Review 4

Which review is the most neutral?

☐ Review 5

☐ Review 1

☐ Review 2

☐ Review 6

Explore the product sentiment dataset

Your data science team has run all of your product reviews through the API and provided you with the average sentiment score and magnitude for each of your products.

Examine the data.

1. Navigate to the **data-to-insight > ecommerce > products** dataset and click the **Preview** tab to see the data.

How many Aluminum Handy Emergency Flashlights have been ordered?

☐ 85
☐ 90
☐ 0
☒ 66
Submit

2. Click the **Schema** tab.

What data type are the sentimentScore and sentimentMagnitude fields?

☒ INTERGER
☐ FLOAT
☐ RECORD
☐ STRING
Submit

Create a query that shows the top 5 products with the most positive sentiment

In the **Query Editor**, write your SQL query.

Possible Solution:

```
#standardSQL
SELECT
  SKU,
  name,
  sentimentScore,
  sentimentMagnitude
FROM
  `data-to-insights.ecommerce.products`
ORDER BY
  sentimentScore DESC
LIMIT 5
```

What product has the highest sentiment?

☐ Stylus Pen w/ LED Light
☒ USB wired soundbar - in store only
☐ G Noise-reducing Bluetooth Headphones
☐ G Noise-reducing Bluetooth Headphones
Submit

Revise your query to show the top 5 products with the most negative sentiment. Filter out NULL values.

Possible Solution:

```
#standardSQL
SELECT
  SKU,
  name,
  sentimentScore,
  sentimentMagnitude
FROM
  `data-to-insights.ecommerce.products`
WHERE sentimentScore IS NOT NULL
ORDER BY
  sentimentScore
LIMIT 5
```

What is the product with the lowest sentiment?

What is the product with the lowest sentiment?



Womens Convertible Vest-Jacket Sea Foam Green



Mens Vintage Henley



4 Womens Vintage Hero Tee Platinum

check7 inch Dog Frisbee

Submit

Click *Check my progress* to verify the objective.

Explore the product sentiment dataset

Check my progress

Join datasets to find insights

It's the first of the month and your inventory team has informed you that the `orderedQuantity` field in the product inventory dataset is out of date. They need your help to query the total sales by product for 08/01/2017 and reference that against the current stock levels in inventory to see which products need to be resupplied first.

Calculate daily sales volume by productSKU

Create a new table in your **ecommerce** dataset with the below requirements:

- Title it `sales_by_sku_20170801`
- Source the data from `data-to-insights.ecommerce.all_sessions_raw`
- Include only distinct results
- Return `productSKU`
- Return the total quantity ordered (`productQuantity`). Hint: Use a `SUM()` with a `IFNULL` condition
- Filter for only sales on 20170801
- ORDER BY the SKUs with the most orders first

Possible Solution:

```
# pull what sold on 08/01/2017
CREATE OR REPLACE TABLE ecommerce.sales_by_sku_20170801 AS
SELECT DISTINCT
  productSKU,
  SUM(IFNULL(productQuantity,0)) AS total_ordered
FROM
  `data-to-insights.ecommerce.all_sessions_raw`
WHERE date = '20170801'
GROUP BY productSKU
ORDER BY total_ordered DESC #462 skus sold
```

Click the **Preview** tab. How many distinct product SKUs were sold?

Answer: 462

True or false: GGOEGOAQ012899 is the top selling product SKU.

checkTrue



False

Next, enrich your sales data with product inventory information by joining the two datasets.

Join sales data and inventory data

Using a join, enrich the website ecommerce data with the following fields from the product inventory dataset:

- `name`
- `stockLevel`
- `restockingLeadTime`
- `sentimentScore`

- sentimentMagnitude

Complete the partially written query:

```
# standardSQL
# join against product inventory to get name
SELECT DISTINCT
  website.productSKU,
  website.total_ordered,
  inventory.name,
  inventory.stockLevel,
  inventory.restockingLeadTime,
  inventory.sentimentScore,
  inventory.sentimentMagnitude
FROM
  ecommerce.sales_by_sku 20170801 AS website
  LEFT JOIN `data-to-insights.ecommerce.products` AS inventory

ORDER BY total_ordered DESC
```

Possible Solution:

```
# standardSQL
# join against product inventory to get name
SELECT DISTINCT
  website.productSKU,
  website.total_ordered,
  inventory.name,
  inventory.stockLevel,
  inventory.restockingLeadTime,
  inventory.sentimentScore,
  inventory.sentimentMagnitude
FROM
  ecommerce.sales_by_sku 20170801 AS website
  LEFT JOIN `data-to-insights.ecommerce.products` AS inventory
  ON website.productSKU = inventory.SKU
ORDER BY total_ordered DESC
```

Modify the query you wrote to now include:

- A calculated field of $(total_ordered / stockLevel)$ and alias it "ratio". Hint: Use `SAFE_DIVIDE(field1,field2)` to avoid divide by 0 errors when the stock level is 0.
- Filter the results to only include products that have gone through 50% or more of their inventory already at the beginning of the month

Possible Solution:

```
#standardSQL
# calculate ratio and filter
SELECT DISTINCT
  website.productSKU,
  website.total_ordered,
  inventory.name,
  inventory.stockLevel,
  inventory.restockingLeadTime,
  inventory.sentimentScore,
  inventory.sentimentMagnitude,

  SAFE_DIVIDE(website.total_ordered, inventory.stockLevel) AS ratio
FROM
  ecommerce.sales_by_sku 20170801 AS website
  LEFT JOIN `data-to-insights.ecommerce.products` AS inventory
  ON website.productSKU = inventory.SKU
```

```
# gone through more than 50% of inventory for the month
WHERE SAFE_DIVIDE(website.total_ordered,inventory.stockLevel) >= .50

ORDER BY total_ordered DESC
```

What is the name of the top selling product and what percent of its inventory has been sold already?



Android Infant Short Sleeve Tee Pewter with 7 product orders out of 2 in stock
checkLeather Journal-Black with 250 product orders out of 354 in stock



Youth Short Sleeve Tee Red with a restocking leadtime of 9

Submit

Click *Check my progress* to verify the objective.

Join datasets to find insights

Check my progress

Append additional records

Your international team has already made in-store sales on 08/02/2017 which you want to record in your daily sales tables.

Create a new empty table to store sales by productSKU for 08/02/2017

For the schema, specify the following fields:

- table name is `ecommerce.sales_by_sku_20170802`
- `productSKU` `STRING`
- `total_ordered` as an `INT64` field

Possible Solution:



```
#standardSQL
CREATE OR REPLACE TABLE ecommerce.sales_by_sku_20170802
(
  productSKU STRING,
  total_ordered INT64
);
```


Confirm you now have two date-shared sales tables (you may have to refresh your browser window):


[FEATURES & INFO](#) [SHORTCUT](#)



Explorer [+ ADD DATA](#)

Viewing pinned projects.

▼  qwiklabs-gcp-00-aa11f8bdb31 

▼  ecommerce

 sales_by_sku_ (2)

▶  data-to-insights 

Insert the sales record provided to you by your sales team:

```
#standardSQL
INSERT INTO ecommerce.sales_by_sku_20170802
(productSKU, total_ordered)
VALUES('GGOEGHPA002910', 101)
```

Confirm the record appears by previewing the table.

Append together historical data

There are multiple ways to append together data that has the same schema. Two common ways are using UNIONS and table wildcards.

- **Union** is an SQL operator that appends together rows from different result sets.
 - **Table wildcards** enable you to query multiple tables using concise SQL statements. Wildcard tables are available only in standard SQL.
- Write a UNION query that will result in all records from the below two tables:

- `ecommerce.sales_by_sku_20170801`
- `ecommerce.sales_by_sku_20170802`

```
#standardSQL
SELECT * FROM ecommerce.sales_by_sku_20170801
UNION ALL
SELECT * FROM ecommerce.sales_by_sku_20170802
```

Note: The difference between a UNION and UNION ALL is that a UNION will not include duplicate records.

What is a pitfall of having many daily sales tables?

Answer: You will have to write many UNION statements chained together.

A better solution is to use the table wildcard filter and `_TABLE_SUFFIX` filter.

Write a query that uses the (*) table wildcard to select all records from `ecommerce.sales_by_sku_` for the year 2017.

Possible Solution:

```
#standardSQL
SELECT * FROM `ecommerce.sales_by_sku_2017*`
```

Modify the previous query to add a filter to limit the results to just 08/02/2017.

Possible Solution:

```
#standardSQL
SELECT * FROM `ecommerce.sales_by_sku_2017*`
WHERE _TABLE_SUFFIX = '0802'
```

Note: Another option to consider is to create a Partitioned Table which automatically can ingest daily sales data into the correct partition.

A UNION ALL join does not include duplicate records.

`close=True`

`check=False`

Click *Check my progress* to verify the objective.

Congratulations!

This concludes this hands-on lab. You explored sample ecommerce data by creating reporting tables and then manipulating views using SQL JOINS and UNIONS.



Finish your Quest

This self-paced lab is part of the Qwiklabs [BigQuery for Data Warehousing](#) Quest. A Quest is a series of related labs that form a learning path. Completing this Quest earns you the badge above, to recognize your achievement. You can make your badge (or badges) public and link to them in your online resume or social media account. [Enroll in this Quest](#) and get immediate completion credit if you've taken this lab. See other available [Qwiklabs Quests](#).

Take your next lab

Continue with another lab in the Quest, for example [Working with JSON, Arrays, and Structs in BigQuery](#), or check out these other labs:

- [Exploring NCAA Data with BigQuery](#)
- [Cloud Composer: Copying BigQuery Tables Across Different Locations](#)

Next steps / learn more

- Already have a Google Analytics account and want to query your own datasets in BigQuery? Follow this [export guide](#).
- If you want to practice with more SQL syntax for JOINS, check out the [BigQuery JOIN documentation](#).
- Try [Google Dataset Search](#) as a resource!

Google Cloud Training & Certification

...helps you make the most of Google Cloud technologies. [Our classes](#) include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. [Certifications](#) help you validate and prove your skill and expertise in Google Cloud technologies.

Manual Last Updated January 29, 2021

Lab Last Tested January 29, 2021

Copyright 2021 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.