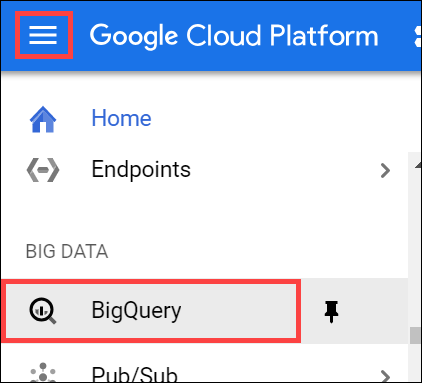
**Open BigQuery and Pin a project to the Resource tree**

1. Click **Navigation menu** > **BigQuery**.



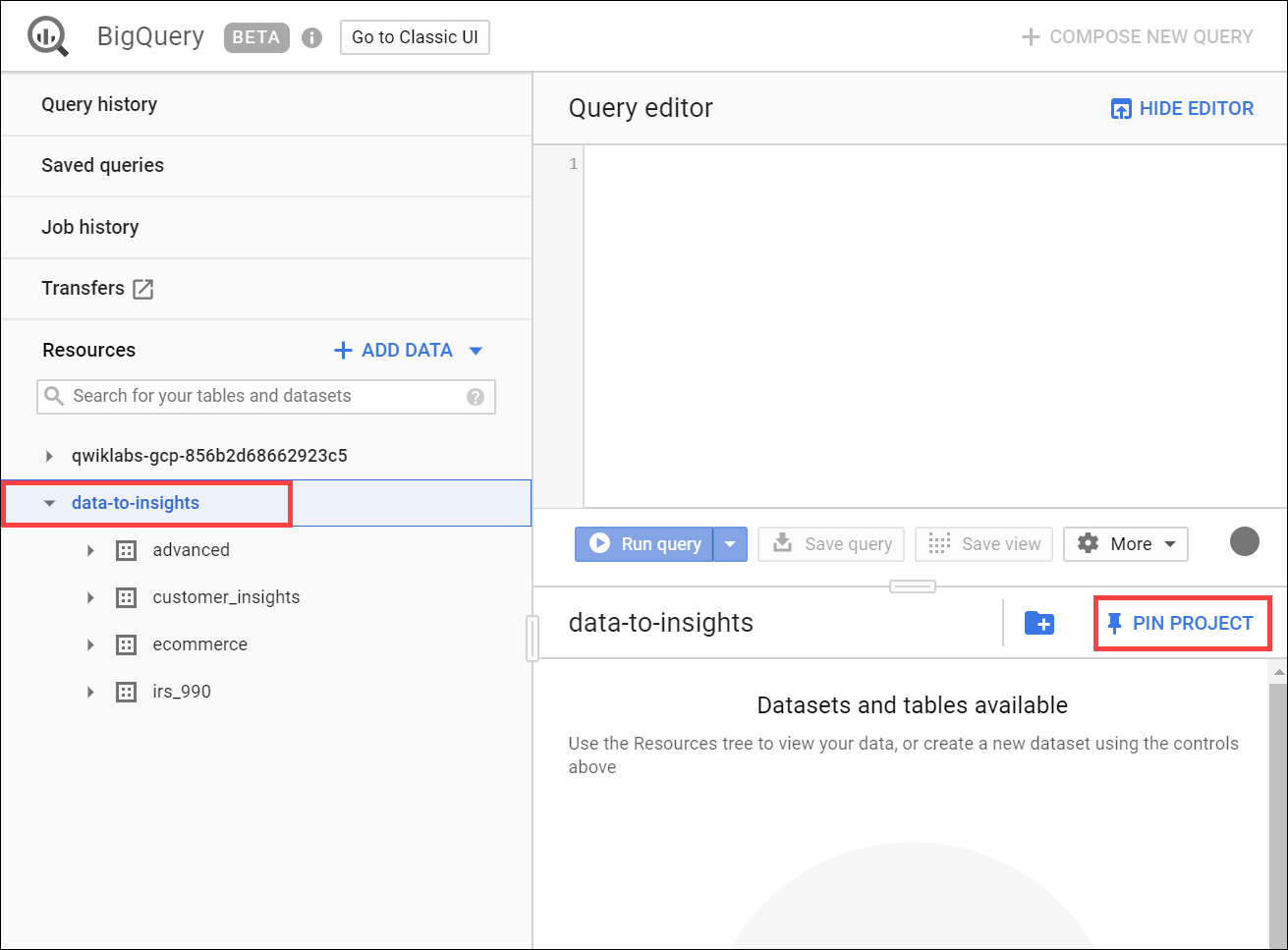
The Welcome to BigQuery in the Cloud Console message box opens.

The Welcome to BigQuery in the Cloud Console message box provides a link to the quickstart guide and UI updates.

1. Click **Done**.

BigQuery public datasets are not displayed by default in the BigQuery web UI. To open the public datasets project, open <https://console.cloud.google.com/bigquery?p=data-to-insights&page=ecommerce> in a new browser window.

1. In the left pane, in the Resource section, click **data-to-insights**. In the right pane, click **Pin Project**.



1. Close this browser window.
2. 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.

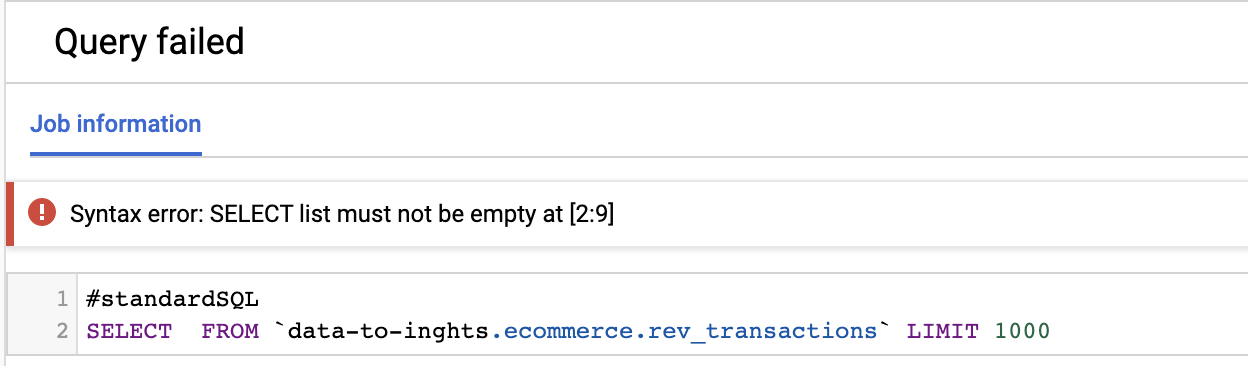
**BigQuery Code editor**

For each activity in the following sections, this lab provides queries with common errors for you to troubleshoot. The lab directs you what to look at and suggests how to correct the syntax and return meaningful results.

To follow along with the troubleshooting and suggestions, copy and paste the query into the BigQuery Query editor. If there are errors, you see a red exclamation point at the line containing the error and in the query validator (bottom corner).



If you run the query with the errors, the query fails and the error is specified in the Job information.



When the query is error free, you see a green checkmark in the query validator. When you see the green checkmark, click **Run** to run the query to view what you get for results.



For information about syntax, see [Standard SQL Query Syntax](https://cloud.google.com/bigquery/docs/reference/standard-sql/query-syntax).

**Find the total number of customers who went through checkout**

Your goal in this section is to construct a query that gives you the number of unique visitors who successfully went through the checkout process for your website. The data is in the rev\_transactions table which your data analyst team has provided. They have also given you example queries to help you get started in your analysis but you're not sure they're written correctly.

Troubleshoot queries that contain query validator, alias, and comma errors

Look at the below query and answer the following question:

#standardSQL

SELECT FROM `data-to-inghts.ecommerce.rev\_transactions` LIMIT 1000

What's wrong with the previous query to view 1000 items?

checkWe have not specified any columns in the SELECT

checkThere is a typo in the dataset name



We are using legacy SQL



There is a typo in the table name

Submit

What about this updated query?

#standardSQL

SELECT \* FROM [data-to-insights:ecommerce.rev\_transactions] LIMIT 1000

What's wrong with the new previous query to view 1000 items?



There is a typo in the dataset name

checkWe are using legacy SQL



There is a typo in the table name



We have not specified any columns in the SELECT

Submit

What about this query that uses Standard SQL?

#standardSQL

SELECT FROM `data-to-insights.ecommerce.rev\_transactions`

What is wrong with the previous query?



We are still using legacy SQL



SELECT clause is returning all columns which leads to poor performance

checkStill no columns defined in SELECT



We are missing an ORDER BY clause

Submit

What about now? This query has a column.

#standardSQL

SELECT

fullVisitorId

FROM `data-to-insights.ecommerce.rev\_transactions`

What is wrong with the previous query?



We are missing a column alias



The page title is missing from the columns in SELECT



We are missing a LIMIT clause



Without aggregations, limits, or sorting, this query is not insightful

Submit

One or more of your choices is incorrect.

What about now? The following query has a page title.

#standardSQL

SELECT fullVisitorId hits\_page\_pageTitle

FROM `data-to-insights.ecommerce.rev\_transactions` LIMIT 1000

How many columns will the previous query return?



3 columns will be returned since we are missing a comma



0, the query will return an error

close~~2, columns named fullVisitorId and hits\_page\_pageTitle~~

check1, a column named hits\_page\_pageTitle

Submit

What about now? The missing comma has been corrected.

#standardSQL

SELECT

fullVisitorId

, hits\_page\_pageTitle

FROM `data-to-insights.ecommerce.rev\_transactions` LIMIT 1000

Answer: This returns results, but are you sure visitors aren't counted twice? Also, returning only one row answers the question of how many unique visitors reached checkout. In the next section you find a way to aggregate your results.

Troubleshoot queries that contain logic errors, GROUP BY statements, and wildcard filters

Aggregate the following query to answer the question: How many unique visitors reached checkout?

#standardSQL

SELECT

fullVisitorId

, hits\_page\_pageTitle

FROM `data-to-insights.ecommerce.rev\_transactions` LIMIT 1000

What about this? An aggregation function, COUNT(), was added.

#standardSQL

SELECT

COUNT(fullVisitorId) AS visitor\_count

, hits\_page\_pageTitle

FROM `data-to-insights.ecommerce.rev\_transactions`

What is wrong with the previous query?

checkThe COUNT() function does not de-deduplicate the same fullVisitorId



A COUNT() function is used when SUM() should be used instead



Nothing, it executes correctly

checkIt is missing a GROUP BY clause

Submit

In this next query, GROUP BY and DISTINCT statements were added.

#standardSQL

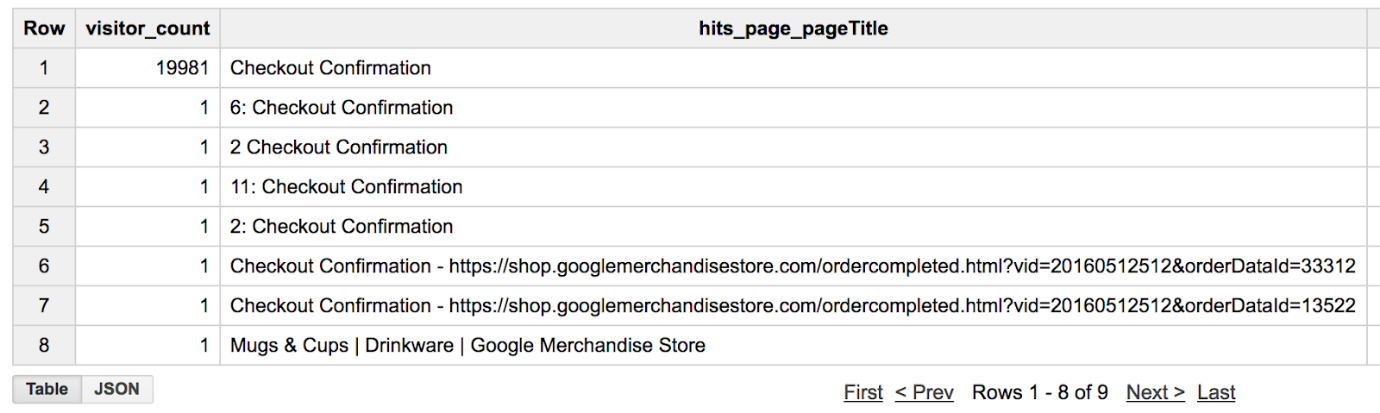
SELECT

COUNT(DISTINCT fullVisitorId) AS visitor\_count

, hits\_page\_pageTitle

FROM `data-to-insights.ecommerce.rev\_transactions`

GROUP BY hits\_page\_pageTitle

**Results** 

Great! The results are good, but they look strange. Filter to just "Checkout Confirmation" in the results.

#standardSQL

SELECT

COUNT(DISTINCT fullVisitorId) AS visitor\_count

, hits\_page\_pageTitle

FROM `data-to-insights.ecommerce.rev\_transactions`

WHERE hits\_page\_pageTitle = "Checkout Confirmation"

GROUP BY hits\_page\_pageTitle

Click *Check my progress* to verify the objective.

Find the total number of customers went through checkout

Check my progress

**List the cities with the most transactions with your ecommerce site**

Troubleshoot ordering, calculated fields, and filtering after aggregating errors

Complete the partially written query:

SELECT

geoNetwork\_city,

totals\_transactions,

COUNT( DISTINCT fullVisitorId) AS distinct\_visitors

FROM

`data-to-insights.ecommerce.rev\_transactions`

GROUP BY

**Possible solution**

#standardSQL

SELECT

geoNetwork\_city,

SUM(totals\_transactions) AS totals\_transactions,

COUNT( DISTINCT fullVisitorId) AS distinct\_visitors

FROM

`data-to-insights.ecommerce.rev\_transactions`

GROUP BY geoNetwork\_city

Update your previous query to order the top cities first.

Which city had the most distinct visitors? Ignore the value: 'not available in this demo dataset'

checkMountain View



Austin

close~~San Jose~~



Los Angeles

Submit

**Possible solution**

#standardSQL

SELECT

geoNetwork\_city,

SUM(totals\_transactions) AS totals\_transactions,

COUNT( DISTINCT fullVisitorId) AS distinct\_visitors

FROM

`data-to-insights.ecommerce.rev\_transactions`

GROUP BY geoNetwork\_city

ORDER BY distinct\_visitors DESC

Update your query and create a new calculated field to return the average number of products per order by city.

**Possible solution**

#standardSQL

SELECT

geoNetwork\_city,

SUM(totals\_transactions) AS total\_products\_ordered,

COUNT( DISTINCT fullVisitorId) AS distinct\_visitors,

SUM(totals\_transactions) / COUNT( DISTINCT fullVisitorId) AS avg\_products\_ordered

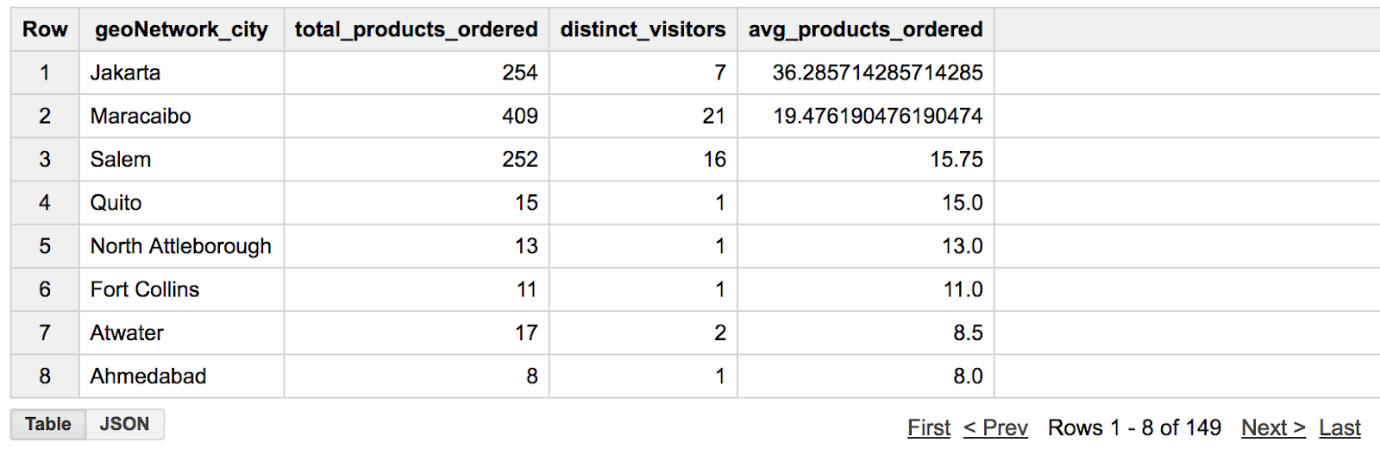
FROM

`data-to-insights.ecommerce.rev\_transactions`

GROUP BY geoNetwork\_city

ORDER BY avg\_products\_ordered DESC

**Results**



Filter your aggregated results to only return cities with more than 20 avg\_products\_ordered.

What's wrong with the following query?

#standardSQL

SELECT

geoNetwork\_city,

SUM(totals\_transactions) AS total\_products\_ordered,

COUNT( DISTINCT fullVisitorId) AS distinct\_visitors,

SUM(totals\_transactions) / COUNT( DISTINCT fullVisitorId) AS avg\_products\_ordered

FROM

`data-to-insights.ecommerce.rev\_transactions`

WHERE avg\_products\_ordered > 20

GROUP BY geoNetwork\_city

ORDER BY avg\_products\_ordered DESC

What is wrong with the previous query?

checkYou cannot filter aggregated fields in the `WHERE` clause (use `HAVING` instead)



You cannot divide non-similar aggregate functions

checkYou cannot filter on aliased fields within the `WHERE` clause



Nothing, it executes scorrectly

Submit

**Possible solution**

#standardSQL

SELECT

geoNetwork\_city,

SUM(totals\_transactions) AS total\_products\_ordered,

COUNT( DISTINCT fullVisitorId) AS distinct\_visitors,

SUM(totals\_transactions) / COUNT( DISTINCT fullVisitorId) AS avg\_products\_ordered

FROM

`data-to-insights.ecommerce.rev\_transactions`

GROUP BY geoNetwork\_city

HAVING avg\_products\_ordered > 20

ORDER BY avg\_products\_ordered DESC

Click *Check my progress* to verify the objective.

List the cities with the most transactions with your ecommerce site

Check my progress

It doesn't look like you've completed this step yet. Try again.

**Find the total number of products in each product category**

Find the top selling products by filtering with NULL values

What's wrong with the following query? How can you fix it?

#standardSQL

SELECT hits\_product\_v2ProductName, hits\_product\_v2ProductCategory

FROM `data-to-insights.ecommerce.rev\_transactions`

GROUP BY 1,2

What is wrong with the previous query?

checkNo aggregate functions are used



Nothing, it executes correctly



There is a typo in the column name

checkLarge GROUP BYs really hurt performance (consider filtering first and/or using aggregation functions)

Submit

What is wrong with the following query?

#standardSQL

SELECT

COUNT(hits\_product\_v2ProductName) as number\_of\_products,

hits\_product\_v2ProductCategory

FROM `data-to-insights.ecommerce.rev\_transactions`

WHERE hits\_product\_v2ProductName IS NOT NULL

GROUP BY hits\_product\_v2ProductCategory

ORDER BY number\_of\_products DESC

What is wrong with the previous query which lists products?



The WHERE clause should include NULL Product Names

close~~The GROUP BY contains an incorrect column~~



Nothing, the query executes correctly

checkThe COUNT() function is not the distinct number of products in each category

Submit

Update the previous query to only count distinct products in each product category.

**Possible solution**

#standardSQL

SELECT

COUNT(DISTINCT hits\_product\_v2ProductName) as number\_of\_products,

hits\_product\_v2ProductCategory

FROM `data-to-insights.ecommerce.rev\_transactions`

WHERE hits\_product\_v2ProductName IS NOT NULL

GROUP BY hits\_product\_v2ProductCategory

ORDER BY number\_of\_products DESC

LIMIT 5

Which category has the most distinct number of products offered?



Office

check(not set)



Electronics



${productitem.product.origCatName}

Submit

Notes:

* (not set) could indicate the product has no category
* ${productitem.product.origCatName} is front-end code to render the category which may indicate the Google Analytics tracking script is firing before the page is fully-rendered

Click *Check my progress* to verify the objective.

Find the total number of products in each product category

Check my progress

**Congratulations!**

You troubleshot and fixed broken queries in BigQuery standard SQL. Remember to use the Query Validator for incorrect query syntax but also to be critical of your query results even if your query executes successfully.

Finish your Quest

This self-paced lab is part of the Qwiklabs [BigQuery Basics for Data Analysts](https://google.qwiklabs.com/quests/69) and [BigQuery for Data Analyis](https://google.qwiklabs.com/quests/55) Quests. 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 a Quest and get immediate completion credit if you've taken this lab. See other available [See other available Qwiklabs Quests](https://google.qwiklabs.com/catalog).