Exploring and Creating an Ecommerce Analytics Pipeline with Cloud Dataprep v1.5

1 hour 30 minutes 1 Credit
Rate Lab

Overview

Cloud Dataprep by Trifacta is an intelligent data service for visually exploring, cleaning, and preparing structured and unstructured data for analysis. In this lab we will explore the Cloud Dataprep UI to build an ecommerce transformation pipeline that will run at a scheduled interval and output results back into BigQuery.

The dataset we will be using is an ecommerce dataset that has millions of Google Analytics records for the Google Merchandise Store loaded into BigQuery. We've made a copy of that dataset for this lab and will be exploring the available fields and row for insights.

Objectives

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

- Connect BigQuery datasets to Cloud Dataprep
- Explore dataset quality with Cloud Dataprep
- Create a data transformation pipeline with Cloud Dataprep
- Schedule transformation jobs outputs to BigQuery

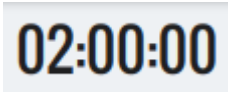
What you'll need

- A Google Cloud Platform project
- The Google Chrome browser. Other browsers are not supported by Cloud Dataprep.

Task 1. Set up your development environment

Set up Qwiklabs

For each lab, you get a new GCP project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, 

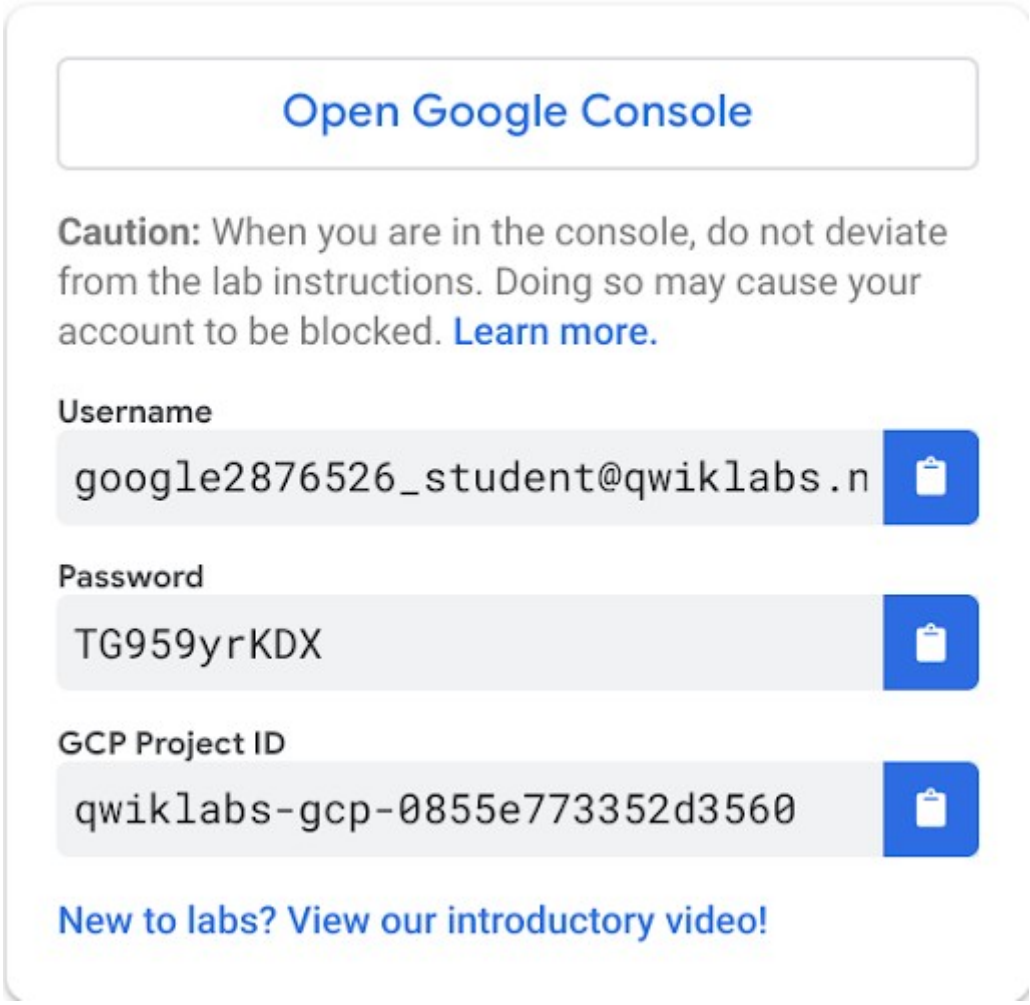
and make sure you can finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.

3. When ready, click 

.

4. Note your lab credentials. You will use them to sign in to Cloud Platform Console.



5. Click **Open Google Console**.
6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

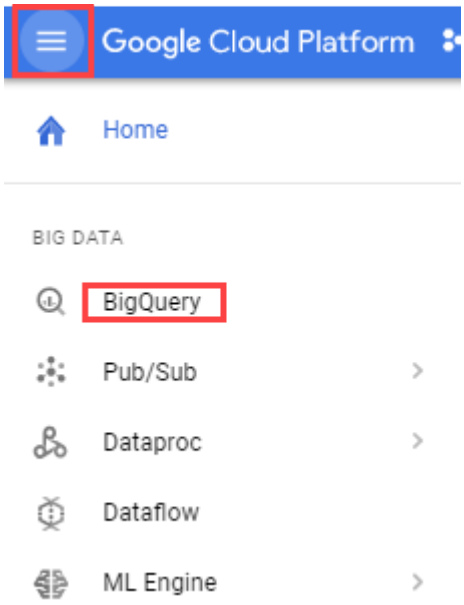
If you use other credentials, you'll get errors or **incur charges**.

7. Accept the terms and skip the recovery resource page.

Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

Open BigQuery Console

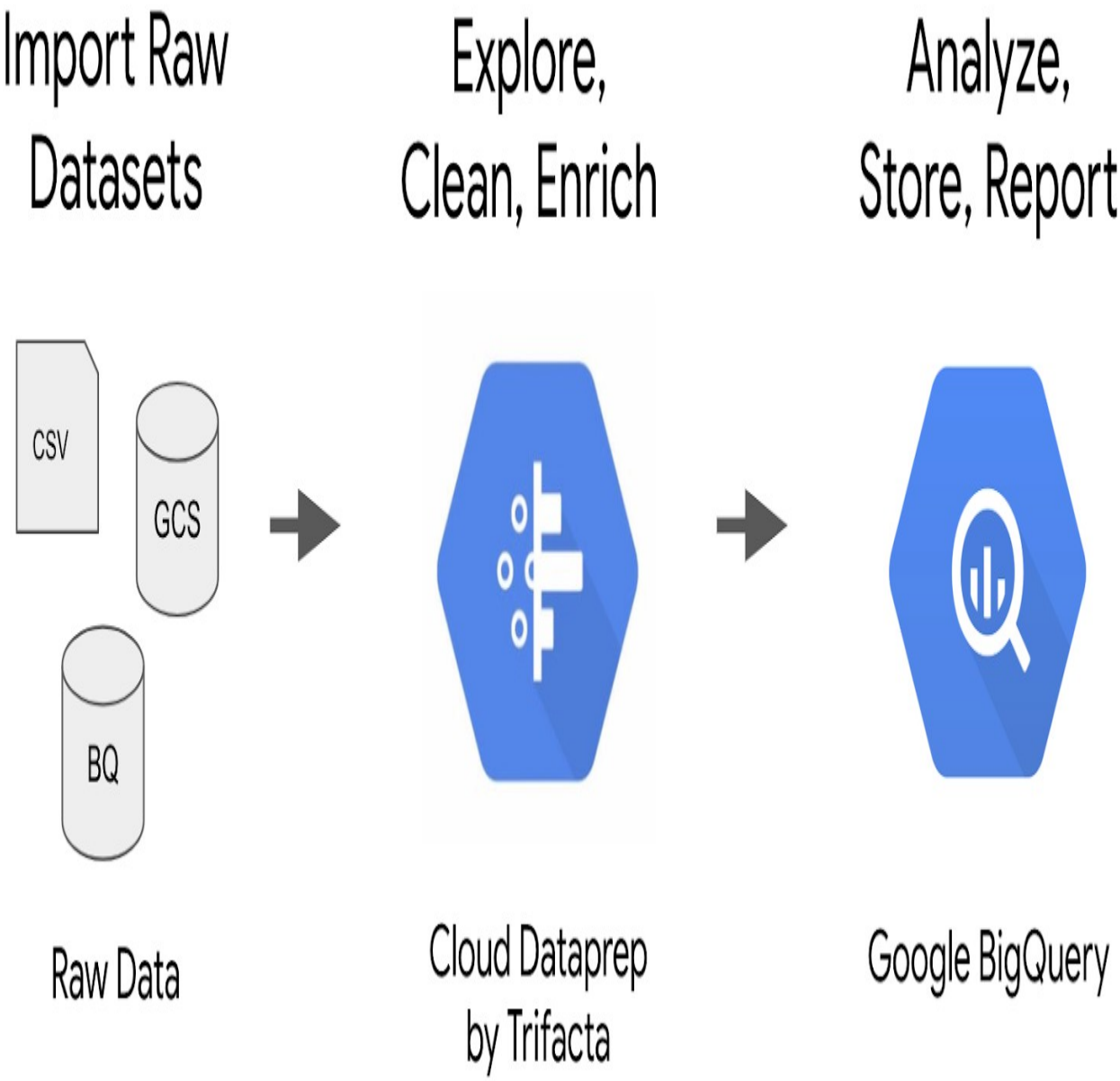
In the Google Cloud Console, select **Navigation menu** > **BigQuery**:



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

Click **Done**.

Although this lab is largely focused on Cloud Dataprep, you need BigQuery as an endpoint for dataset ingestion to the pipeline and as a destination for the output when the pipeline is completed.

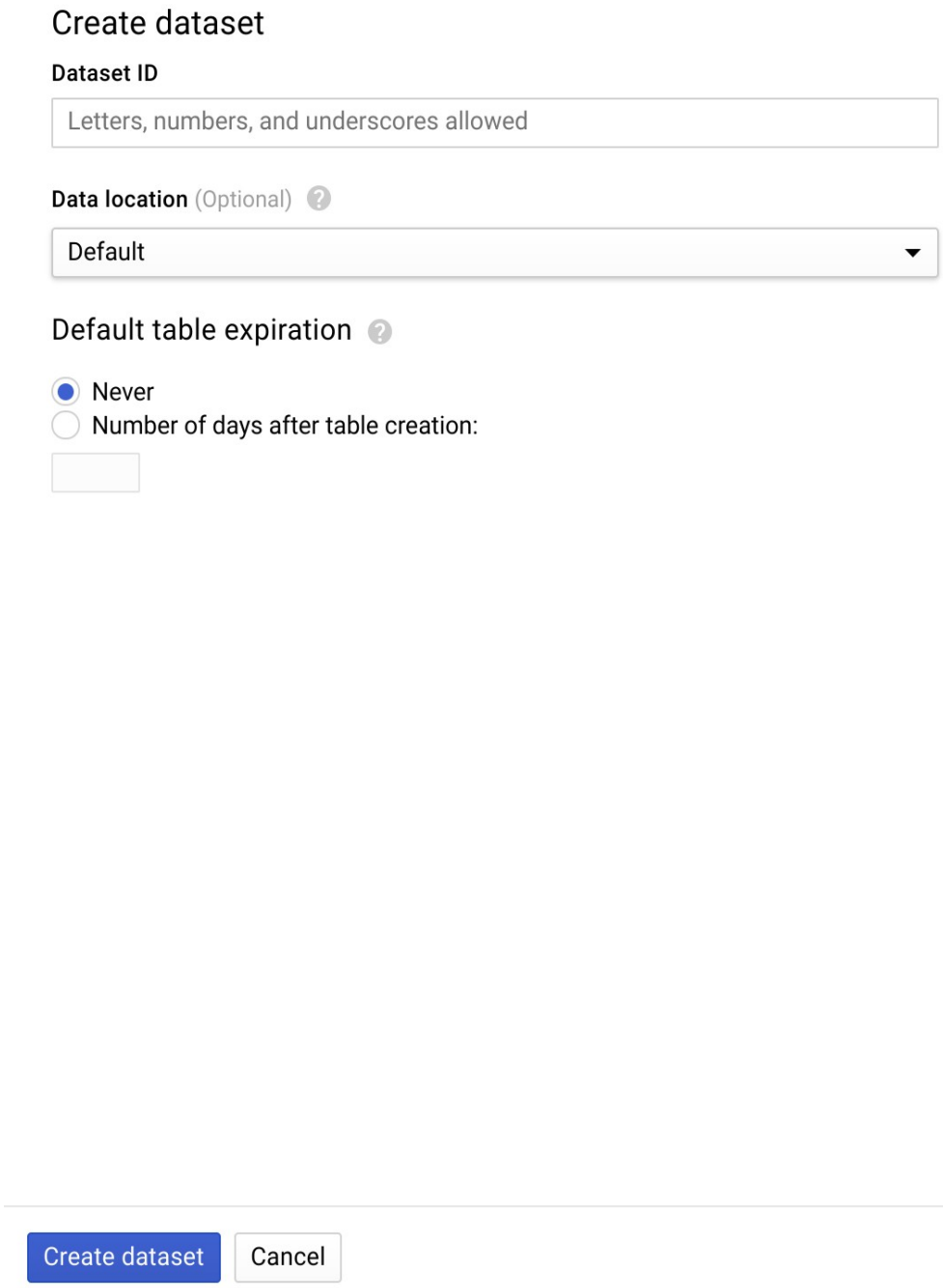


Task 2. Create an empty BigQuery dataset

In this task, you create a new BigQuery dataset to receive the output table of your new pipeline.

1. In the left pane, click on your project name, and then click **Create Dataset**.
2. In the **Create dataset** dialog:

- For **Dataset ID**, type **ecommerce**.
- Leave the other values at their defaults.



3. Click **Create dataset**.
4. Copy and paste this SQL query into the **Query editor** textfield:

```
#standardSQL
 CREATE OR REPLACE TABLE ecommerce.all_sessions_raw_dataprep
 OPTIONS(
   description="Raw data from analyst team to ingest into Cloud Dataprep"
 ) AS
 SELECT * FROM `data-to-insights.ecommerce.all_sessions_raw`
 WHERE date = '20170801'; # limiting to one day of data 56k rows for this lab
```

Click **Run**.

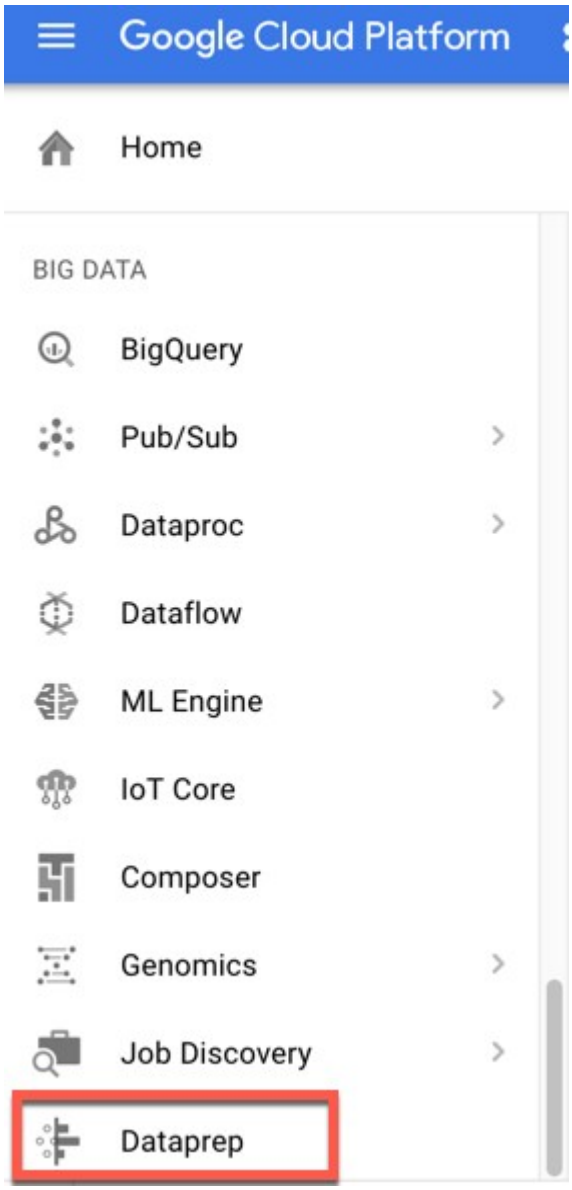
This query copies over a subset of the public raw ecommerce dataset to your own project dataset for you to explore and clean in Cloud Dataprep.

6. Confirm that the new raw data table exists in your project.

Task 3. Open Cloud Dataprep

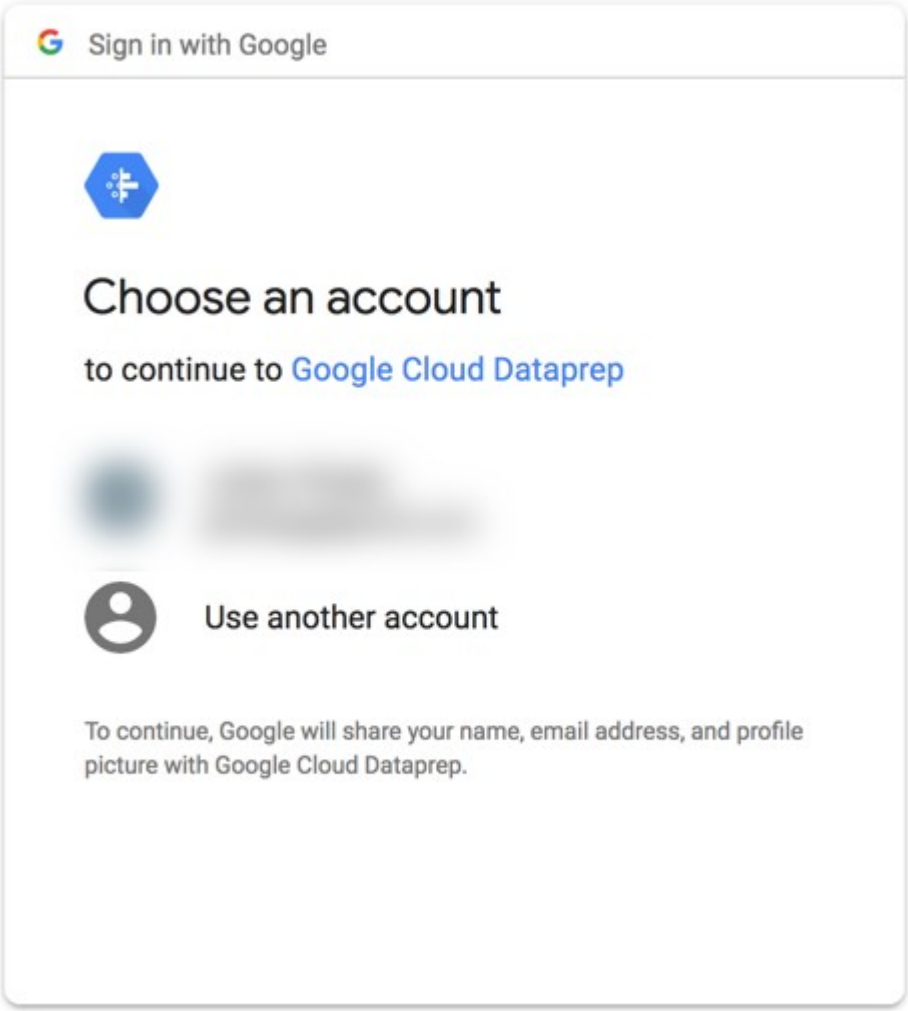
In this task, you accept the terms of service for Google and Trifacta, and then you allow Trifacta to access your project data.

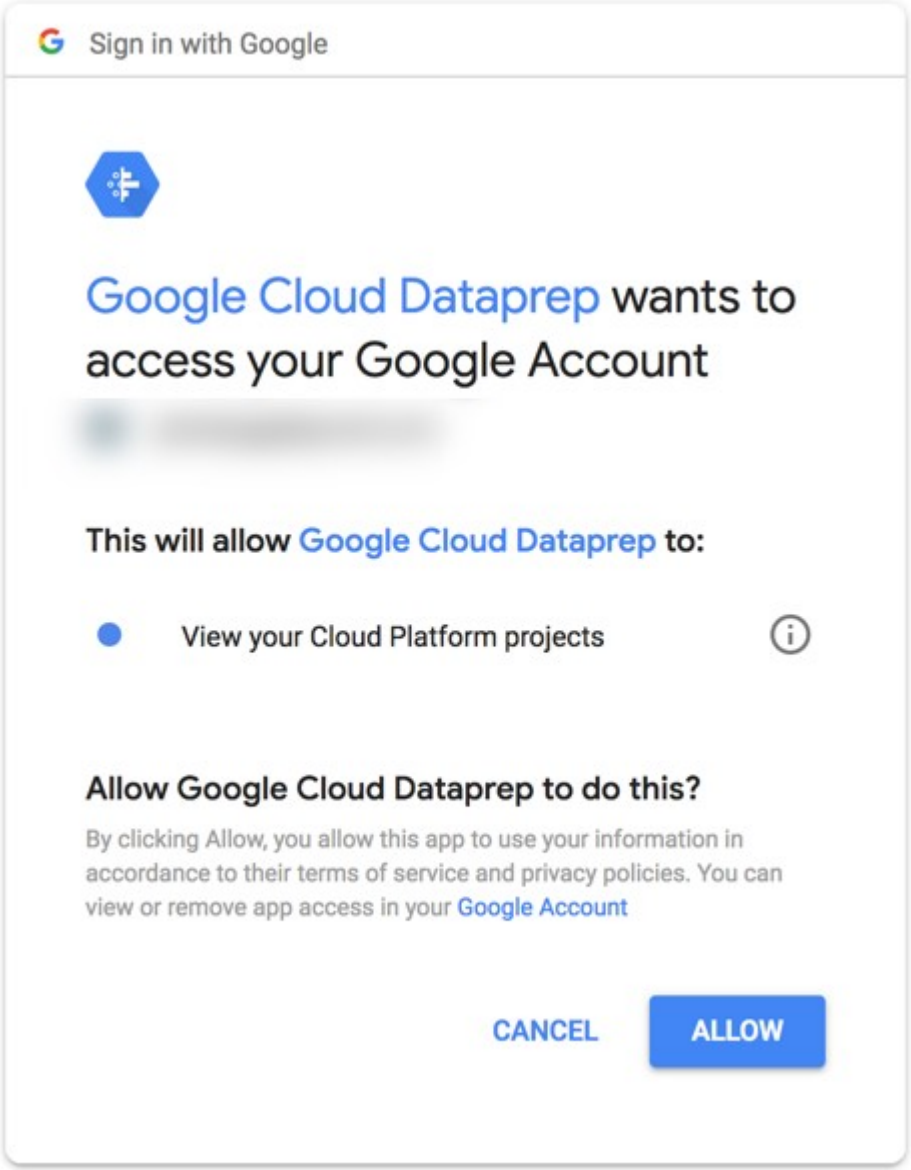
1. Go to the GCP console, and make sure that your lab's project is selected.



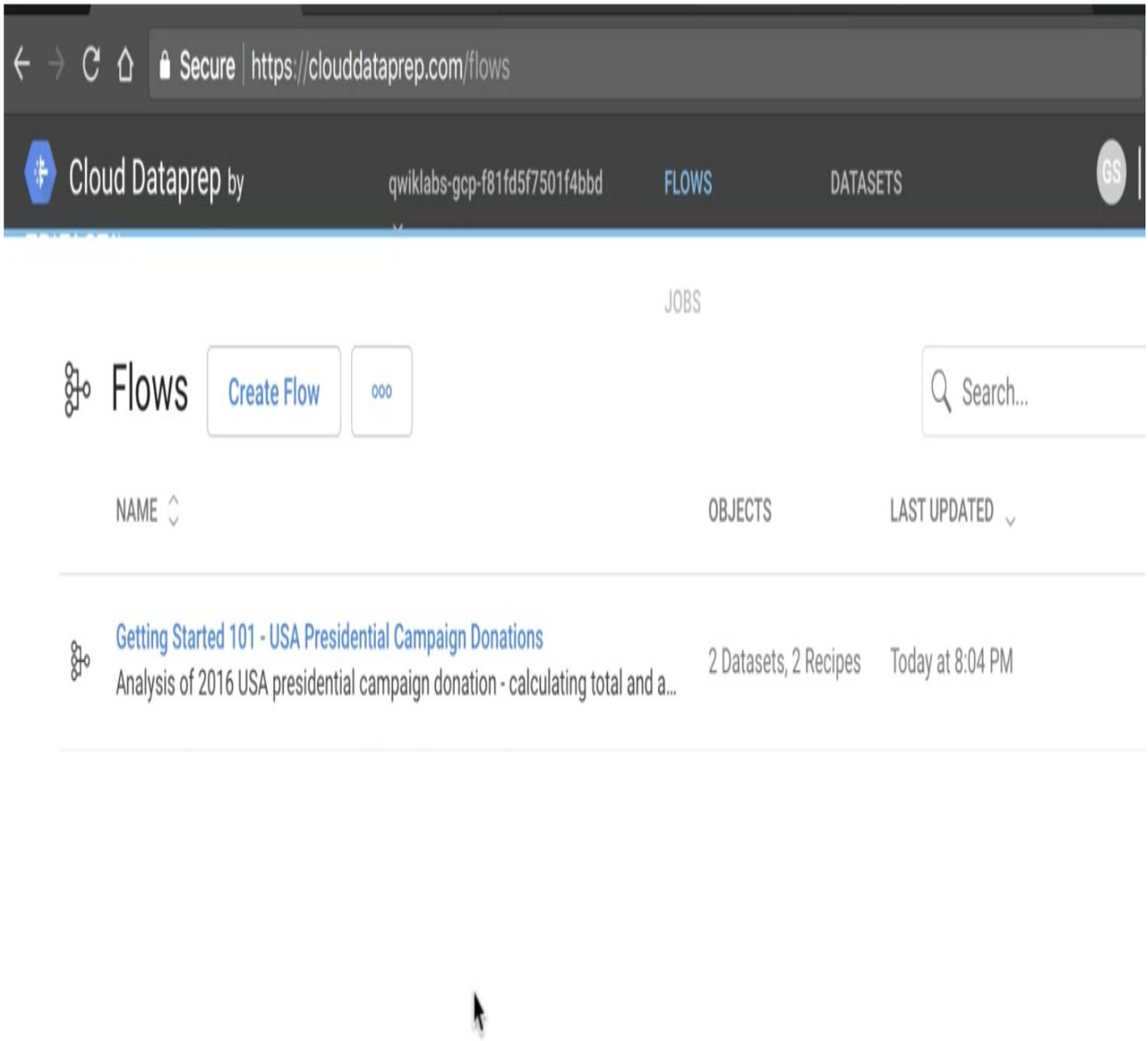
2. In the **Navigation menu** (☰), click **Dataprep**.
3. Select the Terms of Service for Google and Trifacta, and then click **Accept**.
4. In the **Share account information with Trifacta** dialog, select the checkbox, and then click **Agree and Continue**.
5. To **allow Trifacta to access your project data**, click **Allow**. This authorization process might take a few minutes.

6. In the **Sign in with Google** window appears, select your Qwiklab account and then click **Allow**. Click **Accept** if required after checking the checkbox.





7. To use the default location for the storage bucket, click **Continue**. The Flows homepage of Cloud Dataprep appears.



Task 4. Connect BigQuery data to Cloud Dataprep

In this task, you connect Cloud Dataprep to your BigQuery data source.

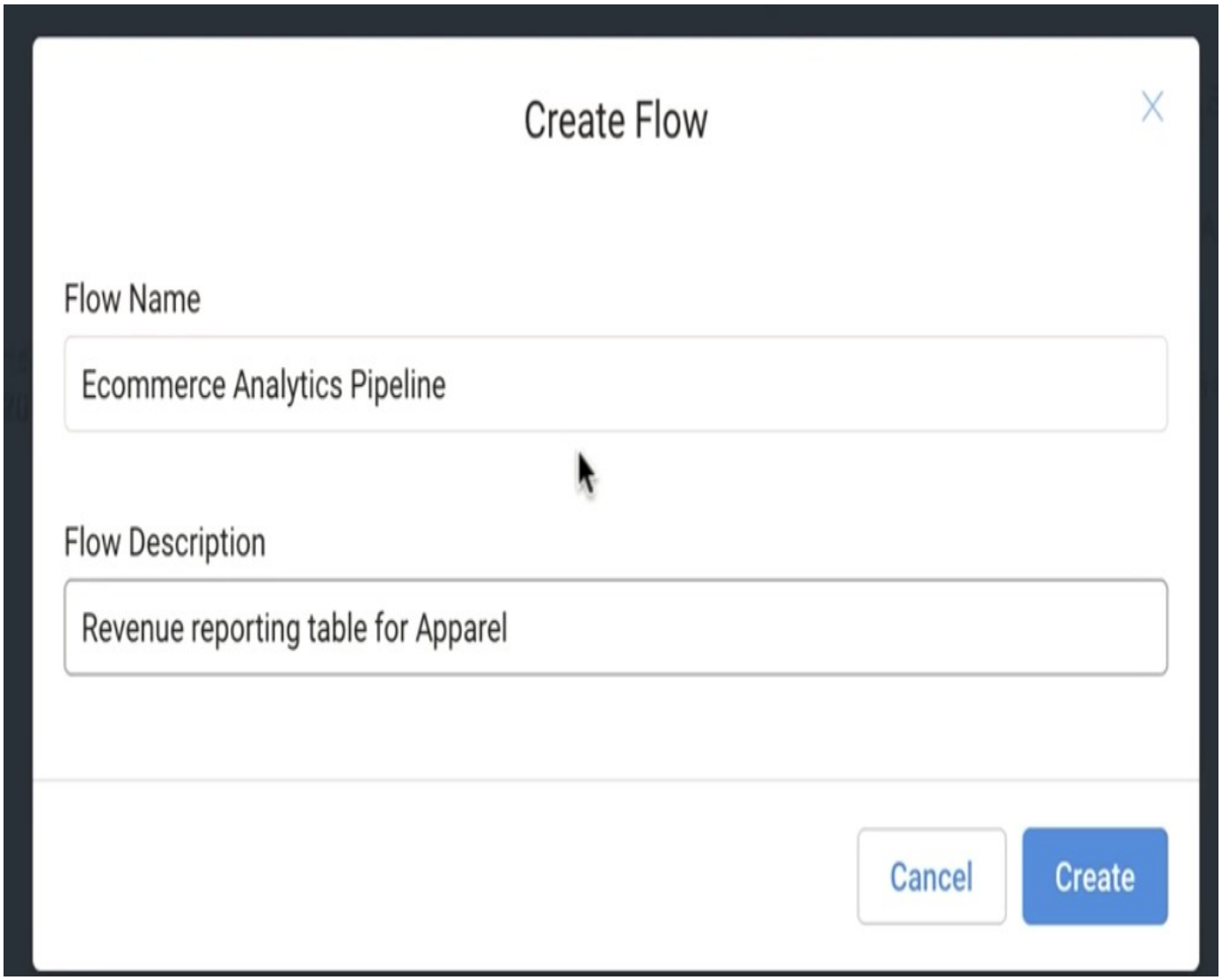
On the Cloud Dataprep page:

1. Click **Create Flow**.

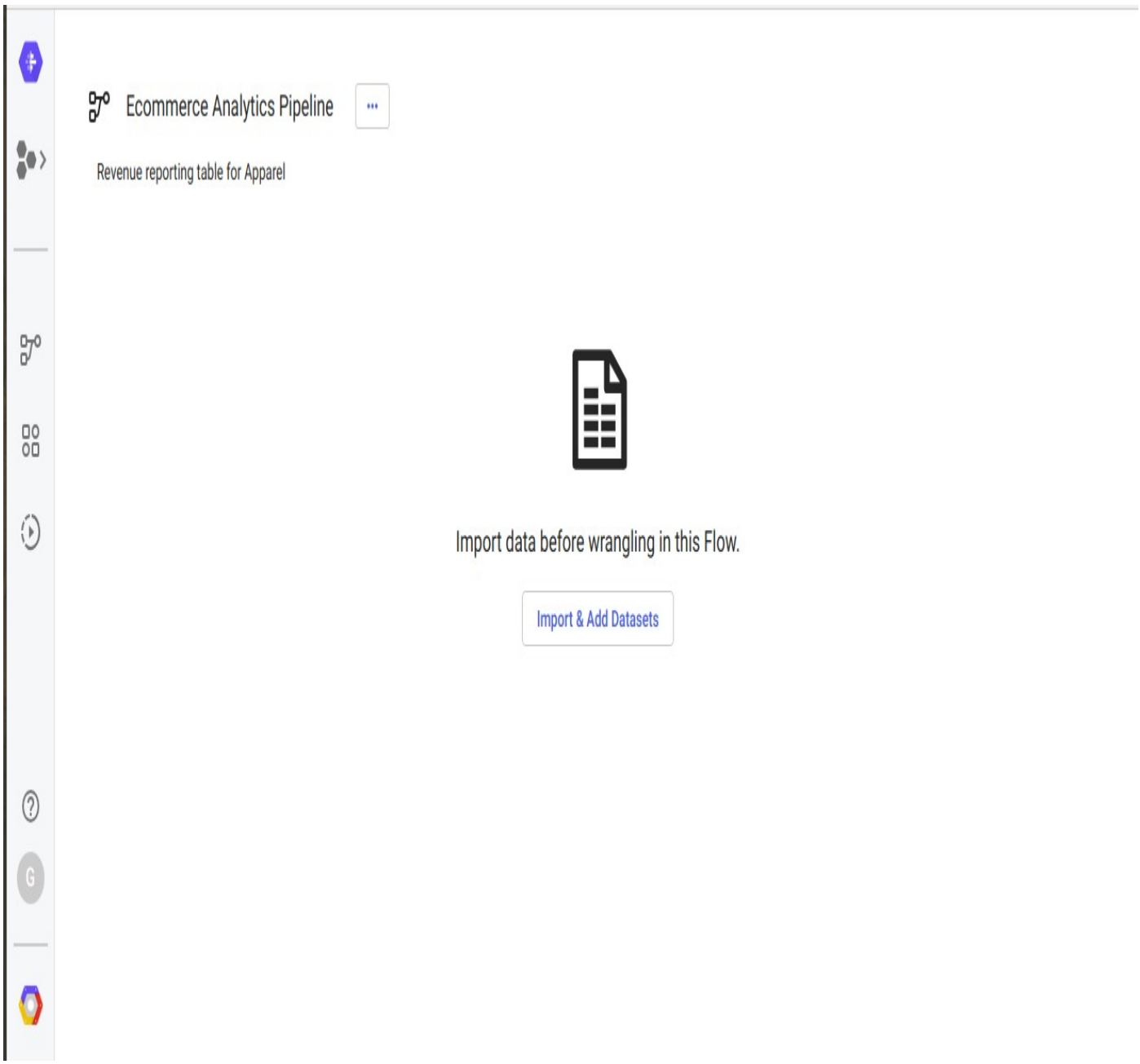
2. In the Create Flow dialog, specify these details:

- For **Flow Name**, type **Ecommerce Analytics Pipeline**
- For **Flow Description**, type **Revenue reporting table for Apparel**

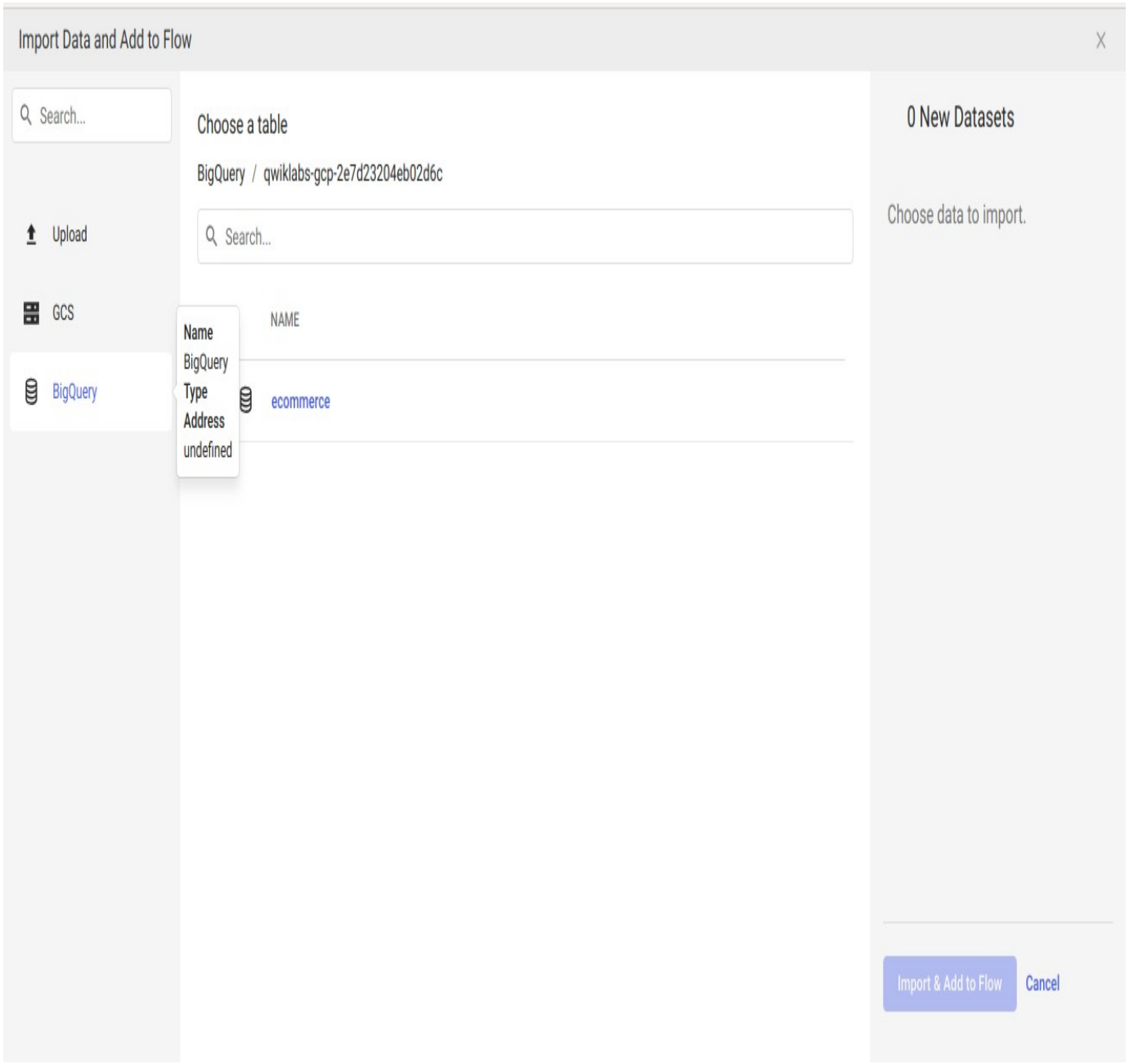
3. Click **Create**.



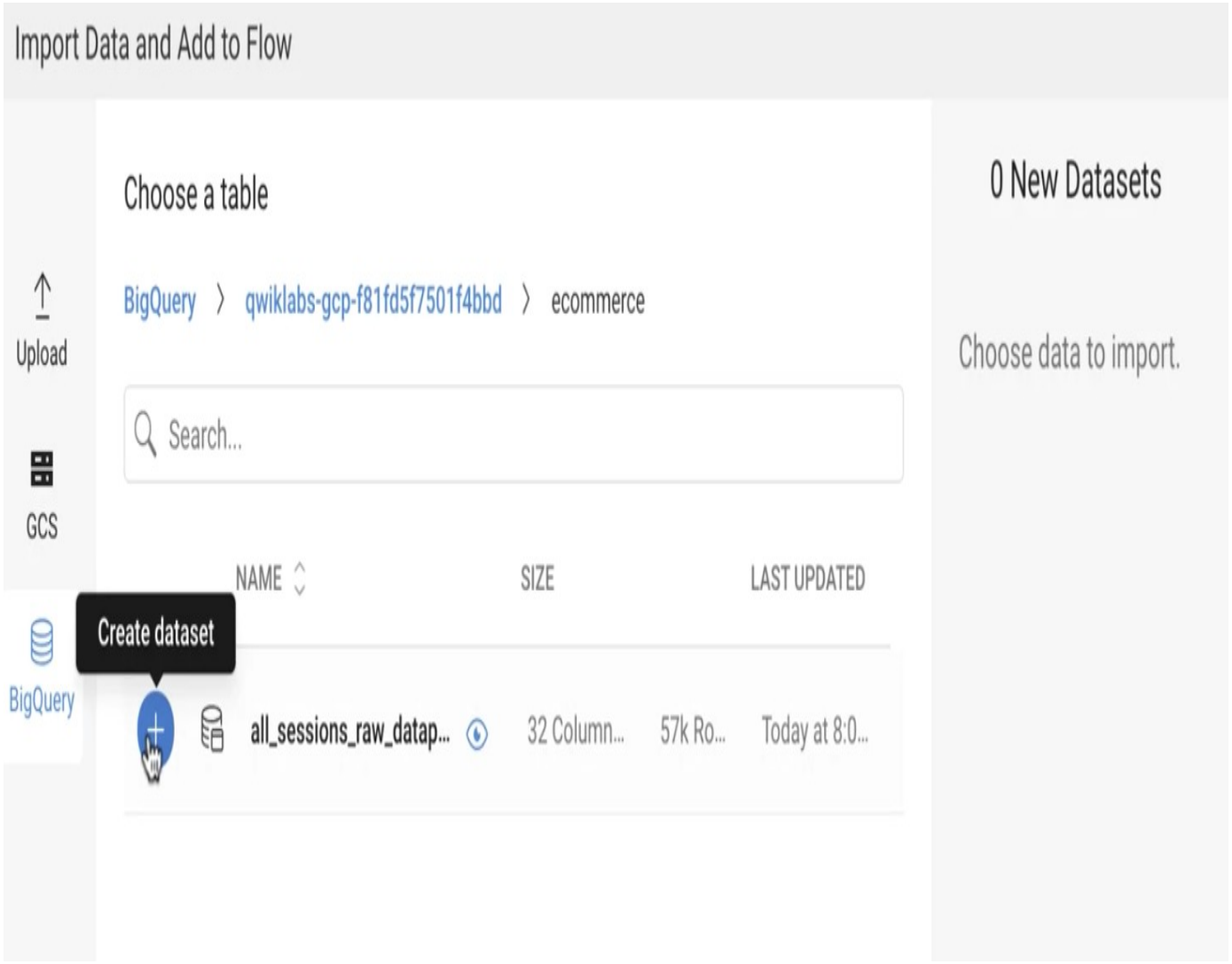
4. Click **Don't show me any helpers** in **What's a flow?** dialog.
5. Click **Import & Add Datasets**.



6. In the left pane, click **BigQuery**.
7. When your **ecommerce** dataset is loaded, click on it.

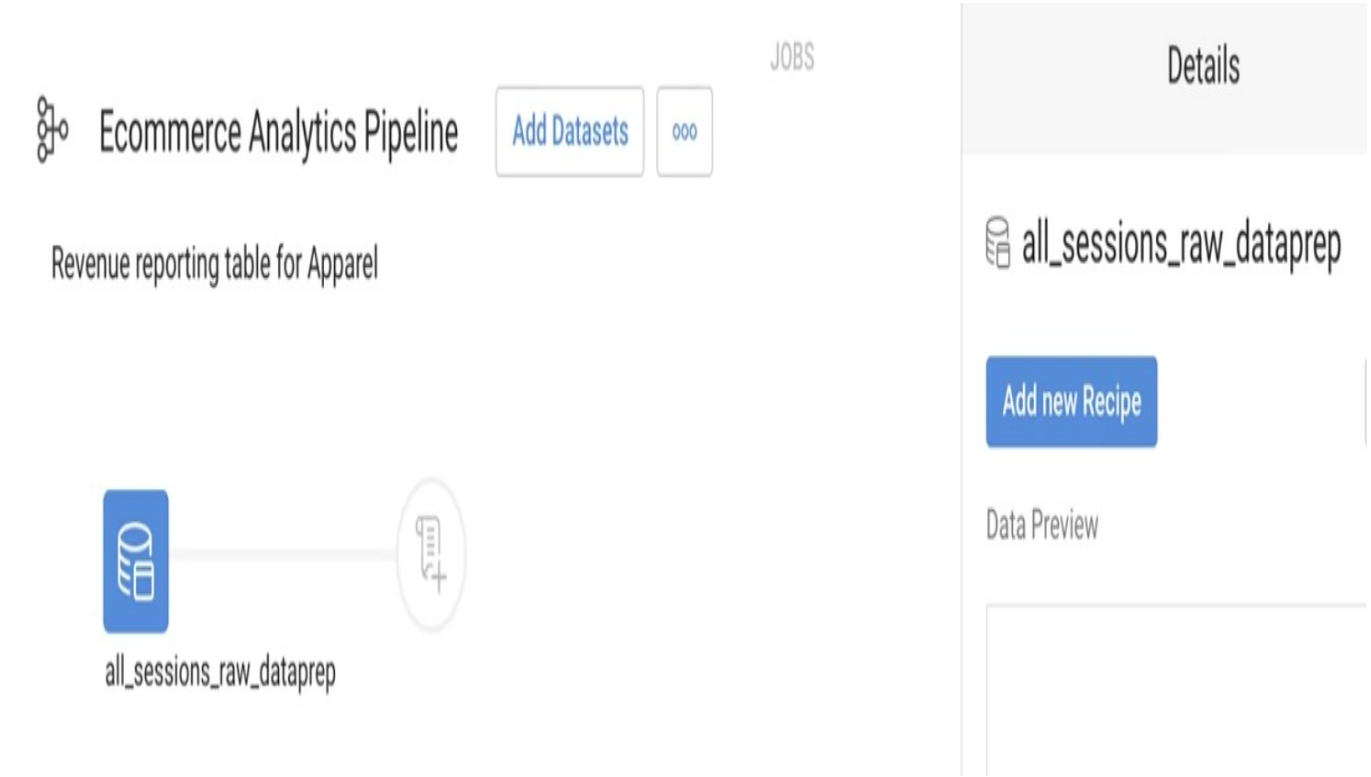


8. To create a dataset, click **Create dataset** icon (+).



9. Click **Import & Add to Flow**.

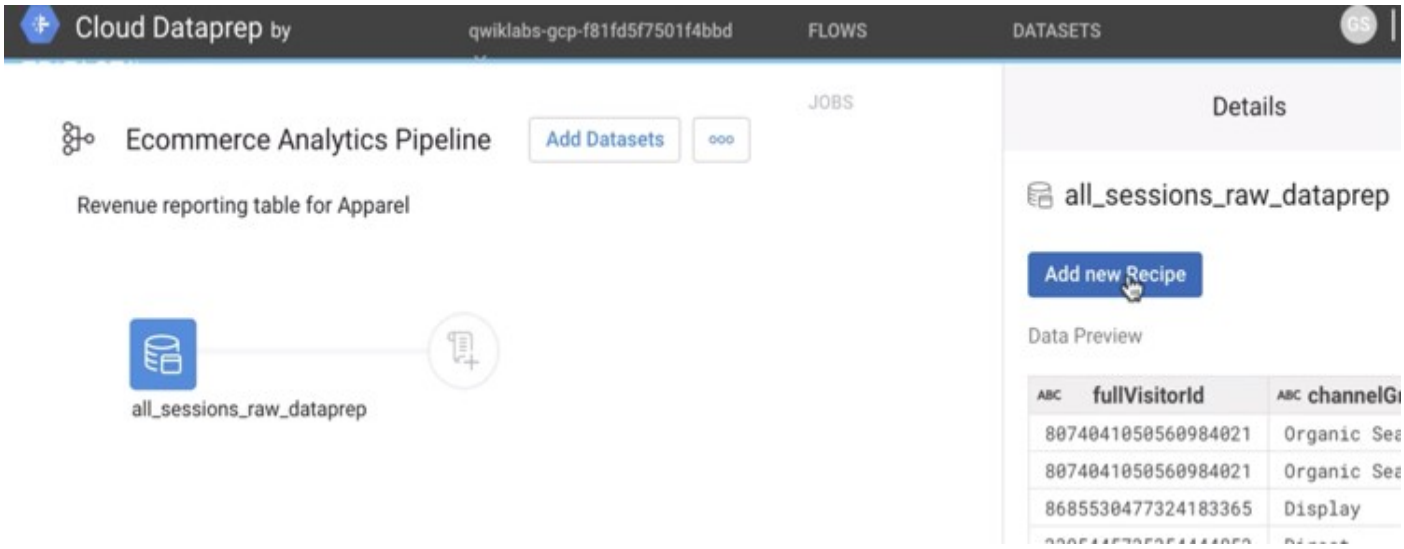
The data source automatically updates. In the right pane, **Add new Recipe** should become an available option. You are ready to go to the next task.



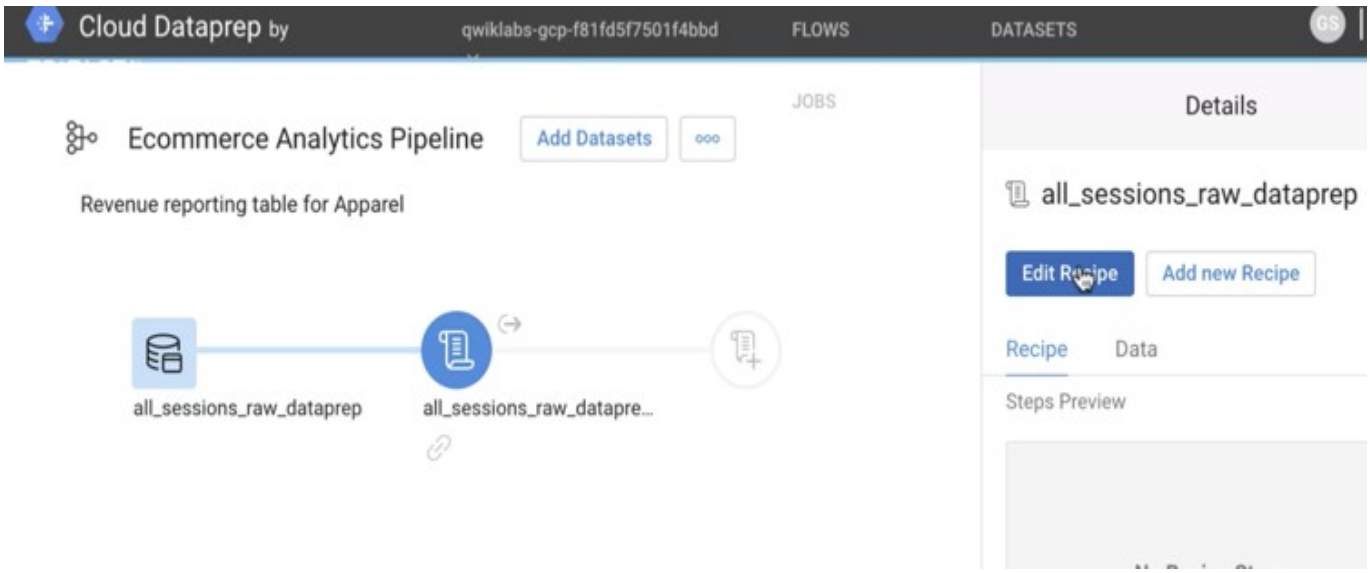
Task 5. Explore ecommerce data fields with a UI

In this task, you load and explore a sample of the dataset within Cloud Dataprep.

1. In the right pane, click **Add new Recipe**.

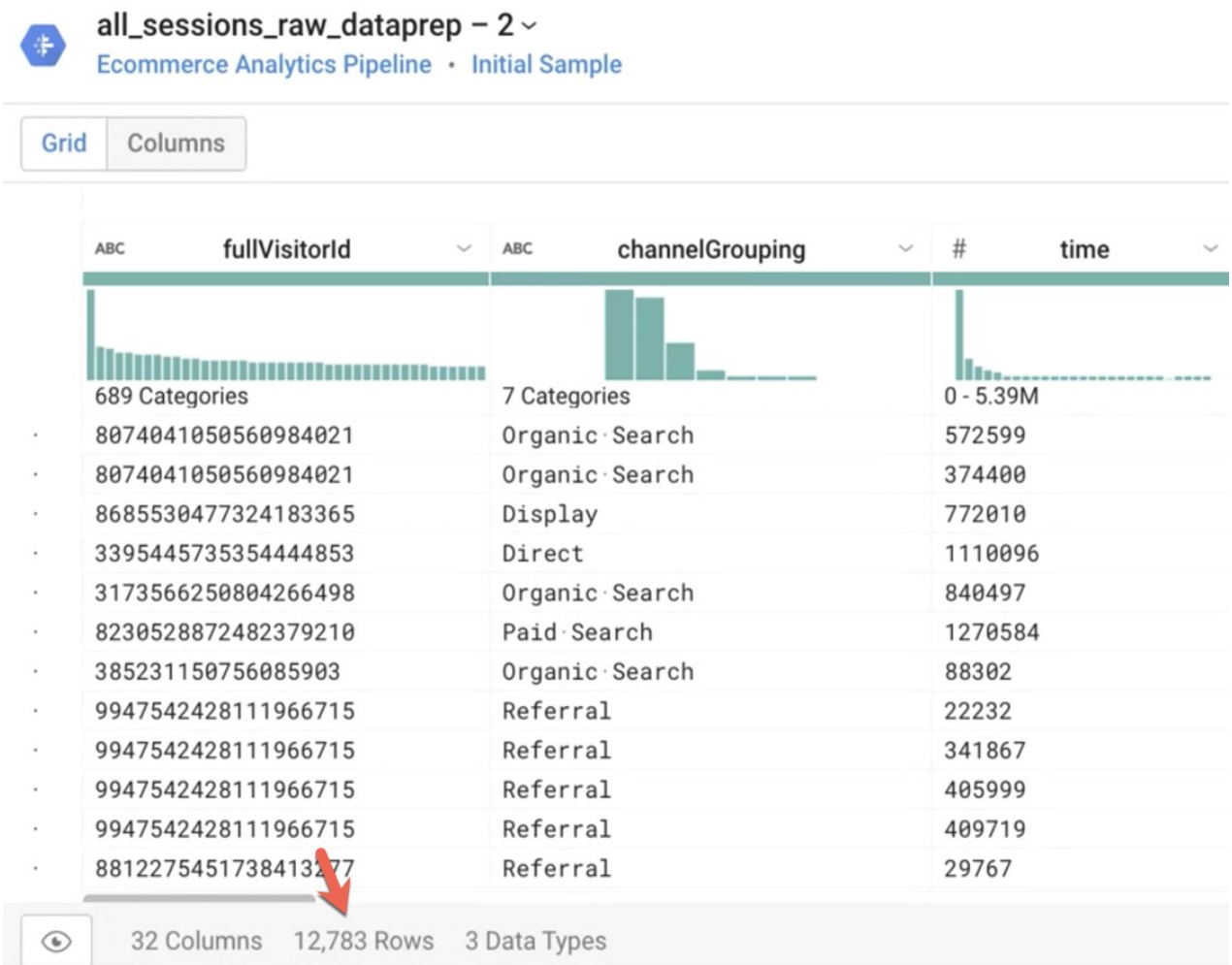


2. Click **Edit Recipe**.

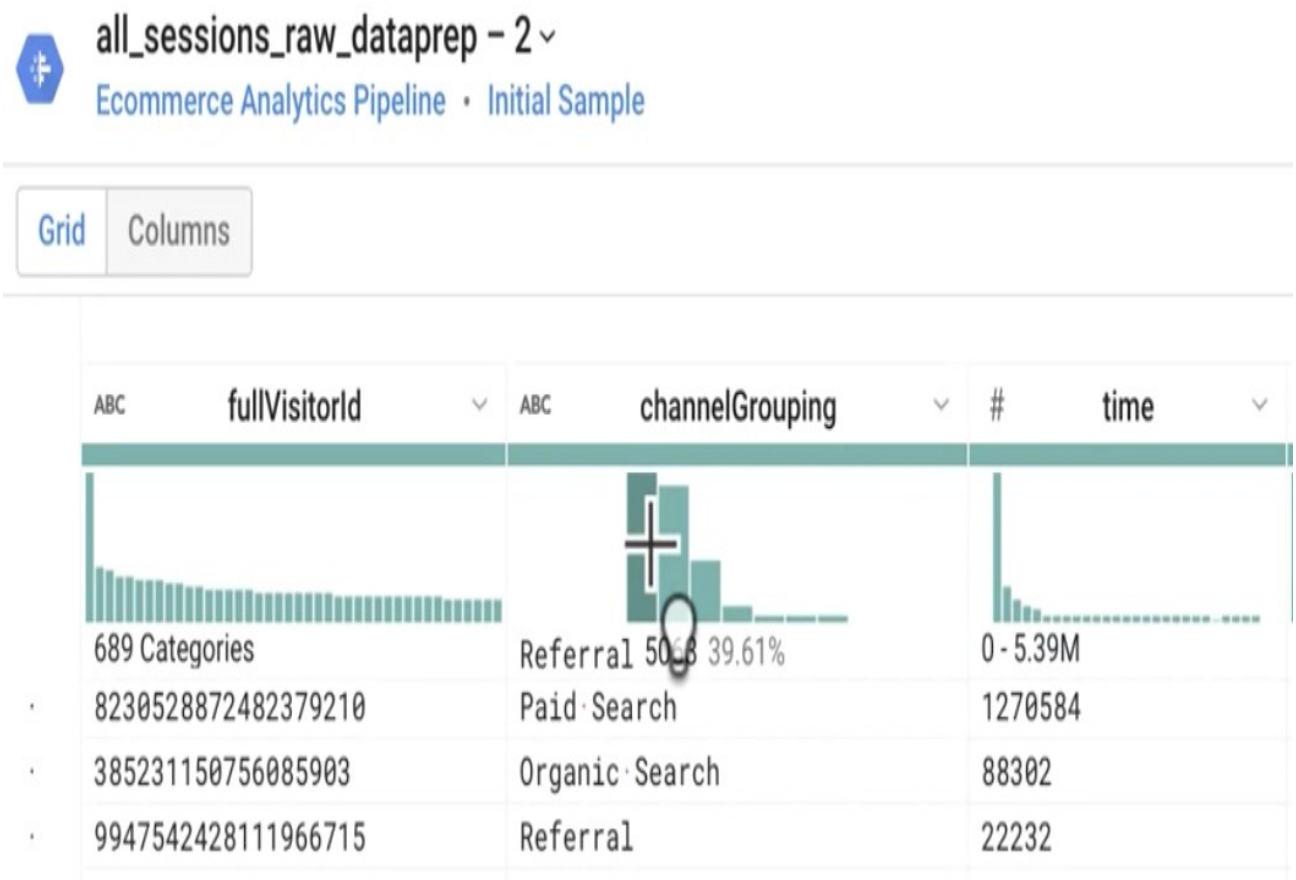


3. Click **Don't show me any helpers** in **The Transformer** dialog if required.

- Cloud Dataprep will load a sample of the source dataset for speed of exploration. **Note**: When your pipeline is run, it will operate over the entire source dataset. How many rows does the sample contain?

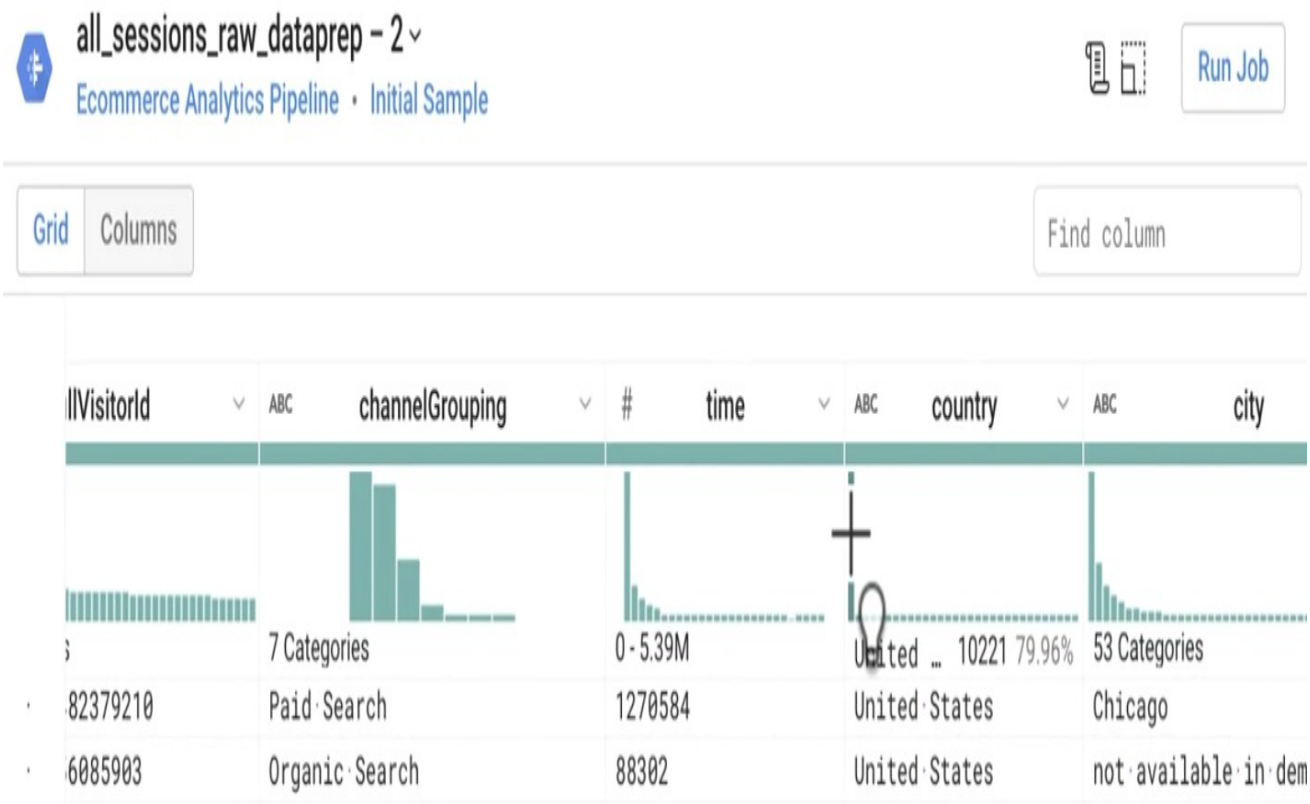


- `Answer:` About 12 thousands rows
- What is the most common value in the **channelGrouping** column?



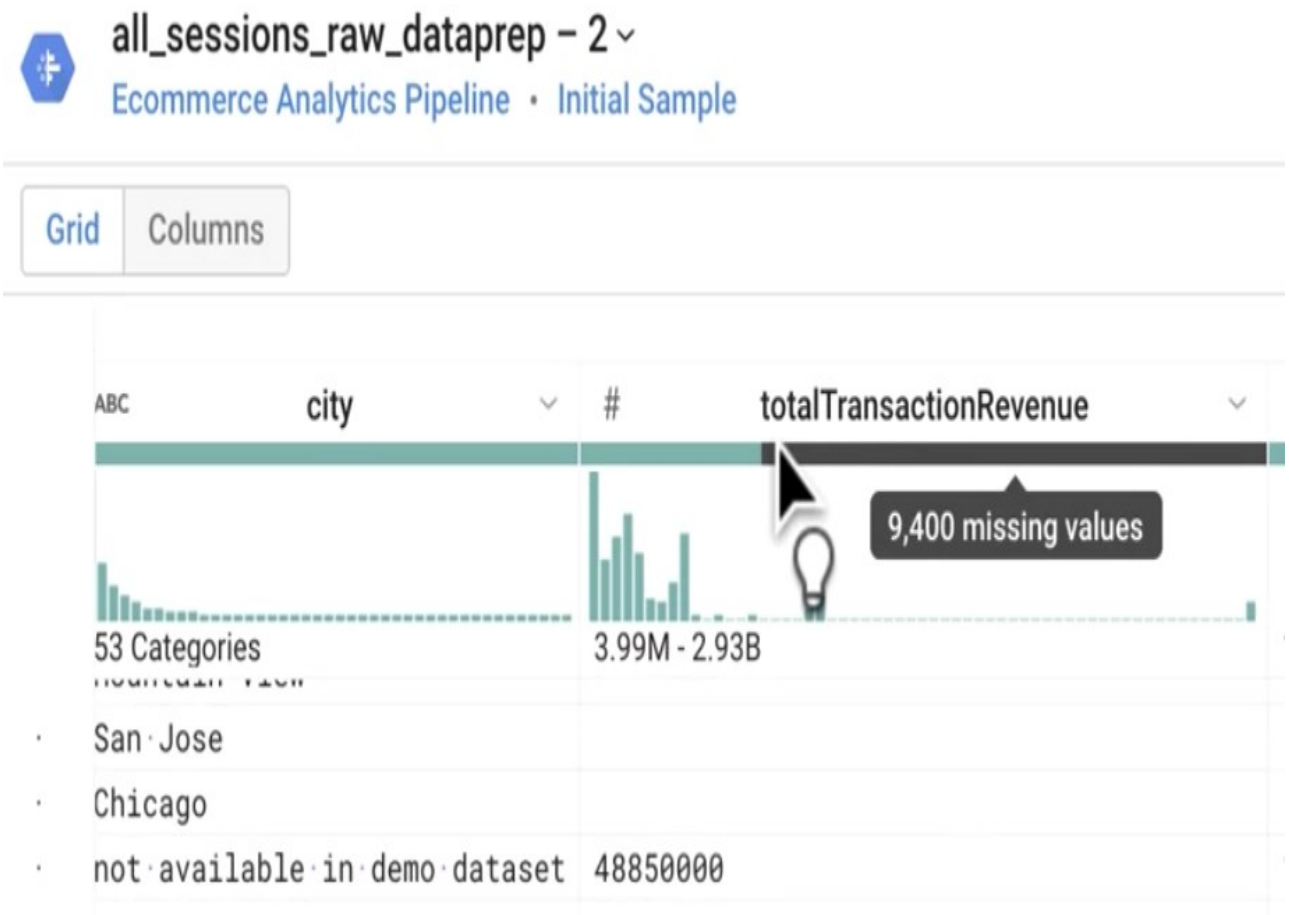
`Answer:` Referral

- What are the top three countries from which sessions are originated?



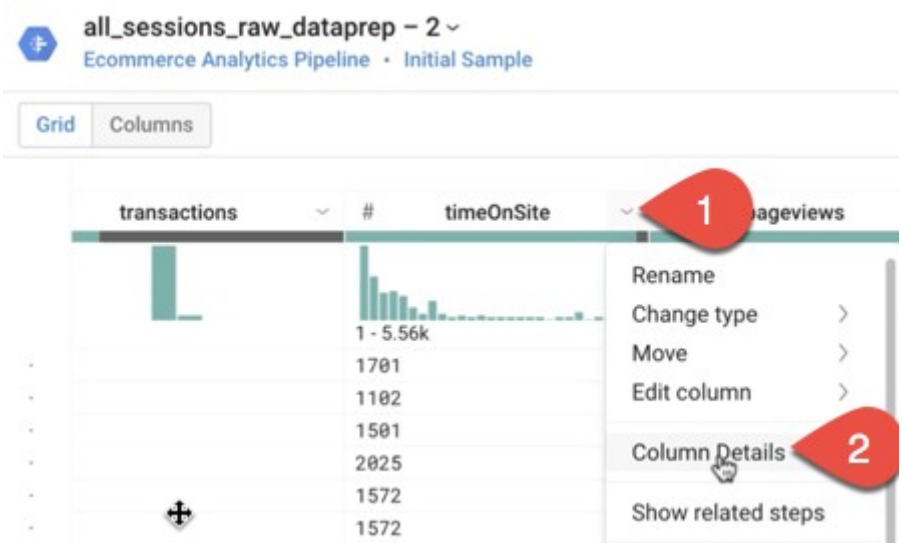
Answer: US, India, United Kingdom

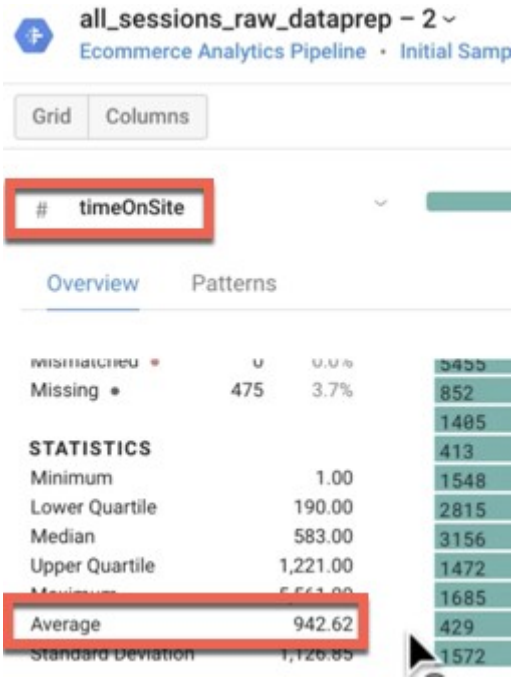
- What does the grey bar under totalTransactionRevenue represent?



`Answer:` Missing values

- What is the average **timeOnSite** in seconds, average **pageviews**, and average **sessionQualityDim** for the data sample? (Hint: Use **Column Details**.)





`Answers:` **Average Time On Site:** 942 seconds (or 15.7 minutes)

Average Pageviews: 20.44 pages

Average Session Quality Dimension: 38.36

Note: Your answers may vary slightly due to the data sample used by Cloud Dataprep

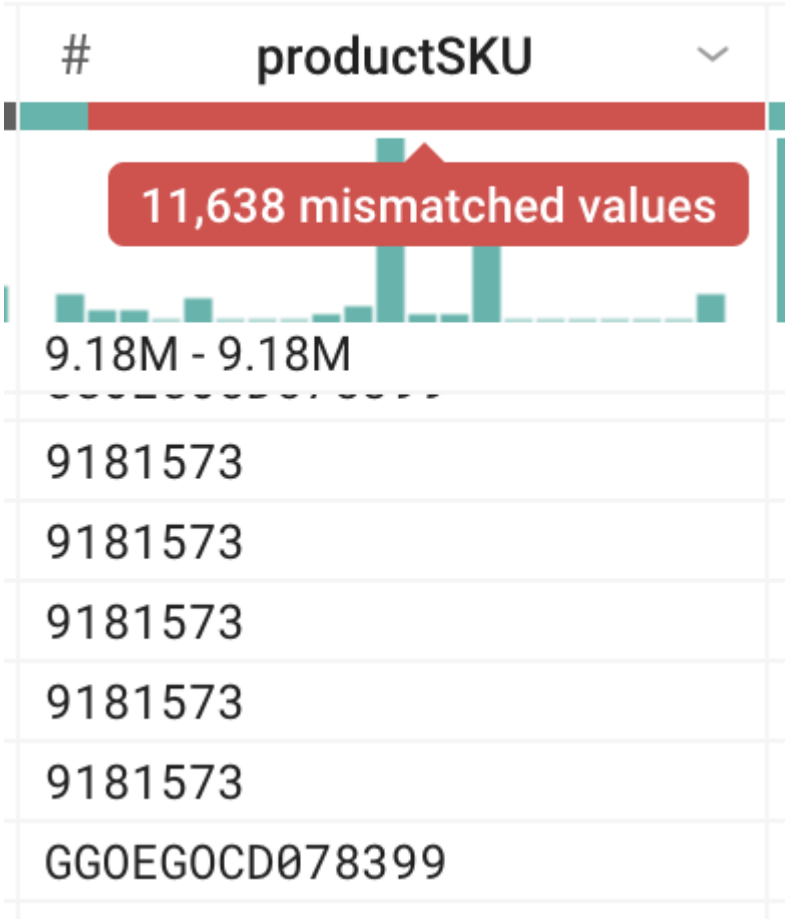
- Looking at the histogram for **sessionQualityDim**, are the data values evenly distributed?

`Answer:` No, they are skewed to lower values (low quality sessions), which is expected.

- What is the date range for the dataset sample?

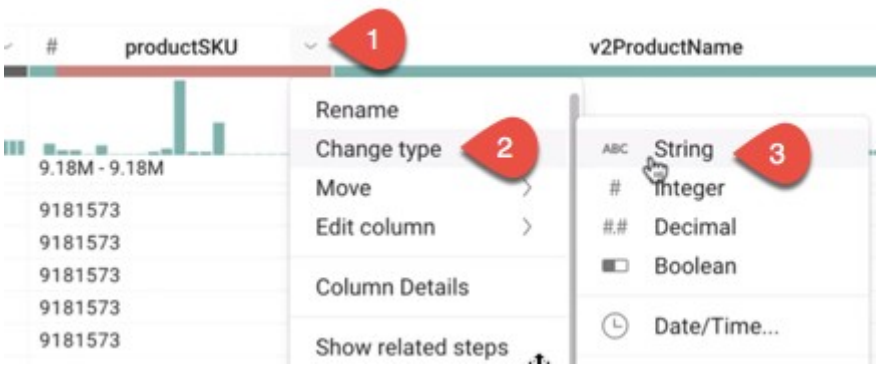
`Answer:` 8/1/2017 (one day of data)

- Why is there a red bar under the **productSKU** column?

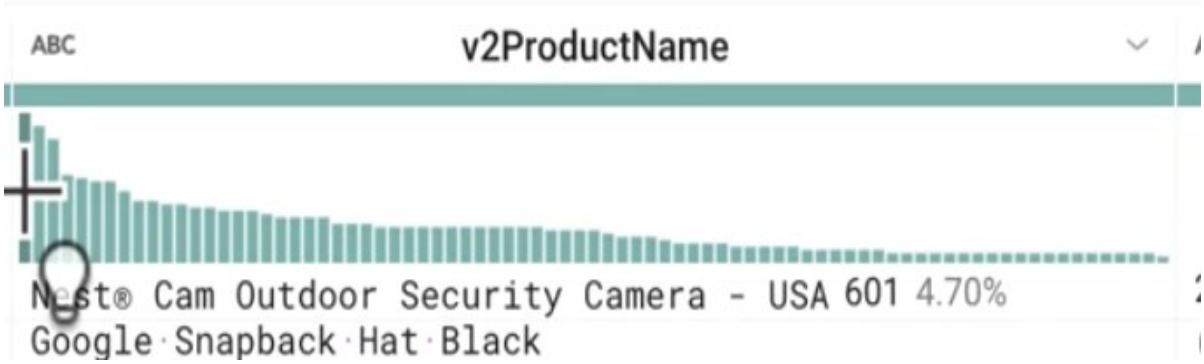


`Answer:` The red bar indicates mismatched values. Cloud Dataprep automatically identified the **productSKU** column type as an integer. Cloud Dataprep also detected some non-integer values and therefore flagged those as mismatched. In fact, the productSKU is not always an integer (for example, a correct value might be “GGOEGOCD078399”). So in this case, Cloud Dataprep incorrectly identified the column type: it should be a string, not an integer. You fix that in the next step.

- To convert the **productSKU** column type to a string data type, open the menu to the right of the **productSKU** column by clicking , then click **Change type > String**.

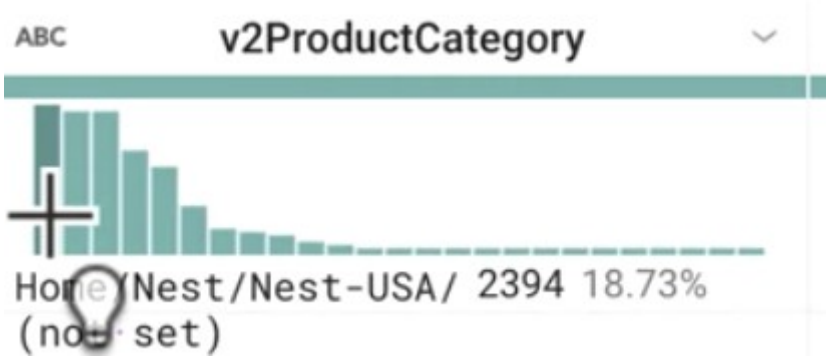


- Looking at **v2ProductName**, what are the most popular products?



`Answer:` Nest products

- Looking at **v2ProductCategory**, what are some of the most popular products? How many categories were sampled?



`Answer:` Nest, (not set), and Apparel are the most popular out of approximately 25 categories.

- True or False: The most common **productVariant** is COLOR.

`Answer:` False. It's (not set) because most products do not have variants (80%+)

- What are the two categories of type?

`Answer:` PAGE and EVENT

- What is the average **productQuantity**?

`Answer:` 3.45 (your answer may vary)

- How many distinct SKUs are in the dataset?

Answer: Over 600+

- What are some of the most popular product names by row count? The most popular categories?

Answer:

Cam Outdoor Security Camera - USA

Cam Indoor Security Camera - USA

Learning Thermostat 3rd Gen-USA - Stainless Steel

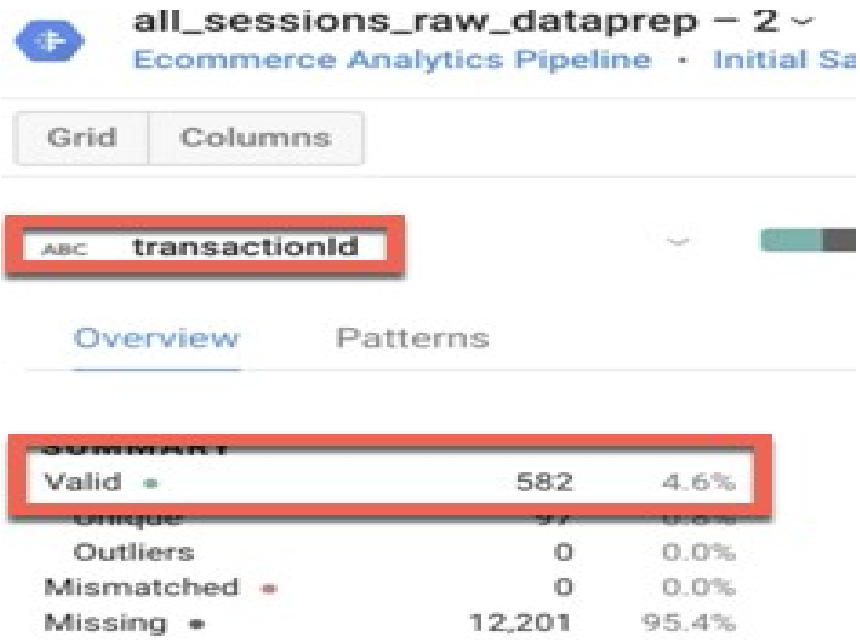
- What is the dominant currency code for transactions?

Answer: **USD** (United States Dollar)

- Are there valid values for **itemQuantity** or item **itemRevenue**?

Answer: No, they are all NULL values.

- What percentage of transaction IDs have a valid value? What does this represent for our ecommerce dataset?



Answer: About 4.6% of transaction IDs have a valid value, which represents the average conversion rate of the website (4.6% of visitors transact).

- How many **eCommerceAction_type** are there, and what is the most popular **eCommerceAction_step**?

Answers:

Six types have data in our sample.

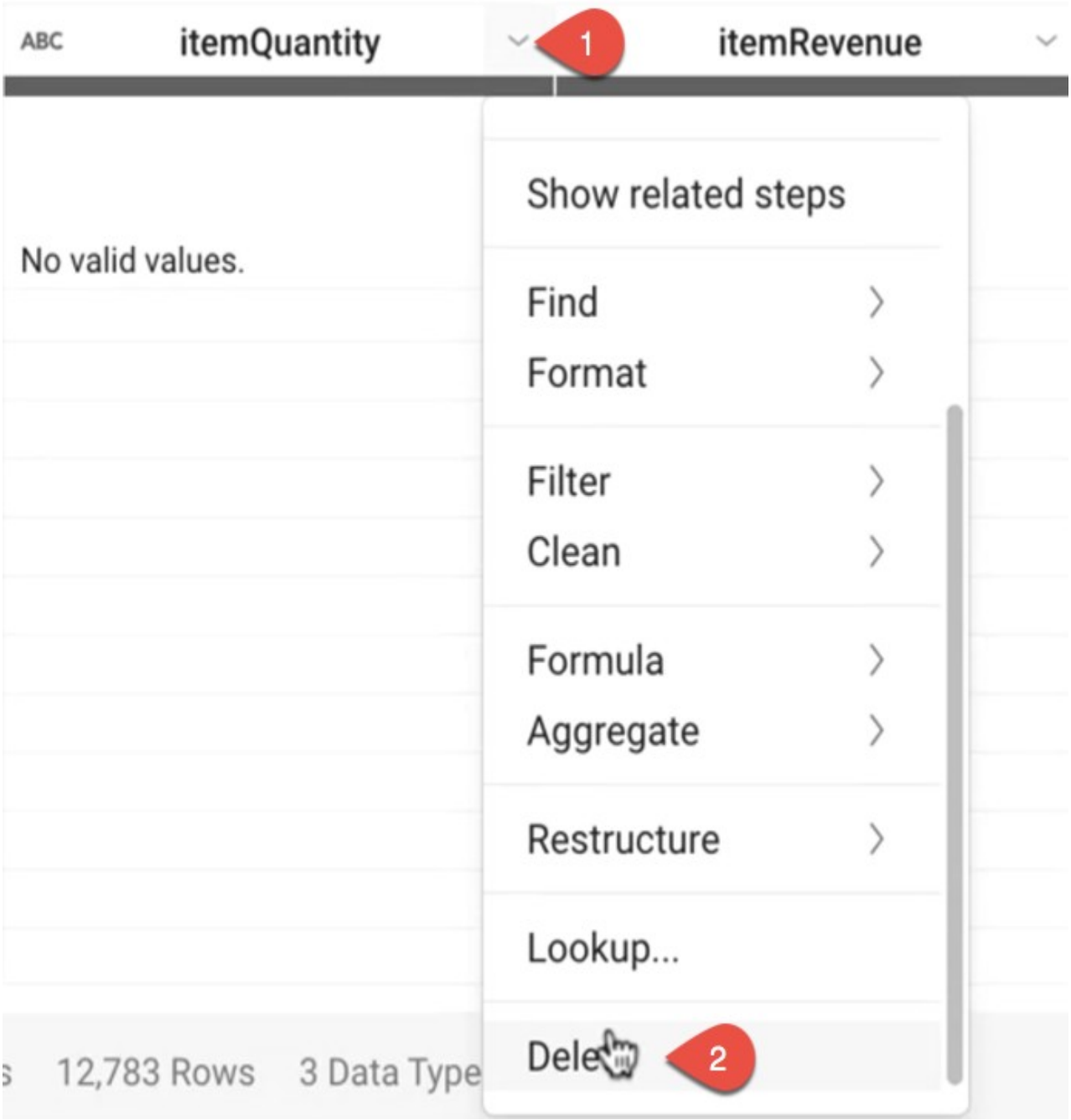
0 or NULL is the most popular.

Task 6. Clean the data

In this task, you clean the data by deleting unused columns, eliminating duplicates, creating calculated fields, and filtering the rows. Deleting columns is common for when fields are depreciated in the schema or have all NULL values.

Delete unused columns

- Select the unwanted column, and then click **Delete**. Do this for the following columns which have all NULL values:



- itemRevenue

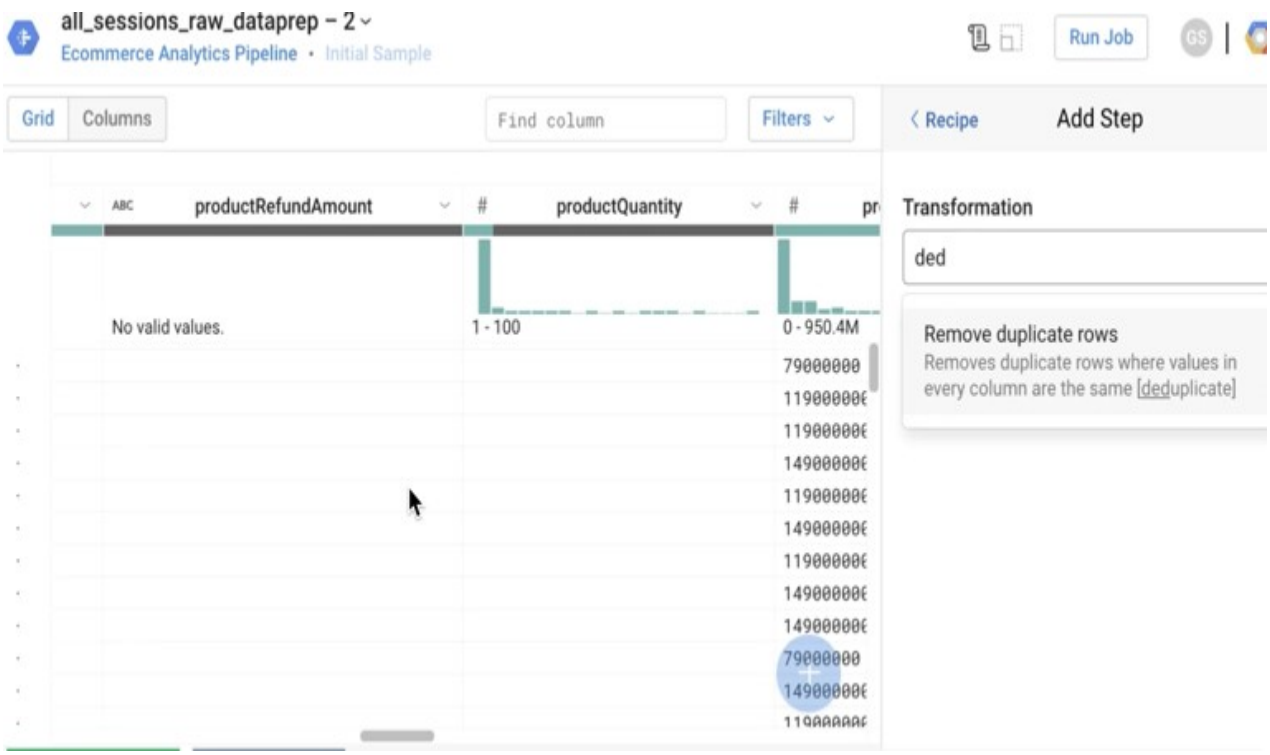
- itemQuantity

Deduplicate rows

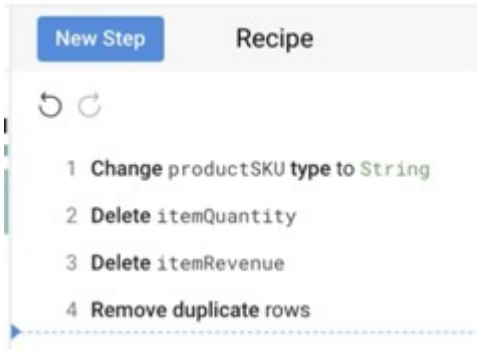
Your team has informed you there may be duplicate session values included in the source dataset. Let’s remove these with a new deduplicate step.

1. Click **Recipe** icon and select **New Step**.

2. In the Transformation box, type **deduplicate** and select **Remove duplicate rows**.
3. Click **Add**.



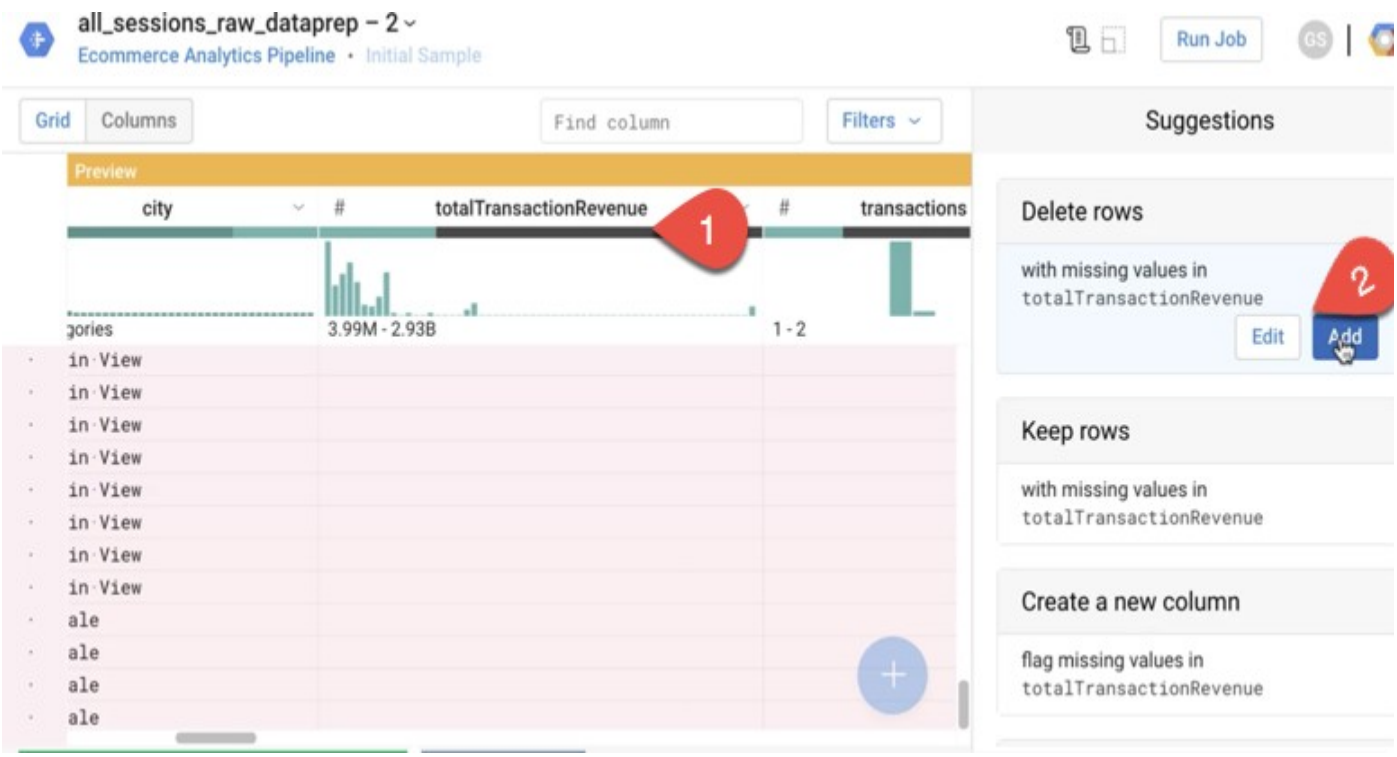
4. Review the recipe created so far:



Filter out sessions without revenue

Your team has asked you to create a table of all user sessions that bought at least one item from the website. Filter out user sessions with NULL revenue.

1. Under the **totalTransactionRevenue** column, click the missing values bar.
2. In the **Suggestions** panel, click **Delete rows** with missing values, and then click **Add** (as shown).



This step filters your dataset to only include transactions with revenue (where **totalTransactionRevenue** is NULL).

Filter out sessions for just Type = ‘PAGE’

The dataset contains both views of website Pages and triggered Events like “viewed product categories” or “added to cart”. To avoid double counting session pageviews, add a filter to only include pageview-related events.

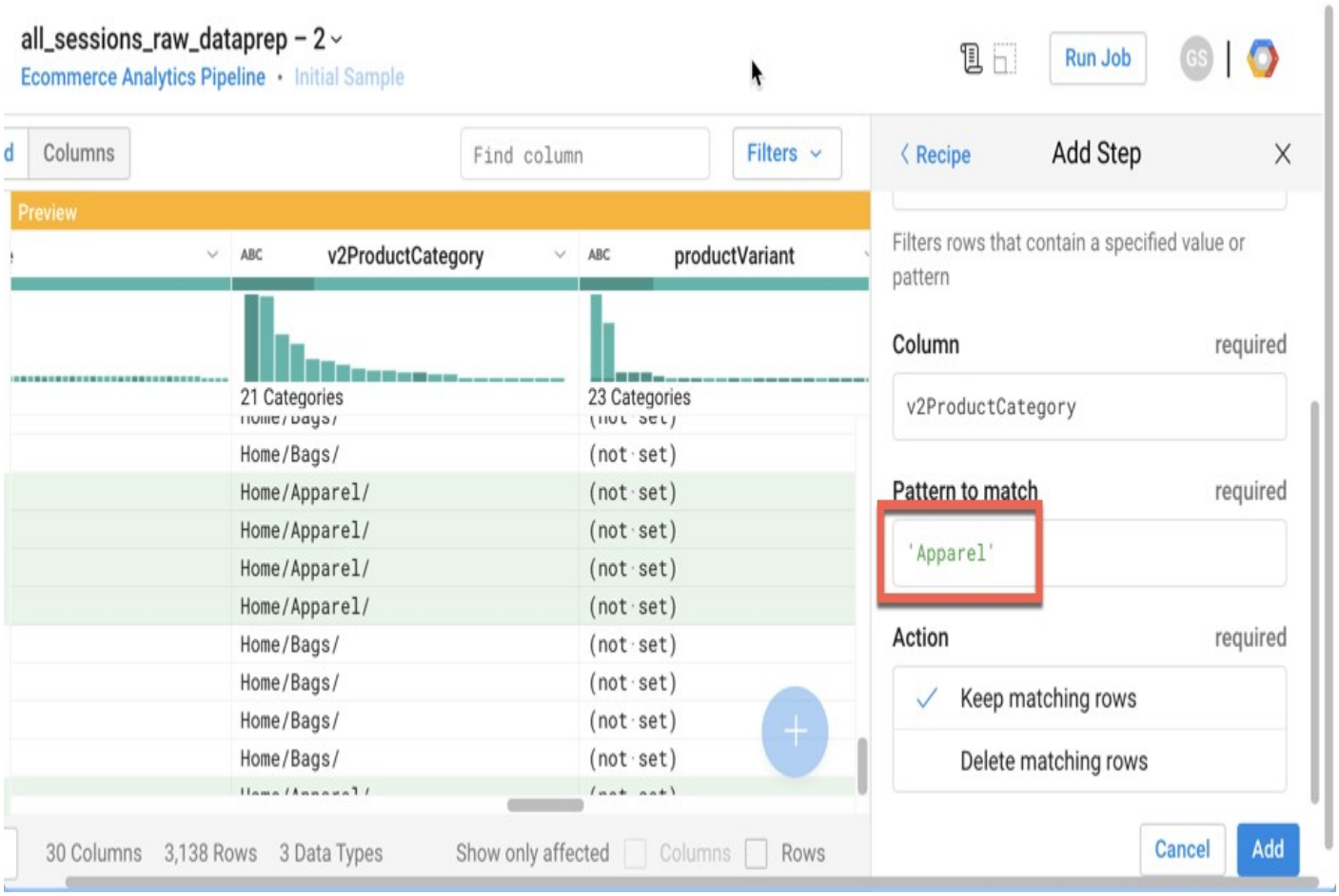
1. In the **type** column, click the bar for PAGE.
2. In the **Suggestions** panel, click Keep rows where type is PAGE, and then click **Add**.

Filter for apparel products

Your team has now asked you to further filter your output to only include transactions in the Apparel category (apparel includes items like T-Shirts and other clothing items)

1. Next to the **v2ProductCategory** column, click the drop down icon.
2. Select **Filter rows** > **On column values**.
3. Select **Contains**.
4. In **Pattern to match** type 'Apparel' (case sensitive) and then click **Add**.

`Note:` Products in the catalog can belong to more than one category (‘Apparel’ and ‘Home/Apparel/’) which is why we are matching any rows that have Apparel anywhere in the category name.



Task 7. Enrich the data

Search your schema documentation for **visitId** and read the description to determine if it is unique across all user sessions or just the user.

VisitId = An identifier for this session. This is part of the value usually stored as the _utmb cookie. This is only unique to the user. For a completely unique ID, you should use a combination of fullVisitorId and visitId.

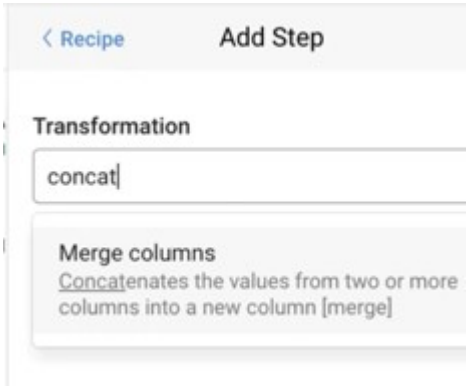
visitId is not unique across all users.

In this task, you add a new concatenated column to create a unique session ID field. Then you will enrich your ecommerce label data with a case statement.

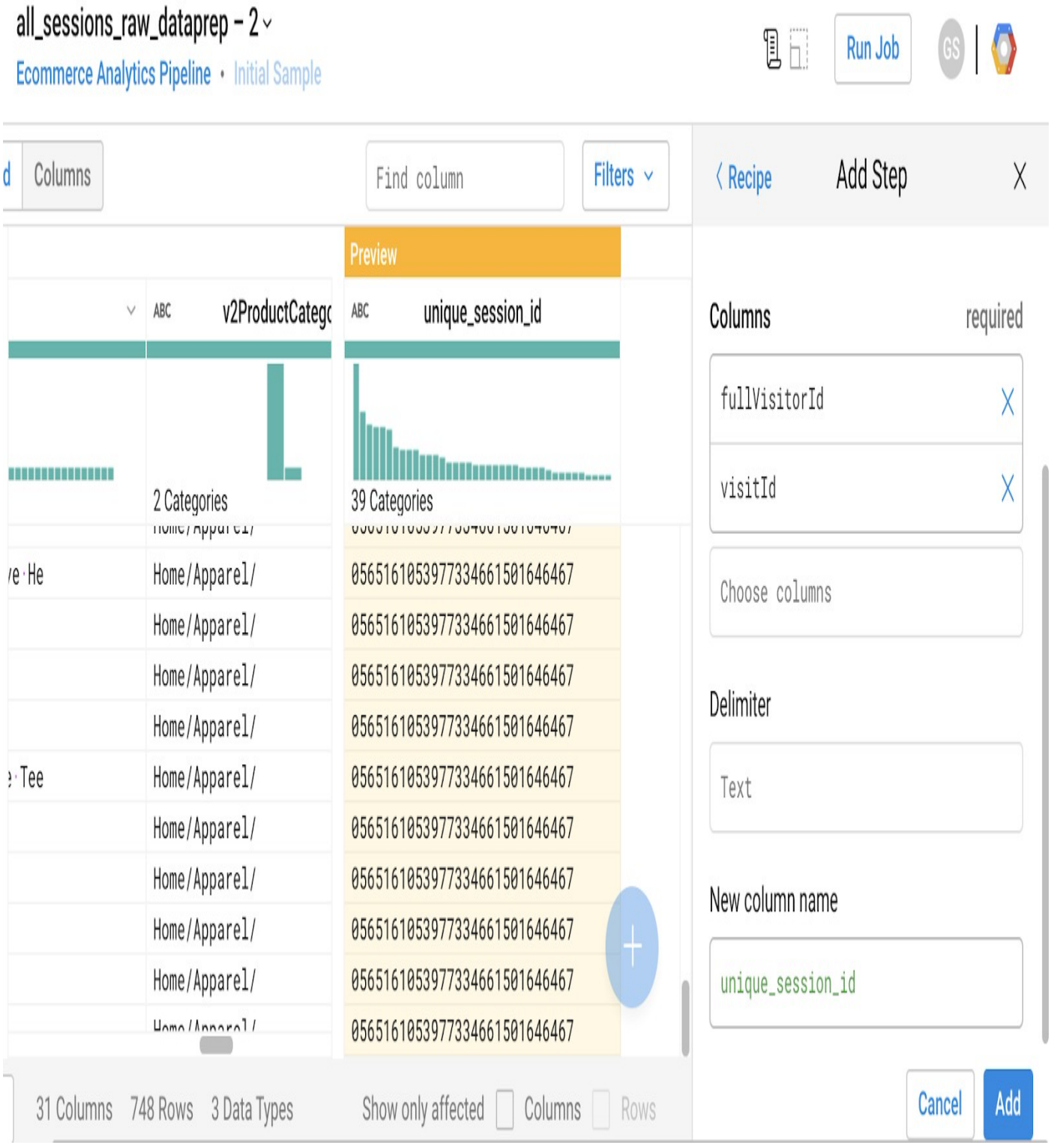
Create a new column for a unique session ID

As you discovered, the dataset has no single column for a unique visitor session. Create a unique ID for each session by concatenating the **fullVisitorID** and **visitId** fields.

1. Click on **New Step**.
2. For **Search transformation**, type **concat**, and then select **Merge columns**.



3. For **Columns**, select **fullVisitorId** and **visitId**.
4. For the **New column name**, type **unique_session_id**, and leave the other inputs as their default values and click **Add**.

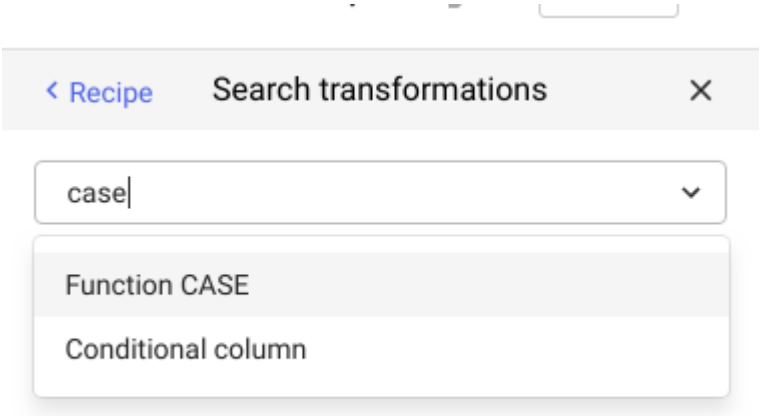


Create a case statement for the ecommerce action type

The **eCommerceAction_type** field is an integer that maps to actual ecommerce actions performed in that session like 3 = “Add to Cart” or 5 = “Check out.” Create a calculated column that maps to the integer value.

1. Click on **New Step**.

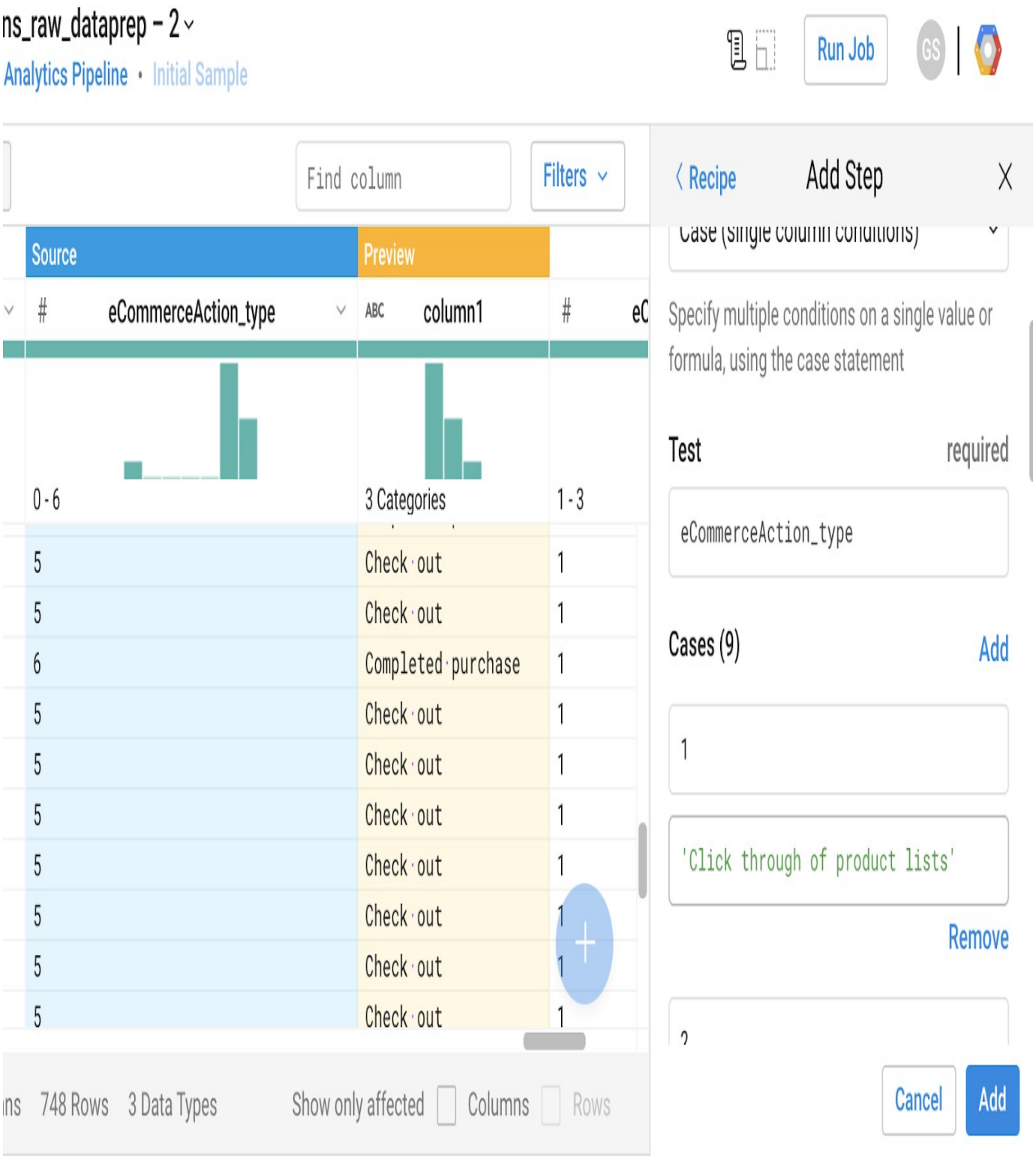
2. In the **Transformation** panel, type **case**, and then select **Conditional column**.



3. Select **Case on single column** from the drop-down.
4. For **Column to evaluate**, specify **eCommerceAction_type**.
5. Next to Cases (X), click **Add** 8 times for a total of 9 cases.
6. For each Case, specify the following mapping values (including the quotes):

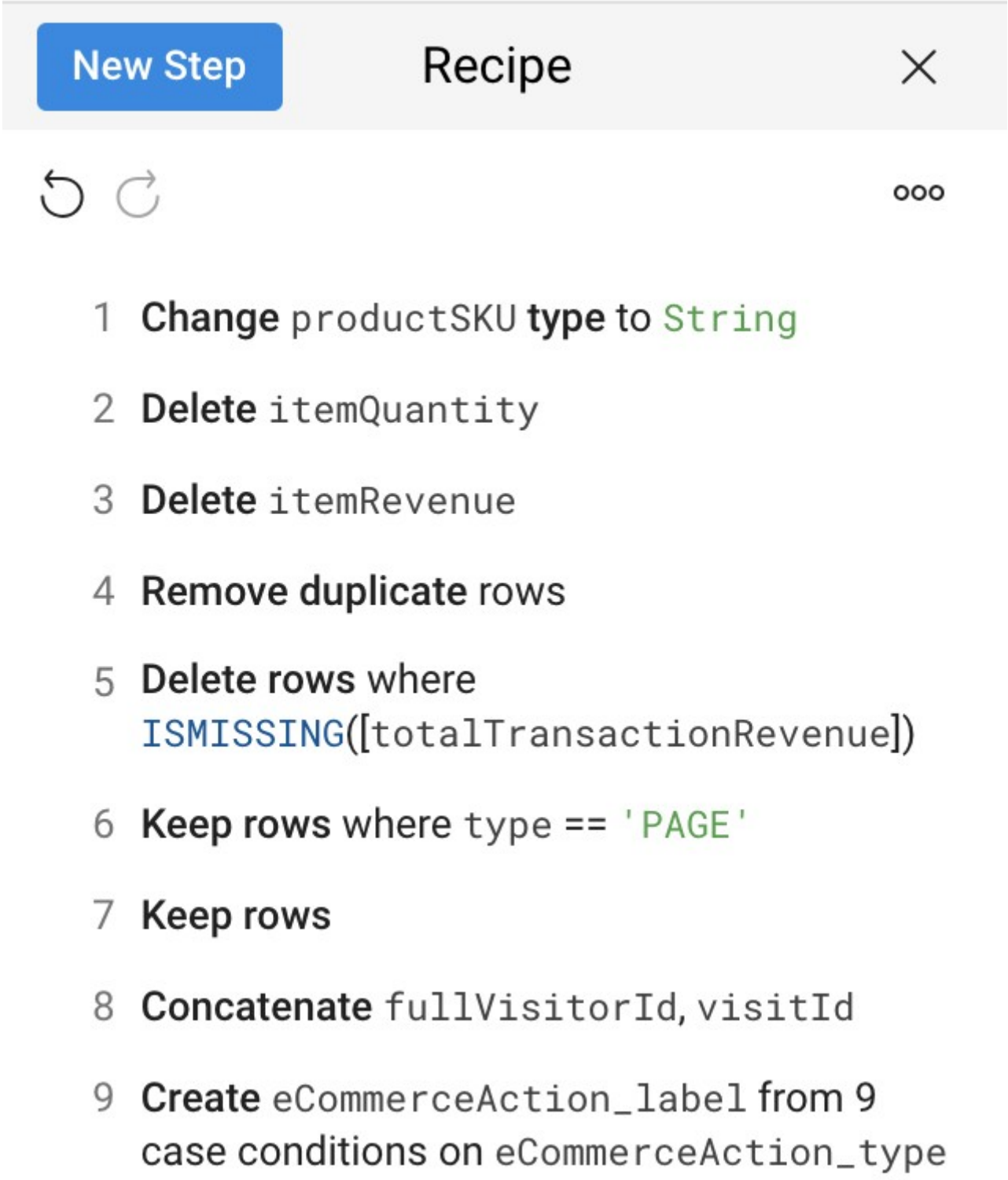
Value to compare	New value
1	'Click through of product lists'
2	'Product detail views'
3	'Add product(s) to cart'
4	'Remove product(s) from cart'
5	'Check out'
6	'Completed purchase'
7	'Refund of purchase'
8	'Checkout options'
0	'Unknown'

Example:



Leave the other fields at their default values.

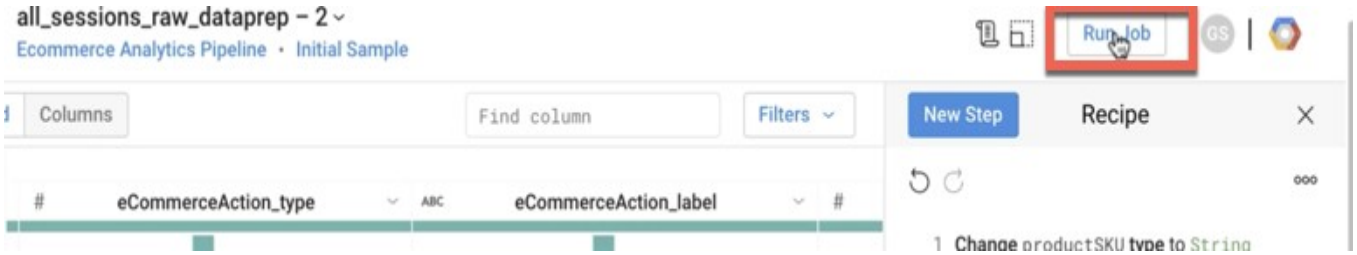
7. For **New column name**, type **eCommerceAction_label**, and then click **Add**.
8. Review the Recipe and compare it to this example:



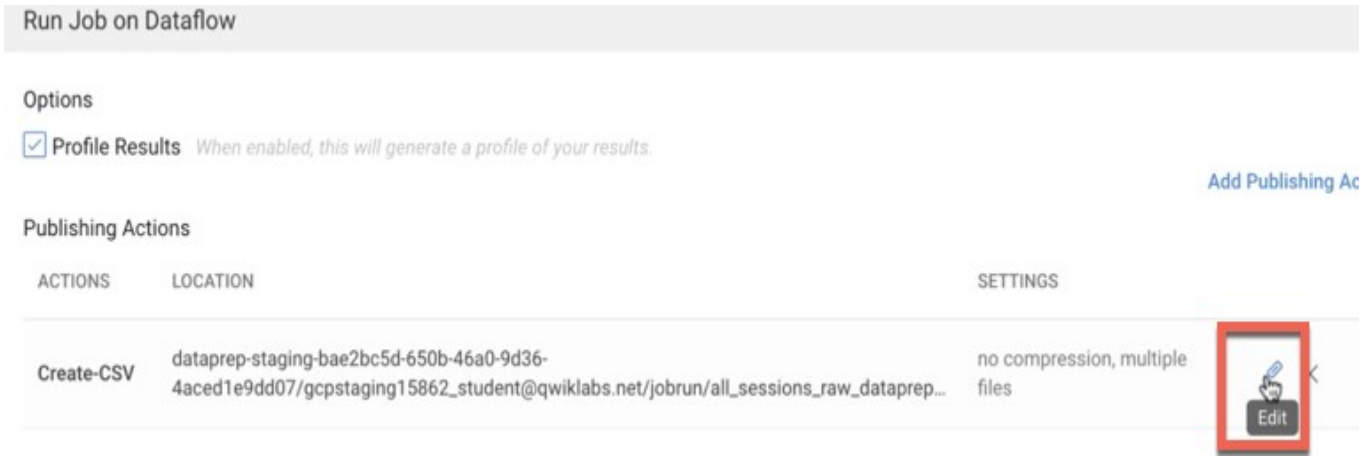
Task 8. Run and schedule Cloud Dataprep jobs to BigQuery

When you are satisfied with the flow, it's time to execute the transformation recipe against your source dataset. To do that, you execute and monitor a Cloud Dataprep job (which starts and runs a Cloud Dataflow job).

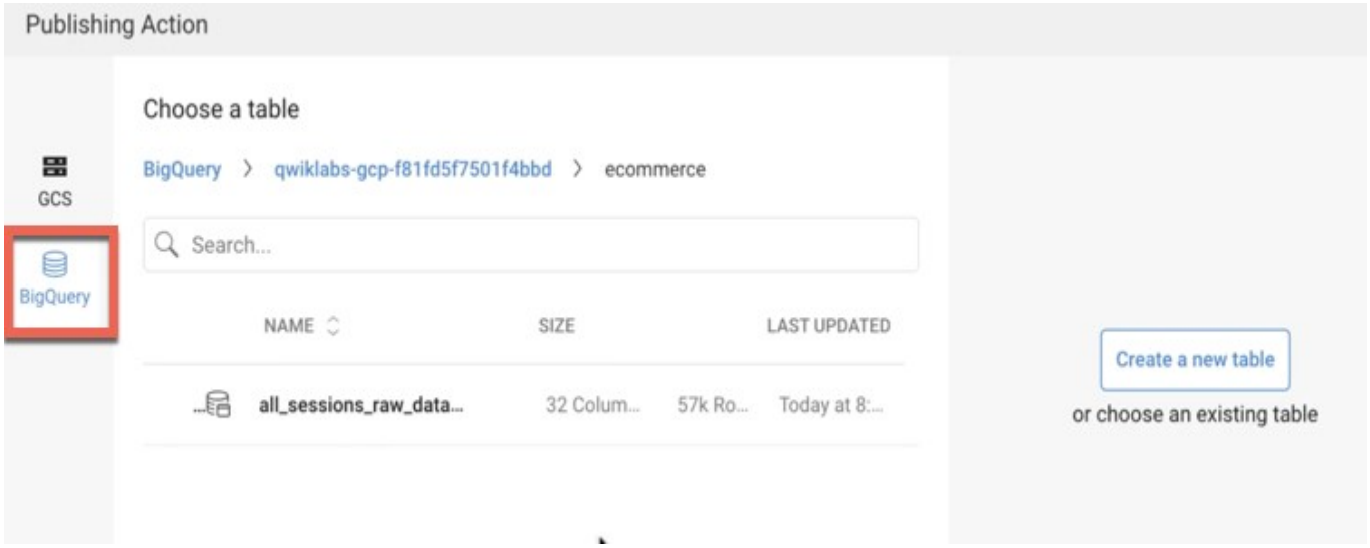
1. From the Transformer page, click **Run Job**.



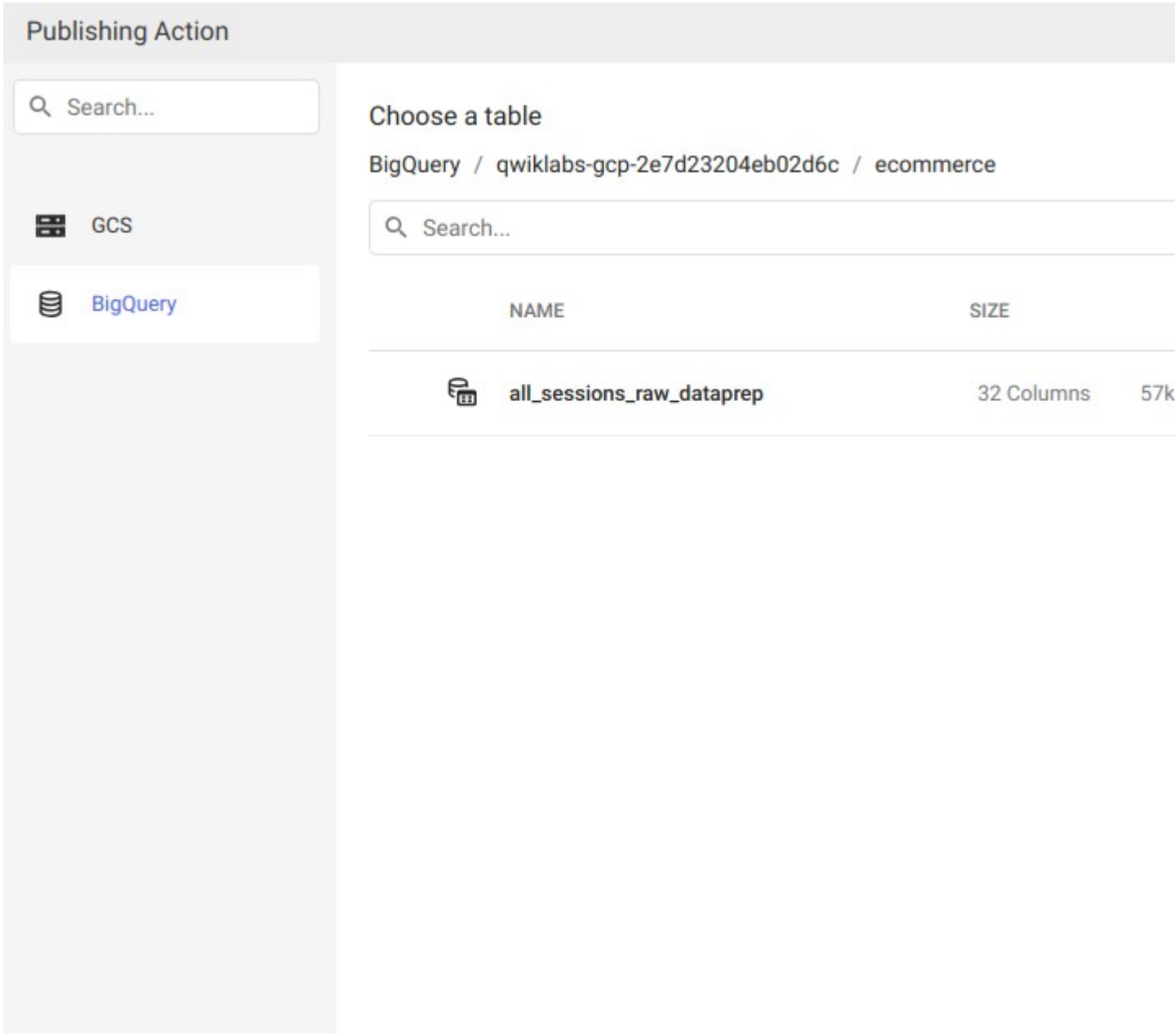
2. Click **Edit**



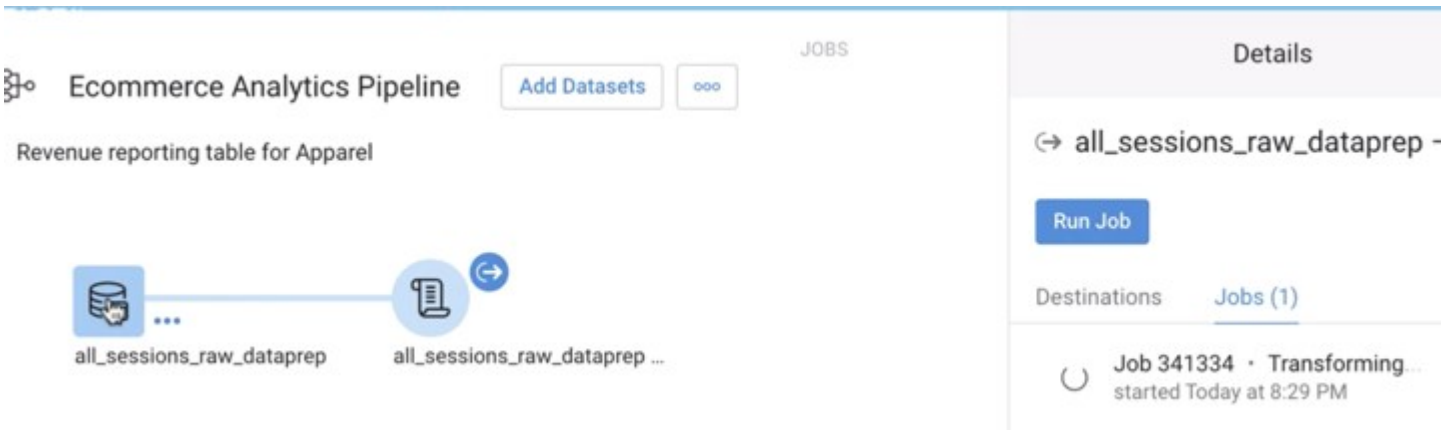
3. Select **BigQuery** and go into your ecommerce dataset, and then click **Create a new table**.



4. Name the output table **apparel_revenue** and select **Drop the table every run**.

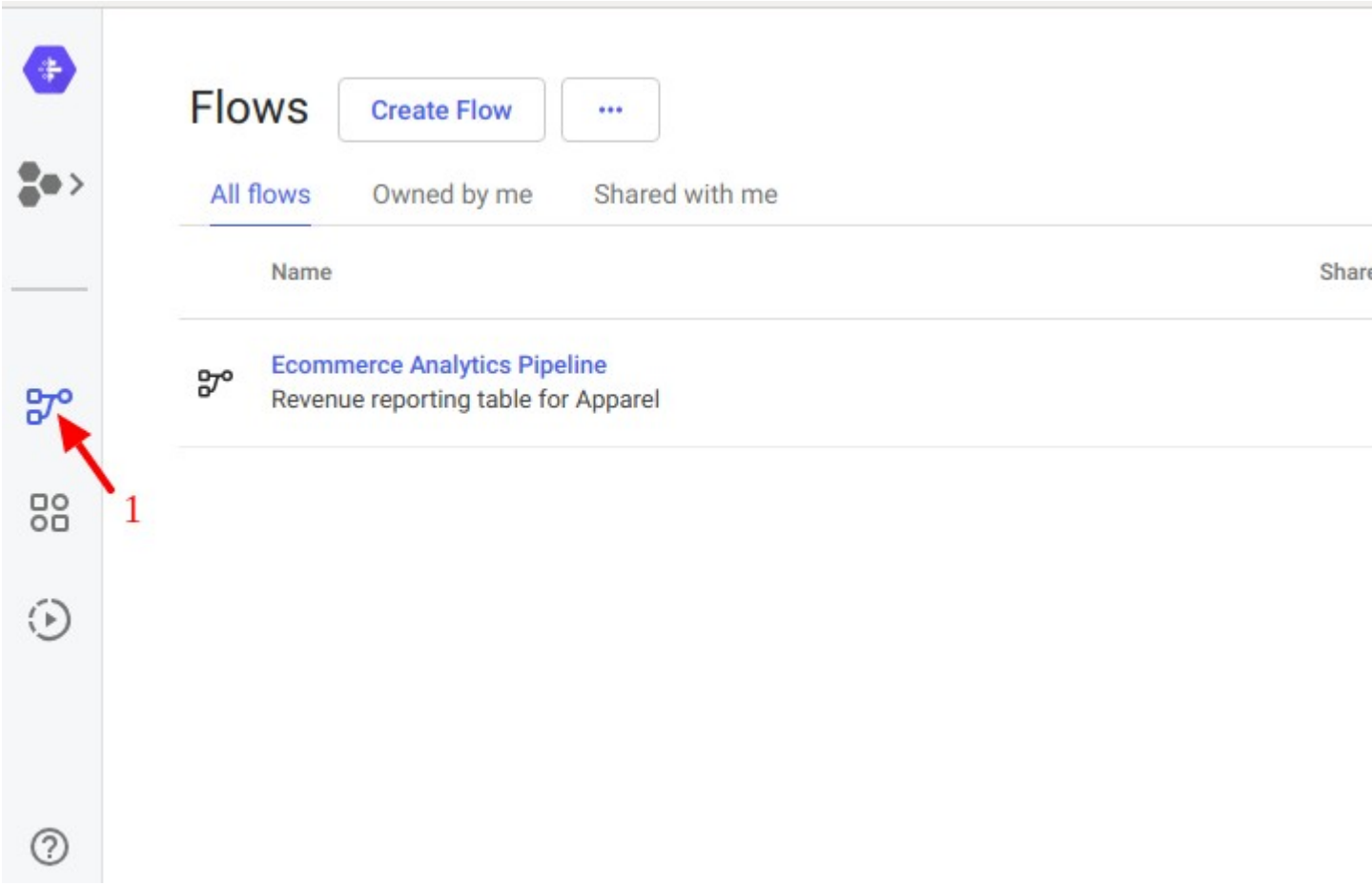


5. Click **Update**.
6. Click **Run Job**.



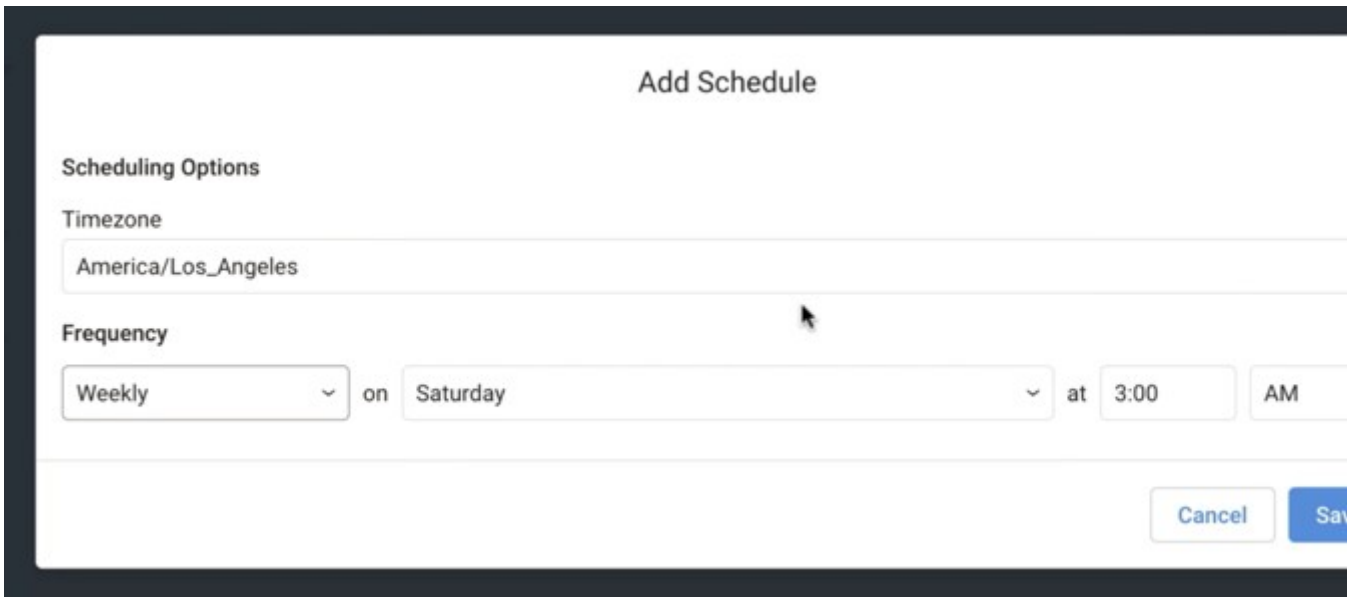
While the job completes (8 - 10 minutes), you can schedule your flow to run automatically.

7. To schedule a flow, click **Flows** > Hover over a flow name > **More Details [...]** > **Schedule Flow**.

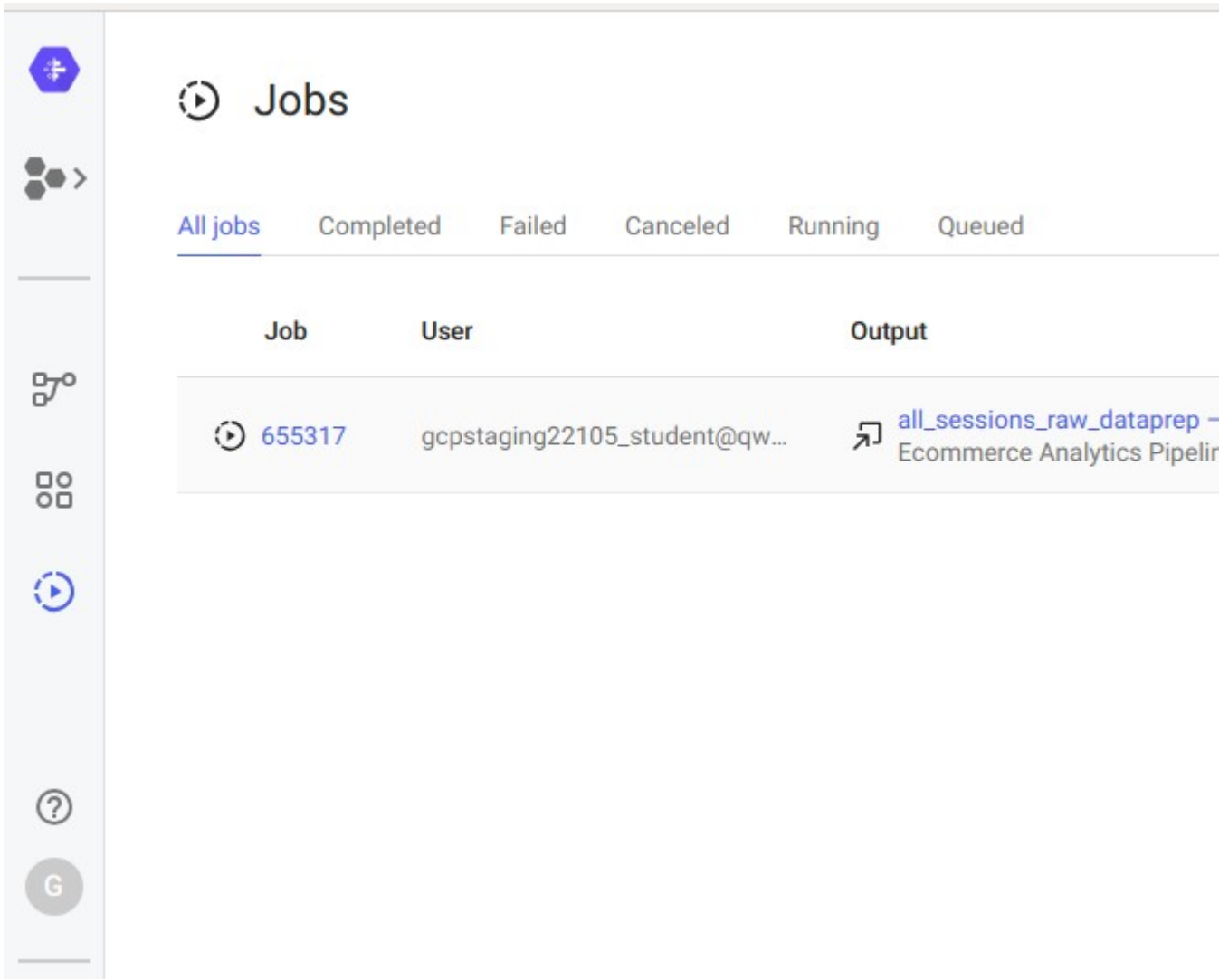


8. In the **Add Schedule** dialog:

- For **Frequency**, select **Weekly**.
- For day of week, select **Saturday**.
- For time, enter **3:00** and select **AM**.
- Click **Save**.

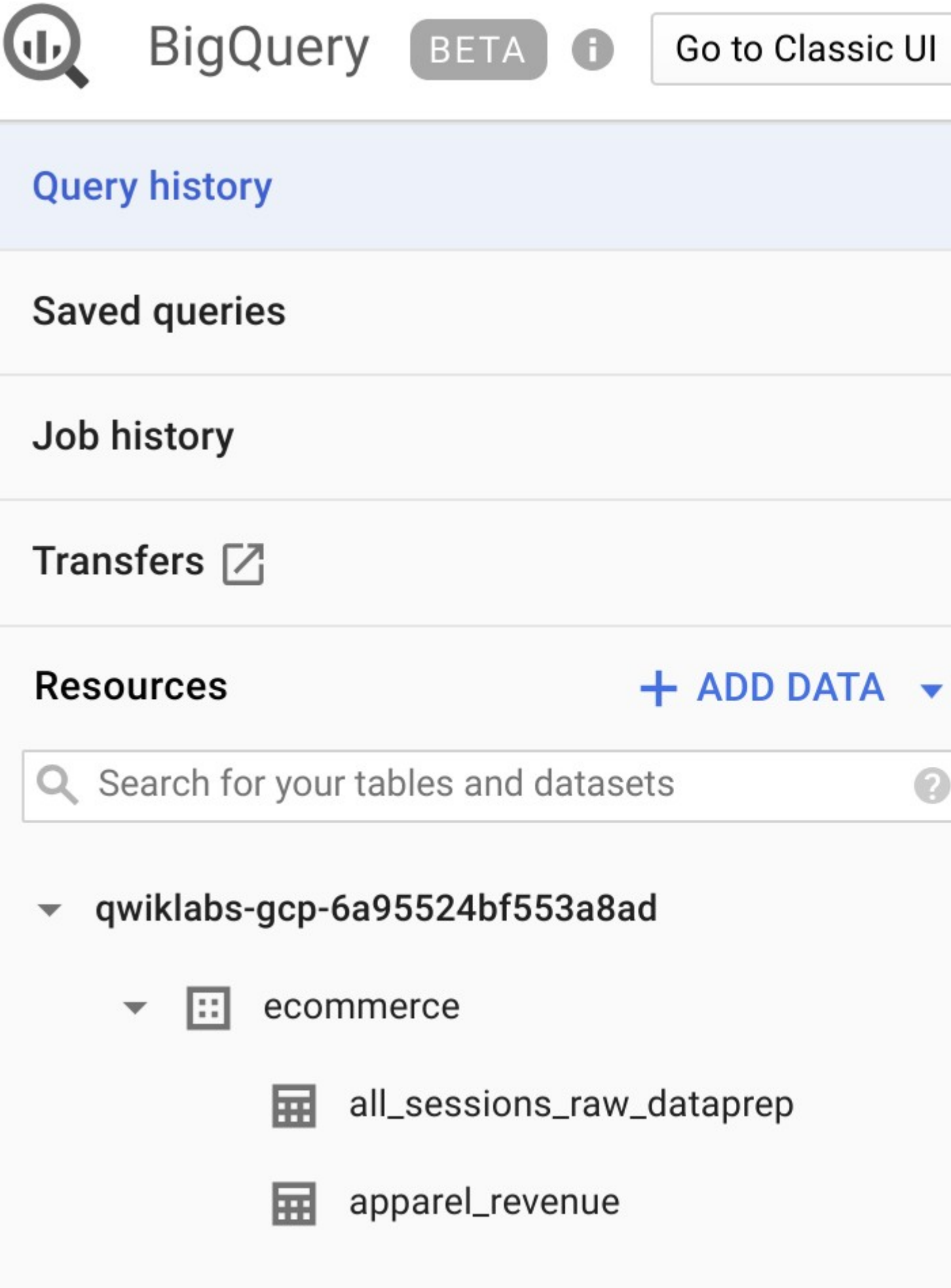


9. Click Jobs to monitor your Cloud Dataprep job.

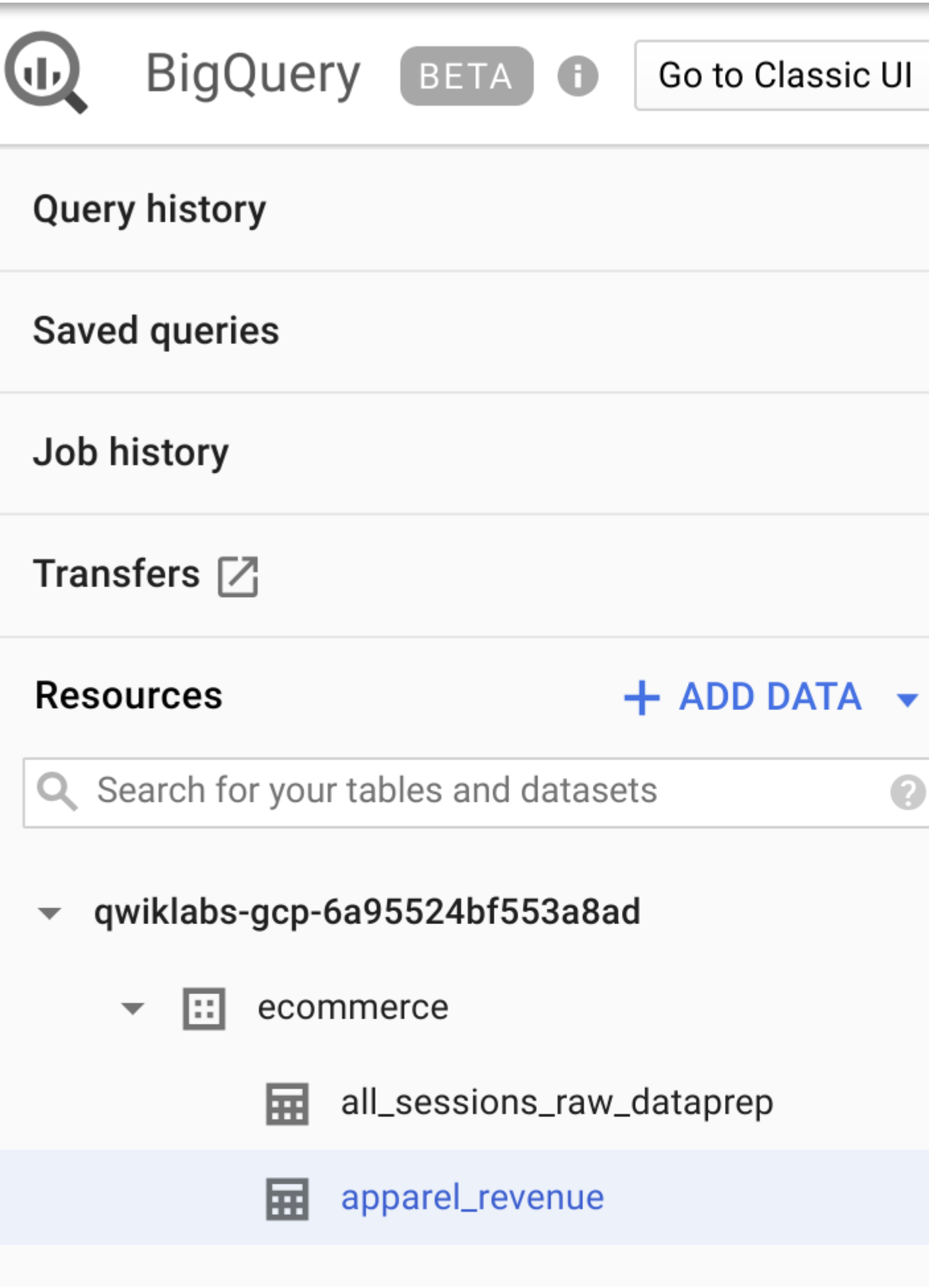


10. Wait 10 - 15 minutes for your job to run

After your Cloud Dataprep job finishes, refresh your BigQuery page and confirm that the output table **apparel_revenue** exists.


BigQuery
BETA
Go to Classic UI
Query history
Saved queries
Job history
Transfers
Resources
ADD DATA
Search for your tables and datasets
qwiklabs-gcp-6a95524bf553a8ad
ecommerce
all_sessions_raw_dataprep
apparel_revenue

Select **Preview** and ensure you have revenue transaction data for Apparel products.


BigQuery
BETA
Go to Classic UI
Query history
Saved queries
Job history
Transfers
Resources
ADD DATA
Search for your tables and datasets
qwiklabs-gcp-6a95524bf553a8ad
ecommerce
all_sessions_raw_dataprep
apparel_revenue