

Creating a Data Transformation Pipeline with Cloud Dataprep

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 you explore the Cloud Dataprep UI to build a data transformation pipeline and outputs results into BigQuery.

The dataset you'll use is an [ecommerce dataset](#) that has millions of Google Analytics session 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.

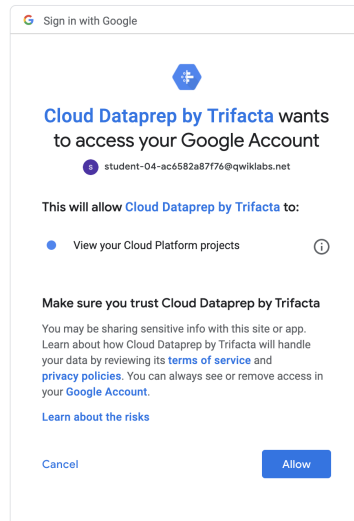
Objectives

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

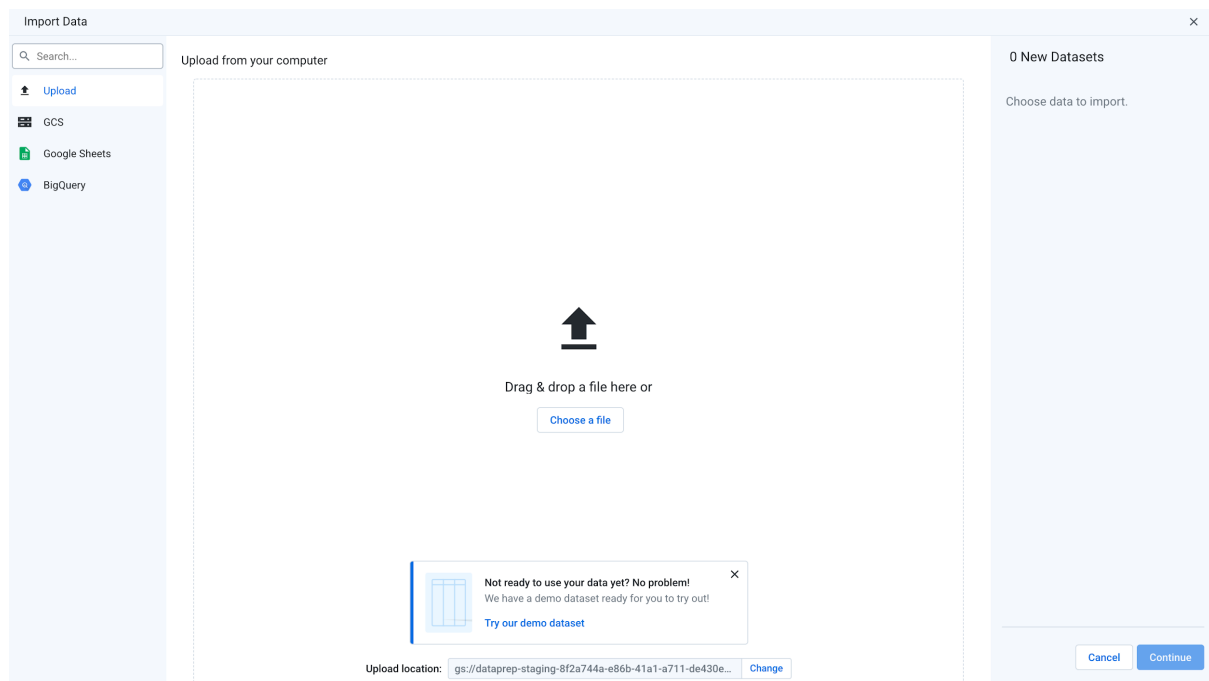
- Connect BigQuery datasets to Cloud Dataprep.
- Explore dataset quality with Cloud Dataprep.
- Create a data transformation pipeline with Cloud Dataprep.
- Run transformation jobs outputs to BigQuery.

Open Google Cloud Dataprep

1. In the Cloud Console go to the **Navigation menu**, and under **Big Data** select **Dataprep**.
2. To get into Cloud Dataprep, check that you agree to Google Dataprep Terms of Service, and then click **Accept**.
3. Click the checkbox and then click **Agree and Continue** when prompted to share account information with Trifacta.
4. Click **Allow** to give Trifacta access to your project.
5. Select your Qwiklabs credentials to sign in and click **Allow**.



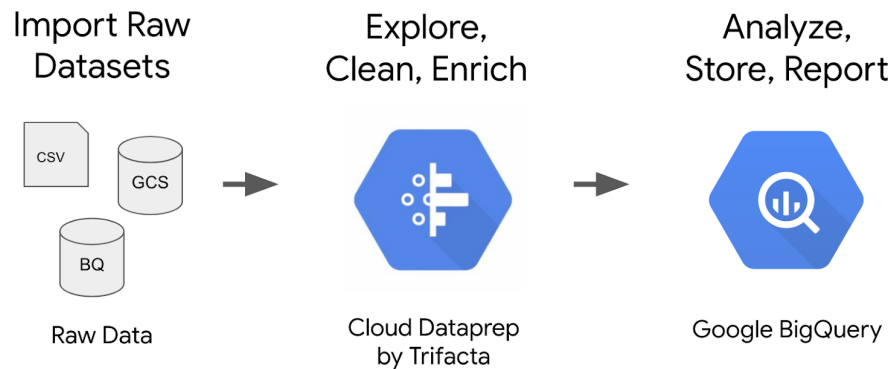
6. Check the box and click **Accept** to agree to Trifacta Terms of Service.
7. If prompted to use the default location for the storage bucket, click **Continue**.
8. For new users, a tutorial will launch, asking you to select datasets. Quit out of this screen by clicking **Cancel** or exiting out.



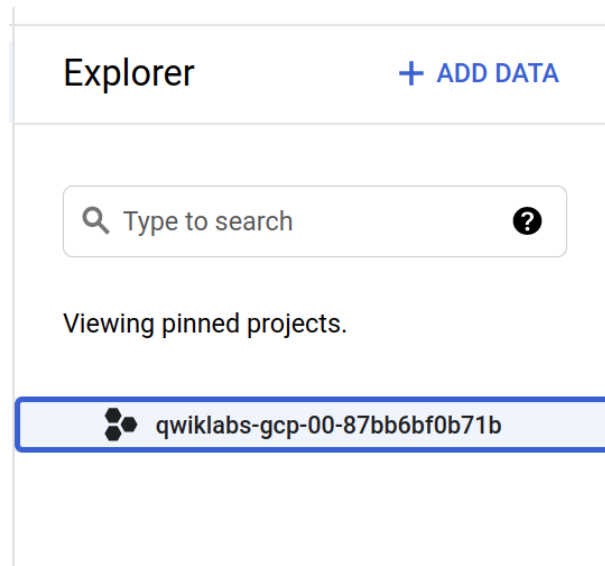
9. Click on the Dataprep icon in the top left corner to go to the home screen.

Creating a BigQuery Dataset

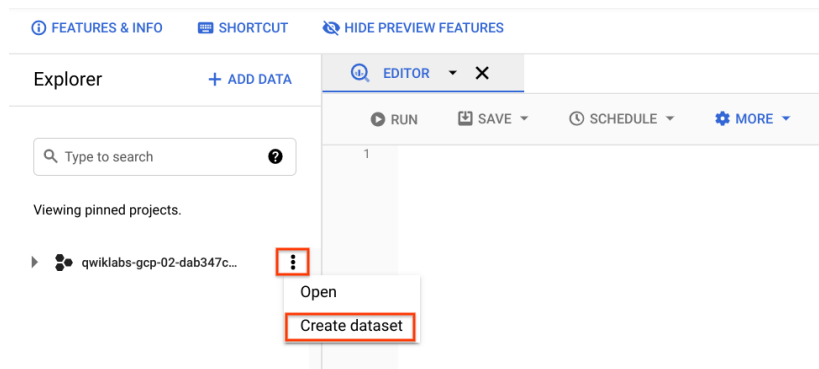
Although this lab is largely focused on Cloud Dataprep, you need BigQuery as an endpoint for dataset ingestion to the pipeline and as a destination for the output when the pipeline is completed.



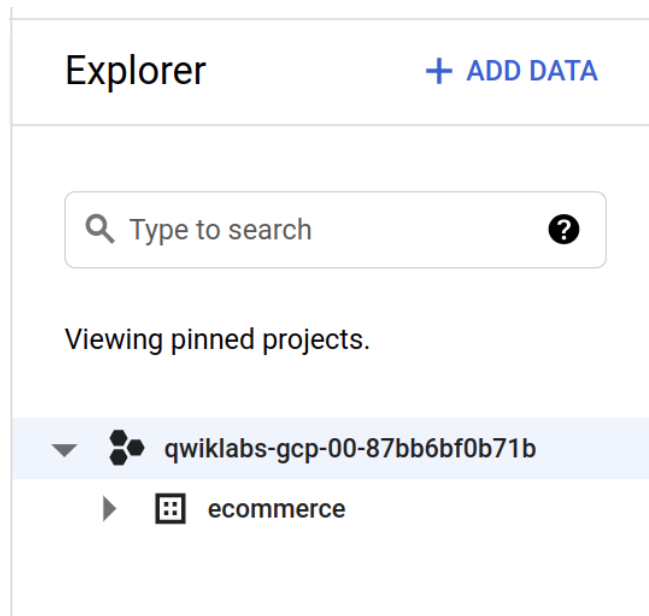
1. In the Cloud Console, select **Navigation menu > BigQuery**.
2. The **Welcome to BigQuery in the Cloud Console** message box opens. This message box provides a link to the quickstart guide and lists UI updates.
3. Click **Done**.
4. In the **Explorer** pane, select your project name:



5. In the left pane, under **Explorer** section, click on the **View actions** icon next to your project ID, then click **Create dataset**.



- For **Dataset ID**, type ecommerce.
 - Leave the other values at their defaults.
6. Click **CREATE DATASET**. You will now see your dataset under your project in the left-hand menu:

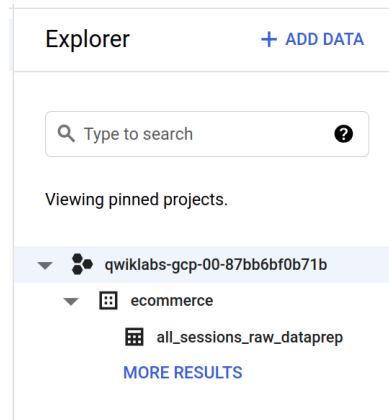


7. Navigate to the query **EDITOR** and copy and paste the following SQL query into it:

```
#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
```

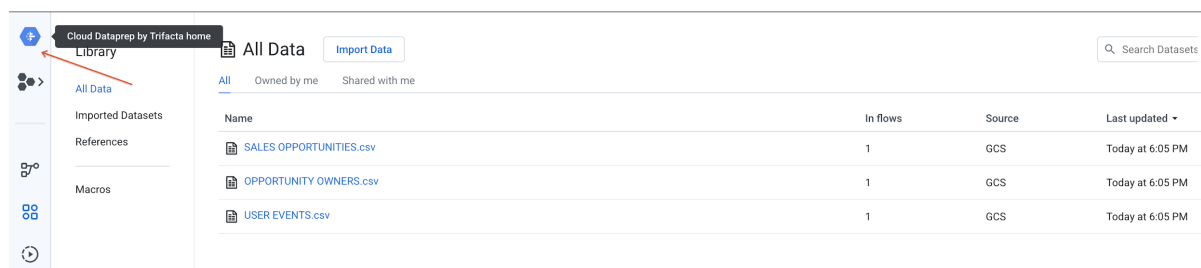
- Click **RUN**. This query copies over a subset of the public raw ecommerce dataset (one day's worth of session data, or about 56 thousand records) into a new table named `all_sessions_raw_dataprep`, which has been added to your ecommerce dataset for you to explore and clean in Cloud Dataprep.
- Confirm that the new table exists in your ecommerce dataset:



Connecting BigQuery data to Cloud Dataprep

In this task, you will connect Cloud Dataprep to your BigQuery data source. On the Cloud Dataprep page:

- Click on **Cloud Dataprep by Trifacta Home** icon.



- Click **Create Flow** in the top-right corner.
- Rename the **Untitled Flow** and specify these details:
 - For **Flow Name**, type Ecommerce Analytics Pipeline
 - For **Flow Description**, type Revenue reporting table

Rename ×

Flow Name

Ecommerce Analytics Pipeline

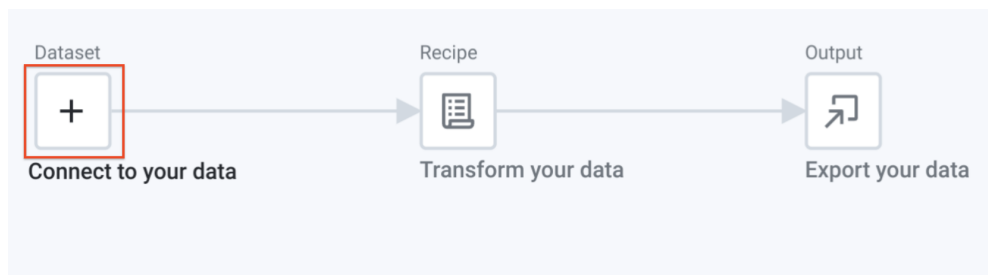
Flow Description

Revenue reporting table

Cancel

OK

4. Click **Ok**.
5. If prompted with a What 's a flow? popup, select **Don't show me any helpers**.
6. Click the **Add Icon** in the Dataset box.



7. In the **Add Datasets to Flow** dialog box, select **Import Datasets**.

Add datasets to flow ×

Q Search...

All (3)

Imported (3)

Reference (0)

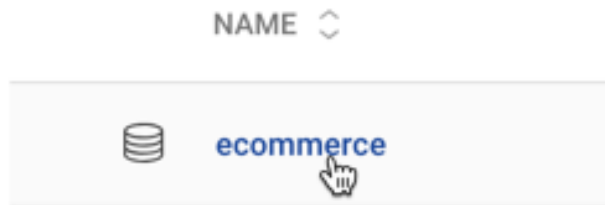
	NAME	SOURCE	LAST UPDATED
<input type="checkbox"/>	SALES OPPORTUNITIES.csv	GCS	Today at 6:05 PM
<input type="checkbox"/>	OPPORTUNITY OWNERS.csv	GCS	Today at 6:05 PM
<input type="checkbox"/>	USER EVENTS.csv	GCS	Today at 6:05 PM

Import datasets

Cancel

Add

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



10. Click on the **Create dataset** icon (+ sign) on the left of the `all_sessions_raw_dataprep` table.

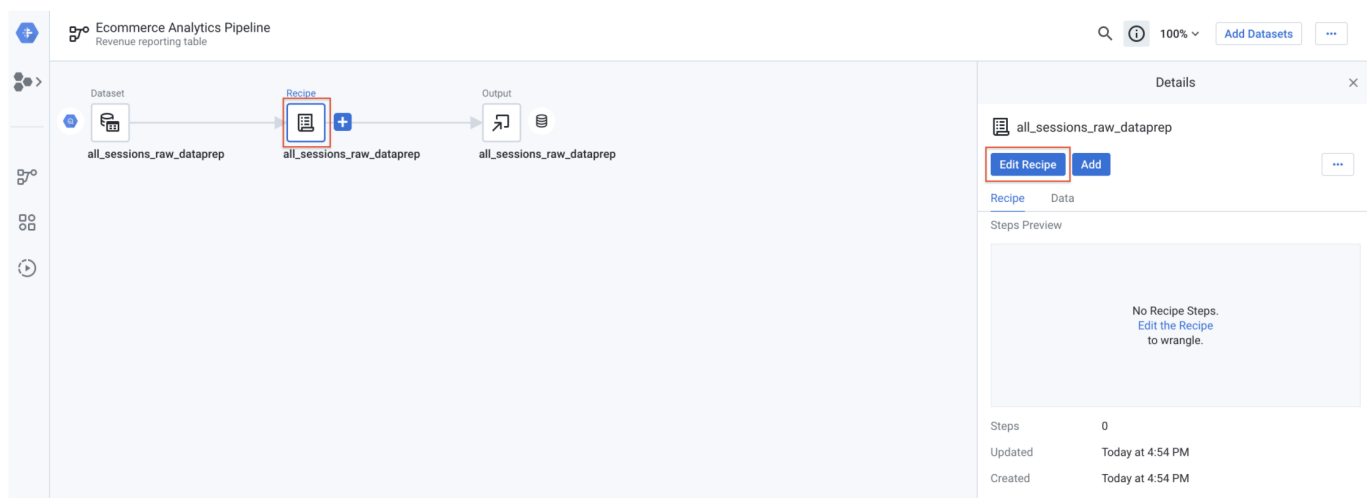
11. Click **Import & Add to Flow** in the bottom right corner.

The data source automatically updates. You are ready to go to the next task.

Exploring ecommerce data fields with a UI

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

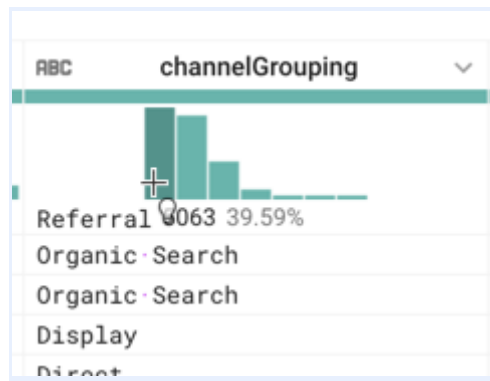
1. Click on the **Recipe icon** and then select **Edit Recipe**.



Cloud Dataprep loads a sample of your dataset into the Transformer view. This process might take a few seconds. You are now ready to start exploring the data!

Answer the following questions:

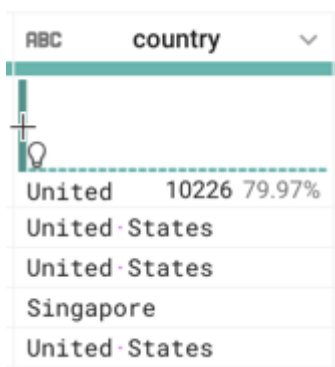
- How many columns are there in the dataset?



Answer: Referral. A [referring site](#) is typically any other website that has a link to your content. An example here is a different website reviewed a product on our ecommerce website and linked to it. This is considered a different acquisition channel than if the visitor came from a search engine.

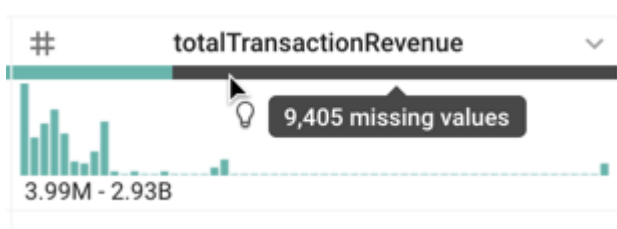
Tip: When looking for a specific column, click the **Find column** icon (🔍) in the top right corner, then start typing the column's name in the **Find column** textfield, then click on the column's name. This will automatically scroll the grid to bring the column on the screen.

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




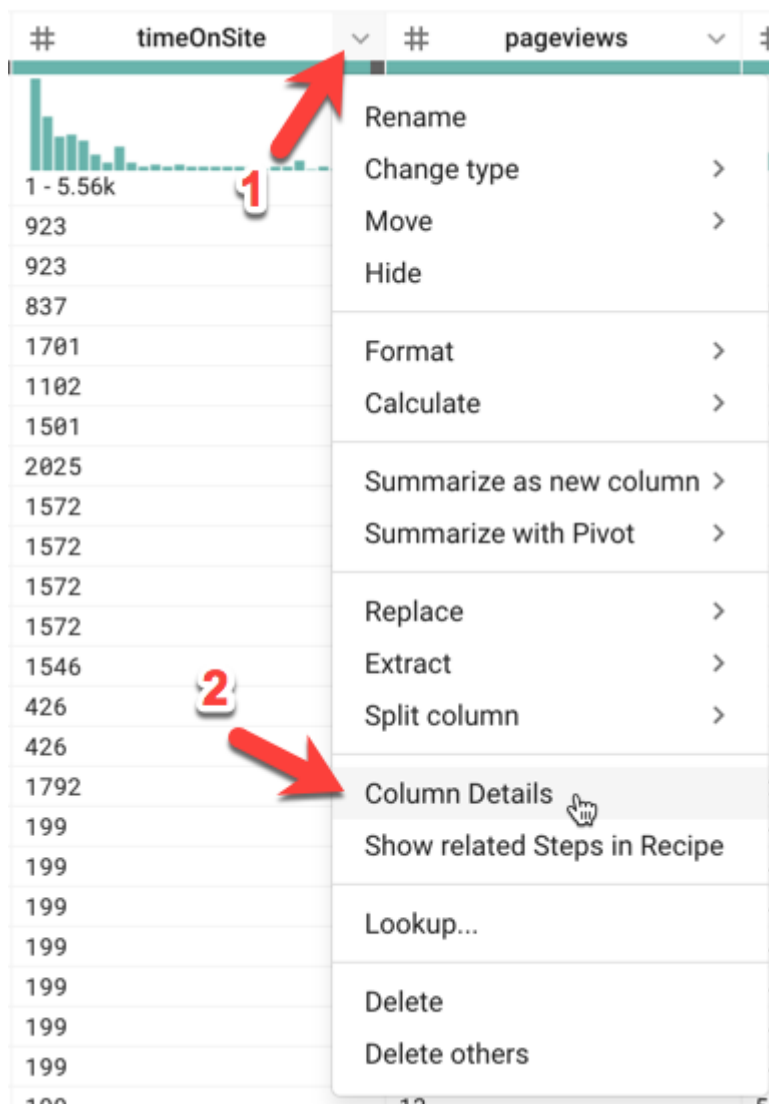
Answer: United States, India, United Kingdom

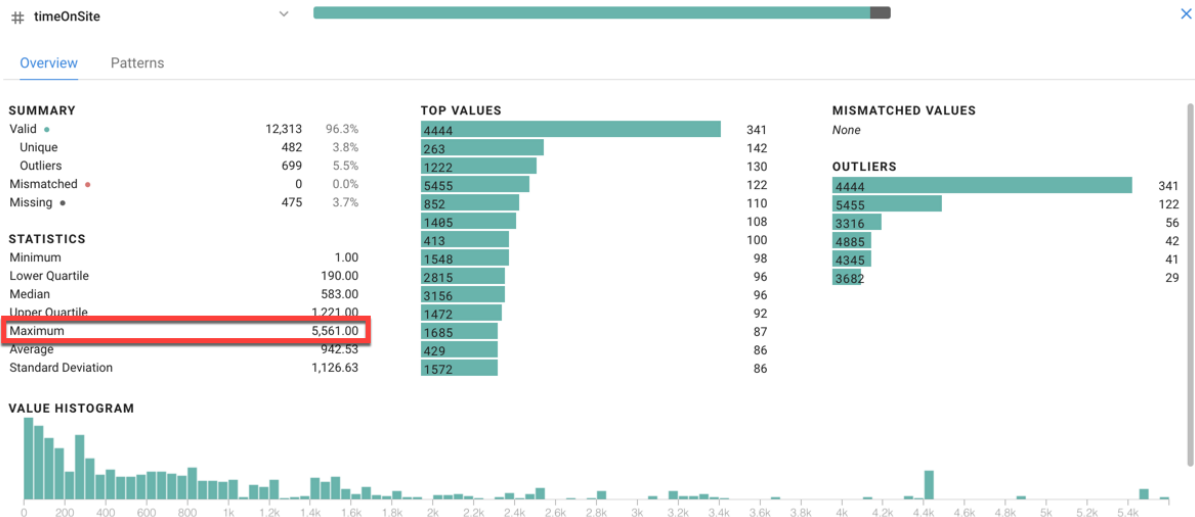
- What does the grey bar under **totalTransactionRevenue** represent?



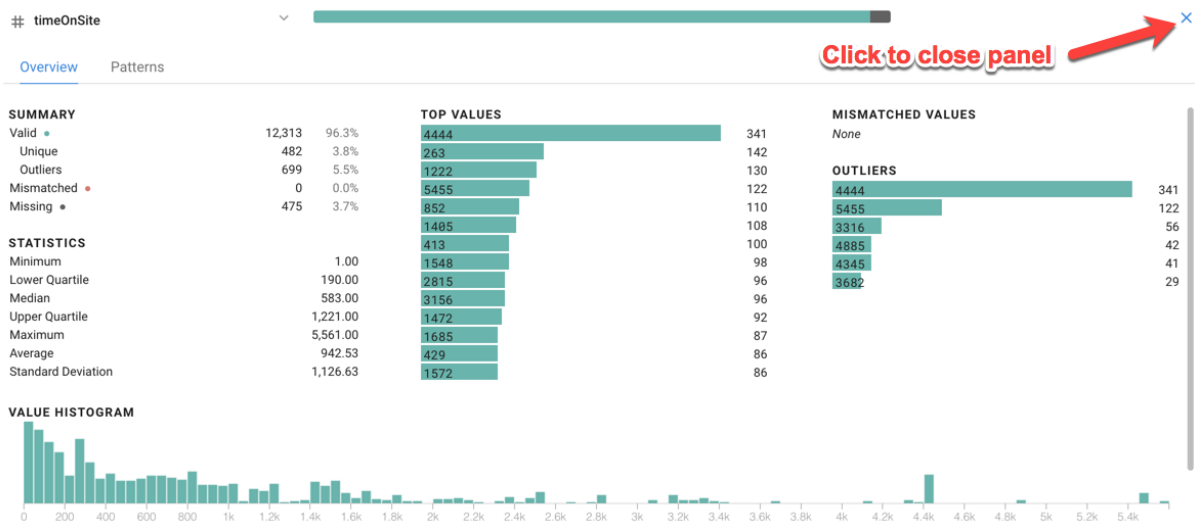
Answer: Missing values for the totalTransactionRevenue field. This means that a lot of sessions in this sample did not generate revenue. Later, we will filter out these values so our final table only has customer transactions and associated revenue.

- What is the maximum timeOnSite in seconds, maximum pageviews, and maximum sessionQualityDim for the data sample? (Hint: Open the menu to the right of the timeOnSite column by clicking  the **Column Details** menu)





To close the details window, click the **Close Column Details** (X) button in the top right corner. Then repeat the process to view details for the pageviews and sessionQualityDim columns.



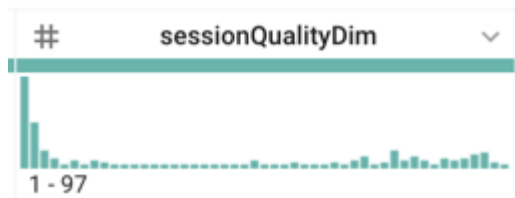
Answers:

- **Maximum Time On Site:** 5,561 seconds (or 92 minutes)
- **Maximum Pageviews:** 155 pages
- **Maximum Session Quality Dimension:** 97

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

Note on averages: Use extra caution when performing aggregations like averages over a column of data. We need to first ensure fields like timeOnSite are only counted once per session. We'll explore the uniqueness of visitor and session data in a later lab.

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

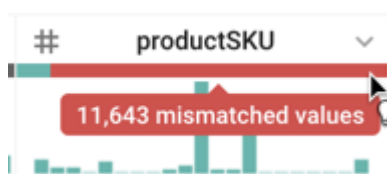


Answer: No, they are skewed to lower values (low quality sessions), which is expected.

- What is the **date** range for the dataset? Hint: Look at **date** field

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

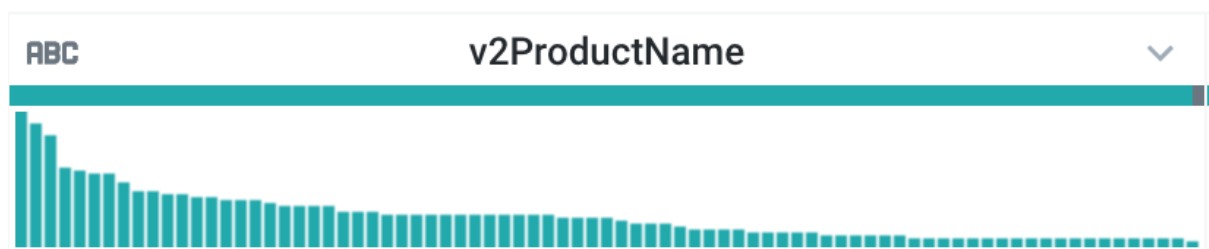
- You might see a red bar under the productSKU column. If so, what might that mean?



Answer: A red bar indicates mismatched values. While sampling data, Cloud Dataprep attempts to automatically identify the type of each column. If you do not see a red bar for the productSKU column, then this means that Cloud Dataprep

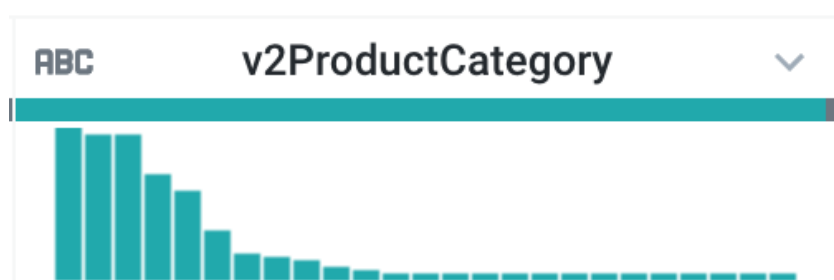
correctly identified the type for the column (i.e. the String type). If you do see a red bar, then this means that Cloud Dataprep found enough number values in its sampling to determine (incorrectly) that the type should be Integer. Cloud Dataprep also detected some non-integer values and therefore flagged those values 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 will fix that later in this lab.

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



Answer: Nest products

- Looking at the v2ProductCategory column, what are some of the most popular product categories?



Answers:

The most popular product categories are:

- **Nest**
- **Bags**
- **(not set)** (which means that some sessions are not associated with a category)
- 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 values in the **type** column?

Answer: PAGE and EVENT

A user can have many different interaction types when browsing your website. Types include recording session data when viewing a PAGE or a special EVENT (like "clicking on a product") and other types. Multiple hit types can be triggered at the exact same time so you will often filter on type to avoid double counting. We'll explore this more in a later analytics lab.

- What is the maximum productQuantity?

Answer: 100 (your answer may vary)

productQuantity indicates how many units of that product were added to cart.

100 means 100 units of a single product was added.

- What is the dominant currencyCode for transactions?

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

- Are there valid values for itemQuantity or itemRevenue?

Answer: No, they are all NULL (or missing) values.

Note: After exploration, in some datasets you may find duplicative or deprecated columns. We will be using productQuantity and productRevenue fields instead and dropping the itemQuantity and itemRevenue fields later in this lab to prevent confusion for our report users.

- What percentage of transactionId values are valid? What does this represent for our ecommerce dataset?

ABC transactionId		
<div>Overview</div> <div>Patterns</div>		
SUMMARY		
Valid	582	4.6%
Unique	97	0.8%
Outliers	0	0.0%
Mismatched	0	0.0%
Missing	12,206	95.4%
STRING LENGTH STATISTICS		
Minimum	15.00	
Lower Quartile	15.00	
Median	15.00	
Upper Quartile	15.00	
Maximum	15.00	
Average	15.00	
Standard Deviation	0.00	

- 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 values are there, and what is the most common value?

Hint: Count the distinct number of histogram columns.



Answers: There are seven values found in our sample. The most common value is zero 0 which indicates that the type is unknown. This makes sense as the majority of the web sessions on our website will not perform any ecommerce actions as they are just browsing.

- Using the [schema](#), what does eCommerceAction_type = 6 represent?

Hint: Search for eCommerceAction type and read the description for the mapping


Answer: 6 maps to "Completed purchase". Later in this lab we will ingest this mapping as part of our data pipeline.

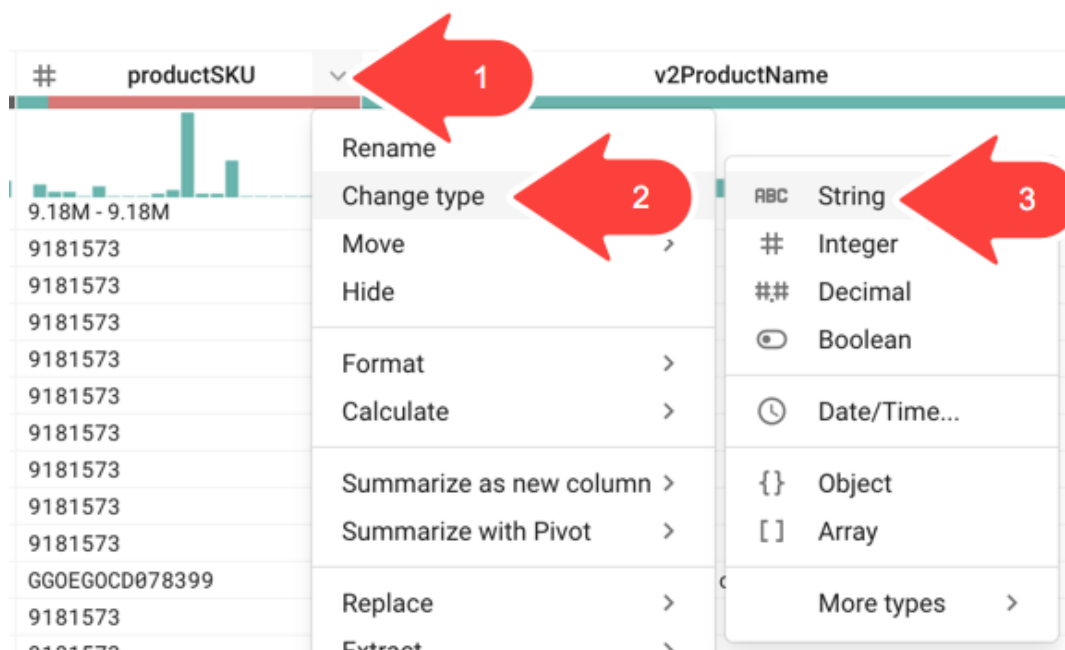
<code>commerceAction.action_type</code>	STRING	The action type. Click through of product lists = 1, Product detail views = 2, Add product(s) to cart = 3, Remove product(s) from cart = 4, Check out = 5, Completed purchase = 6, Refund of purchase = 7, Checkout options = 8, Unknown = 0.
---	--------	---

Cleaning the data

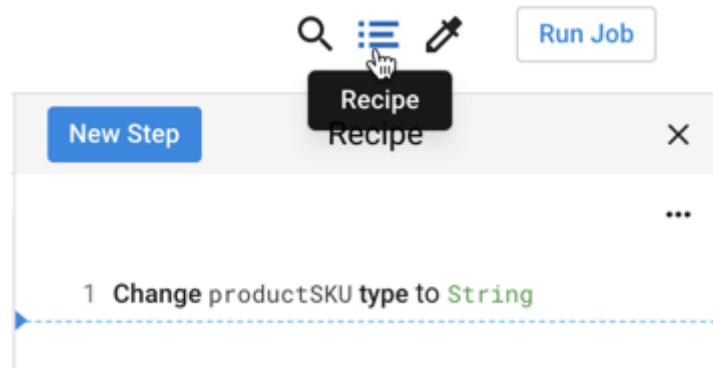
In this task, you will clean the data by deleting unused columns, eliminating duplicates, creating calculated fields, and filtering out unwanted rows.

Converting the productSKU column data type

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



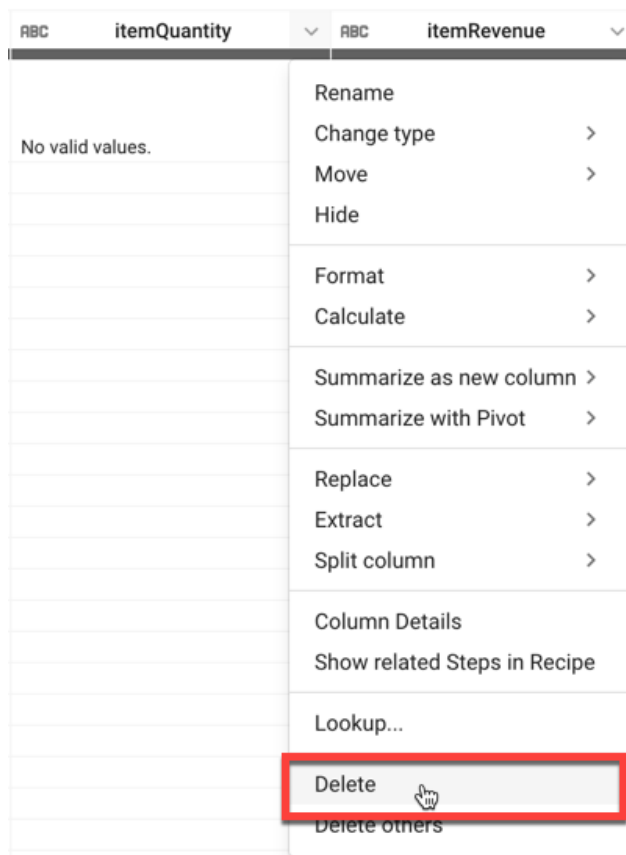
Verify that the first step in your data transformation pipeline was created by clicking on the **Recipe** icon:



Deleting unused columns

As we mentioned earlier, we will be deleting the **itemQuantity** and **itemRevenue** columns as they only contain NULL values are not useful for the purpose of this lab.

- Open the menu for the **itemQuantity** column, and then click **Delete**.

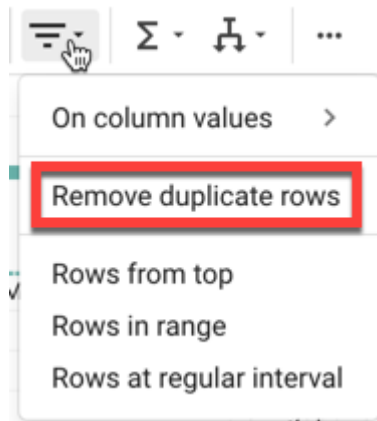


- Repeat the process to delete the **itemRevenue** column.

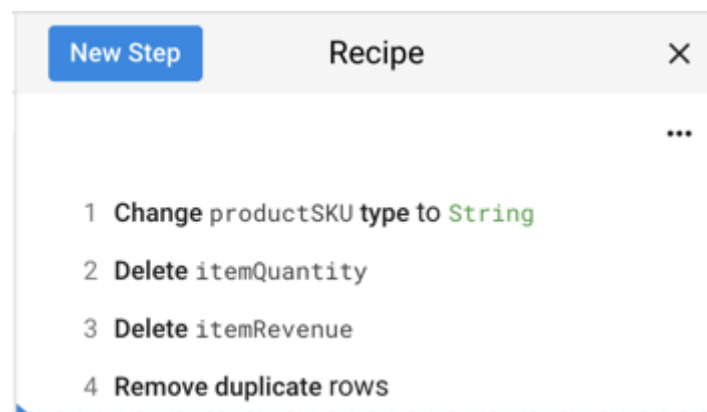
Deduplicating rows

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 the **Filter rows** icon in the toolbar, then click **Remove duplicate rows**.



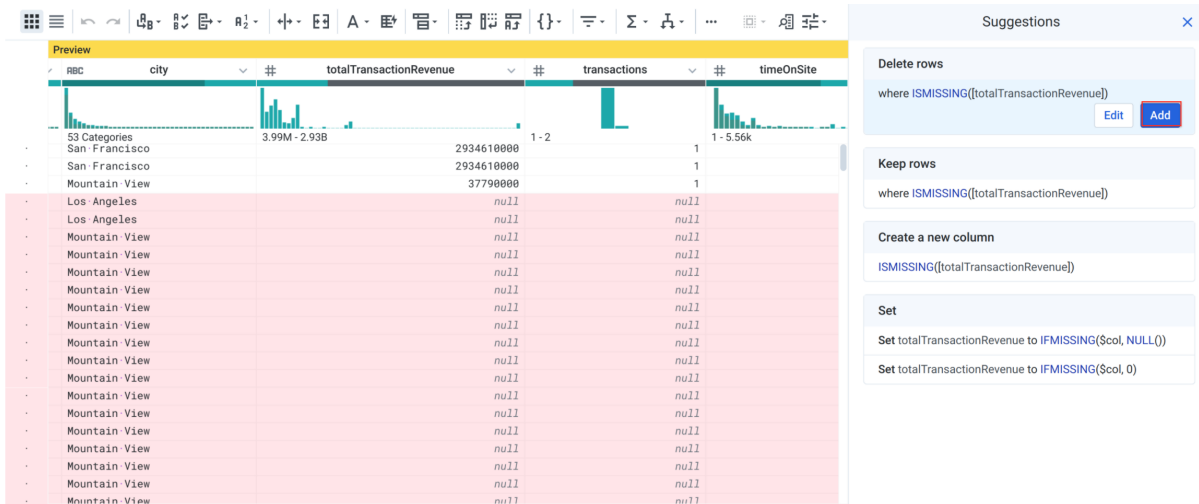
2. Click **Add** in the right-hand panel.
3. Review the recipe that you created so far, it should resemble the following:



Filtering 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.

2. In the **Suggestions** panel, in **Delete rows**, click **Add**.

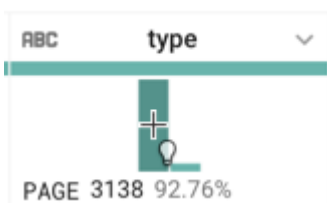


This step filters your dataset to only include transactions with revenue (where **totalTransactionRevenue** is not NULL).

Filtering sessions for PAGE views

The dataset contains sessions of different types, for example **PAGE** (for page views) or **EVENT** (for triggered events like "viewed product categories" or "added to cart"). To avoid double counting session pageviews, add a filter to only include page view related hits.

1. In the histogram below the **type** column, click the bar for **PAGE**. All rows with the type **PAGE** are now highlighted in green.



2. In the **Suggestions** panel, in **Keep rows**, and click **Add**.

The screenshot shows a data table interface. The main table has columns: #, sessionQualityDim, date, #, visitId, RBC, type, RBC, and productRefundAr. The 'date' column is filtered for 'Aug 1 - Aug 1'. The 'visitId' column has a filter '1.5B - 1.5B'. The 'type' column has a filter '2 Categories'. The 'RBC' column has a filter 'No valid values.'. The 'productRefundAr' column has a filter 'null'. The table contains 20 rows of data. On the right, there is a 'Suggestions' panel with the following sections:

- Keep rows**: where type == 'PAGE' (Buttons: Edit, Add)
- Delete rows**: where type == 'PAGE'
- Set**: Set type to IF(type == 'PAGE', NULL(), \$col)
- Create a new column**: type == 'PAGE'
- Deduplicate rows**: where every value is the same

Enriching 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.

As we see, **visitId** is not unique across all users. We will need to create a unique identifier.

Creating 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 the **Merge columns** icon in the toolbar.



2. For **Columns**, select `fullVisitorId` and `visitId`.
3. For **Separator** type a single hyphen character: `-`.
4. For the **New column name**, type `unique_session_id`.

[illegible]

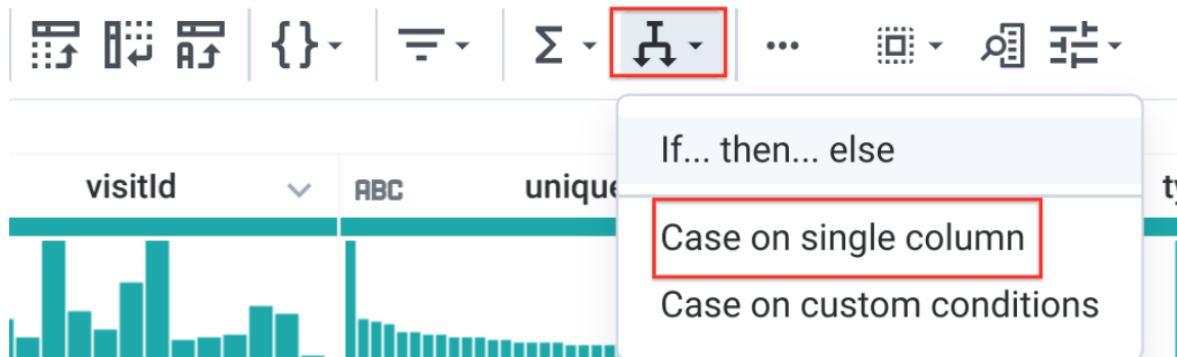
5. Click **Add**.

The `unique_session_id` is now a combination of the `fullVisitorId` and `visitId`. We will explore in a later lab whether each row in this dataset is at the unique session level (one row per user session) or something even more granular.

Creating a case statement for the ecommerce action type

As you saw earlier, values in the `eCommerceAction_type` column are integers that map to actual ecommerce actions performed in that session. For example, 3 = "Add to Cart" or 5 = "Check out". This mapping will not be immediately apparent to our end users so let's create a calculated field that brings in the value name.

1. Click on **Conditions** in the toolbar, then click **Case on single column**.



2. For **Column to evaluate**, specify `eCommerceAction_type`.
3. Next to **Cases (1)**, click **Add** 8 times for a total of 9 cases.

Conditions

×

Condition type

required

Case on single column

▼

Specify multiple conditions on a single value or formula, using the case statement

Column to evaluate

required

Select a column

▼

Cases (1)

+ Add

Comparison

Enter a value or formula

New value

Enter a value or formula

Default value

Edit formula

New column name

Insert new name

Cancel

Add

4. For each **Case**, specify the following mapping values (including the single quote characters):

Comparison	New value
0	'Unknown'
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'

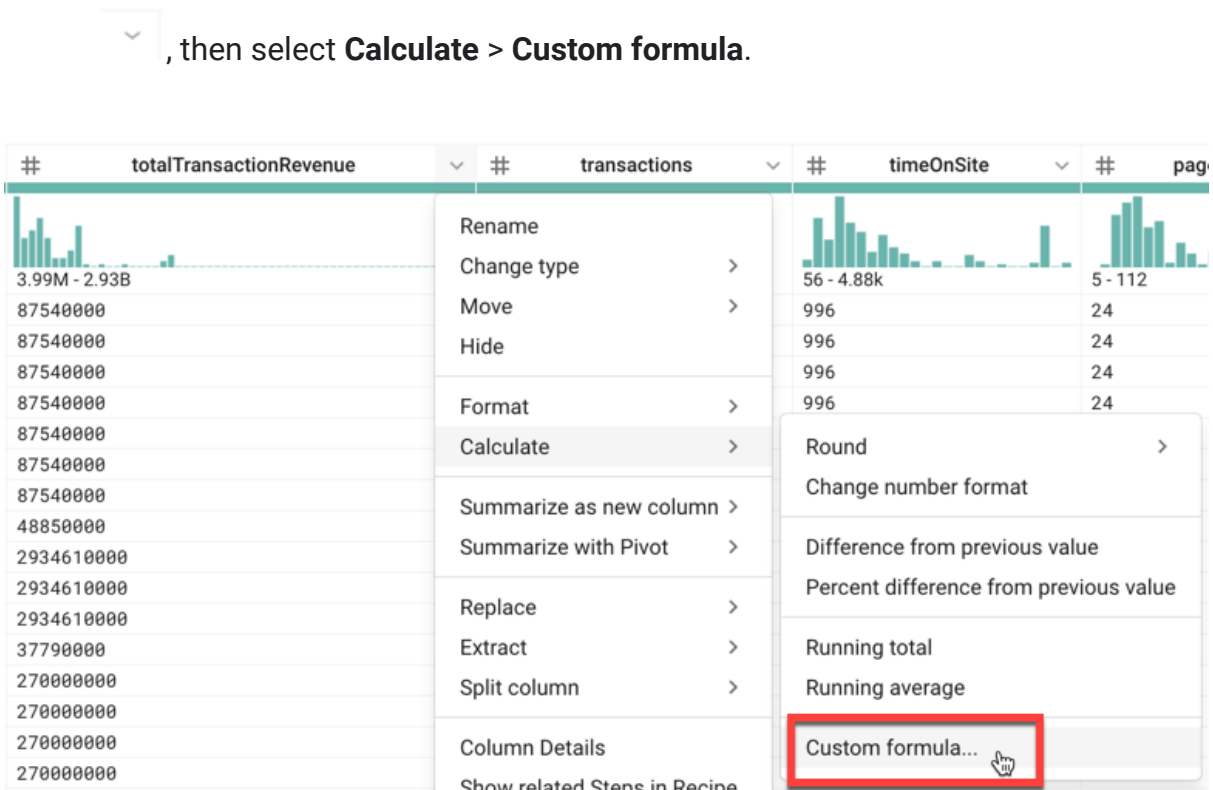
The screenshot shows a data tool interface. On the left is a data table with columns: #, eCommerceAction_type, and eCommerceAction_label. The table contains 20 rows. The first row has values 0-6, 0, and 4 Categories. The subsequent rows have values 0, 0, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 0. The eCommerceAction_label column contains values: Unknown, Unknown, Unknown, Unknown, Product detail views, Unknown, Unknown, Unknown, Unknown, Unknown, Product detail views, Unknown, Unknown, Unknown, Product detail views, Unknown, Unknown, Unknown, Unknown, Product detail views, Unknown, Unknown. On the right is a 'Conditions' panel. It has a 'Condition type' dropdown set to 'Case on single column'. Below it is a text box: 'Specify multiple conditions on a single value or formula, using the case statement'. Then, 'Column to evaluate' is set to '# eCommerceAction_type'. There are three cases listed: Case (9) with comparison '0' and new value 'Unknown'; Case 1 with comparison '1' and new value 'Click through of product lists'; Case 2 with comparison '2' and new value 'Product detail views'; and Case 3 with comparison '3' and an empty new value field. At the bottom of the panel are 'Cancel' and 'Add' buttons.

5. For **New column name**, type eCommerceAction_label. Leave the other fields at their default values.
6. Click **Add**.

Adjusting values in the totalTransactionRevenue column

As mentioned in the [schema](#), the **totalTransactionRevenue** column contains values passed to Analytics multiplied by 10^6 (e.g., 2.40 would be given as 2400000). You now divide contents of that column by 10^6 to get the original values.

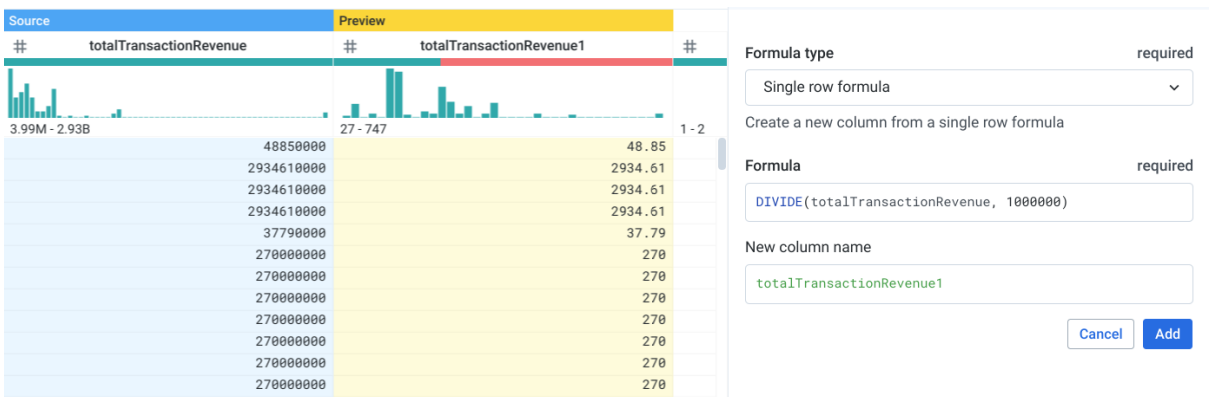
1. Open the menu to the right of the **totalTransactionRevenue** column by clicking



, then select **Calculate** > **Custom formula**.

#	totalTransactionRevenue	#	transactions	#	timeOnSite	#	pag
3.99M - 2.93B				56 - 4.88k		5 - 112	
87540000				996		24	
87540000				996		24	
87540000				996		24	
87540000				996		24	
87540000							
87540000							
87540000							
48850000							
2934610000							
2934610000							
2934610000							
37790000							
270000000							
270000000							
270000000							
270000000							
270000000							

2. For **Formula**, type: `DIVIDE(totalTransactionRevenue, 1000000)` and for **New column name**, type: `totalTransactionRevenue1`. Notice the preview for the transformation:



Source		Preview	
#	totalTransactionRevenue	#	totalTransactionRevenue1
3.99M - 2.93B		27 - 747	
48850000		48.85	
2934610000		2934.61	
2934610000		2934.61	
2934610000		2934.61	
37790000		37.79	
270000000		270	
270000000		270	
270000000		270	
270000000		270	
270000000		270	
270000000		270	
270000000		270	

Formula type required

Single row formula

Create a new column from a single row formula

Formula required


`DIVIDE(totalTransactionRevenue, 1000000)`

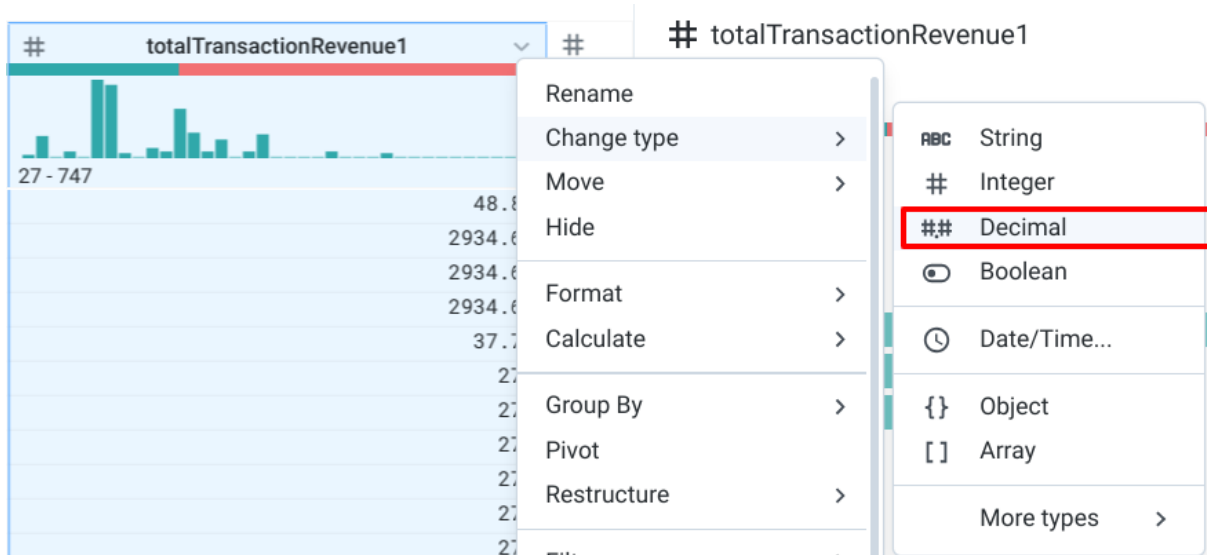
New column name

`totalTransactionRevenue1`

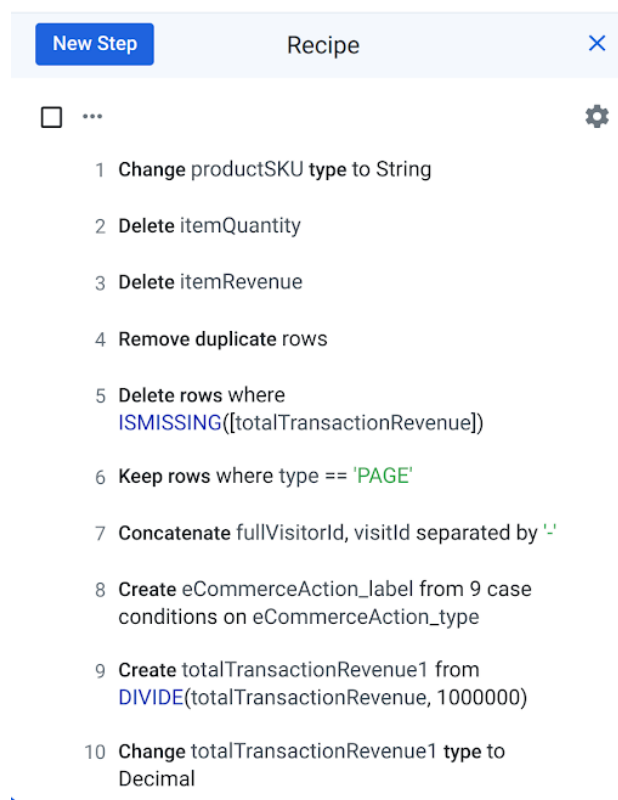
Cancel Add

3. Click **Add**.

4. To convert the new totalTransactionRevenue1 column's type to a decimal data type, open the menu to the right of the totalTransactionRevenue1 column by clicking , then click **Change type > Decimal**.



5. Review the full list of steps in your recipe:



6. You can now click **Run**.

Running Cloud Dataprep jobs to BigQuery

Challenge: Now that you are satisfied with the flow, it's time to execute the transformation recipe against your source dataset. The challenge for you is to load the output of the job into the BigQuery dataset that you created earlier. Make sure you load the output into a separate table and name it `revenue_reporting`.

Once your Cloud Dataprep job is completed, refresh your BigQuery page and confirm that the output table **revenue_reporting** exists.