Step-by-Step: Merge text strings to gain insights

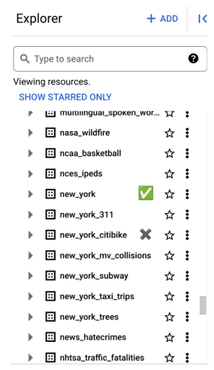
This reading outlines the steps the instructor performs in the following video, Merge text strings to gain insights. In the video, the instructor uses SQL's **`CONCAT`** function to combine strings from multiple columns to create a new column. Additionally, the instructor uses other SQL commands such as **`AVG`**, **`GROUP BY`**, and **`ORDER BY`** to gain insights about the new column.

Keep this step-by-step guide open as you watch the video. It can serve as a helpful reference tool if you need additional context or clarification while following the video steps. This is not a graded activity, but you can complete these steps to practice the skills demonstrated in the video.

What you'll need

If you would like to follow along with the instructor, you will need to log in to your BigQuery account to use the open (public) dataset called **`new_york`**. To access the dataset, make sure you have the **`bigquery-public-data`** project starred in your BigQuery Explorer pane. Then, scroll through the datasets in the **`bigquery-public-data`** project to find the **`new_york`** dataset. The table you will use is called **`citibike_trips`**. Select this table and then select the **Query** button.

Important note: BigQuery has two different databases that contain very similar information: **`new_york`** and **`new_york_citibike`**. Both of these databases contain tables called **`citibike_trips`**. However, these tables are not exactly the same between both databases. This step-by-step and the subsequent video use the **`new_york`** database. You will need to scroll to find this dataset under the **`bigquery_public_data`** project in the Explorer pane; it does not come up in a search.



Example: Use `CONCAT` on the bike sharing dataset

The **`CONCAT`** function can combine data from separate columns to provide new insights.

- In the BigQuery editor, enter **`SELECT`** and press Enter (Windows) or Return (Mac).
- Enter **`usertype,`** on line 2.
- On line 3, enter **`CONCAT(start_station_name," to ", end_station_name)`** to combine the names of the beginning and ending stations for each trip in a new column. This will create one column of routes.
- Enter **`AS route,`** at the end of line 3 to name the column route.
- On line 4, enter **`COUNT (*) as num_trips,`** to count the number of trips. The asterisk tells SQL to count the number of rows you're selecting. Each row represents a trip, so you can count all of the rows you've selected to count the number of trips.
- Next, calculate the average trip duration for each route. On line 5, enter:
  `ROUND(AVG(cast(tripduration as int64)/60),2) AS duration`

  This line of code accomplishes several tasks:

- It uses the **`CAST`** function to cast **`tripduration`** as an integer and divides that number by 60 to convert the number from seconds to minutes.
- It uses the **`AVG`** function to find the average duration of each route.
- It uses the **`ROUND`** function to round the output to 2 decimal places.
- It uses the **`AS`** command to give this output the alias **duration**.

Note 1: BigQuery stores numbers in a 64-bit memory system, which is why there's a 64 after integer in this case.

Note 2: While explaining this code, the instructor says "divide by the number of rows." Instead, they meant "divide by 60."

- Enter **`FROM`** on line 6 and press return.
- Enter **`` `bigquery-public-data.new_york.citibike_trips` ``** on line 7 (enclosed in back-ticks).
- Enter **`GROUP BY`** on line 8.
- Enter **`start_station_name, end_station_name, usertype`** on line 9.
- Enter **`ORDER BY`** on line 10 to tell SQL how to organize this data.
- Enter **`num_trips DESC`** on line 11 to sort it in descending order.
- Enter **`LIMIT 10`** on line 12.
- Your completed query should match the following code:

```
1   SELECT
2     usertype,
3     CONCAT (start_station_name," to ", end_station_name) AS route,
4     COUNT (*) as num_trips,
5     ROUND(AVG(cast(tripduration as int64)/60),2) AS duration
6   FROM
7     `bigquery-public-data.new_york.citibike_trips`
8   GROUP BY
9     start_station_name, end_station_name, usertype
10  ORDER BY
11    num_trips DESC
12  LIMIT 10
```

- Select **RUN** to view the results.

Now you can easily read these route names and trace them back to real places. You can also explore the types of customers taking each route. This type of information can help decision-makers at the bike-sharing company understand their user base in different parts of the city and where to keep more bikes for people to rent.

Go to next item **Completed**