Activity Exemplar: Create a target table in BigQuery

In this activity, you used BigQuery to create a target table to store data you pulled from a dataset of street tree information from San Francisco, California. In your BI role, you'll need to use programs such as BigQuery and Dataflow to move and analyze data with SQL. Now, you've practiced a key part of the Extraction stage of the BI pipeline: pulling data from a source and placing it into its own table.

The exemplar you're about to review will help you evaluate whether you completed the activity correctly. Because this activity involves copying, pasting, and executing a complete SQL query, you will just need to check that your result matches this exemplar.

If you find that the result you received is different from the exemplar provided, double check the formatting of the query you copied. Review the explanation of the SQL query in this activity to learn more about how the SQL query works and how to write your own in your projects.

Access the exemplar

To explore the query result exemplar, download the following attachment:

SanFranciscoTrees
CSV File

Assessment of exemplar

In this activity, you ran the following SQL query to create a target table:

```sql
SELECT
    address,
    COUNT(address) AS number_of_trees
FROM
    `bigquery-public-data.san_francisco_trees.street_trees`
WHERE
    address != "null"
GROUP BY address
ORDER BY number_of_trees DESC
LIMIT 10;
```

The SELECT clause selects the address of each tree. By using the COUNT function, you count the number of trees at each address and return a single row of data per address, instead of per tree. This data is saved as a new column.

The FROM clause is straightforward as it specifies the street_trees table within the San Francisco Street Trees dataset.

The WHERE clause is necessary to ensure that your target table only includes rows that have a value in the address column.

The GROUP BY clause specifies that you're grouping data by the address, and the ORDER BY clause sorts the data in descending order by number_of_trees column.

The LIMIT clause limits the query to return only the top ten rows of data. When working with large datasets, including a limit will decrease the processing time required to return the data.

If you need a refresher on SQL code, review some resources from the Google Data Analytics Certificate: Review Google Data Analytics Certificate content about SQL and Review Google Data Analytics Certificate content about SQL best practices.

The result of this query is a target table with two columns. It features the address column, as well as the total number of trees planted at the address you calculated in the SELECT clause. If properly executed, the first value in the address column is 100x Cargo Way. Next to it, the number_of_trees is 135. If you didn't receive this result, please review the code and run it again.

Furthermore, the target table shows the 10 addresses with the most trees planted by the Department of Public Works in the city of San Francisco:

100x Cargo Way

700 Junipero Serra Blvd

1000 San Jose Ave

1200 Sunset Blvd

1600 Sunset Blvd

2301 Sunset Blvd

1501 Sunset Blvd

2401 Sunset Blvd

100 STAIRWAY5

2601 Sunset blvd

And the number of trees for each address is as follows:

100x Cargo Way: 135

700 Junipero Serra Blvd: 125

1000 San Jose Ave: 113

1200 Sunset Blvd: 110

1600 Sunset Blvd: 102

2301 Sunset Blvd: 94

1501 Sunset Blvd: 93

2401 Sunset Blvd: 92

100 STAIRWAY5: 87

2601 Sunset Blvd: 84

Key takeaways

Target tables are the destination for data during the Extraction stage of a pipeline. You'll use them in your role as a BI professional to store data after pulling it from their sources. Once they're in a target table, you can transform them with BigQuery or Dataflow and load them into reporting tables. You'll learn about the Transform and Load stages of data pipelines later in this course.

Go to next item ✓Completed