**Driving Insights:   
A Comprehensive Analysis of Taxi Operations**

**Team Members:**

*Manoj Velu*

*Nivethitha Avarampalayam Manoharan*

*Varshini Vaisnavi Srinivasan*

1. **Introduction**

**1.1 Objectives:**

* + Understand Demand Patterns and Identify Peak Hours.
  + Analyze Cumulative Revenue Over Time.
  + Explore RatecodeID and Average Payments.
  + Understand Customer Payment Preferences.

**1.2 Impact:**

These objectives drive strategic decisions for taxi services, optimizing deployment, rate structures, and customer interactions. The analysis is the compass guiding taxis toward enhanced efficiency and a more customer-centric service.

1. **Data Overview**

|  |  |
| --- | --- |
| **Field Name** | **Description** |
| VendorID | A code indicating the TPEP provider that provided the record.  1= Creative Mobile Technologies, LLC; 2= VeriFone Inc. |
| tpep\_pickup\_datetime | The date and time when the meter was engaged. |
| tpep\_dropoff\_datetime | The date and time when the meter was disengaged. |
| Passenger\_count | The number of passengers in the vehicle.  This is a driver-entered value |
| Trip\_distance | The elapsed trip distance in miles reported by the taximeter. |
| PULocationID | TLC Taxi Zone in which the taximeter was engaged |
| DOLocationID | TLC Taxi Zone in which the taximeter was disengaged |
| RateCodeID | The final rate code in effect at the end of the trip.  1= Standard rate 2=JFK 3=Newark 4=Nassau or Westchester 5=Negotiated fare 6=Group ride |
| Store\_and\_fwd\_flag | This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka “store and forward,” because the vehicle did not have a connection to the server. Y= store and forward trip N= not a store and forward trip |
| Payment\_type | A numeric code signifying how the passenger paid for the trip. 1= Credit card 2= Cash 3= No charge 4= Dispute |
| Fare\_amount | The time-and-distance fare calculated by the meter |
| Extra | Miscellaneous extras and surcharges. Currently, this only includes the $0.50 and $1 rush hour and overnight charges. |
| MTA\_tax | $0.50 MTA tax that is automatically triggered based on the metered rate in use |
| Improvement\_surcharge | $0.30 improvement surcharge assessed trips at the flag drop. The improvement surcharge began being levied in 2015. |
| Tip\_amount | Tip amount – This field is automatically populated for credit card tips. Cash tips are not included. |
| Tolls\_amount | Total amount of all tolls paid in trip. |
| Total\_amount | The total amount charged to passengers. Does not include cash tips. |

1. **GCP Setup:**

Fig : Created the GCP Account

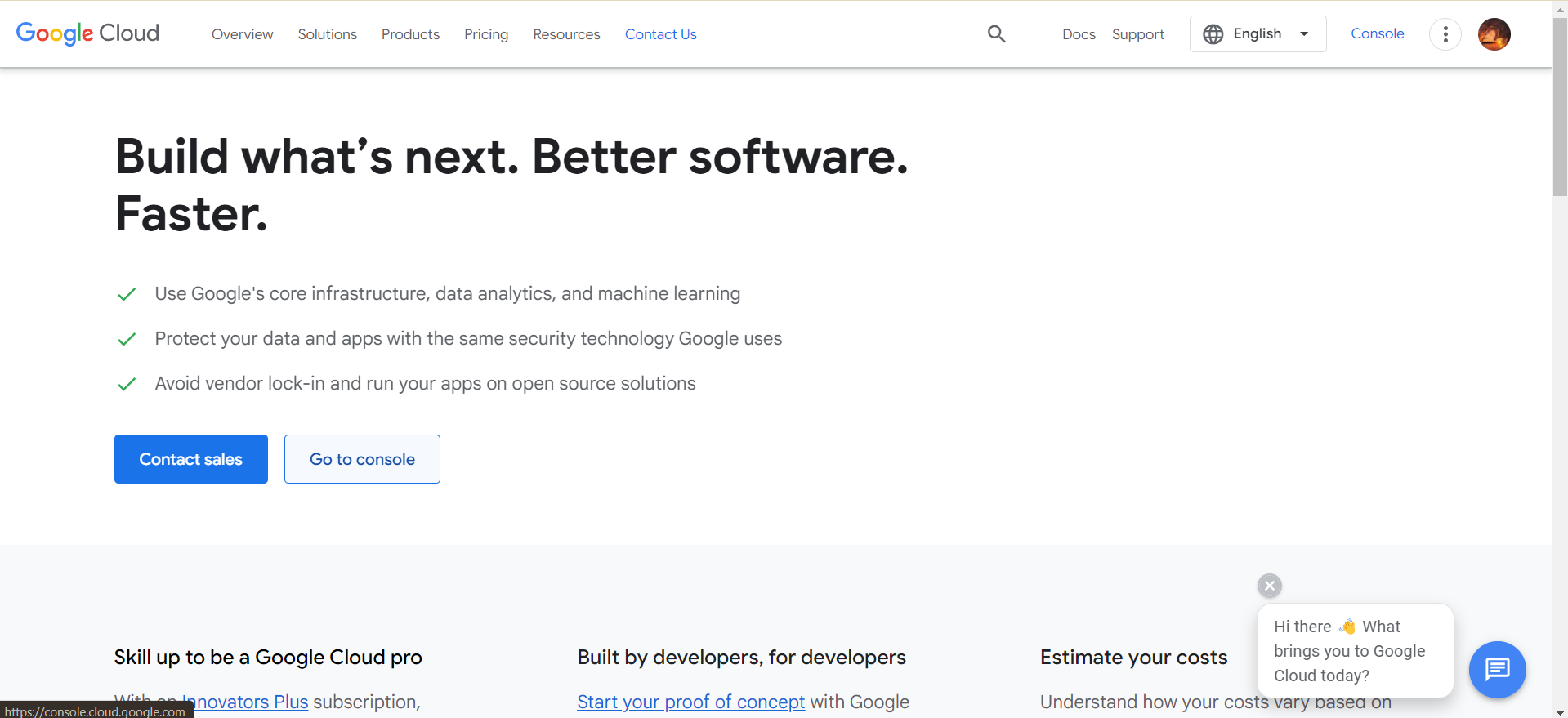


Fig : Opened BigQuery Console

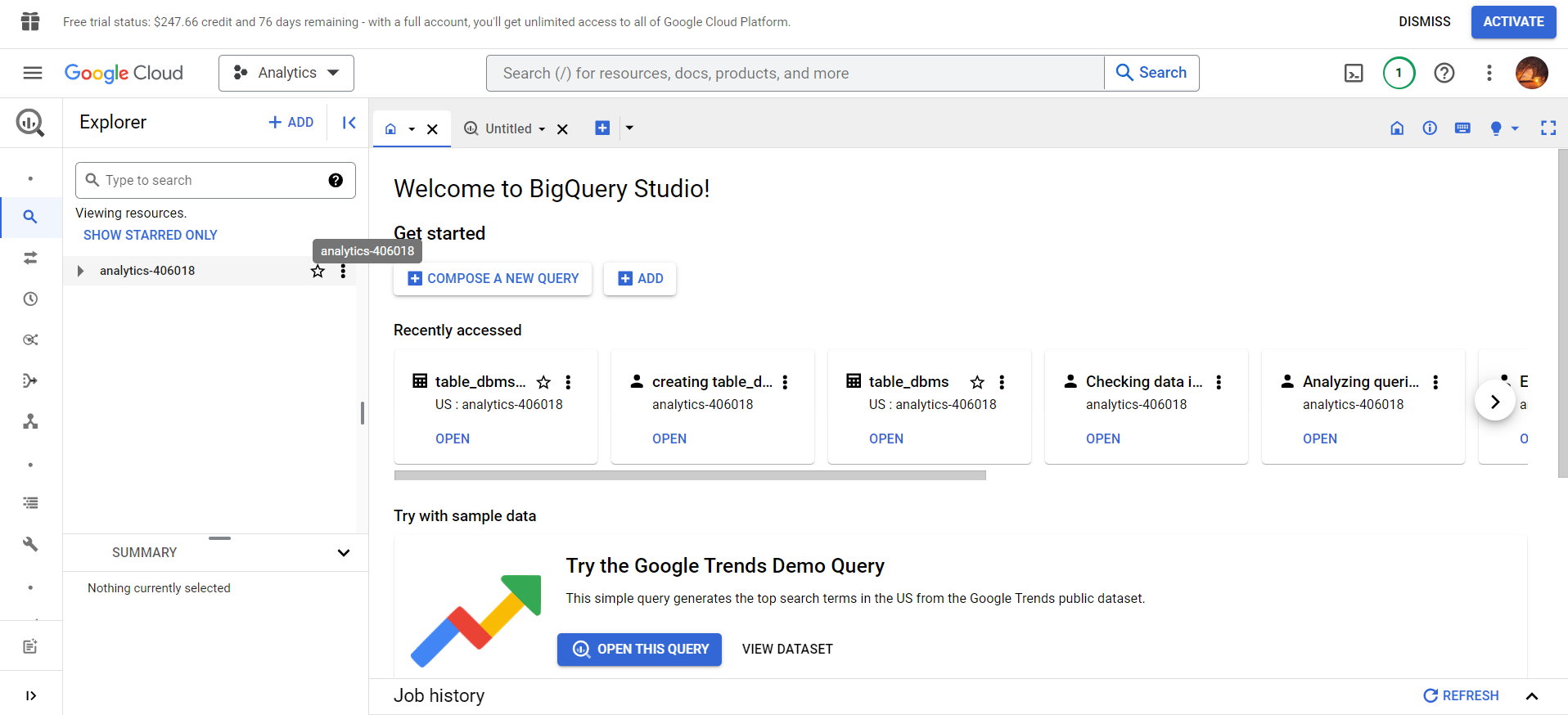


Fig : Created a Project Name called ‘Analytics’

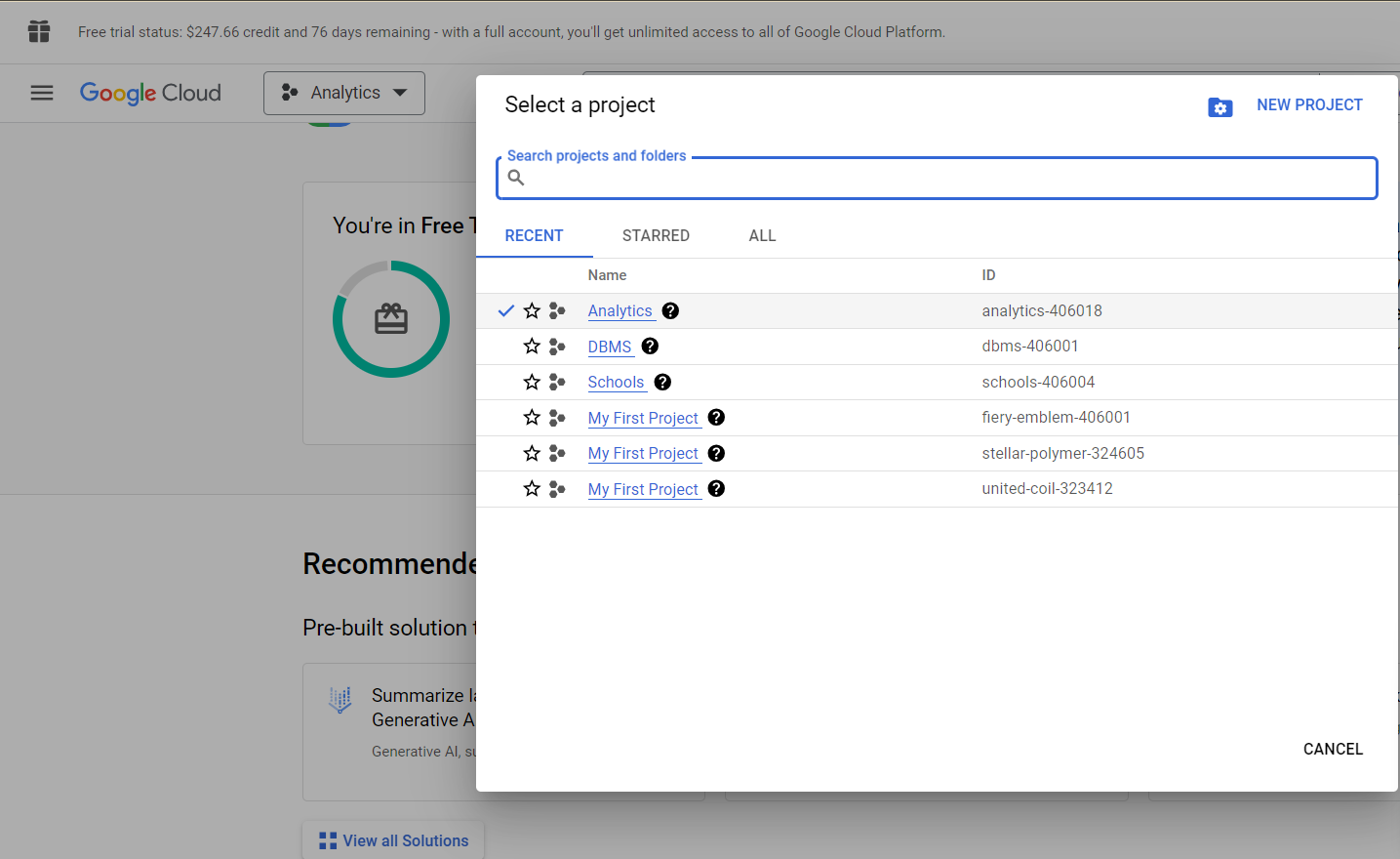


Fig : Created a dataset called ‘data’

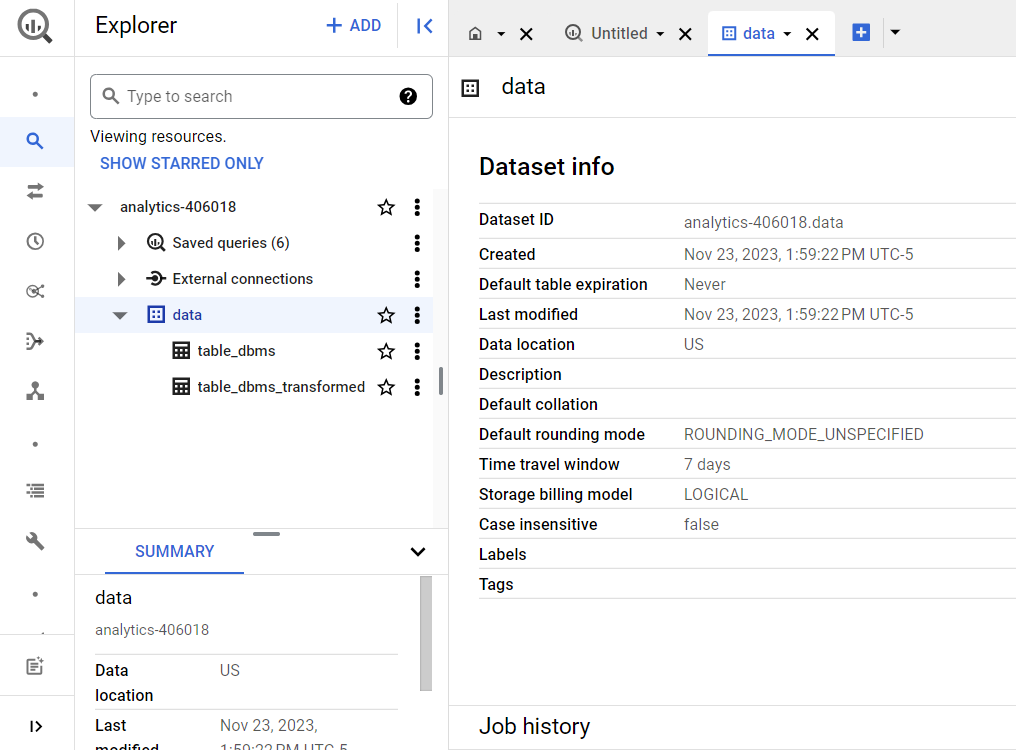


Fig : Created an empty table called ‘table\_dbms’

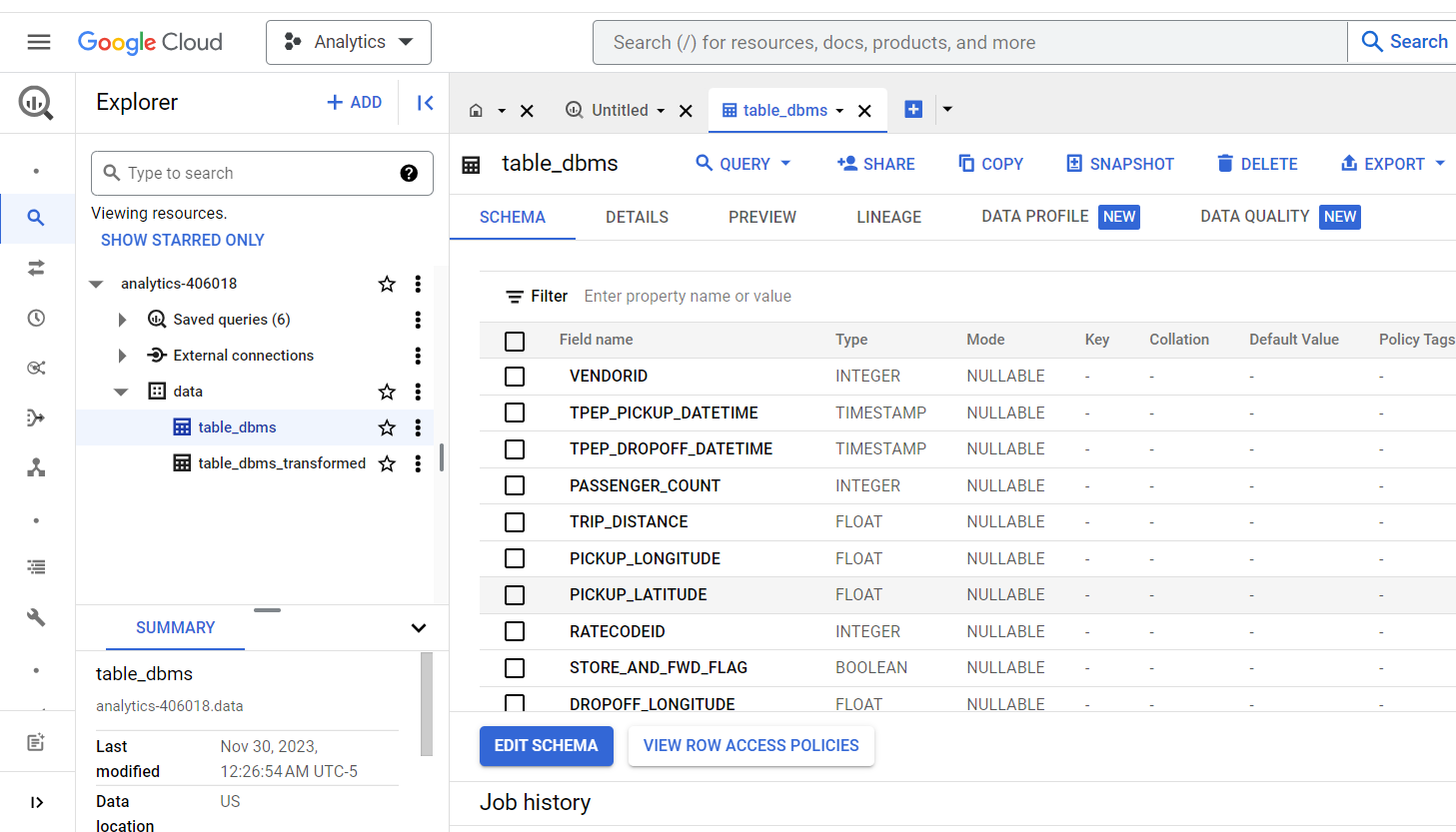
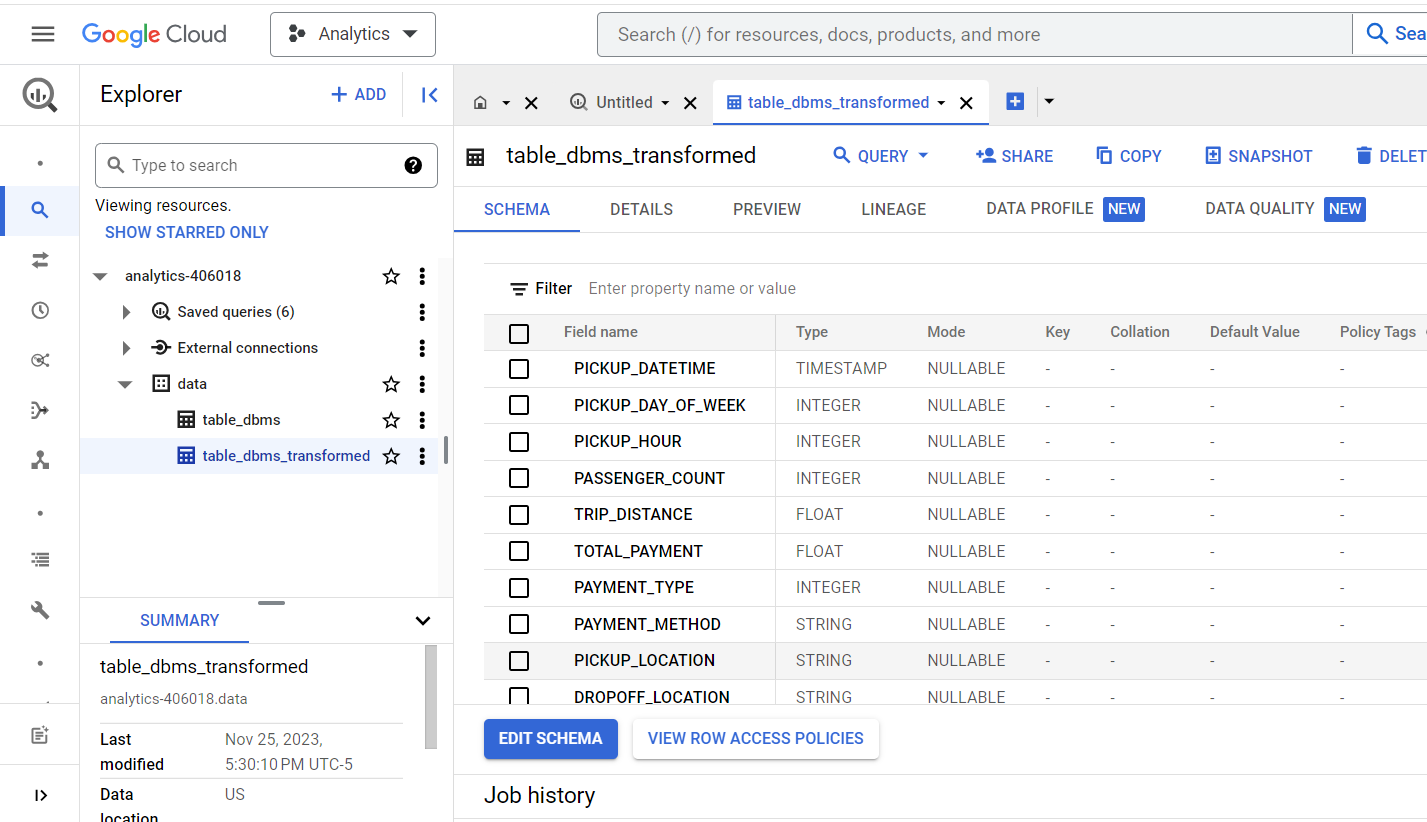


Fig: Created a table called ‘table\_dbms\_transformed’



1. **ETL**

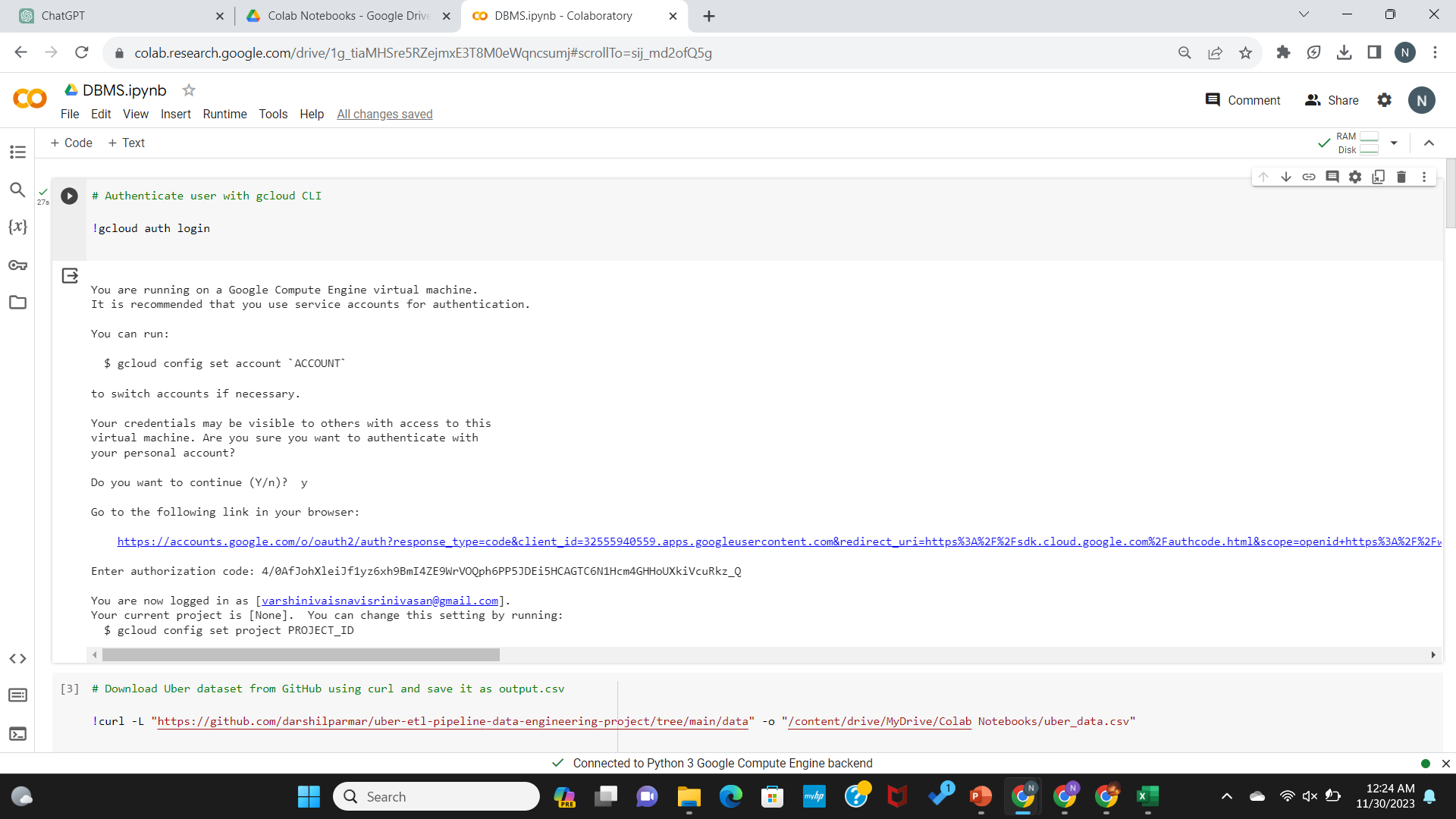
**Step 1:** Authenticate User

- log in securely to GCP using the Google Cloud CLI to ensures proper access to GCP resources.

**Step 2:** Download Uber Dataset

- The dataset is fetched from a GitHub repository using 'curl' command to download and save the dataset.

Fig : Authenticate User and Download Uber Dataset



**Step 3:** List Files in Google Drive

-used 'ls' command for checking if the downloaded file is present in a specified Google Drive location.

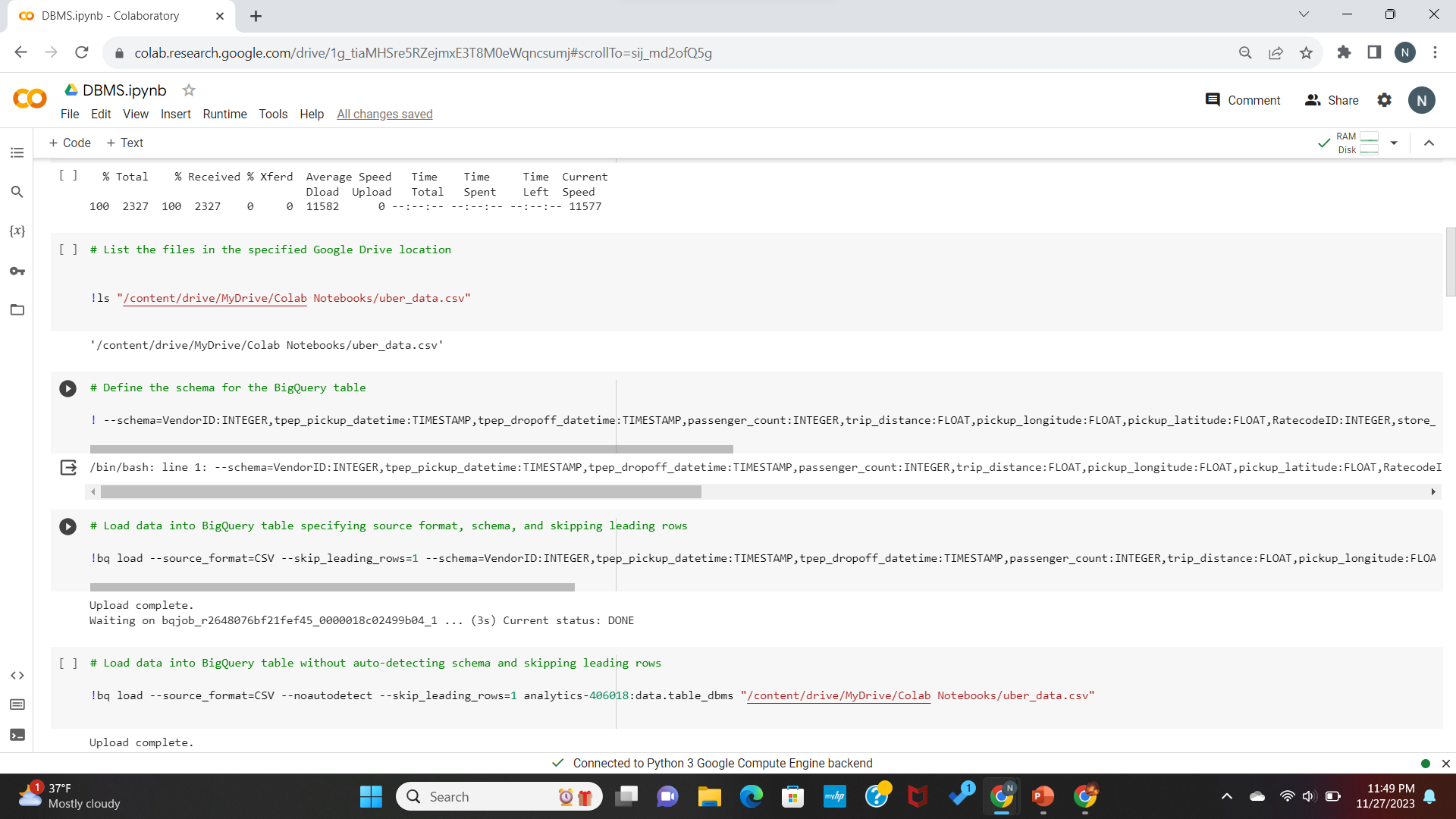
**Step 4:** Define Schema for Big Query

- The schema, or structure, for the Big Query table is defined here. It outlines the types of data each column will contain.

**Step 5:** Load Data into Big Query

- Loaded the dataset into Big Query and the Auto-detection of the schema is used, and leading rows are skipped.

Fig : List Files in Google Drive

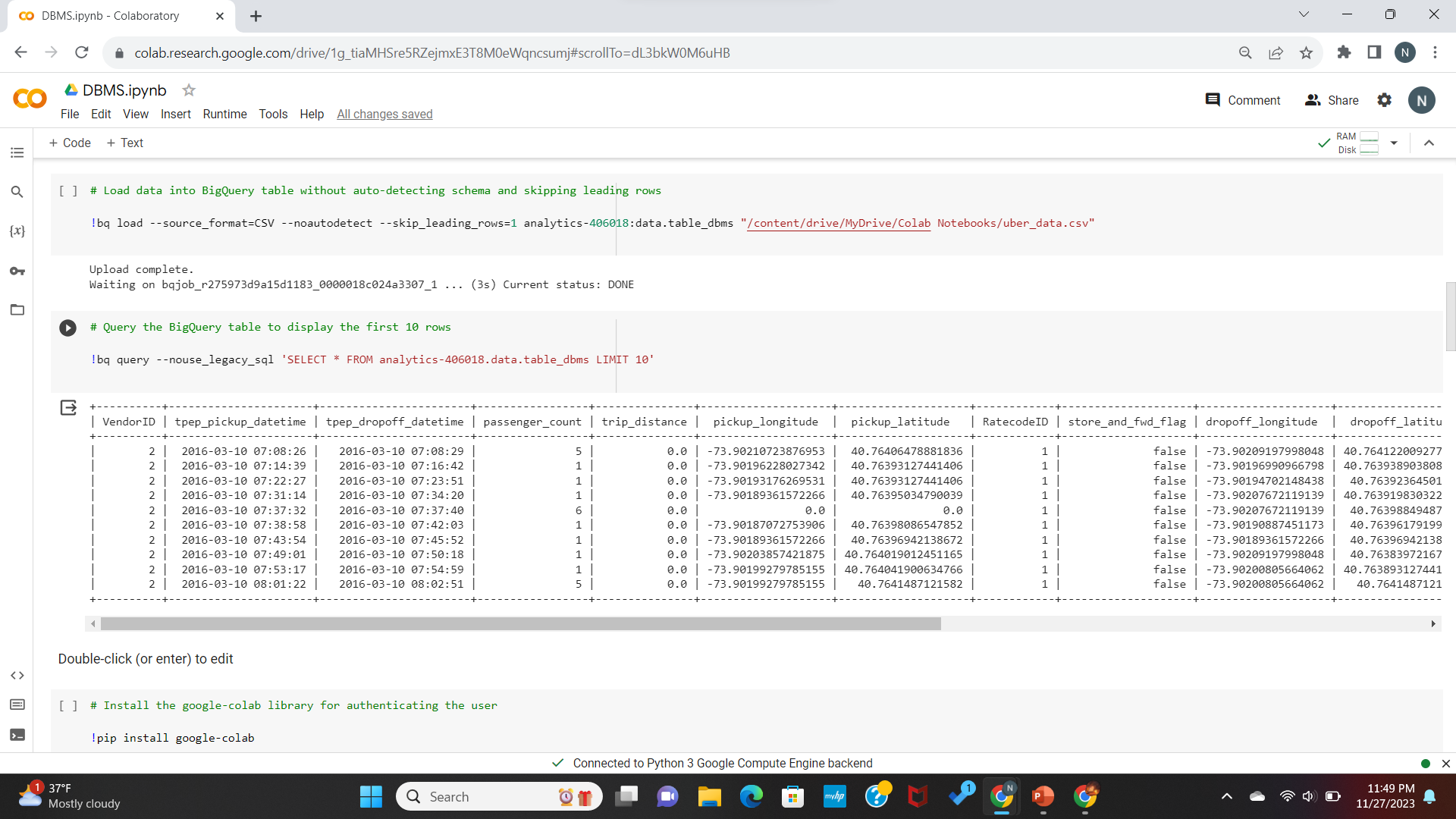


**Step 6:** Load Data into Big Query (No Auto-detection)

- Explored alternative method where the schema is not auto-detected, but a predefined schema is used and loaded the data without auto-detecting the schema.

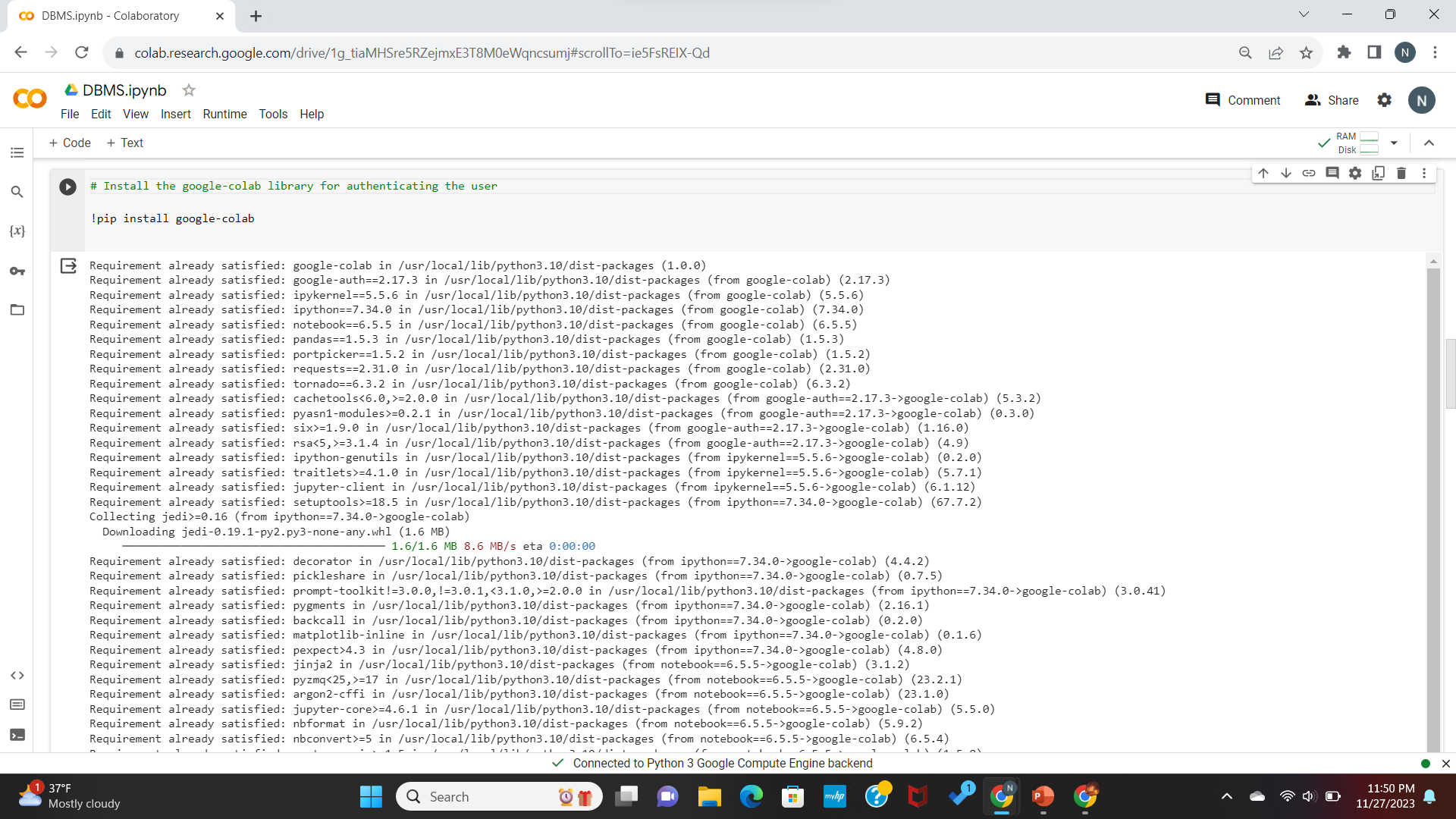
**Step 7:** Query Big Query Table

- Querying the Big Query table to display the first 10 rows of data and provided a glimpse of the data being processed.



**Step 8:** Authenticate User with Google Colab

- Installed 'google-colab' and authenticating the user.



**Step 9:** Set GCP Project ID

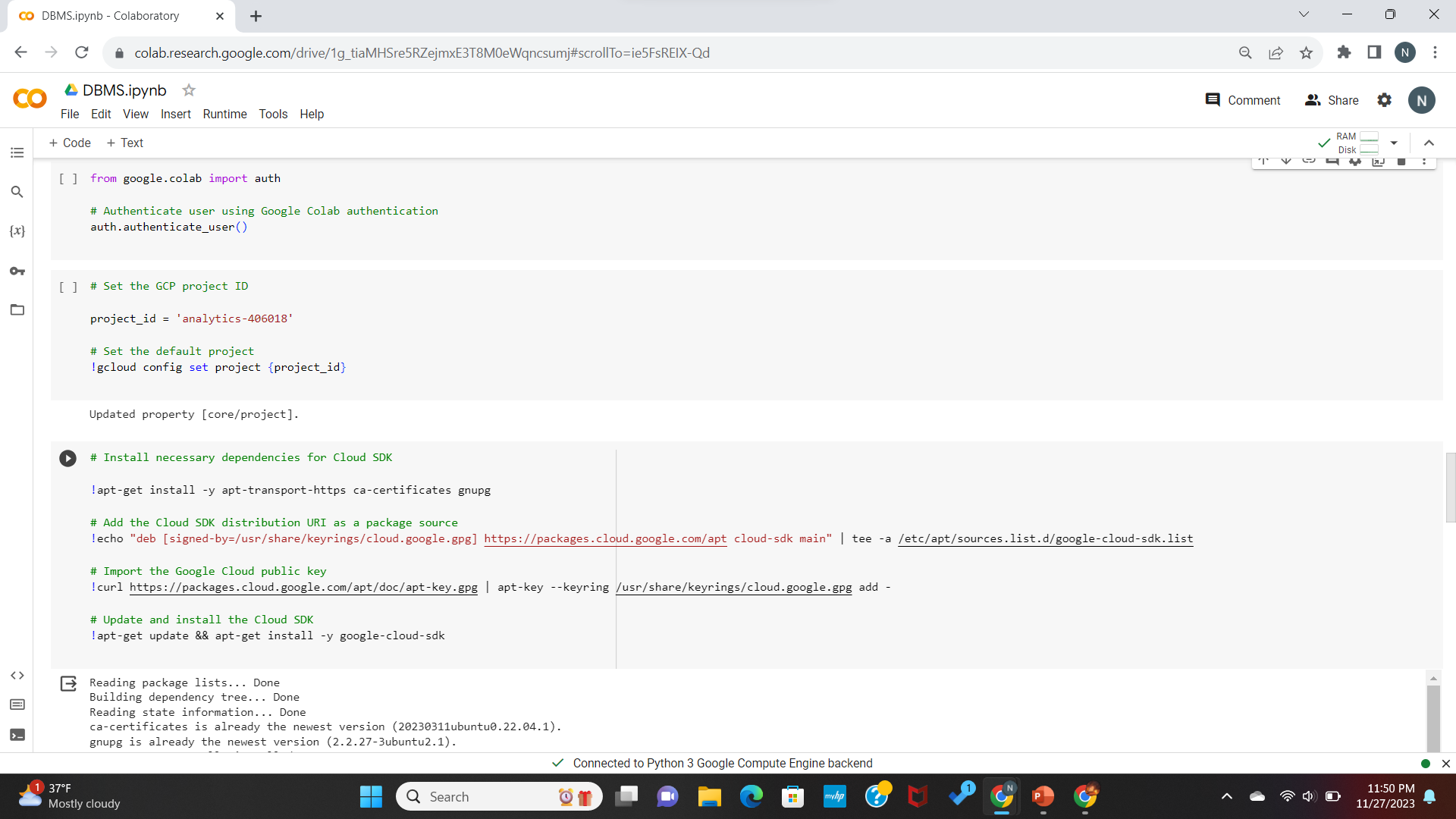
- The 'gcloud config set project' command is used for setting the GCP project ID for further operations.

**Step 10:** Install Dependencies for Cloud SDK

- Installed necessary dependencies for the Google Cloud SDK and Prepared the environment for GCP operations.

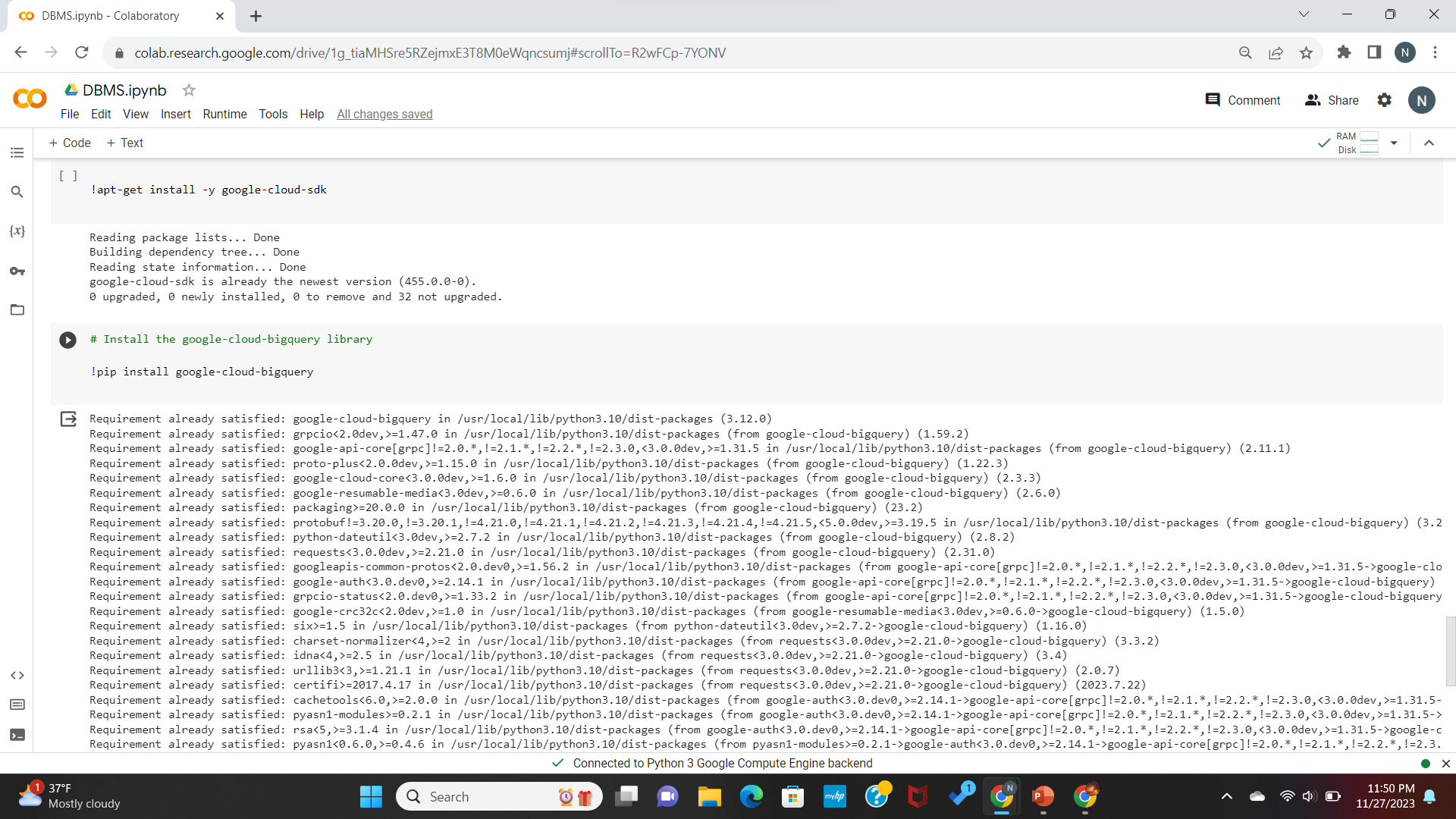
**Step 11:** Install Cloud SDK

- Updating and installing the Cloud SDK.



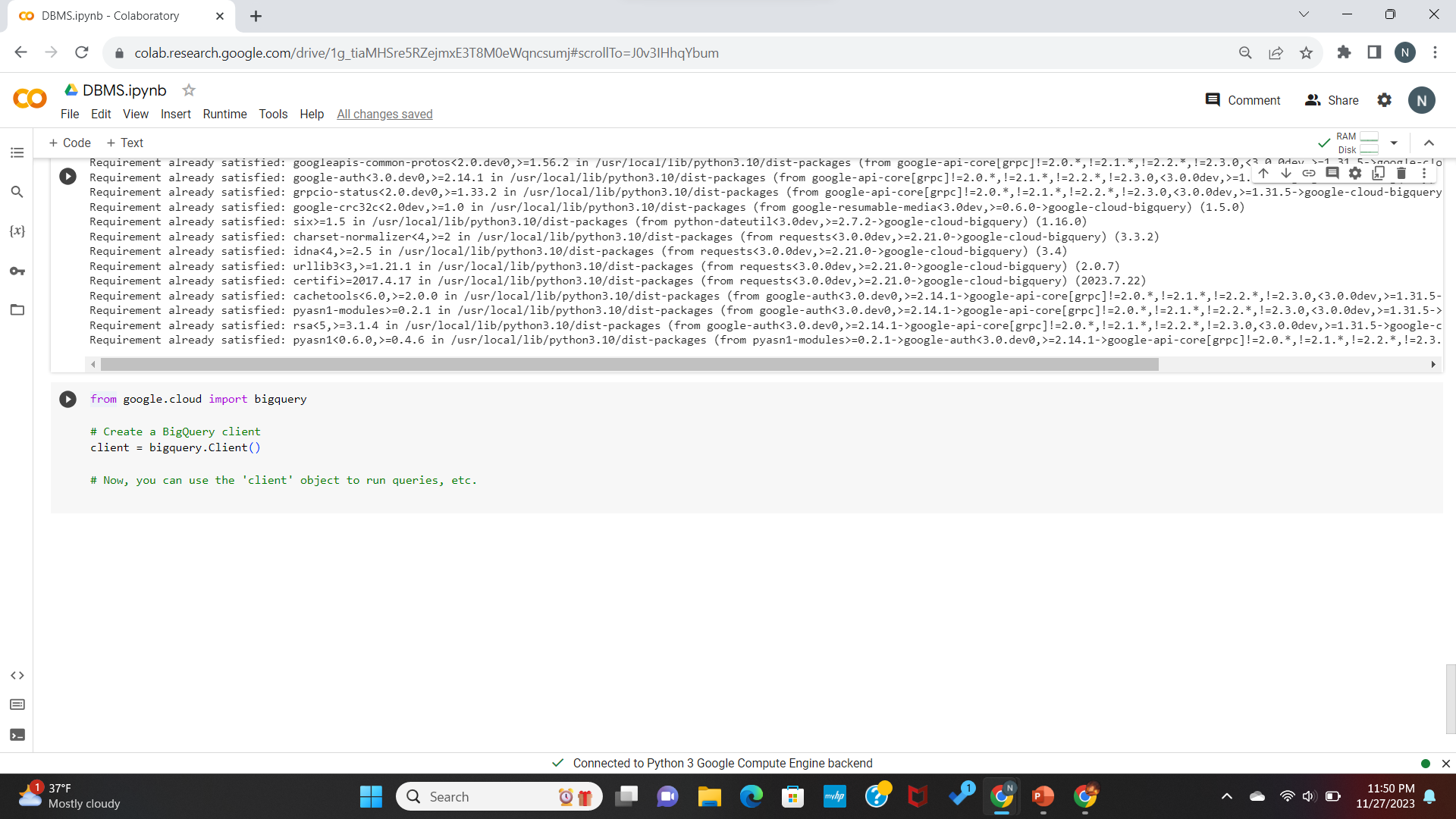
**Step 12:** Install Big Query Library

* Installed Google Cloud Big Query library for Python and Prepared the environment for running Big Query queries.



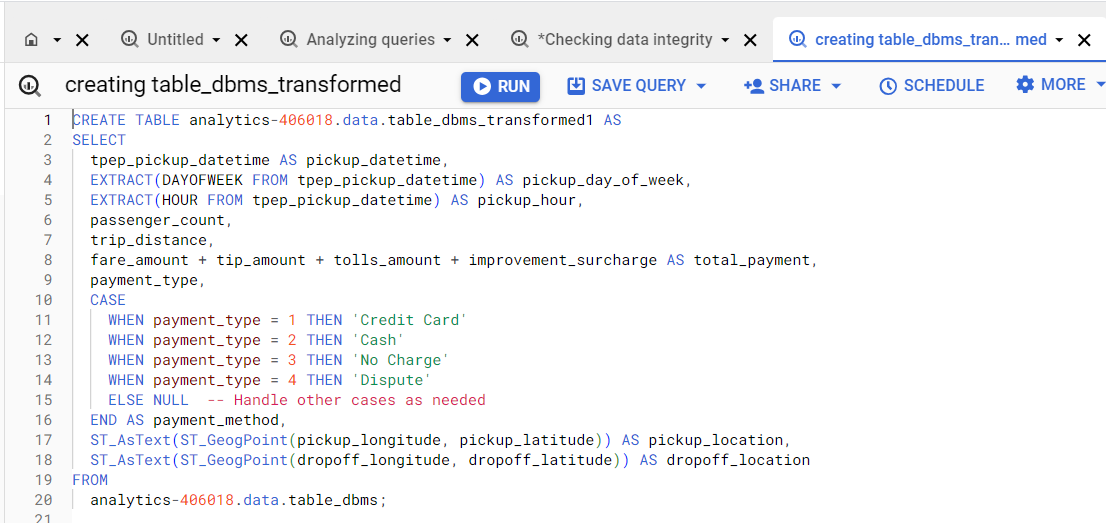
**Step 13:** Create Big Query Client

- Created a Big Query client object for interacting with Big Query services and the client object is now ready for running queries.

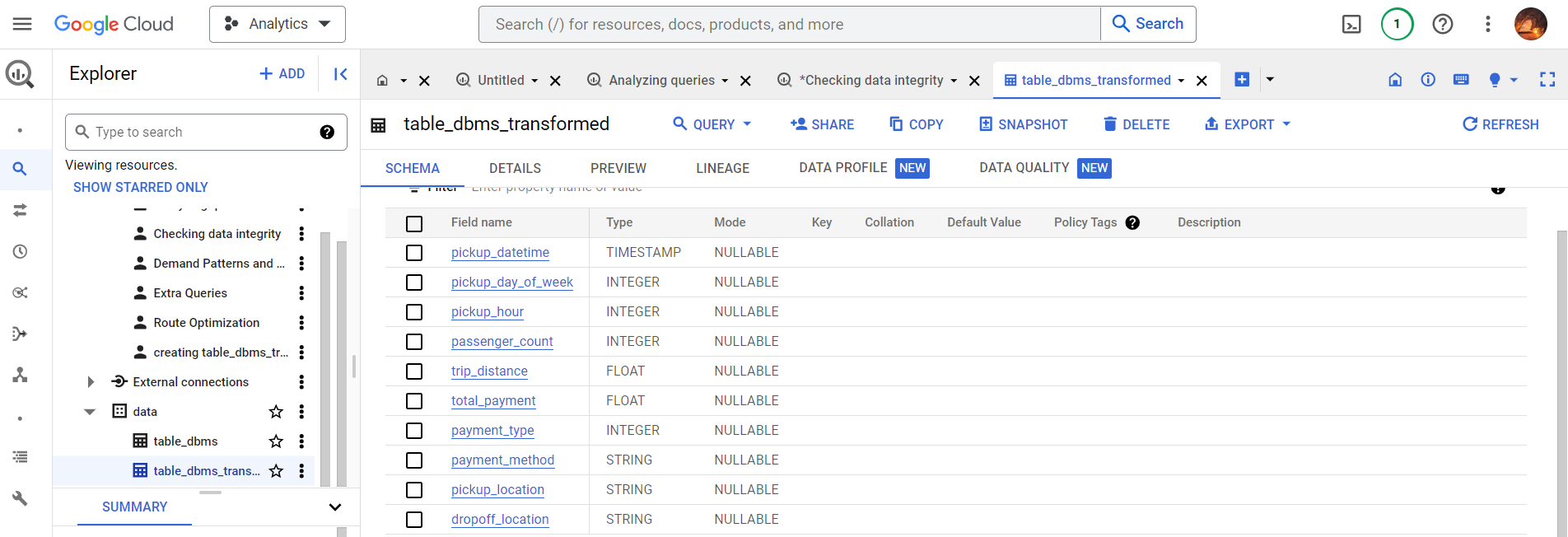


1. **Data Transformation**

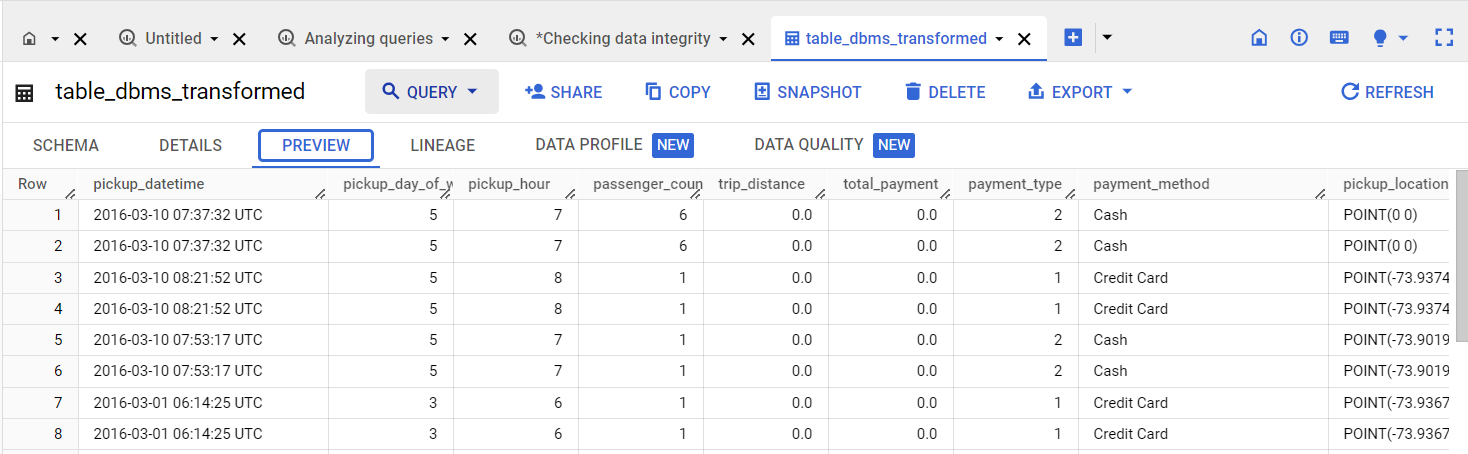
Performed data transformations using SQL on the extracted data. Creation of a new table in Big Query that stores the transformed data. The transformations were meaningful and prepared the data for insightful analysis.



Table\_dbms\_transformed schema



Table\_dbms\_transformed



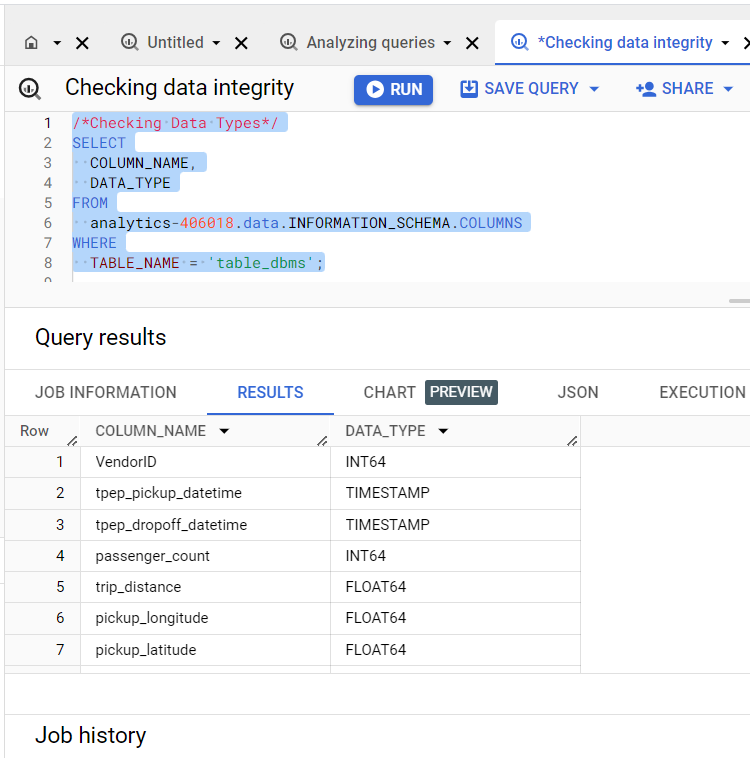
1. **Data Validation**

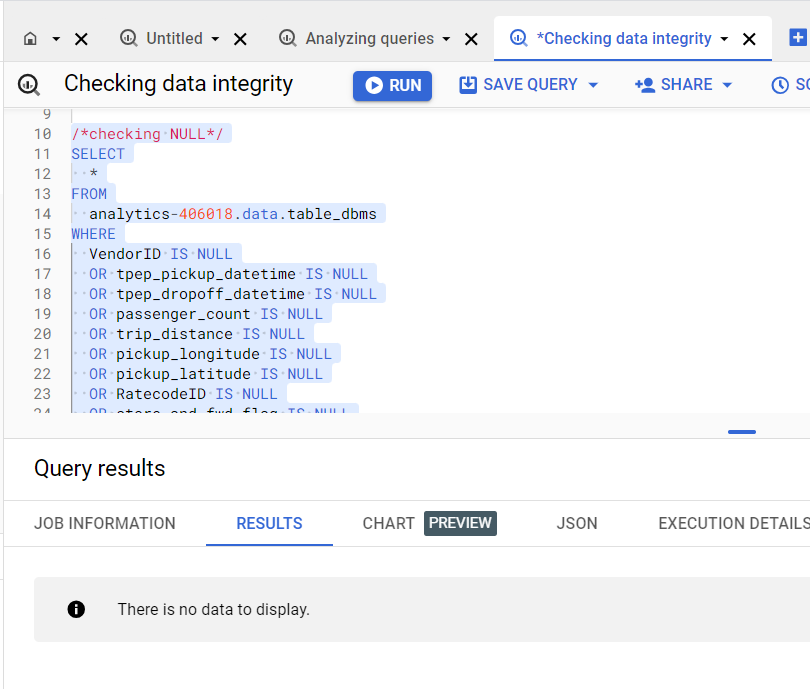
**Null Values:**

* + Checked for null values in columns.
  + No null values found in the dataset.

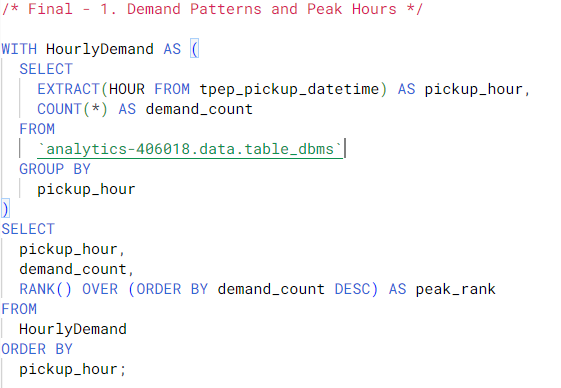
**Data Types:**

* + Verified data types for columns.
  + Ensured that data types align with expectations.

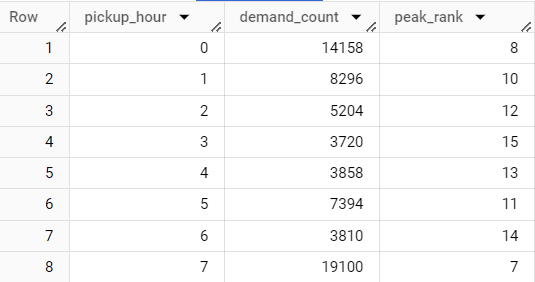




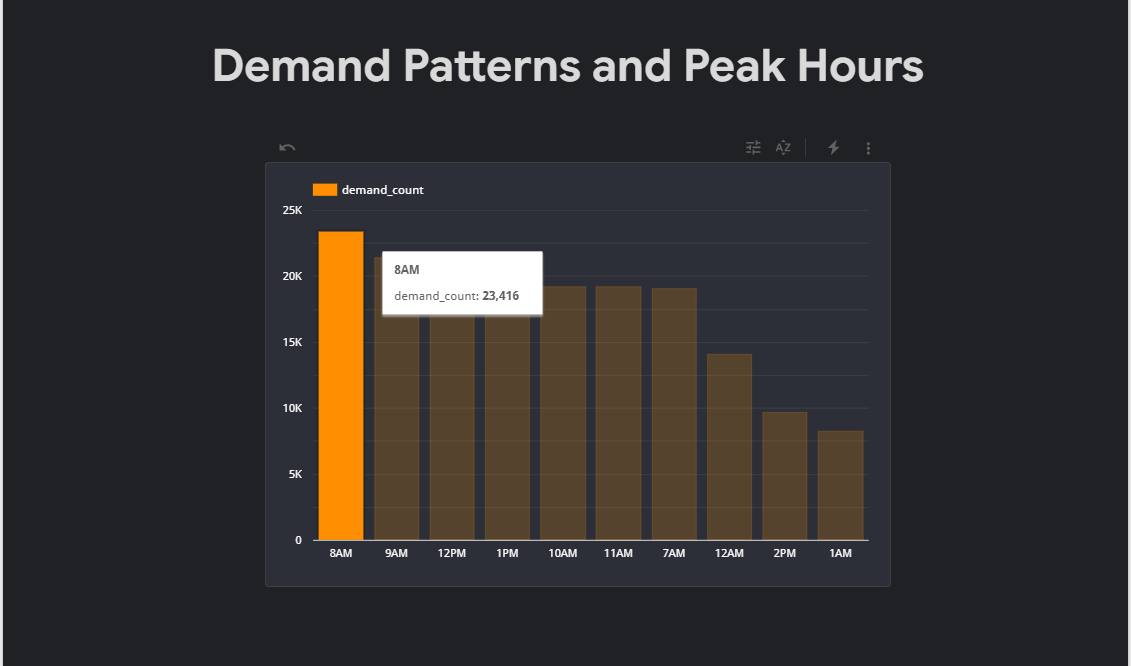
1. **Data Analysis and Insights**
   1. **Demand Pattern and Peak Hours:**



**Query Output:**

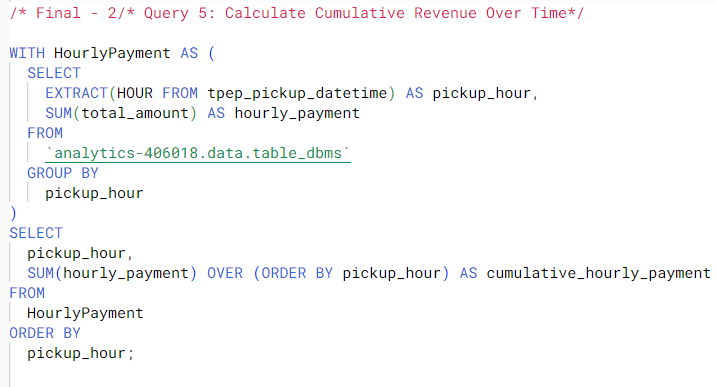


**Looker Studio Visualization**

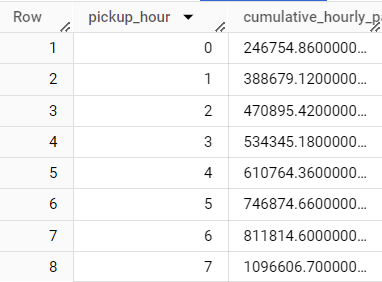


**7.2 Cumulative Revenue Over Time:**

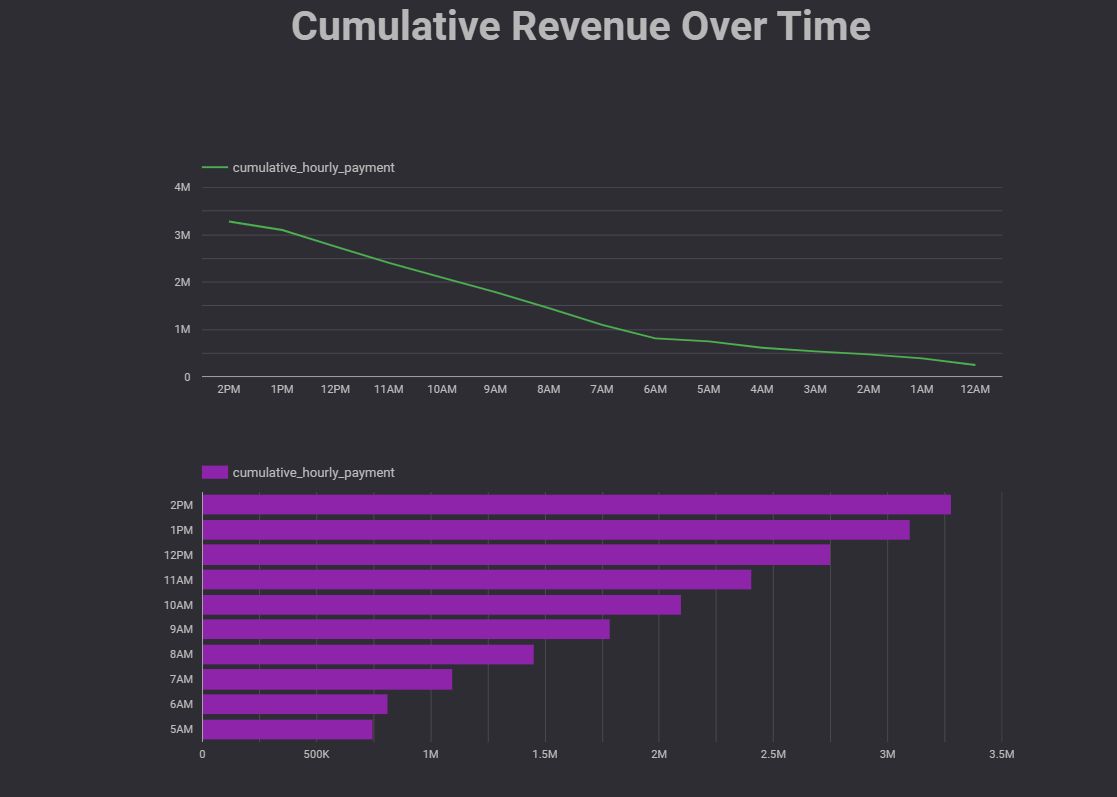
**Query:**



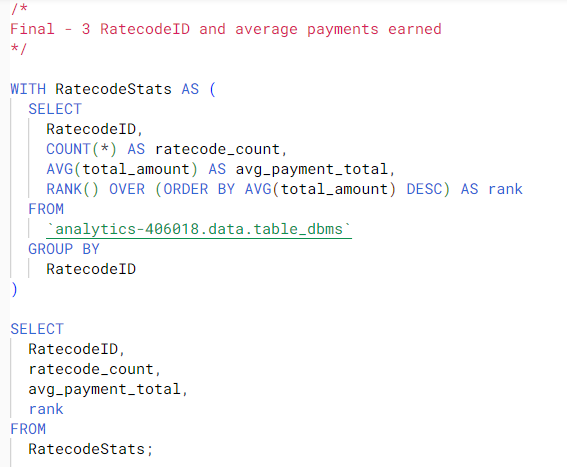
**Query Output:**



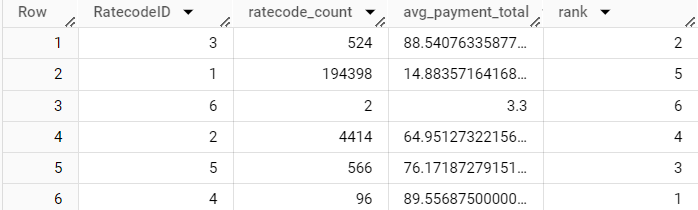
**Looker Studio Visualization:**



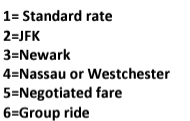
* 1. **Rate code Statistics:**



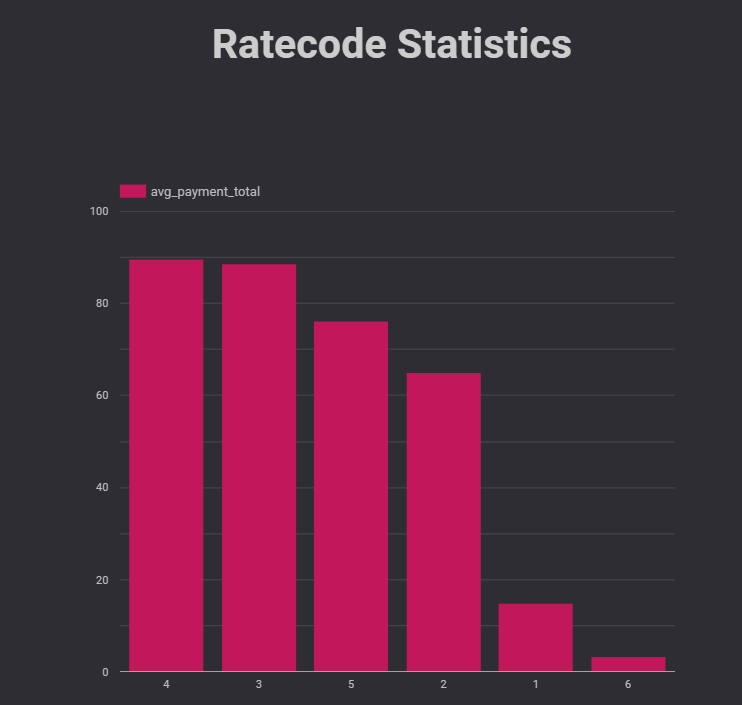
**Output of Query**



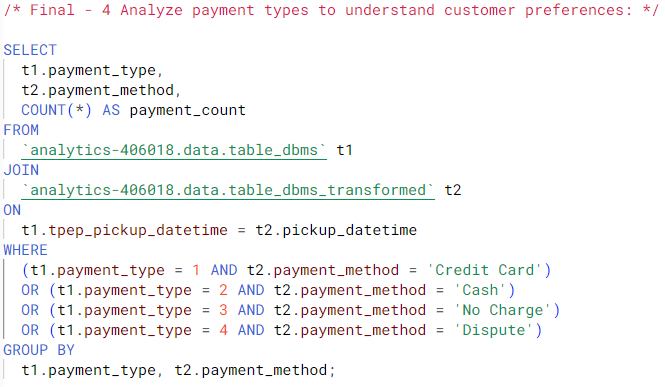
**Rate Codes:**



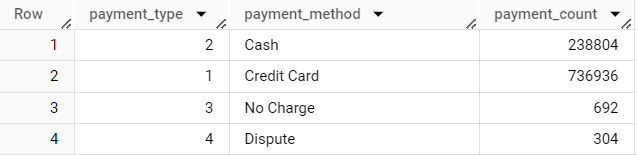
**Looker Studio Visualization**



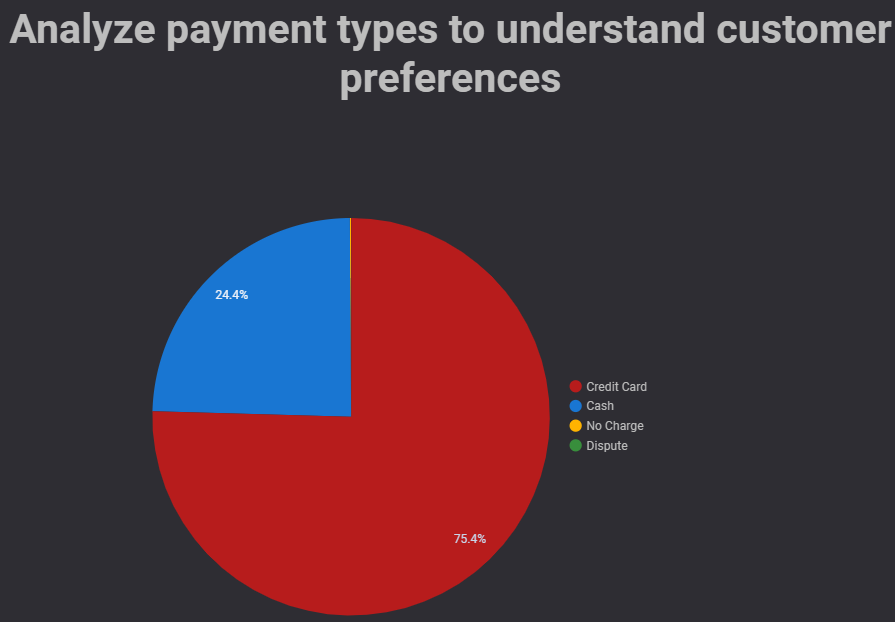
* 1. **Payment Preference:**



**Output of the Query:**



**Looker Studio Visualization**



1. **Recommendations**

* Fleet Optimization for Peak Demand based on Visualizations
* Capitalizing on Business Hour Opportunities I.e. Peak Hours
* Strategic Resource Allocation based on Insights
* Targeted Marketing for High-Average Rate Codes to drive the profit margins upwards
* Optimization of Low-Performing Rate Codes to gain more users
* Enhancing Digital Payment Infrastructure to ensure smooth experience
* Encouraging Cash and Card Flexibility
* Reviewing No Charge Instances
* Effective Dispute Resolution for improved trust

1. **Conclusion**

* We have extracted the dataset and leveraged GCP and its tools to run queries that give us a bigger picture about the dataset.
* This analysis provides some key insights for taxi services to excel in operational demands, revenue generation, and customer satisfaction.
* Leveraging these insights can ensure a thriving future in the competitive transportation industry by exceeding customer expectations.

1. **Reference**

* <https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page>
* <https://www.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_yellow.pdf>
* [https://github.com/darshilparmar/uber-etl-pipeline-data-engineering-project/blob/main/README.md](https://d.docs.live.net/aba215a0a19aeaf3/Documents/Taxi%5b1%5d.pptx)