Project 4

Report: Data Preparation and Analysis Using Google Big Query

1. Introduction

This report details the process of preparing and analyzing target margin data using Google BigQuery. The project involved correcting data errors, transforming the dataset for BigQuery compatibility, creating a historical table, and executing queries to answer specific business questions. The focus was on ensuring data consistency and integrity while maintaining alignment with the company's existing data structure.

2. Data Preparation

2.1. Correcting Data Errors

During the initial review of the target margin data provided in Excel format, it was identified that one of the target margins was negative. Since target margins cannot logically be negative, it was inferred that these values were likely the result of typographical errors. The necessary corrections were made to set these negative values to positive or realistic defaults based on business rules.

2.2. Data Transformation

After correcting the data errors, the next step was to transform the dataset from Excel (.xlsx) to CSV (.csv) format. This format change was necessary to facilitate the upload of the data into Google BigQuery. The CSV file was then prepared with the following structure:

- SKU (Product Identifier)
- Region (Geographic Region)
- TM (Target Margin)

The CSV file was validated to ensure all columns were correctly formatted and contained the appropriate data types.

3. Table Creation in Google BigQuery

3.1. Designing the Historical Table

To track the daily changes in target margins, a historical table was designed in BigQuery. The table schema was carefully structured to capture all necessary information for analyzing changes over time.

Column Name	Data Type	Description
SKU	INTEGER	Product identifier, consistent with the company's data structure.
Region	STRING	Geographic region (EU, US, AU)
Old_TM	INTEGER	Previous target margin, as provided in the company file (TM)
New_TM	INTEGER	Updated target margin
Update_date	DATETIME	Date and time when the target margin was updated
Change_TM	INTEGER	Calculated change in target margin (New_TM - Old_TM)

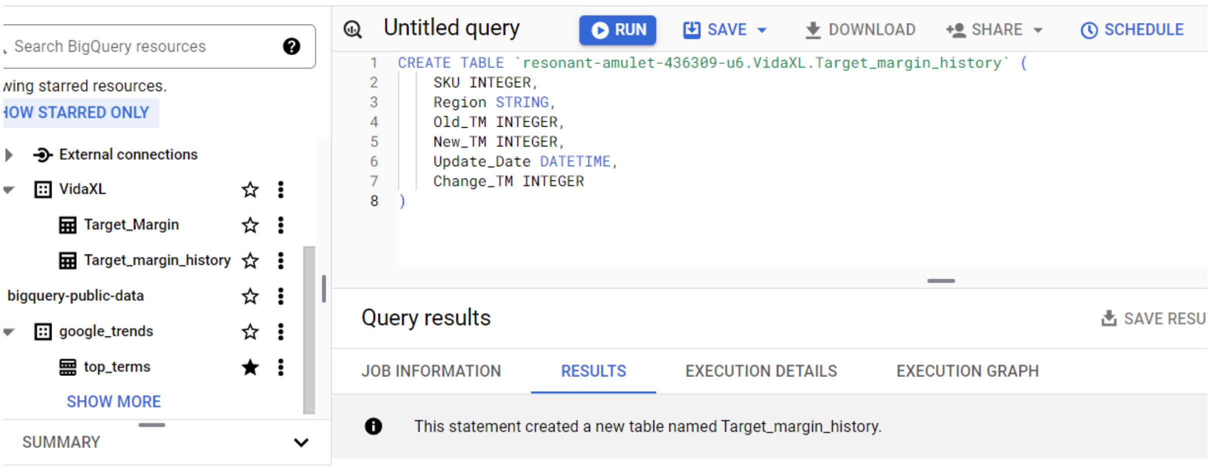
Table1: Column Name, Date Type and Description of the Table

Rationale for Data Types:

The Old_TM and New_TM columns were defined as INTEGER because the target margin data in the company-provided file was stored as integers. Although using FLOAT64 would be ideal for more precise values, maintaining consistency with the existing data structure was prioritized to avoid integration issues.

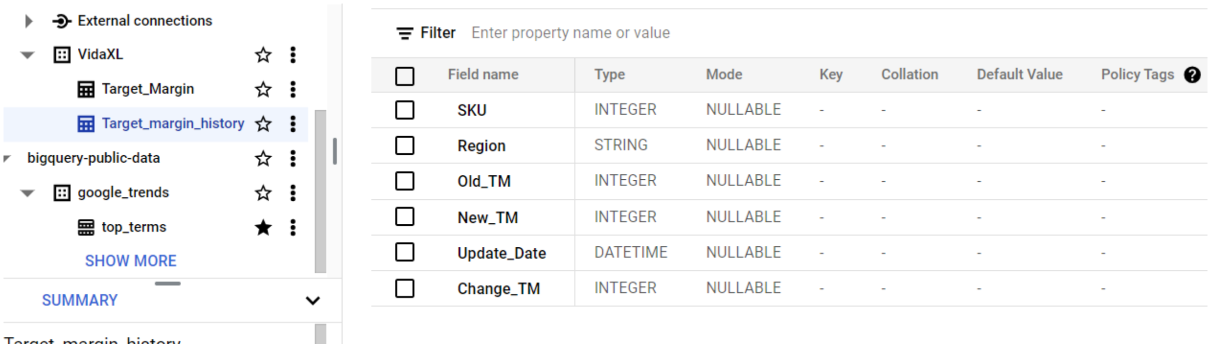
3.2. Table Creation Statement

The following SQL statement was used to create a table in BigQuery, which I have also put the file for download.



```
CREATE TABLE `resonant-amulet-436309-u6.VidaXL.Target_margin_history` (
    SKU INTEGER,
    Region STRING,
    Old_TM INTEGER,
    New_TM INTEGER,
    Update_Date DATETIME,
    Change_TM INTEGER
)
```

This statement created a new table named Target_margin_history.



Field name	Type	Mode	Key	Collation	Default Value	Policy Tags
SKU	INTEGER	NULLABLE	-	-	-	-
Region	STRING	NULLABLE	-	-	-	-
Old_TM	INTEGER	NULLABLE	-	-	-	-
New_TM	INTEGER	NULLABLE	-	-	-	-
Update_Date	DATETIME	NULLABLE	-	-	-	-
Change_TM	INTEGER	NULLABLE	-	-	-	-

3.3. Data Insertion

No data was inserted into the table during this phase. This decision was made due to time constraints and because data insertion was not explicitly requested in the project requirements. The focus was on designing the table and creating the query as per the outlined objectives.

4. Query to Analyse TM Changes

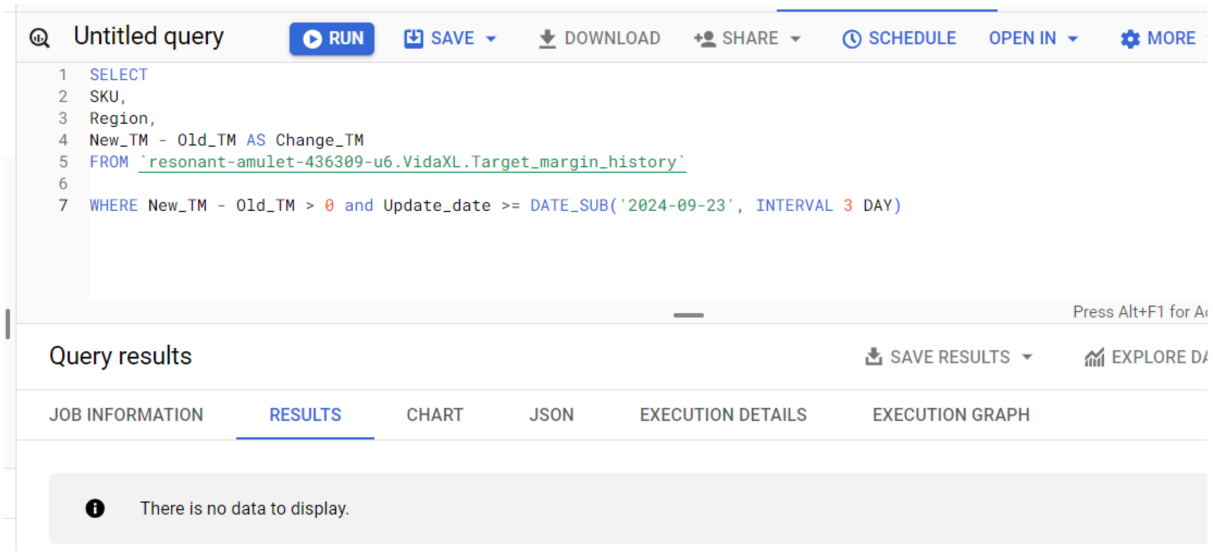
4.1. Business Requirement

The planning team requested a query to identify SKUs whose target margins increased over the last 3 days. This query would provide insights into recent adjustments and help track performance based on the updated margins.

4.2. Query Design

The query was designed to calculate the difference between the new and old target margins (New_TM - Old_TM) and filter for positive changes within the last three days.

SQL Query:



```
SELECT
SKU,
Region,
New_TM - Old_TM AS Change_TM
FROM `resonant-amulet-436309-u6.VidaXL.Target_margin_history`

WHERE New_TM - Old_TM > 0 and Update_date >= DATE_SUB('2024-09-23', INTERVAL 3 DAY)
```

There is no data to display.

4.3. Explanation of the Query

- **Change_TM Calculation:** The difference between New_TM and Old_TM was calculated to identify changes in the target margin.
- **Filtering for Positive Changes:** The condition New_TM - Old_TM > 0 was used to filter for SKUs with increased target margins.

- **Date Filter:** The DATE_SUB function was applied to only include changes made in the last three days from the specified date 2024-09-23. The date 2024-09-23 was used as an example representing the current date when this query was created.

4.4. Note on Data Insertion

Data was not inserted to test the query due to the time constraints and the fact that data insertion was not part of the initial requirements. The primary objective was to design the table structure and create a query that could be used for analysis when data is available.

5. Conclusion

This report outlines the process of correcting data, transforming it for BigQuery compatibility, and designing a historical table to track target margin changes. The decision to use the INTEGER data type for target margins was based on the existing company data structure, ensuring consistency and minimizing potential integration issues. The final query effectively identifies SKUs with increased target margins, providing valuable insights for the planning team.

6. Recommendations

- **Consider Using FLOAT64:** If future data includes decimal values for target margins, it is recommended to update the table schema to use the FLOAT64 data type for better precision.
- **Automation:** Automate the data upload process to BigQuery to streamline the daily update process.
- **Data Insertion for Testing:** When time permits, inserting test data into the table would allow for a more thorough validation of the queries designed.