Final Project – Data Warehousing

1. Executive Summary

In today's data-driven retail environment, organizations must leverage extensive datasets to gain deeper insights into their operations, customers, and products. This project explores a scenario inspired by a mid-sized online retail company seeking to enhance profitability, understand customer buying behaviors, and drive strategic decision-making. By integrating transactional (OLTP) and analytical (OLAP) models, as well as performing dimensionally-modeled data warehousing, we develop a set of analyses and visualizations that inform executive-level business strategies.

The work focuses on several key areas of inquiry. First, we examine whether a customer's commute distance to a store location correlates with their average spending, potentially guiding location-based promotions or personalized marketing. Next, we investigate month-over-month sales growth trends to identify seasonal patterns and operational improvements. We also assess the impact of customer demographics—such as income or marital status—on overall sales, enabling more targeted marketing efforts. Further, the analysis delves into customer lifetime value to identify the most profitable customer segments for long-term retention strategies. Lastly, we explore which products yield the highest profit margins and segment customers by their yearly spending patterns, guiding product selection, inventory management, and personalized recommendations.

Through dimensional modeling, analytic SQL queries, and potential visualization platforms, this project offers a coherent strategy for transforming raw data into actionable business insights. The ultimate goal is to improve decision-making, allocate marketing resources effectively, and drive sustained profitability.

2. Problem Statement

As online retail competition intensifies, decision-makers face a complex web of challenges: maintaining growth, enhancing profitability, and consistently meeting customer expectations. The complexity is compounded by varied customer demographics, dynamic product lines, and evolving market conditions.

This project addresses the core challenge of turning disparate, large-scale transactional data into meaningful insights that inform strategic actions. Specifically, the organization seeks to answer critical questions:

Customer Behavior & Geography: Does the distance customers commute to shop correlate with their average sales amounts, and how can this inform tailored marketing or logistics strategies?

Revenue Growth Patterns: Can identifying month-over-month sales trends help forecast demand, optimize inventory, and schedule targeted promotions to drive steady growth?

Customer Demographics: How do demographic factors—such as marital status, gender, income, or education level—influence purchasing behaviors and overall sales, and what does this mean for segmentation and personalization?

Customer Lifetime Value (CLV): Which customers generate the most long-term value, and how can retention strategies be refined to nurture these relationships over time?

Product Profitability: Which products yield the highest margins, and how can product mix adjustments or supplier negotiations improve profitability?

Customer Spending Segmentation: How can we categorize customers by their yearly spending patterns to inform differentiated marketing approaches and loyalty programs?

By addressing these questions through a robust data warehousing architecture, dimensional modeling, and analytic queries, the organization aims to reduce uncertainty, identify growth opportunities, and ultimately make data-driven decisions that enhance long-term success.

3. Literature Review

Implementing data warehousing and business intelligence solutions centers on key concepts like dimensional modeling, data cube design, and the integration of OLTP and OLAP systems. Foundational authors such as Ralph Kimball and Bill Inmon have established best practices that inform essential architectural decisions for advanced analytics.

Dimensional modeling, as advocated by Kimball and Ross (2013) in *The Data Warehouse Toolkit*, differentiates analytical storage structures from operational relational schemas. Star and snowflake schemas organize data around conformed dimensions and fact tables, enhancing clarity and query efficiency. Key principles include defining the "grain" of fact tables, identifying relevant "dimensions" (e.g., customers, products, time), and specifying "facts" (quantitative measures like sales amounts). These choices facilitate complex queries and integrated analysis across fact tables.

Data cube design extends dimensional modeling by providing a framework for multi-dimensional analysis. Cubes aggregate facts across multiple dimensions, enabling operations like drilling down, rolling up, and slicing data to uncover patterns and trends. This simplifies navigation of complex datasets and supports diverse decision-making scenarios, from growth analyses to customer segmentation. Gray et al. (1997) highlight that pre-aggregated data cubes enhance OLAP performance by speeding up query responses.

Effective SQL aggregation is crucial for leveraging dimensional models and data cubes. Fact tables optimized for SQL aggregate functions (SUM, AVG, COUNT) enable the generation of key metrics like monthly revenue trends and customer lifetime value. Well-structured schemas and appropriate indexing facilitate efficient SQL-based aggregation, supporting rapid responses for BI dashboards and ad-hoc queries.

The distinction between OLTP and OLAP environments is critical. OLTP systems manage day-to-day transactions using normalized schemas for data integrity and minimal redundancy. In contrast, OLAP systems handle read-heavy analytical workloads, employing denormalized schemas like star and snowflake designs to reduce join complexity and accelerate queries (Inmon, 2005; Kimball & Ross, 2013). This separation allows data warehouses to support enterprise-wide analytics while OLTP systems maintain operational efficiency.

Modern BI platforms continue to evolve, incorporating techniques like columnar storage and Massively Parallel Processing (MPP) architectures in cloud-based data warehouses such as BigQuery, Snowflake, and Azure Synapse. While foundational principles remain relevant, these platforms simplify aspects like indexing and partitioning. Nonetheless, core concepts of dimensional modeling, cube structures, and SQL aggregations persist as guiding practices for robust, scalable BI solutions.

In conclusion, dimensional modeling, data cube design, SQL aggregation strategies, and the differentiation between OLAP and OLTP form a comprehensive framework for data warehousing and business intelligence. Adhering to these principles enables efficient data exploration and transforms transactional data into actionable insights, facilitating informed and strategic decision-making.

4. Data Collection and Preparation

The data collection process for this project involves sourcing, integrating, and refining datasets that capture customer attributes, product details, sales transactions, and temporal information. Data was sourced from:

Below is an overview of the tables in our dataset,

DIM_Customer:

Contains customer-related information such as CustomerKey (unique identifier), demographics (gender, marital status), location, and purchase behavior (income, number of cars, children). This table helps to link customers to sales data for customer segmentation and analysis.

DIM_Date:

Stores date-related information, including DateKey (unique identifier), specific day, month, and year details, along with fiscal periods and calendar breakdowns. Useful for time-based analysis such as month-on-month growth, sales trends, and time-series analysis.

DIM_Product:

Provides product-related details such as ProductKey (unique identifier), name, pricing, inventory data (safety stock, reorder points), and specifications like size, weight, and color. It helps in product-level analysis such as sales, profitability, and stock levels.

FACT_Product:

Tracks product movements and quantities over time, including attributes like ProductKey, DateKey, and transaction details (e.g., units in, units out). This table supports analysis related to inventory flow and product performance over time.

FACT_Sales_Internet:

Contains sales transaction data from internet-based orders. It includes ProductKey, CustomerKey, sales amounts, order quantities, discounts, and tax information. This table is essential for sales and customer analytics such as total revenue, profit margins, and customer spending patterns.

OLTP_Product:

Provides detailed product information such as ProductID, name, product number, cost, price, inventory attributes (safety stock, reorder point), and product specifications (size, color, weight). This table is used for real-time product management and helps to monitor product performance and inventory health.

OLTP_Sales:

Contains transactional sales data, including SalesOrderID, SalesOrderDetailID, OrderQty, ProductID, SpecialOfferID, unit price, and discounts. This table is used for real-time sales reporting, tracking sales per order, and monitoring order quantities for each product.

Migrating Data to Google Cloud Platform

We have used GCP as a **Cloud Data Warehouse Solution** for storage, processing and performing analytics. Below are the steps we followed,

1. Sign In to GCP Console

- Open the Google Cloud (console.cloud.google.com).

- Sign in with the Google account and select or create a project.

2. Access BigQuery Studio

- In the Google Cloud Console, use the navigation menu on the left to find BigQuery under the "Big Data" section.

- Click on BigQuery to open the BigQuery Console.

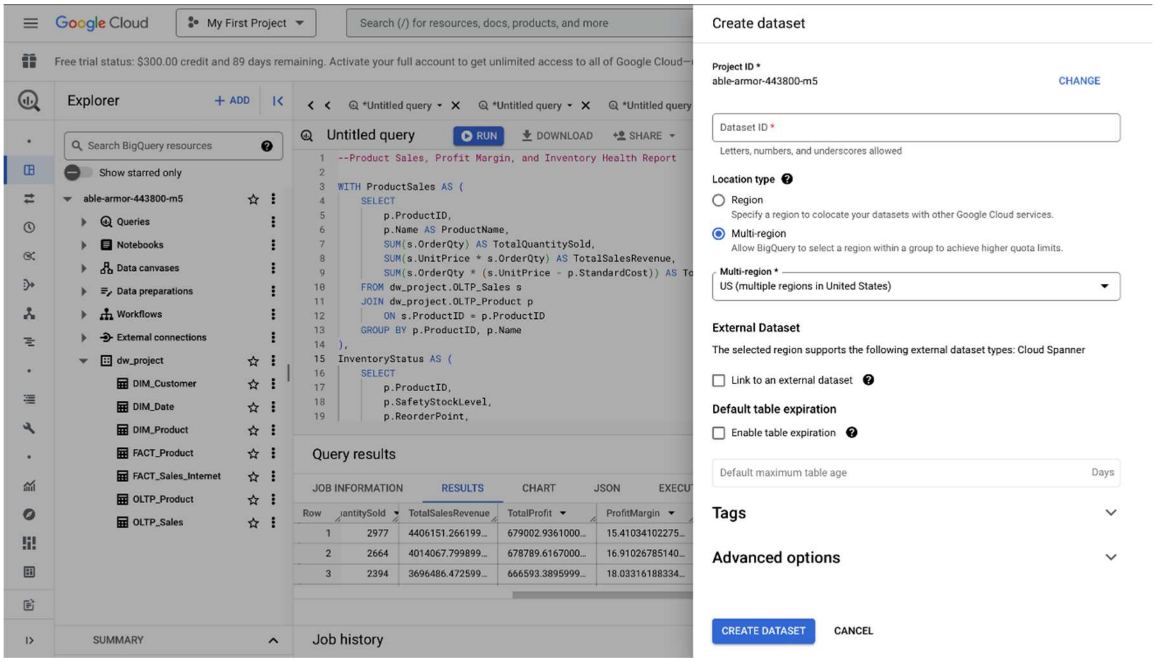
4. Create a Dataset (available under Project ID)

- In the BigQuery console, click the Project Name on the left panel.

- Click Create Dataset at the top.

- Enter a name for the dataset and configure settings as needed.

- Click Create Dataset.



6. Upload Your Data

As for this project, our data was small – we preferred a bit easier approach by directly uploading the CSV file into the BigQuery Dataset. For Big Data, the proper approach would be to use Google Could Storage (Storage Bucket) followed by ETL operations. Eventually, the final step is importing the data into BigQuery.

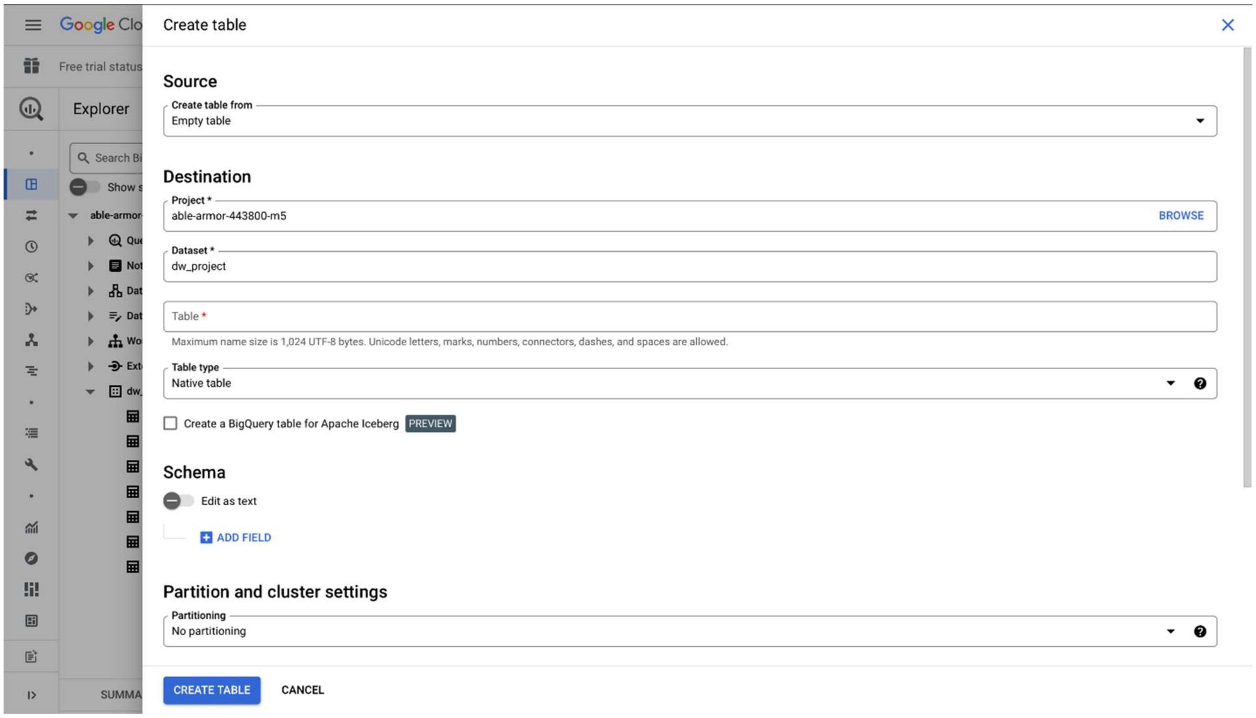
- In BigQuery Studio, click on the + Create button.

- Select Upload Data from the options.

- Choose the file you want to upload.

- Select the dataset you want to upload the file to or create a new one.

- Specify the file format (CSV, JSON, etc.), delimiter (for CSV), and other options (e.g., schema, headers).



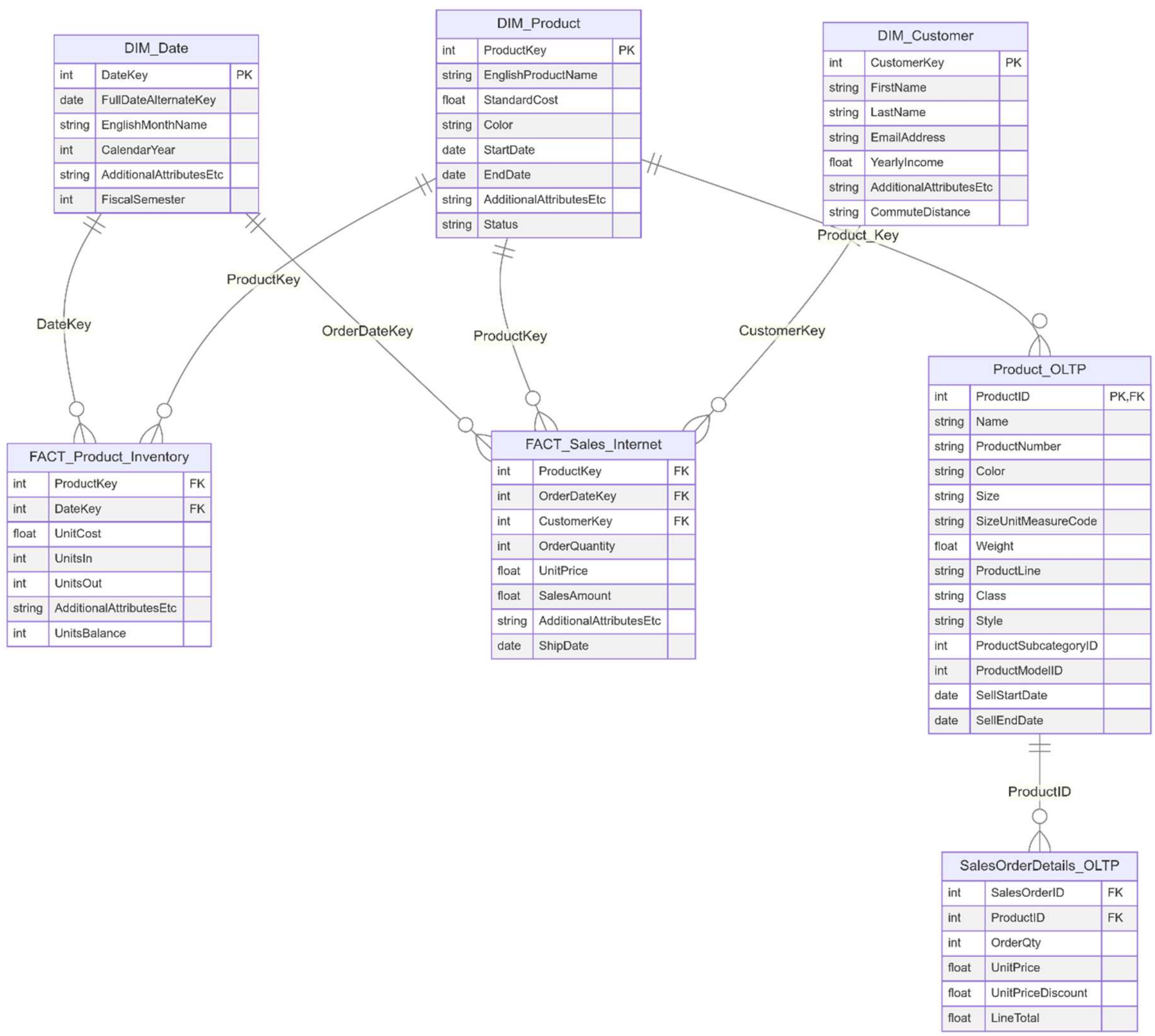
Tools and Platforms:
Data may be collected and stored within a cloud-based environment, such as Google BigQuery, Snowflake, or Azure Synapse, or housed on-premises in a data warehouse appliance. Extract, Transform, Load (ETL) or Extract, Load, Transform (ELT) processes can be orchestrated using tools like Apache Airflow, dbt, or native cloud services.

Quality Assurance and Governance:
Throughout the data collection and preparation phases, quality checks (row counts, referential integrity checks, duplication tests) and metadata documentation ensure the reliability of the final data warehouse environment. This governance framework helps maintain trust in the analytics produced downstream.

By meticulously collecting and preparing the data, the project sets the stage for robust dimensional modeling, efficient OLAP-style queries, and the advanced analytics required to address the research questions—ranging from commute distance correlations and monthly growth trends to demographic-based customer segmentation and product profitability.

5. <u>Database Design</u>

Dimensional Modelling using ERD



- **Fact Tables**:
  - Sales_Table:
    - Stores measurable business metrics (e.g., SalesAmount, TaxAmt, OrderQuantity) with keys to link to dimensions like ProductKey, CustomerKey, and DateKey.
  - Product_Inventory_Table:
    - Captures metrics related to inventory or product movement (e.g., UnitsIn, UnitsOut, UnitCost).

- **Dimension Tables**:
  - DIM_Product:
    - Includes descriptive attributes like ProductLine, ModelName, and multilingual descriptions, enhancing slicing and dicing capabilities.
  - DIM_Date:
    - Provides time-based granularity, enabling analysis across multiple time hierarchies such as days, weeks, quarters, and fiscal periods.
  - DIM_Customer:
    - Offers customer-level details like YearlyIncome, Gender, and GeographyKey to analyze sales by demographics or location.
- **Hierarchies and Granularity**:
  - Dimensions like DIM_Date have natural hierarchies (e.g., Day → Month → Quarter → Year).
  - DIM_Product includes hierarchies like ProductSubcategory → ProductLine.
  - Fact tables provide transaction-level granularity.

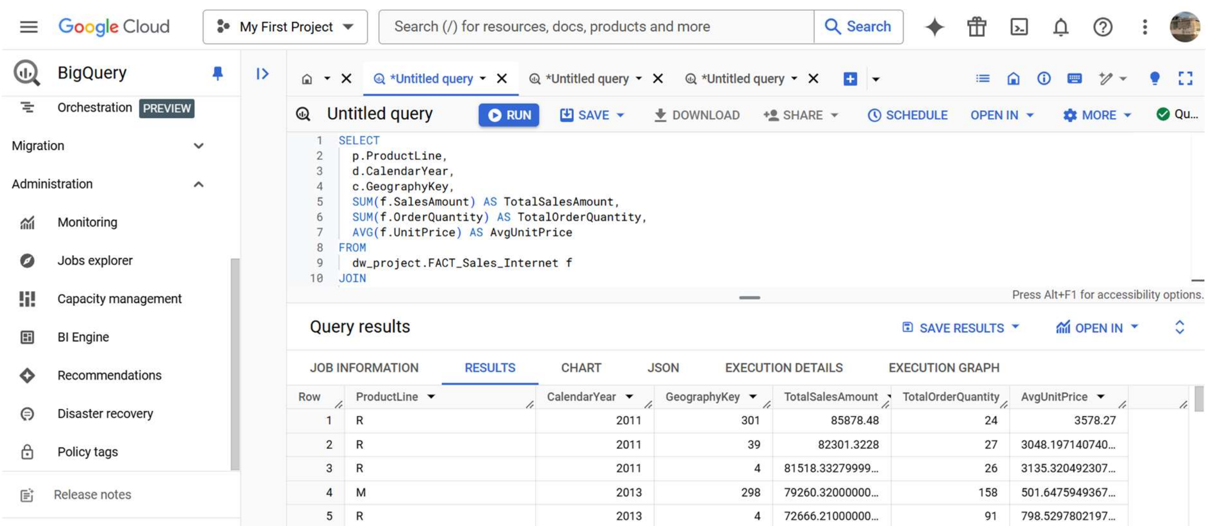
Appropriating Data Cubes Design through our ERD.

  - Data cubes are often used in OLAP for summarizing data in multiple dimensions. The cube for our data include measures like total sales or quantities, broken down by dimensions like products, time, and customer.
  - To design a data cube, we have to define measures (numeric values we want to aggregate) and dimensions (categories for summarizing data).
  - These cubes allow slicing (e.g., filtering by product category), dicing (e.g., comparing sales across regions), and rolling up (e.g., aggregating by year)

Creating Data Cubes in Google Big Query Studio:

1. Sales Cube

Goal: Analyze SalesAmount by ProductLine, Year, and GeographyKey.

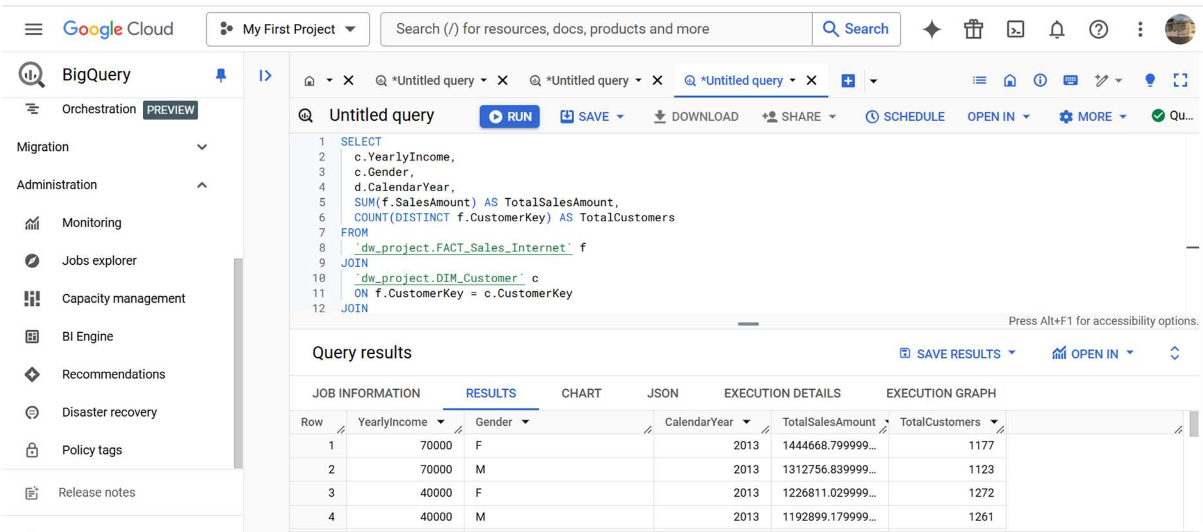


Explanation:

- **Dimensions**: ProductLine, CalendarYear, GeographyKey.
- **Measures**: SUM(SalesAmount), SUM(OrderQuantity), AVG(UnitPrice).
- **Analysis**: This query generates a cube showing total sales and order quantities for each product line in each geography for every year.

2. Customer Insights Cube

Goal: Analyze SalesAmount by YearlyIncome, Gender, and Year.



Explanation:

- **Dimensions**: YearlyIncome, Gender, CalendarYear.
- **Measures**: SUM(SalesAmount), COUNT(DISTINCT CustomerKey).
- **Analysis**: This query builds a cube for analyzing spending trends by income bracket and gender over time.

6. Exploratory Data Analysis
7. Reporting, Modeling and Storytelling

- Queries:

OLAP systems are primarily used for data analysis and business intelligence purposes. They allow users to query large datasets quickly and perform complex aggregations, calculations, and trend analyses. Here are some common use cases of OLAP:

- Business Reporting and Dashboards

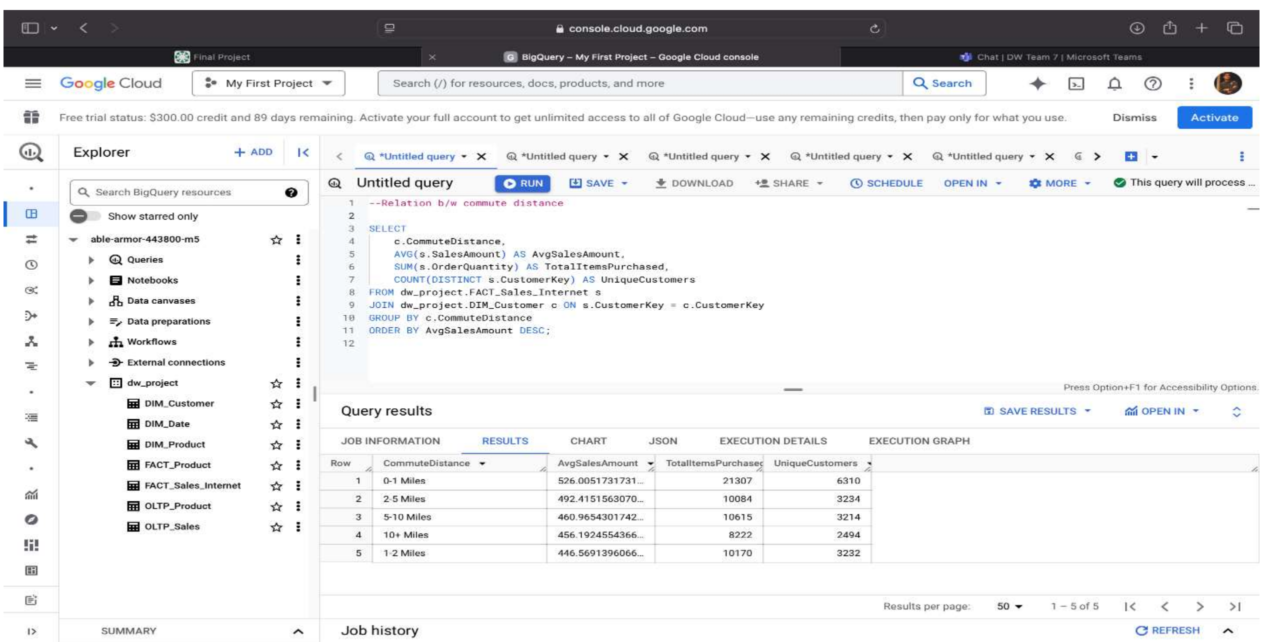
- Data Warehousing and Trend Analysis

- Customer Segmentation

- Financial Planning and Forecasting

Our Project mainly consists of an OLAP Data Mart with 2 FACTs and 3 Dims. Below are some analytical SQL queries using this DataMart,

1. Relation Between Commute Distance and Sales



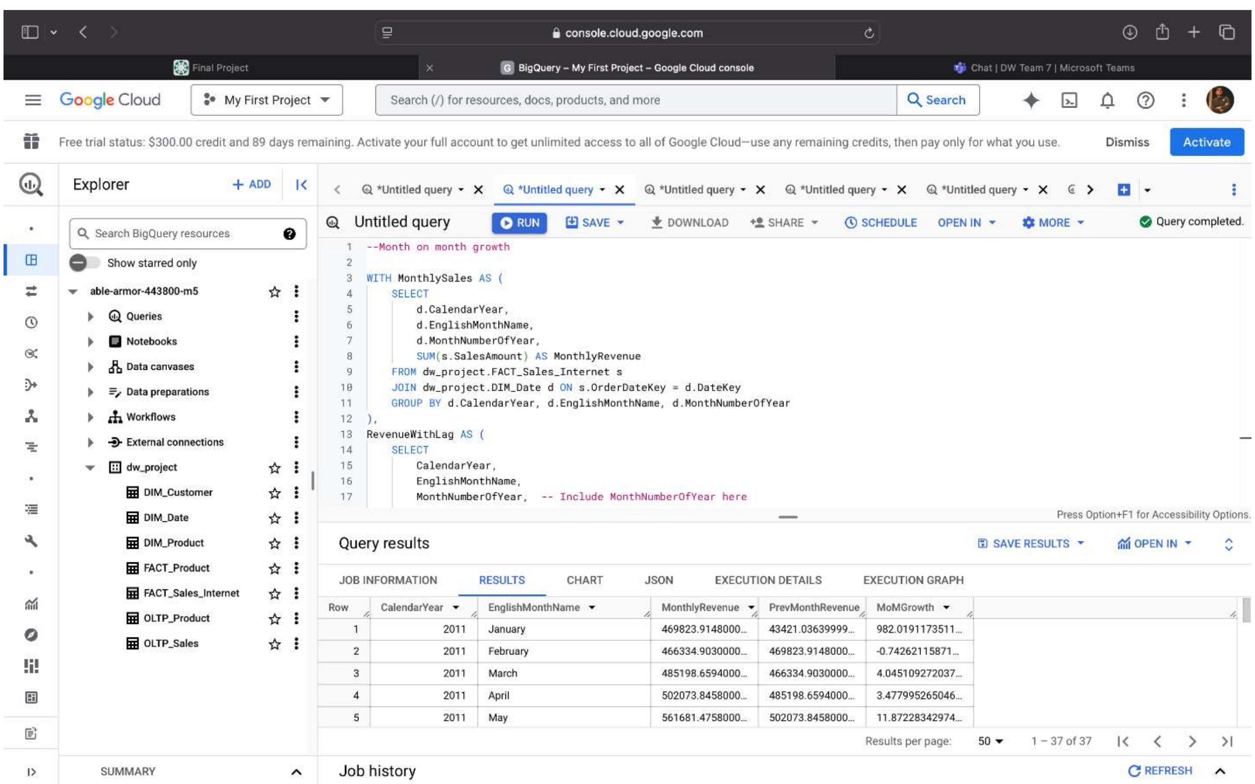
Purpose:
- This query examines how the commute distance of customers relates to their purchasing behavior.

- Metrics like average sales amount, total items purchased, and unique customers are calculated for each commute distance category.

Use Cases:
- Business Targeting: Identify customer groups based on commute distance and create targeted marketing campaigns.
- Resource Allocation: Align resources and product availability for regions with high-spending commuters.

Insights:
- Uncovers if customers with longer or shorter commutes spend more on average.
- Determines if certain commute groups purchase more items, which could suggest lifestyle patterns.

2. Month-on-Month (MoM) Growth in Sales



Full Query:

```sql
--Month on month growth

WITH MonthlySales AS (
    SELECT
        d.CalendarYear,
        d.EnglishMonthName,
        d.MonthNumberOfYear,
        SUM(s.SalesAmount) AS MonthlyRevenue
    FROM dw_project.FACT_Sales_Internet s
    JOIN dw_project.DIM_Date d ON s.OrderDateKey = d.DateKey
    GROUP BY d.CalendarYear, d.EnglishMonthName, d.MonthNumberOfYear
),
RevenueWithLag AS (
    SELECT
        CalendarYear,
        EnglishMonthName,
        MonthNumberOfYear,  -- Include MonthNumberOfYear here
        MonthlyRevenue,
        LAG(MonthlyRevenue) OVER (ORDER BY CalendarYear, MonthNumberOfYear) AS PrevMonthRevenue
    FROM MonthlySales
)
SELECT
    CalendarYear,
    EnglishMonthName,
    MonthlyRevenue,
    PrevMonthRevenue,
    ((MonthlyRevenue - PrevMonthRevenue) / PrevMonthRevenue) * 100 AS MoMGrowth
FROM RevenueWithLag
WHERE PrevMonthRevenue IS NOT NULL
ORDER BY CalendarYear, MonthNumberOfYear;  -- Ensure proper ordering
```

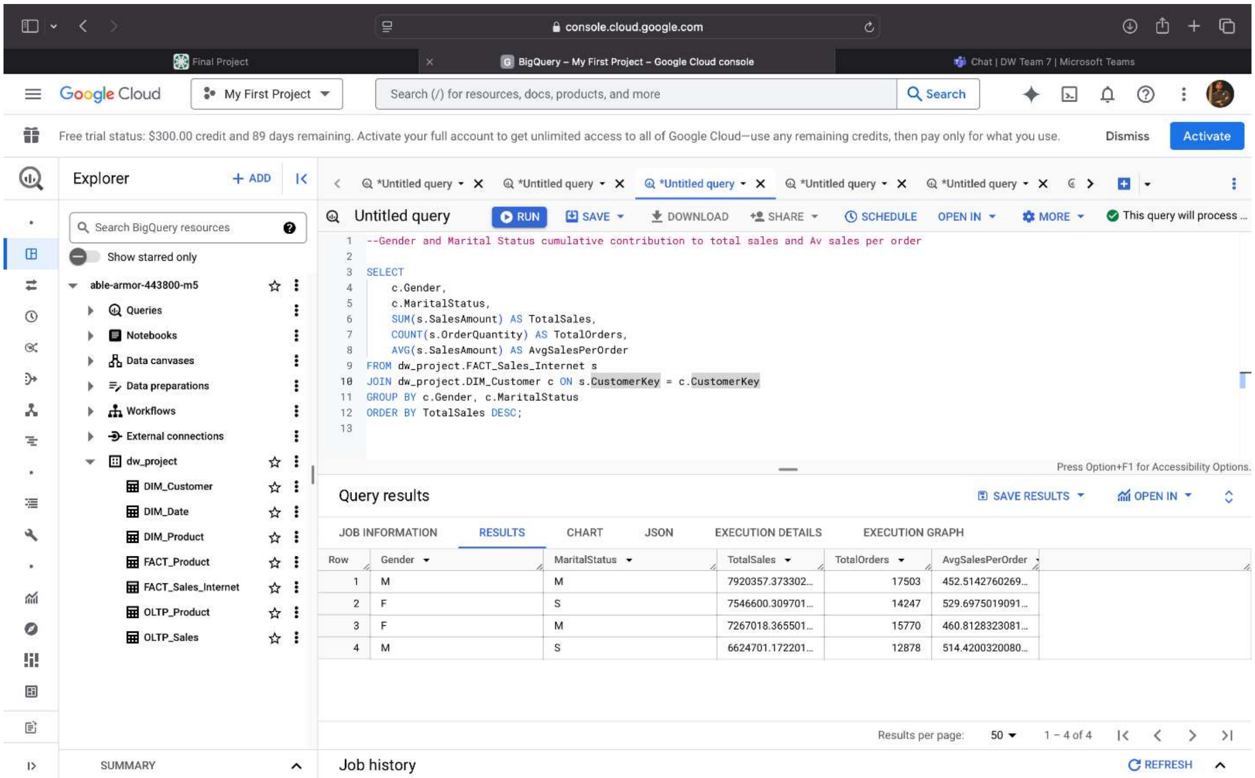
Purpose:
- Tracks sales growth or decline by comparing the monthly revenue to the previous month.
- Calculates the percentage growth or drop (MoM Growth).

Use Cases:
- Trend Analysis: Evaluate business performance over time and identify seasonal patterns.
- Strategy Adjustment: Pinpoint months with poor performance and investigate causes, such as marketing efforts, product availability, or customer demand.

Insights:

- Shows months with significant growth or decline in revenue.
- Identifies trends to forecast future performance and plan marketing campaigns or sales initiatives.

3. Gender and Marital Status Contribution

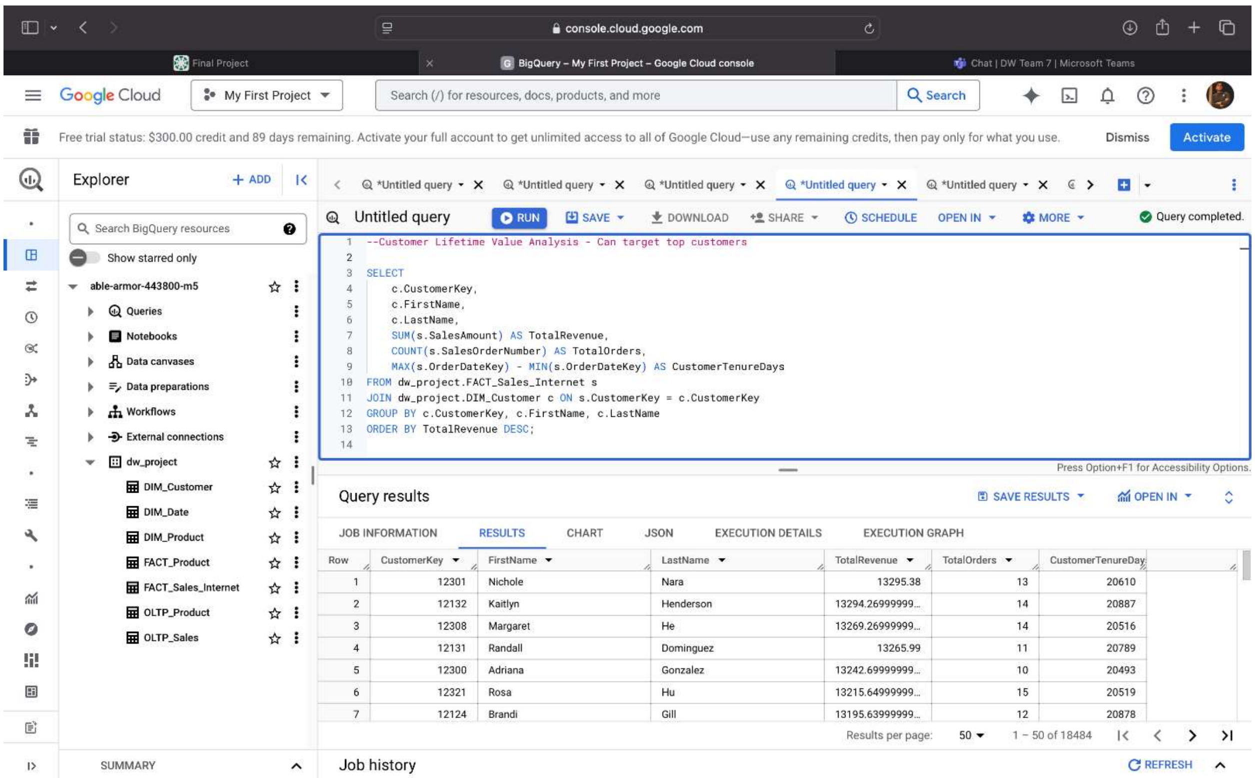


Purpose:
- Breaks down sales data by customer gender and marital status to understand their contributions to overall sales.

Use Cases:
- Demographic Insights: Help businesses tailor products, offers, and communication strategies for specific demographic groups.
- Product Recommendations: Identify differences in spending habits between groups to recommend products.

Insights:
- Indicates which demographic groups generate the most revenue.
- Reveals purchasing patterns, such as whether single or married customers spend more.

4. Customer Lifetime Value Analysis



Purpose:

- Calculates customer lifetime value (CLV) by aggregating total spending, order count, and tenure (days active).

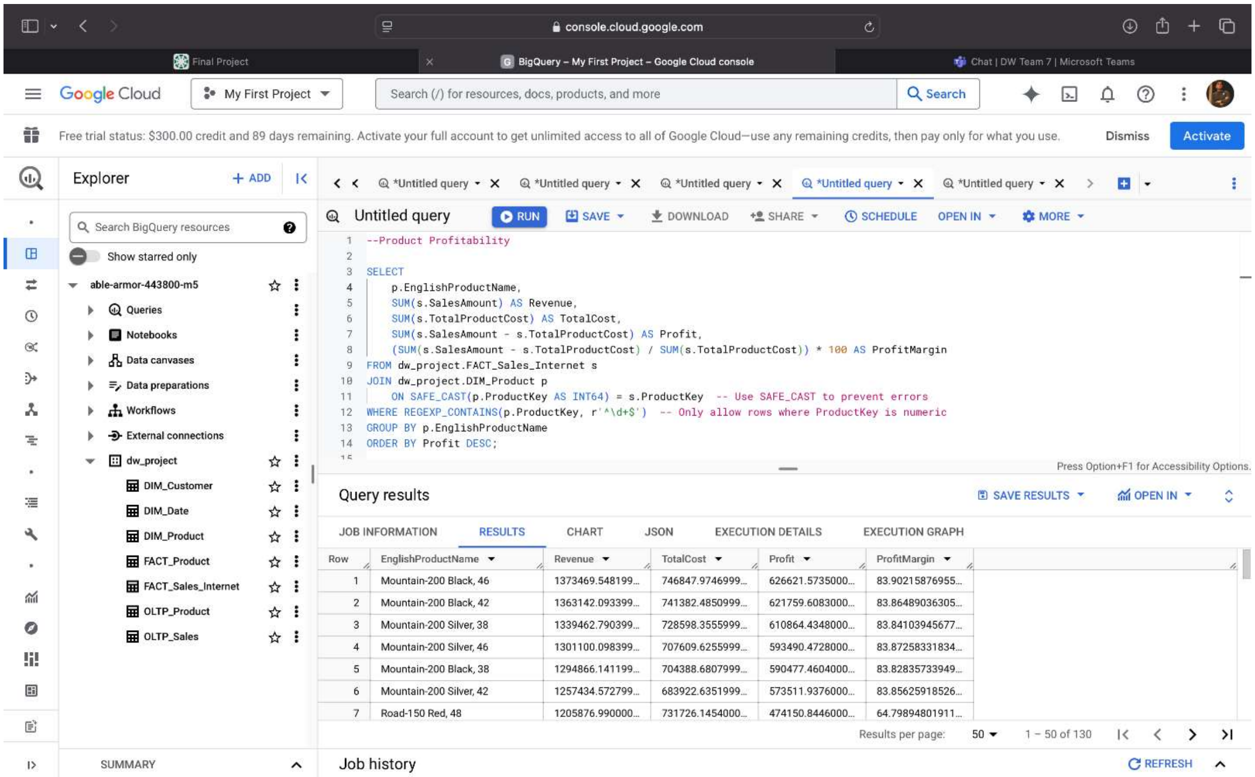
Use Cases:

- Customer Retention: Focus on high-value customers by offering personalized rewards or discounts.
- Profitability Analysis: Identify which customers contribute the most to revenue.

Insights:

- Highlights the most loyal and high-revenue customers.
- Provides data to improve customer retention strategies and increase repeat purchases.

5. Product Profitability



Purpose:
- Evaluates product profitability by comparing revenue to costs and calculating profit margins.

Use Cases:
- Inventory Decisions: Identify top-performing products to prioritize in inventory and promotions.
- Pricing Strategies: Adjust prices for low-margin products to improve profitability.

Insights:
- Highlights the most profitable products.
- Identifies products with low profitability for further investigation.

6. Customer Segmentation by Spending Category

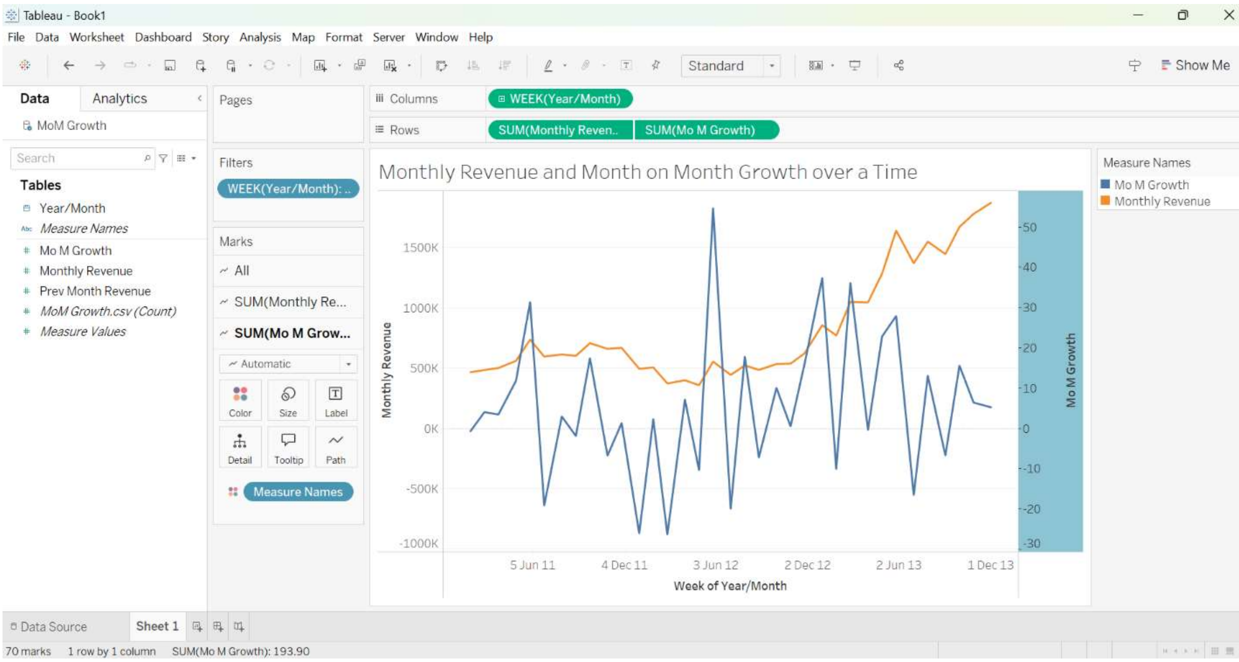


Tableau - Book1
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Show Me
Data
Analytics
MoM Growth
Search
Tables
Year/Month
Measure Names
Mo M Growth
Monthly Revenue
Prev Month Revenue
MoM Growth.csv (Count)
Measure Values
Pages
Filters
WEEK(Year/Month): ..
Marks
All
SUM(Monthly Re...
SUM(Mo M Grow...
Automatic
Color
Size
Label
Detail
Tooltip
Path
Measure Names
Columns
WEEK(Year/Month)
Rows
SUM(Monthly Reven..
SUM(Mo M Growth)
Monthly Revenue and Month on Month Growth over a Time
Monthly Revenue
1500K
1000K
500K
0K
-500K
-1000K
Mo M Growth
50
40
30
20
10
0
-10
-20
-30
5 Jun 11
4 Dec 11
3 Jun 12
2 Dec 12
2 Jun 13
1 Dec 13
Week of Year/Month
Measure Names
Mo M Growth
Monthly Revenue
Data Source
Sheet 1
70 marks
1 row by 1 column
SUM(Mo M Growth): 193.90

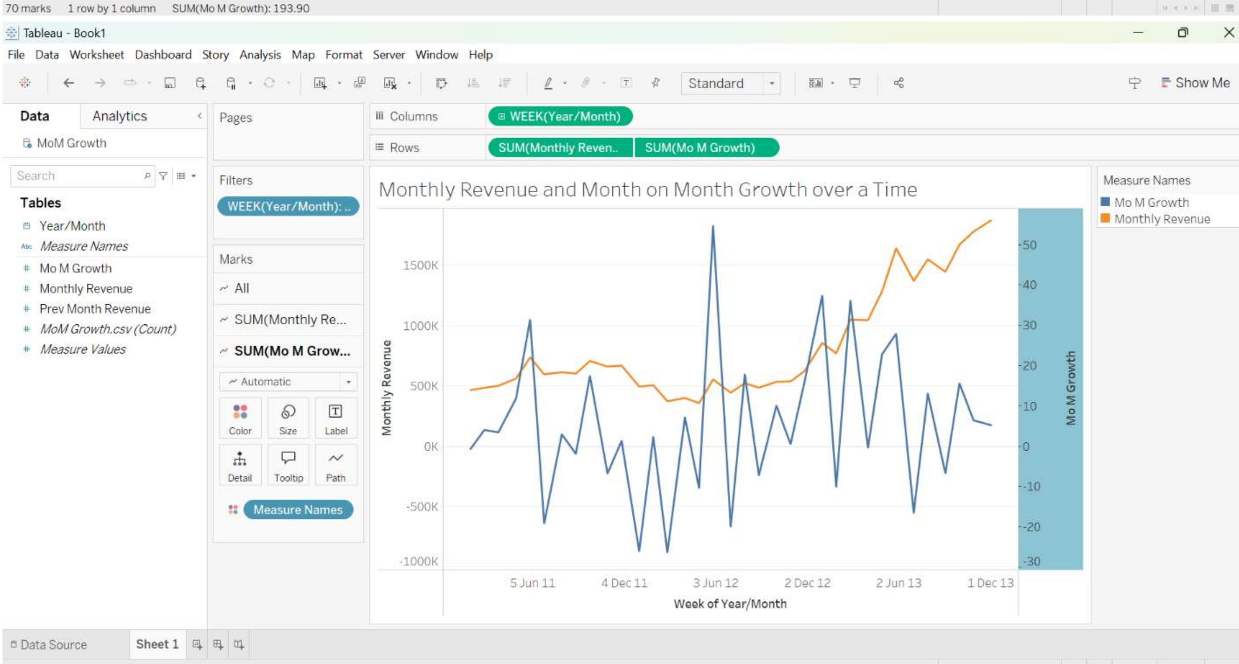


Tableau - Book1
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Show Me
Data
Analytics
MoM Growth
Search
Tables
Year/Month
Measure Names
Mo M Growth
Monthly Revenue
Prev Month Revenue
MoM Growth.csv (Count)
Measure Values
Pages
Filters
WEEK(Year/Month): ..
Marks
All
SUM(Monthly Re...
SUM(Mo M Grow...
Automatic
Color
Size
Label
Detail
Tooltip
Path
Measure Names
Columns
WEEK(Year/Month)
Rows
SUM(Monthly Reven..
SUM(Mo M Growth)
Monthly Revenue and Month on Month Growth over a Time
Monthly Revenue
1500K
1000K
500K
0K
-500K
-1000K
Mo M Growth
50
40
30
20
10
0
-10
-20
-30
5 Jun 11
4 Dec 11
3 Jun 12
2 Dec 12
2 Jun 13
1 Dec 13
Week of Year/Month
Measure Names
Mo M Growth
Monthly Revenue
Data Source
Sheet 1
70 marks
1 row by 1 column
SUM(Mo M Growth): 193.90

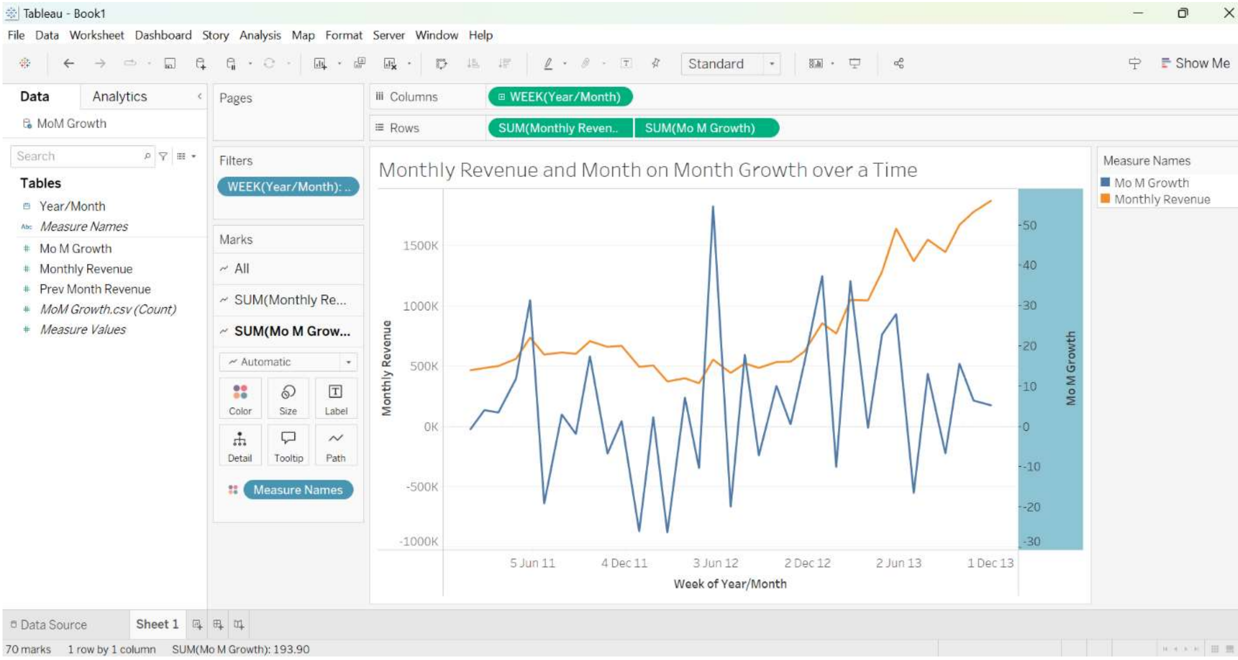
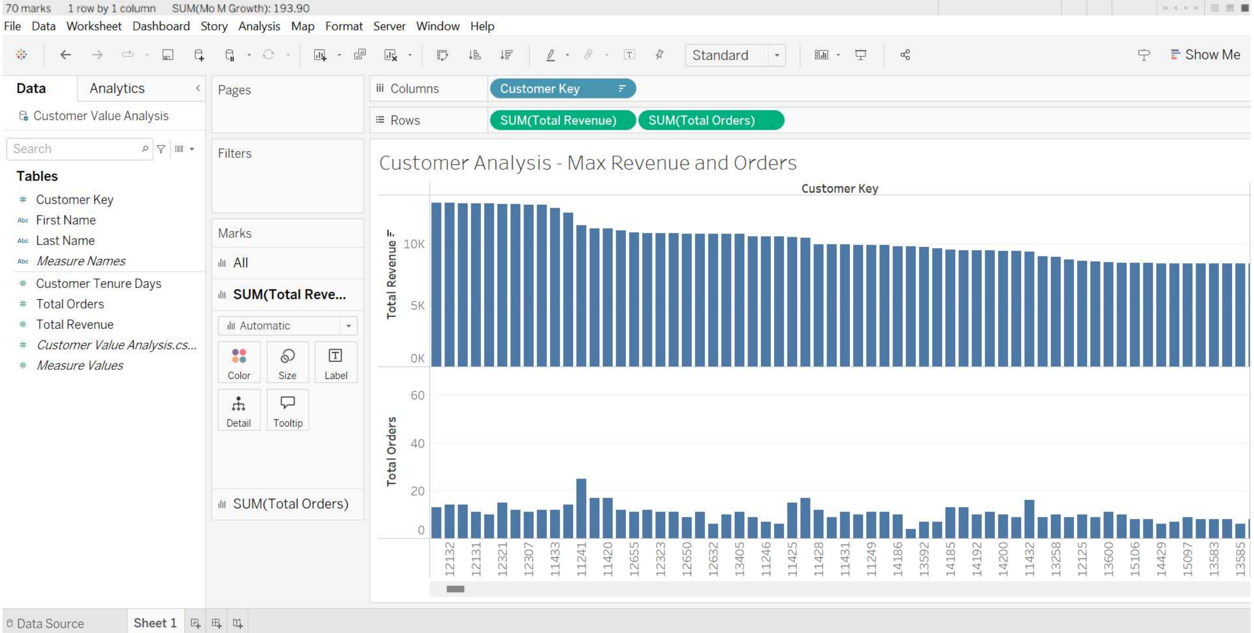
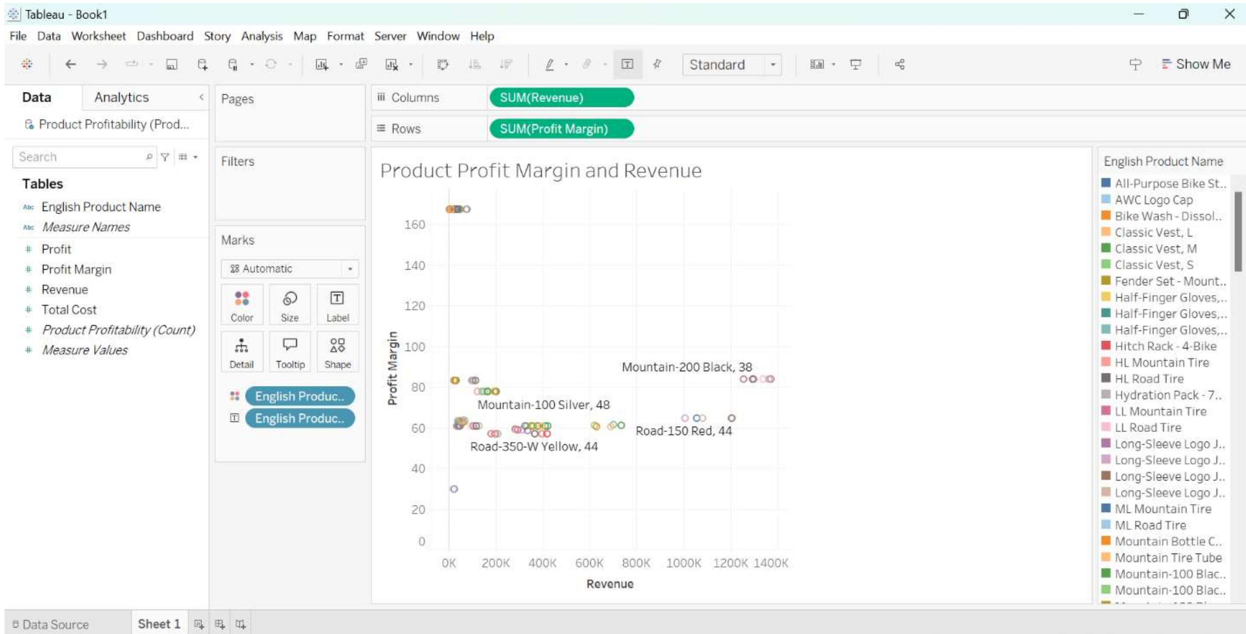


Tableau - Book1
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Show Me
Data
Analytics
MoM Growth
Search
Tables
Year/Month
Measure Names
Mo M Growth
Monthly Revenue
Prev Month Revenue
MoM Growth.csv (Count)
Measure Values
Pages
Filters
WEEK(Year/Month): ..
Marks
All
SUM(Monthly Re...
SUM(Mo M Grow...
Automatic
Color
Size
Label
Detail
Tooltip
Path
Measure Names
Columns
WEEK(Year/Month)
Rows
SUM(Monthly Reven..
SUM(Mo M Growth)
Monthly Revenue and Month on Month Growth over a Time
Monthly Revenue
1500K
1000K
500K
0K
-500K
-1000K
Mo M Growth
50
40
30
20
10
0
-10
-20
-30
5 Jun 11
4 Dec 11
3 Jun 12
2 Dec 12
2 Jun 13
1 Dec 13
Week of Year/Month
Measure Names
Mo M Growth
Monthly Revenue
Data Source
Sheet 1
70 marks 1 row by 1 column SUM(Mo M Growth): 193.90


File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Show Me
Data
Analytics
Customer Value Analysis
Search
Tables
Customer Key
First Name
Last Name
Measure Names
Customer Tenure Days
Total Orders
Total Revenue
Customer Value Analysis.cs...
Measure Values
Pages
Filters
Marks
All
SUM(Total Reve...
Automatic
Color
Size
Label
Detail
Tooltip
SUM(Total Orders)
Columns
Customer Key
Rows
SUM(Total Revenue)
SUM(Total Orders)
Customer Analysis - Max Revenue and Orders
Customer Key
Total Revenue
10K
5K
0K
Total Orders
60
40
20
0
12132
12131
12321
12307
11433
11241
11420
12655
12323
12650
12632
13405
11246
11425
11428
11431
11249
14186
13592
14185
14192
14200
11432
13258
12125
13600
15106
14429
15097
13583
13585
Data Source
Sheet 1



8. Conclusion