TUNIS BUSINESS SCHOOL
UNIVERSITY OF TUNIS

REPORT

IT 300
BUSINESS INTELLIGENCE & DATABASE ADMINISTRATION

Business Intelligence Research
Superstore Sales & Performance Data in the United States

Authors:
Majd Hamdi
Seif Ben Soltana

Submitted to:
Prof. Ameni Azzouz
Prof. Manel Abdelkader

TABLE OF CONTENTS

1. Introduction

This business intelligence project is focused on analyzing superstore sales and performance data in the United States.
The study aims to find out how different factors, such as product categories, customer segments, geographical regions, and sales dates influence the revenue and profit margins of the superstore. By understanding these relationships, we can gain valuable insights into the superstore's strengths and weaknesses, which can help the management of the company to make informed decisions and optimize their business strategies.
Through this project, we aim to improve the superstore's profitability and competitiveness in the retail industry.

2. Data Preparation

We will be employing the **ELT method**, which is to extract, load, then transform the data within our **Snowflake cloud data warehouse** using **SnowSQL** as our manipulation language.

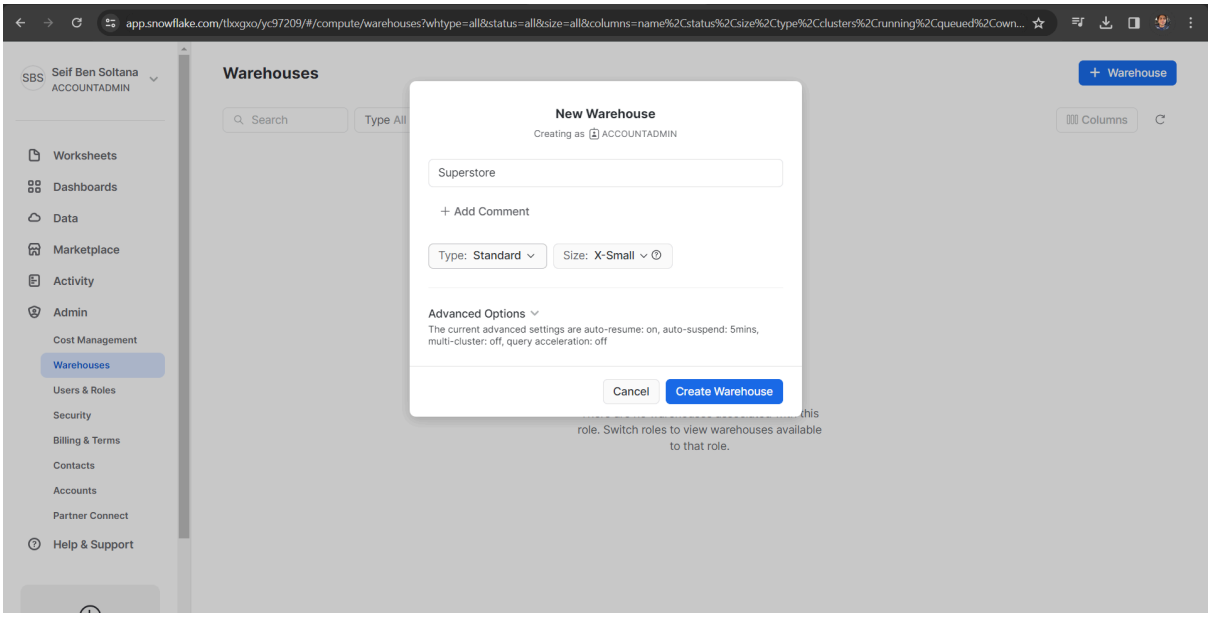
2.1. Data Extraction

We retrieved the Superstore Sales Data dataset from Datawonders' website.
We will be using 2 Excel sheets from the US (Sample) - SuperStore dataset:
The first sheet "Order" contains 21 columns;
The second Sheet "People" contains 2 columns.

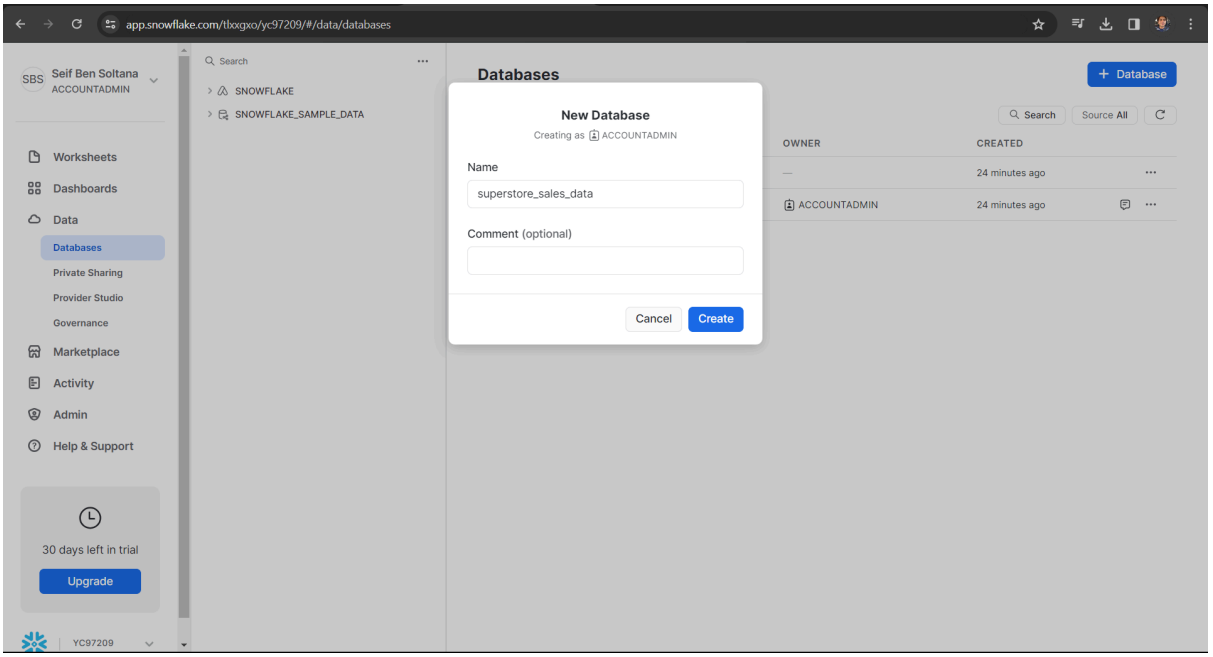
2.2. Data Loading

Since the dataset is of high quality and is already clean (no duplicates , no NaN values, and no corrupted data). Most of the work will be focused on splitting the data into distinct dimensions and choosing the adequate primary and foreign keys to establish the needed relationships between the tables.

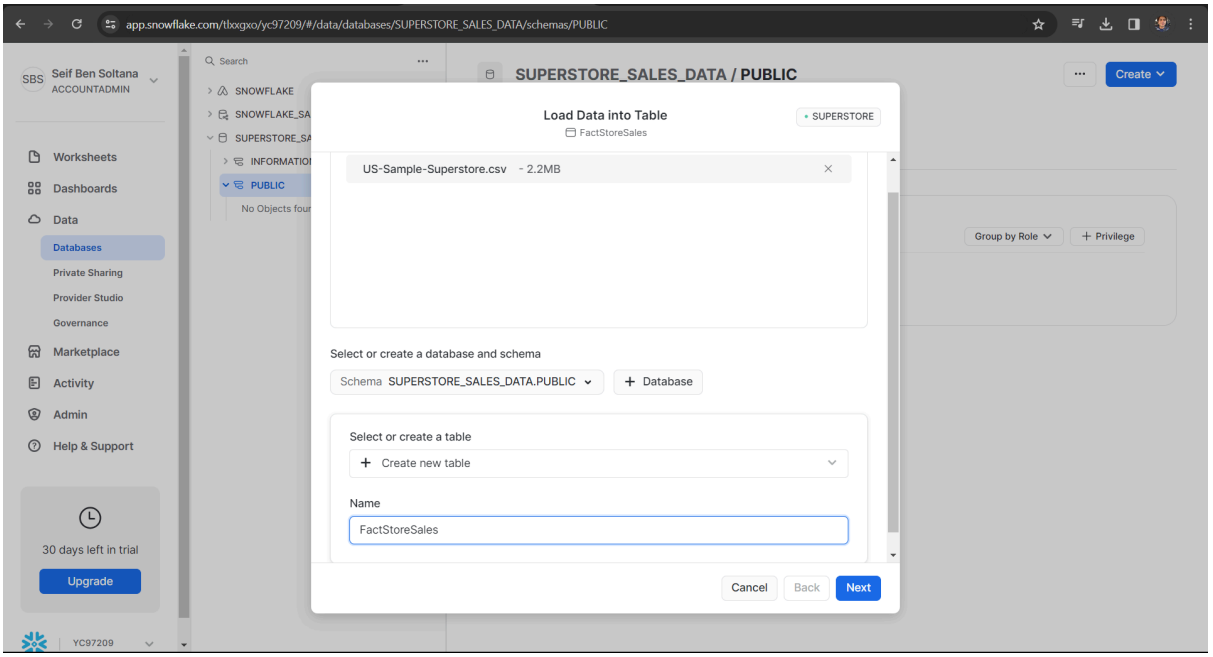
After creating a Data Warehouse ‘Superstore’ in **Snowflake**:



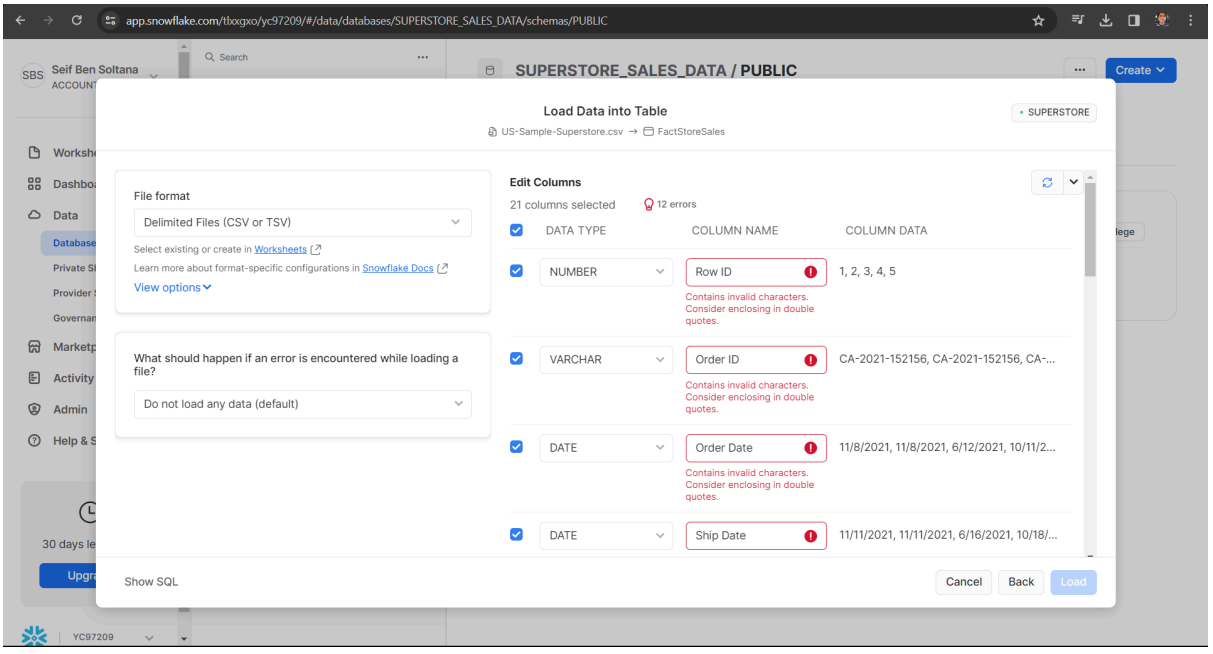
We create a new Database “superstore_sales_data”:



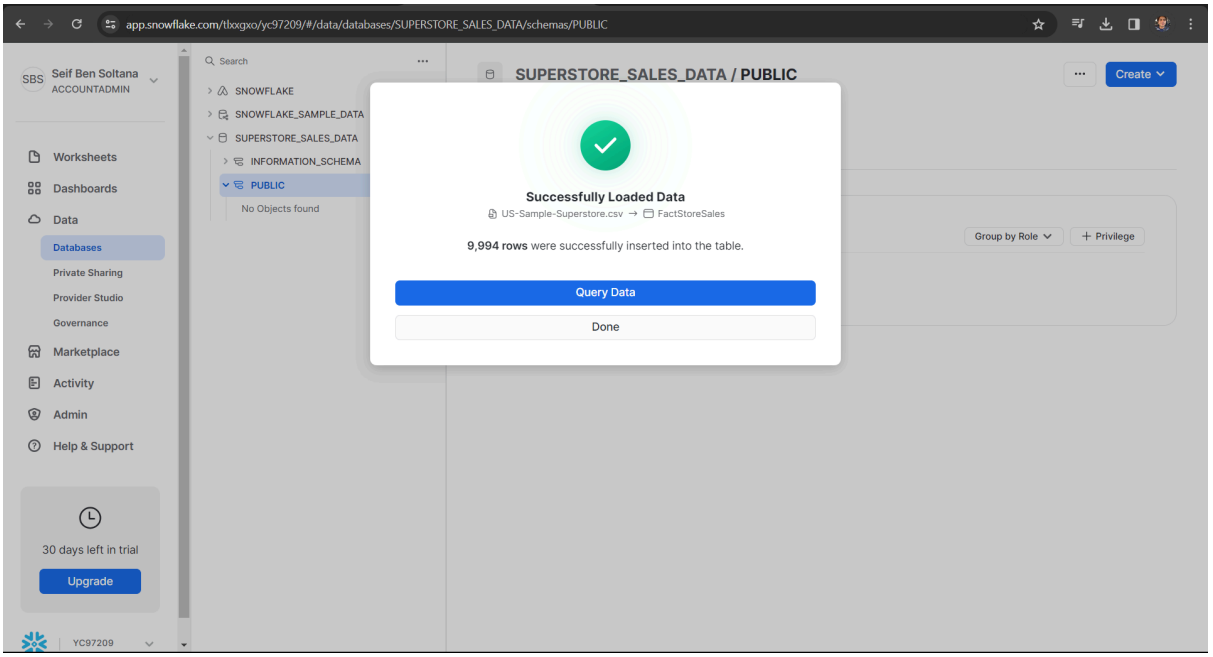
Within this database, we load our data from the “US-Sample-Superstore.csv” file into Table:

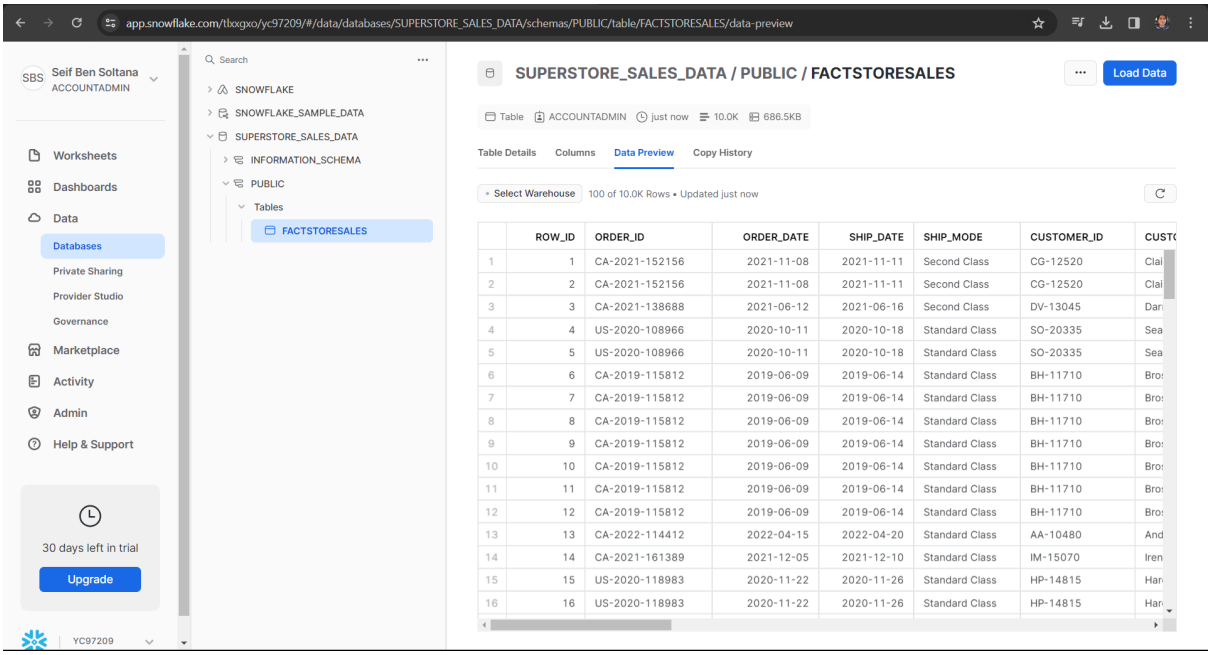


We transform the DATA TYPE and COLUMN NAME accordingly:

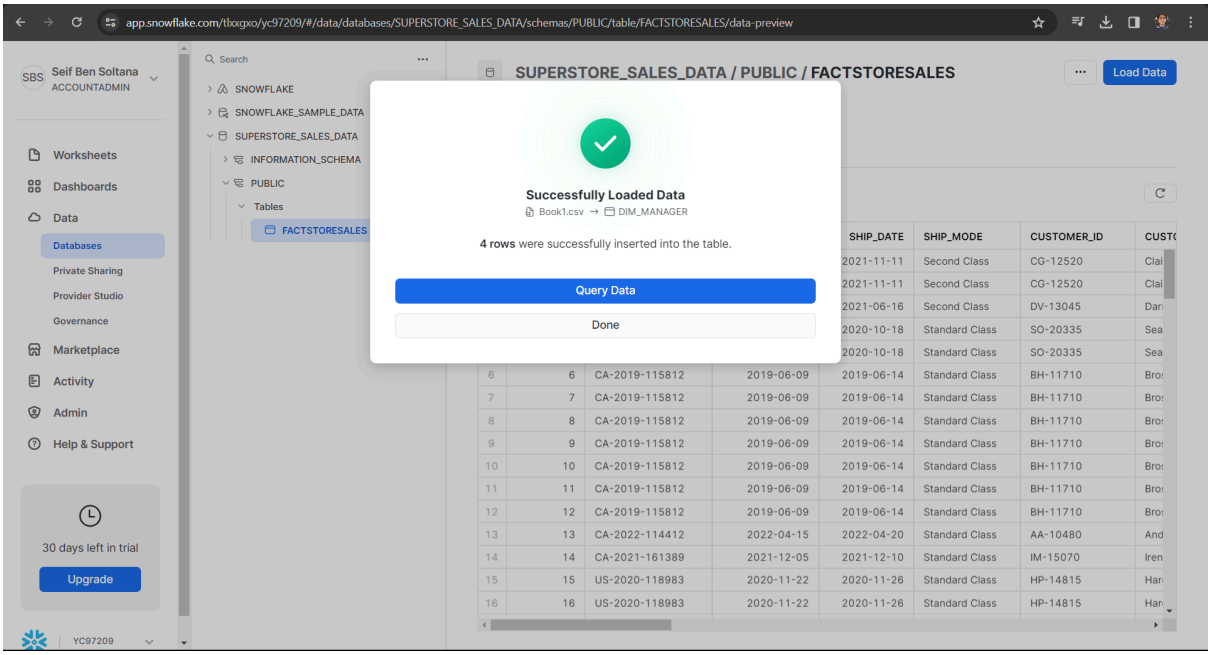


and we load the table into our “SUPERSTORE_SALES_DATA” database:





Now, we repeat the same steps with the file “Book.csv” that contains informations about the manager:



Please note that:
“US-Sample-Superstore.csv” will be loaded as FactStoreSales.
“Book.csv” will be loaded as DIM_MANAGER.

2.3. Data Transformation

Main points to solve during the transformation phase:

- Decide which dimensions to extract from the main table fact and assign the proper columns to them.
- Some PRODUCT_ID values refer to more than one product name and some POSTAL_CODE values refer to more than one city → These columns can not become primary keys in the new dimensions → We create new columns to solve this.

Using **SnowSQL**:

```
      ALTER TABLE FactStoreSales
      ADD (PRODUCT_PK INTEGER);
// We give all distinct combinations of (PRODUCT_ID,PRODUCT_NAME)
a unique number.
           MERGE INTO FactStoreSales o
           USING ( SELECT ROWID as rid,
                  DENSE_RANK() OVER (ORDER BY PRODUCT_ID,
PRODUCT_NAME)
      as new_id
                FROM FactStoreSales
                   ) t
           ON (o.ROWID = t.rid)
           WHEN MATCHED THEN UPDATE SET o.PRODUCT_PK = t.new_id;
//We extract the needed columns to the new DIM_PRODUCT.
```

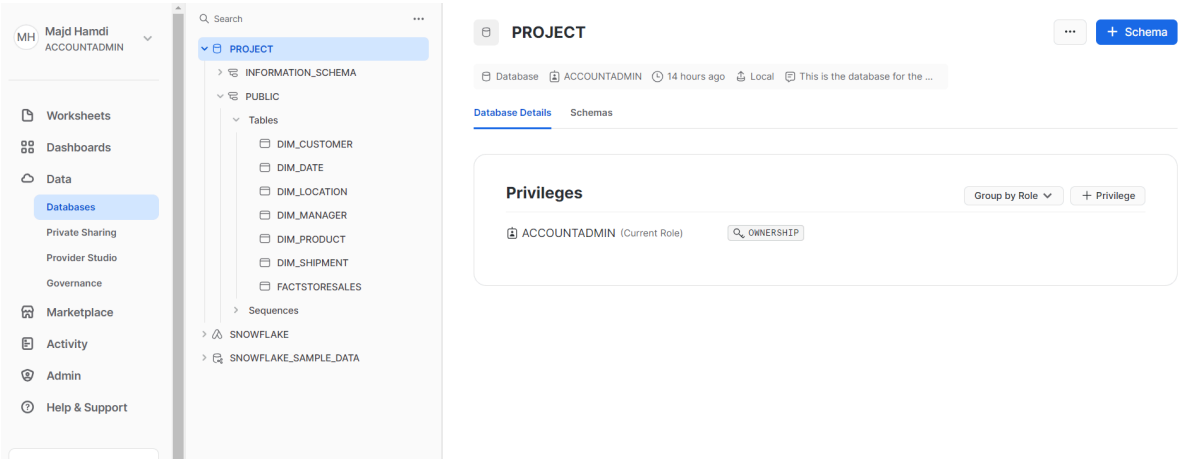
```
CREATE TABLE DIM_PRODUCT AS
SELECT DISTINCT PRODUCT_PK,PRODUCT_ID, PRODUCT_NAME,
CATEGORY,SUB_CATEGORY
      FROM FactStoreSales;
```

- In the case of exporting the final tables as .csv file format , one must replace all commas (,) from the following columns : PRODUCT_NAME, SALES,DISCOUNT,PROFIT into dots “.” (for example) otherwise they will be interpreted as field separators.

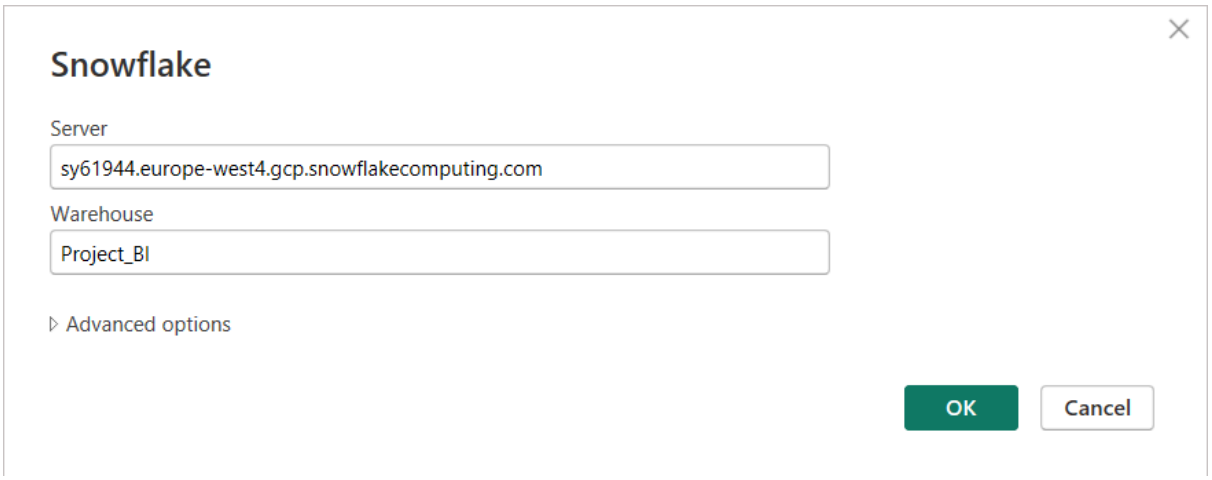
```
UPDATE DIM_PRODUCT
SET PRODUCT_NAME = REPLACE(PRODUCT_NAME, ',', '.');
```

refer to ***Manipulate_data.pdf*** file for **<u>detailed sql queries.</u>**

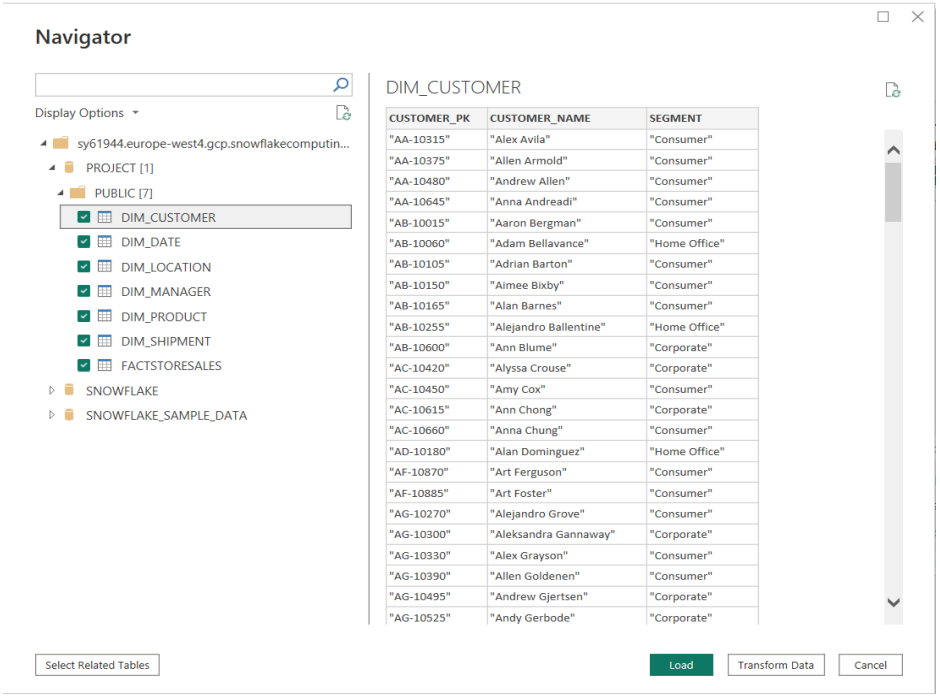
In the following screenshot, you can see the fact table and 6 dimension tables in our ‘PROJECT’ database in our Snowflake Data Warehouse:



Using **Power BI** we link to **Snowflake** warehouse **,** open the database “PROJECT” (which is the same database as “SUPERSTORE_SALES_DATA” in Majd’s Snowflake account) and load our tables:



Loading the fact and dimension tables into Microsoft Power BI:



2.4 Data Modeling

The resulting **dimension tables** are :

DIM_CUSTOMER:
- CUSTOMER_PK
- CUSTOMER_NAME
- SEGMENT

DIM_DATE:
- FULL_DATE

DIM_LOCATION:
- LOCATION_PK
- CITY
- POSTAL_CODE
- REGION
- STATE

DIM_MANAGER:
- MANAGER_PK
- MANAGER_FIRST_NAME
- MANAGER_LAST_NAME

DIM_PRODUCT:
- PRODUCT_PK
- PRODUCT_ID
- CATEGORY
- PRODUCT_NAME
- SUBCATEGORY

DIM_SHIPMENT:
- SHIP_MODE

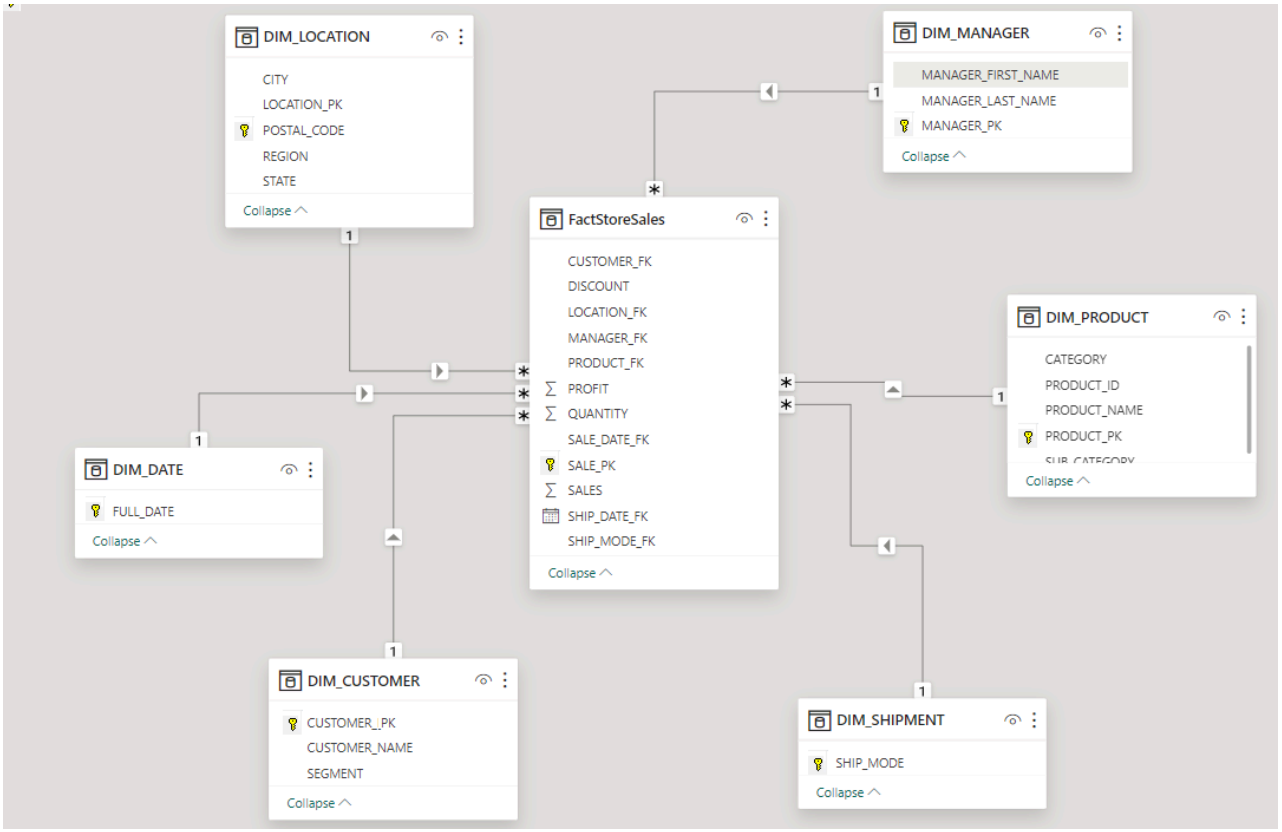
The **fact table**:

FactStoreSales:
- SALE_PK
- SALE_DATE_FK
- SHIP_DATE_FK
- SHIP_MODE_FK
- CUSTOMER_FK
- LOCATION_FK
- MANAGER_FK
- SHIPMENT_FK
- SALES
- QUANTITY
- PROFIT
- DISCOUNT

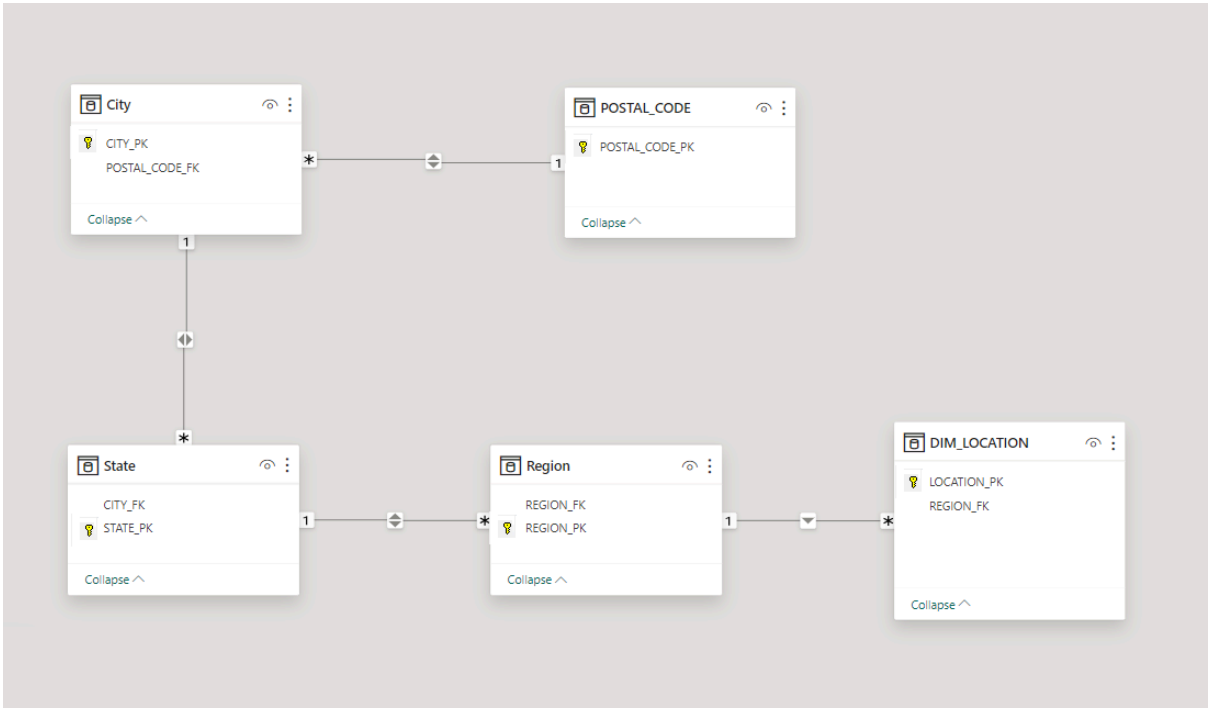
The **measures**:

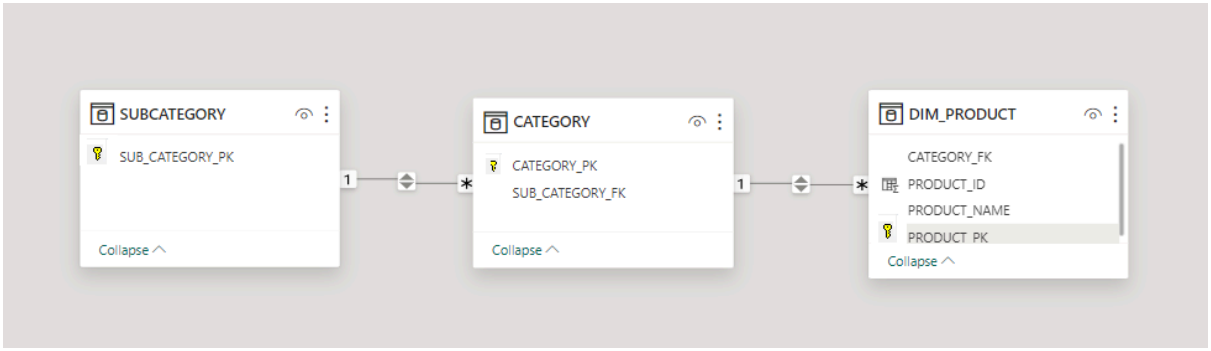
- Total Sales
- Total Quantity
- Total Profit
- Discount

The resulting **Star Schema**:



We chose to work with a **Star Schema instead of a Snowflake Schema** as it is faster and recommended for our analysis in Microsoft Power BI. In fact, Data Analysis Expressions (DAX) are the heart of Power BI's calculations and measures. A star schema simplifies DAX expressions, making them more readable and faster to write.
However, we can **normalize** our data structure and transform it to a **Snowflake Schema**:


City
CITY_PK
POSTAL_CODE_FK
Collapse
POSTAL_CODE
POSTAL_CODE_PK
Collapse
State
CITY_FK
STATE_PK
Collapse
Region
REGION_FK
REGION_PK
Collapse
DIM_LOCATION
LOCATION_PK
REGION_FK
Collapse


SUBCATEGORY
SUB_CATEGORY_PK
Collapse
CATEGORY
CATEGORY_PK
SUB_CATEGORY_FK
Collapse
DIM_PRODUCT
CATEGORY_FK
PRODUCT_ID
PRODUCT_NAME
PRODUCT_PK
Collapse

2.5 ROLAP Process:

We use Snowflake as our data warehouse and **SnowSQL** as our **ROLAP manipulation language**. Our ROLAP system queries data from relational tables in Snowflake and generates analytical results on demand, enabling us to perform agile and versatile analysis.
Below are examples of queries we wrote to aggregate information into insightful business intelligence reports that will help find patterns and trends

Discount amount grouped by category and sub_category:

```sql
// discount amount grouped by category and sub category
SELECT
    dp.CATEGORY,
    dp.SUB_CATEGORY,
    SUM(fs.DISCOUNT * fs.SALES) AS discount_amount
FROM
    FactStoreSales fs
JOIN
    DIM_PRODUCT dp ON fs.PRODUCT_FK = dp.PRODUCT_PK
GROUP BY
    dp.CATEGORY,
    dp.SUB_CATEGORY
ORDER BY
    dp.CATEGORY,
    dp.SUB_CATEGORY;
```

Results | Chart

	CATEGORY	SUB_CATEGORY	DISCOUNT_AMOUNT
1	Furniture	Bookcases	20800.591041
2	Furniture	Chairs	49814.8001
3	Furniture	Furnishings	8708.9528
4	Furniture	Tables	44192.1971
5	Office Supplies	Appliances	7490.2997
6	Office Supplies	Art	1820.8304
7	Office Supplies	Binders	43745.0793

Query Details
Query duration 1.1s
Rows 17
Query ID 01b1decc-0000-d91c-0...

CATEGORY
Office Supplies 9

Average shipping time for each shipping mode :

```sql
// average shipping time for each shipping mode
SELECT
    SHIP_MODE_FK,
    AVG(fs.SHIP_DATE_FK - fs.SALE_DATE_FK) AS avg_shipping_time
FROM
    FactStoreSales fs
GROUP BY
    fs.SHIP_MODE_FK
ORDER BY
    avg_shipping_time DESC;
```

Results | Chart

	SHIP_MODE_FK	AVG_SHIPPING_TIME
1	Standard Class	5.007205
2	Second Class	3.238046
3	First Class	2.182055
4	Same Day	0.044199

Query Details
Query duration 252ms
Rows 4
Query ID 01b1dee3-0000-d91c-0...

SHIP_MODE_FK

Display monthly sales over time Including the total sales and the number of distinct customers:

```sql
// Display monthly sales over time, including the total sales and the number of distinct customers
SELECT
    TO_CHAR(dd.FULL_DATE_PK, 'YYYY-MM') AS month,
    COUNT(DISTINCT fs.CUSTOMER_FK) AS distinct_customers,
    SUM(fs.SALES) AS total_sales
FROM
    DIM_DATE dd
LEFT JOIN
    FactStoreSales fs ON dd.FULL_DATE_PK = fs.SALE_DATE_FK
GROUP BY
    month
ORDER BY
    month;
```

Results | Chart

	MONTH	DISTINCT_CUSTOMERS	TOTAL_SALES
1	2019-01	32	14236.895
2	2019-02	27	4519.892
3	2019-03	69	55691.009
4	2019-04	64	28295.345
5	2019-05	67	23648.287
6	2019-06	63	34595.1276
7	2019-07	65	33946.393
8	2019-08	70	27909.4685
9	2019-09	118	81777.3508

Query Details

Query duration 573ms

Rows 60

Query ID 01b1decf-0000-d91c-0...

MONTH

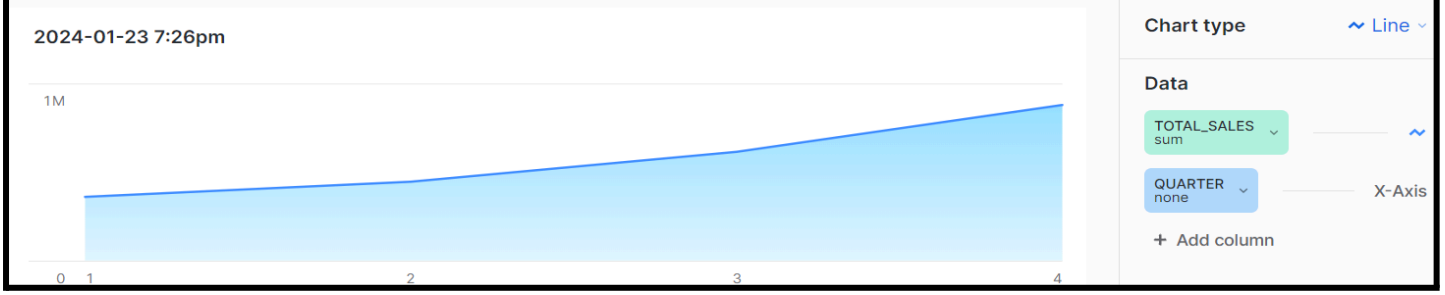
100% filled

DISTINCT_CUSTOMERS

Display quarter sales to identify present patterns :

```sql
// display quarter sales to identify present patters
SELECT
    EXTRACT(QUARTER FROM dd.FULL_DATE_PK) AS quarter,
    SUM(fs.SALES) AS total_sales
FROM
    DIM_DATE dd
LEFT JOIN
    FactStoreSales fs ON dd.FULL_DATE_PK = fs.SALE_DATE_FK
GROUP BY
    quarter
ORDER BY
    quarter;
```

Results | Chart



Display the total profit per margin:

```sql
//Display the total profit per manager
SELECT
    dm.MANAGER_FIRST_NAME,
    dm.MANAGER_LAST_NAME,
    SUM(fs.PROFIT) AS total_profit
FROM
    factstoresales fs
JOIN
    DIM_MANAGER dm ON fs.MANAGER_FK = dm.MANAGER_PK
GROUP BY
    dm.MANAGER_FIRST_NAME,
    dm.MANAGER_LAST_NAME;
```

Results | Chart

	MANAGER_FIRST_NAME	MANAGER_LAST_NAME	TOTAL_PROFIT
1	Fred	Suzuki	46749.4303
2	Sadie	Pawthorne	108418.4489
3	Roxanne	Rodriguez	39706.3625
4	Chuck	Magee	91522.78

Query Details
Query duration 29ms
Rows 4
Query ID 01b1de20-0000-d885-...

Total sales, average profit total quantity, minimum quantity, maximum quantity, and number of distinct customers for each city:

```sql
//total sales, average profit, total quantity, minimum quantity, maximum quantity, and number of distinct customers for each city
SELECT
    dl.CITY,
    SUM(fs.SALES) AS total_sales,
    AVG(fs.PROFIT) AS average_profit,
    SUM(fs.QUANTITY) AS total_quantity,
    MIN(fs.QUANTITY) AS min_quantity,
    MAX(fs.QUANTITY) AS max_quantity,
    COUNT(DISTINCT fs.CUSTOMER_FK) AS distinct_customers
FROM
    FactStoreSales fs
```

Results | Chart

	CITY	TOTAL_SALES	AVERAGE_PROFIT	TOTAL_QUANTITY	MIN_QUANTITY	MAX_QUAN
1	New York City	256368.161	67.799982186	3417	1	
2	Los Angeles	175851.341	40.75067992	2879	1	
3	Seattle	119540.742	68.121721262	1590	1	
4	San Francisco	112669.092	34.328206667	1935	1	
5	Philadelphia	109077.013	-25.768654376	1981	1	
6	Houston	64504.7604	-26.93248939	1466	1	
7	Chicago	48539.541	-21.192894268	1132	1	
8	San Diego	47521.029	37.512917647	670	1	
9	Jacksonville	44713.183	-18.59068	429	1	
10	Springfield	43054.342	38.041088344	649	1	

Query Details
Query duration 20ms
Rows 531
Query ID 01b1deee-0000-d91c-0...

CITY
100% filled

TOTAL_SALES
1.392 256368.161

```sql
//total sales, average profit, total quantity, minimum quantity, maximum quantity, and number of distinct customers for each city
SELECT
    dl.CITY,
    SUM(fs.SALES) AS total_sales,
    AVG(fs.PROFIT) AS average_profit,
    SUM(fs.QUANTITY) AS total_quantity,
    MIN(fs.QUANTITY) AS min_quantity,
    MAX(fs.QUANTITY) AS max_quantity,
    COUNT(DISTINCT fs.CUSTOMER_FK) AS distinct_customers
FROM
    FactStoreSales fs
JOIN
    DIM_LOCATION dl ON fs.LOCATION_FK = dl.LOCATION_PK
GROUP BY
    dl.CITY
ORDER BY
    total_sales DESC;
```

Results | Chart

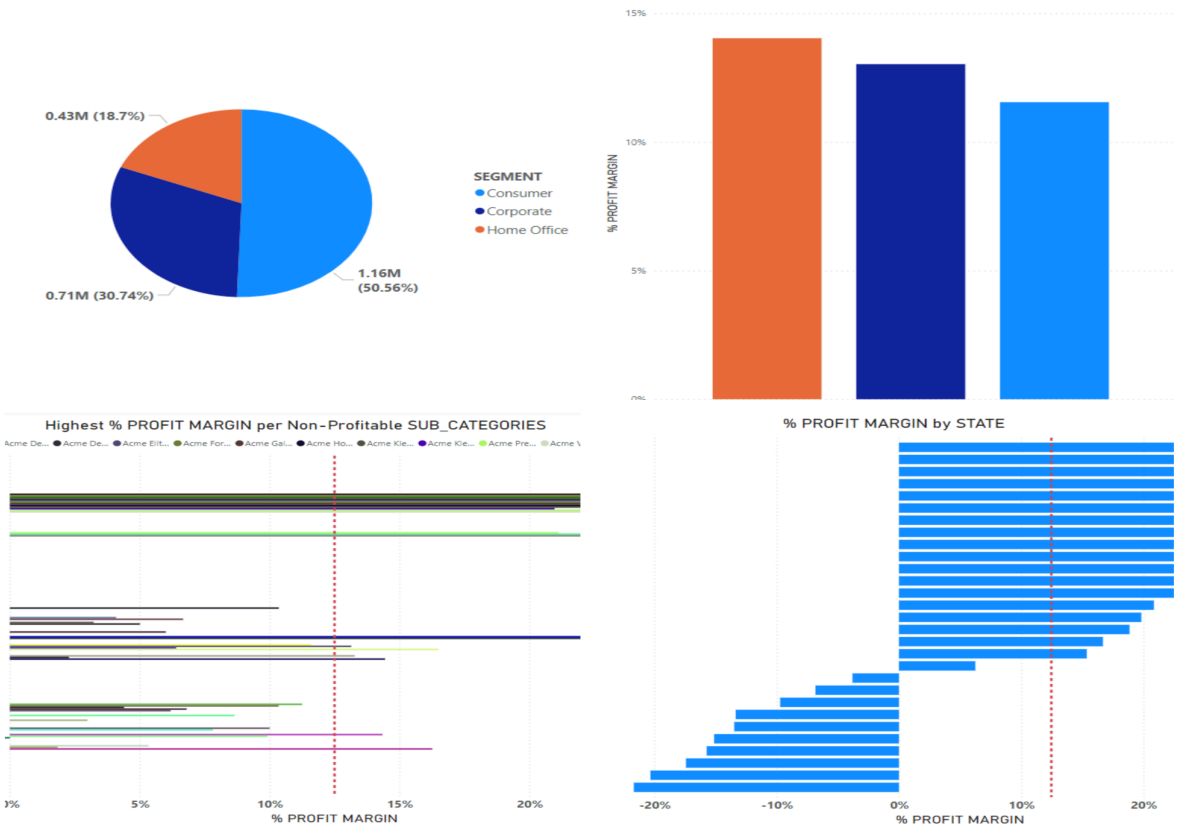
	AVERAGE_PROFIT	TOTAL_QUANTITY	MIN_QUANTITY	MAX_QUANTITY	DISTINCT_CUSTOMERS
513	1.2348	6	6	6	1
514	4.7304	3	3	3	1
515	3.792	3	3	3	1
516	3.6288	2	2	2	1
517	4.4955	1	1	1	1
518	-14.7708	2	2	2	1

AVERAGE_PROFIT
-200.61916 642.88575

TOTAL_QUANTITY
1 3417

3. Data Analysis & Recommandations

We used **Microsoft Power BI** to visualize and analyze our data.
Here is a sample of the visualizations we created with Microsoft Power BI:



Please refer to the "***Insights and Recommandations.pdf***" file in our project to find detailed insights with visuals.

After having performed an analysis on products, customer segments, geography, monthly sales, and managers, we recommend the management of the company to:

- Reassess the discount strategy, especially in the Supplies sub-category where high-margin items are sold at a loss;
- Consider to stop selling unprofitable products, especially within Tables and Bookcases subcategories, where negative profit margins are observed;
- Explore strategies to diversify product sales within the Consumer segment, to reduce its dependency on low profit-margin Furniture products;
- Consider targeted marketing and promotions for Technology items to capitalize on their high-profit margins;
- Allocate additional resources and marketing efforts in its top-performing states, namely California, New York, and Texas, which collectively contribute to most of the Superstore's revenue;

- Analyze the factors causing low or negative profit margins in low-performing states. The company can:
    - Implement targeted strategies to improve profitability, such as adjusting pricing, refining product offerings, or optimizing supply chain operations;
    - Close its stores and stop operating in these states.
- The company should devise strategies to cope with the seasonal fluctuations in revenue, especially during the first quarter of the year, when sales are typically low;
- Evaluate the factors influencing the low profit margins in the South and Central regions. The company can:
    - Provide additional training and/or resources to these managers to help improve regional profitability;
    - Replace them with new managers.