Master's in Data Science and Engineering

Data Warehouses

Developed By:
Bruno Fernandes - up202108871
Hugo Abelheira - up202409899
Tiago Coelho - up202105004

April 1, 2025

Contents

List of Figures

1 Introduction: Unveiling the Potential of Data-Driven Decision Making

1.1 Project Context and Objectives

In today's dynamic business environment, data-driven decision-making is paramount for sustained growth and competitive advantage. The Super Store Data Warehouse project was conceived to address the growing need for a centralized, integrated, and analytical data repository. By consolidating data from disparate operational systems into a unified dimensional model, the Data Warehouse enables stakeholders to gain actionable insights into sales patterns, customer behavior, and product performance.

The primary objectives of this project were:

- **Dimensional Modeling:** To design and implement a dimensional model that facilitates efficient and intuitive data analysis.
- **Consolidation of Data:** To extract, transform, and load data from the Superstore dataset into a structured Data Warehouse.
- **Business Intelligence:** To empower stakeholders with the ability to generate meaningful reports and visualizations for informed decision-making.

1.2 The Power of Dimensional Modeling

Dimensional modeling, a cornerstone of Data Warehouse design, provides a structured approach to organizing data for analytical purposes. By denormalizing data into fact and dimension tables, dimensional models optimize query performance and enhance data accessibility. The dimensional bus matrix served as a blueprint for the Data Warehouse, outlining the key business processes and their corresponding dimensions and facts.

2 Strategic Planning: The Blueprint for Data Integration

2.1 Unpacking the Dimensional Bus Matrix:

The dimensional bus matrix, a crucial artifact in the design phase, outlined the core business processes of the Superstore and their associated dimensions. This matrix facilitated the identification of conformed dimensions, ensuring consistency and uniformity across different business processes.

SuperStore Data Warehouse

Dimensional bus matrix

Data mart	Star	Dimension	Calendar	CalendarMonth	Customer	Shipping	Location	Product	Category	State	Region
Sales	Item		x		X		X	X			
	Orders		X		X	X	X				
	OrderM			X						X	
	ProductPerformance			X					X	X	
	ShippingBehavior					X			X		X
	ShippingBehaviorS					X			X	X	

Figure 1: Bus Matrix of our Super Store

Our matrix centers on a "Sales" data mart, built around four main fact tables: Item, Orders, OrderM, and ProductPerformance, alongside ShippingBehavior and ShippingBehaviorS for shipping analysis.

- **Fact Tables and Dimensions:**
  - The Item fact table, the most detailed, links to Calendar, Customer, Location, and Product.
  - Orders aggregates data at the order level, using the same dimensions as Item, plus Shipping.
  - OrderM focuses on monthly state-level analysis, using CalendarMonth and State.
  - ProductPerformance analyses product performance by category and state monthly, using CalendarMonth, Category and State.
  - ShippingBehavior and ShippingBehaviorS analyses the shipping behavior, using Shipping, Location, Category and State.
- **Conformed Dimensions:** Calendar, Customer, Location, and State are conformed, ensuring consistent analysis across tables.
- **Granularity:** The matrix supports various levels of detail, from individual items to monthly state summaries.
- **Shipping Analysis:** Dedicated fact tables emphasize shipping behavior analysis, crucial for logistics optimization.

2.2 Data Dictionaries: A Foundation for Data Understanding

Data dictionaries are vital for a Data Warehouse, detailing each dimension and fact table's attributes, definitions, and relationships. They ensure clarity, consistency, and facilitate effective data analysis.

2.2.1 Dimensions Dictionaries

The dimension data dictionaries, covering tables like Calendar, Customer, and Product, are crucial for ensuring clarity and consistency within the Data Warehouse. They provide detailed definitions for each attribute, specifying its purpose, data type, and role as a key. This level of detail is essential for accurate data interpretation and usage across the organization.

Furthermore, these dictionaries include information about hierarchical relationships and Slowly Changing Dimension (SCD) types, which are vital for understanding data evolution and table relationships. By standardizing attribute definitions and data types, they promote data governance and ensure consistent data usage. Ultimately, these dictionaries serve as indispensable tools for data analysts and business users, facilitating efficient data retrieval, analysis, and informed decision-making.

Name	Description	SCD	Version	1.0	Date	2025-03-31
Calendar	Stores information about transaction and shipping dates	Type 1	Hierarchy	Day < Month < Year		
Attribute	Description	Level	Key	Type	Size	Precision
calendar_id	Unique identifier for date	Date	PK	ID		
full_date	Full date (YYYY-MM-DD)	Date	UK	DATE		
year_id	Year Surrogate	Year	LK	INT		
year_number	Year of the date	Year	UK	INT		0
month_id	Month Surrogate	Month	LK	INT		
month_number	Month of the date	Month	UK	INT		0
month_name	Name of the month	Month		VARCHAR	15	
day_id	Day Surrogate	Day	LK	INT		
day_number	Day of the date	Day	UK	INT		0

Figure 2: Calendar Dimension Dictionary

2.2.2 Fact Tables Dictionaries

The fact table data dictionaries, encompassing tables such as Item, Orders, and OrderM, are crucial for understanding the quantitative measures and analytical focus of the Data Warehouse. They provide detailed definitions for each measure, specifying its calculation, unit of measurement, and business significance. This ensures that all users interpret the factual data consistently and accurately.

For example, the "sales" measure in the Item fact table is defined as the "price of the transaction (product price * quantity * (1 - discount))," providing clarity on its calculation. Similarly, the "lost_value" measure is defined as the "difference between full price of the product and the discounted price," highlighting its relevance for loss analysis. Furthermore, these dictionaries clearly outline the dimensions associated with each fact table, clarifying the context in which the measures should be analyzed. This ensures that users understand the granularity and scope of the data, facilitating meaningful analysis and informed decision-making.

Star	**Orders**	Version	1.0	Date	2025-03-31
Granularity	One order instance				
Dimensions					
OrderCalendar	Calendar				
ShippingCalendar	Calendar				
Customer	Customer				
Location	Location				
Shipping	Shipping				
Measures					
sales_order	Total price of the order (an order may contain different products)				
quantity_order	Total number of products in the order				
order_code	Identifier of the order (degenerate)				
lost_value_order	Difference between full price of the order and the discounted price				
profit_order	Total profit of the order				

Figure 3: Orders Fact Table Dictionary

3 The Dimensional Model: The Architecture of Information

3.1 Star Schemas

The dimensional model employed a star schema, characterized by a central fact table surrounded by dimension tables. This schema optimized query performance and simplified data analysis. The model comprised the following key tables:

- **Fact Tables:** Item, Orders, OrderM, ProductPerformance;
- **Dimensions:** Customer, Product, Location, Calendar, Shipping, State, Region, Category and CalendarMonth;

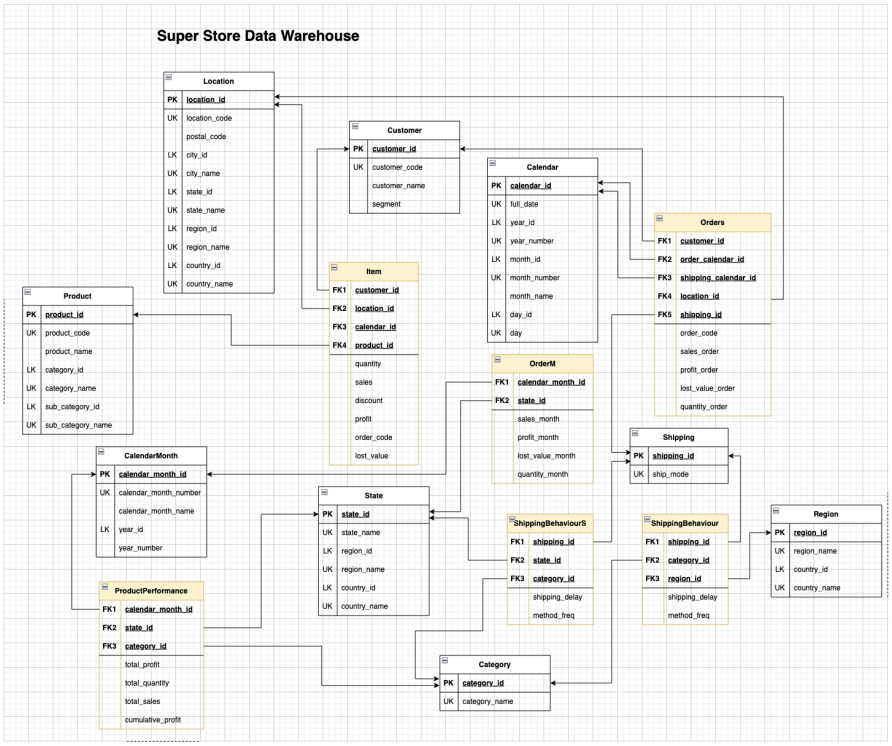

Figure 4: ETL Diagram of the Item Fact Table

3.2 Navigating the Dimensions

Each dimension table was meticulously designed to capture the relevant attributes and hierarchies. For instance, the Customer dimension included attributes such as customer name, segment, and location, while the Product dimension encompassed product name, category, and subcategory.

3.3 The Facts: The Heart of Analysis

The fact tables, the core of the dimensional model, stored the quantitative measures of the business processes. The Item fact table, for example, recorded sales, quantity, discount and profit for each product sold. The Orders fact table recorded sales, quantity, lost value and profit for each order. The OrderM fact table recorded monthly sales, quantity, lost value and profit for each state. And the ProductPerformance fact table recorded monthly sales, quantity, profit and cumulative profit for each product category and state.

4 The Data Journey: Extraction, Transformation, and Loading (ETL)

4.1 Unveiling the Data Sources

The primary data source for this project was the Superstore dataset, a CSV file containing transactional data. The dataset was analyzed for data quality and consistency, and appropriate transformations were implemented to address any issues.

4.2 The Art of Transformation: Shaping the Data

The ETL process, implemented using Python and pandas, involved a series of transformations to prepare the data for loading into the Data Warehouse. These transformations included data cleansing, data standardization, and data aggregation. The ETL process was designed using BPMN diagrams, providing a visual representation of the data flow and transformations. The BPMN diagrams facilitated communication and collaboration among the development team.

4.3 ETL Overview Diagram

The ETL (Extract, Transform, Load) Overview Diagram provides a high-level visual representation of the data flow from the original data sources to the final Data Warehouse. This diagram captures the general sequence of ETL operations, highlighting the main extraction, transformation, and loading stages, as well as the dependencies between different processes. It serves as an essential guide for understanding the overall ETL architecture and how data is moved and transformed throughout the process.

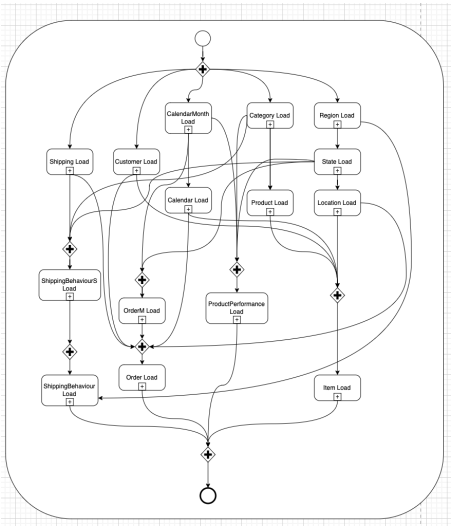

Figure 5: ETL Overview

4.4 Process Specific ETL Diagrams

The ETL diagram for the "Item" table details the transformation and loading process for this fact table. It illustrates the data flow from the source (an Excel file) to the "Item" table in the Data Warehouse. The diagram highlights lookup steps to retrieve foreign keys from dimension tables (Customer, Calendar, Location, and Product) based on input data. It also shows error handling for unmatched records, logged to an error file. Finally, it depicts the insertion of transformed data into the "Item" table. This diagram provides detailed documentation for the "Item" table's ETL process, aiding in understanding, maintenance, and debugging. Similar diagrams were developed for each table (dimension and fact), offering a comprehensive view of the ETL pipeline.

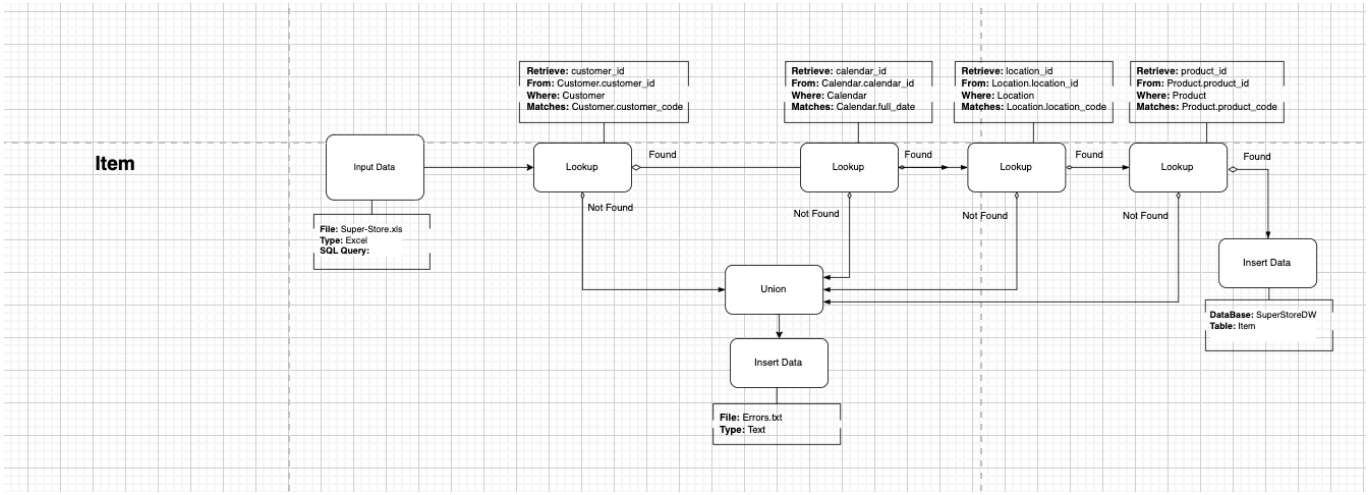

Figure 6: ETL Diagram of the Item Fact Table

4.5 Loading the Data Warehouse: The Destination

The transformed data was loaded into a MySQL database, the chosen platform for the Data Warehouse. The loading process was optimized for performance, and data integrity was ensured through referential constraints and data validation.

5 Exploring the Data Universe: Queries and Analysis

5.1 The Power of Queries: Extracting Knowledge

SQL queries were used to extract meaningful insights from the Data Warehouse. These queries enabled stakeholders to analyze sales trends, identify customer segments, and evaluate product performance.

5.2 Unveiling the Patterns: Data Analysis

Data analysis techniques, such as aggregation, filtering, and sorting, were employed to uncover patterns and trends in the data. For instance, sales data was analyzed to identify top-performing products and customer segments.

5.3 Queries Examples

5.3.1 Query 1

Visualize the products with the highest sales volume, which can guide inventory strategies, marketing, and demand analysis.

```sql
SELECT
    p.product_name,
    SUM(i.quantity) AS total_quantity
FROM Item i
JOIN Product p ON i.product_id = p.product_id
GROUP BY p.product_name
ORDER BY total_quantity DESC
LIMIT 10;
```

Listing 1: SQL Query for Monthly Sales

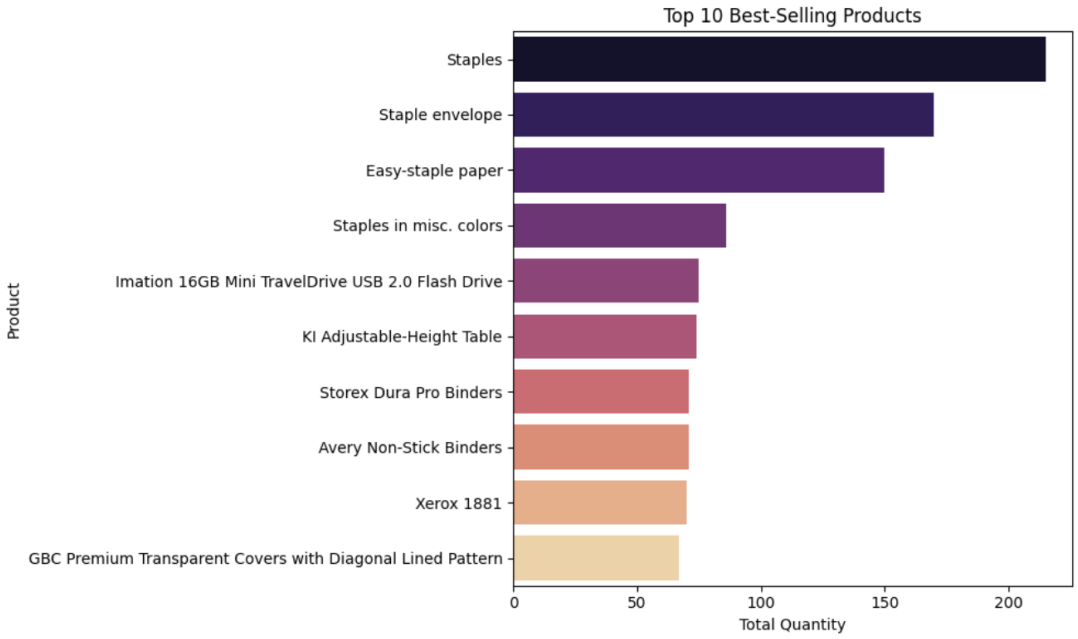

Figure 7: Query 1 Visualization

5.3.2 Query 2

The chart shows, for a specific state, how sales accumulate over time. This visualization is useful for understanding trends and revenue progression by state.

```sql
SELECT
    s.state_name,
    c.full_date,
    SUM(o.sales_order) OVER (PARTITION BY s.state_name ORDER BY c.full_date) AS
        running_total
FROM Orders o
JOIN Location l ON o.location_id = l.location_id
JOIN State s ON l.state_id = s.state_id
JOIN Calendar c ON o.order_calendar_id = c.calendar_id
ORDER BY s.state_name, c.full_date;
```

Listing 2: SQL Query for Monthly Sales

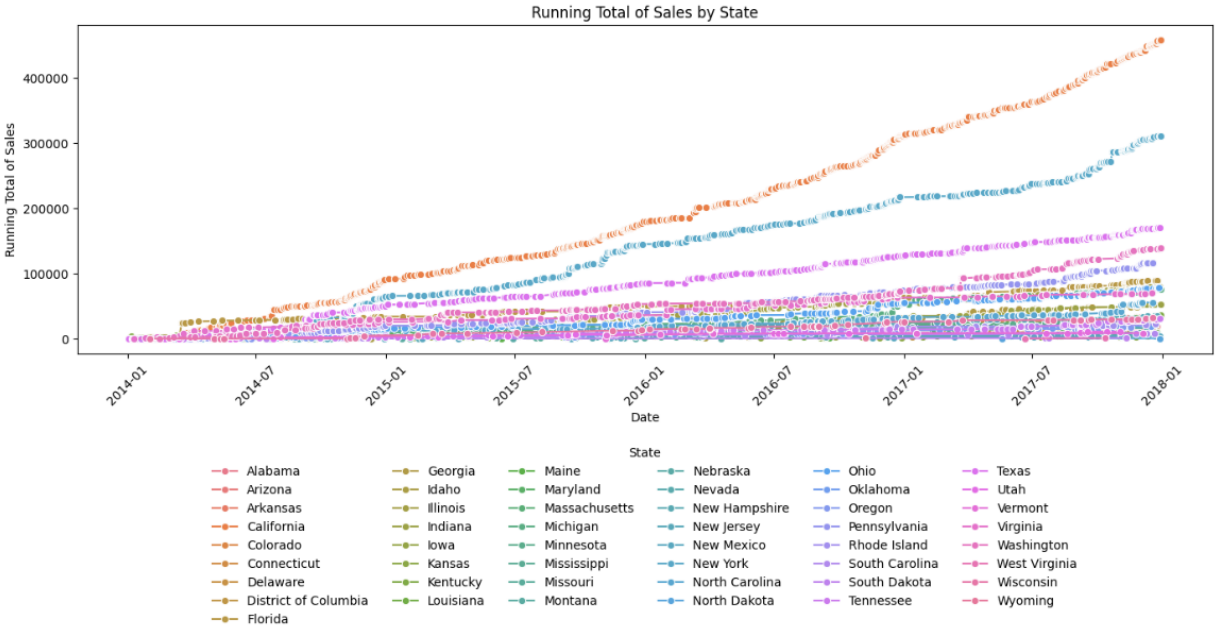

Figure 8: Query 2 Visualization

5.3.3 Query 3

It allows identifying which shipping method has the highest or lowest average time between order and shipment, helping to analyze logistical efficiency.

```sql
SELECT
    sh.ship_mode,
    AVG(DATEDIFF(c2.full_date, c1.full_date)) AS average_delivery_time
FROM Orders o
JOIN Shipping sh ON o.shipping_id = sh.shipping_id
JOIN Calendar c1 ON o.order_calendar_id = c1.calendar_id
JOIN Calendar c2 ON o.shipping_calendar_id = c2.calendar_id
GROUP BY sh.ship_mode
ORDER BY average_delivery_time;
```

Listing 3: SQL Query for Monthly Sales

Figure 9: Query 3 Visualization

6 Visualization: Transforming Data into Information

6.1 The Power of Visualization: Telling Stories with Data

PowerBI, a data visualization tool, was used to create interactive dashboards and reports. This visualization enabled stakeholders to explore the data and gain actionable insights.

6.2 Interactive Dashboards: The Business Vision

Interactive dashboards provided a comprehensive view of the business, allowing stakeholders to monitor key performance indicators (KPIs) and track progress toward goals. Additionally, we created specific files for visualization, such as generating an Excel file with multiple sheets to adapt the data format for PowerBI compatibility.

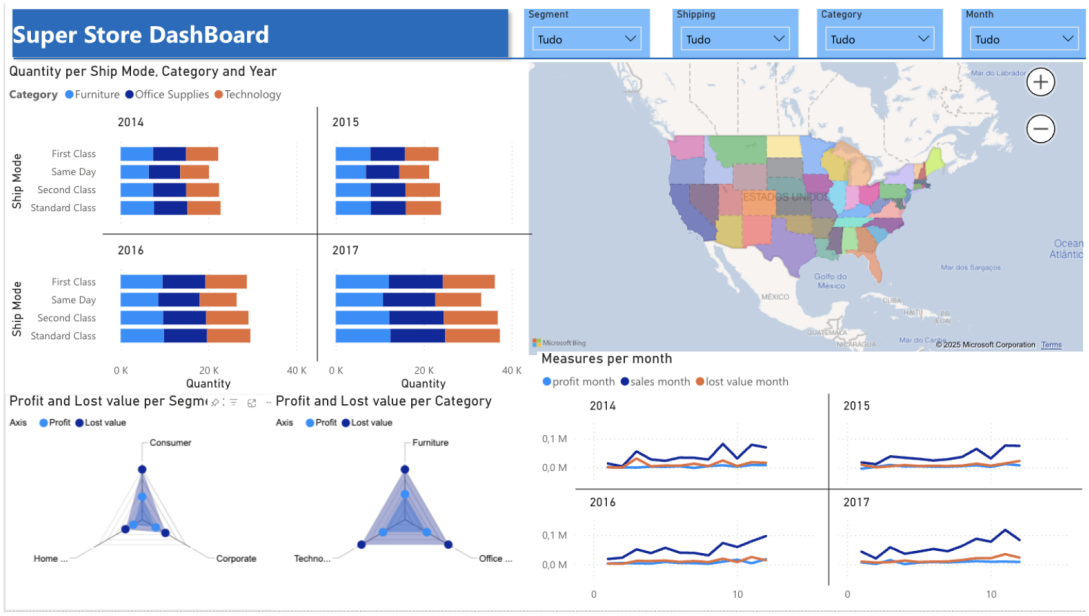

Figure 10: PowerBI Visualization

7 Final Reflections: Learnings and Challenges

7.1 The Legacy of the Data Warehouse: Advantages and Challenges

The Data Warehouse offered several advantages over operational databases, including improved query performance, enhanced data accessibility, and support for complex analytical queries. However, the project also faced challenges, such as data quality issues and the complexity of the ETL process.

7.2 The Future of Data Analysis: Expanding Horizons

The field of data analysis is constantly evolving, with new tools and techniques emerging regularly. The Super Store Data Warehouse project provides a solid foundation for future data analysis initiatives.

7.3 Conclusion

The development of the Super Store Data Warehouse has demonstrated the transformative power of structured data integration in enabling data-driven decision-making. By consolidating disparate operational data into a well-designed dimensional model, the project has enhanced data accessibility, optimized query performance, and provided stakeholders with valuable insights into sales trends, customer behavior, and logistics efficiency. Despite the challenges encountered—such as ensuring data quality and managing the complexity of the ETL process—these obstacles were addressed through meticulous data modeling, robust transformation pipelines, and strategic planning. The result is a scalable and efficient analytical platform that empowers business intelligence initiatives.

Looking ahead, the continuous evolution of data analysis techniques, coupled with advancements in cloud computing, artificial intelligence, and real-time analytics, presents new opportunities for further enhancing the Data Warehouse. Future improvements could involve integrating predictive analytics, automating ETL processes, and incorporating more diverse data sources to refine decision-making capabilities. The foundation established by this project serves as a stepping stone for future innovations, ensuring that businesses remain agile and competitive in an increasingly data-driven world.

A Appendices

A.1 Data Dictionaries

- Dimensions:

Name	Description	SCD	Version	1.0	Date	2025-03-31
CalendarMonth	Stores information about transaction and shipping dates	Type 1	Hierarchy	Month < Year		
Attribute	Description	Level	Key	Type	Size	Precision
calendar_month_id	Unique identifier for month	Month	PK	ID		
calendar_month_number	Month number	Date	UK	INT		0
calendar_month_name	Month name	Date	UK	VARCHAR	15	
year_id	Year Surrogate	Year	LK	NUMBER		
year_number	Year of the month	Date	UK	INT		0

Figure 11: Calendar Month Dimension Dictionary

Name	Description	SCD	Version	1.0	Date	2025-03-31
Customer	Stores information about the customers and their segments	Type 1	Hierarchy			
Attribute	Description	Level	Key	Type	Size	Precision
customer_id	Identifier for customer	Customer	PK	ID	20	
customer_code	Code for the customer	Customer	UK	VARCHAR	50	
customer_name	Name of the customer	Customer		VARCHAR	100	
segment	Customer segment	Customer		VARCHAR	20	

Figure 12: Customer Dimension Dictionary

Name	Description	SCD	Version	1.0	Date	2025-03-31
Location	Stores geographical information related to customers	Type 1	Hierarchy	Postal_Code < City < State < Region < Country		
Attribute	Description	Level	Key	Type	Size	Precision
location_id	Unique identifier for location	Location	PK	ID		
location_code	Code for the location	Location	UK	VARCHAR		
country_id	Country Surrogate	Country	LK	ID		
country_name	Customers' Country	Country	UK	VARCHAR	50	
region_id	Region Surrogate	Region	LK	ID		
region_name	Customers' Region	Region	UK	VARCHAR	50	
state_id	State Surrogate	State	LK	ID		
state_name	Customers' State	State	UK	VARCHAR	50	
city_id	City Surrogate	City	LK	ID		
city_name	Customers' City	City	UK	VARCHAR	50	
postal_code	Customers' Postal Code	Location		VARCHAR	15	

Figure 13: Location Dimension Dictionary

Name	Description	SCD	Version	1.0	Date	2025-03-31
Product	Stores details about the products sold	Type 1	Hierarchy	Sub-Category < Category		
Attribute	Description	Level	Key	Type	Size	Precision
product_id	Identifier for product	Product	PK	ID		
product_code	Product identification	Product	UK	VARCHAR	50	
product_name	Name of the product	Product		VARCHAR	100	
category_id	Category Surrogate	Category	LK	ID		
category_name	Product category	Category	UK	VARCHAR	50	
sub_category_id	Sub-Category Surrogate	Sub-Category	LK	ID		
sub_category_name	Product sub-category	Sub-Category	UK	VARCHAR	50	

Figure 14: Product Dimension Dictionary

Name	Description	SCD	Version	1.0	Date	2025-03-31
Shipping	Stores information about shipping modes	Type 1	Hierarchy			
Attribute	Description	Level	Key	Type	Size	Precision
shipping_id	Identifier for shipping mode	Shipping	PK	ID		
ship_mode	Shipping method	Shipping	UK	VARCHAR	50	

Figure 15: Shipping Dimension Dictionary

Name	Description	SCD	Version	1.0	Date	2025-03-31
Category	Stores information about the products' category	Type 1	Hierarchy			
Attribute	Description	Level	Key	Type	Size	Precision
category_id	Identifier for product category	Category	PK	ID		
category_name	Name of the category	Category	UK	VARCHAR	50	

Figure 16: Category Dimension Dictionary

Name	Description	SCD	Version	1.0	Date	2025-03-31
State	Stores information about the customer's state	Type 1	Hierarchy	State < Region < Country		
Attribute	Description	Level	Key	Type	Size	Precision
state_id	Identifier for state	State	PK	ID		
state_name	Name of the state	State	UK	VARCHAR	50	
region_id	Region Surrogate	Region	LK	ID		
region_name	Customers' Region	Region	UK	VARCHAR	50	
country_id	Country Surrogate	Country	LK	ID		
country_name	Customers' Country	Country	UK	VARCHAR	50	

Figure 17: State Dimension Dictionary

Name	Description	SCD	Version	1.0	Date	2025-03-31
Region	Stores information about the customer's region	Type 1	Hierarchy	Region < Country		
Attribute	Description	Level	Key	Type	Size	Precision
region_id	Region Surrogate	Region	PK	ID		
region_name	Customers' Region	Region	UK	VARCHAR	50	
country_id	Country Surrogate	Country	LK	ID		
country_name	Customers' Country	Country	UK	VARCHAR	50	

Figure 18: Region Dimension Dictionary

- Fact Tables:

Star	**Item**	Version	1.0	Date	2025-03-31
Granularity	Individual transaction of a product within an order				
Dimensions					
Customer	Customer				
Location	Location				
Calendar	Calendar				
Product	Product				
Measures					
quantity	The amount of products of that instance				
sales	The price of the transaction (product price * quantity * (1 - discount))				
discount	Discount applied to the item transaction				
order_code	Identifier of the order this item belongs to				
lost_value	Difference between full price of the product and the discounted price				
profit	Total profit of the transaction (Sales - cost)				

Figure 19: Item Fact Table Dictionary

Star	**OrderM**	Version	1.0	Date	2025-03-31
Granularity	Monthly aggregation of all orders by state				
Dimensions					
CalendarMonth	CalendarMonth				
State	State				
sales_month	Total revenue of the month				
quantity_month	Total number of products sold in the month				
lost_value_month	Difference between full price of the monthly orders without and with discount				
profit_month	Total profit of the month				

Figure 20: OrderM Fact Table Dictionary

Star	**ProductPerformance**	Version	1.0	Date	2025-03-31
Granularity	Product statistics per month and state				
Dimensions					
Category	Category				
State	State				
CalendarMonth	CalendarMonth				
Measures					
total_sales	Monthly revenue of a category of products				
total_profit	Monthly profit of a category of products				
cumulative_profit	Monthly cumulative profit for a category of products				
total_quantity	Quantity of products sold in each month				

Figure 21: ProductPerformance Fact Table Dictionary

Star	**ShippingBehaviour**	Version	1.0	Date	2025-03-31
Granularity	Frequency of shipping methods used per category and region				
Dimensions					
Shipping	Shipping				
Category	Category				
Region	Region				
Measures					
shipping_delay	Average difference between the order date and the shipping date				
method_freq	Number of times (absolute frequency) a shipping method is used				

Figure 22: ShippingBehaviour Fact Table Dictionary

Star	**ShippingBehaviourS**	Version	1.0	Date	2025-03-31
Granularity	Frequency of shipping methods used per category and state				
Dimensions					
Shipping	Shipping				
Category	Category				
State	State				
Measures					
shipping_delay	Average difference between the order date and the shipping date				
method_freq	Number of times (absolute frequency) a shipping method is used				

Figure 23: ShippingBehaviourS Fact Table Dictionary

A.2 ETL Diagrams

Below are all the remaining ETL process diagrams used in this project:

- Dimensions:

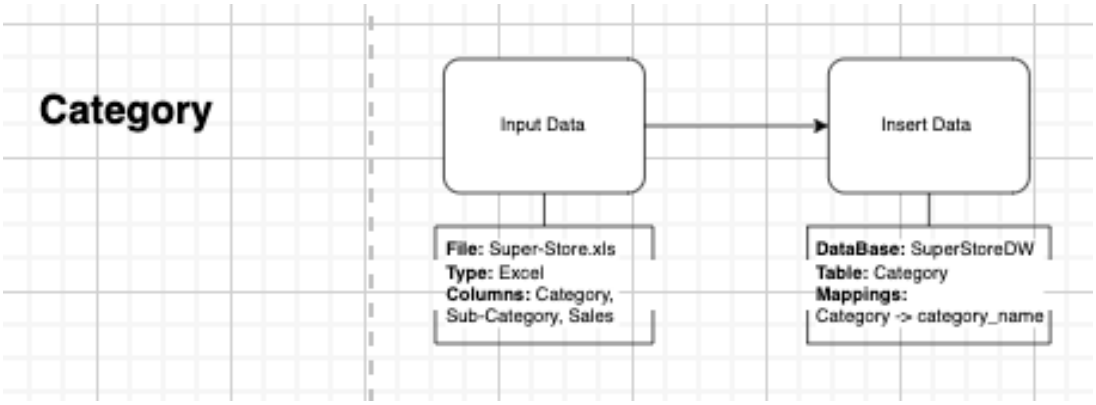

Figure 24: Category ETL Design

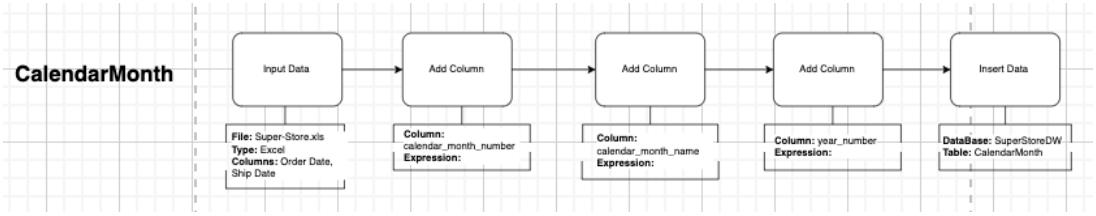

Figure 25: CalendarMonth ETL Design

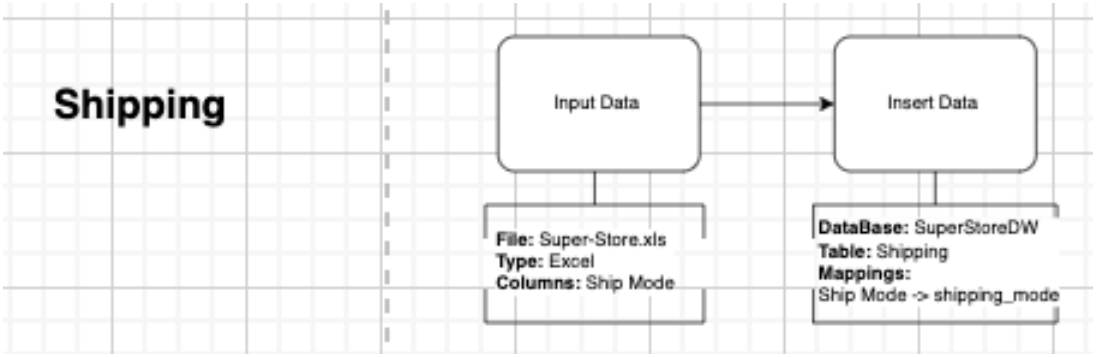

Figure 26: Shipping ETL Design

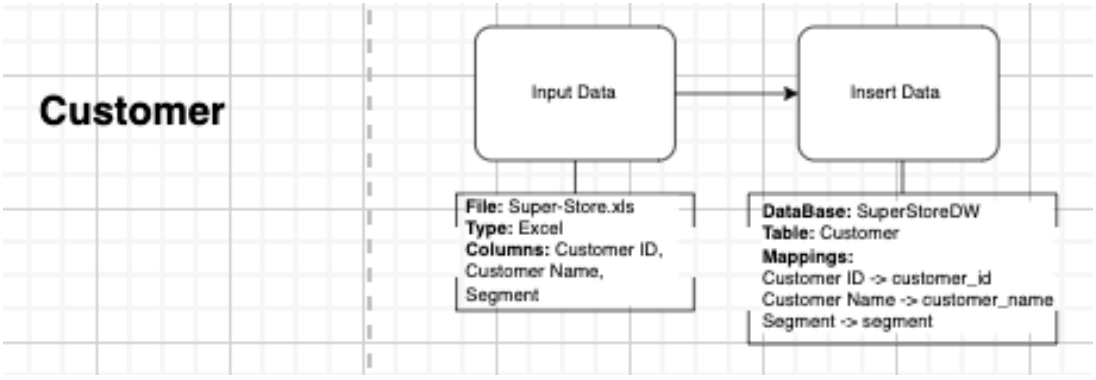

Figure 27: Customer ETL Design

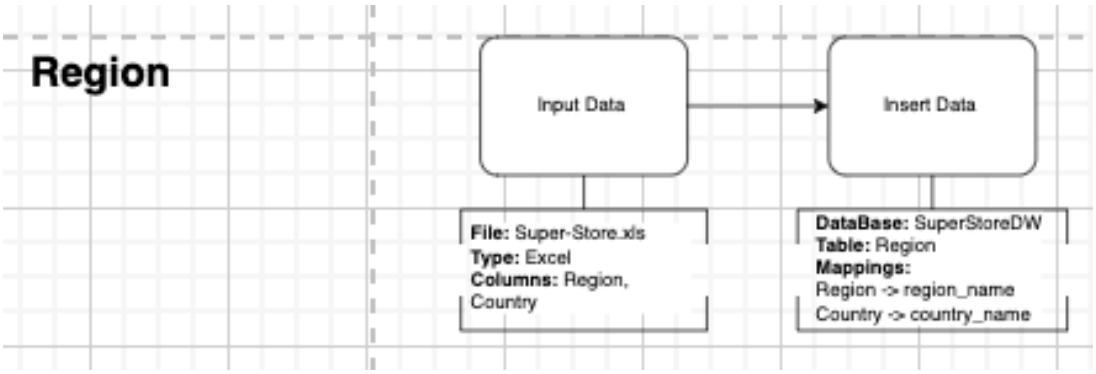

Figure 28: Region ETL Design

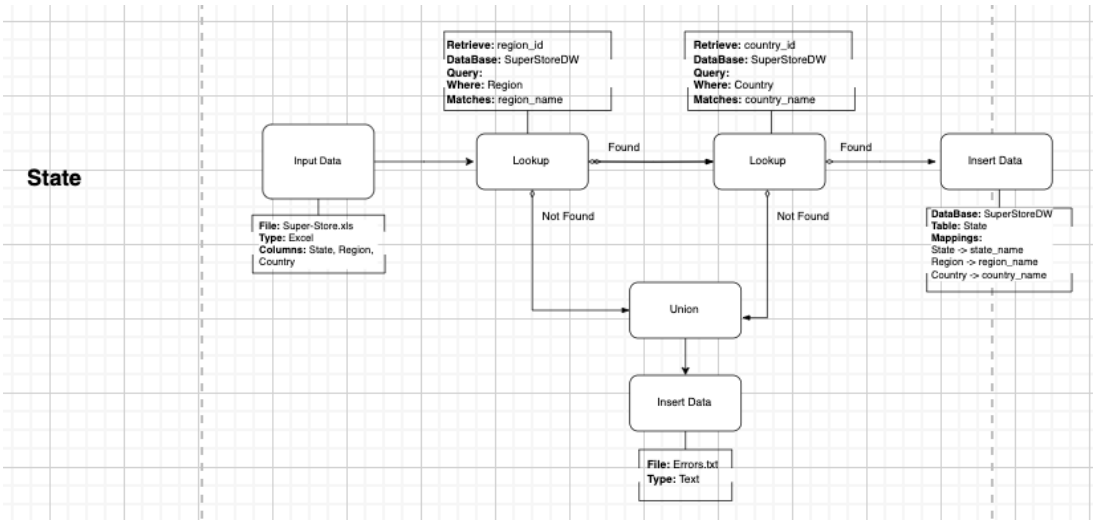

Figure 29: State ETL Design

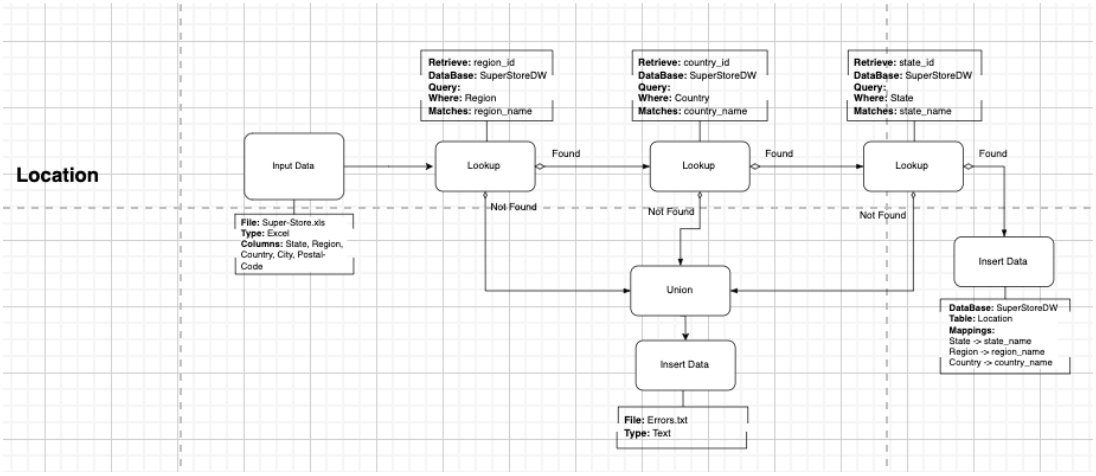

Figure 30: Location ETL Design

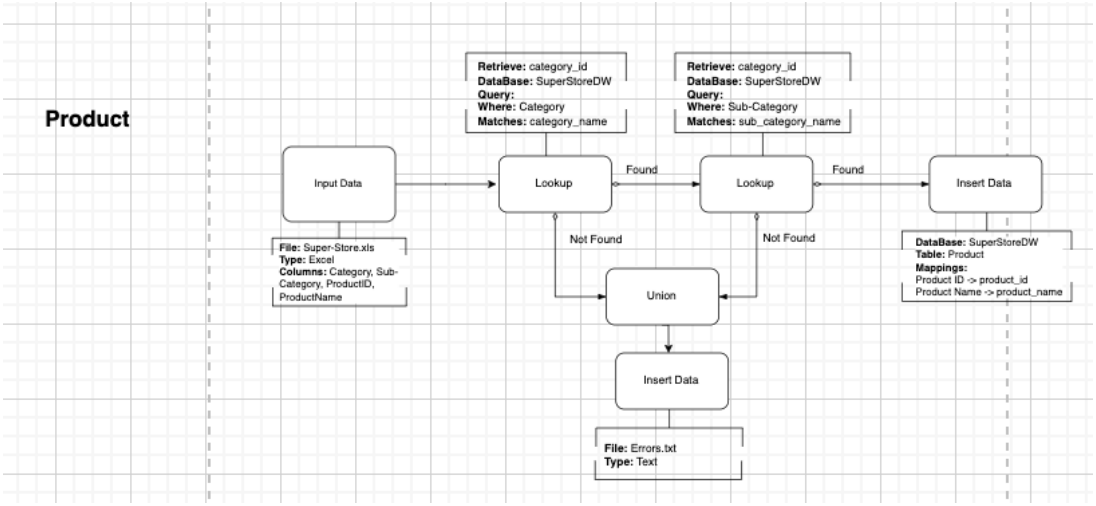

Figure 31: Product ETL Design

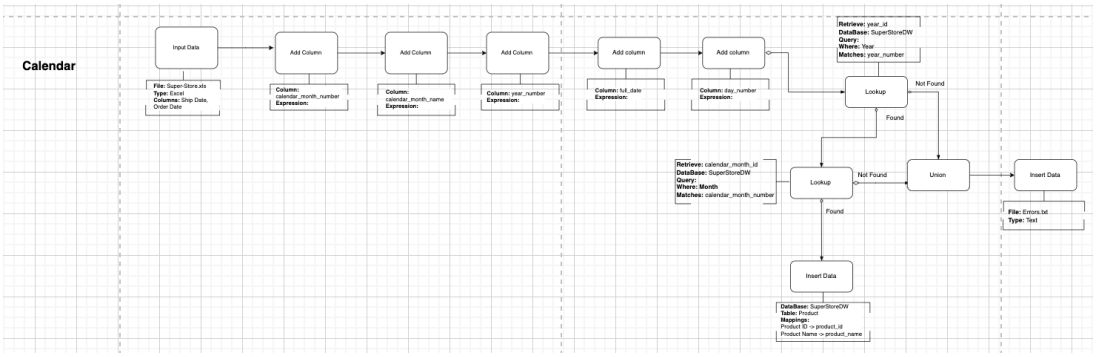

Figure 32: Calendar ETL Design

- Fact Tables:

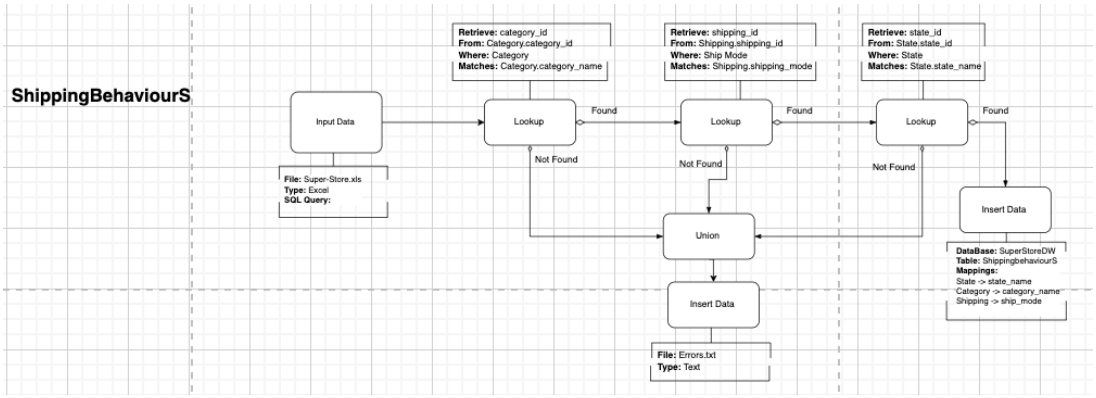

Figure 33: ShippingBehaviourS ETL Design

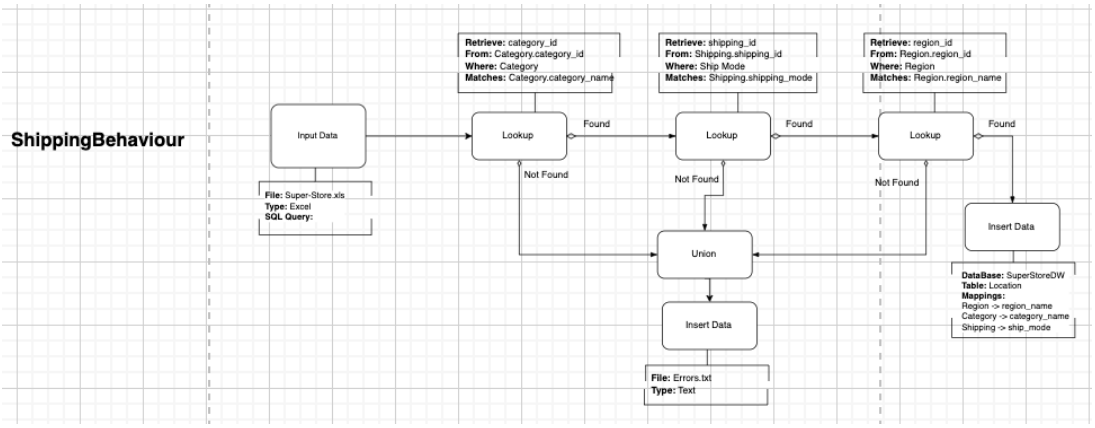

Figure 34: ShippingBehaviour ETL Design

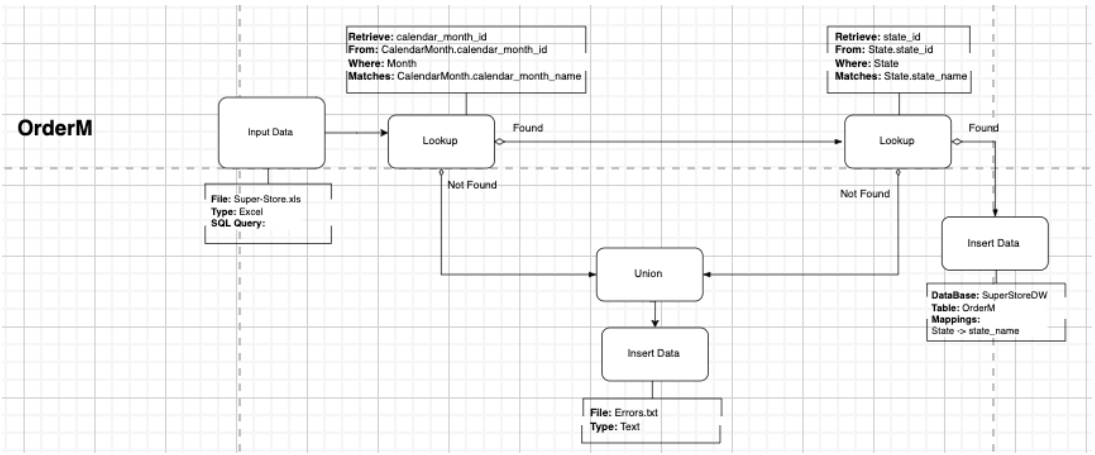

Figure 35: OrderM ETL Design

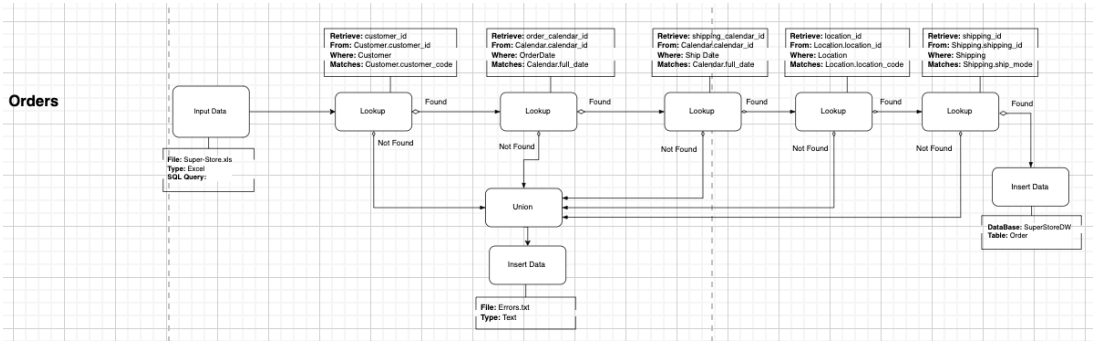

Figure 36: Orders ETL Design

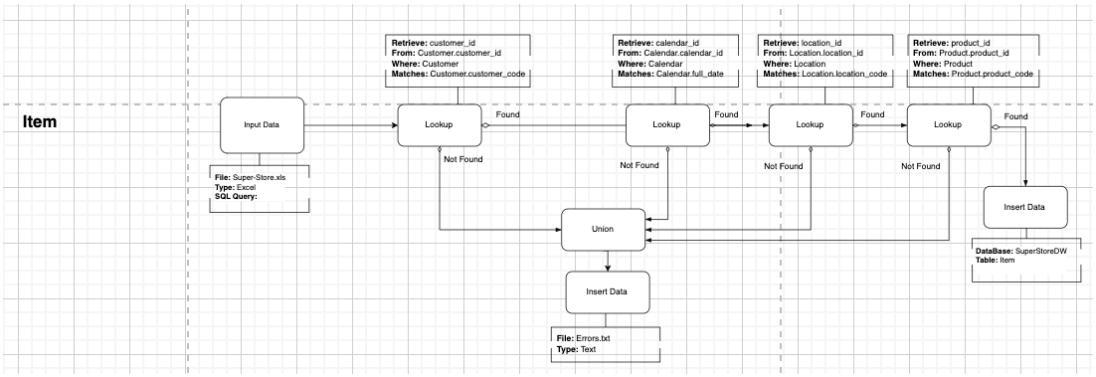

Figure 37: Item ETL Design

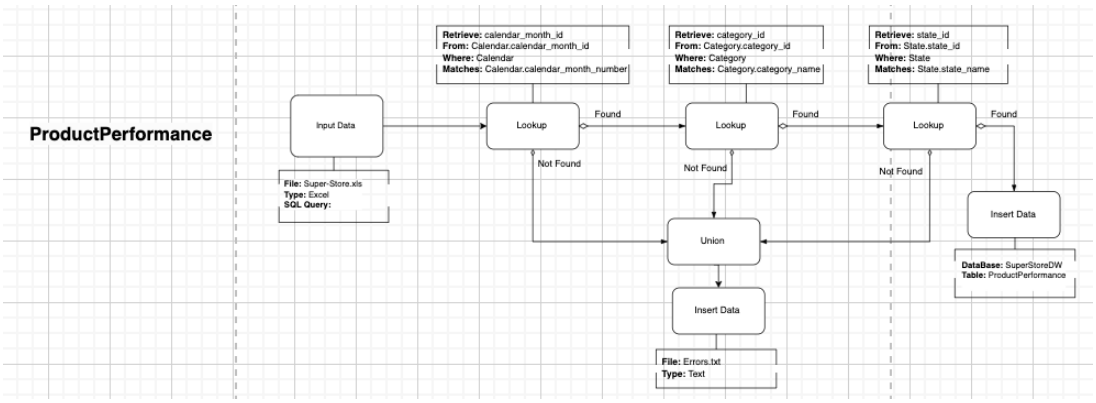

Figure 38: ProductPerformance ETL Design

A.3 List of Queries

Here is a list of the remaining SQL queries developed for the project:

- Query 4

```sql
SELECT
    cm.year_number AS year,
    cm.calendar_month_number AS month_num,
    cm.calendar_month_name AS month_name,
    SUM(o.sales_order) AS total_sales
FROM Orders o
JOIN CalendarMonth cm ON o.order_calendar_id = cm.calendar_month_id
GROUP BY cm.year_number, cm.calendar_month_number, cm.calendar_month_name
ORDER BY cm.year_number, cm.calendar_month_number;
```

Listing 4: SQL Query for Monthly Sales

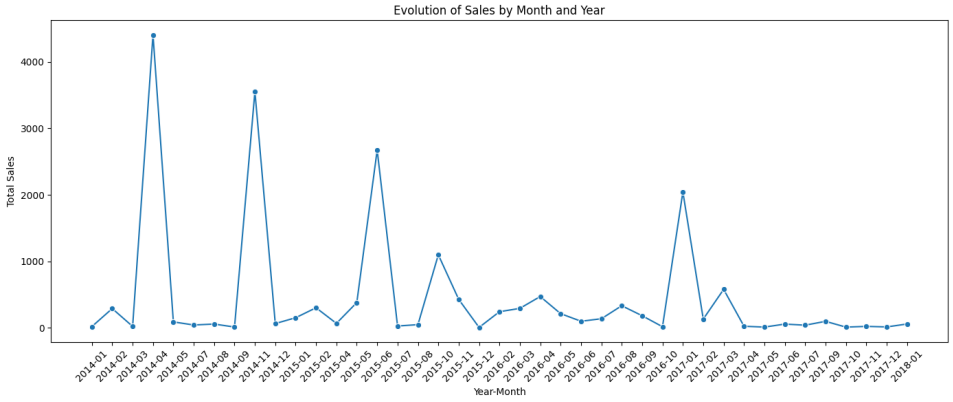

Figure 39: Query 4 Visualization

- Query 5:

```sql
SELECT
    r.region_name AS region,
    SUM(o.profit_order) AS total_profit
FROM Orders o
JOIN Location l ON o.location_id = l.location_id
JOIN State s ON l.state_id = s.state_id
JOIN Region r ON s.region_id = r.region_id
GROUP BY r.region_name
ORDER BY total_profit DESC;
```

Listing 5: SQL Query for Monthly Sales

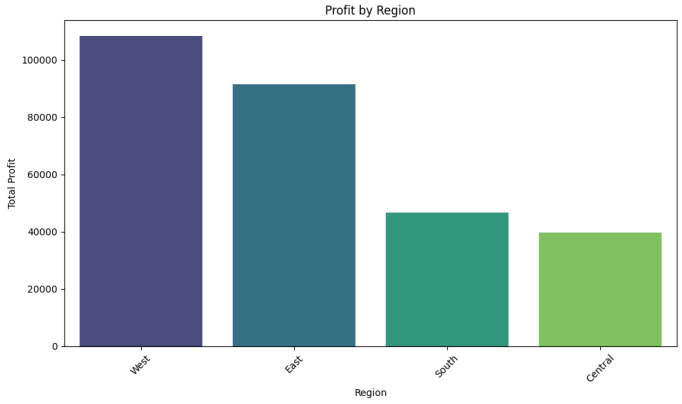

Figure 40: Query 5 Visualization

- Query 6:

```sql
SELECT
    IF(GROUPING(c.segment), 'TOTAL', c.segment) AS segment,
    SUM(o.sales_order) AS total_sales
FROM Orders o
JOIN Customer c ON o.customer_id = c.customer_id
GROUP BY c.segment WITH ROLLUP;
```

Listing 6: SQL Query for Monthly Sales

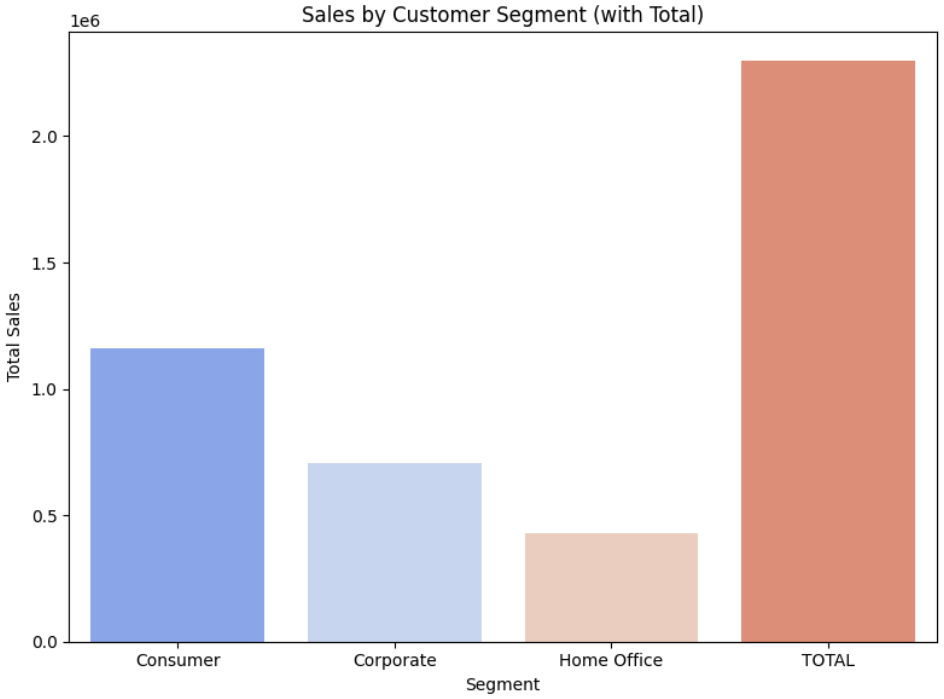

Figure 41: Query 6 Visualization

- Query 7:

```sql
SELECT
    product_name,
    total_profit,
    RANK() OVER (ORDER BY total_profit DESC) AS ranking
FROM (
    SELECT
        p.product_name,
        SUM(i.profit) AS total_profit
    FROM Item i
    JOIN Product p ON i.product_id = p.product_id
    GROUP BY p.product_name
) AS sub
ORDER BY ranking
LIMIT 10;
```

Listing 7: SQL Query for Monthly Sales

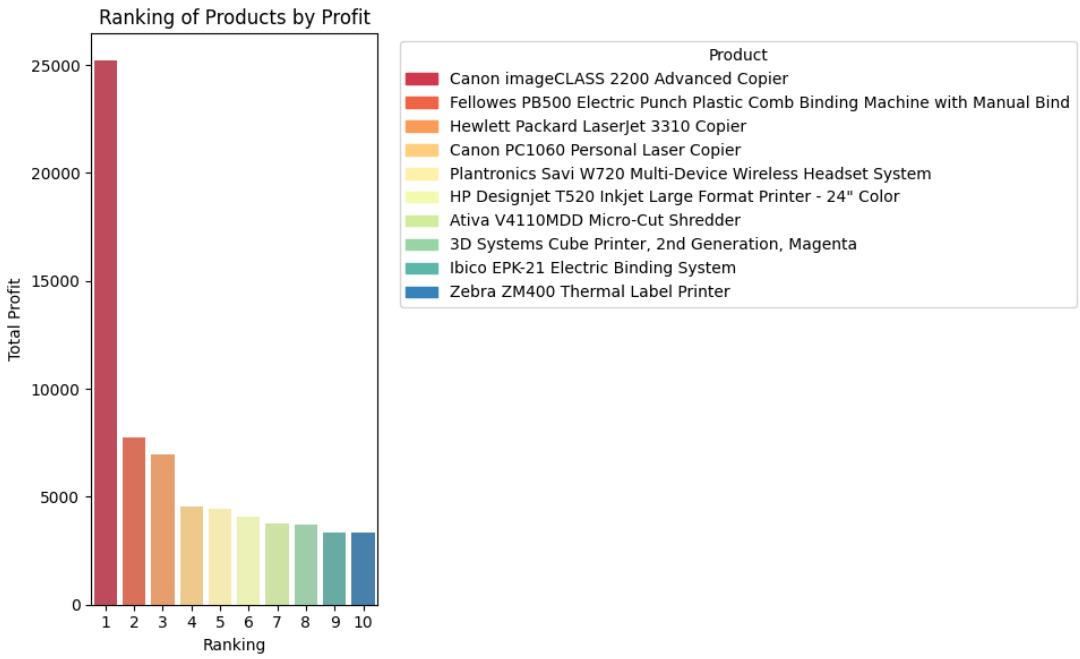

Figure 42: Query 7 Visualization

- Query 8:

```sql
SELECT
    sh.ship_mode,
    ROUND(SUM(o.lost_value_order), 2) AS lost_value
FROM Orders o
JOIN Shipping sh ON o.shipping_id = sh.shipping_id
GROUP BY sh.ship_mode
ORDER BY lost_value DESC;
```

Listing 8: SQL Query for Monthly Sales

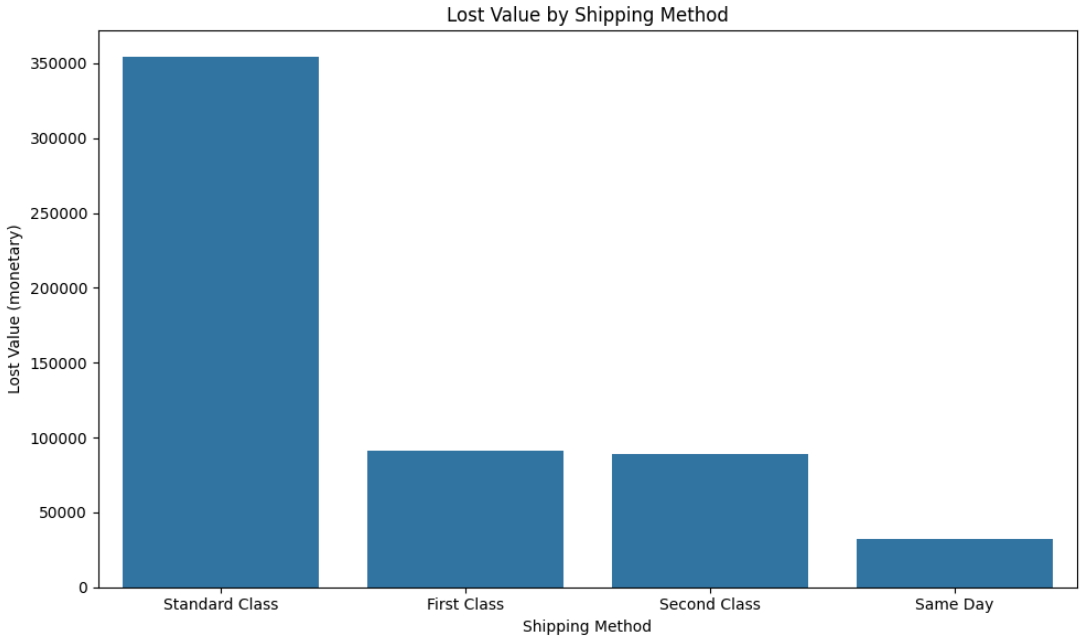

Figure 43: Query 8 Visualization

- Query 9:

```sql
SELECT
    cat.category_name,
    SUM(i.sales) AS total_sales,
    SUM(i.profit) AS total_profit,
    ROUND(AVG(i.discount), 2) AS average_discount
FROM Item i
JOIN Product p ON i.product_id = p.product_id
JOIN Category cat ON p.category_id = cat.category_id
GROUP BY cat.category_name
ORDER BY total_sales DESC;
```

Listing 9: SQL Query for Monthly Sales

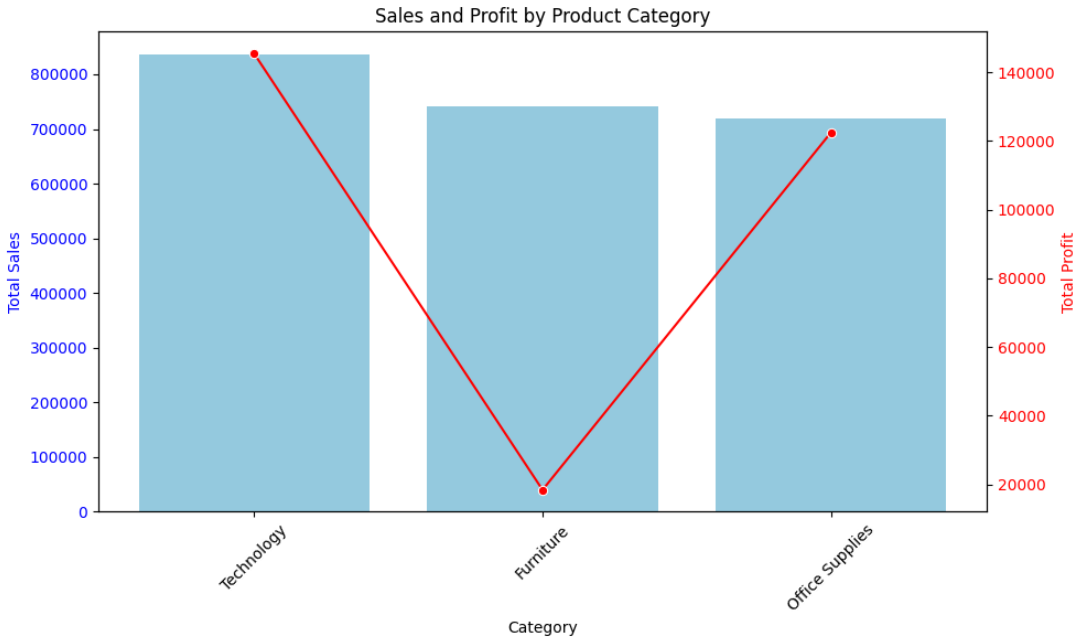

Figure 44: Query 9 Visualization

- Query 10:

```sql
SELECT
    s.state_name,
    SUM(o.sales_order) AS total_sales,
    SUM(SUM(o.sales_order)) OVER (ORDER BY s.state_name) AS accumulated_sales,
    ROUND(SUM(SUM(o.sales_order)) OVER (ORDER BY s.state_name) /
          (SELECT SUM(sales_order) FROM Orders) * 100, 2) AS cumulative_percentage
FROM Orders o
JOIN Location l ON o.location_id = l.location_id
JOIN State s ON l.state_id = s.state_id
GROUP BY s.state_name
ORDER BY cumulative_percentage;
```

Listing 10: SQL Query for Monthly Sales

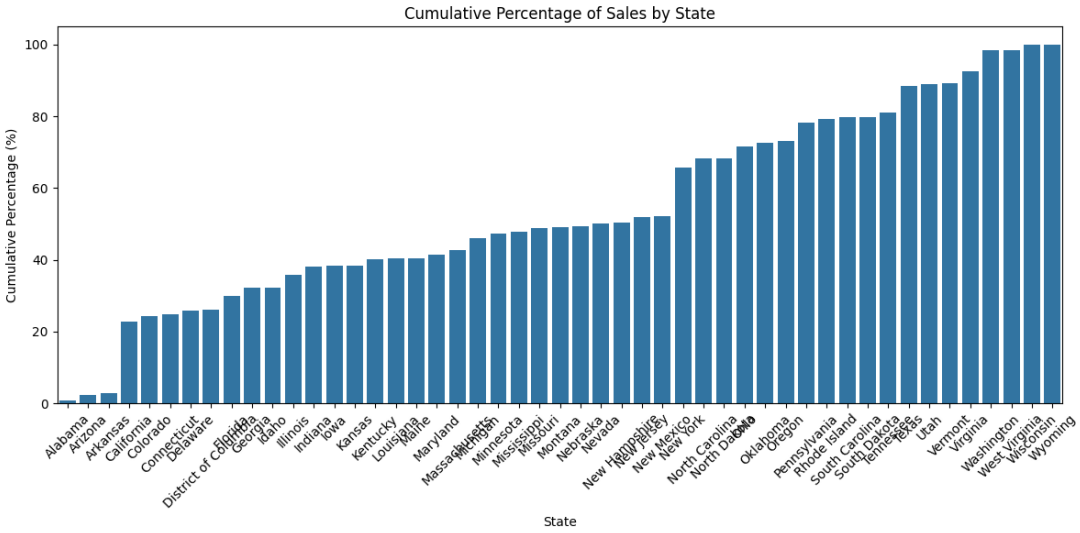

Figure 45: Query 10 Visualization

- Query 11:

```sql
SELECT
    s.state_name,
    cm.year_number,
    cm.calendar_month_number,
    cm.calendar_month_name,
    om.sales_month,
    om.profit_month,
    om.quantity_month,
    om.lost_value_month
FROM OrderM om
JOIN CalendarMonth cm ON om.calendar_month_id = cm.calendar_month_id
JOIN State s ON om.state_id = s.state_id
ORDER BY s.state_name, cm.year_number, cm.calendar_month_number;
```

Listing 11: SQL Query for Monthly Sales

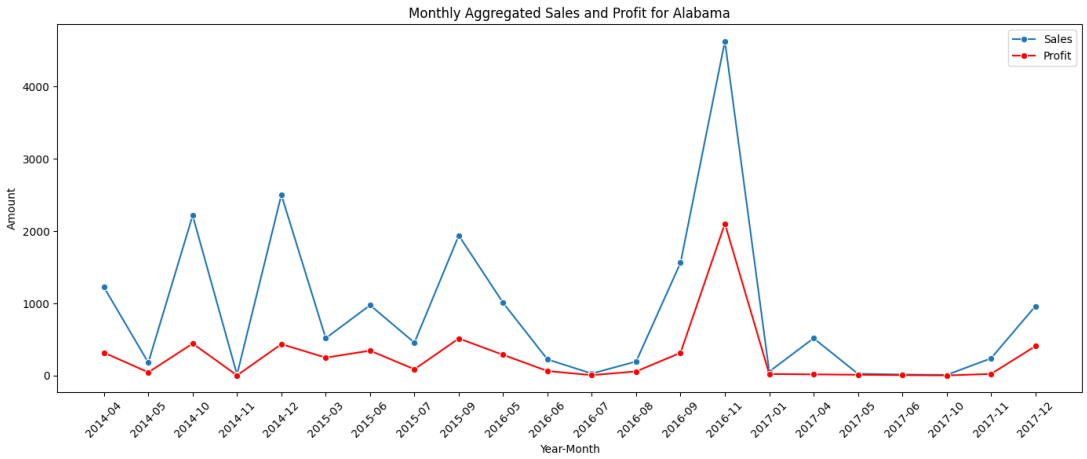

Figure 46: Query 11 Visualization

- Query 12:

```sql
WITH sales_cte AS (
  SELECT
    cat.category_name,
    p.product_name,
    i.sales,
    SUM(i.sales) OVER (PARTITION BY cat.category_name ORDER BY i.sales DESC ROWS
        UNBOUNDED PRECEDING) AS running_sales,
    SUM(i.sales) OVER (PARTITION BY cat.category_name) AS total_category_sales
  FROM Item i
  JOIN Product p ON i.product_id = p.product_id
  JOIN Category cat ON p.category_id = cat.category_id
)
SELECT
    category_name,
    product_name,
    sales,
    running_sales,
    total_category_sales
FROM sales_cte
WHERE running_sales - sales < 0.8 * total_category_sales
ORDER BY category_name, running_sales DESC
LIMIT 10;
```

Listing 12: SQL Query for Monthly Sales

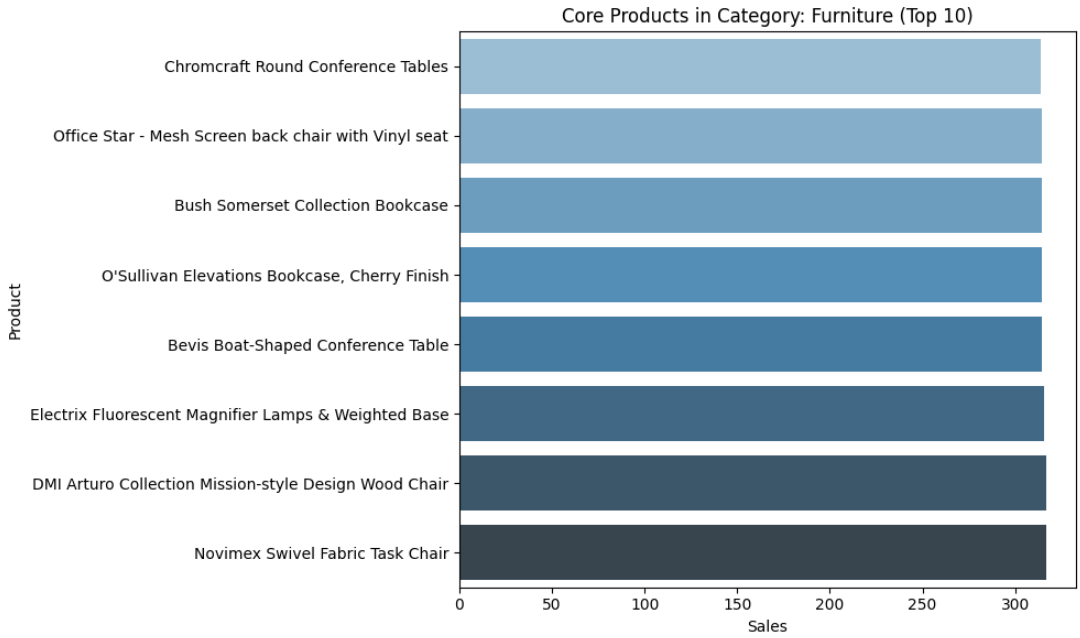

Figure 47: Query 12 Visualization

- Query 13:

```sql
SELECT
    cat.category_name,
    SUBSTRING_INDEX(
        GROUP_CONCAT(p.product_name ORDER BY i.sales DESC SEPARATOR ', '),
        ', ',
        10
    ) AS top_products,
    SUM(i.sales) AS total_sales
FROM Item i
JOIN Product p ON i.product_id = p.product_id
JOIN Category cat ON p.category_id = cat.category_id
GROUP BY cat.category_name;
```

Listing 13: SQL Query for Monthly Sales

	category_name	top_products	total_sales
0	Furniture	HON 5400 Series Task Chairs for Big and Tall, Riverside Palais Royal Lawyers Bookcase, Royale Cherry Finish, Chromcraft Bull-Nose Wood Oval Conference Tables & Bases, Riverside Palais Royal Lawyers Bookcase, Royale Cherry Finish, Sauder Forest Hills Library, Woodland Oak Finish, HON 5400 Series Task Chairs for Big and Tall, Bretford Rectangular Conference Table Tops	741999.98
1	Office Supplies	GBC Ibimaster 500 Manual ProClick Binding System, Ibico EPK-21 Electric Binding System, High Speed Automatic Electric Letter Opener, Fellowes PB500 Electric Punch Plastic Comb Binding Machine with Manual Bind, GBC DocuBind P400 Electric Binding System, Fellowes PB500 Electric Punch Plastic Comb Binding Machine with Manual Bind, High Speed Automatic Electric Letter Opener, Martin Yale Chadless Opener Electric Letter Opener, Ibico EPK-21 Electric Binding System, GBC DocuBind P400 Electric Binding System	719046.99
2	Technology	Cisco TelePresence System EX90 Videoconferencing Unit, Canon imageCLASS 2200 Advanced Copier, Canon imageCLASS 2200 Advanced Copier, Canon imageCLASS 2200 Advanced Copier, Canon imageCLASS 2200 Advanced Copier, 3D Systems Cube Printer, 2nd Generation, Magenta, HP Designjet T520 Inkjet Large Format Printer - 24" Color, Canon imageCLASS 2200 Advanced Copier	836154.10

Figure 48: Query 13 Visualization