Data Storage Solutions For Data Analytics

---Data Warehouse and Business Analysis

Zhilei Wu

1. Introduction

In an increasingly competitive global business environment, the construction of data warehouses is no longer an option for enterprises, but a strategic measure that must be taken. This report details how to use advanced data management technologies to build a data warehouse system for a Russian underwear trading company that can refine key business insights. By effectively integrating operational data and using business intelligence tools SSRS and Tableau for in-depth analysis, this report will demonstrate how data warehouses can help company leaders more comprehensively understand business processes, identify market opportunities, and on this basis, formulate more precise business strategies. [1], [2]

2. Business context and requirements

The business background of this report is based on the specific needs of a well-established Russian underwear trading company. As the company's operations expand and market competition intensifies, there is an urgent need to establish a centralized database system to centrally manage its dispersed operational data, thereby improving the availability of information and the effectiveness of decision-making. The establishment of a data warehouse not only helps the company store historical data but also supports complex data analysis to reveal potential business insights. In addition, efficient data visualization will become a powerful tool for providing insights and guiding decision-making for the company.[3], [4]

2.1 The main business requirements are as follows

1. Build a centralized data management system to achieve unified storage, management, and access to data.
2. Establish a data warehouse that not only meets the needs of storing historical data but also provides support for advanced data analysis.
3. Implement effective data visualization by providing management with clear business insights through vivid reports and charts.[2]

2.2 Stakeholders

1. **Procurement Team:** Needs insights into supply chain dynamics to optimize inventory and purchasing strategies.
2. **Marketing Department:** Focuses on market trends and customer behavior analysis to develop precise marketing plans.

3. **Senior Management:** Seeks an overall business view to support strategic planning and decision-making.

3. Data warehouse construction

3.1 Business Understanding and Database Architecture

3.1.1 Business understanding and database architecture design

In this project, we meticulously analyzed the operating model of the Russian underwear trading company and designed the database structure accordingly. The core business processes involve purchasing from suppliers, inventory management, order processing, sales tracking, and payment processing. For these business processes, we constructed a database model that encompasses 11 entities and their associations, ensuring that the data structure comprehensively reflects all aspects of the company's operations.

3.1.2 Database architecture design

We carefully designed the following main database entities based on actual business needs:

1. **Inventory Transactions:** Detailed records of inventory changes.
2. **Purchase Orders:** Tracking information on goods ordered from suppliers.
3. **Products:** Contains the product catalog and detailed specification information.
4. **Order Details and Orders**: Captures every detail of each sale.
5. **Payments and Payment Methods:** Tracks financial transactions and payment preferences.

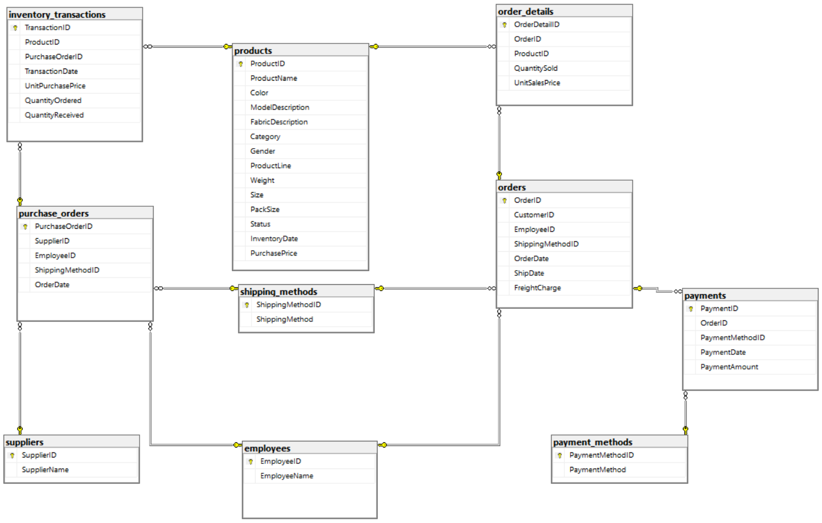


Figure 1 Database structure diagram

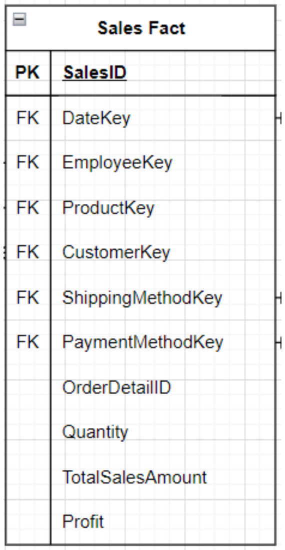
The relationships among these entities are linked through foreign keys, forming an efficient relational database model that facilitates data storage and retrieval.

3.2 Dimensional model design and implementation

3.2.1 Designing dimensions and fact sheets

Considering the company's business analysis needs, we chose a star schema design for the structure of the data warehouse, aiming to simplify the data analysis process. The data model includes two core fact tables and six dimension tables:

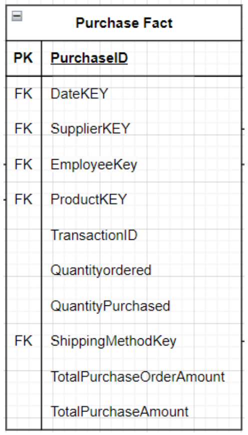
1. **Sales Fact Table:** Centralizes sales data, providing a basis for market trend analysis



Sales Fact	
PK	SalesID
FK	DateKey
FK	EmployeeKey
FK	ProductKey
FK	CustomerKey
FK	ShippingMethodKey
FK	PaymentMethodKey
	OrderDetailID
	Quantity
	TotalSalesAmount
	Profit

Figure 2 Sales Fact Table

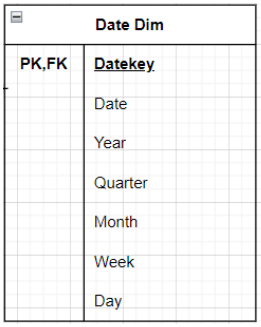
2. **Purchase Fact Table:** Detailed purchasing data supports supply chain analysis



Purchase Fact	
PK	PurchaseID
FK	DateKEY
FK	SupplierKEY
FK	EmployeeKey
FK	ProductKEY
	TransactionID
	Quantityordered
	QuantityPurchased
FK	ShippingMethodKey
	TotalPurchaseOrderAmount
	TotalPurchaseAmount

Figure 3 Purchase Fact Table

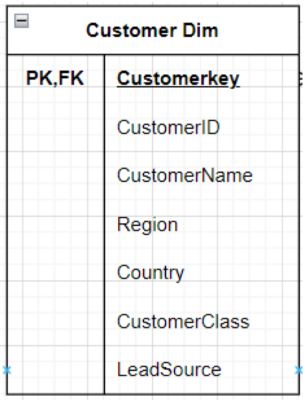
3. **Time Dimension (Date Dim):** Supports time series analysis



Date Dim	
PK,FK	**Datekey**
	Date
	Year
	Quarter
	Month
	Week
	Day

Figure 4 Time Dimension

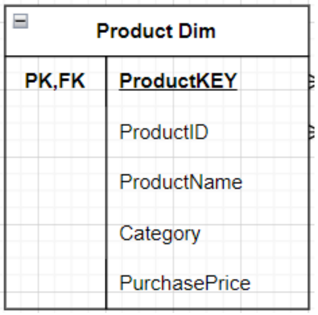
4. **Customer Dimension (Customer Dim**): Provides a basis for market segmentation



Customer Dim	
PK,FK	**Customerkey**
	CustomerID
	CustomerName
	Region
	Country
	CustomerClass
	LeadSource

Figure 5 Customer Dimension

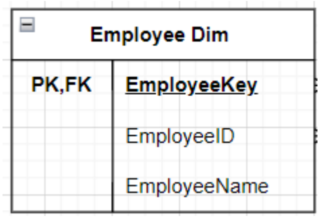
5. **Product Dimension:** Facilitates product performance analysis



Product Dim	
PK,FK	**ProductKEY**
	ProductID
	ProductName
	Category
	PurchasePrice

Figure 6 Product Dimension

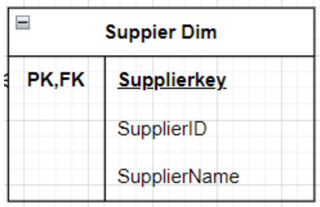
6. **Employee Dimension:** Reflects staff performance



Employee Dim	
PK,FK	**EmployeeKey**
	EmployeeID
	EmployeeName

Figure 7 Employee Dimension

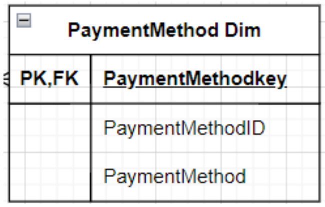
7. **Supplier Dimension:** Evaluates supplier performance



Suppier Dim	
PK,FK	**Supplierkey**
	SupplierID
	SupplierName

Figure 8 Supplier Dimension

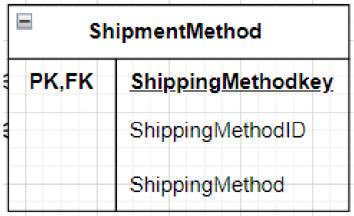
8. **Payment Method Dimension:** Analyzes financial flows



PaymentMethod Dim	
PK,FK	**PaymentMethodkey**
	PaymentMethodID
	PaymentMethod

Figure 9 Payment Method Dimension

9. **Transportation Method Dimension:** Optimizes logistics decisions



ShipmentMethod	
PK,FK	**ShippingMethodkey**
	ShippingMethodID
	ShippingMethod

Figure 10 Transportation Method Dimension

10. **Dimension and Fact Tables ER Diagram:** Ensures efficient data integration

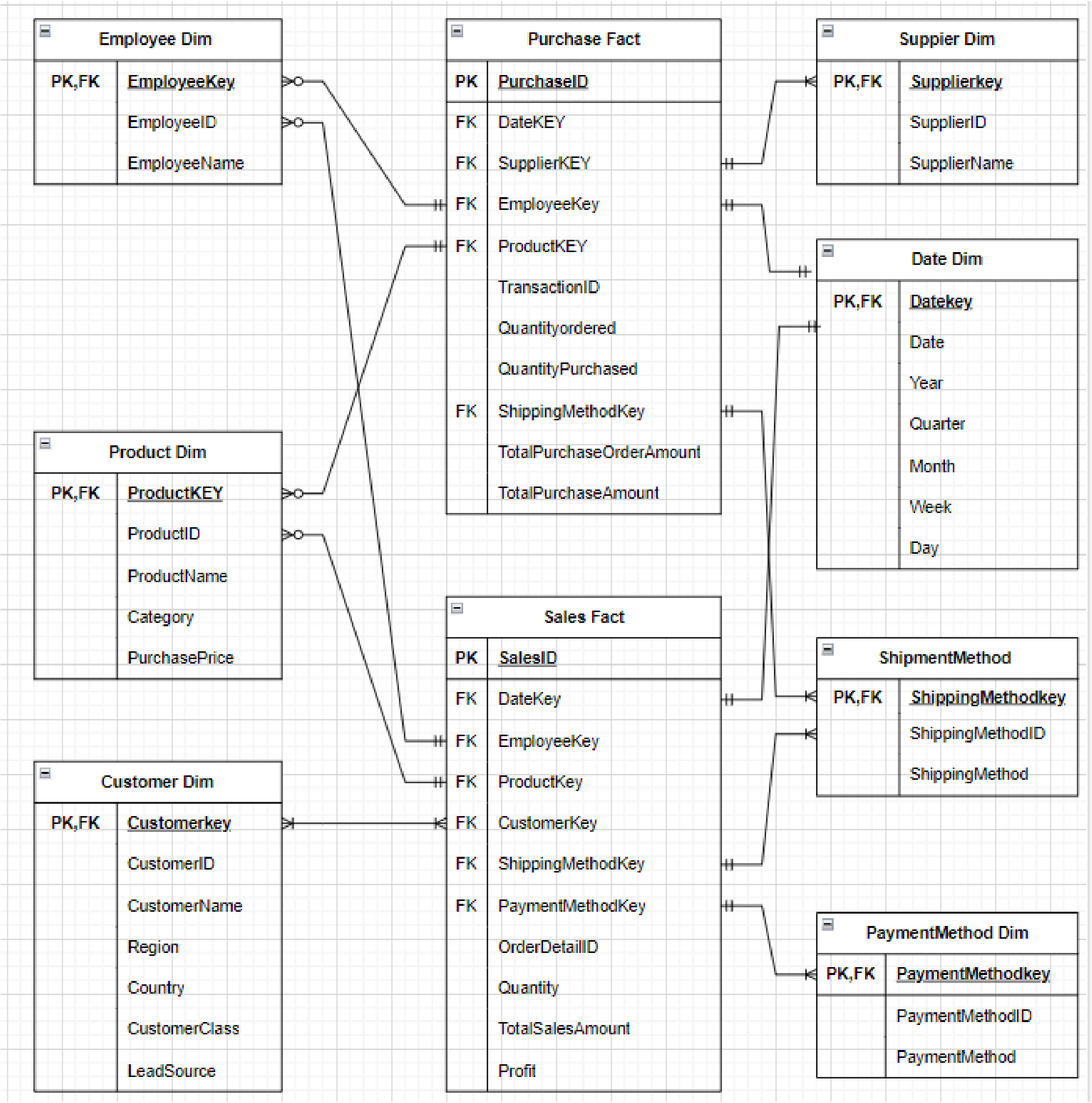


Figure 11 Dimension and Fact Tables ER Diagram

3.2.2 Dimension and fact table construction

The construction of fact tables and dimension tables was completed by writing and executing SQL scripts in SQL Server Management Studio (SSMS). Our detailed design and execution ensured the accurate implementation and efficient operation of the data model.

The details of running in SSMS are shown below:

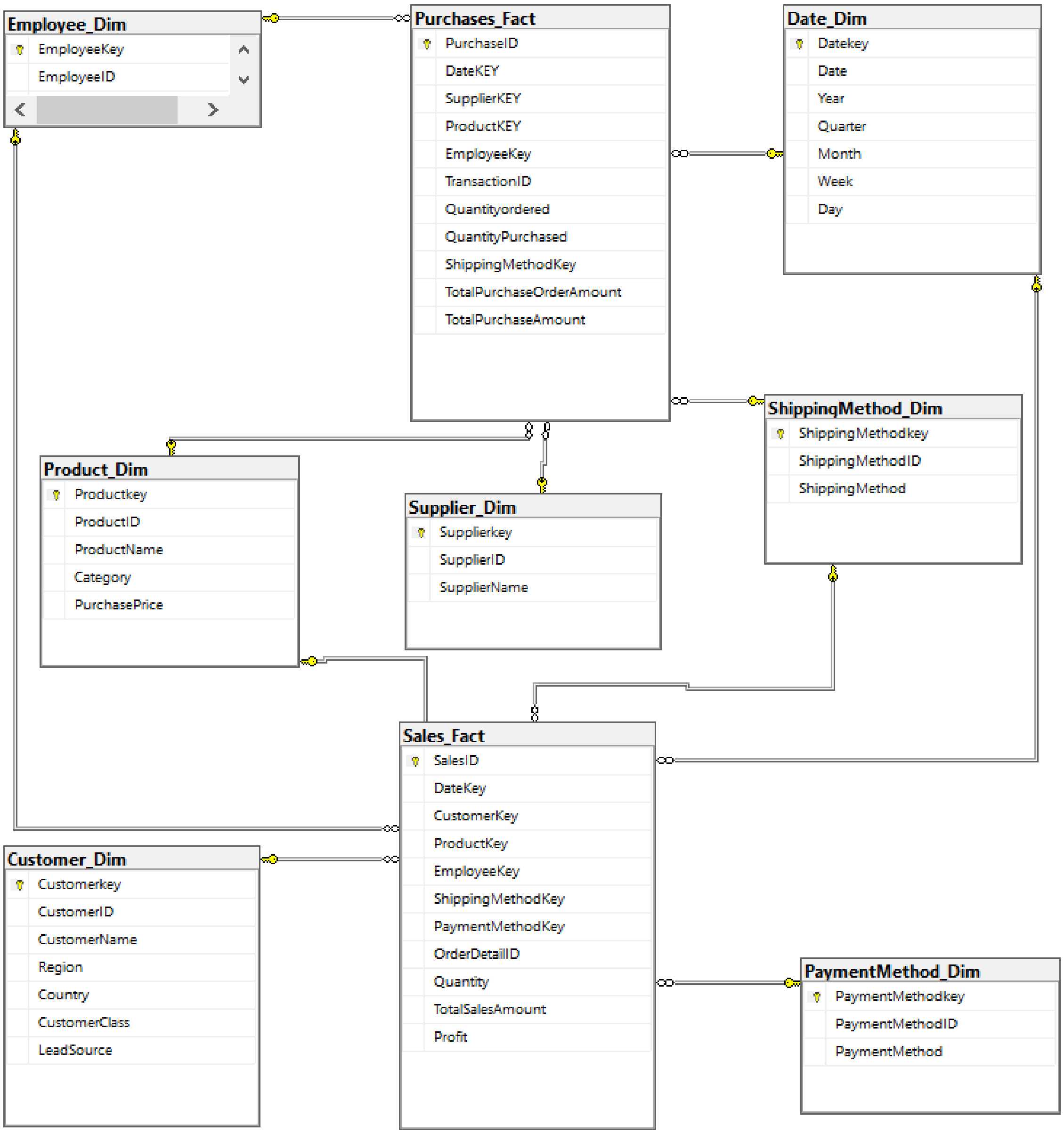


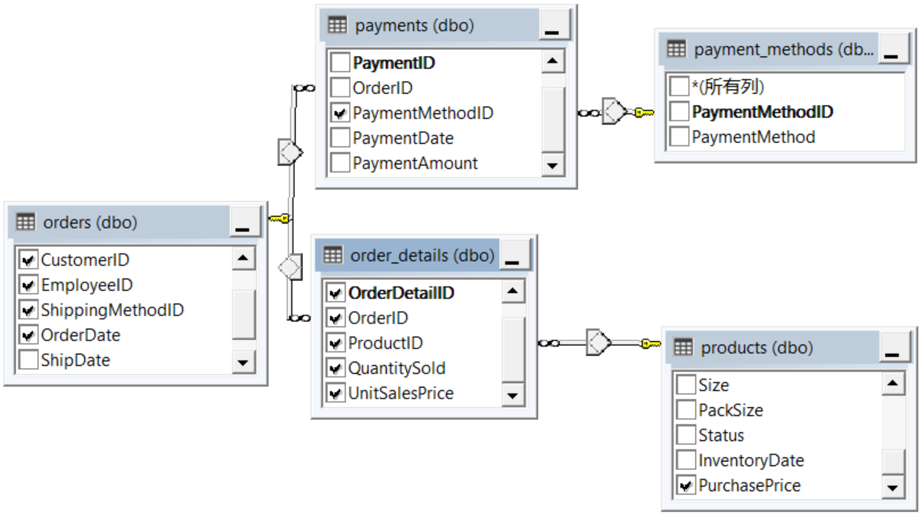
Figure 12 Dimension and fact table structure

3.3 ETL process implementation

In the ETL (Extract, Transform, Load) process, we used Microsoft SQL Server Integration Services (SSIS) as the primary tool. Through each step of ETL, we ensured the efficient and accurate transfer of data from data sources to the data warehouse. [5]

3.3.1 Data Extraction

The data extraction process involved pulling data from data sources, establishing OLEDB connection managers. This data includes sales records, payment information, customer details, etc., and the extraction process was automated to ensure data timeliness and accuracy, as shown in the following diagram:



payments (dbo)
- □ PaymentID
- □ OrderID
- ☑ PaymentMethodID
- □ PaymentDate
- □ PaymentAmount

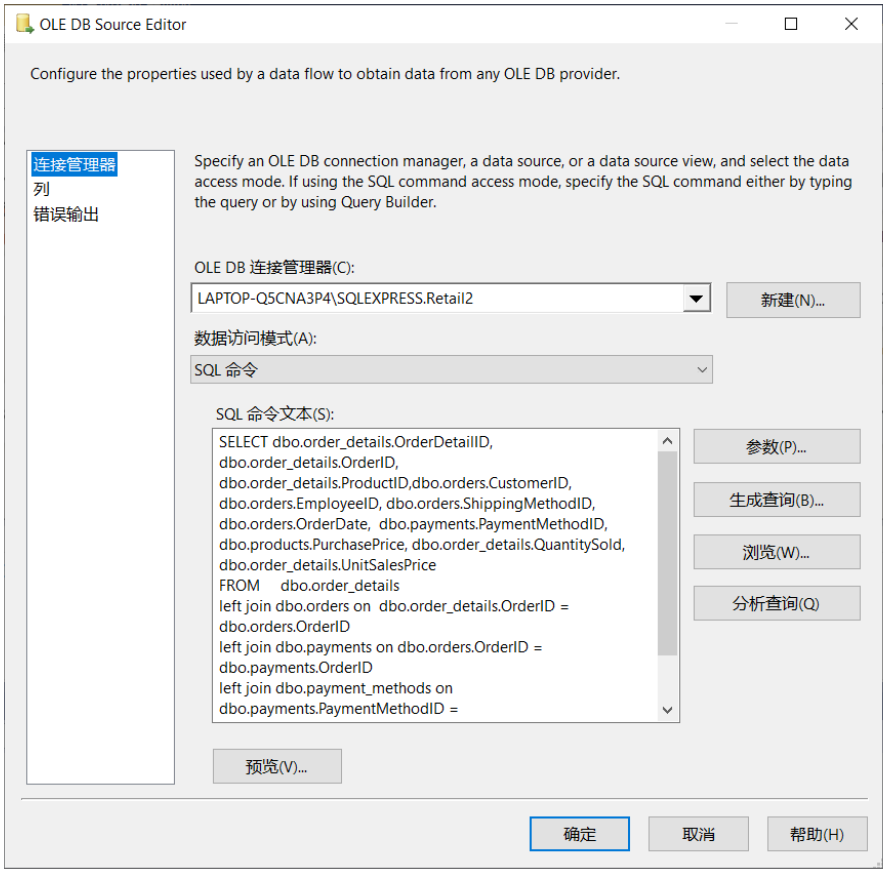
payment_methods (db...
- □ *(所有列)
- □ PaymentMethodID
- □ PaymentMethod

orders (dbo)
- ☑ CustomerID
- ☑ EmployeeID
- ☑ ShippingMethodID
- ☑ OrderDate
- □ ShipDate

order_details (dbo)
- ☑ OrderDetailID
- ☑ OrderID
- ☑ ProductID
- ☑ QuantitySold
- ☑ UnitSalesPrice

products (dbo)
- □ Size
- □ PackSize
- □ Status
- □ InventoryDate
- ☑ PurchasePrice

Figure 13 Data extraction from data sources



OLE DB Source Editor

Configure the properties used by a data flow to obtain data from any OLE DB provider.

Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.

OLE DB 连接管理器(C): LAPTOP-Q5CNA3P4\SQLEXPRESS.Retail2

数据访问模式(A): SQL 命令

SQL 命令文本(S):

```
SELECT dbo.order_details.OrderDetailID,
dbo.order_details.OrderID,
dbo.order_details.ProductID,dbo.orders.CustomerID,
dbo.orders.EmployeeID, dbo.orders.ShippingMethodID,
dbo.orders.OrderDate,  dbo.payments.PaymentMethodID,
dbo.products.PurchasePrice, dbo.order_details.QuantitySold,
dbo.order_details.UnitSalesPrice
FROM     dbo.order_details
left join dbo.orders on  dbo.order_details.OrderID =
dbo.orders.OrderID
left join dbo.payments on dbo.orders.OrderID =
dbo.payments.OrderID
left join dbo.payment_methods on
dbo.payments.PaymentMethodID =
```

Figure 14 Data extraction by automated process

3.3.2 Data Transformation

During the transformation phase, we cleaned the data by removing missing values, inconsistent formats, and incorrect data entries. We also calculated new fields and identified primary keys for linking to dimension tables, ensuring data consistency and accuracy.

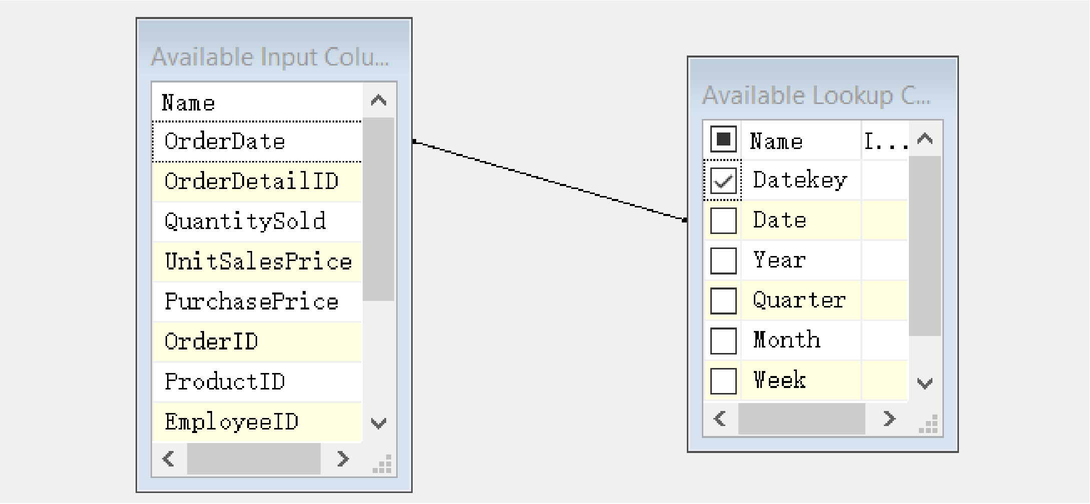
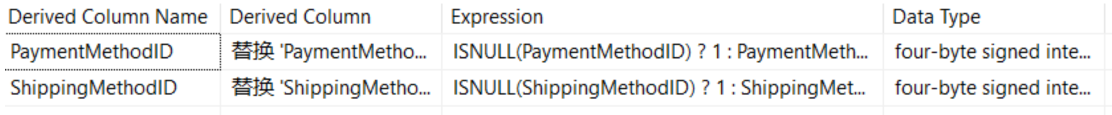
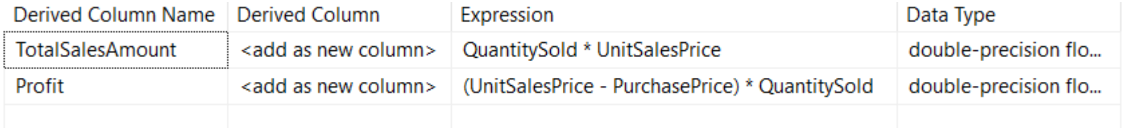


Figure 15 Joining the primary key of a dimension table



Derived Column Name	Derived Column	Expression	Data Type
PaymentMethodID	替换 'PaymentMetho...	ISNULL(PaymentMethodID) ? 1 : PaymentMeth...	four-byte signed inte...
ShippingMethodID	替换 'ShippingMetho...	ISNULL(ShippingMethodID) ? 1 : ShippingMet...	four-byte signed inte...

Figure 16 Missing Value Processing Chart

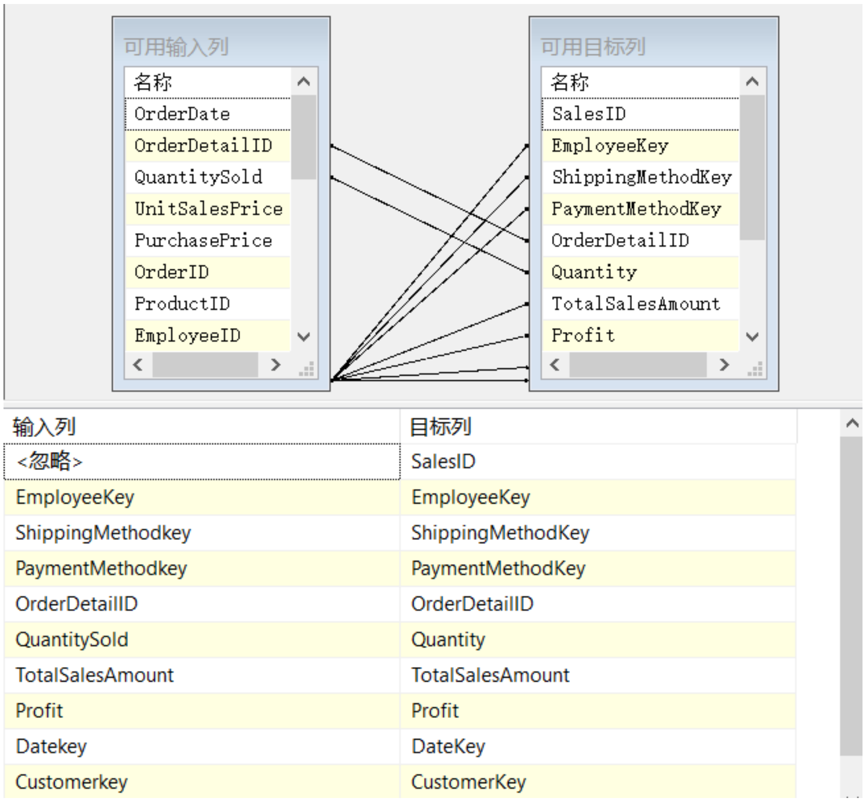


Derived Column Name	Derived Column	Expression	Data Type
TotalSalesAmount	<add as new column>	QuantitySold * UnitSalesPrice	double-precision flo...
Profit	<add as new column>	(UnitSalesPrice - PurchasePrice) * QuantitySold	double-precision flo...

Figure 17 New Field Calculation Chart

3.3.3 Data Loading

The loading step involved populating the transformed data into the predefined star schema. This involved inserting data into the fact tables and corresponding dimension tables, and linking the appropriate primary and foreign keys.



可用输入列

名称
OrderDate
OrderDetailID
QuantitySold
UnitSalesPrice
PurchasePrice
OrderID
ProductID
EmployeeID

可用目标列

名称
SalesID
EmployeeKey
ShippingMethodKey
PaymentMethodKey
OrderDetailID
Quantity
TotalSalesAmount
Profit

输入列	目标列
<忽略>	SalesID
EmployeeKey	EmployeeKey
ShippingMethodkey	ShippingMethodKey
PaymentMethodkey	PaymentMethodKey
OrderDetailID	OrderDetailID
QuantitySold	Quantity
TotalSalesAmount	TotalSalesAmount
Profit	Profit
Datekey	DateKey
Customerkey	CustomerKey

Figure 18 Data Loading processing

3.4 Data Warehouse Validation

By comparing the number of rows imported in the SSIS project with the source data query results, we verified the success of the data loading. Once all operations were completed, the data warehouse's structure and design were consistent with the expected outcomes, marking the successful establishment of the data warehouse.

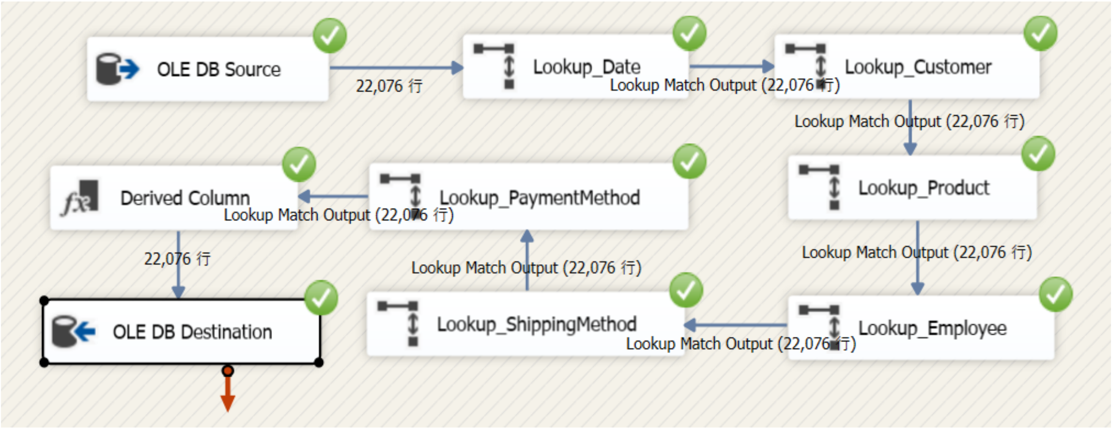

OLE DB Source
22,076 行
Lookup_Date
Lookup Match Output (22,076 行)
Lookup_Customer
Lookup Match Output (22,076 行)
Lookup_Product
Lookup Match Output (22,076 行)
Lookup_Employee
Lookup Match Output (22,076 行)
Lookup_ShippingMethod
Lookup Match Output (22,076 行)
Lookup_PaymentMethod
Lookup Match Output (22,076 行)
Derived Column
22,076 行
OLE DB Destination

Figure 19 Sales Fact Table ETL Validation

The dimension table is built with reference to the same steps above and the results are as follows:

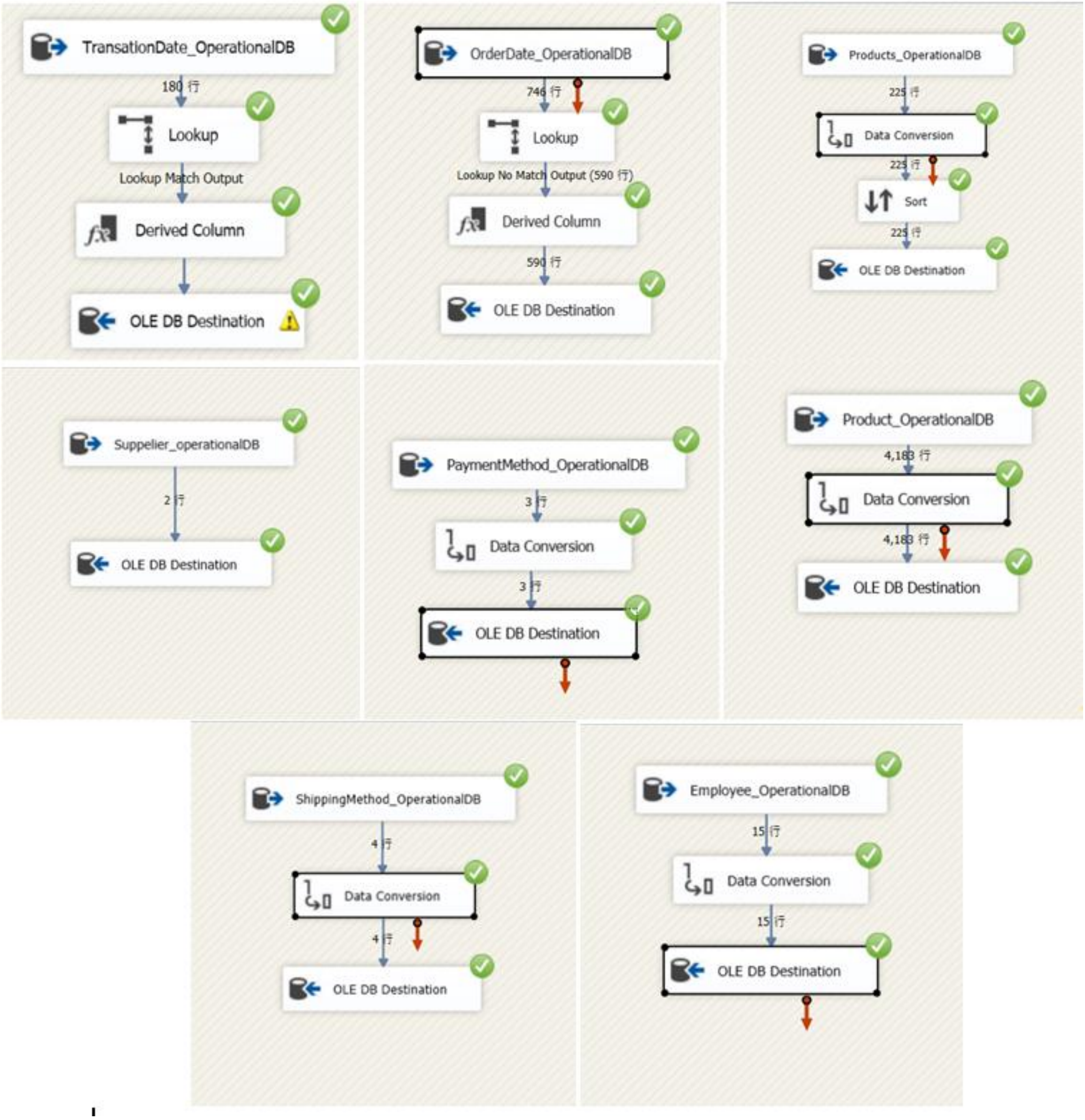


Figure 20 Other Dimension Table ETL Validation

3.5 Data warehouse deployment validation

With the deployment of the data warehouse, all stakeholders are now able to utilize it for their respective purposes.

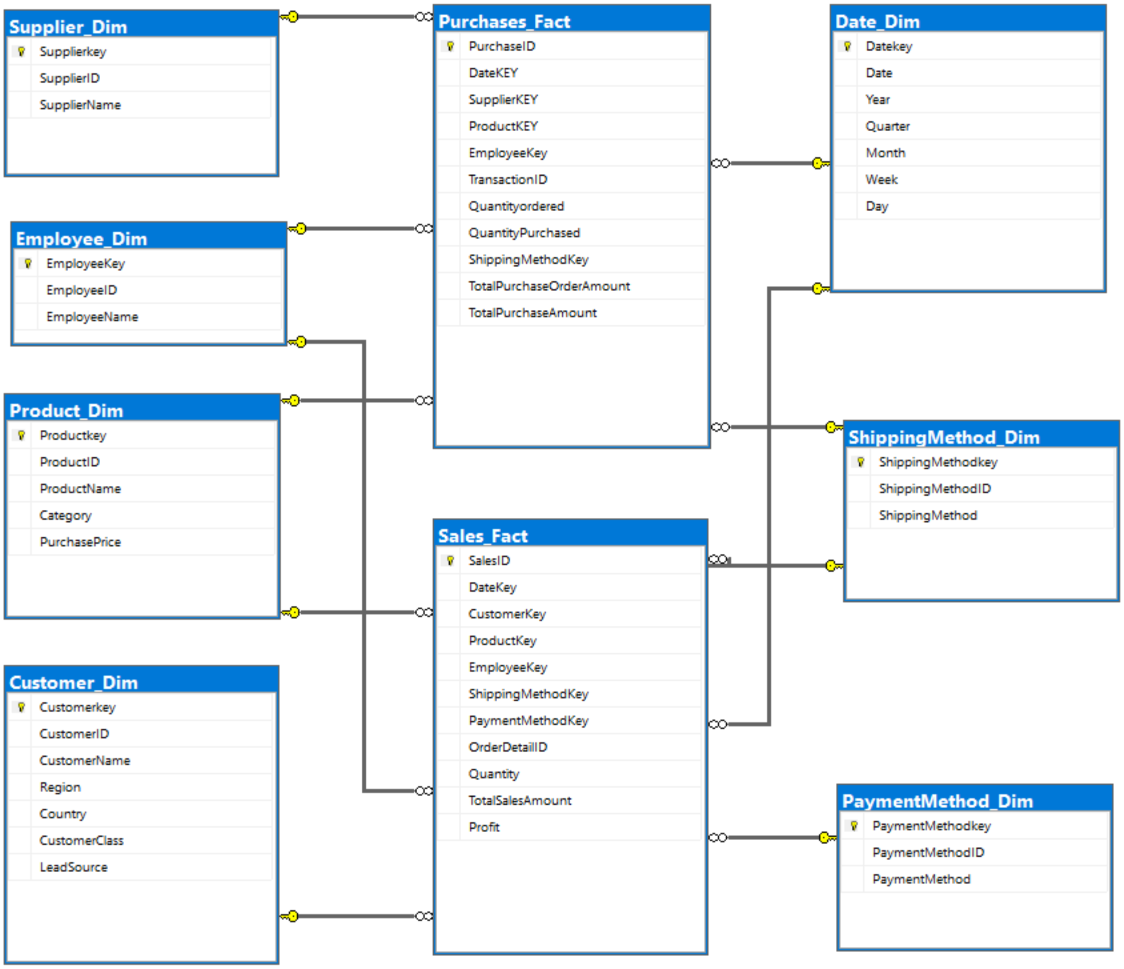


Figure 21 Data Warehouse Diagram

4. Business Analysis

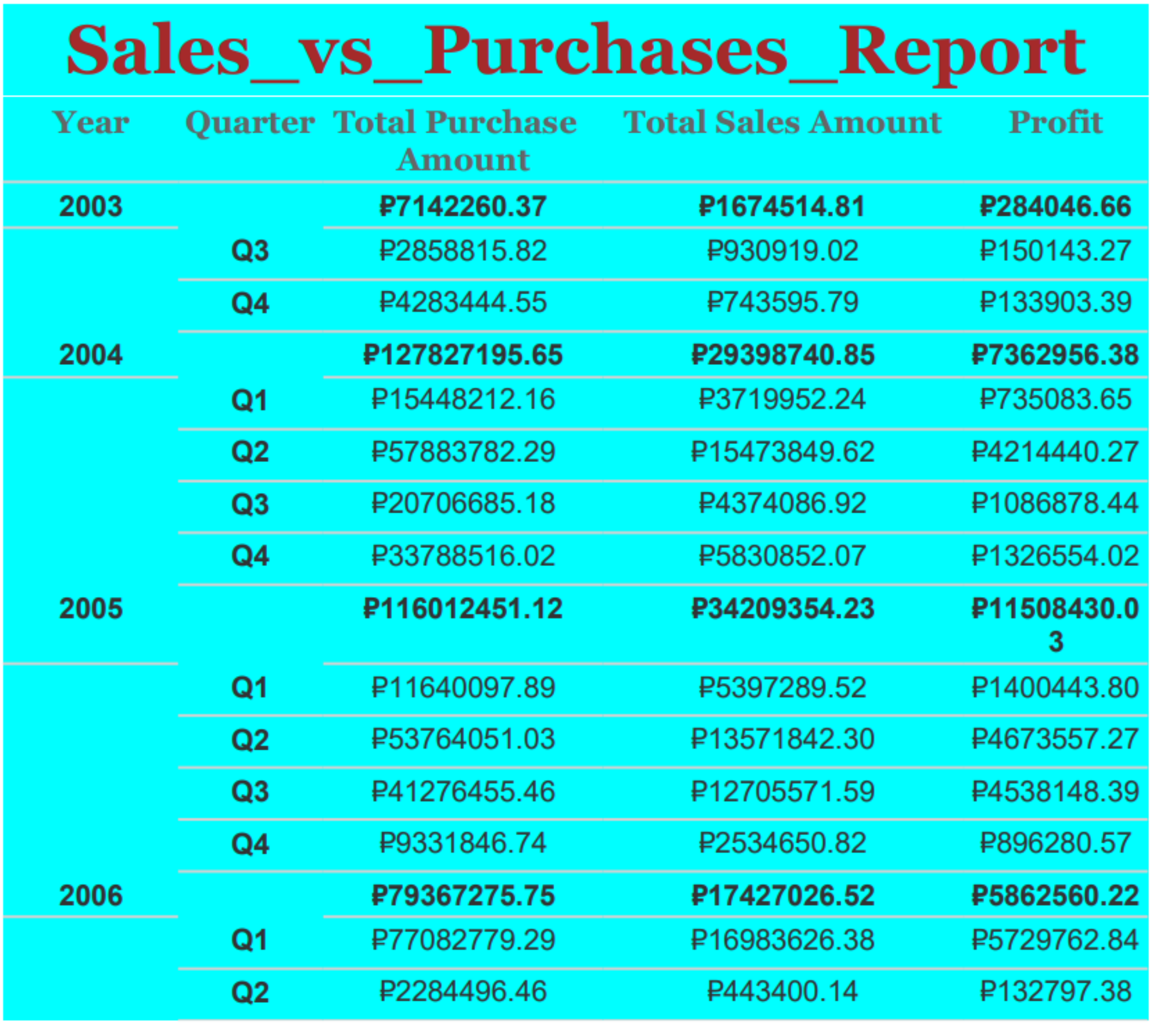
4.1 SSRS report analysis

After the data warehouse was established, we used SQL Server Reporting Services (SSRS) to generate a series of reports to support management decisions and optimize business processes. The following is an overview of key reports:

Report 1: Sales and Purchase Analysis Report

- **Stakeholders and Purpose of Analysis:**

This report aims to provide the procurement and marketing teams with an in-depth analysis of sales and purchasing data, revealing trends in quarterly sales and purchasing activities. This helps the teams to predict market movements, optimize inventory management, and provide data support for strategic decision-making by the upper management.



Sales_vs_Purchases_Report

Year	Quarter	Total Purchase Amount	Total Sales Amount	Profit
2003		**₱7142260.37**	**₱1674514.81**	**₱284046.66**
	Q3	₱2858815.82	₱930919.02	₱150143.27
	Q4	₱4283444.55	₱743595.79	₱133903.39
2004		**₱127827195.65**	**₱29398740.85**	**₱7362956.38**
	Q1	₱15448212.16	₱3719952.24	₱735083.65
	Q2	₱57883782.29	₱15473849.62	₱4214440.27
	Q3	₱20706685.18	₱4374086.92	₱1086878.44
	Q4	₱33788516.02	₱5830852.07	₱1326554.02
2005		**₱116012451.12**	**₱34209354.23**	**₱11508430.03**
	Q1	₱11640097.89	₱5397289.52	₱1400443.80
	Q2	₱53764051.03	₱13571842.30	₱4673557.27
	Q3	₱41276455.46	₱12705571.59	₱4538148.39
	Q4	₱9331846.74	₱2534650.82	₱896280.57
2006		**₱79367275.75**	**₱17427026.52**	**₱5862560.22**
	Q1	₱77082779.29	₱16983626.38	₱5729762.84
	Q2	₱2284496.46	₱443400.14	₱132797.38

Figure 22 Data Sales and Purchase Analysis Report

- **Analysis Method:**

By synthesizing quarterly data between 2003 and 2006, focusing on total purchases, total sales and profit for each quarter, these metrics are critical to understanding business performance.

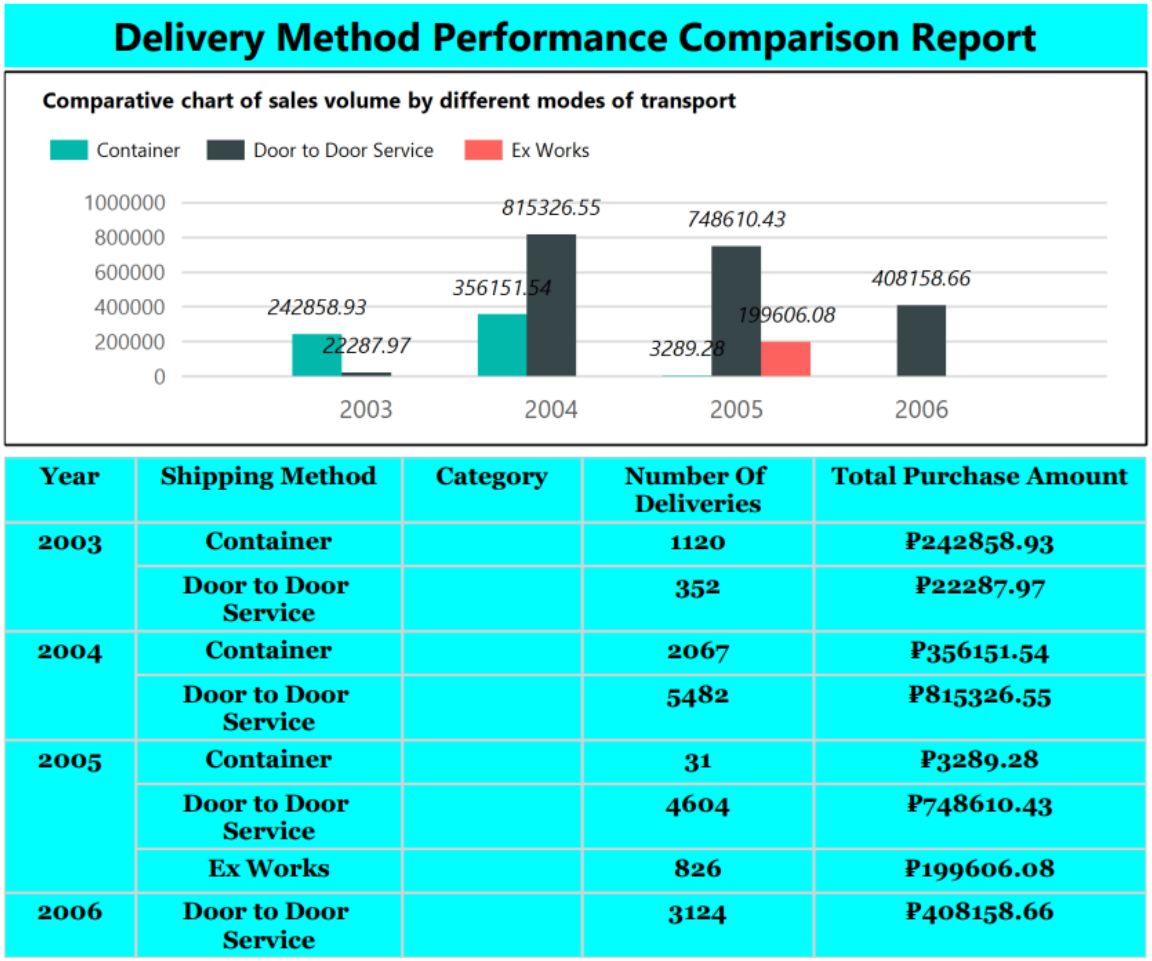
- **Business Insight:**

The numbers show increasing sales and profits over the years, particularly strong in Q4, which may be due to holiday sales. The sharp profit rise in 2006's Q2 suggests effective strategies. Overall, the company is growing steadily, highlighting the need for smart planning around peak seasons to boost profits.

Report 2: Product Transportation Method Analysis Report

- **Stakeholders and Purpose of Analysis:**

This report is prepared for the logistics management team and senior management to analyze the impact of different transportation methods on sales and purchasing, thereby optimizing logistics costs.



Delivery Method Performance Comparison Report

Year	Shipping Method	Category	Number Of Deliveries	Total Purchase Amount
2003	Container		1120	₱242858.93
	Door to Door Service		352	₱22287.97
2004	Container		2067	₱356151.54
	Door to Door Service		5482	₱815326.55
2005	Container		31	₱3289.28
	Door to Door Service		4604	₱748610.43
	Ex Works		826	₱199606.08
2006	Door to Door Service		3124	₱408158.66

Figure 23 Product Transportation Method Analysis Report

- **Methods of analysis:**

By collating and analysing data on the number of deliveries and total purchase value of different modes of transport between 2003 and 2006, the performance of different logistics services in terms of cost and efficiency is highlighted.

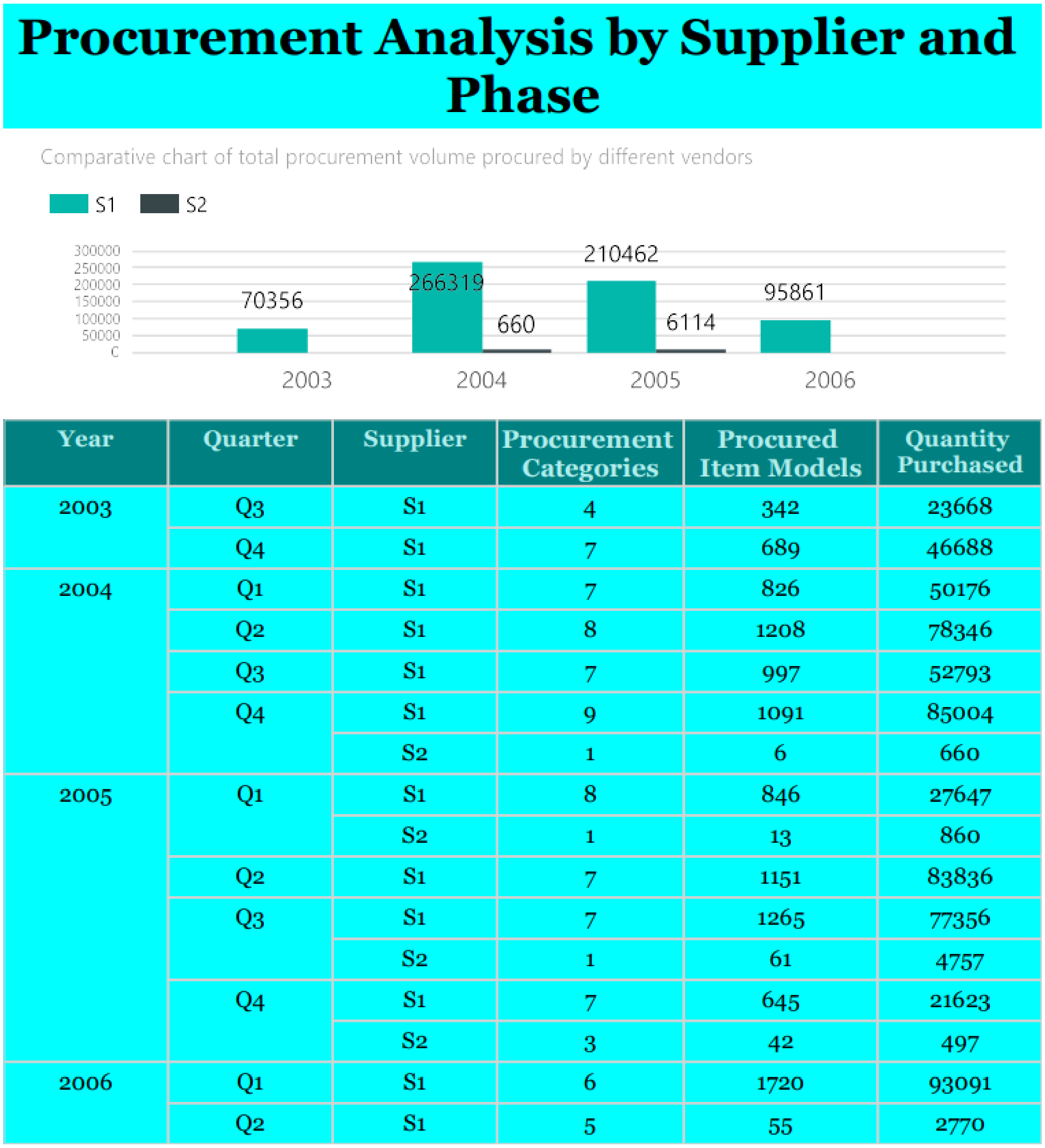
- **Business Insights:**

The chart indicates that Door to Door service is increasing in popularity and effectiveness, as seen in the rising number of deliveries and purchase amounts from 2003 to 2006. This suggests customers value the convenience it offers. The use of containers shows a significant purchase amount in 2004, but with fewer deliveries compared to Door to Door, hinting at its use for larger, less frequent shipments. Ex Works, introduced in 2005, has yet to match the performance of the other methods. The company may consider focusing on the profitable Door to Door service and exploring how to optimize container use for efficiency.

Report 3: Purchasing by Supplier Analysis Report

- **Stakeholders and Purpose of Analysis:**

The supply chain management team and the purchasing department are the main audience for this report. It aims to analyze different suppliers' categories of purchases, models, and purchase volumes, to assess supplier performance and guide future purchasing strategies.



Procurement Analysis by Supplier and Phase

Comparative chart of total procurement volume procured by different vendors

Year	Quarter	Supplier	Procurement Categories	Procured Item Models	Quantity Purchased
2003	Q3	S1	4	342	23668
	Q4	S1	7	689	46688
2004	Q1	S1	7	826	50176
	Q2	S1	8	1208	78346
	Q3	S1	7	997	52793
	Q4	S1	9	1091	85004
		S2	1	6	660
2005	Q1	S1	8	846	27647
		S2	1	13	860
	Q2	S1	7	1151	83836
	Q3	S1	7	1265	77356
		S2	1	61	4757
	Q4	S1	7	645	21623
		S2	3	42	497
2006	Q1	S1	6	1720	93091
	Q2	S1	5	55	2770

Figure 24 Purchasing by Supplier Analysis Report

- **Methods of analysis:**

The SSRS tool was used to show in detail the number of categories purchased, the types of products purchased, and the quantities purchased by suppliers S1 and S2, based on data from 2003 to 2006, broken down by quarter.

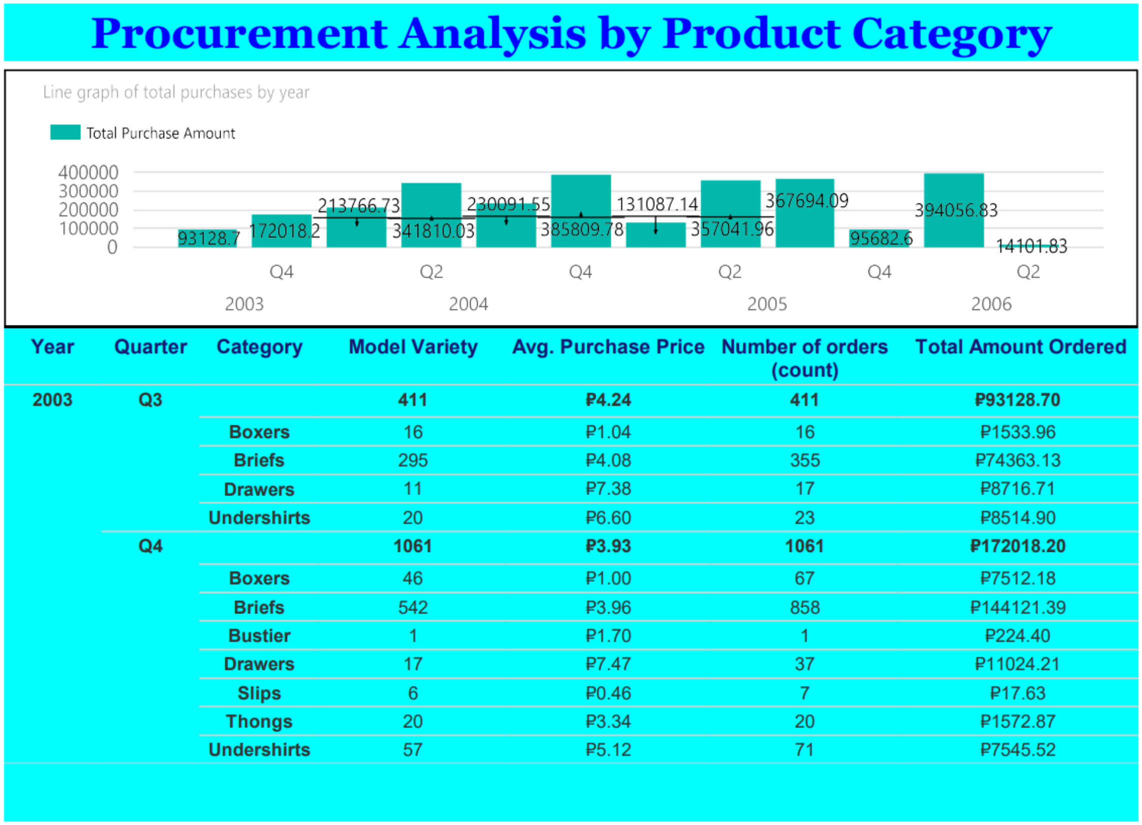
- **Business Insights:**

The analysis suggests Supplier S1 is the primary source, with a consistent increase in procurement volume over the years, peaking in Q4 of 2004. Despite a varied number of procurement categories and item models, S1's quantity purchased consistently grows, indicating a strong and possibly exclusive partnership. Supplier S2, while introduced in 2004, contributes a smaller portion and sees a modest increase in volume by 2006. The company might benefit from evaluating S2’s growth potential or using it for strategic sourcing to mitigate risks associated with dependence on a single supplier.

Report 4: Product Category Purchasing Analysis Report

- **Stakeholders and Purpose of Analysis:**

This report is prepared for the purchasing department and product management team, aimed at analysing the company's product purchasing composition, including the variety of product models, average purchasing prices, and order quantities within each product category.



Procurement Analysis by Product Category

Year	Quarter	Category	Model Variety	Avg. Purchase Price	Number of orders (count)	Total Amount Ordered
2003	**Q3**		**411**	**₱4.24**	**411**	**₱93128.70**
		Boxers	16	₱1.04	16	₱1533.96
		Briefs	295	₱4.08	355	₱74363.13
		Drawers	11	₱7.38	17	₱8716.71
		Undershirts	20	₱6.60	23	₱8514.90
	Q4		**1061**	**₱3.93**	**1061**	**₱172018.20**
		Boxers	46	₱1.00	67	₱7512.18
		Briefs	542	₱3.96	858	₱144121.39
		Bustier	1	₱1.70	1	₱224.40
		Drawers	17	₱7.47	37	₱11024.21
		Slips	6	₱0.46	7	₱17.63
		Thongs	20	₱3.34	20	₱1572.87
		Undershirts	57	₱5.12	71	₱7545.52

Figure 25 Product Category Purchasing Analysis Report

- **Methods of analysis：**

Purchases in different product categories between 2003 and 2006 were analyzed, including the diversity of models, average purchase price, number of orders and total purchase amount for each category.

- **Business Insights:**

A detailed review of the data reveals differing purchasing strategies across various product categories. The significant increase in variety and orders for Q4 compared to Q3 indicates a possible seasonal demand. This procurement pattern highlights the importance of optimizing inventory to align with consumer demand and suggests that detailed analysis of product performance could further refine procurement strategies.

4.2 Tableau Visual Analysis

4.2.1 Business Visualization

- Customer Distribution Map:

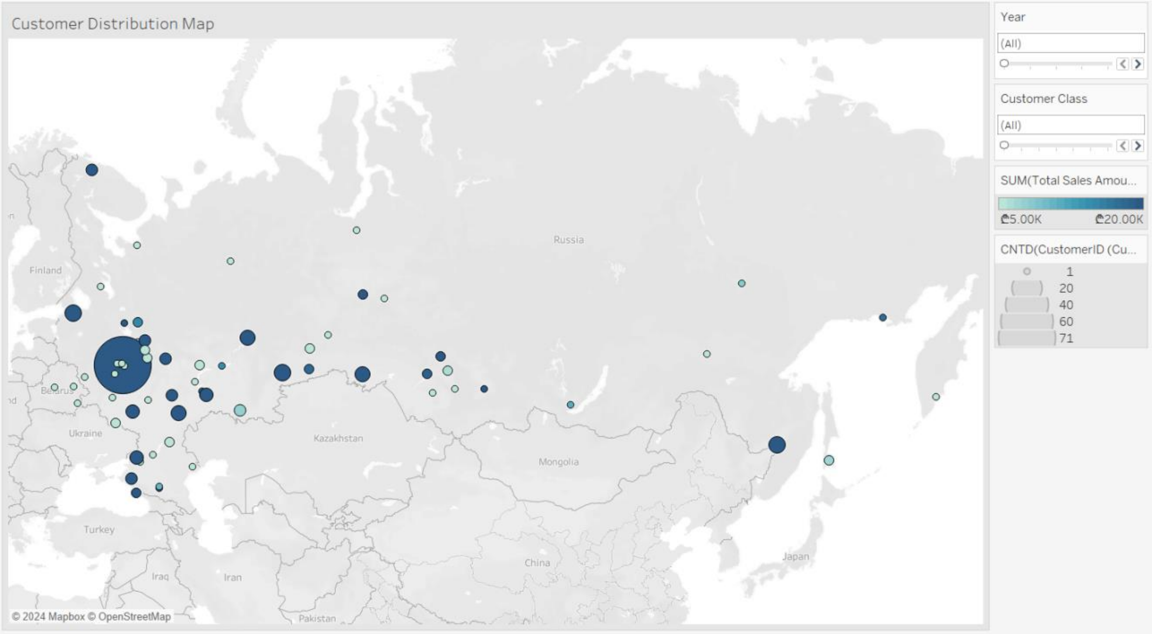


Figure 26 Customer Distribution Map

This map, visualized through Tableau, displays the distribution of the company's customers across different geographic locations. Each point on the map represents a city, with the size of the point corresponding to the number of customers in that city, and the shade of color indicating the total sales amount. It can be observed that Moscow has the highest concentration of customers and also the highest sales volume, accounting for nearly half of the company's total sales. This offers a clear perspective on the regions where the company's market is mainly concentrated.[6]

- Quarterly Sales And Profit Comparison:

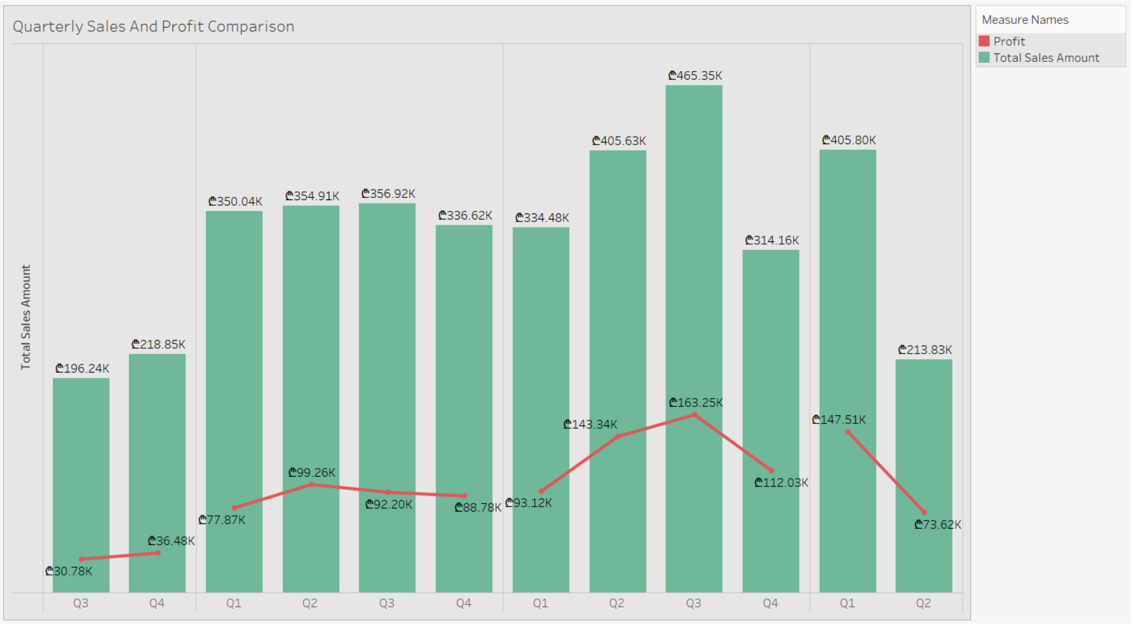


Figure 27 Quarterly Sales And Profit Comparison

This chart shows a comparison between total sales and profits by quarter. The bar graph represents the total sales for each quarter, while the line chart shows the corresponding profits. From this visualization, it is easy to identify which quarters performed better and the relationship between sales and profits.

- Top 10 Profitable Regions:

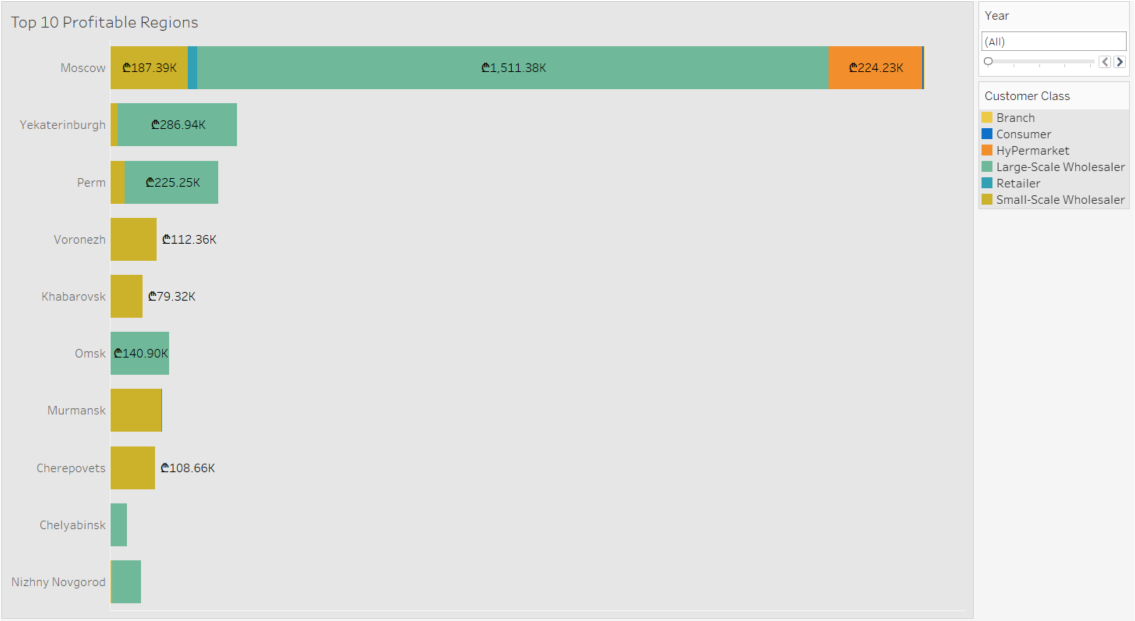


Figure 28 Top 10 Profitable Regions

This bar chart ranks the company's top ten most profitable regions. Each bar represents the profit of a region, with different colors distinguishing various customer categories. For example, the Moscow region, as a large wholesaler, has the highest profit, amounting to 187.39 thousand euros, while the sales amount to 1511.38 thousand euros. This view allows for quick identification of the company's most lucrative market areas.

- Analysis Of Customer Lead Sources:

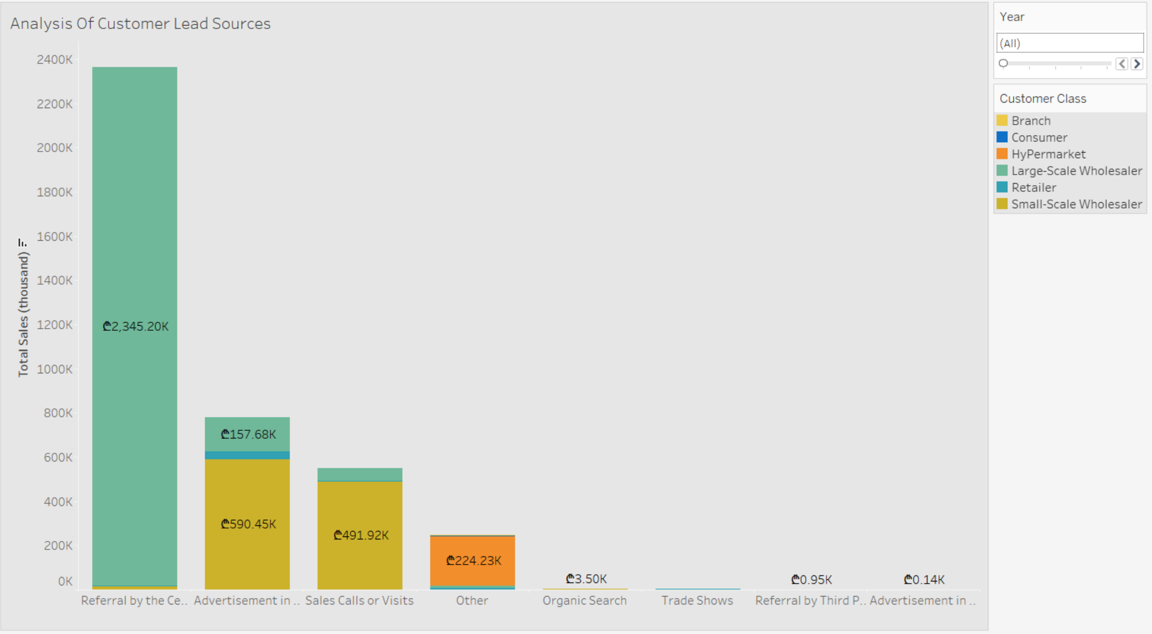


Figure 29 Data Analysis Of Customer Lead Sources

This bar chart analyzes the contribution of different lead sources to sales. It shows that sales brought in by referrals from the central office are the highest, reaching 2345.20 thousand euros. Other channels such as advertising, sales visits, or other methods also contribute, but to a significantly lesser extent compared to referrals. This indicates that central office referrals are the most effective way of customer acquisition for the company.

4.2.2 Business Analysis Dashboard

We have also designed a comprehensive business analysis dashboard that monitors multiple key business indicators on a single interface, providing senior management with a real-time, comprehensive view of business performance.

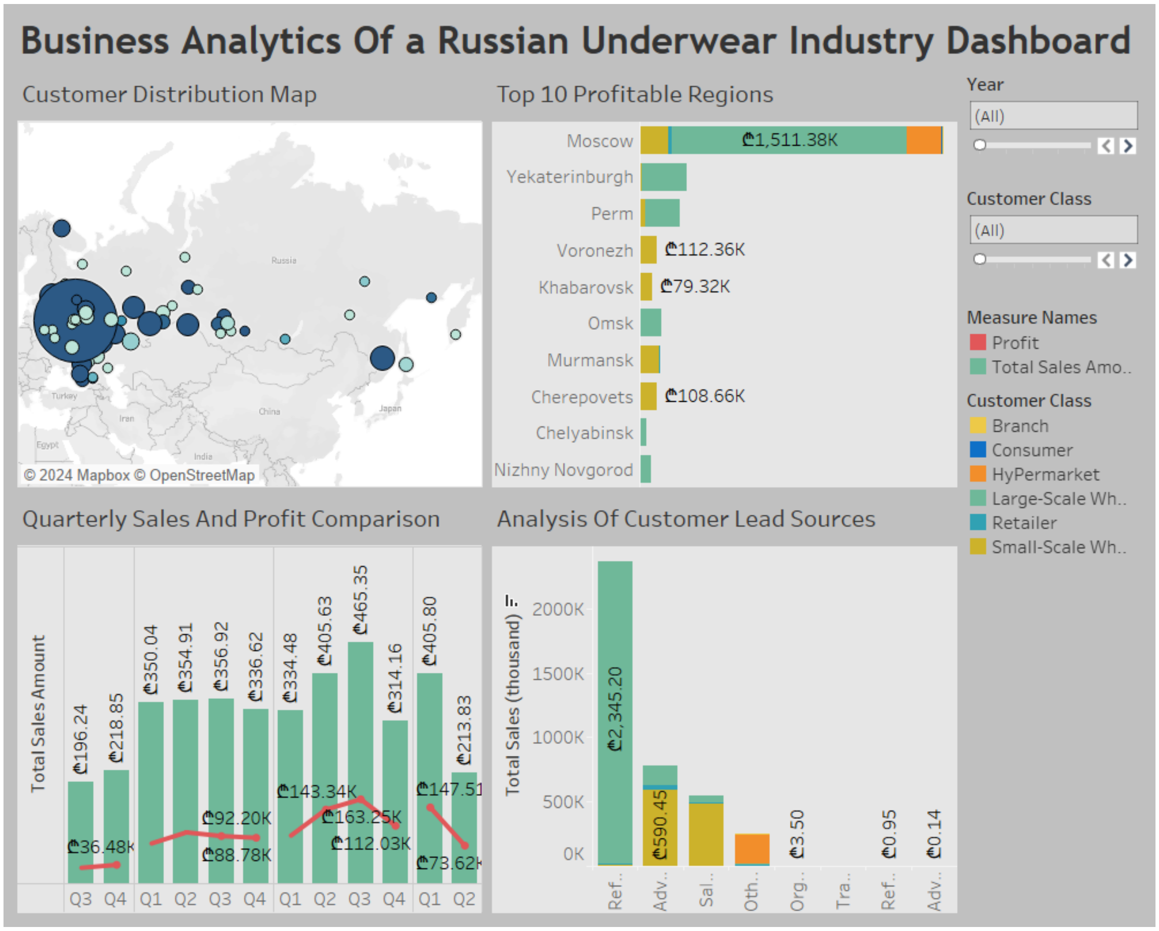


Figure 30 Business Analysis Dashboard

4.3 Data Warehouse Feedback and Improvements

Based on the generated reports and visualizations, we collected feedback from various departments. These valuable opinions help us to continue optimizing the design of the data warehouse to ensure that the results of data analysis better serve the company's business needs.

4.4 Decision Support and Strategic Recommendations

Combining SSRS reports and Tableau visual analysis, we proposed a series of data-driven strategic suggestions for the company, aiming to optimize purchasing strategies, improve sales efficiency, refine logistics management, and strengthen marketing activities.

5. Performance Comparison between Graph Database and Relational Database

5.1 Relational Database vs. Graph Database

This section will explore the comparison between relational databases and graph databases in terms of data query retrieval, aiming to evaluate the performance differences between the two types of databases when processing the same dataset.[7]

5.2 Performance Comparison between Graph Database and Relational Database

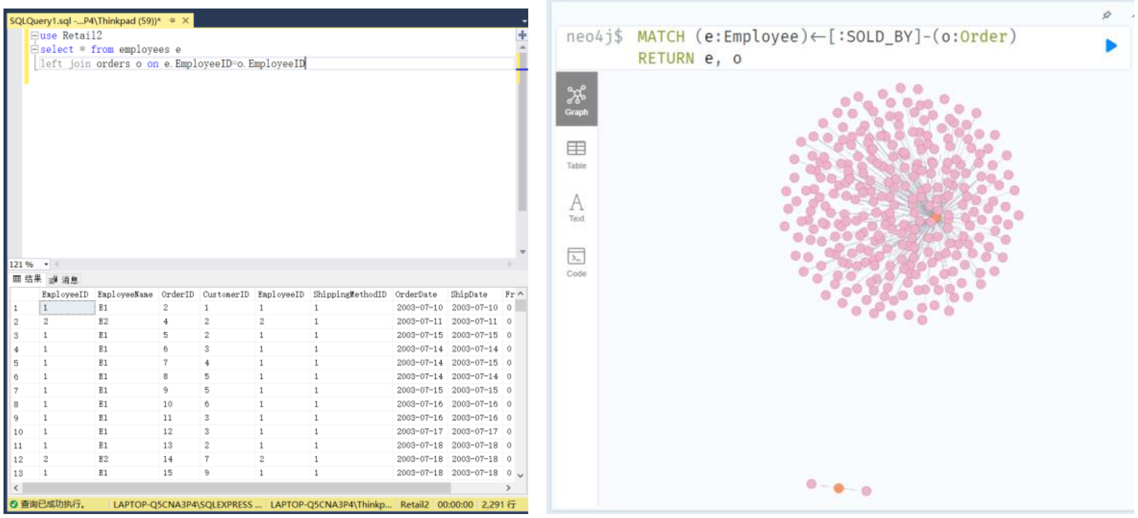
5.2.1 Comparison of 'SELECT ' Query in SQL vs. Neo4j

Query Objective: Retrieve all orders sold by sales staff
Database Statement:

- **SQL:** SELECT * FROM employees e LEFT JOIN orders o ON e.EmployeeID = o.EmployeeID;

- **Neo4j:** MATCH (e:Employee)<-[:SOLD_BY]-(o:Order) RETURN e, o;

Analysis: In SQL, JOIN operations are necessary to associate employees with orders, which adds complexity to the query. In contrast, Neo4j simplifies the query process by visually representing the connections between employees and orders using graph relationships.



```
use Retail2
select * from employees e
left join orders o on e.EmployeeID=o.EmployeeID
```

	EmployeeID	EmployeeName	OrderID	CustomerID	EmployeeID	ShippingMethodID	OrderDate	ShipDate	Fr
1	1	E1	2	1	1	1	2003-07-10	2003-07-10	0
2	2	E2	4	2	2	1	2003-07-11	2003-07-11	0
3	1	E1	5	2	1	1	2003-07-15	2003-07-15	0
4	1	E1	6	3	1	1	2003-07-14	2003-07-14	0
5	1	E1	7	4	1	1	2003-07-14	2003-07-15	0
6	1	E1	8	5	1	1	2003-07-14	2003-07-14	0
7	1	E1	9	5	1	1	2003-07-15	2003-07-15	0
8	1	E1	10	6	1	1	2003-07-16	2003-07-16	0
9	1	E1	11	3	1	1	2003-07-16	2003-07-16	0
10	1	E1	12	3	1	1	2003-07-17	2003-07-17	0
11	1	E1	13	2	1	1	2003-07-18	2003-07-18	0
12	2	E2	14	7	2	1	2003-07-18	2003-07-18	0
13	1	E1	15	9	1	1	2003-07-18	2003-07-18	0

```
查询已成功执行。 LAPTOP-Q5CNA3P4\SQLEXPRESS ... LAPTOP-Q5CNA3P4\Thinkp... Retail2 00:00:00 2,291 行
```

```
neo4j$ MATCH (e:Employee)←[:SOLD_BY]-(o:Order)
       RETURN e, o
```

Figure 31 Results of Retrieve all orders sold by sales staff

5.2.2 Comparison of Filter Query in SQL vs. Neo4j

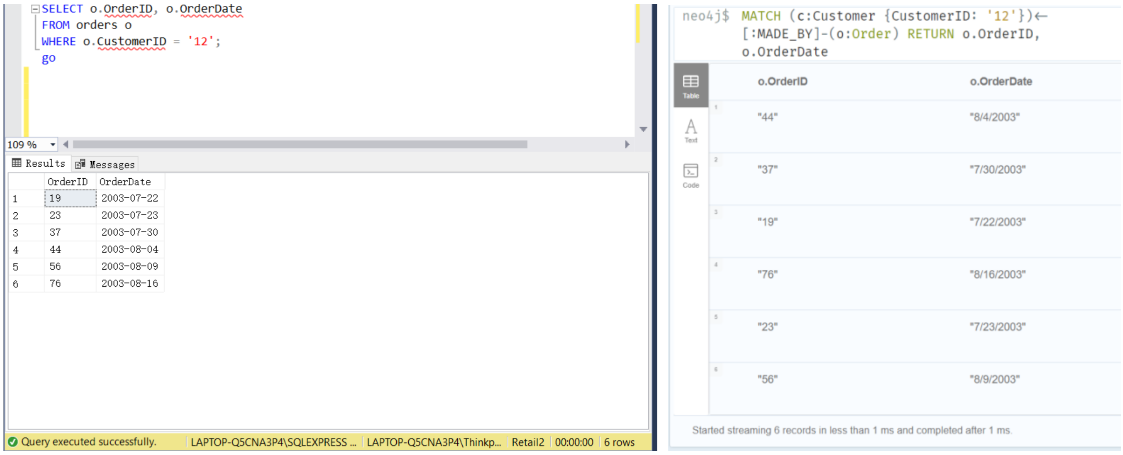
Query Objective: Retrieve all orders for a specific customer

Database Statements:

- **SQL:** SELECT o.OrderID, o.OrderDate FROM orders o WHERE o.CustomerID = '12';

- **Neo4j:** MATCH (c:Customer {CustomerID: '12'})<-[:MADE_BY]-(o:Order) RETURN o.OrderID, o.OrderDate;

Analysis: SQL queries use WHERE clauses to filter specific customer orders, while Neo4j directly uses graphical relationships to define the connection between a customer whose customer ID is 12 and an order. Although both are presented as tables, the use of graphical relationships makes the query statements simpler and more straightforward.



```
SELECT o.OrderID, o.OrderDate
FROM orders o
WHERE o.CustomerID = '12';
go
```

	OrderID	OrderDate
1	19	2003-07-22
2	23	2003-07-23
3	37	2003-07-30
4	44	2003-08-04
5	56	2003-08-09
6	76	2003-08-16

Query executed successfully. | LAPTOP-Q5CNA3P4\SQLEXPRESS ... | LAPTOP-Q5CNA3P4\Thinkp... | Retail2 | 00:00:00 | 6 rows

```
neo4j$ MATCH (c:Customer {CustomerID: '12'})←
       [:MADE_BY]-(o:Order) RETURN o.OrderID,
       o.OrderDate
```

	o.OrderID	o.OrderDate
1	"44"	"8/4/2003"
2	"37"	"7/30/2003"
3	"19"	"7/22/2003"
4	"76"	"8/16/2003"
5	"23"	"7/23/2003"
6	"56"	"8/9/2003"

Started streaming 6 records in less than 1 ms and completed after 1 ms.

Figure 32 Results of Retrieve all orders for a specific customer

5.2.3 Comparison of Group By Query in SQL vs. Neo4j

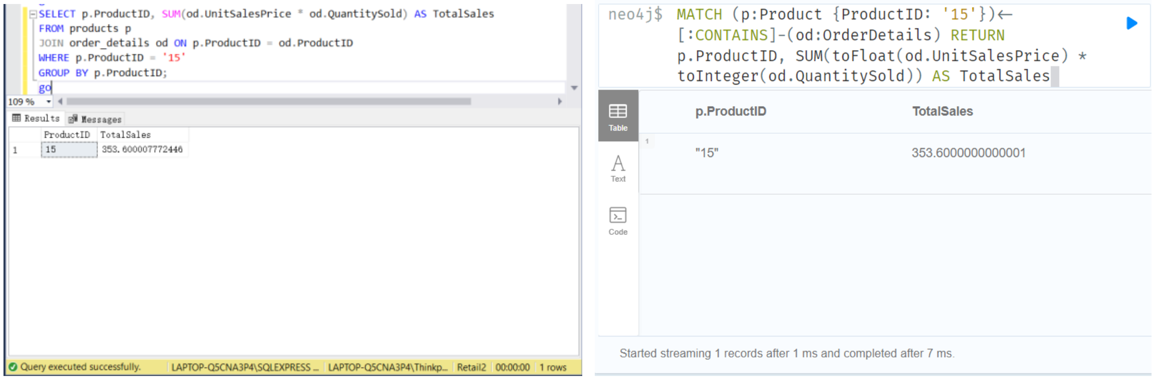
Query Objective: Retrieve the total sales of a specific product

Database Statements:

- **SQL:** SELECT p.ProductID, SUM(od.UnitSalesPrice * od.QuantitySold) AS TotalSales FROM products p JOIN order_details od ON p.ProductID = od.ProductID WHERE p.ProductID = '15' GROUP BY p.ProductID;

- **Neo4j:** MATCH (p:Product {ProductID: '15'})<-[:CONTAINS]-(od:OrderDetails) RETURN p.ProductID, SUM(toFloat(od.UnitSalesPrice) * toInteger(od.QuantitySold)) AS TotalSales;

Analysis: When performing the aggregation function calculation, SQL needs to group in order to perform the aggregation calculation to find the total sales amount, whereas Neo4j can directly perform the aggregation operation through the nodes and relationships of the graph, showing the syntactic simplicity and efficiency of graph databases in dealing with aggregated queries.



```
SELECT p.ProductID, SUM(od.UnitSalesPrice * od.QuantitySold) AS TotalSales
FROM products p
JOIN order_details od ON p.ProductID = od.ProductID
WHERE p.ProductID = '15'
GROUP BY p.ProductID;
go
```

```
   ProductID  TotalSales
1  15         353.600007772446
```

```
Query executed successfully.   LAPTOP-Q5CNA3P4\SQLEXPRESS ...   LAPTOP-Q5CNA3P4\Thinkp...   Retail2   00:00:00   1 rows
```

```
neo4j$ MATCH (p:Product {ProductID: '15'})←
       [:CONTAINS]-(od:OrderDetails) RETURN
       p.ProductID, SUM(toFloat(od.UnitSalesPrice) *
       toInteger(od.QuantitySold)) AS TotalSales
```

```
   p.ProductID   TotalSales
1  "15"          353.6000000000001
```

```
Started streaming 1 records after 1 ms and completed after 7 ms.
```

Figure 33 Results of Retrieve inventory transactions within a specific time frame

5.2.4 Both Bounds Query Comparison in SQL vs. Neo4j

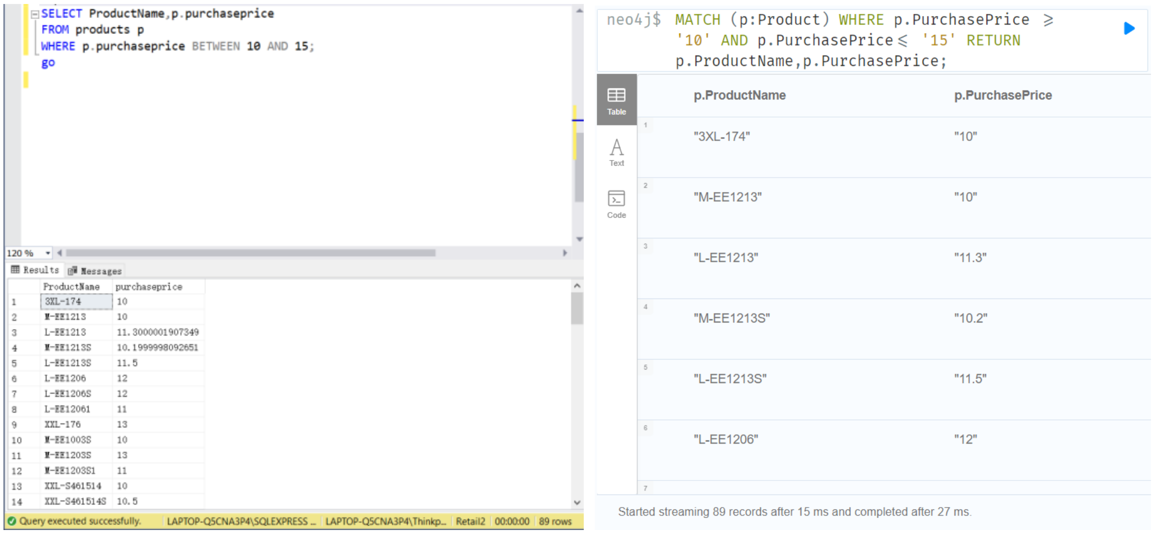
Query Objective: Retrieve products records within a specific price period

Database Statements:

- **SQL:** SELECT ProductName,p.purchaseprice FROM products p WHERE p.purchaseprice BETWEEN 10 AND 15;

- **Neo4j:** MATCH (p:Product) WHERE p.PurchasePrice >= '10' AND p.PurchasePrice<= '15' RETURN p.ProductName,p.PurchasePrice;

Analysis: SQL queries use the BETWEEN clause to specify a range of product purchase prices for precise price segment queries. Neo4j also provides explicit price boundaries, but its query representation is more intuitive and easier to understand and maintain.



```
SELECT ProductName,p.purchaseprice
FROM products p
WHERE p.purchaseprice BETWEEN 10 AND 15;
go
```

	ProductName	purchaseprice
1	3XL-174	10
2	M-EE1213	10
3	L-EE1213	11.3000001907349
4	M-EE1213S	10.1999998092651
5	L-EE1213S	11.5
6	L-EE1206	12
7	L-EE1206S	12
8	L-EE12061	11
9	XXL-176	13
10	M-EE1003S	10
11	M-EE1203S	13
12	M-EE1203S1	11
13	XXL-S461514	10
14	XXL-S461514S	10.5

Query executed successfully. | LAPTOP-Q5CNA3P4\SQLEXPRESS ... | LAPTOP-Q5CNA3P4\Thinkp... | Retail2 | 00:00:00 | 89 rows

```
neo4j$ MATCH (p:Product) WHERE p.PurchasePrice ≥
       '10' AND p.PurchasePrice ≤ '15' RETURN
       p.ProductName,p.PurchasePrice;
```

	p.ProductName	p.PurchasePrice
1	"3XL-174"	"10"
2	"M-EE1213"	"10"
3	"L-EE1213"	"11.3"
4	"M-EE1213S"	"10.2"
5	"L-EE1213S"	"11.5"
6	"L-EE1206"	"12"

Started streaming 89 records after 15 ms and completed after 27 ms.

Figure 34 Results of Retrieve inventory transaction records within a specific time period

5.2.5 Join Query Comparison in SQL vs. Neo4j

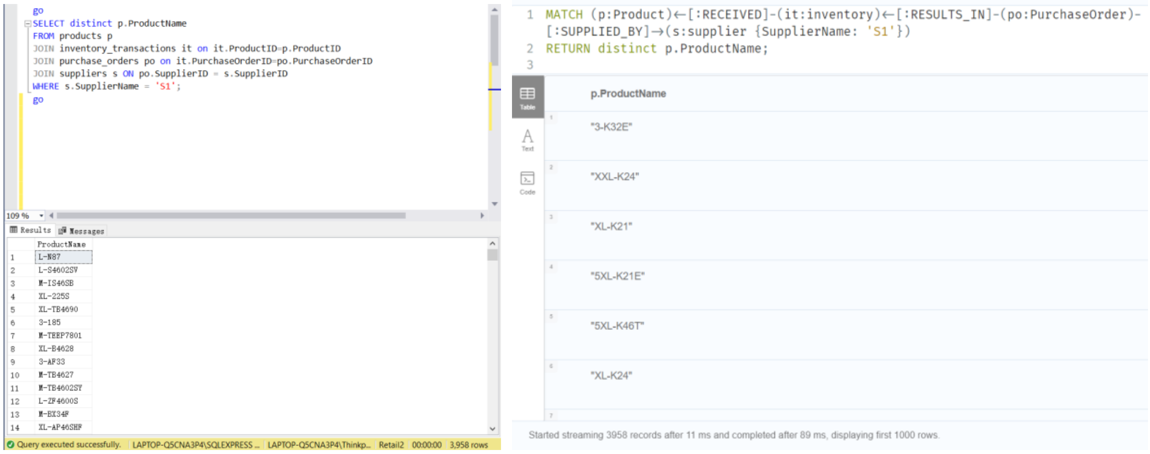
Query Objective: Retrieve all products supplied by Supplier S1

Database Statements:

- **SQL:** SELECT DISTINCT p.ProductName FROM products p JOIN inventory_transactions it ON it.ProductID = p.ProductID JOIN purchase_orders po ON it.PurchaseOrderID = po.PurchaseOrderID JOIN suppliers s ON po.SupplierID = s.SupplierID WHERE s.SupplierName = 'S1';

- **Neo4j:** MATCH (p:Product)<-[:RECEIVED]-(it:Inventory)<-[:RESULTS_IN]-(po:PurchaseOrder)-[:SUPPLIED_BY]->(s:Supplier {SupplierName: 'S1'}) RETURN DISTINCT p.ProductName;

Analysis: SQL queries involve multiple JOIN operations to connect related tables, which increases the complexity of the query. In contrast, Neo4j shows the relationship path between products and suppliers directly through its graph database structure, making the statements more concise and potentially more efficient.



```
go
SELECT distinct p.ProductName
FROM products p
JOIN inventory_transactions it on it.ProductID=p.ProductID
JOIN purchase_orders po on it.PurchaseOrderID=po.PurchaseOrderID
JOIN suppliers s ON po.SupplierID = s.SupplierID
WHERE s.SupplierName = 'S1';
go
```

	ProductName
1	L-N87
2	L-S4602SV
3	M-IS46SB
4	XL-225S
5	XL-TB4690
6	3-185
7	M-TEEP7801
8	XL-B4628
9	3-AF33
10	M-TB4627
11	M-TB4602SY
12	L-ZF4600S
13	M-BX34F
14	XL-AP46SHF

Query executed successfully. | LAPTOP-Q5CNA3P4\SQLEXPRESS ... | LAPTOP-Q5CNA3P4\Thinkp... | Retail2 | 00:00:00 | 3,958 rows

```
MATCH (p:Product)←[:RECEIVED]-(it:inventory)←[:RESULTS_IN]-(po:PurchaseOrder)-
[:SUPPLIED_BY]→(s:supplier {SupplierName: 'S1'})
RETURN distinct p.ProductName;
```

	p.ProductName
1	"3-K32E"
2	"XXL-K24"
3	"XL-K21"
4	"5XL-K21E"
5	"5XL-K46T"
6	"XL-K24"

Started streaming 3958 records after 11 ms and completed after 89 ms, displaying first 1000 rows.

Figure 35 Results of Retrieve all products supplied by Supplier S1

5.2.6 Count Query Comparison in SQL vs. Neo4j

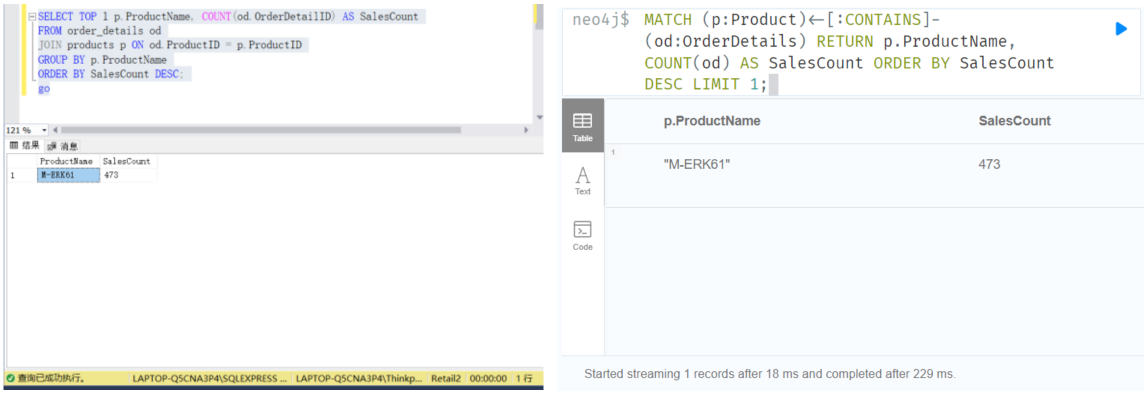
Query Objective: Identify the product with the most sales records

Database Statements:

- **SQL:** SELECT TOP 1 p.ProductName, COUNT(od.OrderDetailID) AS SalesCount FROM order_details od JOIN products p ON od.ProductID = p.ProductID GROUP BY p.ProductName ORDER BY COUNT(od.OrderDetailID) DESC;

- **Neo4j:** MATCH (p:Product)<-[:CONTAINS]-(od:OrderDetails) RETURN p.ProductName, COUNT(od) AS SalesCount ORDER BY SalesCount DESC LIMIT 1;

Analysis: In SQL, the query involves aggregate functions and sorting, requiring explicit use of GROUP BY and ORDER BY statements; Neo4j, on the other hand, completes the counting and sorting directly through graph relationships and built-in functions, showing the directness and efficiency of graph databases in handling such aggregate data.



```
SELECT TOP 1 p.ProductName, COUNT(od.OrderDetailID) AS SalesCount
FROM order_details od
JOIN products p ON od.ProductID = p.ProductID
GROUP BY p.ProductName
ORDER BY SalesCount DESC;
go
```

	ProductName	SalesCount
1	M-ERK61	473

查询已成功执行。 | LAPTOP-Q5CNA3P4\SQLEXPRESS ... | LAPTOP-Q5CNA3P4\Thinkp... | Retail2 | 00:00:00 | 1 行

```
neo4j$ MATCH (p:Product)←[:CONTAINS]-
       (od:OrderDetails) RETURN p.ProductName,
       COUNT(od) AS SalesCount ORDER BY SalesCount
       DESC LIMIT 1;
```

	p.ProductName	SalesCount
1	"M-ERK61"	473

Started streaming 1 records after 18 ms and completed after 229 ms.

Figure 36 Results of Identify the product with the most sales records

5.2.7 Like Query Comparison in SQL vs. Neo4j

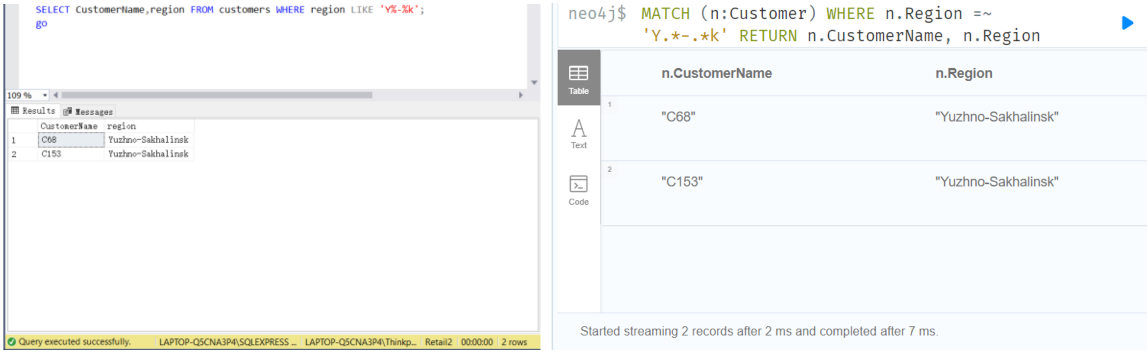
Query Objective: Fuzzily search for customer names in a specific region

Database Statements:

- **SQL:** SELECT DISTINCT region FROM customers WHERE region LIKE 'Y%-%k';

- **Neo4j:** MATCH (n:Customer) WHERE n.Region =~ 'Y.*-.*k' RETURN n.CustomerName, n.Region;

Analysis: SQL uses the LIKE clause for fuzzy matching, whereas Neo4j uses regular expressions, offering more flexible matching options. Graph databases demonstrate a stronger expressive capability and possible performance advantages when handling fuzzy match queries.



```
SELECT CustomerName,region FROM customers WHERE region LIKE 'Y%-%k';
go
```

	CustomerName	region
1	C68	Yuzhno-Sakhalinsk
2	C153	Yuzhno-Sakhalinsk

Query executed successfully. | LAPTOP-Q5CNA3P4\SQLEXPRESS ... | LAPTOP-Q5CNA3P4\Thinkp... | Retail2 | 00:00:00 | 2 rows

```
neo4j$ MATCH (n:Customer) WHERE n.Region =~
'Y.*-.*k' RETURN n.CustomerName, n.Region
```

	n.CustomerName	n.Region
1	"C68"	"Yuzhno-Sakhalinsk"
2	"C153"	"Yuzhno-Sakhalinsk"

Started streaming 2 records after 2 ms and completed after 7 ms.

Figure 37 Results of Fuzzily search for customer names in a specific region

By comparing and analysing, we have deduced the applicability of relational databases and graph databases in different business use cases, providing guidance for the selection of enterprise information system architecture.

6. Conclusion

In this project, our primary task was to build a data warehouse for a Russian underwear trading company. Combining the company's business background and requirements, we successfully created a database model with 11 entities, using a star schema to optimize data storage and support advanced data analysis.

During the process of importing data sources, we faced several technical challenges, especially in the data extraction phase. Due to improper initial configurations of the connection manager, data could not be correctly imported. We resolved this issue by switching to an OLEDB connection manager. Additionally, when using auto-generated SQL for table connections, we encountered issues with incomplete data imports, which we resolved by modifying the relevant SQL codes. These experiences taught us the importance of thorough technical evaluations and pre-testing at the start of the project to ensure compatibility and efficiency of all data management tools.

To meet the needs of stakeholders, we developed SSRS reports and Tableau visualizations and dashboards, helping company leaders gain a comprehensive understanding of business processes, identify market opportunities, and thereby formulate more precise business strategies. The successful implementation of the project not only enhanced the company's data-driven decision-making capabilities but also strengthened our experience in data warehouse construction and business analysis.

Additionally, we explored the performance differences between graph databases and relational databases in data retrieval, to understand the appropriate data construction scenarios for each, providing valuable experience for future projects. This project underscored the importance of efficient data management and analysis tools in modern business competition. Despite some challenges, through team effort and technical adjustments, we ultimately achieved the project goals, creating substantial business value for the company.

7. Reference

[1] A. Nambiar and D. Mundra, “An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management,” Big Data and Cognitive Computing, vol. 6, no. 4. MDPI, Dec. 01, 2022. doi: 10.3390/bdcc6040132.

[2] “CRISP-DM The New Blueprint for Data Mining Colin Shearer (Fall 2000)”.

[3] “Full details and actions for The data warehouse toolkit: the definitive guide to dimensional modeling.” Accessed: Apr. 13, 2024. [Online]. Available: https://www.vlebooks.com/Product/Index/1027730?page=0&startBookmarkId=-1

[4] “Full details and actions for Agile Data Science 2.0: Building Full-stack Data Analytics Applications With Spark.” Accessed: Apr. 13, 2024. [Online]. Available: https://www.vlebooks.com/Product/Index/1097874?page=0&startBookmarkId=-1

[5] A. Dhaouadi, K. Bousselmi, M. M. Gammoudi, S. Monnet, and S. Hammoudi, “Data Warehousing Process Modeling from Classical Approaches to New Trends: Main Features and Comparisons,” Data (Basel), vol. 7, no. 8, Aug. 2022, doi: 10.3390/data7080113.

[6] “Tableau Community.” Accessed: Apr. 13, 2024. [Online]. Available: https://www.tableau.com/community

[7] “Neo4j Graph Database & Analytics | Graph Database Management System.” Accessed: Apr. 13, 2024. [Online]. Available: https://neo4j.com/?utm_source=Google&utm_medium=PaidSearch&utm_campaign=Evergreenutm_content%3DEMEA-Search-SEMBrand-Evergreen-None-SEM-SEM-NonABM&utm_term=neo4j&utm_adgroup=core-brand&gad_source=1&gclid=Cj0KCQjw2uiwBhCXARIsACMvIU049f8mqTiz5Q-TUtIRSEApUS2ffxg6jDXkK9FBIw-c1b0XRnoSNOgaArt1EALw_wcB

8.Appendix

```
use Retail_DB2
go
CREATE TABLE Date_Dim (
    Datekey INT PRIMARY KEY IDENTITY,
    Date DATE NOT NULL,
    Year INT,
    Quarter varChar(25),
    Month INT,
    Week varchar(25),
    Day INT
);
go
CREATE TABLE Customer_Dim (
       Customerkey INT PRIMARY KEY IDENTITY,
    CustomerID INT,
    CustomerName VARCHAR(255),
    Region VARCHAR(255),
    Country VARCHAR(255),
    CustomerClass VARCHAR(255),
    LeadSource VARCHAR(255)
);
go
CREATE TABLE Supplier_Dim (
       Supplierkey INT PRIMARY KEY IDENTITY,
    SupplierID INT ,
    SupplierName VARCHAR(255),
);
go
CREATE TABLE Product_Dim (
    Productkey INT PRIMARY KEY IDENTITY,
       ProductID INT,
    ProductName VARCHAR(255),
    Category VARCHAR(255),
    PurchasePrice DECIMAL(10,2)
);
go
CREATE TABLE PaymentMethod_Dim (
    PaymentMethodkey INT PRIMARY KEY IDENTITY,
       PaymentMethodID INT,
    PaymentMethod VARCHAR(255)
);
go
CREATE TABLE ShippingMethod_Dim (
    ShippingMethodkey INT PRIMARY KEY IDENTITY,
       ShippingMethodID INT,
    ShippingMethod VARCHAR(255)
);
go
CREATE TABLE Employee_Dim (
    EmployeeKey INT PRIMARY KEY IDENTITY(1,1),
    EmployeeID INT unique NOT NULL,
    EmployeeName VARCHAR(255)
);
go
CREATE TABLE Sales_Fact (
    SalesID INT PRIMARY KEY IDENTITY,
    DateKey INT,
    CustomerKey INT,
    ProductKey INT,
       EmployeeKey INT,
       ShippingMethodKey INT,
    PaymentMethodKey INT,
    OrderDetailID INT,
    Quantity INT,
    TotalSalesAmount DECIMAL(10,2),
    Profit DECIMAL(10,2),
    FOREIGN KEY (DateKey) REFERENCES Date_Dim(Datekey),
    FOREIGN KEY (CustomerKey) REFERENCES Customer_Dim(Customerkey),
    FOREIGN KEY (ProductKey) REFERENCES Product_Dim(Productkey),
    FOREIGN KEY (EmployeeKey) REFERENCES Employee_Dim(EmployeeKey),
       FOREIGN KEY (ShippingMethodKey) REFERENCES
ShippingMethod_Dim(ShippingMethodkey),
    FOREIGN KEY (PaymentMethodKey) REFERENCES PaymentMethod_Dim(PaymentMethodkey)
);
go
CREATE TABLE Purchases_Fact (
    PurchaseID INT PRIMARY KEY IDENTITY,
    DateKEY INT,
    SupplierKEY INT,
    ProductKEY INT,
       EmployeeKey INT,
       TransactionID INT,
       Quantityordered INT,
    QuantityPurchased INT,
       ShippingMethodKey INT,
    TotalPurchaseOrderAmount DECIMAL(10,2),
       TotalPurchaseAmount DECIMAL(10,2),
    FOREIGN KEY (DateKEY) REFERENCES Date_Dim(Datekey),
    FOREIGN KEY (SupplierKEY) REFERENCES Supplier_Dim(Supplierkey),
       FOREIGN KEY (EmployeeKey) REFERENCES Employee_Dim(EmployeeKey),
    FOREIGN KEY (ProductKEY) REFERENCES Product_Dim(Productkey),
       FOREIGN KEY (ShippingMethodKey) REFERENCES
ShippingMethod_Dim(ShippingMethodkey)
```