FlipKart Data Warehouse Project - Final Report

1. Summary of the Project Approach and Implementation

The goal of this data warehouse project was to turn raw FlipKart e-commerce data into an analytical tool that lets businesses learn about their data and make decisions based on that data. The execution was done in a planned way, with five steps that all fit together:

Data Warehouse Platform and Tools Used:

Data warehouse Platform: Microsoft SQL Server

- It can be used as a data warehouse platform because it has columnstore indexes for analytical queries, the ability to partition big tables, and the best performance for star schema designs.
- Selected because it supports dimensional modeling ideas by default and works well with both normalized and star schema designs

ETL Tool Used: T-SQL Microsoft SQL Server

- Used for tasks like getting data, changing it, and loading it.
- Picked because it gives straight access to databases, works well with sets, and makes workflow easier without needing any extra tools.

1.1 Data Sourcing

To start the project, three related datasets were chosen: Customers_FlipKart, Products_FlipKart, and Sales_FlipKart. These datasets covered the most important parts of an e-commerce business. These files had information about customers' backgrounds, what they bought, how much they paid, and records of transactions.

There was a lot of information about customers in the Customers_FlipKart dataset. It included demographic information like birth year, level of schooling, marital status, income, and number of children. It also had useful information about how people behave, such as how recently they bought something, their reward scores, and how much they spent on different types of products.

The Products_FlipKart dataset had a lot of information about the catalog of products, like names of products, sections, subcategories, brands, prices, and stock levels. A product_id, which was the main key, was used to mark each item as unique.

The Sales_FlipKart dataset was the main fact table in the raw data. It had records of transactions that included order times, quantities, discounts, sales amounts, payment methods, and shipping information. In the raw data format, this dataset had a basic star schema because it used foreign keys (customer_id and product_id) to connect customers to goods.

A detailed data dictionary was made to list the attributes and connections between these datasets. This dictionary defined fields clearly and laid the groundwork for the method to integrating data.

1.2 Normalized Database Implementation

In the second step, the raw data had to be changed into a Third Normal Form (3NF) database format that was properly normalized. Before moving on to the dimensional model, this important step made sure that the data was correct and consistent.
Before normalization could happen, the raw datasets had to be carefully looked over to find entities, attributes, and connections. This study showed that the original three datasets needed to be broken up into seven properly adjusted tables:

A. Customers: This table held basic information about each customer, like their primary key (customer_id), year_birth, school, marital_status, income, number of children, and the date they registered. We got rid of duplicate data and made a single point of truth for customer information by separating demographic data.

B. CustomerPurchaseStats: This table kept track of how customers bought things and was related to the Customers table by the customer_id foreign key. It had information like how much was spent on different types of products, deals that were bought, and preferred ways to make purchases. By getting rid of transitive relationships, this separation followed the rules of 3NF.

C. Categories: A generated category_id served as the main key for this lookup table, which held unique combos of category and subcategory data. This step of leveling kept category information from being used more than once in different product records.

D. Products: This table had information about products, such as the main key (product_id), the name of the product, its price, its brand, and a foreign key relationship to the Categories table. By getting rid of category duplication, this framework made storage more efficient.

E. Locations: The main key in this table was location_id, which kept unique combinations of country, state, and city information. This way, location hierarchies wouldn't be repeated across transaction records.

F. Orders: The order_id was the primary key in this table, which held information about the order header. It also had foreign key links to the Customers and Locations tables, which held the core transaction data.

G. OrderDetails: This table, OrderDetails, held transaction line items with transaction_id as the main key and foreign key relationships to the Orders and Products tables. It had information like quantity, discount, and payment details.

This normalized structure got rid of duplicate data while keeping data consistency by using the right primary and foreign key relationships. By using 3NF, we made sure that each non-key attribute only depended on the main key and not on any other non-key attributes. This made the base for future ETL processes stable.

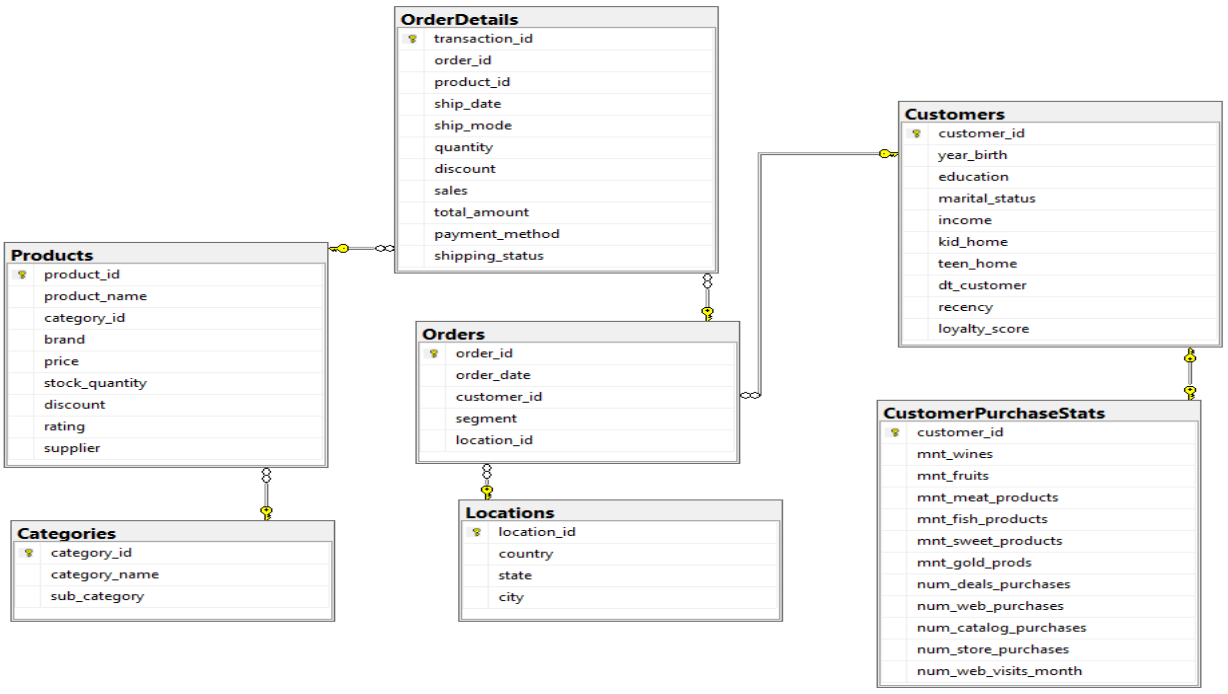
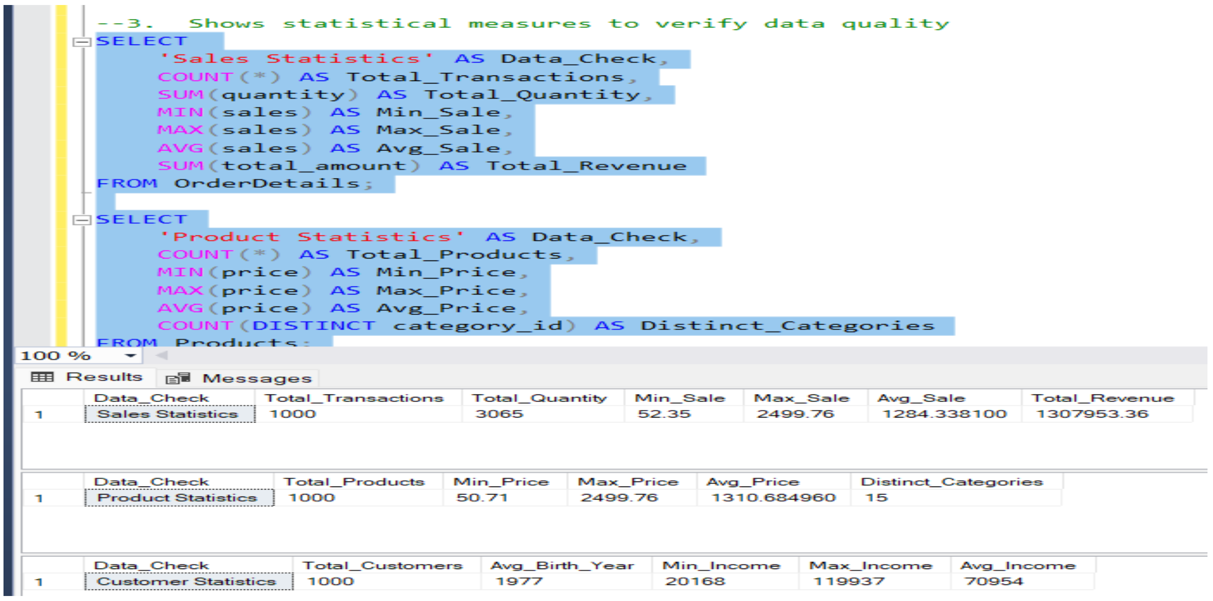


Fig 1- ER Diagram

There were confirmation queries that made sure all 1,000 records from each source file were loaded correctly into the normalized structure, and that all relationships maintained their referential integrity.



```
--3.  Shows statistical measures to verify data quality
SELECT
    'Sales Statistics' AS Data_Check,
    COUNT(*) AS Total_Transactions,
    SUM(quantity) AS Total_Quantity,
    MIN(sales) AS Min_Sale,
    MAX(sales) AS Max_Sale,
    AVG(sales) AS Avg_Sale,
    SUM(total_amount) AS Total_Revenue
FROM OrderDetails;

SELECT
    'Product Statistics' AS Data_Check,
    COUNT(*) AS Total_Products,
    MIN(price) AS Min_Price,
    MAX(price) AS Max_Price,
    AVG(price) AS Avg_Price,
    COUNT(DISTINCT category_id) AS Distinct_Categories
FROM Products;
```

100 %

Results | Messages

	Data_Check	Total_Transactions	Total_Quantity	Min_Sale	Max_Sale	Avg_Sale	Total_Revenue
1	Sales Statistics	1000	3065	52.35	2499.76	1284.338100	1307953.36

	Data_Check	Total_Products	Min_Price	Max_Price	Avg_Price	Distinct_Categories
1	Product Statistics	1000	50.71	2499.76	1310.684960	15

	Data_Check	Total_Customers	Avg_Birth_Year	Min_Income	Max_Income	Avg_Income
1	Customer Statistics	1000	1977	20168	119937	70954

Fig 2- Verifying data integrity with statistics

1.3 ETL Implementation(T-SQL SQL Server)

It was the ETL process that connected the normalized business database (DB_FlipKart) to the dimensional data warehouse (FlipKart_EDW).

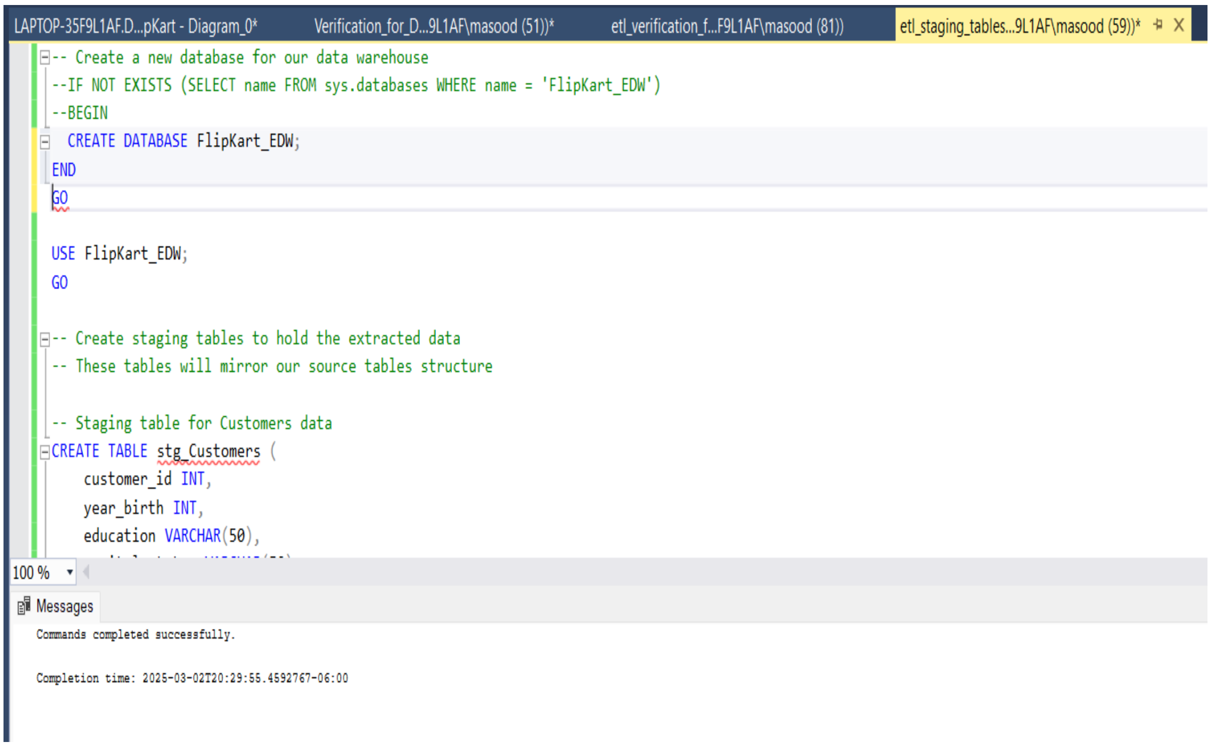
We used T-SQL in SQL Server to perform ETL process.

This process of change was carefully planned and carried out in four separate steps:

Phase 1: Extraction

During the extraction process, data was taken from the normalized database tables and put into staging tables in the data warehouse database. This staging method made a clear separation between the operational and analytical settings, which let changes happen without affecting the source system.

We made staging tables for each entity (stg_Customers, stg_CustomerPurchaseStats, stg_Products, stg_Categories, stg_Orders, stg_OrderDetails, and stg_Locations) to hold the extracted data briefly. **We used T-SQL in SQL Server** to do INSERT-SELECT actions to move data from tables that have been normalized to tables that are being used for staging.


LAPTOP-35F9L1AF.D...pKart - Diagram_0*
Verification_for_D...9L1AF\masood (51))*
etl_verification_f...F9L1AF\masood (81))
etl_staging_tables...9L1AF\masood (59))*
100 %
Messages

```
-- Create a new database for our data warehouse
--IF NOT EXISTS (SELECT name FROM sys.databases WHERE name = 'FlipKart_EDW')
--BEGIN
  CREATE DATABASE FlipKart_EDW;
END
GO

USE FlipKart_EDW;
GO

-- Create staging tables to hold the extracted data
-- These tables will mirror our source tables structure

-- Staging table for Customers data
CREATE TABLE stg_Customers (
    customer_id INT,
    year_birth INT,
    education VARCHAR(50),
```

```
Commands completed successfully.

Completion time: 2025-03-02T20:29:55.4592767-06:00
```

Fig 3- ETL_Extraction

Phase 2: Dimension Creation

In the second step, the staged data were turned into properly structured dimension tables. There are now six main dimensions:

- Customer Dimension: Used SCD Type 2 tracking and a derived CustomerSegment property to sort customers into two groups based on their loyalty scores: Premium and Standard.

- Product Dimension: Made with denormalized category and subcategory data from the Categories table and a derived StockStatus property to show the status of inventory.

- The Location Dimension was created with geographic hierarchies and a Region property that was derived to make geographic analysis easier.

- Date Dimension: Makes up a full set of dates from 2010 to 2025, with different date attributes for time-based research.

- Payment Method Dimension: This dimension is made up of the different payment methods that were used in the transaction data.

- Shipping Method: This dimension is made up of the different shipping ways that were listed in the transaction data.

Each dimension was given surrogate keys to improve the speed of joins and the right tracking fields for changes in the past when they were needed.

Phase 3: Fact Table Population

Each dimension was given surrogate keys to improve the speed of joins and the right tracking fields for changes in the past when they were needed.We filled out two fact tables after setting up all the measurement tables:

The Sales Fact Table (fact_Sales) held information about individual sales, including the amount sold, the price, the total amount, and the number of sales. Through substitute keys, it kept connections with all dimensions.

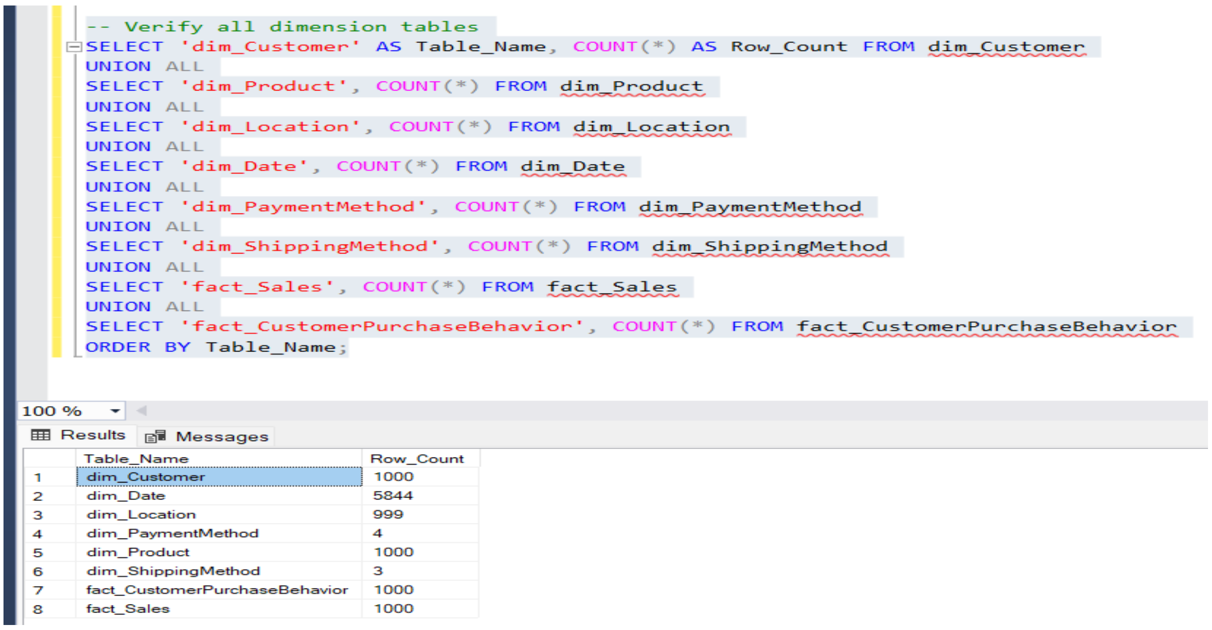
Customer Purchase Behavior fact table (fact_CustomerPurchaseBehavior) shows how customers buy things. This secondary fact table showed how much customers spent on different types of products and through different channels of purchase. It made the statistics more customer-centered.

To load the fact table, staging tables and dimension tables had to be joined together to get the right surrogate keys, derived measures had to be calculated, and proper referential integrity had to be ensured.

Phase 4: Verification

In the last step, thorough checks were made to make sure the data quality and consistency throughout the ETL process:

- Checking the dimension table made sure that the record numbers and SCD Type 2 implementations were correct.
- Checking the facts in the table made sure that the measurements were right and that the references were correct.
- Data quality checks showed that key fields did not have any NULL values and that relationships were being kept up to date.



```
-- Verify all dimension tables
SELECT 'dim_Customer' AS Table_Name, COUNT(*) AS Row_Count FROM dim_Customer
UNION ALL
SELECT 'dim_Product', COUNT(*) FROM dim_Product
UNION ALL
SELECT 'dim_Location', COUNT(*) FROM dim_Location
UNION ALL
SELECT 'dim_Date', COUNT(*) FROM dim_Date
UNION ALL
SELECT 'dim_PaymentMethod', COUNT(*) FROM dim_PaymentMethod
UNION ALL
SELECT 'dim_ShippingMethod', COUNT(*) FROM dim_ShippingMethod
UNION ALL
SELECT 'fact_Sales', COUNT(*) FROM fact_Sales
UNION ALL
SELECT 'fact_CustomerPurchaseBehavior', COUNT(*) FROM fact_CustomerPurchaseBehavior
ORDER BY Table_Name;
```

	Table_Name	Row_Count
1	dim_Customer	1000
2	dim_Date	5844
3	dim_Location	999
4	dim_PaymentMethod	4
5	dim_Product	1000
6	dim_ShippingMethod	3
7	fact_CustomerPurchaseBehavior	1000
8	fact_Sales	1000

Fig 4- Dimension and Fact Tables Verification

1.4 Dimensional Modeling

The dimensional model used a star schema design and had two fact tables (fact_Sales and fact_CustomerPurchaseBehavior) and six dimension tables

(Customer, Product, Location, Date, PaymentMethod, and ShippingMethod). This setup made the database better for analytical queries by:

- Denormalization makes the question structure easier to understand.
- Giving business-oriented table and property names that are easy to understand
- Making aggregation and screening work more efficiently
- Keeping track of the past through SCD Type 2 application

A dimensional bus matrix was used to show how the facts and variables related to each other in the model.

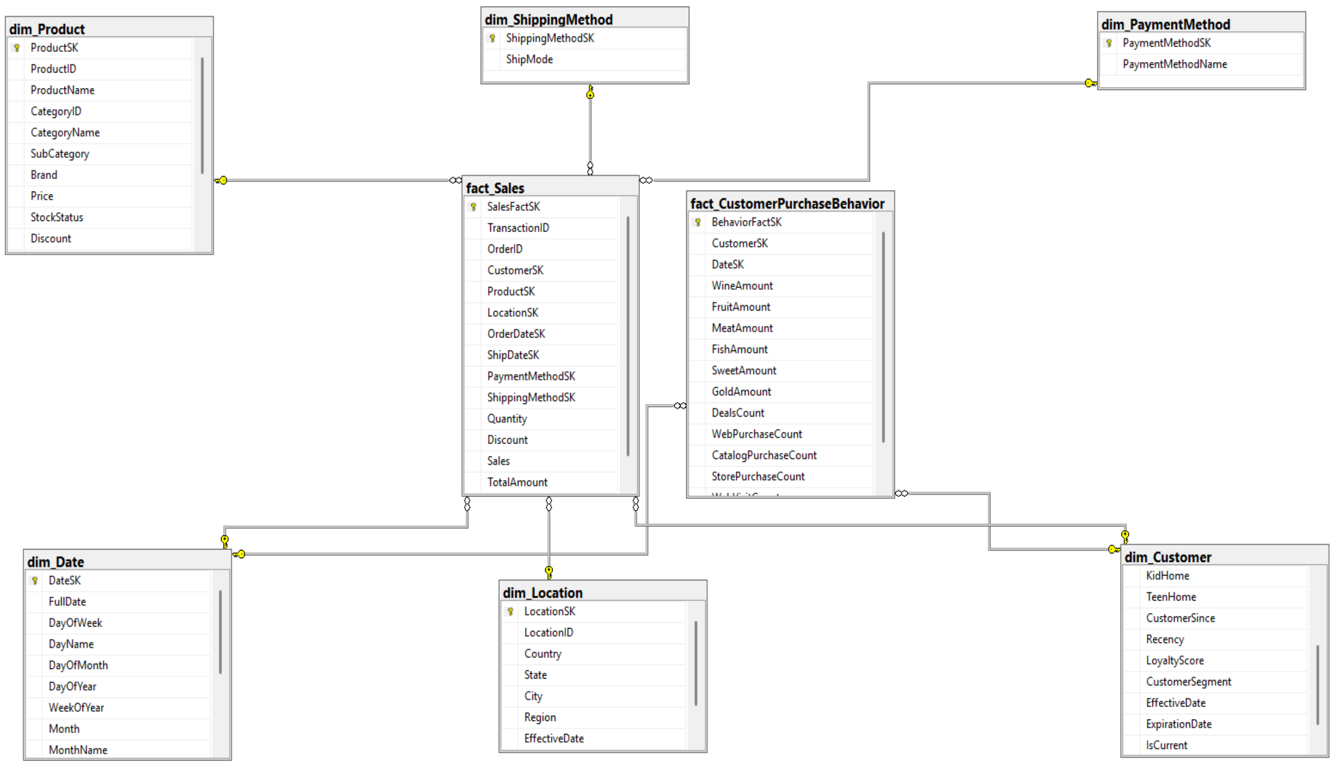


Fig 4.a Star Schema Diagram

1.5 Analytical Querying

Lastly, a set of analytical questions were created to show what the dimensional model could do and get useful business information from the data warehouse. It was these queries that showed how powerful the star schema is for allowing complex business analysis with simple SQL.
There were four different kinds of analytical questions put in place:

a. Sales Performance Analysis

It looked at how sales have changed over time and how well a product is doing. To give you an example, the "Sales Performance by Product Category and Quarter" query showed seasonal trends that were unique to each product category and each KPI.
Visualizing this data made it easy to see clear patterns.

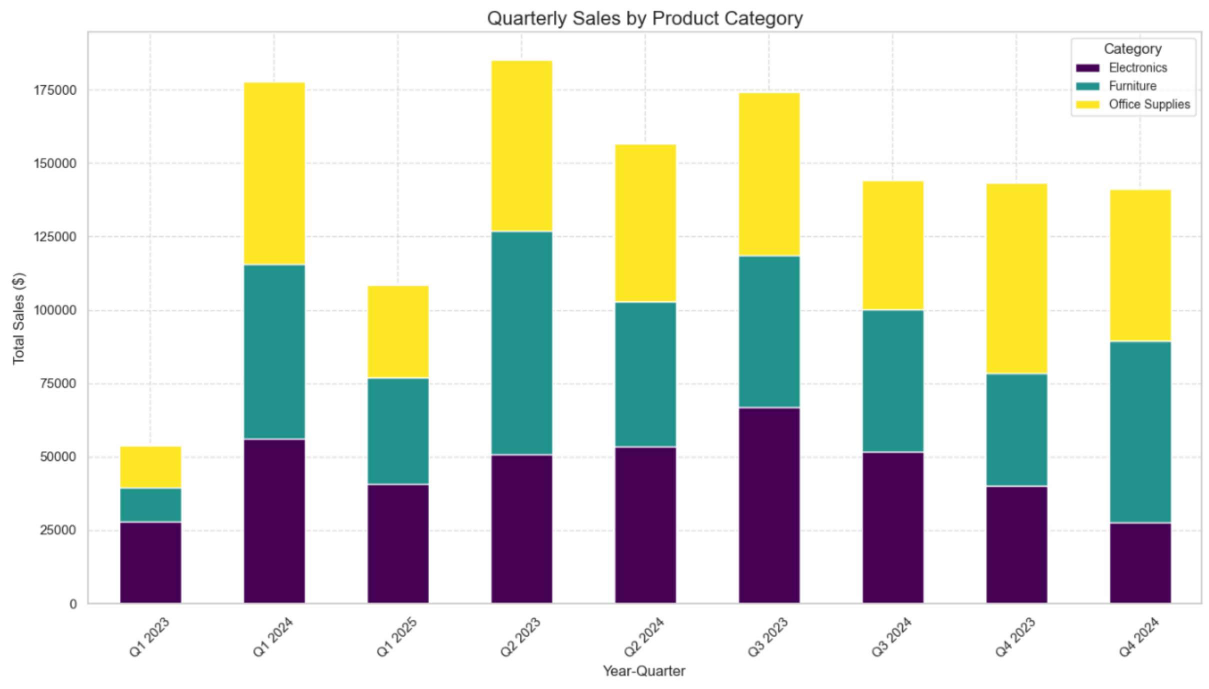


Fig 5- Stacked Bar Chart

b. Customer Segmentation Analysis

The queries looked at the behavior and revenue of different types of customers. The query "Customer Segment Profitability Analysis" broke down key performance measures by customer segment, which showed patterns that were not predicted.

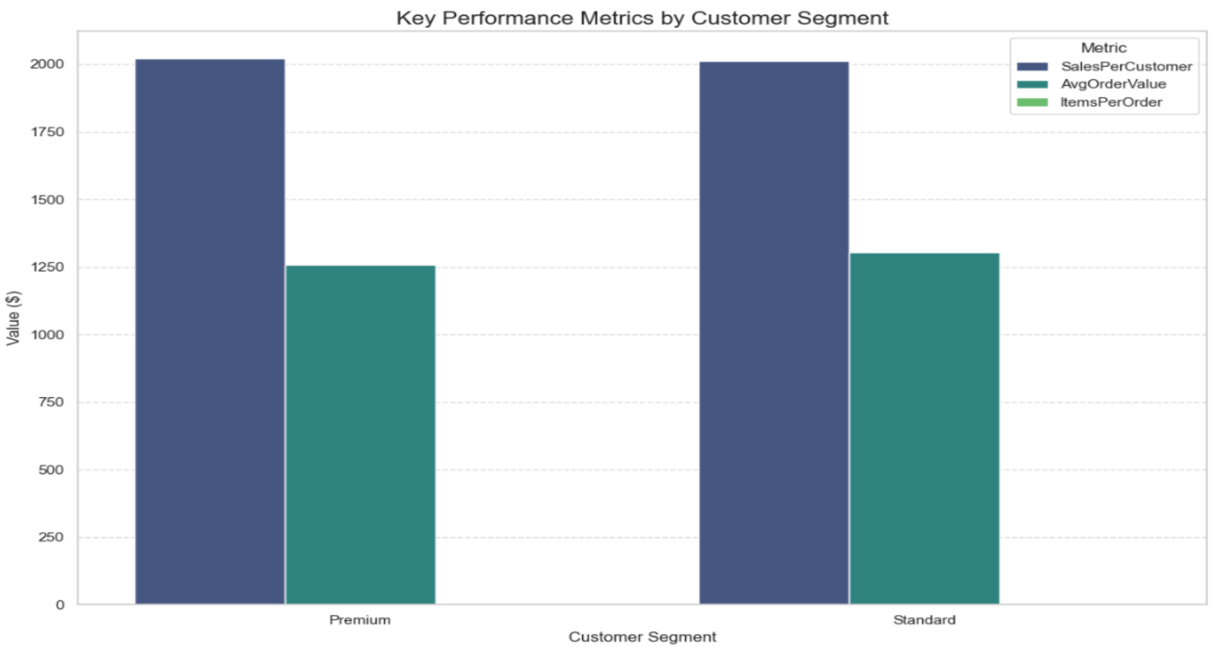


Fig 6- Grouped Bar chart

c. Geographic Sales Analysis

These inquiries looked at how well sales were doing in different parts of the world, including how efficient shipping was. The "Regional Sales Performance with Shipping Analysis" question showed that all sales ($1,284,338) were from the US, which suggests that the business could grow internationally.

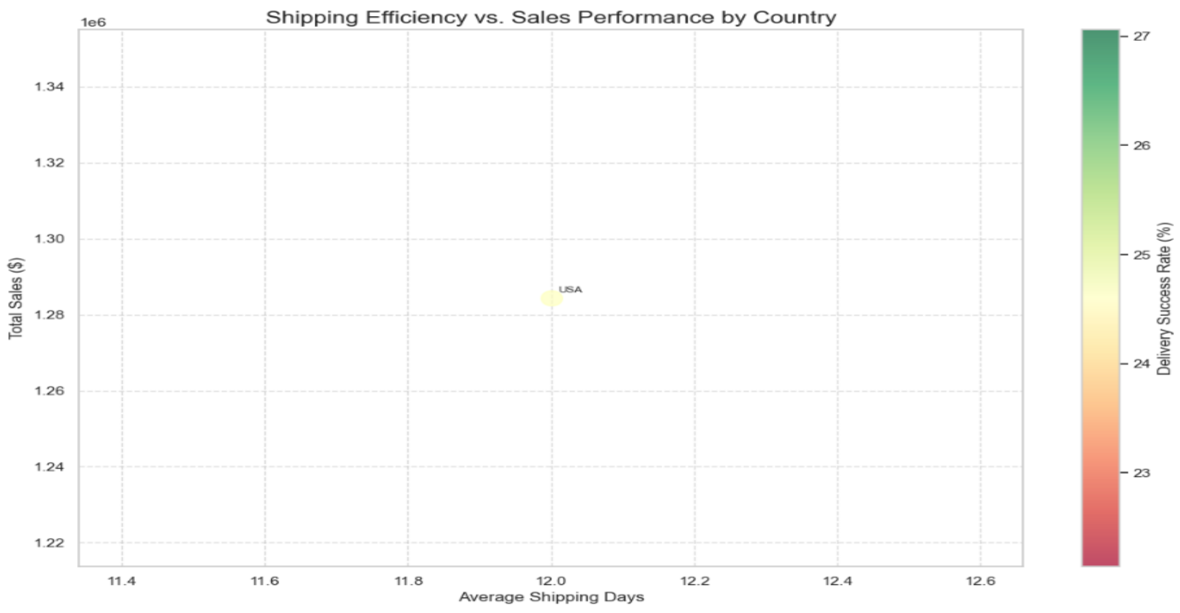


Fig 7- Scatter Plot for Geographic sales

Complex Multi-Dimensional Analysis

These queries gave us a lot of different business information. Time, product, place, customer, and fulfillment factors were used in "Comprehensive Sales Analysis by Multiple Dimensions" to show complex patterns.

The data showed that Premium customers, who were willing to pay more for faster shipping, chose Same Day shipping. Standard customers, on the other hand, used a wider range of shipping choices, with Express shipping being the most popular for office supplies.

The "Advanced Year-over-Year Comparative Analysis" of customer lifetime value and cross-category shopping trends showed that the Premium and Standard sectors had the same lifetime value ($2,020 vs. $2,013), even though they bought different things more often.

Each query had both tabular and graphic outputs (bar charts, scatter plots, heatmaps, and radar charts) so that business users could see the results. These business ideas showed how useful the data warehouse is for making decisions based on data.

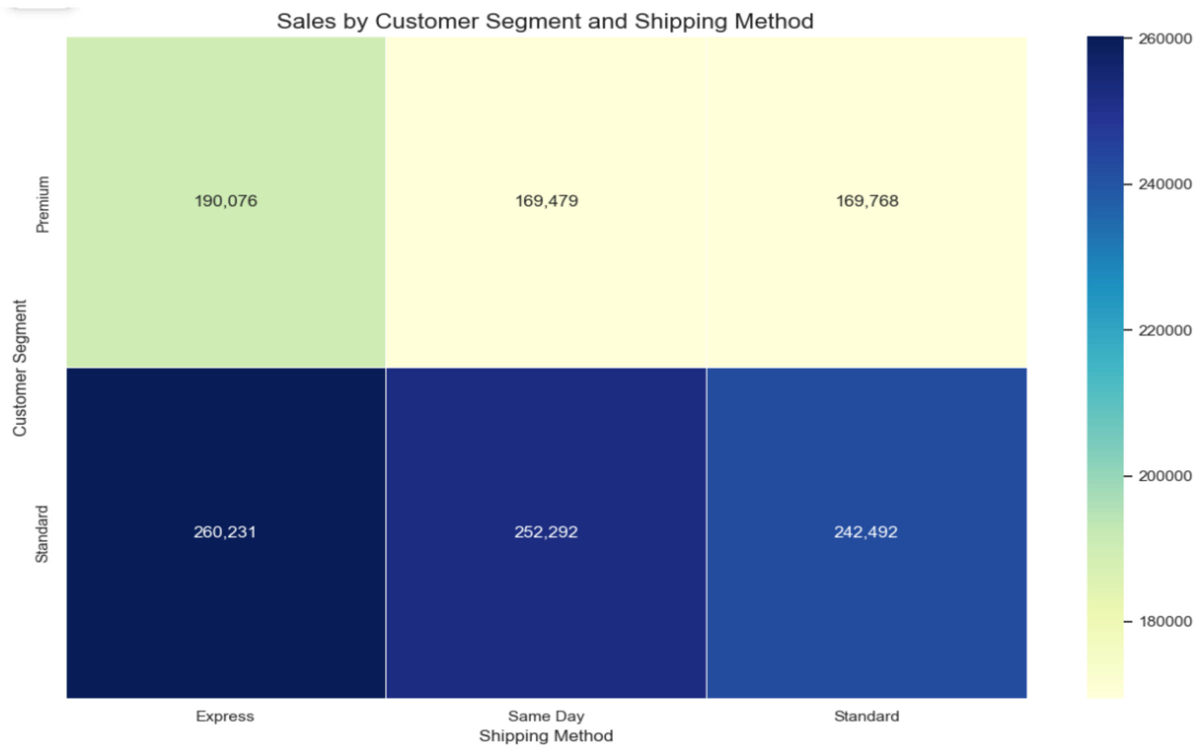


Fig 8- Heatmap of Sales by Customer Segment and Shipping Method

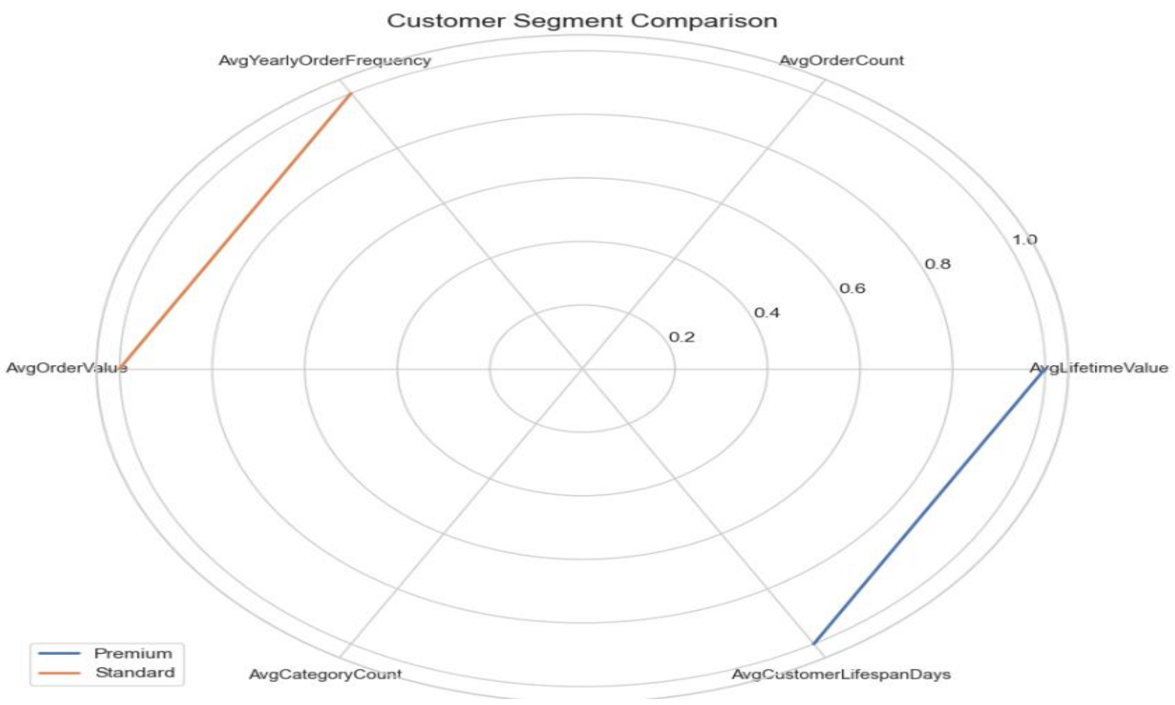


Fig 9- Radar Chart for Customer Segment Comparison

2. Discussion of Challenges Encountered and Solutions Applied

During the project's execution, a number of major problems were found and dealt with in a planned way:

2.1 Data Quality and Standardization

Challenge: The raw data had different formats and NULL values, and it needed to be standardized across all datasets.

Solution: During the extraction step, a structured data cleaning process was put in place. Among these were:

- Making text fields the same to ensure accuracy
- Taking care of NULL values with ISNULL() calls
- Making string dates into correct DATE data types
- Checking groups of numbers and fixing outliers

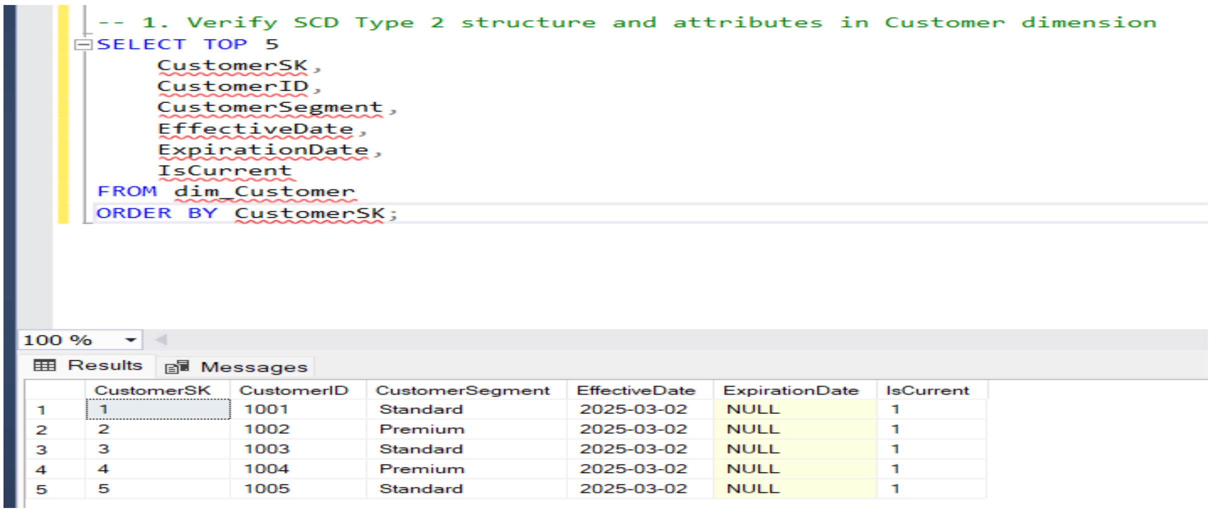
The fact that the record numbers and value distributions were the same in both the source and target systems showed that the data had been cleaned and standardized correctly.

2.2 Implementing Slowly Changing Dimensions (SCD Type 2)

Problem: It was hard to keep track of changes in dimension traits over time because they needed complicated logic and extra fields that weren't in the source data.
Solution: An all-encompassing plan for implementing SCD Type 2 was made, which included:

- Adding tracking fields for EffectiveDate, ExpirationDate, and IsCurrent
- Creating rules to help find and handle present and past records
- Making sure that fact tables always use the most up-to-date versions of dimension records
- Putting together a system for verification to check the SCD implementation

The end result was a strong system that could keep track of changes over time to customer traits, product details, and location data, which let them be analyzed in the past.



```sql
-- 1. Verify SCD Type 2 structure and attributes in Customer dimension
SELECT TOP 5
    CustomerSK,
    CustomerID,
    CustomerSegment,
    EffectiveDate,
    ExpirationDate,
    IsCurrent
FROM dim_Customer
ORDER BY CustomerSK;
```

100 %

Results | Messages

	CustomerSK	CustomerID	CustomerSegment	EffectiveDate	ExpirationDate	IsCurrent
1	1	1001	Standard	2025-03-02	NULL	1
2	2	1002	Premium	2025-03-02	NULL	1
3	3	1003	Standard	2025-03-02	NULL	1
4	4	1004	Premium	2025-03-02	NULL	1
5	5	1005	Standard	2025-03-02	NULL	1

Fig 10- SCD Type 2 Verification

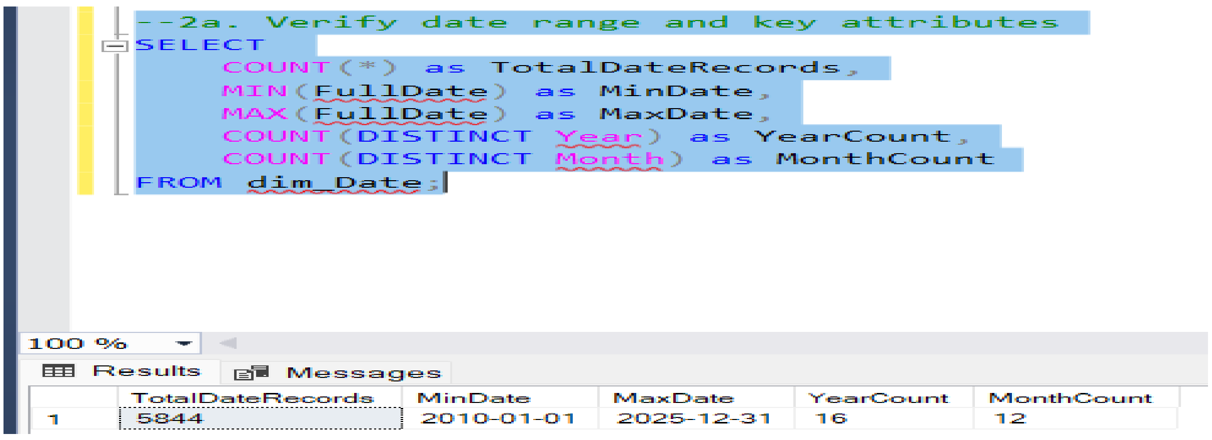
2.3 Date Dimension Generation

Problem: To make a full date dimension with business-relevant traits, data that wasn't in the source systems had to be created.

Solution: A custom process for creating date dimensions was put in place that:

- Made all the dates in the range that was asked for (2010–2025)
- Date properties like day of the week, month, quarter, and year were calculated.
- Weekends and holidays were named.
- Made substitute keys based on date values to make joins go more smoothly.

This detailed date dimension made it possible to do time-based research using a variety of time scales and business calendars.



```
--2a. Verify date range and key attributes
SELECT
    COUNT(*) as TotalDateRecords,
    MIN(FullDate) as MinDate,
    MAX(FullDate) as MaxDate,
    COUNT(DISTINCT Year) as YearCount,
    COUNT(DISTINCT Month) as MonthCount
FROM dim_Date;
```

	TotalDateRecords	MinDate	MaxDate	YearCount	MonthCount
1	5844	2010-01-01	2025-12-31	16	12

Fig 11- Verify date range and key date attributes

2.4 Surrogate Key Management

Problem: During the ETL process, it was hard and prone to mistakes to keep natural keys and surrogate keys consistent.

Solution: A methodical approach to managing surrogate keys was put in place:

- Making replacement keys that are unique for each dimension
- keeping track of how natural keys and substitute keys are mapped
- Making sure lookups are always done while the fact table is loading
- Adding steps to check the security of the key

This method made sure that there were no NULL values in any foreign key fields, so there was perfect referential integrity between the fact and dimension tables.

2.5 Performance Optimization for Complex Queries

Problem: Analytical questions that were too complicated and had a lot of joins and aggregations could slow things down.

Solution: To improve query speed, the dimensional model was made better by:

- Denormalizing dimension tables to make joins easier
- Using derived attributes to reduce the amount of processing needed during query execution
- Setting up the right indexing techniques
- Using substitute keys to make joins run faster

Because of these improvements, even the most complicated multidimensional studies could be run quickly.

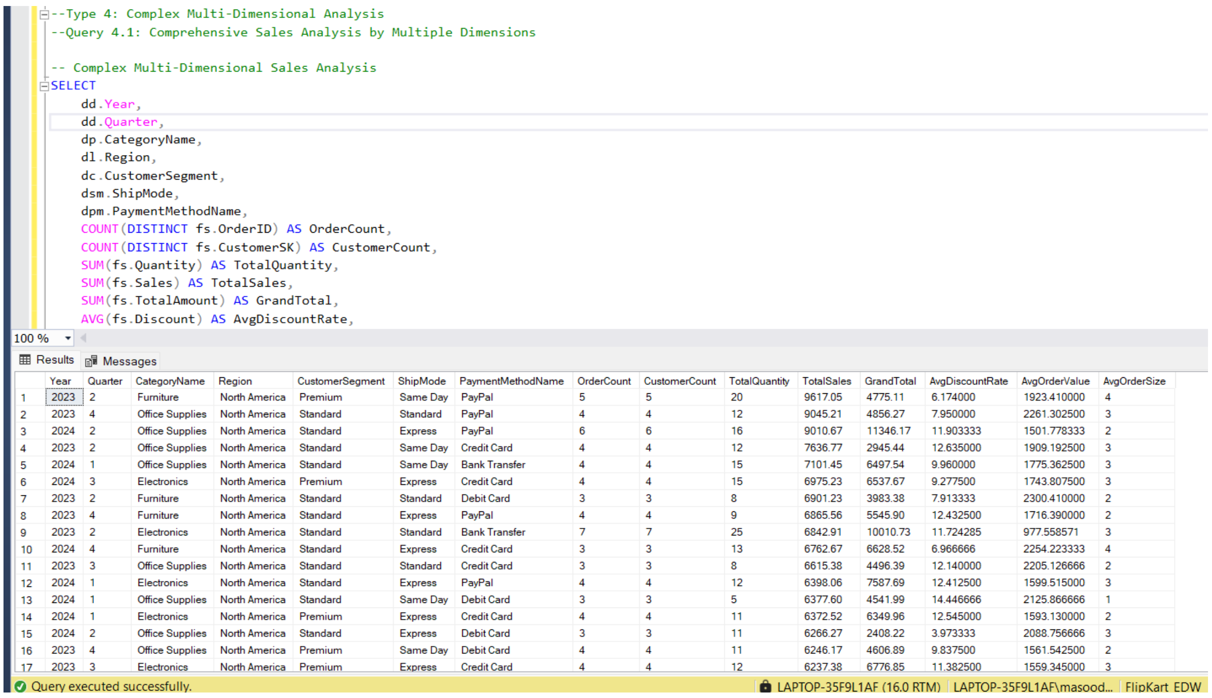
3. Analysis of the Effectiveness of the Dimensional Model

Several important factors can be used to judge how well the dimensional model works:

3.1 Query Simplification and Performance

The star schema design made analytical searches a lot easier. In order to run searches against the normalized database, you would have had to do complicated joins across seven or more tables. But with the dimensional model, you could run simple queries that made it clear how facts and dimensions related to each other.

For instance, the complicated multidimensional analysis query joined six dimensions to the fact table without any performance problems, showing how well the model works. Because of this simplification, business researchers can use the data warehouse without needing to know a lot about SQL.



```sql
--Type 4: Complex Multi-Dimensional Analysis
--Query 4.1: Comprehensive Sales Analysis by Multiple Dimensions

-- Complex Multi-Dimensional Sales Analysis
SELECT
    dd.Year,
    dd.Quarter,
    dp.CategoryName,
    dl.Region,
    dc.CustomerSegment,
    dsm.ShipMode,
    dpm.PaymentMethodName,
    COUNT(DISTINCT fs.OrderID) AS OrderCount,
    COUNT(DISTINCT fs.CustomerSK) AS CustomerCount,
    SUM(fs.Quantity) AS TotalQuantity,
    SUM(fs.Sales) AS TotalSales,
    SUM(fs.TotalAmount) AS GrandTotal,
    AVG(fs.Discount) AS AvgDiscountRate,
```

	Year	Quarter	CategoryName	Region	CustomerSegment	ShipMode	PaymentMethodName	OrderCount	CustomerCount	TotalQuantity	TotalSales	GrandTotal	AvgDiscountRate	AvgOrderValue	AvgOrderSize
1	2023	2	Furniture	North America	Premium	Same Day	PayPal	5	5	20	9617.05	4775.11	6.174000	1923.410000	4
2	2023	4	Office Supplies	North America	Standard	Standard	PayPal	4	4	12	9045.21	4856.27	7.950000	2261.302500	3
3	2024	2	Office Supplies	North America	Standard	Express	PayPal	6	6	16	9010.67	11346.17	11.903333	1501.778333	2
4	2023	2	Office Supplies	North America	Standard	Same Day	Credit Card	4	4	12	7636.77	2945.44	12.635000	1909.192500	3
5	2024	1	Office Supplies	North America	Standard	Same Day	Bank Transfer	4	4	15	7101.45	6497.54	9.960000	1775.362500	3
6	2024	3	Electronics	North America	Premium	Express	Credit Card	4	4	15	6975.23	6537.67	9.277500	1743.807500	3
7	2023	2	Furniture	North America	Standard	Standard	Debit Card	3	3	8	6901.23	3983.38	7.913333	2300.410000	2
8	2023	4	Furniture	North America	Standard	Express	PayPal	4	4	9	6865.56	5545.90	12.432500	1716.390000	2
9	2023	2	Electronics	North America	Standard	Standard	Bank Transfer	7	7	25	6842.91	10010.73	11.724285	977.558571	3
10	2024	4	Furniture	North America	Standard	Express	Credit Card	3	3	13	6762.67	6628.52	6.966666	2254.223333	4
11	2023	3	Office Supplies	North America	Standard	Standard	Credit Card	3	3	8	6615.38	4496.39	12.140000	2205.126666	2
12	2024	1	Electronics	North America	Standard	Express	PayPal	4	4	12	6398.06	7587.69	12.412500	1599.515000	3
13	2024	1	Office Supplies	North America	Standard	Same Day	Debit Card	3	3	5	6377.60	4541.99	14.446666	2125.866666	1
14	2024	1	Electronics	North America	Premium	Express	Credit Card	4	4	11	6372.52	6349.96	12.545000	1593.130000	2
15	2024	2	Office Supplies	North America	Standard	Express	Debit Card	3	3	11	6266.27	2408.22	3.973333	2088.756666	3
16	2023	4	Office Supplies	North America	Premium	Same Day	Debit Card	4	4	11	6246.17	4606.89	9.837500	1561.542500	2
17	2023	3	Electronics	North America	Premium	Express	Credit Card	4	4	12	6237.38	6776.85	11.382500	1559.345000	3

```
Query executed successfully.    LAPTOP-35F9L1AF (16.0 RTM) | LAPTOP-35F9L1AF\masood... | FlipKart_EDW
```

Fig 12- Complex Multi-Dimensional Sales Analysis

3.2 Support for Business Requirements

The dimensional model worked well for all the business studies that were needed:

a. Sales Analysis: The model let you look at sales in great depth by product, time, place, and type of customer.

b. Customer Segmentation: The customer dimension with derived segments made it easier to look at revenue by type of customer.

c. Geographic Analysis: The location dimension with its hierarchies and areas made it easier to look at how well different regions were doing.

d. Time-Based Analysis: The date factor helped find seasonal patterns and looked for trends.

The detailed analytical queries showed that the model met all of the research and reporting needs of the business.

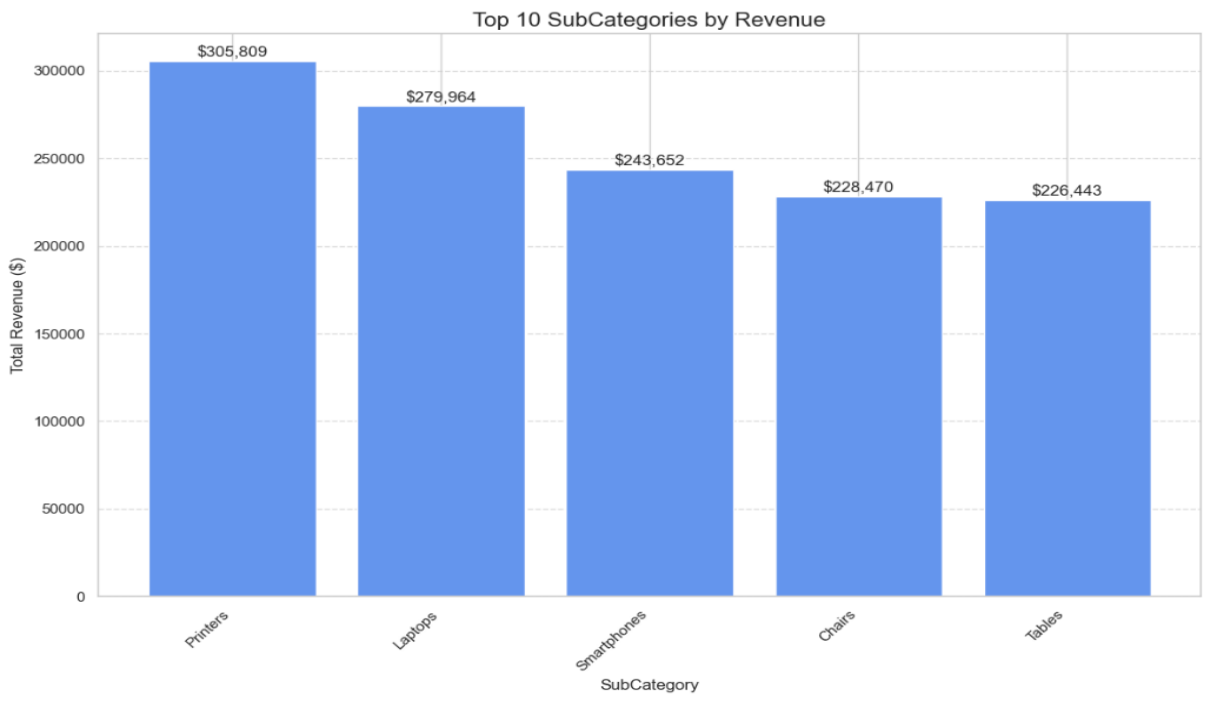


Fig 13- Bar Chart- Top 5 SubCategories by Revenue

3.3 Historical Tracking Capabilities

Implementing SCD Type 2 gave strong historical tracking tools for important variables. All records in the current implementation are marked as "current" (IsCurrent=1) because this is the first load. However, the structure is there to keep track of changes to customer segments, product attributes, and location information in the future.
As the data warehouse changes, this feature will become more useful because it lets you look at how changes in traits affect business performance over time.

3.4 Business Value of Derived Attributes

The derived qualities made the dimensional model much more useful for business:

- Targeted marketing analysis was possible by putting customers into groups based on their loyalty numbers, such as Premium or Standard.
- Status of the Product Stock: Sorting goods by inventory levels (High, Medium, Low) helped with inventory management analysis.
- Geographic Regions: Putting places into regions made it easier to study geography.

These derived traits turned technical data into information that was useful for business, making the model more useful for people making decisions.

3.5 Scalability and Extensibility

Scalability and flexibility were taken into account when the dimensional model was created:

- It's easy for the star schema to add new dimensions or make current ones bigger.
- As the business needs change, the fact tables can be updated to include new measurements.
- The SCD Type 2 application will be able to keep track of the past even as the data changes.

This gives the data warehouse the ability to change with the needs of the business without having to be completely redesigned.

4. Recommendations for Future Improvements

Based on the experience of putting the current model into action and a study of it, the following suggestions for future improvements are made:

4.1 Enhanced Customer Segmentation

The current customer segmentation shows that the lifetime value of customers in the Premium and Standard groups is about the same ($2,020 vs. $2,013). This suggests that the segmentation criteria could be made more specific.

<u>Recommendation:</u> Use a more advanced classification model that takes into account more than just loyalty scores, such as

- RFM stands for "Recency, Frequency, and Monetary."
- Preferences for purchase categories
- Ways to get new customers
- How people react to sales and deals

This better grouping would make it easier to target marketing efforts with more useful differences.

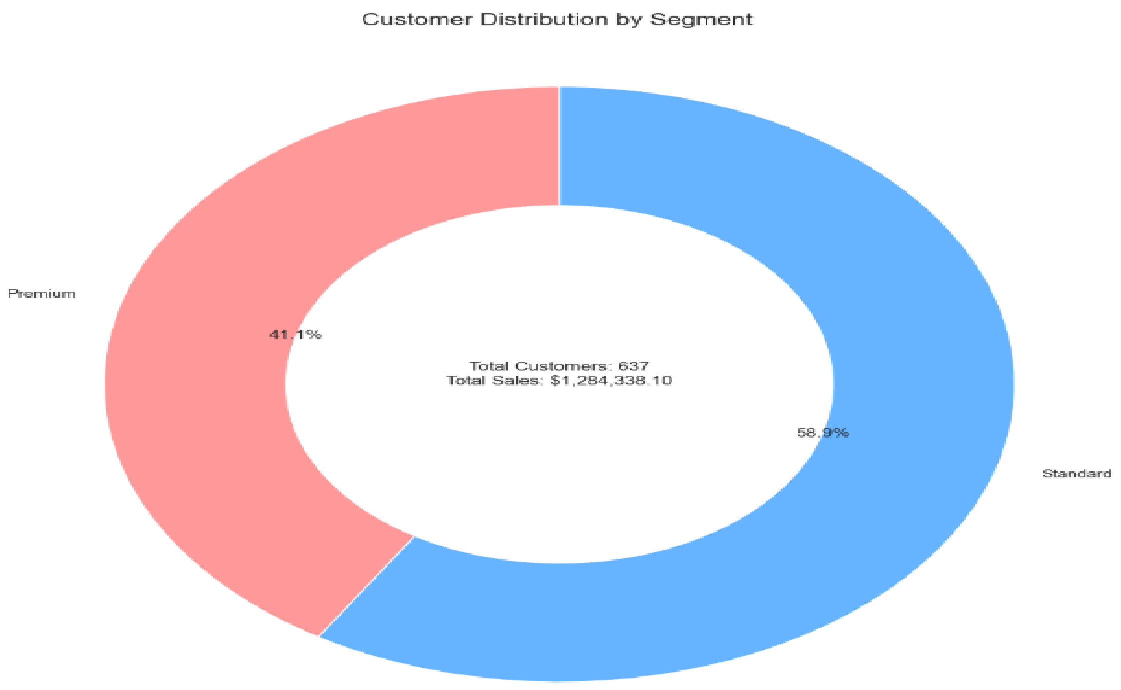


Fig 14- Donut Chart- Customer Distribution by Segment

4.2 Implementation of Real-Time Data Integration

The current ETL process is set up to work in batches, which means that it might not get the most recent transaction data for analysis.

Recommendation: Come up with a way to integrate info in real time or almost real time that:

- Records events as they happen
- Updates the detailed model bit by bit
- Offers more up-to-date information for making operating decisions
- Utilizes the SCD Type 2 method to keep track of the past

Change data capture (CDC) methods or streaming data integration tools could be used to make this happen.

4.3 Expansion of Product Hierarchy and Attributes

There is a simple hierarchy of categories and subcategories in the product dimension right now. This could be improved to allow for more in-depth product research.

Recommendation: Add the following to the product dimension:

- More layers of the hierarchy, such as product family and brand category
- More information about the object, like its size, color, and materials
- Better derived attributes (for example, stage of the product lifecycle and margin group)

This expansion would make it possible to analyze product success at a more detailed level and help with managing product portfolios.

4.4 Geographic Expansion and Enhancement

The current data only shows places in North America, which makes it harder to look at the global market.

Recommendation: The dimensional model for global growth should be ready by:

- Making the place factor bigger to help global markets
- Adding support for more than one language for product and group names
- Adding the ability to change currencies in fact tables
- Setting up regional systems that match how businesses report

This improvement would help with studying foreign markets and growing businesses around the world.

4.5 Automated Data Quality Monitoring

At the moment, checking the quality of the data is done by hand using SQL searches.
Recommendation: Set up automated tracking of data quality that:

- Checks regularly for problems with data accuracy
- Checks that business rules and limits are followed
- Warns managers about possible data issues
- Keeps track of data quality metrics over time

This technology would keep the data quality high and lower the chance of making bad business decisions based on bad data.

5. Conclusion

With the help of the FlipKart data warehouse project, raw e-commerce data was turned into an advanced analytical tool that could meet a wide range of business intelligence needs. The project set up a strong base for making decisions based on data by following a methodical approach for data collection, normalization, ETL processing, dimensional modeling, and analytical queries.

The star schema form of the dimensional model made complex queries easier to understand while keeping data integrity and the ability to track changes over time. A number of business-focused queries showed how analytically powerful the implemented solution was. They gave useful information about sales performance, customer behavior, geographic patterns, and relationships across multiple dimensions.

Even though the current implementation meets all the needs of the project, the suggested future enhancements would make it even more valuable to the business by improving customer segmentation, allowing real-time integration, adding more product and geographic dimensions, incorporating advanced analytics, and automating monitoring of data quality.

It's not just a technical accomplishment that this data warehouse is, it's also a strategic business tool that can help you make better decisions and gain a competitive edge through data-driven insights.