17/11/2025

BSAN7206: Assessment 2

DATA WAREHOUSING IMPLEMENTATION

Data analysis of Handlebar Haven with strategic insights and recommendations for growth

MADELINE COERT
YING HONG
RACHEL LOW
GEORGE ONISHI
CALEB WONG

WORD COUNT: 3,490

Executive Summary

Handlebar Haven is a successful global business in bicycle manufacturing and retailing. To lead the business to the next level, the CEO needs to know which area to focus its business expansion. However, the CEO and the management do not have the relevant facts and figures to support the decisions. Specifically, it needs to analyse sales volumes and profitability in these key areas: customer, product, sales territories, time period and product categories.

Analysis of this data revealed several key performance drivers. The most profitable customers are primarily located in the United States, with San Francisco consistently appearing as a top-performing area. The Classic Vest (Medium) and the Bike Wash Dissolver emerged as the highest-margin products, while the Northwest region of the United States generated the strongest territorial profitability and sales turnover. Across all years, sales peaked on Sundays and during the first quarter, with 2024 delivering the highest total profit. The analysis also identified product lines that consistently underperform, including Scooter Racks, Caps, Forks, Derailleurs and Headsets. These sub-categories generate low returns and show limited growth potential, making them suitable candidates for product rationalisation.

This report discusses in detail the business analytics methodologies utilised, data extraction and transformation tasks performed, and the business insights gathered from the exploratory data analysis.

Table of Contents

1.1. Business Analytics Framework

Handlebar Haven (HH) operates world-wide with presence in 3 continents. Vast amounts of data are captured daily by its transactional systems. To derive strategic business insights from the data collected, a systematic approach to understand the data is needed and hence business analytics (BA) is crucial. BA plays a key role within organisations by transforming complex and fragmented data into actionable insights that drive strategic decision-making. It bridges the gap between raw information and meaningful business outcomes, allowing companies to base their strategies on evidence rather than intuition. A strong analytics capability enhances competitiveness, operational efficiency and long-term planning.

The BA framework provides the architecture for this transformation through:

1. Source data systems **-** supply raw data for analysis, internal or external
2. Data staging area - temporary storage for extracted data; data is cleansed, transformed, arranged in relational model
3. Data warehouse and metadata - Central repository for transformed data and data dictionary
4. Analytics and Reporting **-** Data is presented in user friendly formats i.e. reports, dashboards, tables, charts using data visualisation tools. Where users can perform further analysis on their own.

These components support the continuous flow of data and ensure insights are both timely and reliable, reinforcing the organisation's ability to make data-driven decisions. Refer to Figure 3.1 for the key components of HH's BA framework.

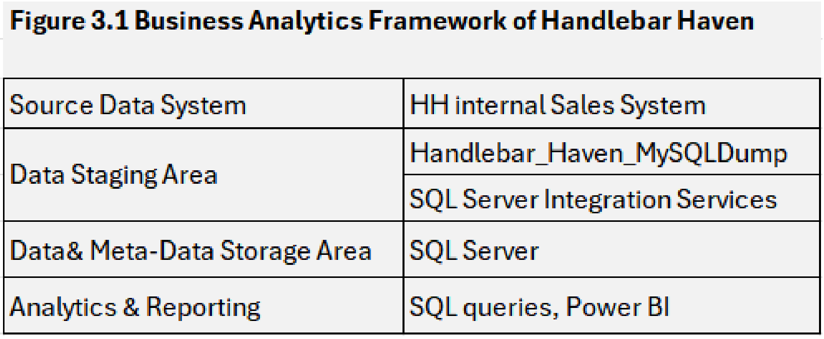


Figure 3.1 Business Analytics Framework of Handlebar Haven

Source Data System	HH internal Sales System
Data Staging Area	Handlebar_Haven_MySQLDump
	SQL Server Integration Services
Data& Meta-Data Storage Area	SQL Server
Analytics & Reporting	SQL queries, Power BI

An essential part of this process is the implementation of a dimensional model, by demoralising fragmented tables stored in operational systems and making them more user friendly for understandable stakeholders. It is business process specific and it captures quantitative(fact) and descriptive(dimensions) data at its lowest granular level. By organising data into fact and dimension tables, this model enables faster query performance and data slicing, supports historical trend analysis and enhances data consistency. Its usability is a key advantage as it allows business users to explore data intuitively across dimensions such as time, product and region (Kimball & Ross, 2013). Complementing this is structured data integration, which merges multiple data sources into a unified warehouse through ETL (Extract, Transform, Load) processes. This integration improves data accessibility, quality and reliability by cleaning and standardising inputs before analysis, ensuring that decision-makers are working from one "single source of truth".

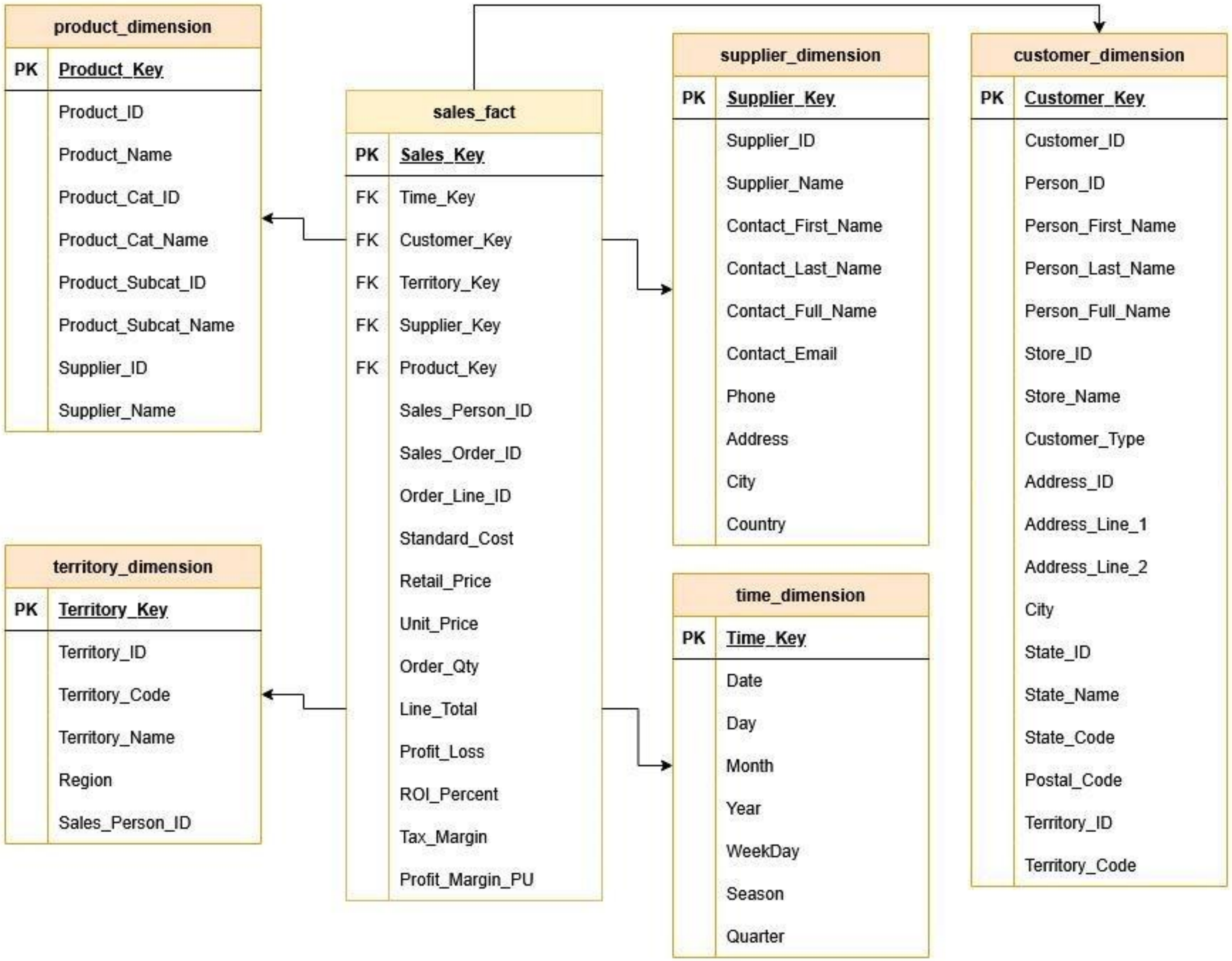
Business analytics also plays a vital role in addressing data silos and fostering evidence-based decision-making. In many organisations, valuable information is trapped within isolated systems across departments. BA overcomes this challenge through integrated data infrastructures that combine different systems into a centralised platform. This integration standardises metrics and definitions of business terms, creating consistency across the organisation and enabling cross-functional collaboration. Dashboards and visualisations further enhance communication by making analytical outputs accessible and comprehensible to a wide range of stakeholders. By using descriptive, predictive and prescriptive analytics, BA helps organisations understand what has happened, why it occurred and what actions should follow. This structured, data-driven approach replaces intuition-based decision-making with clear, empirical evidence that supports effective policy and strategy development.

HH is entering a critical stage of expansion and needs transparent, accurate insights to guide its growth strategy. By implementing an integrated analytics framework, the company can identify which customers, products, territories and timeframes drive the most profitability. This evidence-based system will allow the leadership team to allocate resources strategically, refine marketing efforts and make confident, data-backed decisions that support sustainable business growth.

1.2. Design of a Dimensional Model:

The dimensional model for HH follows a Kimball-style star schema centred on the Fact_Sales table, surrounded by five key dimensions: Product, Customer, Territory, Time and Supplier. The updated schema includes a new Season attribute added above Quarter in the Time hierarchy, allowing for seasonal trend analysis in addition to existing day, month, quarter and year groupings. See the next page for the final implemented star schema.

1.2.1. Final Implemented Star Schema



product_dimension	
PK	Product_Key
	Product_ID
	Product_Name
	Product_Cat_ID
	Product_Cat_Name
	Product_Subcat_ID
	Product_Subcat_Name
	Supplier_ID
	Supplier_Name

territory_dimension	
PK	Territory_Key
	Territory_ID
	Territory_Code
	Territory_Name
	Region
	Sales_Person_ID

sales_fact	
PK	Sales_Key
FK	Time_Key
FK	Customer_Key
FK	Territory_Key
FK	Supplier_Key
FK	Product_Key
	Sales_Person_ID
	Sales_Order_ID
	Order_Line_ID
	Standard_Cost
	Retail_Price
	Unit_Price
	Order_Qty
	Line_Total
	Profit_Loss
	ROI_Percent
	Tax_Margin
	Profit_Margin_PU

supplier_dimension	
PK	Supplier_Key
	Supplier_ID
	Supplier_Name
	Contact_First_Name
	Contact_Last_Name
	Contact_Full_Name
	Contact_Email
	Phone
	Address
	City
	Country

time_dimension	
PK	Time_Key
	Date
	Day
	Month
	Year
	WeekDay
	Season
	Quarter

customer_dimension	
PK	Customer_Key
	Customer_ID
	Person_ID
	Person_First_Name
	Person_Last_Name
	Person_Full_Name
	Store_ID
	Store_Name
	Customer_Type
	Address_ID
	Address_Line_1
	Address_Line_2
	City
	State_ID
	State_Name
	State_Code
	Postal_Code
	Territory_ID
	Territory_Code

1.2.2. Rationale for Selecting the Model

The group selected this model because it provides a strong balance of accuracy, completeness, and analytical flexibility. The fact table captures sales at the most granular level (individual order line items) supporting precise profitability and performance analysis. Conformed dimensions, particularly Product, Time and Territory, allow for consistent reporting across business processes. The structure supports intuitive drill-down and roll-up across product, customer, geographic and temporal hierarchies. The addition of the Season attribute enhances temporal analysis by enabling clearer identification of cyclical patterns. Its simplicity and extensibility make the model suitable for future analytical needs such as marketing performance or inventory optimisation.

1.2.3. Modifications

This updated schema builds on the initial assignment design with several enhancements aimed at improving analytical depth and business relevance. The most significant change is the addition of the Season attribute to the Time dimension, positioned above Quarter to support seasonal trend analysis and improve forecasting accuracy. The Product dimension has been refined to include both category and subcategory identifiers and names, creating clearer hierarchies for product-level performance reporting. The Customer dimension was expanded to capture customer type, store information, and a more detailed address structure including state, postal code, and territory identifiers, enabling granular segmentation by location and customer profile. Similarly, the Supplier dimension was enriched with contact details such as names, email addresses and phone numbers to facilitate supplier relationship management and communication analysis. Finally, the Territory dimension now includes region, territory code, and sales person identifiers, strengthening geographic and sales team performance tracking. Together, these enhancements enhance the model's descriptive power, analytical flexibility and alignment with HH's operational and strategic decision-making needs.

1.2.4. Discussion

Grain of Each Fact Table:
The Fact_Sales table is designed at the transaction level, with one row per order line item. This level of granularity enables detailed profitability analysis at the SKU level and allows executives to aggregate results by customer, territory, or time period without losing accuracy.

Included and Conformed Dimensions:
Five core dimensions include Product, Customer, Territory, Time, and Supplier support all analytical perspectives. Product and Time are conformed dimensions, ensuring consistency between sales and purchasing analysis. Each dimension supports natural hierarchies (e.g., Product → Subcategory → Category; Day → Month → Quarter → Season → Year) that enable drill-down and roll-up capabilities within BI tools.

Key Business Processes Supported:
The schema supports HH's key analytical needs, including sales performance tracking,

supplier and customer profitability analysis and regional sales monitoring. The enhanced Season attribute enables identification of peak and off-peak demand periods**,** improving forecasting, stock management and marketing strategy.

1.3. Data Integration and ETL Process

We used SSIS (SQL Server Integration Services) to integrate data from HH's transactional sales system into a data warehouse based on the proposed dimensional model. The objective of the data warehouse is to enable analytical querying of the data to produce high level, strategic data insights that answer the client's key business questions. Therefore the data warehouse will contain both current and historical data that can be easily accessed and queried according to the business requirements.

1.3.1. Overview of the ETL Process

ETL consists of three steps:

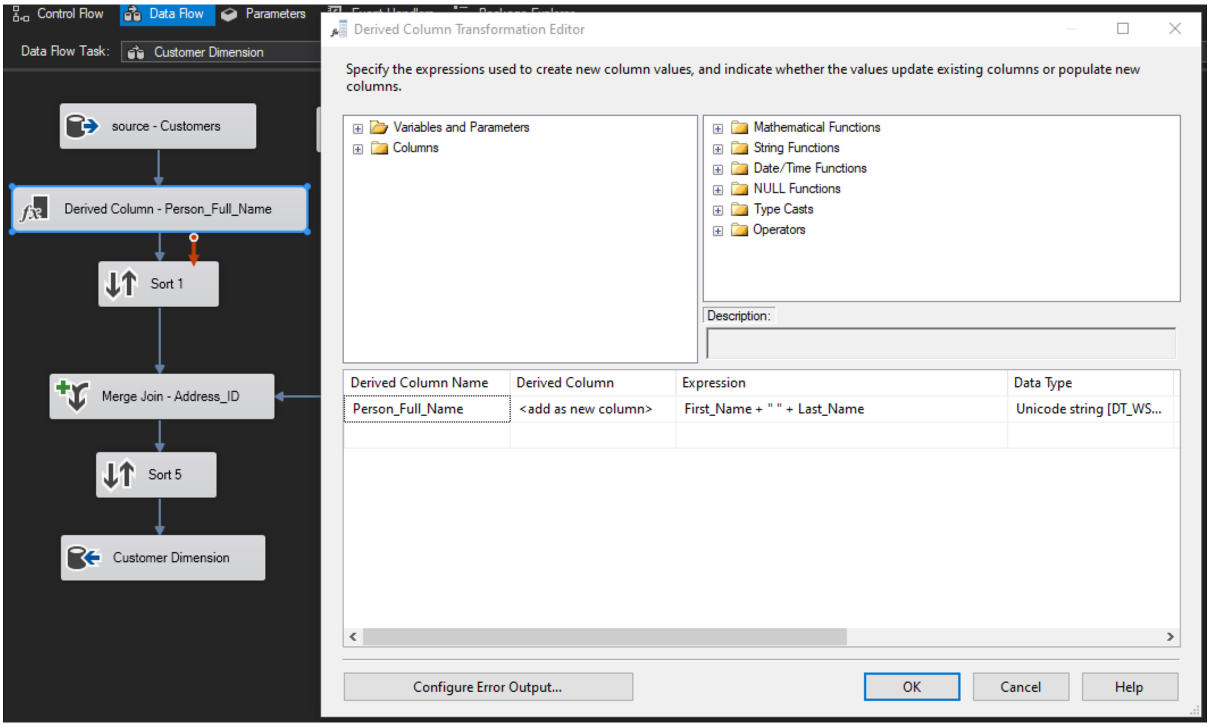
1. **Extraction**: Data from a source system (HH's sales system) are drawn out for transformation
2. **Transformation**: Extracted data are cleansed and transformed. Typical transformations include derived values, removal of duplicated records, standardisation of data formatting, etc. to comply with constraints and requirements set by the data warehouse. Data attributes are also mapped to their corresponding attributes within the data warehouse in this step.
3. **Loading**: Transformed data are integrated into the data warehouse according to the parameters set in the transformation step, with dimensional tables being populated first, followed by fact tables. Once loaded, the new database is ready for analytical querying.

In this scenario:

- **Extraction**: HH has provided its operational database as a MySQL Server dump for access via MySQL Workbench.
- **Transformation** / **Loading**: Visual Studio with an SSIS plugin will be used to design the data warehouse, apply transformations to ensure compliance with warehouse requirements, and then to load the data into the data warehouse.

1.3.2. SSIS Transformations

CustomerDimension

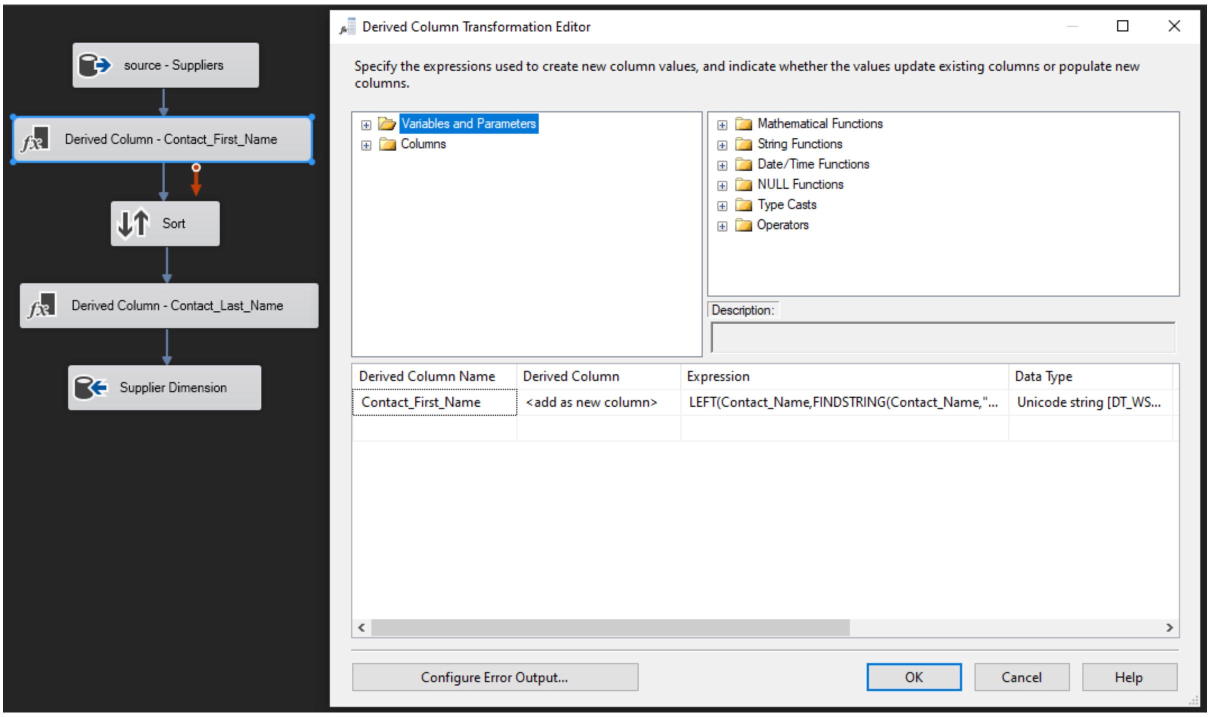


Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

Derived Column Name	Derived Column	Expression	Data Type
Person_Full_Name	<add as new column>	First_Name + " " + Last_Name	Unicode string [DT_WS...

Component Name	Component Type	Transformation	Notes
Person_Full_Name	Derived column	First_Name + " " + Last_Name	• Column added to distinguish customer individuals from other non-customer contacts • Warehouse convention to ensure all individuals have First names, Last names and Full names all captured separately

SupplierDimension

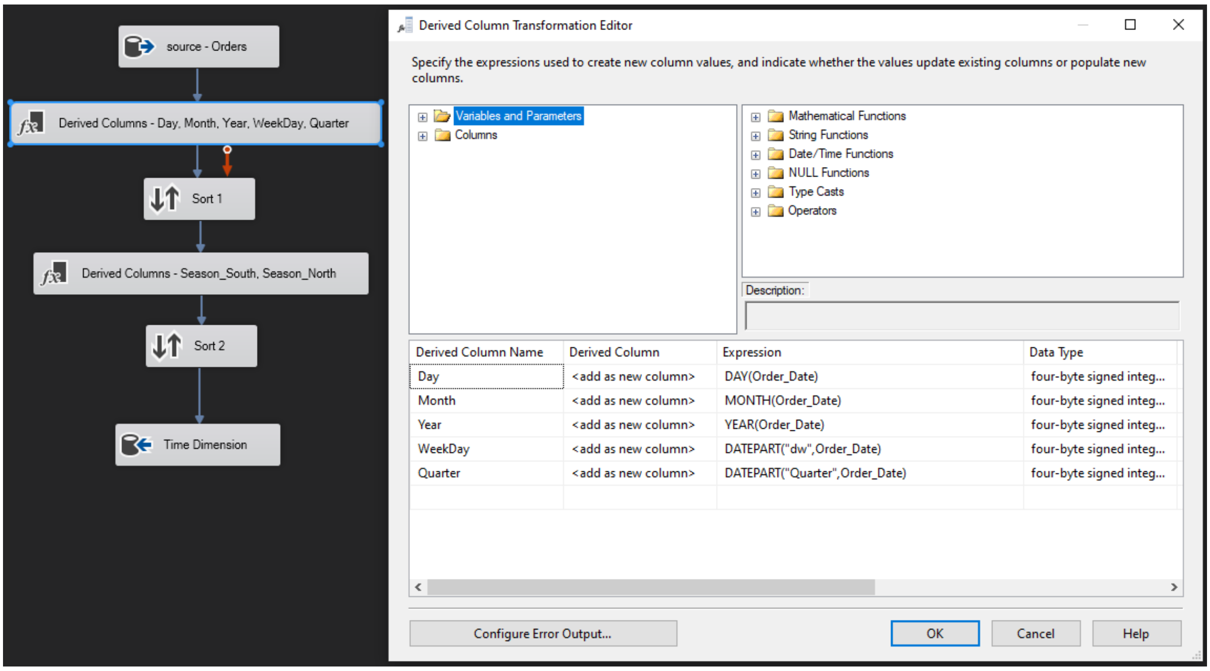


Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

Derived Column Name	Derived Column	Expression	Data Type
Contact_First_Name	<add as new column>	LEFT(Contact_Name,FINDSTRING(Contact_Name,"...	Unicode string [DT_WS...

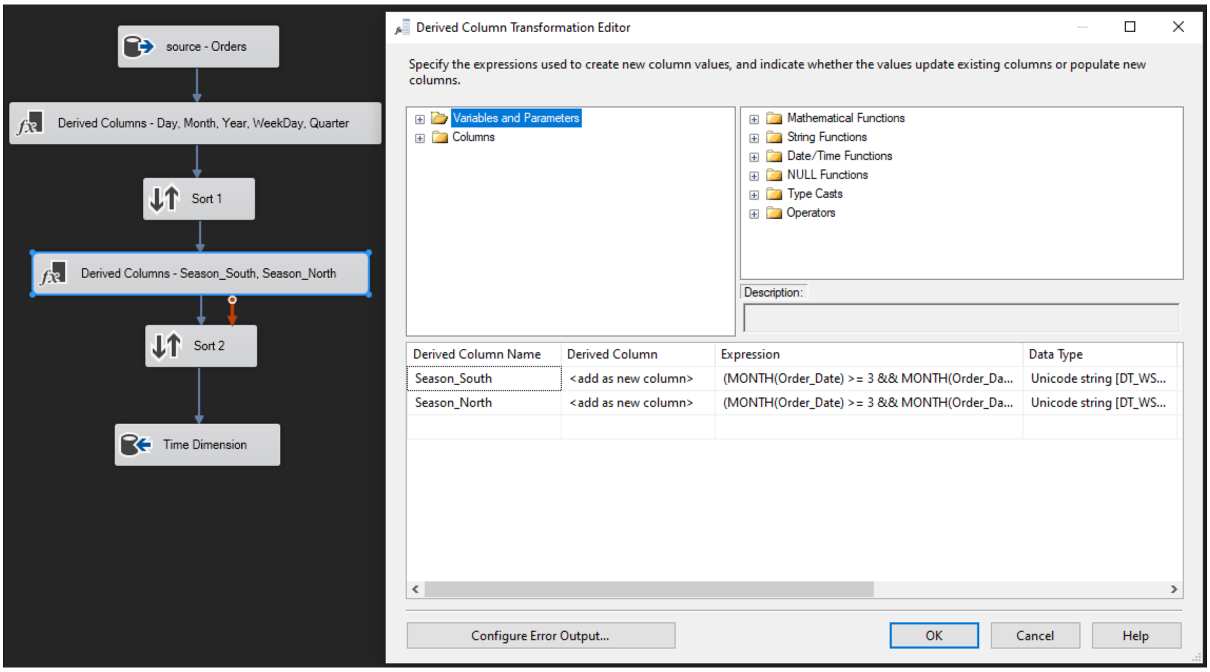
Component Name	**Component Type**	**Transformation**	**Notes**
Contact_First_Name	Derived column	LEFT(Contact_Name,FINDSTRING (Contact_Name," ",1) - 1)	• Columns added to distinguish supplier contacts from other individuals represented, eg Customers, Sales staff. • Warehouse convention to ensure all individuals have First, Last and Full names all captured separately. • Transform provided Full name values into separate First and Last names.
Contact_Last_Name	Derived column	RIGHT(Contact_Name,LEN (Contact_Name) - FINDSTRING(Contact_Name," ",1))	

TimeDimension



Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

Derived Column Name	Derived Column	Expression	Data Type
Day	<add as new column>	DAY(Order_Date)	four-byte signed integ...
Month	<add as new column>	MONTH(Order_Date)	four-byte signed integ...
Year	<add as new column>	YEAR(Order_Date)	four-byte signed integ...
WeekDay	<add as new column>	DATEPART("dw",Order_Date)	four-byte signed integ...
Quarter	<add as new column>	DATEPART("Quarter",Order_Date)	four-byte signed integ...

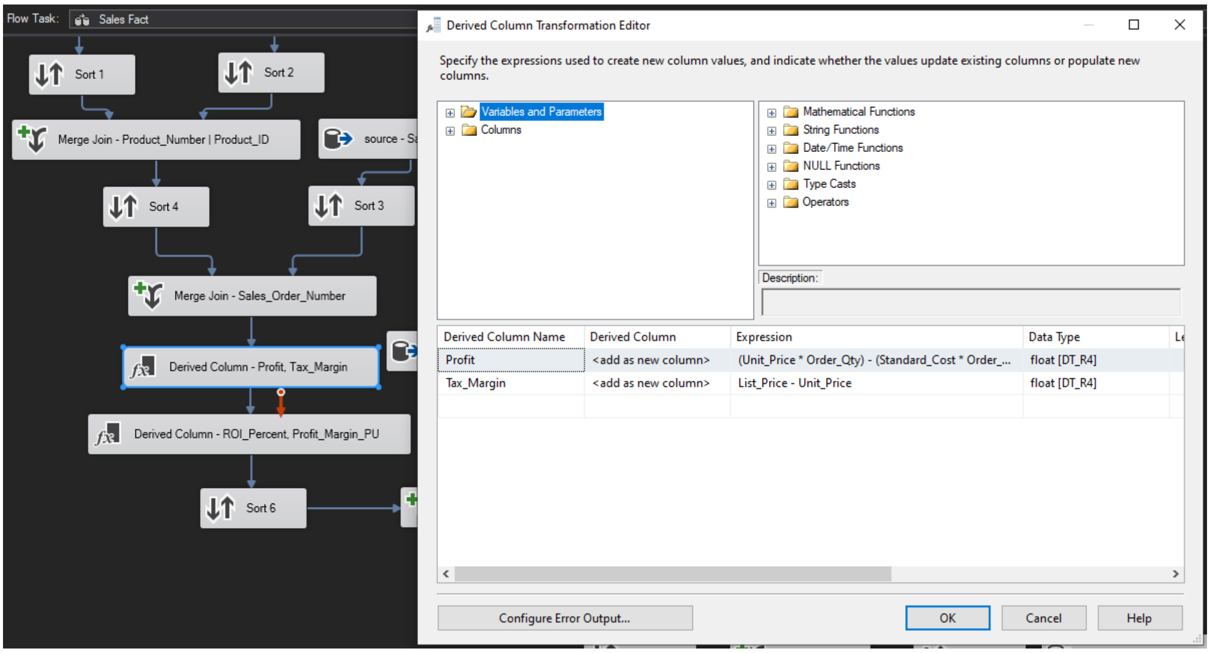


Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

Derived Column Name	Derived Column	Expression	Data Type
Season_South	<add as new column>	(MONTH(Order_Date) >= 3 && MONTH(Order_Da...	Unicode string [DT_WS...
Season_North	<add as new column>	(MONTH(Order_Date) >= 3 && MONTH(Order_Da...	Unicode string [DT_WS...

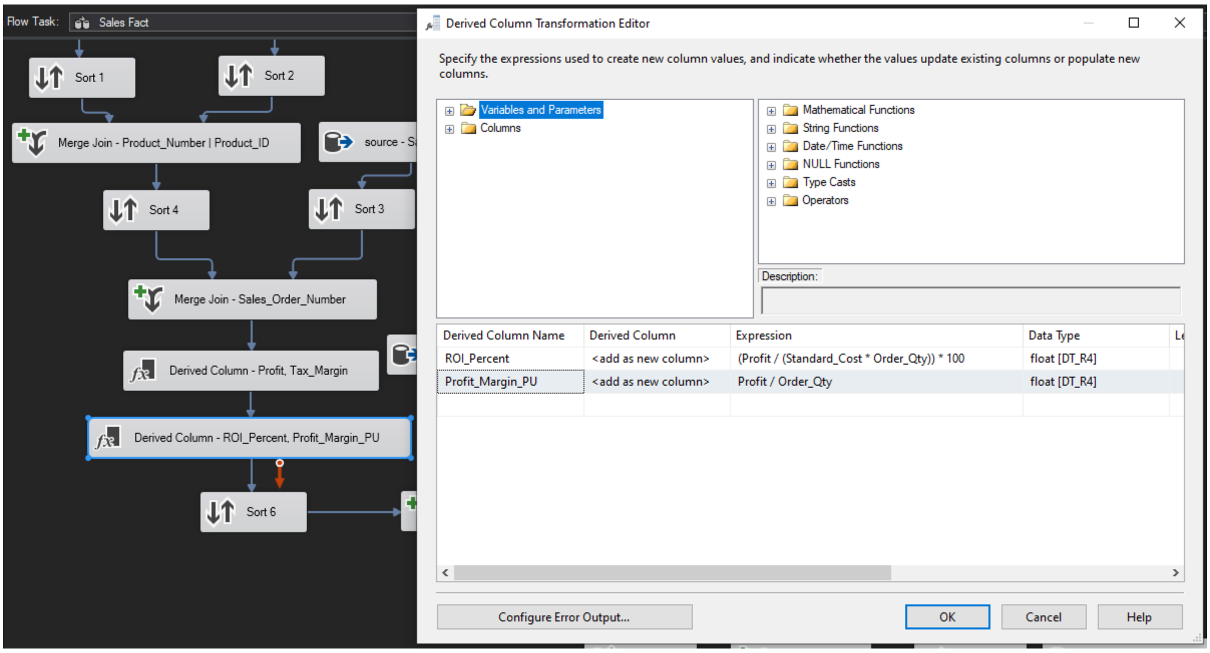
Component Name	**Component Type**	**Transformation**	**Notes**
Day	Derived column	DAY(Order_Date)	• Derives values based on Order_Date value provided from Sales_Order source table. • Separates Sales_Order into component values that can be used to better inform analytics.
Month	Derived column	MONTH(Order_Date)	
Year	Derived column	YEAR(Order_Date)	
WeekDay	Derived column	DATEPART("dw",Order_Date)	
Quarter	Derived column	DATEPART("Quarter",Order_Date)	
Season_North	Derived column	(MONTH(Order_Date) >= 3 && MONTH(Order_Date) <= 5) ? "Spring" : (MONTH(Order_Date) >= 6 && MONTH(Order_Date) <= 8) ? "Summer" : (MONTH(Order_Date) >= 9 && MONTH(Order_Date) <= 11) ? "Autumn" : "Winter"	• Seasons derived based on [Month], with separate attributes based on northern or southern hemisphere seasons. • Enables season-based analyses without adding geographic attributes to TimeDimension.
Season_South	Derived column	(MONTH(Order_Date) >= 3 && MONTH(Order_Date) <= 5) ? "Autumn" : (MONTH(Order_Date) >= 6 && MONTH(Order_Date) <= 8) ? "Winter" : (MONTH(Order_Date) >= 9 && MONTH(Order_Date) <= 11) ? "Spring" : "Summer"	

SalesFact


Flow Task:
Sales Fact
Sort 1
Sort 2
Merge Join - Product_Number | Product_ID
source - S
Sort 4
Sort 3
Merge Join - Sales_Order_Number
Derived Column - Profit, Tax_Margin
Derived Column - ROI_Percent, Profit_Margin_PU
Sort 6
Derived Column Transformation Editor
Variables and Parameters
Columns
Mathematical Functions
String Functions
Date/Time Functions
NULL Functions
Type Casts
Operators
Description:
Configure Error Output...
OK
Cancel
Help

Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

Derived Column Name	Derived Column	Expression	Data Type
Profit	<add as new column>	(Unit_Price * Order_Qty) - (Standard_Cost * Order_...	float [DT_R4]
Tax_Margin	<add as new column>	List_Price - Unit_Price	float [DT_R4]


Flow Task:
Sales Fact
Sort 1
Sort 2
Merge Join - Product_Number | Product_ID
source - S
Sort 4
Sort 3
Merge Join - Sales_Order_Number
Derived Column - Profit, Tax_Margin
Derived Column - ROI_Percent, Profit_Margin_PU
Sort 6
Derived Column Transformation Editor
Variables and Parameters
Columns
Mathematical Functions
String Functions
Date/Time Functions
NULL Functions
Type Casts
Operators
Description:
Configure Error Output...
OK
Cancel
Help

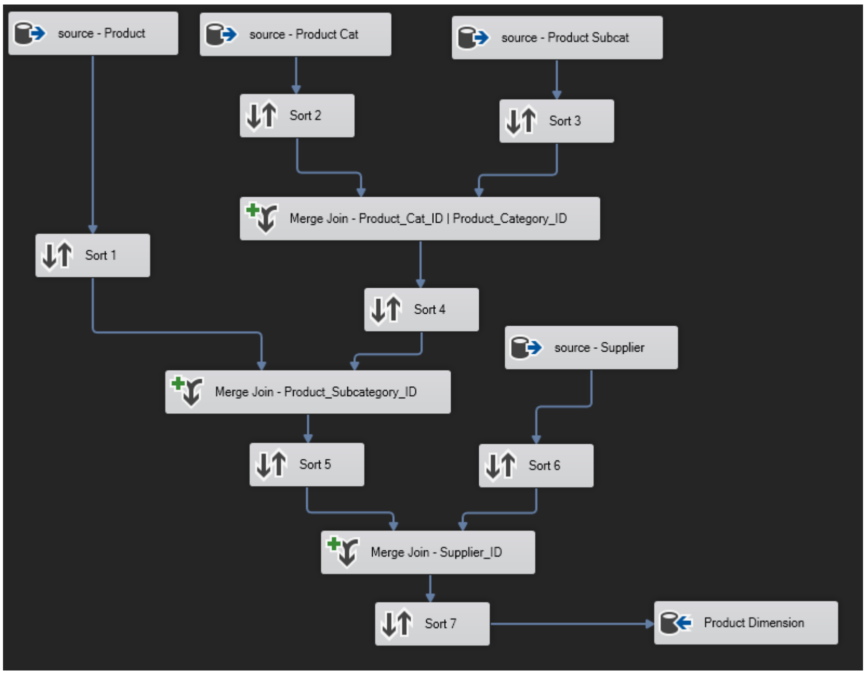
Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

Derived Column Name	Derived Column	Expression	Data Type
ROI_Percent	<add as new column>	(Profit / (Standard_Cost * Order_Qty)) * 100	float [DT_R4]
Profit_Margin_PU	<add as new column>	Profit / Order_Qty	float [DT_R4]

Component Name	Component Type	Transformation	Notes
Profit_Loss	Derived column	(Unit_Price * Order_Qty) - (Standard_Cost * Order_Qty)	• Total profit or loss for the sales line
Tax_Margin	Derived column	List_Price - Unit_Price	• Revenue gained from tax on products sold
ROI_Percent	Derived column	(Profit / (Standard_Cost * Order_Qty)) * 100	• Profit/Loss percentage by product sale
Profit_Margin_PU	Derived column	Profit / Order_Qty	• Profit/Loss margin per unit sold

1.3.3. Data Sourcing and Mapping

ProductDimension



`ProductDimension` sources data from `products`, `product_categories`, `product_sub_category` and `suppliers` tables from the HH database.

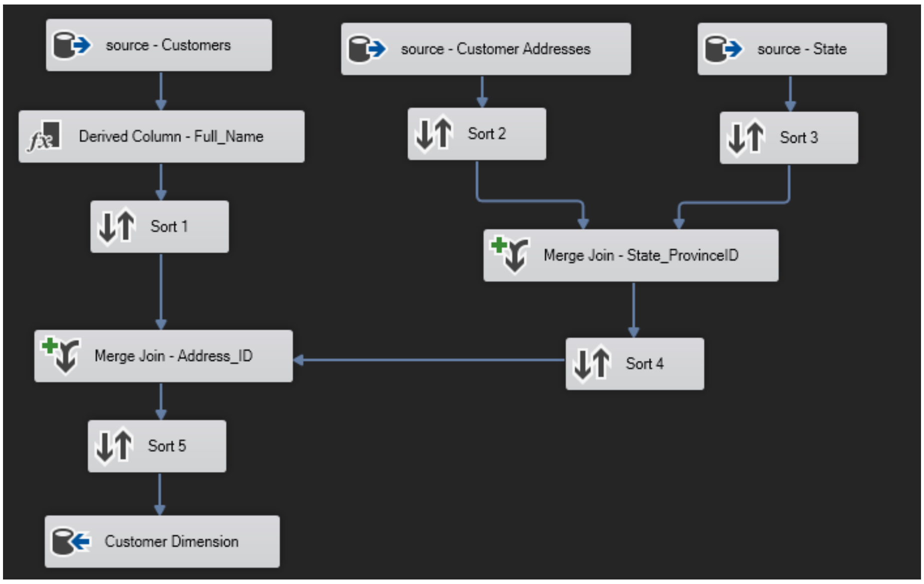
Data flow:

1. Source attributes from `product_categories` and `product_sub_category` tables, sort based on *`Product_Cat_ID`* and *`Product_Category_ID`* columns;
2. Merge join `product_categories` and `product_sub_category` data extracts via Left Outer Join and *`Product_Cat_ID`* and *`Product_Category_ID`* as Join Keys;
3. Source attributes from products table, sort via *`Product_Subcategory_ID`*;
4. Merge Join products extract with `product_categories+product_sub_category` combined extract via Left Outer Join with *`Product_Subcategory_ID`* as Join Key. Sort resulting data via *`Supplier_ID`*;
5. Source attributes from suppliers table, sort via *`Supplier_ID`*;
6. Merge Join `products+product_categories+product_sub_category` combined extract with suppliers extract via Inner Join using *`Supplier_ID`* as Join Key. Reorder using *`Product_ID`* values;
7. Create `ProductDimension` table in MySQL Server via script (see appendix);
8. Load data into `ProductDimension` table.

Mapping:

Product Dimension Attribute	**Source Database Attribute**	**Source Database Table**	**Notes**
Product_Key			Surrogate Primary Key
Product_ID	Product_ID	product	
Product_Name	Product_Name	product	
Product_Cat_ID	Product_Cat_ID	product_categories	
Product_Cat_Name	Cat_Name	product_categories	Attribute renamed to match naming convention
Product_Subcat_ID	Product_Subcategory_ID	product_sub_category	Attribute name shortened to conform with naming convention
Product_Subcat_Name	Product_Name	product_sub_category	Attribute renamed during ETL to better represent data and conform with naming convention
Supplier_ID	Supplier_ID	suppliers	
Supplier_Name	Supplier_Name	suppliers	

CustomerDimension



`CustomerDimension` sources data from the `hh.customers`, `hh.customer_address` and `hh.state` tables.

Data flow:

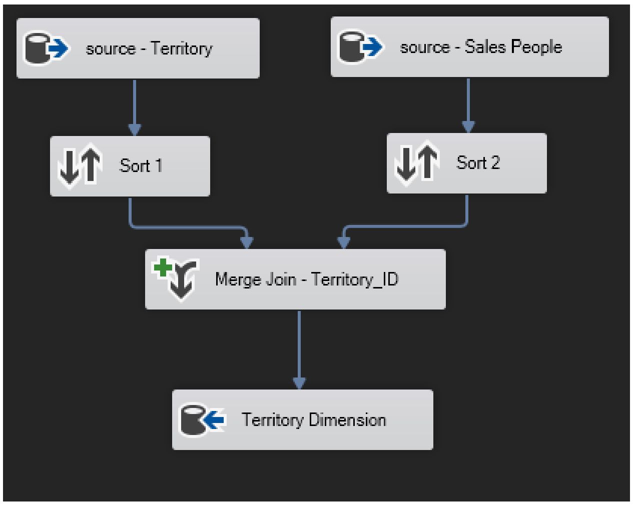
1. Source attributes from `customer_address` and `state` tables, sort data *`State_ProvinceID`* column.

2. Merge Join `customer_address` extract with state extract via Left Outer Join with `State_ProvinceID` as Join Key. Sort resulting data via `Address_ID`.
3. Source attributes from customer table.
4. Create `Custoner_Full_Name` column from customer extract by combining `First_Name` with `Last_Name`, sort data via `Address_ID` column.
5. Merge Join customers extract with `customer_address+state` combined extract via Inner Join using `Address_ID` as Join Key. Reorder using `Customer_ID` values.
6. Create `CustomerDimension` table in MySQL Server via script (see appendix)
7. Load data into `CustomerDimension` table.

Mapping:

CustomerDimension Attribute	Source Database Attribute	Source Database Table	Notes
Customer_Key			Surrogate Primary Key
Customer_ID	Customer_ID	customers	
Person_ID	Person_ID	customers	
Person_First_Name	First_Name	customers	
Person_Last_Name	Last_Name	customers	
Person_Full_Name			Derived column: First_Name + " " + Last_Name
Store_ID	Store_ID	customers	
Store_Name	Store_Name	customers	
Customer_Type	Person_Type_Description	customers	
Address_ID	Address_ID	customer_address	
Address_Line_1	Address_Line_1	customer_address	
Address_Line_2	Address_Line_2	customer_address	
City	Client_City	customer_address	
State_ID	State_Province_ID	customer_address	
State_Name	State_Name	state	
State_Code	State_Province_Code	state	
Postal_Code	Postal_Code	customer_address	
Territory_ID	Territory_ID	state	
Territory _Code	Country_Region_Code	state	

TerritoryDimension



TerritoryDimension sources data from the `hh.sales_territory` and `hh.sales_persons` tables.

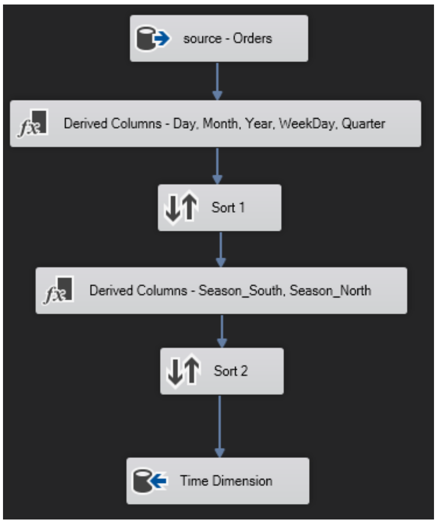
Data flow:

1. Source attributes from `state_territory` and `sales_person`, sort data by *`Territory_ID`*.
2. Merge Join `state_territory`, extract with `sales_person`, extract via Left Outer Join with *`Territory_ID`* as Join Key.
3. Sort data via *`Country_Name`* and *`Territory_ID`*.
4. Load data into `TerritoryDimension` table.

Mapping:

TerritoryDimension Attribute	Source DB Attribute	Source DB Table	Notes
Territory_Key			Surrogate Primary Key
Territory_ID	Territory_ID	sales_territory	
Territory_ Code	Country_Region_Code	state	
Territory_Name	Territory_Name	sales_territory	
Region	Country_Group	sales_territory	
Sales_Person_ID	Sales_Person_ID	sales_persons	

TimeDimension



source - Orders
Derived Columns - Day, Month, Year, WeekDay, Quarter
Sort 1
Derived Columns - Season_South, Season_North
Sort 2
Time Dimension

`TimeDimension` sources data from `hh.sales_orders`.

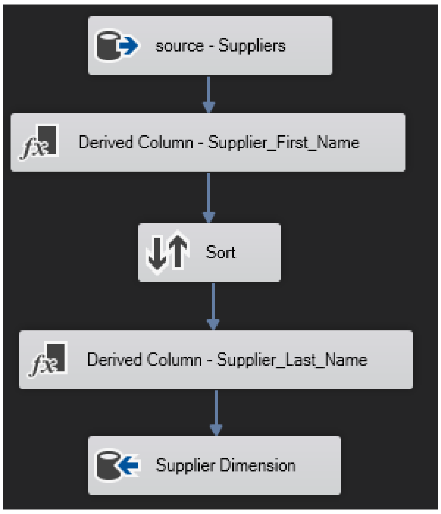
Data flow:

1. Load *`Order_Date`* from `Sales_Order` table.
2. Create Derived Columns to represent *`Day`*, *`Month`*, *`Year`*, *`WeekDay`* and *`Quarter`* using values loaded from *`Order_Date`* column. Sort values based on *`Order_Date`* column.
3. Create additional Derived Columns for *`Season_North`* and *`Season_South`* values.
4. Sort data via *`Order_Date`* column, remove any rows with duplicate values.
5. Load data into `TimeDimension` table.

Mapping:

Time Dimension Attribute	**Source Database Attribute**	**Source Database Table**	**Notes**
Time_Key			Surrogate Primary Key
Date	Order_Date	sales_orders	
Day			Derived column
Month			Derived column
Year			Derived column
Weekday			Derived column
Season_North			Derived column
Season_South			Derived column
Quarter			Derived column

SupplierDimension


source - Suppliers
Derived Column - Supplier_First_Name
Sort
Derived Column - Supplier_Last_Name
Supplier Dimension

`SupplierDimension` sources data from `hh.suppliers`.

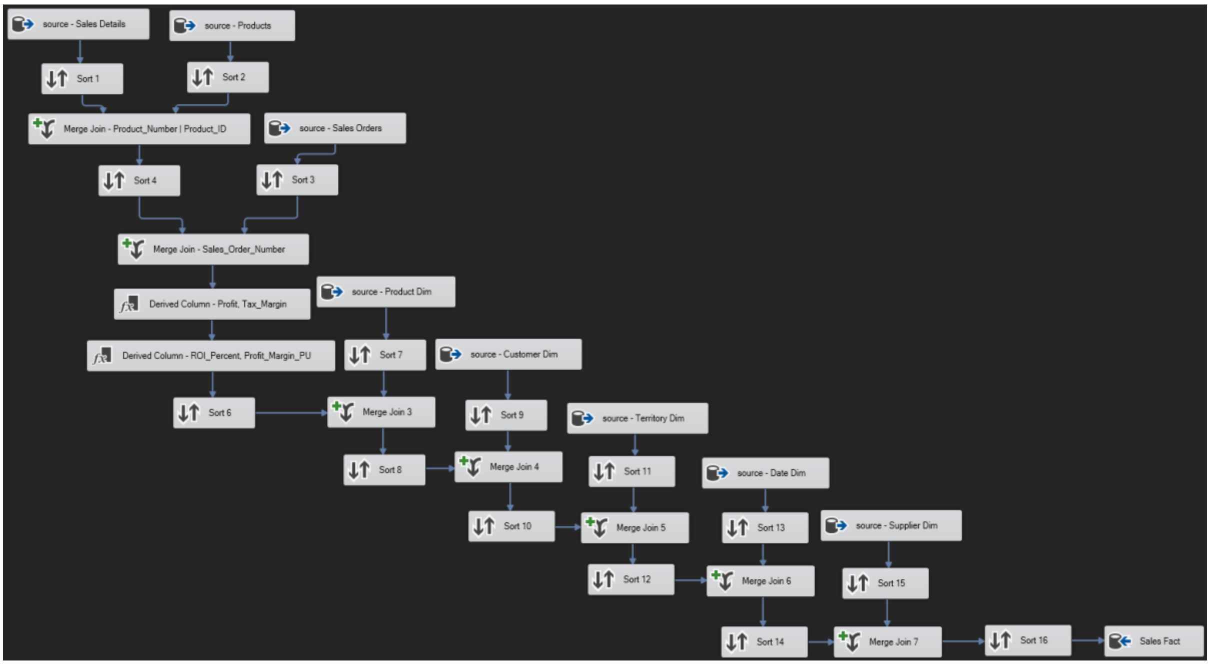
Data flow:

1. Load all data from `suppliers` table.
2. Create Derived Columns for *`Supplier_First_Name`* and *`Supplier_Last_Name`* using values from *`Contact_Name`*.
3. Sort values via *`Supplier_ID`* column.
4. Load data into `SupplierDimension` table.

Mapping:

Supplier Dimension Attribute	**Source Database Attribute**	**Source Database Table**	**Notes**
Supplier_Key			Surrogate Primary Key
Supplier_ID	Supplier_ID	suppliers	
Supplier_Name	Supplier_Name	suppliers	
Contact_First_Name			Derived column
Contact_Last_Name			Derived column
Contact_Full_Name	Contact_Name	suppliers	
Contact_Email	Contact_Email	suppliers	
Phone	Phone	suppliers	
Address	Address	suppliers	
City	City	suppliers	
Country	Country	suppliers	

SalesFact



`SalesFact` sources data from `hh.sales_orders`, `hh.sales_order_details` and `hh.product.`

Data Flow:

1. Source attributes from `sales_order_detail` and `product` tables, sort data by *`Product_Number`* / *`Product_ID`* columns.
2. Merge Join `sales_order_detail` table extract with `product` table extract via Left Outer Join with *`Product_Number`* / *`Product`* as Join Key. Sort data via *`Sales_Order_Number`* column.
3. Source attributes from `sales_orders` table, sort via *`Sales_Order_Number`* column.
4. Merge Join `sales_order_details`+`product` combined extract with `sales_orders` extract via Inner Join using *`Sales_Order_Number`* as Join Key.
5. Create *`Profit_Loss`*, *`ROI_Percent`*, *`Tax_Margin`* and *`Profit_Margin_PU`* columns using derived columns.
6. Use Merge Joins via Left Outer Joins to replace *`Product_ID`*, *`Customer_ID`*, *`Territory_ID`*, *`Order_Date`* and *`Supplier_ID`* values with the *`Product_Key`*, *`Customer_Key`*, *`Territory_Key`*, *`Supplier_Key`* and *`Time_Key`* from respective dimensions. Note Merge Join functions were used to imitate Lookup function which does not function with ODBC connection to MySQL databases.
7. Sort data via *`Sales_Order_Number`* column.
8. Load data into `SalesFact` table.

Mapping:

Sales Fact Attribute	Source Database Attribute	Source Database Table	Notes
Sales_Key			Surrogate Primary Key
Time_Key			Foreign Key
Customer_Key			Foreign Key
Territory_Key			Foreign Key
Supplier_Key			Foreign Key
Product_Key			Foreign Key
Sales_Person_ID	Sales_Person_ID	sales_orders	
Sales_Order_ID	Sales_Order_Number	sales_order_details	
Order_Line_ID	Order_Line_ID	sales_order_details	
Standard_Cost	Standard_Cost	products	
Retail_Price	List_Price	products	
Unit_Price	Unit_Price	sales_order_details	
Order_Qty	Order_Qty	sales_order_details	
Line_Total	Line_Total	sales_order_details	
Profit_Loss			Derived column
ROI_Percent			Derived column
Tax_Margin			Derived column
Profit_Margin_PU			Derived column

1.3.4. Metric Calculations

The SalesFact attributes `Profit_Loss`, `Tax_Margin`, `ROI_Percent`, `Profit_Margin_PU` are derived from `Unit_Price`, `Standard_Cost` and `List_Price` values contained in the `product` and `sales_order_details` tables from HH's source database. These transformations were developed during the dimensional model design phase and then implemented at the 'Transform' step of the ETL process (refer to section 1.3.1).

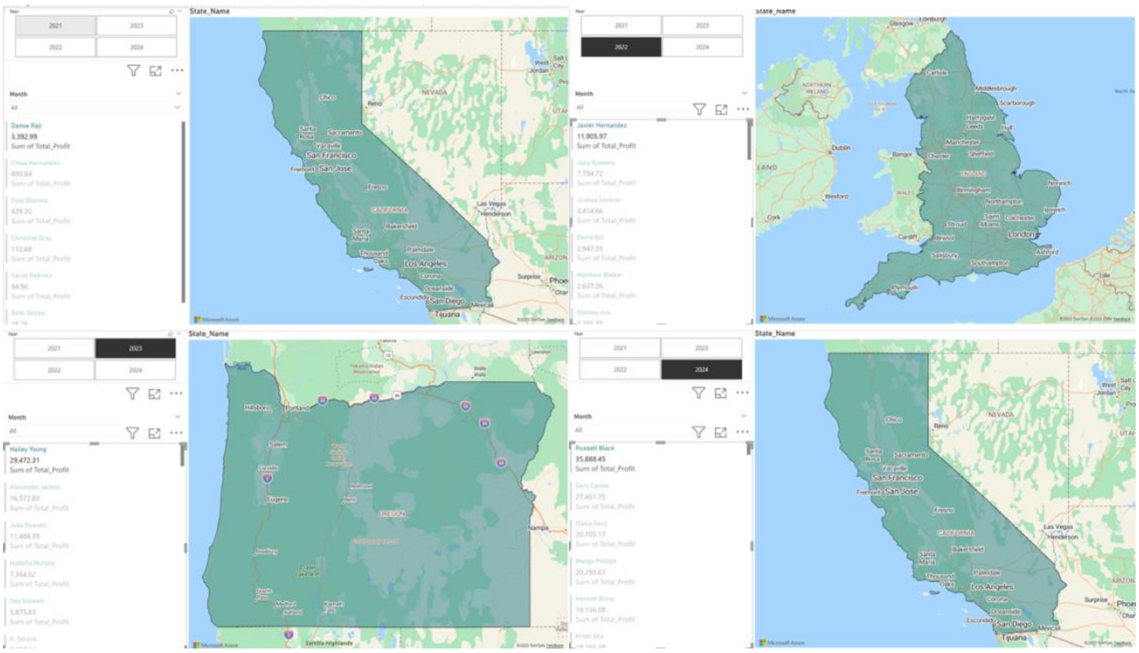
HH's business questions all relate to determining profitability in conjunction with other dimensions, such as Seasons, Territories, and Product Categories. The data from HH's operational database provides costs and prices associated with products, but did not include any calculations to show how each would relate to the other. Therefore, the attributes we added into our data warehouse in SalesFact serve as key performance metrics that enable detailed analytical querying that provide insights that respond to HH's business questions, covered in the next section.

1.4. Business Insights and Recommendations

1.4.1. Key Customers

The query used was to retrieve summarised sales performance for each customer across specific time periods. It joins the sales *FactSales* with the time and *CustomerDimension* tables. For every customer and each year, quarter and month, it calculates total units sold, total sales value and total profit by summing the corresponding fields in the *FactSales* table. The results are grouped by customer information and time attributes to ensure accurate aggregation, and then ordered by profit in descending order so that the most profitable periods appear first.

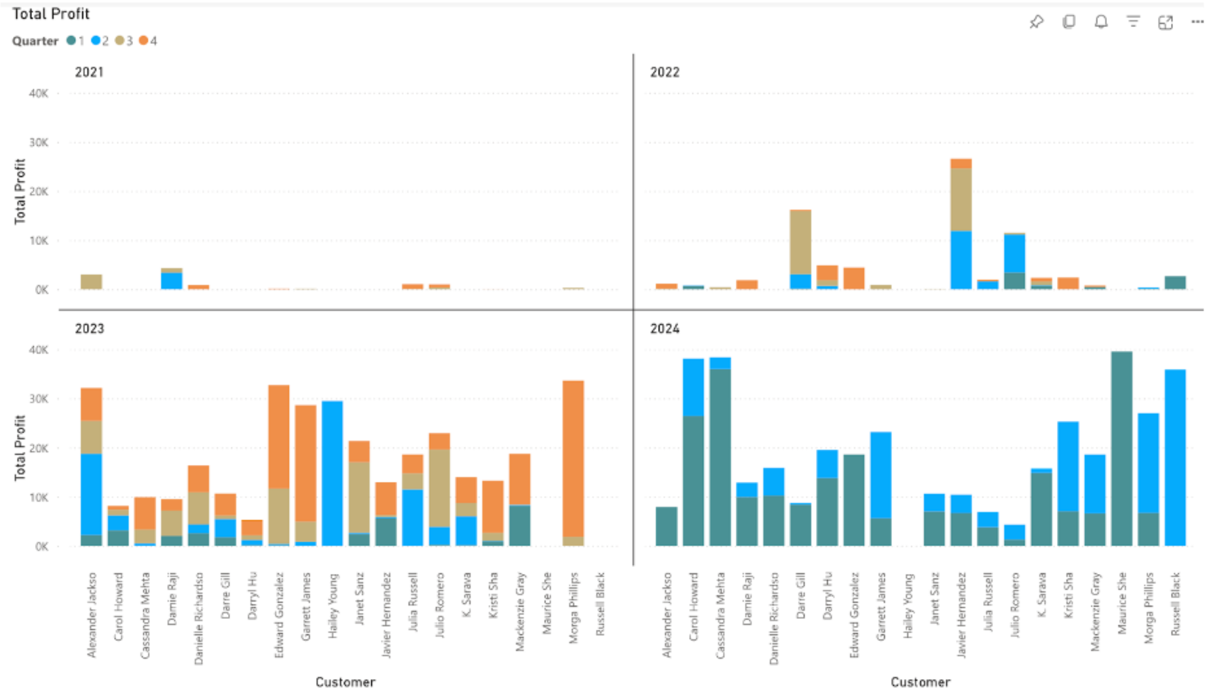
In 2021, the company's most profitable customer was Damie Raji of San Francisco, United States. In 2022, this distinction was held by Javier Hernandez from England, United Kingdom. In 2023, Hailey Young, based in Oregon, United States, emerged as the most profitable customer. By 2024, the leading customer in terms of profitability was Russell Black, also from San Francisco, United States. Figures below visualises the key customers on a map by user.



From a quarterly perspective, Maurice She (California, United States) and Cassandra Mehta (Nord, France) were the primary customers contributing to profitability in Q1. In Q2, the key customers were Russell Black (San Francisco, United States) and Hailey Young (Oregon, United States). For Q4, the principal customers were Morga Phillips(California, United States) and Garrett James (Washington, United States).

Profit generated in Q3 was notably lower than in the other quarters, with Julio Romero (British Columbia, Canada) identified as the primary customer for that period. In summary, the majority of the company's key customers are located within the United States. This suggests that the company's current marketing, distribution, and e-commerce operations in the U.S. are performing effectively.

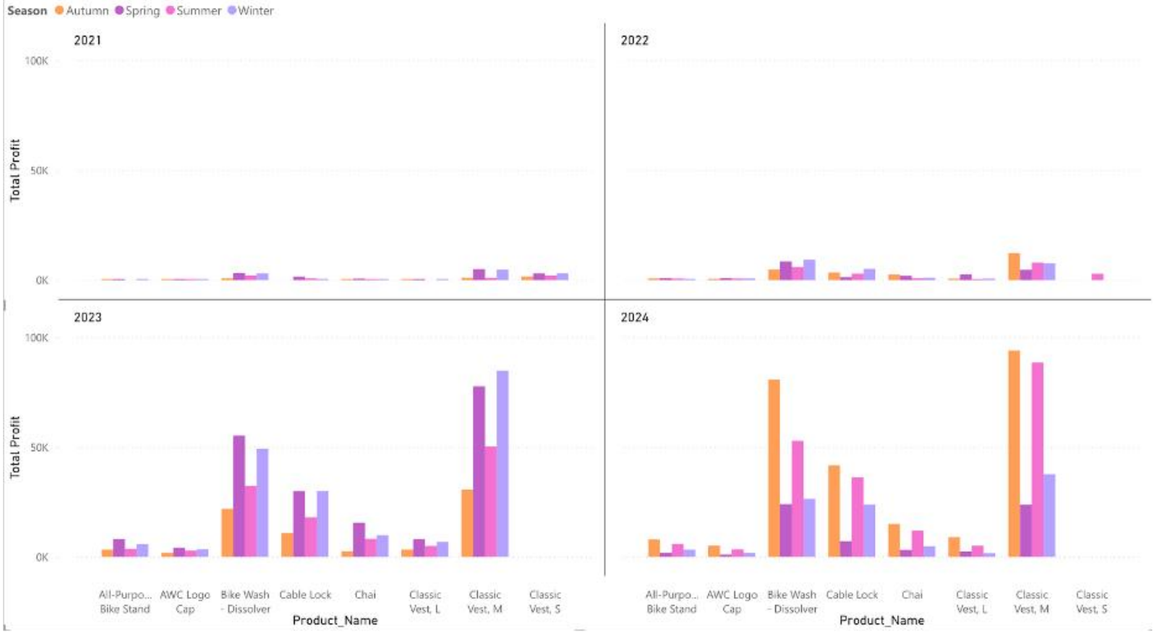
Therefore, it is recommended that HH strengthen their presence in the U.S. by enhancing marketing, retailer engagement and stock availability in top-performing states. This region consistently delivers the highest-profit customers hence, representing the lowest-risk, highest-return opportunity for scalable growth. Prioritising additional investment in this region allows HH to maximise returns by expanding what is already working, rather than diverting resources into less predictable markets.



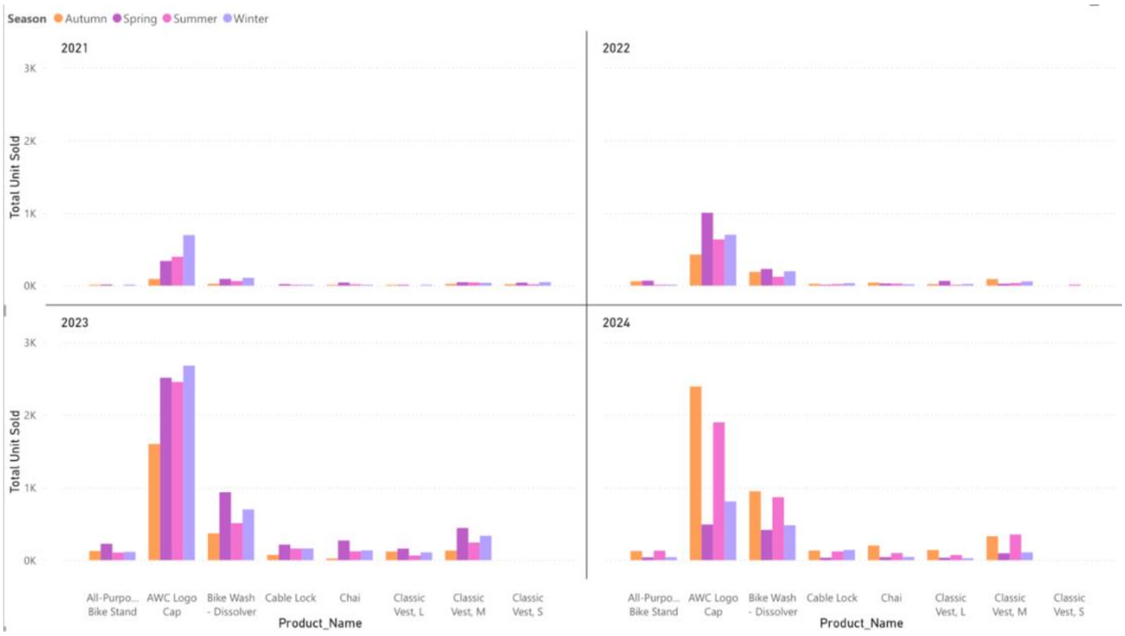
1.4.2. Most Profitable Products

We designed a query to SELECT the *FactSales* metrics related to profitability, then INNER JOIN and GROUP BY dimensions that incorporate product, temporal and geographic details. A CASE expression was used to implement the required geography-based season logic (northern vs southern hemisphere).

The Classic Vest (Medium size) has been the most profitable product from 2021 through 2024, followed by the Bike Wash – Dissolver. These two items consistently rank as the most profitable across all seasons.



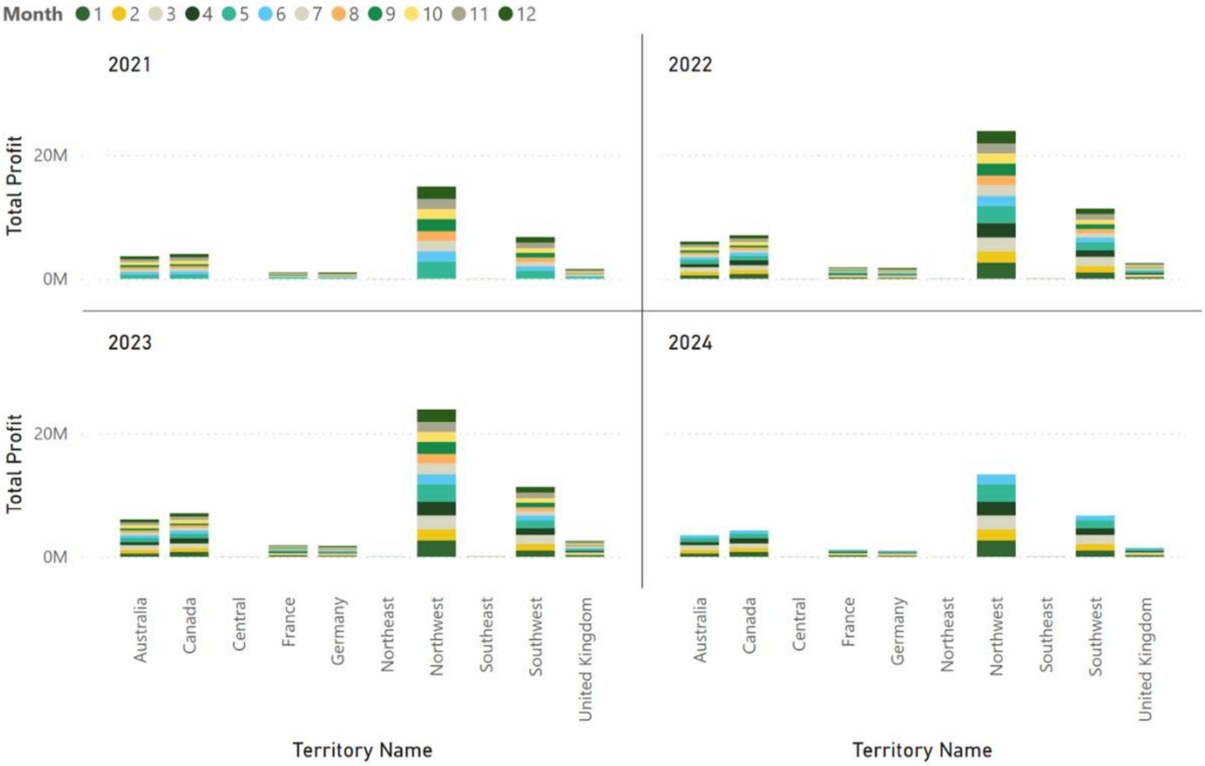
However, when examining total units sold, the AWC Logo Cap emerges as the top-selling product, with sales volume approximately four to five times higher than that of the Classic Vest (M). This disparity indicates that the profit margin of the Classic Vest is significantly higher than that of other products.



The analysis suggests that HH should prioritise production, stocking and targeted promotion of high-margin products such as the Classic Vest (M) and Bike Wash Dissolver, as these items consistently generate the highest profitability across years and seasons. These products deliver significantly higher margins than high-volume items like the AWC Logo Cap, meaning each unit sold contributes more to overall profit. By ensuring these top-performing products are well-stocked and more prominently marketed, HH can increase total profitability without relying solely on higher sales volumes.

1.4.3. Most Profitable Territory

We designed a SQL query that joins the *FactSales* table with *TimeDimension* table, *TerritoryDimension* table, and *SupplierDimension* table to include descriptive information for each transaction. The results are grouped by the three-dimension attributes and finally sorted in descending order of total profit, highlighting the most profitable supplier–territory–month combinations first. Since 2020, the Northwest region of the United States has consistently been the most profitable territory, generating the highest number of transactions monthly. In terms of total profit, the Northwest outperforms all other regions, earning nearly twice as much as the Southwest, which ranks as the second most profitable territory. However, when examining sales volume, the difference between the two regions is less pronounced. Although the Northwest still leads, its total units sold are only about 1.3 times higher than those of the Southwest, the second-highest region in sales quantity.

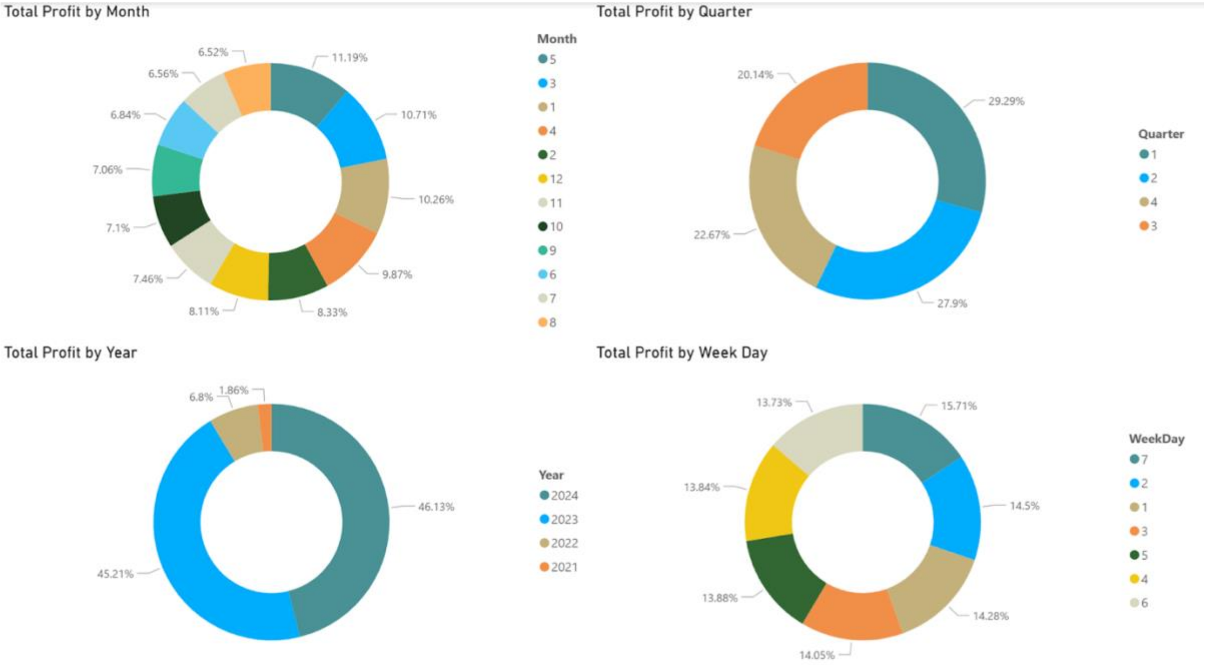


Based on these findings, HH should enhance operational efficiency and customer reach in the Northwest by analysing what specific factors are driving its sustained profitability such as delivery speed, retailer density, or product mix. This analysis should formalise these as best-practice models to replicate in the Southwest and other emerging territories. Although the Northwest leads significantly in total profit, the relatively small difference in sales volume compared to the Southwest suggests that profit gaps may be driven by operational or structural advantages rather than demand alone. By identifying and standardising these successful practices, HH can elevate the performance of other territories without relying solely on increased marketing spend.

1.4.4. Most Profitable Time periods

A SQL query is used to join the *FactSales* table with the *TimeDimension* table to access detailed date attributes, then aggregates data by calculating totals for units sold, sales revenue, profit, ROI profit, and profit/loss. The results are grouped by the selected time attributes to provide a clear breakdown of performance by day and period, and are ordered chronologically with the highest-profit entries appearing first within each group.

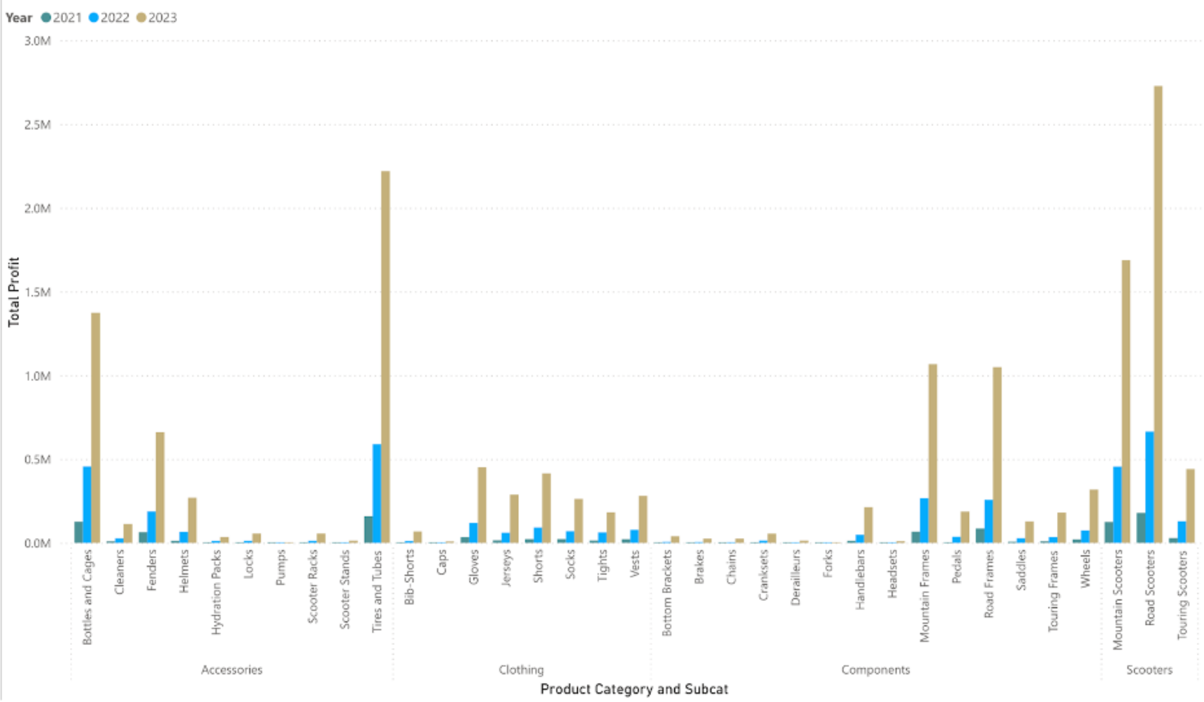
Based on the analysis of unit sales, dollar sales, and profit margins across multiple timeframes, several clear patterns emerge regarding the most profitable periods. March, 2023, Q1, and Sundays consistently stand out as the most profitable times of the year. While profitability peaks during these periods, the highest total sales—in terms of revenue—and the highest unit sales both occur in May, 2023, Q1, and also on Sundays. This distinction suggests that although May generates greater sales volume and revenue, March delivers superior margins, making it the most profitable month overall. These insights highlight the importance of seasonal and weekly trends in driving both sales performance and profitability.



These results indicate that HH should optimise promotional timing and inventory planning by aligning high-margin product campaigns with March and other peak-profit periods, while using May's high sales volume to drive broader promotional activity and inventory turnover. Sundays, which consistently show elevated weekly demand, should be treated as a natural high-traffic day. HH should support strategies such as ensuring stock availability, faster order fulfilment, or subtle weekend-focused digital promotions on Sundays. By differentiating between margin-peak months and volume-peak periods, HH can more effectively balance profitability with sales throughput and translate seasonal insights into targeted, revenue-enhancing actions.

1.4.5 Low-Value Categories/Sub-categories

By joining the *FactSales* table with the *TimeDimension* table and *ProductDimension* table, it enriches each sales record with descriptive product information and date attributes. The query then aggregates key metrics—such as total units sold, total sales, total profit, ROI percentage, and profit or loss—for each combination of product category, subcategory, year, quarter, and month. The grouped results allow for clear comparison of product performance over time, and the final output is ordered chronologically, with periods showing the lowest profit appearing first. The figure below shows total profit of each product category and subcategory by year. Scooters have the highest profit in the most recent years. For other product categories, quite a few items are not giving much return on investment.



For example, in ‘Accessories’, 'Tires and Tubes' consistently held the largest portion of profit across all years, demonstrating strong and increasing dominance. 'Bottles and Cages' also emerged as a substantial and growing profit contributor. Subcategories such as 'Pumps', 'Scooter Stands', 'Locks', 'Hydration Packs', and 'Scooter Racks' generally provided smaller but steady profits, with some exhibiting modest growth. Similarly, Caps within the ‘Clothing’ category, as well as Derailleurs, Forks, and Headsets within ‘Components’, contribute only a small proportion of the total profit and exhibit generally steady trends over time. Notably, Forks in the ‘Components’ category have even shown a gradual decline in performance across the years.

Thus, the following items can be considered to replace:

- Scooter racks
- Caps
- Forks
- Derailleurs
- Headsets

Removing these persistently underperforming products will reduce inventory and operational inefficiencies while allowing HH to reallocate resources toward high-margin, high-growth items. This targeted rationalisation strengthens the product portfolio by ensuring investment is focused on categories that reliably drive value and support long-term profitability.

Conclusion

The completed data warehouse gives HH a reliable foundation for ongoing analytics, turning previously disconnected sales data into clear, strategic insight. Beyond highlighting profitable customers, products, territories and time periods, the analysis points to actions that can directly improve performance. These include expanding in high-return U.S. markets, prioritising high-margin products, applying Northwest operational strengths across regions and discontinuing consistently low-value items. Together, these recommendations provide a focused pathway for improving profitability and efficiency. The warehouse now equips HH with a scalable, evidence based decision-making environment that will continue to support targeted growth as the business expands.

1.5. AI Usage

All analysis, arguments, and written content represent the authors' own work. All AI outputs were critically reviewed, edited for accuracy, and finalised by the authors.

References

Kimball, R., & Ross, M. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling*. John Wiley & Sons. Retrieved from: https://books.google.com.au/books?hl=en&lr=&id=4rFXzk8wAB8C&oi=fnd&pg=PR27&dq=importance+dimensional+modelling+data+analytics+&ots=3q7TRfZ5QK&sig=PIUpnkD9DPiLLY71AzTxjxkfgRY#v=onepage&q=importance%20dimensional%20modelling%20data%20analytics&f=false

Appendices

Appendix 1 – Data Dictionary

A1-1. sales_fact

Attribute	Type	Description
Sales_Key	INT	Surrogate Sales key generated during ETL process
Time_Key	INT	Foreign surrogate key to Time dimension table
Customer_Key	INT	Foreign surrogate key to Customer dimension
Territory_Key	INT	Foreign surrogate key to Territory dimension
Supplier_Key	INT	Foreign surrogate key to Supplier dimension
Product_Key	INT	Foreign surrogate key to Product dimension
Sales_Person_ID	SMALLINT	Unique natural identifier of salesperson that processed the sales transaction
Sales_Order_Number	NVARCHAR (50)	Unique natural identifier for the sales transaction
Order_Line_ID	SMALLINT	Identifier that specifies position of sale transaction within Sales Order
Standard_Cost	FLOAT	Initial price of product when purchased from supplier
Retail_Price	FLOAT	Retail price of product, includes tax component
Unit_Price	FLOAT	Unit price of a product, where tax component has been removed
Order_Qty	SMALLINT	Specifies number of products sold during sales transaction
Line_Total	FLOAT	Specifies total dollar revenue generated from sales transaction
Profit_Loss	FLOAT	Total profit made / loss incurred from sales transaction
ROI_Profit	FLOAT	Percentage of profit/loss made from sales transaction
Tax_Margin	FLOAT	Total tax that has been applied to the product
Profit_Margin_PU	FLOAT	Profit / Loss margin per unit of product sold

A1-2. product_dimension

Attribute Name	Type	Description
Product_Key	INT	Surrogate Product key generated during ETL process
Product_ID	NVARCHAR (50)	Unique natural Product key
Product_Name	NVARCHAR (50)	Name of product
Product_Cat_ID	SMALLINT	Unique category identification code
Product_Cat_Name	NVARCHAR (50)	Name of product category
Product_Subcat_ID	SMALLINT	Unique subcategory identification code

Product_Subcat_Name	NVARCHAR (50)	Name of product subcategory
Supplier_ID	SMALLINT	Natural Supplier key
Supplier_Name	NVARCHAR (100)	Name of product's supplier

A1-3. territory_dimension

Attribute Name	Type	Description
Territory_Key	INT	Surrogate Territory key generated during ETL process
Territory_ID	SMALLINT	Unique natural Territory identifier
Territory_Code	NVARCHAR (50)	Territory's identifier code
Territory_Name	NVARCHAR (50)	Name of territory
Region	NVARCHAR (50)	Sales country group that territory falls within
Sales_Person_ID	SMALLINT	Identifier that specifies salesperson assigned to sales Region

A1-4. supplier_dimension

Attribute Name	Type	Description
Supplier_Key	INT	Surrogate Supplier key generated during ETL process
Supplier_ID	INT	Unique natural Supplier Key
Supplier_Name	NVARCHAR (100)	Name of supplier
Contact_First_Name	NVARCHAR (100)	First name of supplier's contact
Contact_Last_Name	NVARCHAR (100)	Last name of supplier's contact
Contact_Full_Name	NVARCHAR (100)	Full name of supplier contact
Contact_Email	NVARCHAR (100)	Email address for supplier
Phone	NVARCHAR (20)	Phone contact number for supplier
Address	NVARCHAR (200)	Street address for supplier
City	NVARCHAR (100)	City where supplier is located
Country	NVARCHAR (100)	Country where supplier is located

A1-5. time_dimension

Attribute Name	Type	Description
Time_Key	INT	Surrogate Time key generated during ETL process
Date	INT	Full date
Day	INT	Day value of date
Month	INT	Month value of date
Year	INT	Year value of date
WeekDay	INT	Day of week that date falls on: • 1 – Monday

		- 2 – Tuesday - 3 – Wednesday - 4 – Thursday - 5 – Friday - 6 – Saturday - 7 - Sunday
Season_North	NVARCHAR (6)	Season if date occurred in the Northern Hemisphere
Season_South	NVARCHAR (6)	Season if date occurred in the Southern Hemisphere
Quarter	INT	Quarter in year that date falls on: - 1 – Jan (1) – March (3) - 2 – Apr (4) – Jun (6) - 3 – Jul (7) – Sept (9) - 4 – Oct (10) – Dec (12)

A1-6. customer_dimension

Attribute Name	Type	Description
Customer_Key	INT	Surrogate customer key generated during ETL process
Customer_ID	SMALLINT	Unique natural customer identifier
Person_ID	SMALLINT	Unique natural individual/personal customer identifier
Person_First_Name	NVARCHAR (50)	Customer’s first name
Person_Last_Name	NVARCHAR (50)	Customer’s last name
Person_Full_Name	NVARCHAR (101)	Customer’s full name
Store_ID	NVARCHAR (50)	Unique natural identifier for business customers
Store_Name	NVARCHAR (50)	Name of business
Customer_Type	NVARCHAR (50)	Specifies if a customer is: - An individual retail customer - A business - A contact associated with a business
Address_ID	SMALLINT	Unique natural identifier for customer address
Address_Line_1	NVARCHAR (50)	First address line for customer
Address_Line_2	NVARCHAR (50)	Second address line for customer
City	NVARCHAR (50)	City where customer is located
State _ID	SMALLINT	Unique state identifier
State_Name	NVARCHAR (50)	State where customer is located
State _Code	NVARCHAR (50)	State identification code
Postal_Code	NVARCHAR (50)	Customer’s postal code
Territory_ID	SMALLINT	Unique natural Territory identifier
Territory_Code	NVARCHAR (50)	Territory’s identifier code

Appendix 2 – SSIS Transformations

Refer to the ZIP file attached to this report.

Appendix 3 – SQL

The following SQL commands were used to create the dimensional model table structures in the MySQL data warehouse:

A3-1. Product Dimension

```
create table Product_Dimension (
            Product_Key int not null auto_increment,
   Product_ID NVARCHAR(50),
   Product_Name NVARCHAR(50),
   Product_Cat_ID SMALLINT,
   Product_Cat_Name NVARCHAR(50),
   Product_Subcat_ID SMALLINT,
   Product_Subcat_Name NVARCHAR(50),
   Supplier_ID Smallint,
   Supplier_Name NVARCHAR(100),
   Primary Key (Product_Key)
);
```

A3-2. Customer Dimension

```
CREATE TABLE customer_dimension (
   Customer_Key int not null auto_increment,
   Customer_ID SMALLINT,
   Person_ID SMALLINT,
   Person_First_Name NVARCHAR(50),
   Person_Last_Name NVARCHAR(50),
   Person_Full_Name NVARCHAR(101),
   Store_ID NVARCHAR(50),
   Store_Name NVARCHAR(50),
   Customer_Type NVARCHAR(50),
   Address_ID SMALLINT,
   Address_Line_1 NVARCHAR(50),
            Address_Line_2 NVARCHAR(50),
   City NVARCHAR(50),
   State_ID SMALLINT,
   State_Name NVARCHAR(50),
   State_Code NVARCHAR(50),
   Postal_Code NVARCHAR(50),
   Territory_ID SMALLINT,
   Territory_Code NVARCHAR(50),
   Primary Key (Customer_key)
);
```

A3-3. Territory Dimension

```
CREATE TABLE territory_dimension (
   Territory_Key int not null auto_increment,
   Territory_ID SMALLINT,
   Territory_Code NVARCHAR(50),
   Territory_Name NVARCHAR(50),
   Region NVARCHAR(50),
   Sales_Person_ID SMALLINT,
   Primary Key (territory_key)
);
```

A3-4. Supplier Dimension

```
CREATE TABLE supplier_dimension (
   Supplier_Key int not null auto_increment,
   Supplier_ID INT,
   Supplier_Name NVARCHAR(100),
   Contact_First_Name NVARCHAR(100),
   Contact_Last_Name NVARCHAR(100),
   Contact_Full_Name NVARCHAR(100),
   Contact_Email NVARCHAR(100),
   Phone NVARCHAR(20),
   Address NVARCHAR(200),
   City NVARCHAR(100),
   Country NVARCHAR(100),
   Primary Key (supplier_key)
);
```

A3-5. Time Dimension

```
CREATE TABLE time_dimension (
   Time_Key int not null auto_increment,
   Date DATE,
   Day INT,
   Month INT,
   Year INT,
   WeekDay INT,
   Season_North NVARCHAR(6),
   Season_South NVARCHAR(6),
   Quarter INT,
   Primary Key (Time_Key)
);
```

A3-6. Sales Fact

```
CREATE TABLE Sales_Fact (
   Sales_Key int not null auto_increment,
   Time_Key INT,
   Customer_Key INT,
   Territory_Key INT,
   Supplier_Key INT,
   Product_Key INT,
   Sales_Person_ID SMALLINT,
            Sales_Order_ID NVARCHAR(50),
   Order_Line_ID SMALLINT,
   Standard_Cost float,
   Retail_Price float,
   Unit_Price float,
   Order_Qty SMALLINT,
   Line_Total float,
   Profit_Loss float,
   ROI_Percent float,
   Tax_Margin float,
   Profit_Margin_PU float,
   Primary Key (Sales_Key)
);
```