**AdventureWorks**

# Table of Contents

**Section Title Page**

1. **Project Overview 3**
2. **Business Context 3**
3. **Project Scope 3**
4. **Key Entities and Relationships 4**
5. **Technical Approach 8**
6. **Data Model 8** 
   1. **Schema Design: Snowflake 8**
   2. **Detailed Table Structures 9**
7. **Code 17**
8. **Outputs 20**
9. **In-Depth Analysis of the Data Model 24**
10. **Challenges 42**
11. **Deployment Strategies 43**

1. **Considerations and Future Expansion 44**
2. **Conclusion 44**
3. **Project Overview**

AdventureWorks, a prominent bicycle manufacturing and sales company, requires a robust SQL database model to serve as the foundation for their sales data warehouse. This project aims to design and implement a comprehensive database schema that will efficiently store and organize all critical sales-related information.

1. **Company Overview** 
   * **Core Business:** 
     + Primarily focuses on bicycle manufacturing and sales o Operates with multiple product lines and categories o Maintains a complex supply chain network
     + Has a diverse customer base with different customer types
   * **Current Business State:** 
     + Experiencing significant growth phase o Managing increasing sales volumes o Dealing with growing data complexity o Operating across multiple territories and regions o Maintaining relationships with various suppliers and vendors
   * **Operational Challenges:** 
     + Lacks centralized data management system o Struggling with data organization and structure o Limited ability to perform efficient analysis o Difficulty in making data-driven decisions o Managing complex inventory across multiple warehouses
   * **Business Infrastructure:** 
     + Multiple warehouses and storage facilities o Various sales territories and regions o Departmentalized organizational structure o Multiple customer touchpoints o Complex supply chain network

1. **Project Scope**

**Data Warehouse Development:**

* + Design and implementation of a comprehensive SQL database model
  + Creation of a robust sales data warehouse
  + Development of snowflake schema architecture
  + Implementation of fact and dimension tables **Key Focus Areas:**
  + Sales transaction management
  + Inventory tracking and control
  + Purchase order processing
  + Customer relationship management
  + Product hierarchy management
  + Employee and department organization
  + Territory and location management

**Technical Implementation:**

* + SQL Server database implementation
  + Data integrity and constraint management
  + Query optimization and indexing
  + Performance tuning and scalability
  + Integration with existing systems

1. **Key Entities and Relationships**

1. **Sales (Fact Table - FactSales)** 
   * Primary entity capturing all sales transactions • Key Attributes:
     + SalesKey (Primary Key) o Order details (date, quantity, prices) o Financial metrics (total sales, costs, discounts) o Transaction identifiers (order numbers, PO numbers)
   * **Business Rules:** 
     + All sales must have a valid customer and product o Sales amount must be greater than or equal to cost o Orders must have valid payment methods
   * **Links to dimensions:** o Customer o Product o Employee (Salesperson) o Territory
     + ShipMethod
     + Date (multiple references for order, due, and ship dates)
2. **Purchasing (Fact Table - FactPurchasing)** 
   * **Captures all procurement transactions** • **Key Attributes:** 
     + PurchaseKey (Primary Key) o Order quantities and costs o Purchase order details o Quality ratings o Urgency flags
   * **Business Rules:** 
     + All purchases must have approved suppliers o Quality ratings required for received goods o Emergency orders must be flagged
   * **Links to dimensions:** o Supplier o Product o Employee
     + Warehouse
     + ShipMethod
     + Date (multiple references for order, due, and received dates)
3. **Inventory (Fact Table - FactInventory)** 
   * **Tracks all inventory movements and status** • **Key Attributes:** 
     + InventoryKey (Primary Key) o Stock quantities and values o Location information o Quality status o Reorder points
   * **Business Rules:** 
     + Stock levels must be non-negative o All movements must have valid transaction types o Quality status must be maintained
   * **Links to dimensions:** o Product o Warehouse o Date
4. **Product Hierarchy**

**1. Product (DimProduct)** o **Core product information** o **Key Attributes:**

* + - * Product details (name, cost, price)
      * Physical attributes (color, size, weight)
      * Manufacturing information
      * Status indicators o **Business Rules:**
      * Must belong to a valid subcategory
      * Standard cost must be less than list price
      * Safety stock levels must be defined

**2. Subcategory (DimProductSubcategory)** o **Product grouping level** o **Key Attributes:**

* + - * Subcategory identification
      * Names and descriptions
      * Marketing materials (images) o **Business Rules:**
      * **Must belong to a valid category**
      * **Must have unique subcategory ID**

**3. Category (DimProductCategory)** o **Highest level product grouping** o **Key Attributes:**

* + - * Category identification
      * Names and descriptions
      * Marketing materials
    - **Business Rules:** 
      * Must have unique category ID
      * Must maintain valid product hierarchy

1. **Customer Hierarchy**

**1. Customer (DimCustomer)** o **Customer master data** o **Key Attributes:**

* + - * Personal information
      * Contact details
      * Location information
      * Account details o **Business Rules:**
      * Must have valid customer type
      * Credit limits must be enforced
      * Valid contact information required

**2. Customer Type (DimCustomerType)** o **Customer classification** o **Key Attributes:**

* + - * Type definitions
      * Discount rules
      * Payment terms o **Business Rules:**
      * Standard discount rates must be defined
      * Minimum order amounts may apply

1. **Employee Hierarchy**

**1. Employee (DimEmployee)** o **Employee master data** o **Key Attributes:**

* + - * Personal information
      * Contact details
      * Employment details
      * Status indicators o **Business Rules:**
      * Must belong to valid department
      * Must have unique employee ID
      * Valid contact information required

**2. Department (DimDepartment)** o **Organizational structure** o **Key Attributes:**

* + - * Department information
      * Budget details
      * Management structure o **Business Rules:**
      * Must have valid manager
      * Budget amounts must be defined

1. **Territory (DimTerritory)** 
   * **Geographic sales regions** • **Key Attributes:** 
     + Region and country information o Organizational groupings o Sales targets
   * **Business Rules:** 
     + Must have valid manager o Sales targets must be defined o Valid cost center required

**Key Relationships**

1. **Sales-Related Relationships** 
   * **Sales → Product (Many-to-One)** o Each sale must reference exactly one product o Products can have multiple sales
   * **Sales → Customer (Many-to-One)** o Each sale must be associated with one customer o Customers can have multiple sales
   * **Sales → Employee (Many-to-One)** o Each sale must have one responsible employee o Employees can have multiple sales
   * **Sales → Territory (Many-to-One)** o Each sale belongs to one territory o Territories can have multiple sales
2. **Product Hierarchy Relationships** 
   * **Product → Subcategory (Many-to-One)** o Each product belongs to one subcategory o Subcategories can have multiple products
   * **Subcategory → Category (Many-to-One)** o Each subcategory belongs to one category o Categories can have multiple subcategories
3. **Customer Hierarchy Relationships** 
   * **Customer → CustomerType (Many-to-One)** o Each customer has one customer type o Customer types can have multiple custo**mers**
4. **Employee Hierarchy Relationships** 
   * **Employee → Department (Many-to-One)** o Each employee belongs to one department o Departments can have multiple employees
5. **Inventory-Related Relationships** 
   * **Inventory → Product (Many-to-One)** o Each inventory record relates to one product o Products can have multiple inventory records
   * **Inventory → Warehouse (Many-to-One)** o Each inventory record belongs to one warehouse o Warehouses can have multiple inventory records
6. **Purchase-Related Relationships** 
   * **Purchasing → Supplier (Many-to-One)** o Each purchase must have one supplier o Suppliers can have multiple purchases
   * **Purchasing → Product (Many-to-One)** o Each purchase line relates to one product o Products can have multiple purchases

1. **Technical Approach** 
   * Utilize SQL Server or another suitable relational database management system
   * Implement a snowflake schema design
   * Use appropriate data types to ensure efficient storage and querying
   * Implement constraints (primary keys, foreign keys, unique constraints) to maintain data integrity
   * Design indexes to optimize query performance

1. **Data Model**



**6.1. Schema Design: Snowflake**

The chosen snowflake schema for AdventureWorks' data warehouse is a sophisticated design that balances normalization with query performance. This schema type is particularly well-suited for the company's needs due to the following reasons:

1. Hierarchical Data Representation: The product hierarchy (Category -> Subcategory -> Product) is perfectly represented in a snowflake schema.
2. Reduced Data Redundancy: By normalizing dimension tables, the schema significantly reduces data redundancy.
3. Improved Data Integrity: The normalized structure ensures that category and subcategory information is stored in separate tables, reducing the risk of data inconsistencies.
4. Flexible Querying: The snowflake design allows for more flexible querying, enabling easy aggregation at different levels of the product hierarchy.
5. Scalability: The schema can easily accommodate growth without requiring significant structural changes.

**6.2. Detailed Table Structures**

**FactSales**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| SalesKey | INT | PK | Unique identifier for each sales transaction |
| OrderDateKey | INT | FK | Reference to date dimension for order date |
| DueDateKey | INT | FK | Reference to date dimension for due date |
| ShipDateKey | INT | FK | Reference to date dimension for ship date |
| CustomerKey | INT | FK | Reference to customer dimension |
| EmployeeKey | INT | FK | Reference to employee who processed the sale |
| ProductKey | INT | FK | Reference to product dimension |
| SalesTerritoryKey | INT | FK | Reference to sales territory dimension |
| ShipMethodKey | INT | FK | Reference to shipping method dimension |
| CurrencyKey | INT | FK | Reference to currency dimension |
| Quantity | INT |  | Number of units sold |
| UnitPrice | DECIMAL |  | Price per unit |
| UnitCost | DECIMAL |  | Cost per unit |
| TotalSales | DECIMAL |  | Total sales amount |
| Cost | DECIMAL |  | Total cost amount |
| Discount | DECIMAL |  | Discount amount applied |
| TaxAmount | DECIMAL |  | Tax amount |
| Freight | DECIMAL |  | Shipping cost |
| SalesOrderNumber | VARCHAR |  | Unique sales order identifier |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| PONumber | VARCHAR |  | Purchase order number from customer |
| PaymentMethod | VARCHAR |  | Method of payment |
| Status | VARCHAR |  | Current status of the order |

**FactPurchasing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key** | **Description** |
| PurchaseKey | INT | **Type**PK | Unique identifier for each purchase transaction |
| OrderDateKey | INT | FK | Reference to date dimension for order date |
| DueDateKey | INT | FK | Reference to date dimension for due date |
| ReceivedDateKey | INT | FK | Reference to date dimension for receive date |
| SupplierKey | INT | FK | Reference to supplier dimension |
| ProductKey | INT | FK | Reference to product dimension |
| EmployeeKey | INT | FK | Reference to employee who processed the |
| ShipMethodKey | INT | FK | purchaseReference to shipping method dimension |
| WarehouseKey | INT | FK | Reference to warehouse dimension |
| CurrencyKey | INT | FK | Reference to currency dimension |
| Quantity | INT |  | Number of units purchased |
| UnitCost | DECIMAL |  | Cost per unit |
| TotalCost | DECIMAL |  | Total cost amount |
| Freight | DECIMAL |  | Shipping cost |
| TaxAmount | DECIMAL |  | Tax amount |
| PurchaseOrderNumber | VARCHAR |  | Unique purchase order identifier |
| Status | VARCHAR |  | Current status of the purchase order |
| PaymentTerms | VARCHAR |  | Payment terms agreed with supplier |
| QualityRating | INT |  | Quality rating of received goods (1-5) |
| IsUrgent | BIT |  | Flag for urgent orders |

**FactInventory**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| InventoryKey | INT | PK | Unique identifier for each inventory record |
| DateKey | INT | FK | Reference to date dimension |
| ProductKey | INT | FK | Reference to product dimension |
| WarehouseKey | INT | FK | Reference to warehouse dimension |
| Quantity | INT |  | Current quantity in stock |
| UnitCost | DECIMAL |  | Cost per unit |
| TotalValue | DECIMAL |  | Total inventory value |
| TransactionType | VARCHAR |  | Type of inventory transaction |
| ReferenceNumber | VARCHAR |  | Reference to related transaction |
| BatchNumber | VARCHAR |  | Production batch number |
| ExpiryDateKey | INT | FK | Reference to date dimension for expiry date |
| ShelfLocation | VARCHAR |  | Storage location in warehouse |
| ReorderPoint | INT |  | Minimum quantity to trigger reorder |
| MaximumStock | INT |  | Maximum allowed quantity |
| StockOutDuration | INT |  | Duration of stock-out in hours |
| QualityStatus | VARCHAR |  | Current quality status of inventory |

**DimProduct**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| ProductKey | INT | PK | Unique identifier for each product |
| SubcategoryKey | INT | FK | Reference to product subcategory |
| ProductID | VARCHAR |  | Business identifier for the product |
| ProductName | VARCHAR |  | Name of the product |
| StandardCost | DECIMAL |  | Standard manufacturing cost |
| ListPrice | DECIMAL |  | Suggested retail price |
| Color | VARCHAR |  | Product color |
| Size | VARCHAR |  | Product size |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| Weight | DECIMAL |  | Product weight |
| UnitMeasure | VARCHAR |  | Unit of measurement |
| ProductLine | VARCHAR |  | Product line category |
| Style | VARCHAR |  | Style classification |
| ModelName | VARCHAR |  | Model name |
| Description | TEXT |  | Detailed product description |
| SafetyStockLevel | INT |  | Minimum safety stock quantity |
| ReorderPoint | INT |  | Quantity triggering reorder |
| DaysToManufacture | INT |  | Production time in days |
| ProductStatus | VARCHAR |  | Current product status |
| IsDiscontinued | BIT |  | Discontinued flag |
| DiscontinuedDate | DATE |  | Date product was discontinued |

**DimProductSubcategory**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| SubcategoryKey | INT | PK | Unique identifier for subcategory |
| CategoryKey | INT | FK | Reference to product category |
| SubcategoryID | VARCHAR |  | Business identifier for subcategory |
| SubcategoryName | VARCHAR |  | Name of the subcategory |
| Description | TEXT |  | Detailed subcategory description |
| ImageURL | VARCHAR |  | URL to subcategory image |

**DimProductCategory**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| CategoryKey | INT | PK | Unique identifier for category |
| CategoryID | VARCHAR |  | Business identifier for category |
| CategoryName | VARCHAR |  | Name of the category |
| Description | TEXT |  | Detailed category description |
| ImageURL | VARCHAR |  | URL to category image |

**DimCustomer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| CustomerKey | INT | PK | Unique identifier for customer |
| CustomerTypeKey | INT | FK | Reference to customer type |
| CustomerID | VARCHAR |  | Business identifier for customer |
| FirstName | VARCHAR |  | Customer's first name |
| LastName | VARCHAR |  | Customer's last name |
| Email | VARCHAR |  | Customer's email address |
| Phone | VARCHAR |  | Customer's phone number |
| AddressLine1 | VARCHAR |  | Primary address |
| City | VARCHAR |  | City name |
| StateProvince | VARCHAR |  | State or province |
| PostalCode | VARCHAR |  | Postal code |
| Country | VARCHAR |  | Country name |
| CreditLimit | DECIMAL |  | Approved credit limit |
| AccountOpenedDate | DATE |  | Account creation date |
| CustomerStatus | VARCHAR |  | Current customer status |

**DimCustomerType**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| CustomerTypeKey | INT | PK | Unique identifier for customer type |
| TypeName | VARCHAR |  | Name of customer type |
| Description | TEXT |  | Detailed type description |
| DiscountPercent | DECIMAL |  | Discount applicable to customer type |

**DimEmployee**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| EmployeeKey | INT | PK | Unique identifier for employee |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| EmployeeID | VARCHAR |  | Business identifier for employee |
| FirstName | VARCHAR |  | Employee's first name |
| LastName | VARCHAR |  | Employee's last name |
| Email | VARCHAR |  | Employee's email address |
| Phone | VARCHAR |  | Employee's phone number |
| HireDate | DATE |  | Hire date |
| PositionTitle | VARCHAR |  | Job title |
| DepartmentKey | INT | FK | Reference to department dimension |
| LocationKey | INT | FK | Reference to location dimension |
| IsSalesRep | BIT |  | Flag indicating if employee is a sales representative |
| IsManager | BIT |  | Flag indicating if employee is a manager |
| ReportsToKey | INT | FK | Reference to employee who they report to |
| EmploymentStatus | VARCHAR |  | Current employment status |

**DimDepartment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| DepartmentKey | INT | PK | Unique identifier for department |
| DepartmentID | VARCHAR |  | Business identifier for department |
| DepartmentName | VARCHAR |  | Name of the department |
| Description | TEXT |  | Detailed department description |
| Floor | INT |  | Floor number |
| ManagerKey | INT | FK | Reference to employee who manages the department |

**DimSupplier**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| SupplierKey | INT | PK | Unique identifier for supplier |
| SupplierID | VARCHAR |  | Business identifier for supplier |
| CompanyName | VARCHAR |  | Supplier company name |
| ContactName | VARCHAR |  | Primary contact person |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| Email | VARCHAR |  | Contact email address |
| Phone | VARCHAR |  | Contact phone number |
| Address | VARCHAR |  | Business address |
| City | VARCHAR |  | City location |
| Country | VARCHAR |  | Country location |
| CreditRating | INT |  | Credit rating (1-5) |
| PreferredVendorStatus | BIT |  | Preferred vendor flag |

**DimWarehouse**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| WarehouseKey | INT | PK | Unique identifier for warehouse |
| WarehouseName | VARCHAR |  | Name of warehouse |
| Location | VARCHAR |  | Geographic location |
| Country | VARCHAR |  | Country location |
| Capacity | INT |  | Storage capacity |
| Address | VARCHAR |  | Physical address |
| ManagerEmployeeKey | INT |  | Reference to warehouse manager |
| IsAutomated | BIT |  | Automated facility flag |
| TemperatureControlled | BIT |  | Temperature control flag |

**DimTerritory**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| SalesTerritoryKey | INT | PK | Unique identifier for territory |
| Region | VARCHAR |  | Geographic region |
| Country | VARCHAR |  | Country name |
| Group | VARCHAR |  | Territory group |
| Division | VARCHAR |  | Business division |
| ManagerEmployeeKey | INT |  | Reference to territory manager |
| CostCenter | VARCHAR |  | Cost center code |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| SalesTarget | DECIMAL |  | Annual sales target |

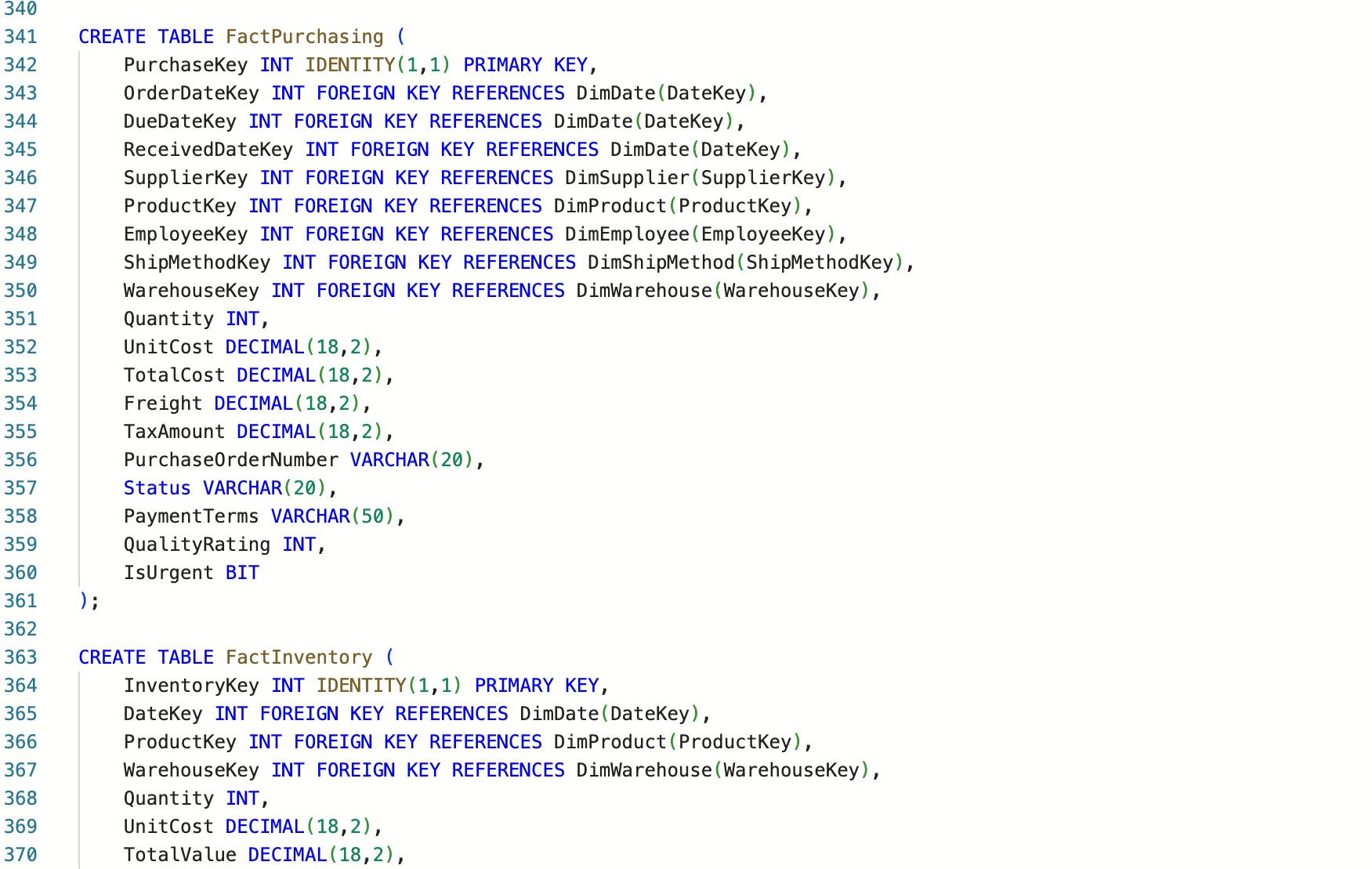
**DimDate**

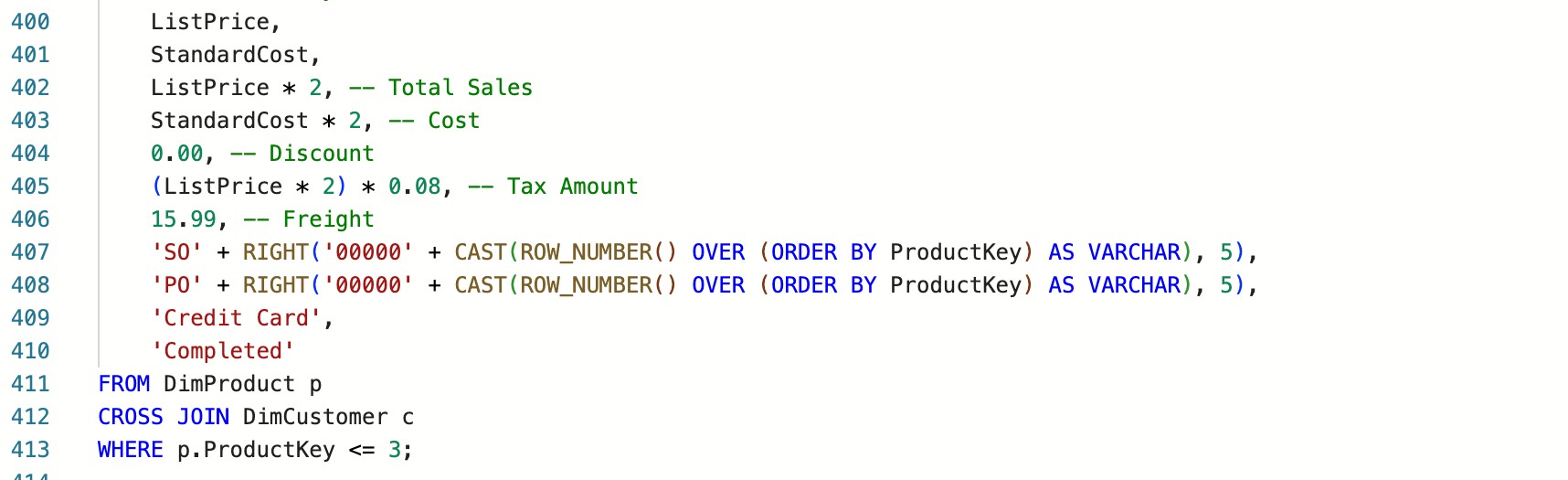
|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| DateKey | INT | PK | Unique identifier for date |
| FullDate | DATE |  | Calendar date |
| Year | INT |  | Calendar year |
| Quarter | INT |  | Calendar quarter (1-4) |
| Month | INT |  | Month number (1-12) |
| MonthName | VARCHAR |  | Month name |
| DayOfWeek | INT |  | Day of week (1-7) |
| DayName | VARCHAR |  | Day name |
| FiscalYear | INT |  | Fiscal year |
| IsHoliday | BIT |  | Holiday flag |

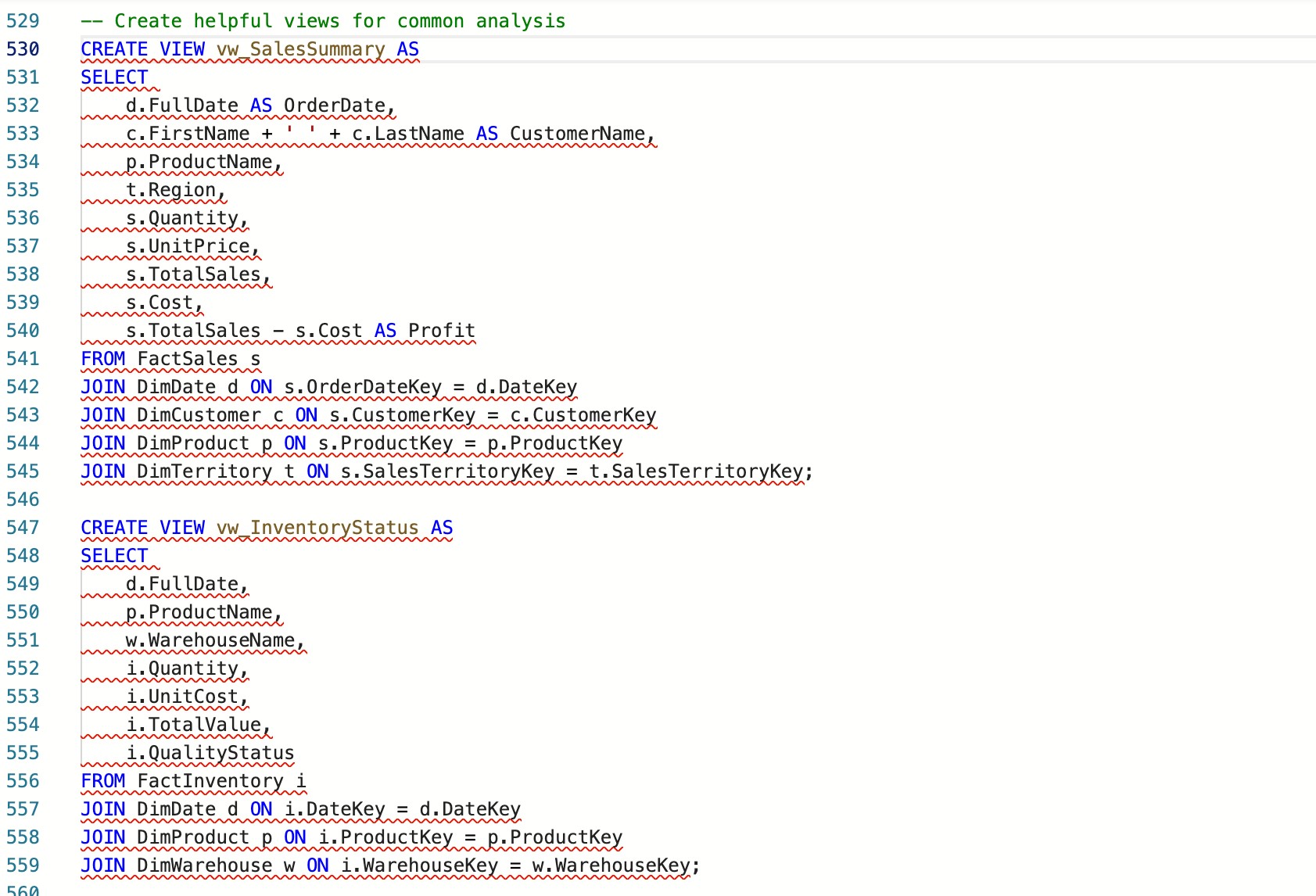
**DimShipMethod**

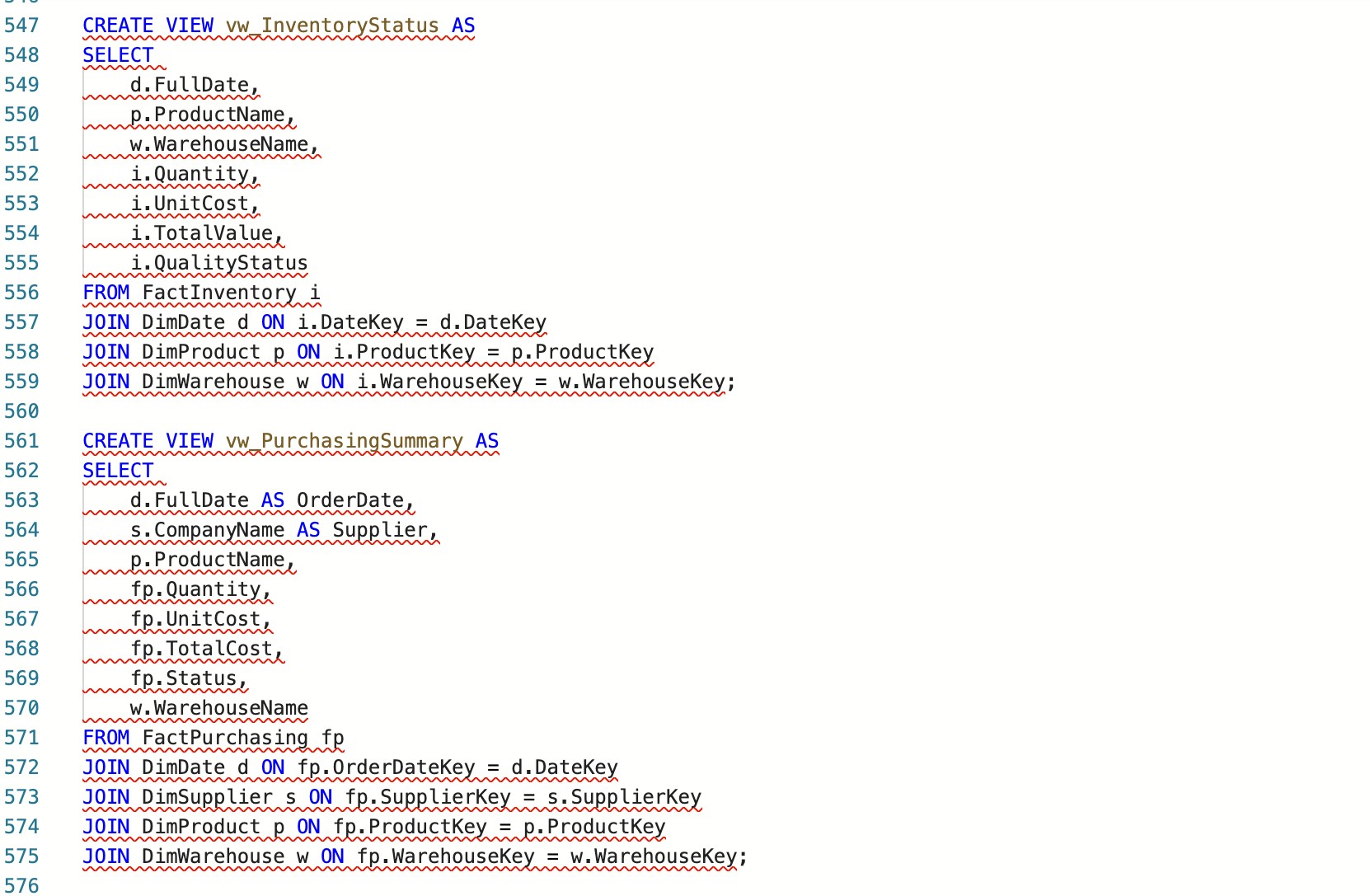
|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Key Type** | **Description** |
| ShipMethodKey | INT | PK | Unique identifier for shipping method |
| MethodName | VARCHAR |  | Name of shipping method |
| BaseRate | DECIMAL |  | Base shipping rate |
| EstimatedDays | INT |  | Estimated delivery days |
| CarrierName | VARCHAR |  | Shipping carrier name |
| TrackingAvailable | BIT |  | Tracking availability flag |
| MaxWeight | DECIMAL |  | Maximum weight limit |
| ActiveFlag | BIT |  | Active method flag |

1. **Code**







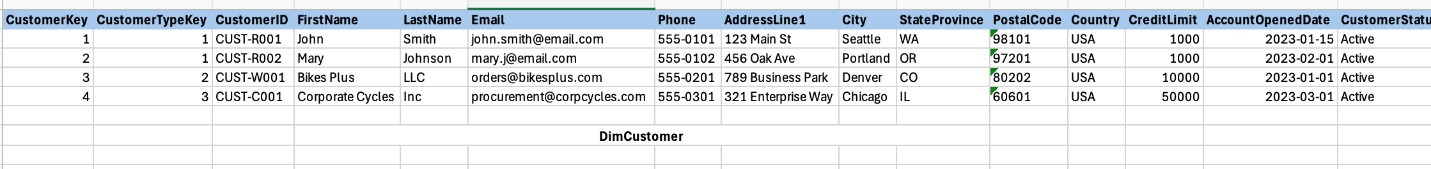


1. **Outputs**

**DimCustomerType**



**DimCustomer**



**DimDate**



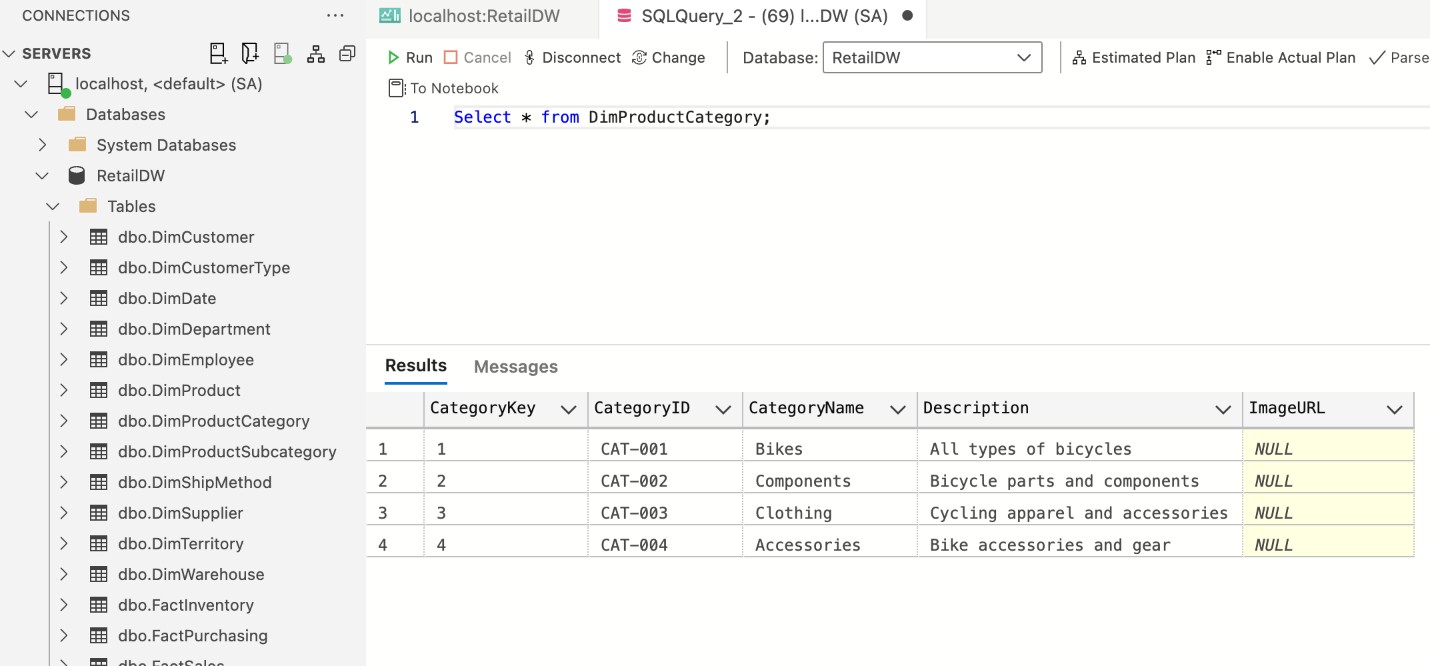
**DimDepartment**



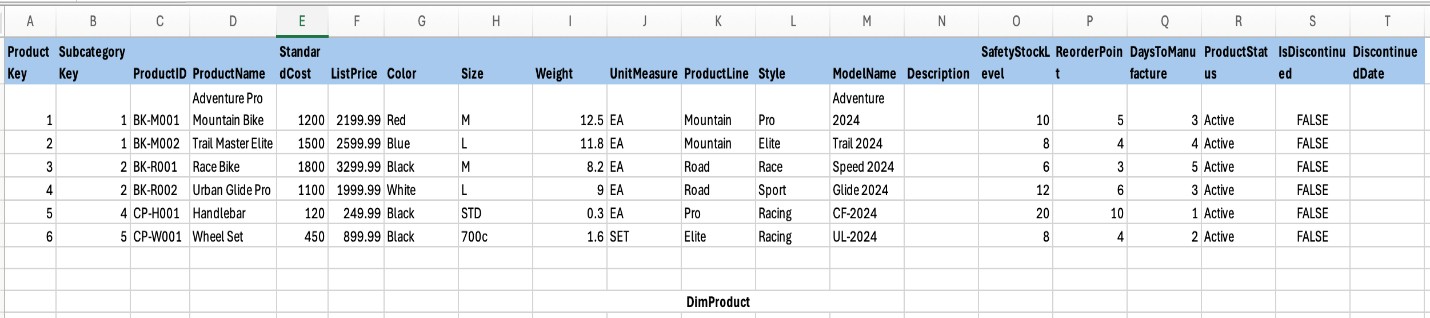
**DimEmployee**



**DimProductCategory**



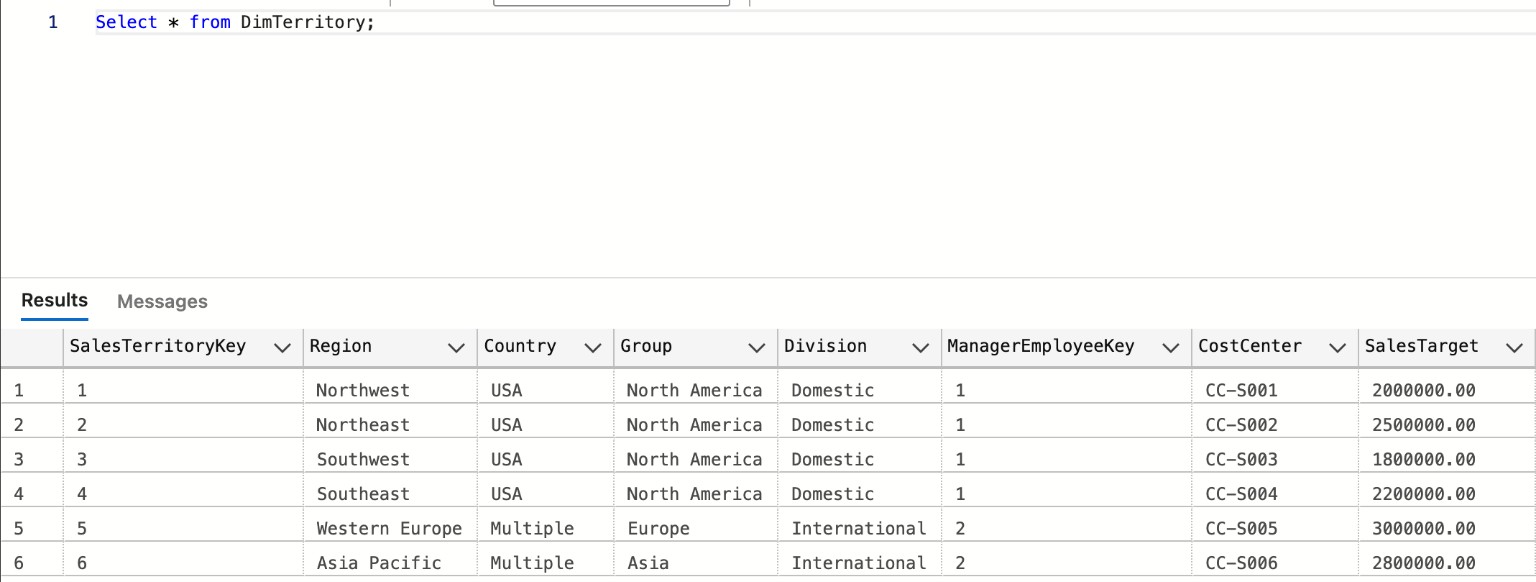
**DimProduct**



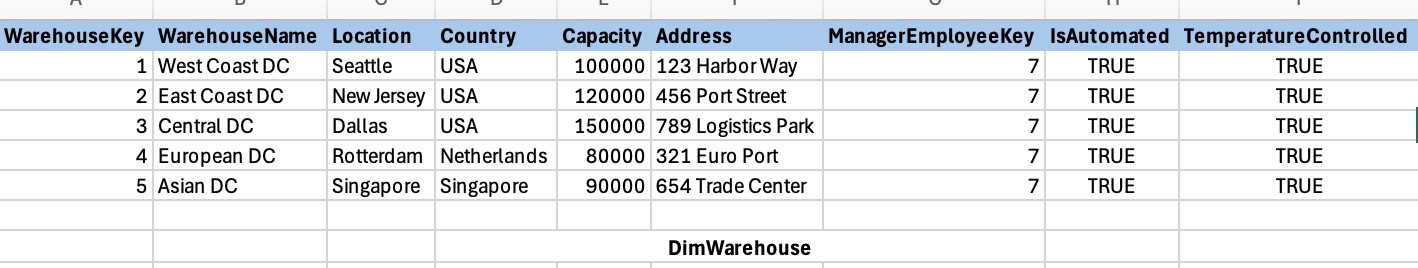
**DimSupplier**



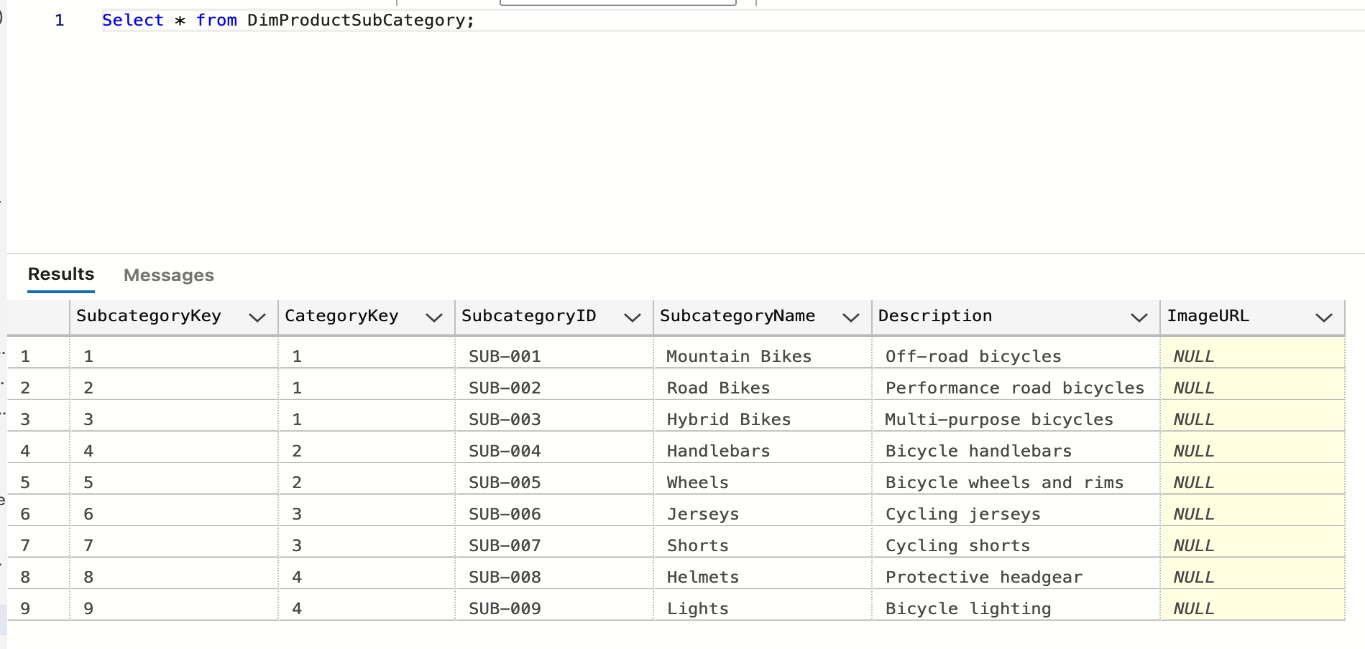
**DimTerritory**



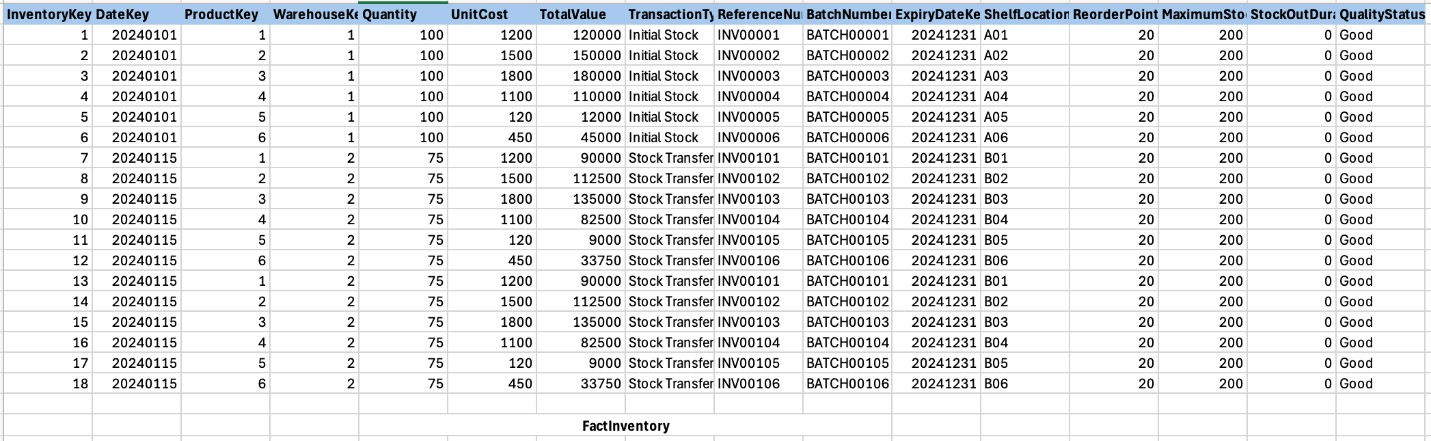
**DimWarehouse**



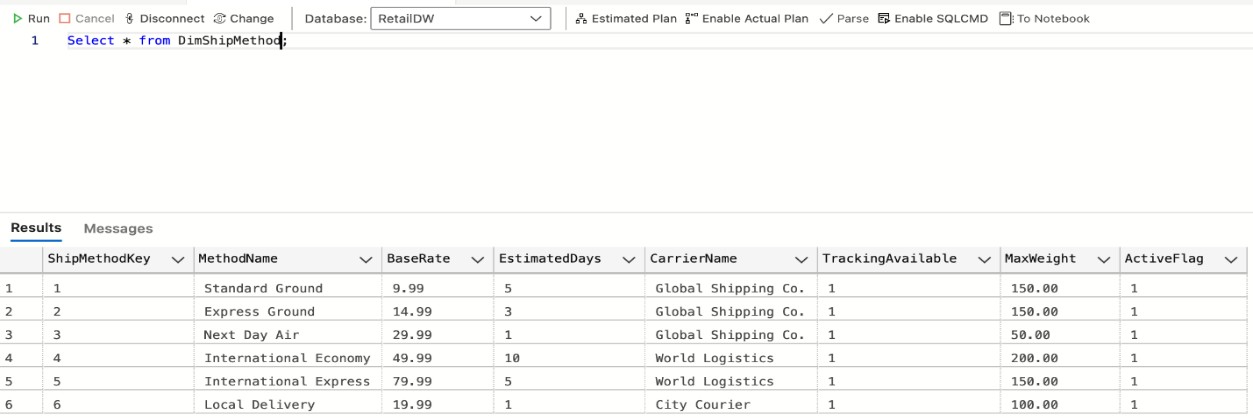
**DimProductSubCategory**



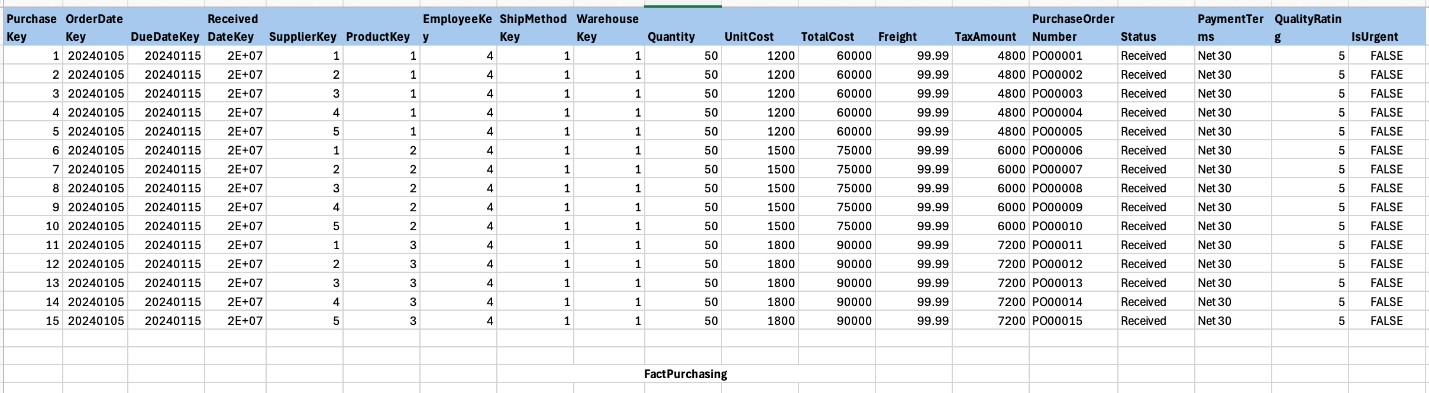
**FactInventory**



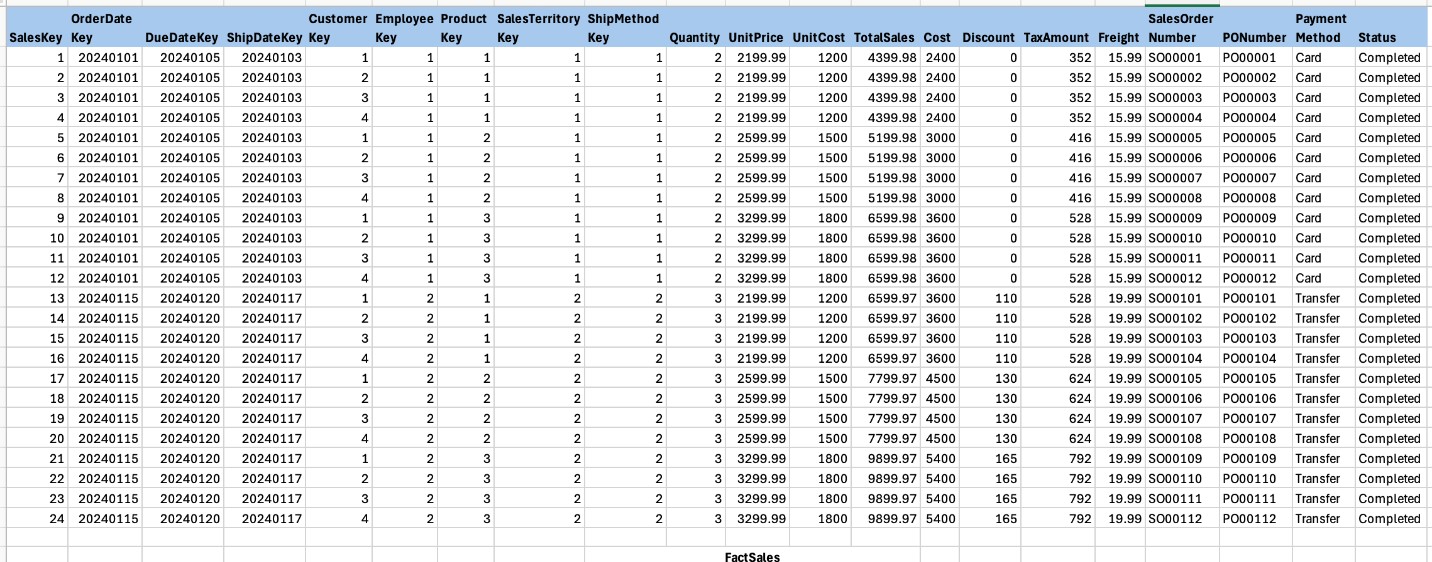
**DimShipMethod**



**FactPurchasing**



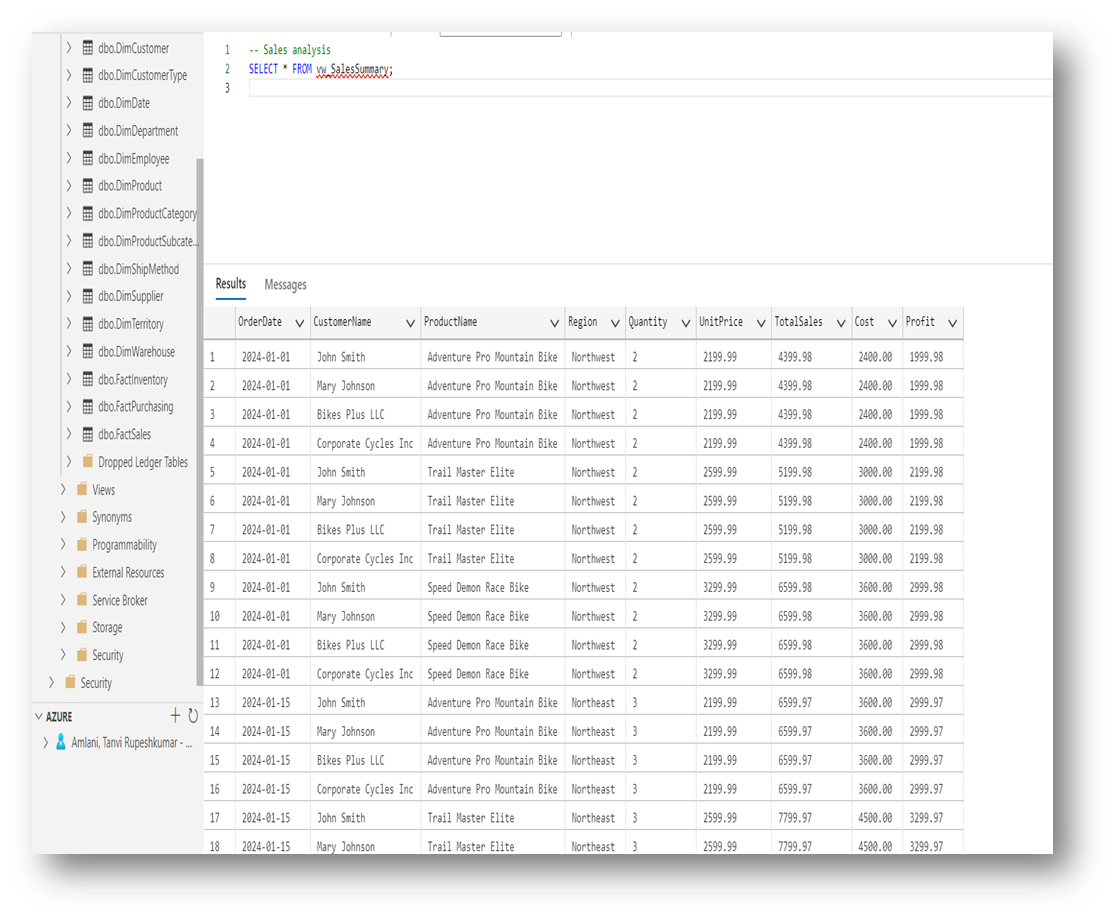
**FactSales**



**9. In-Depth Analysis of the Data Model**

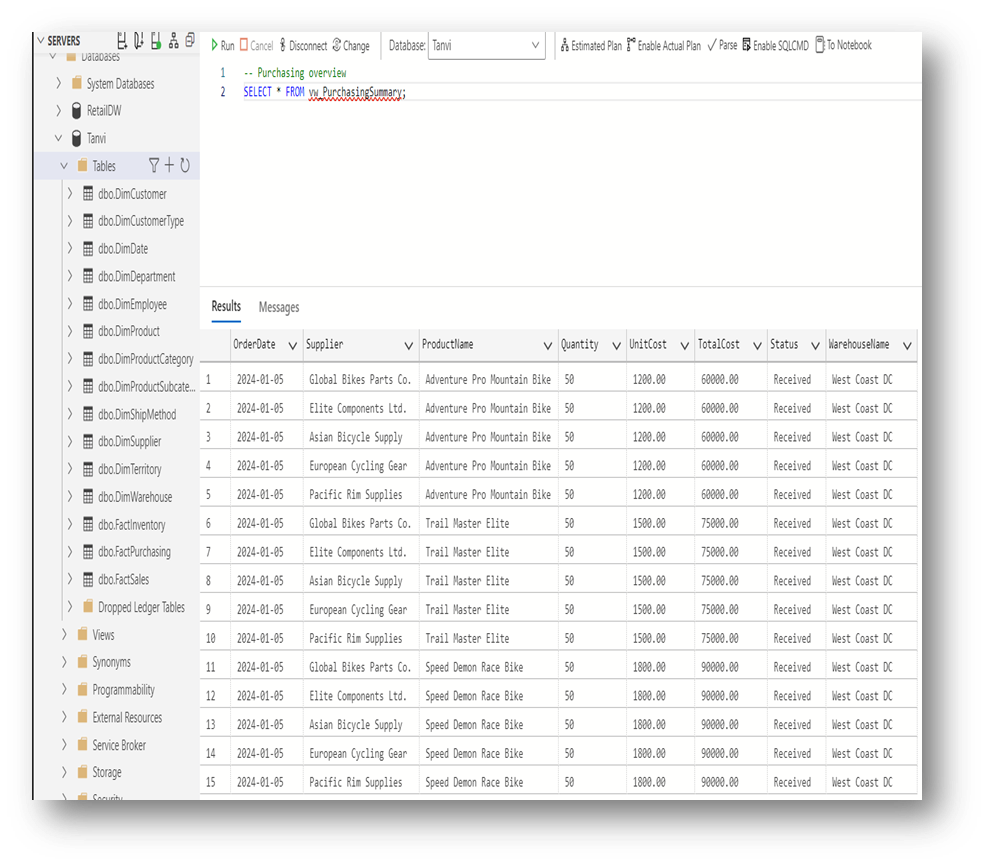
**Fact Table Analysis: Sales**

* **Grain:** Individual line item level of a sales order
* **Measures:** Quantity, UnitPrice, TotalSales, Cost, Discount, TaxAmount, Freight
* **Calculated Measures:** Profit (TotalSales - Cost), Margin ((TotalSales - Cost)/TotalSales)
* **Dimensional References:** References to Customer, Product, Territory, Employee, Date, ShipMethod
* **Date Dimensions:** OrderDateKey, DueDateKey, ShipDateKey for tracking order lifecycle



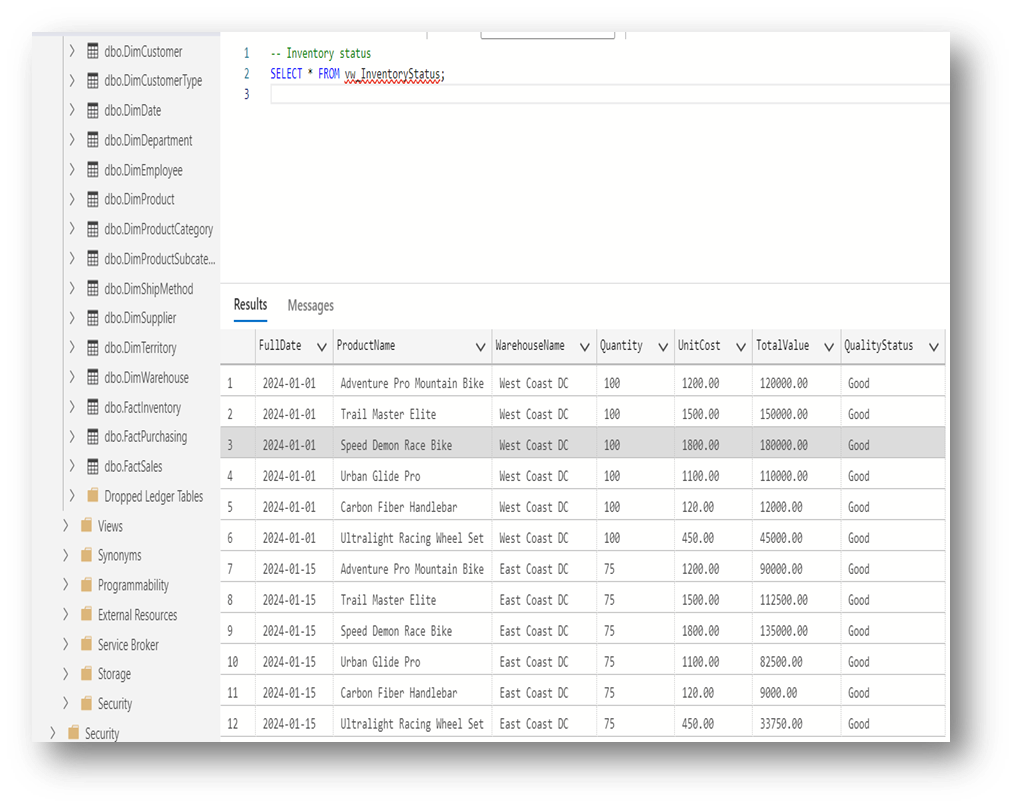
**Fact Table Analysis: Purchasing**

* **Grain:** Individual line item level of a purchase order
* **Measures:** Quantity, UnitCost, TotalCost, Freight, TaxAmount, QualityRating
* **Dimensional References:** References to Supplier, Product, Employee, Warehouse, Date, ShipMethod
* **Transaction Tracking:** PurchaseOrderNumber, Status, PaymentTerms
* **Special Attributes:** IsUrgent flag for priority tracking

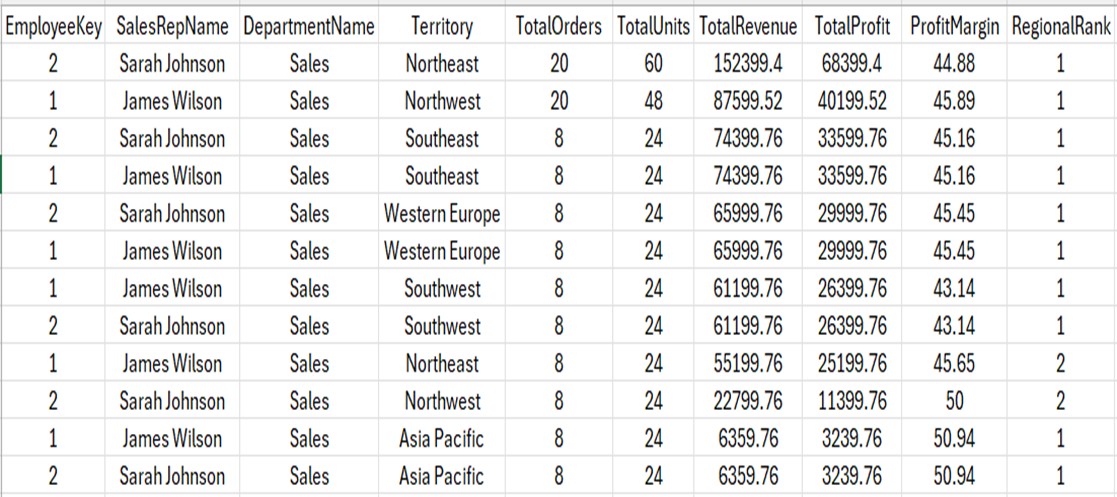


**Fact Table Analysis: Inventory**

* **Grain:** Daily snapshot of product inventory by warehouse
* **Measures:** Quantity, UnitCost, TotalValue, StockOutDuration
* **Tracking Elements:** TransactionType, BatchNumber, ShelfLocation
* **Inventory Controls:** ReorderPoint, MaximumStock, QualityStatus
* **Date References:** Transaction date and Expiry date tracking



**Total Sales Representative Analysis**



The analysis highlights the performance metrics of the top sales representatives across various territories. The dataset provides a comprehensive view of sales activity, enabling data-driven insights for informed decision-making and performance benchmarking.

**Grain**

The dataset captures a detailed snapshot of sales representatives' performance segmented by territory and time. Each record represents a unique combination of employee, region, and performance metrics.

**Key Measures**

* **Total Orders:** The total number of sales orders processed by a sales representative in each territory.
* **Total Units:** The total quantity of units sold within the respective region.
* **Total Revenue:** The revenue generated from the sales activities in each territory.
* **Total Profit:** The profit earned from sales activities after accounting for costs.
* **Profit Margin:** A measure of profitability, calculated as the percentage of profit relative to the revenue.

**Tracking Elements**

* **Territory:** Provides a regional breakdown of sales performance, covering regions such as Northeast, Northwest, Southeast, Western Europe, Southwest, and Asia Pacific.
* **Regional Rank:** Indicates the ranking of each sales representative based on their performance within the respective territory.

**Inventory-Like Controls for Sales**

* **Performance Benchmarks:** Regions with higher TotalRevenue and TotalProfit set the benchmarks for other territories. Territories like Northeast and Northwest exhibit strong sales metrics, indicating their importance in the sales strategy.
* **Profitability Management:** Territories with higher ProfitMargin, such as Asia Pacific and Northeast, are critical for maintaining profitability. Focus on these regions can maximize revenue-to-profit conversion rates.

**Observations and Insights**

1. **Top Performers:**
   * **Sarah Johnson:**
     + Achieved the highest Total Revenue of $152,399.4 in the Northeast, with a Profit Margin of 44.88%.
     + Displayed consistent performance across other territories such as Southeast, Western Europe, and Southwest. o **James Wilson:**
     + Excelled in the Northwest and Northeast territories with Total Revenue figures of $87,599.52 and $55,199.76, respectively.
     + Recorded the highest Profit Margin of 50.94% in the Asia Pacific region, despite lower Total Revenue.
2. **Regional Performance:**
   * **High-Performing Regions:** Territories such as Northeast and Northwest are notable for generating high revenue and profit figures.
   * **Profitability Focus:** Asia Pacific demonstrated the highest Profit Margin (50.94%), indicating a significant potential for profitability despite lower Total Orders and Total Revenue.
3. **Actionable Insights:**

**Boost Sales in High-Margin Territories:** Strategies to expand sales efforts in territories like Asia Pacific can enhance overall profitability.

* + **Address Underperforming Regions:** Focus on improving sales in regions with lower Total Revenue, such as Western Europe and Southwest, to balance performance across all territories.

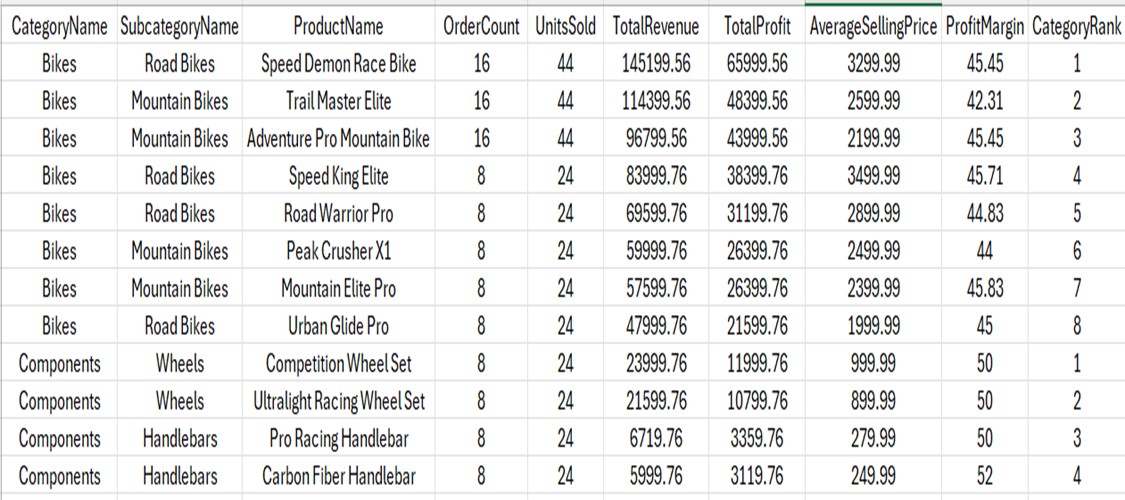
**Date Tracking Suggestions**

To enrich the analysis, integrating date references such as transaction dates and quarterly or monthly performance snapshots will enable trend analysis and provide insights into seasonal or periodic fluctuations in sales performance.

## Summary

The analysis identifies Sarah Johnson and James Wilson as the top-performing sales representatives, excelling across multiple regions. The Northeast and Northwest territories emerge as key revenue drivers, while Asia Pacific demonstrates the highest profitability. These insights provide a solid foundation for targeted strategies to enhance sales performance and profitability across all territories. A strategic focus on high-margin regions and improvement in underperforming territories will contribute to achieving overall sales excellence.

**Product category performance**



This analysis provides insights into the performance of various product categories, subcategories, and individual products. The data highlights key sales and profitability metrics, offering a detailed view of which product lines are driving revenue and profit.

**Grain**

The dataset is organized by product categories and subcategories, with a focus on individual products' performance metrics. Each record represents a unique product within its respective category and subcategory.

**Key Measures**

* **Order Count:** The total number of orders placed for each product.
* **Units Sold:** The total quantity of units sold for each product.
* **Total Revenue:** The revenue generated from the sales of each product.
* **Total Profit:** The profit earned from each product’s sales.
* **Average Selling Price:** The average price at which the product was sold.
* **Profit Margin:** The percentage of profit relative to the revenue.
* **Category Rank:** The ranking of products within their category based on sales or profitability metrics.

**Tracking Elements**

* **Category Name:** Groups products into high-level categories (e.g., Bikes, Components).
* **Subcategory Name:** Provides a breakdown of product types within each category (e.g., Road Bikes, Mountain Bikes, Wheels, Handlebars).

**Inventory-Like Controls for Products**

* **Performance Benchmarks:** Products with higher Total Revenue and Total Profit set the benchmark for their respective categories. For example, the "Speed Demon Race Bike" and "Competition Wheel Set" are standout performers in their categories.
* **Profitability Targets:** Products like "Ultralight Racing Wheel Set" and "Carbon Fiber Handlebar" exhibit high Profit Margins (50% and 52%, respectively), highlighting their profitability potential.

**Observations and Insights**

1. **Category Performance:**
   * **Bikes:** Dominates revenue generation with products like "Speed Demon Race

Bike" contributing $145,199.56 in Total Revenue and $65,999.56 in Total Profit. The Road Bikes and Mountain Bikes subcategories are particularly strong performers. o **Components:** While generating less revenue than Bikes, products in this category (e.g., "Ultralight Racing Wheel Set") achieve the highest Profit Margins, showcasing their profitability.

1. **Product Performance:**
   * **Top Performers:**
     + "Speed Demon Race Bike" is the leading product in terms of both Total Revenue and Total Profit within the Bikes category.
     + "Competition Wheel Set" and "Ultralight Racing Wheel Set" excel in the Components category, with Profit Margins of 50%.
   * **High Profit Margin Products:** Products like "Carbon Fiber Handlebar" (52%) and "Ultralight Racing Wheel Set" (50%) emphasize profitability over high revenue.
2. **Subcategory Highlights:**

**Road Bikes:** The leading subcategory, with multiple top-selling products like

"Speed Demon Race Bike" and "Speed King Elite." o **Mountain Bikes:** Consistently performs well, with products such as "Trail Master

Elite" and "Adventure Pro Mountain Bike." o **Wheels and Handlebars (Components):** High-margin subcategories with steady performance and potential for revenue expansion.

1. **Actionable Insights:**
   * **Focus on High-Margin Products:** Expand marketing and sales efforts for highmargin products like "Ultralight Racing Wheel Set" and "Carbon Fiber

Handlebar." o **Boost Sales of Mid-Level Performers:** Products with moderate revenue and profitability, such as "Urban Glide Pro," could benefit from targeted promotions to improve their market share.

* + **Expand High-Performing Subcategories:** Increase the inventory or marketing focus on Road Bikes and Wheels to capture more market demand.

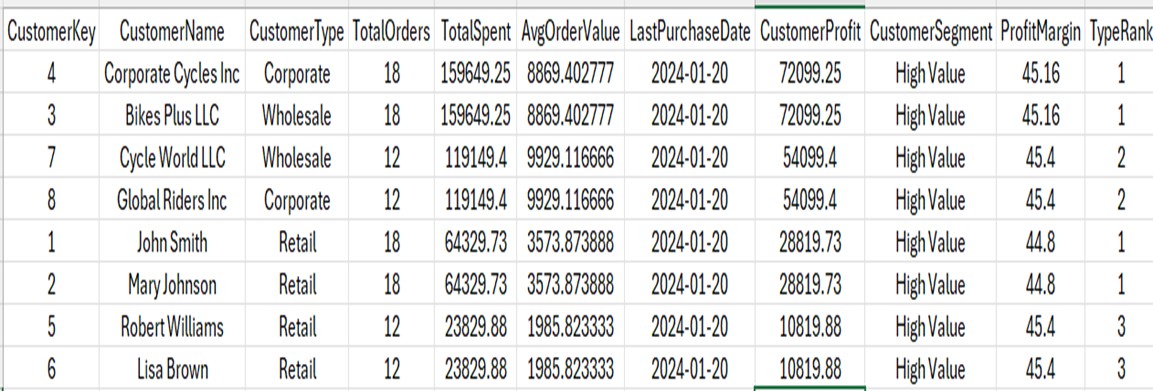
**Date Tracking**

Integrating date references such as sales periods and trends over time would provide insights into product demand fluctuations and seasonal performance.

## Summary

The analysis reveals that the **Bikes** category leads in Total Revenue, with standout products like "Speed Demon Race Bike." The **Components** category demonstrates exceptional profitability, with products like the “Ultralight Racing Wheel Set" achieving a 50% Profit Margin. Highperforming subcategories, such as Road Bikes and Wheels, are crucial for revenue and profit growth. To maximize profitability, the focus should shift to promoting high-margin products and scaling mid-performing products to improve their market presence.

**Customer Segmentation Analysis**



This analysis categorizes customers based on their spending patterns, profit contribution, and order history. The dataset enables the identification of high-value customers and provides actionable insights for improving customer relationship management and revenue generation.

**Grain**

The dataset represents customer-level segmentation based on sales activity, profit contribution, and average order value. Each row corresponds to a unique customer.

**Key Measures**

* **Total Orders:** The total number of orders placed by the customer.
* **Total Spent:** The total amount spent by the customer across all orders.
* **Avg Order Value:** The average value of each order, calculated as Total Spent divided by Total Orders.
* **Customer Profit:** The total profit generated from a customer’s purchases.
* **Profit Margin:** The percentage of profit relative to Total Spent.
* **Type Rank:** The rank of customers within each type (Corporate, Wholesale, Retail) based on their contribution to profit or spending.

**Tracking Elements**

* **Customer Type:** Categorizes customers as Corporate, Wholesale, or Retail based on their purchasing behavior.
* **Customer Segment:** Segments customers into High-Value or other tiers based on Total Spent and Customer Profit.
* **Last Purchase Date:** Tracks the most recent transaction date for each customer.

**Insights and Observations**

1. **Customer Segmentation:**
   * **High-Value Customers:** All customers in the dataset fall under the High-Value segment, highlighting their significant contribution to revenue and profit.
   * **Top Segments by Type:**
     + **Corporate:** Customers like "Corporate Cycles Inc" and "Global Riders Inc" are significant contributors, with Total Spent figures of $159,649.25 and $119,149.4, respectively.
     + **Wholesale:** Customers such as "Bikes Plus LLC" and "Cycle World LLC" exhibit high spending and profit, with Avg Order Values exceeding $8,800.
     + **Retail:** Individual customers like "John Smith" and "Mary Johnson" show moderate spending levels, with Total Spent values of $64,239.73 each.
2. **Profitability Analysis:**
   * Customers such as "Corporate Cycles Inc" and "Bikes Plus LLC" have the highest Customer Profit of $72,099.25 each, with a Profit Margin of 45.16%.

Retail customers like "Robert Williams" and "Lisa Brown" generate lower overall profit ($10,819.88 each) but maintain consistent Profit Margins of 45.4%.

1. **Spending Patterns:**
   * **Corporate and Wholesale Customers:** Exhibit significantly higher Avg Order Values (over $8,800) compared to Retail customers, whose AvgOrderValues range from $1,985 to $3,573. o **Retail Customers:** Show a higher frequency of smaller transactions, indicating potential for growth through targeted upselling and cross-selling strategies.
2. **Actionable Insights:**
   * **Enhance Corporate Relationships:** Strengthen engagement with top Corporate and Wholesale customers like "Corporate Cycles Inc" and "Bikes Plus LLC" through loyalty programs or tailored offers.
   * **Upsell and Cross-Sell Opportunities:** Focus on Retail customers like "John Smith" and "Mary Johnson" to encourage larger transactions and increase Avg Order Value.
   * **Retention Strategy:** Regularly follow up with high-value customers based on their Last Purchase Date to ensure consistent engagement and repeat purchases.

**Customer Relationship Recommendations**

* **High-Value Focus:** Develop targeted marketing campaigns for High-Value customers to maintain loyalty and maximize their lifetime value.
* **Segmentation-Based Strategies:** Tailor strategies for each Customer Type. For example:

o **Corporate:** Offer bulk discounts or custom solutions. o **Wholesale:** Focus on volume-based incentives. o **Retail:** Introduce referral programs or bundle deals to increase Avg Order Value.

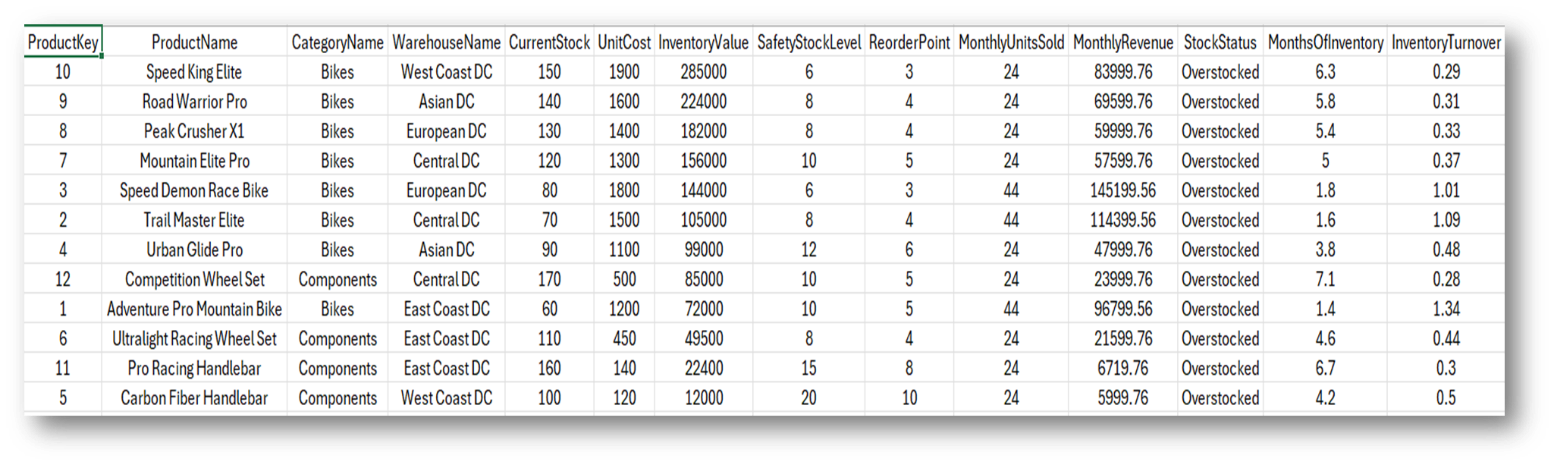
**Date Tracking Suggestions**

Incorporating historical transaction trends and periodic sales data can provide deeper insights into customer behavior and help forecast future spending patterns.

## Summary

The analysis identifies "Corporate Cycles Inc" and "Bikes Plus LLC" as the most valuable customers, contributing the highest profit and revenue. Wholesale and Corporate segments dominate in terms of spending and Avg Order Value, while Retail customers provide opportunities for growth through targeted engagement. Implementing retention strategies and focusing on high-value customers will enhance profitability and strengthen customer relationships across all segments.

**Inventory and Stock Analysis**



This analysis provides insights into the inventory and stock levels of various products across multiple warehouses. The dataset highlights inventory metrics such as current stock levels, safety stock levels, reorder points, and inventory turnover, enabling better inventory management and optimization.

**Grain**

The dataset represents inventory details at the product level across different warehouses. Each row corresponds to a unique product stocked in a specific location.

**Key Measures**

* **Current Stock:** The current quantity of the product available in the warehouse.
* **Unit Cost:** The cost per unit of the product.
* **Inventory Value:** The total value of the current stock calculated as Current Stock multiplied by Unit Cost.
* **Safety Stock Level:** The minimum stock level required to avoid stockouts.
* **Reorder Point:** The stock level at which replenishment orders should be placed.
* **Monthly Units Sold:** The total number of units sold per month.
* **Monthly Revenue:** The revenue generated by selling the product monthly.
* **Inventory Turnover:** A measure of how quickly inventory is sold and replaced, calculated as Monthly Units Sold divided by Current Stock.
* **Months Of Inventory:** Indicates how long the current stock will last, calculated as Current Stock divided by Monthly Units Sold.

**Tracking Elements**

* **ProductName and Category Name:** Identify the product and its respective category (e.g., Bikes, Components).
* **Warehouse Name:** Tracks the warehouse location where the product is stocked.
* **Stock Status:** Flags inventory as "Overstocked" if stock levels significantly exceed demand.

**Insights and Observations**

1. **Stock Levels and Inventory Value:**
   * The **"Speed King Elite"** in the West Coast DC has the highest Current Stock (150 units) with an Inventory Value of $285,000. This product also generates high Monthly Revenue ($83,999.76). o The **"Pro Racing Handlebar"** has the lowest Inventory Value of $22,400 at East Coast DC due to lower Unit Cost and Current Stock.
2. **Safety Stock and Reorder Points:**
   * Products such as **"Ultralight Racing Wheel Set"** and **"Carbon Fiber Handlebar"** have higher Safety Stock Level and Reorder Point thresholds, reflecting their higher variability in demand and the need for consistent replenishment.
   * **"Speed Demon Race Bike"** has a Safety Stock Level of 6 units and a Reorder Point of 3, indicating high turnover and frequent replenishment needs.
3. **Inventory Turnover and Overstocking:**
   * **High Turnover Products:** Products like **"Pro Racing Handlebar"** and **"Urban Glide Pro"** have high Inventory Turnover rates (1.34 and 1.09, respectively), indicating efficient inventory management. o **Overstocked Products:** Most products are flagged as "Overstocked," with Months of Inventory exceeding optimal thresholds (e.g., "Competition Wheel Set" with 7.1 months and **"Peak Crusher X1"** with 5.4 months).
4. **Warehouse-Specific Performance:**
   * **West Coast DC:** Stocks high-value products like "Speed King Elite" but shows a longer Months of Inventory (6.3 months) compared to other regions.
   * **East Coast DC:** Handles low-cost components like **"Carbon Fiber Handlebar"**, reflecting lower Inventory Value but efficient turnover.

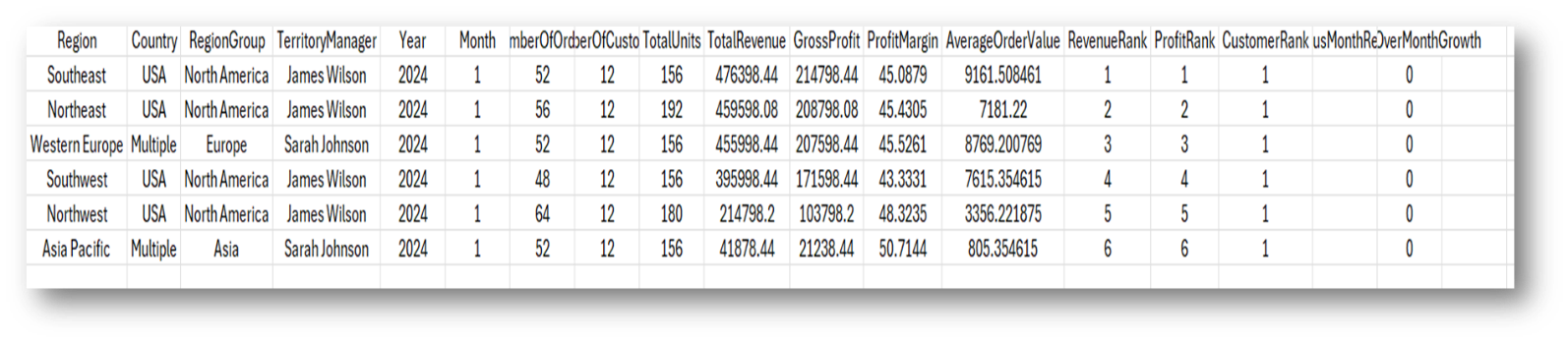
**Actionable Insights**

1. **Reduce Overstocking:** Focus on products with excessive Months of Inventory, such as **"Competition Wheel Set"** and **"Peak Crusher X1"**, to free up warehouse space and reduce holding costs.
2. **Optimize Replenishment Cycles:** Adjust reorder points for products like **"Speed Demon Race Bike"** and **"Urban Glide Pro"** to match their high turnover rates and minimize stockouts.
3. **Warehouse-Level Adjustments:** Review inventory levels at warehouses like West Coast DC to balance stock across locations and reduce overstocking in high-value products.
4. **Increase Inventory Efficiency:** Promote high-turnover products like **the “Pro Racing Handlebar"** to maintain consistent sales momentum and reduce holding periods

**Summary**

The analysis highlights significant overstocking in most products, with inventory levels exceeding demand in multiple warehouses. High-value products like **"Speed King Elite"** and **"Speed Demon Race Bike"** drive Monthly Revenue, while low-cost items like **"Pro Racing Handlebar"** exhibit efficient turnover. Implementing strategies to reduce overstocking, optimize reorder points, and balance inventory across warehouses will improve overall inventory management and operational efficiency.

**Regional Sales Performance**



This analysis focuses on the sales performance across various regions managed by different

Territory Managers. Key metrics such as Total Revenue, Gross Profit, Profit Margin, and Average Order Value are used to evaluate regional performance. The data also includes rankings by revenue, profit, and customers.

**Grain**

The dataset represents regional sales performance for a specific year (2024) and month (January). Each row corresponds to a unique region's performance summary.

**Key Measures**

* **Total Revenue:** The total revenue generated in the region during the reporting period.
* **Gross Profit:** The total profit earned after subtracting costs from Total Revenue.
* **Profit Margin:** The percentage of Gross Profit relative to Total Revenue.
* **Average Order Value:** The average revenue generated per order.
* **Revenue Rank:** Regional rank based on Total Revenue.
* **Profit Rank:** Regional rank based on Gross Profit.
* **Customer Rank:** Rank based on the number of customers.
* **Month Revenue Month Growth:** Tracks month-over-month revenue growth for each region.

**Tracking Elements**

* **Region:** Represents the geographical area.
* **Country and Region Group:** Categorizes regions into broader geographical groups (e.g., North America, Europe, Asia).
* **Territory Manager:** Identifies the manager responsible for sales in each region.

**Insights and Observations**

1. **Top-Performing Regions:**
   * **Southeast (USA):** Leads in both Total Revenue ($476,398.44) and Gross Profit ($214,798.44), with the highest Average Order Value of $9,161.51. Managed by James Wilson, this region exhibits a Profit Margin of 45.09%.
   * **Northeast (USA):** Ranks second with Total Revenue of $459,598.08 and Gross Profit of $208,798.08. The Profit Margin of 45.43% is slightly higher than Southeast.
2. **Moderate Performers:**
   * **Western Europe (Multiple):** Generates a Total Revenue of $455,998.44 and a Gross Profit of $207,598.44. Managed by Sarah Johnson, this region has the second-highest Average Order Value ($8,769.20) but a slightly lower Profit Margin (45.53%). o **Southwest (USA):** Achieves $395,998.44 in Total Revenue with a Gross Profit of $171,598.44. It has the lowest Profit Margin among the top four regions (43.33%).
3. **Underperforming Regions:**
   * **Northwest (USA):** Generates Total Revenue of $214,798.12 and a Gross Profit of $103,798.2, with the lowest Average Order Value of $3,356.22. This indicates low-value transactions. o **Asia Pacific (Multiple):** Shows the lowest Total Revenue ($41,878.44) but the highest Profit Margin (50.71%). Managed by Sarah Johnson, this region has high profitability despite low revenue.
4. **Profitability Insights:**
   * Asia Pacific stands out with the highest Profit Margin, indicating cost-effective operations or premium pricing. o Regions like Southeast and Northeast maintain a balance of high revenue and profitability, making them vital for overall growth.
5. **Territory Management Performance:**
   * **James Wilson:** Manages North America (Southeast, Northeast, Southwest,

Northwest) and oversees the highest-revenue regions (Southeast and Northeast).

* + **Sarah Johnson:** Manages Western Europe and Asia Pacific, with a focus on high-margin operations, as seen in Asia Pacific.

**Actionable Insights**

1. **Growth Potential in Asia Pacific:**
   * Despite being the lowest in revenue, Asia Pacific has significant profitability. Expanding operations or increasing transaction volumes could drive revenue growth while maintaining margins.
2. **Focus on Low-Value Transactions in Northwest:**
   * The Northwest region requires strategies to increase Average Order Value, such as targeted marketing campaigns or product bundling.
3. **Sustain Momentum in High-Revenue Regions:**
   * Strengthen sales strategies in Southeast and Northeast, leveraging their high revenue and profitability to sustain their leadership positions.
4. **Enhance Performance in Western Europe:**
   * Western Europe has strong revenue but slightly lower Profit Margin. Cost optimization or pricing strategies could improve overall profitability.

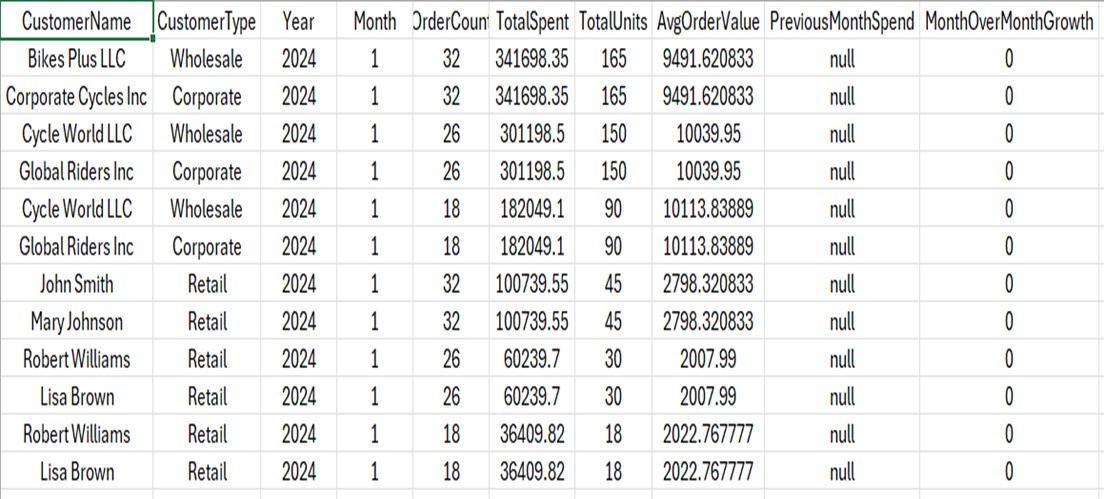
**Summary**

The Southeast region leads in revenue and profit, followed closely by Northeast and Western

Europe. Asia Pacific emerges as the most profitable region, while Northwest lags in Average

Order Value and overall performance. James Wilson oversees the top-performing regions, while Sarah Johnson manages high-margin operations. To maximize growth, focus on expanding profitable regions like Asia Pacific and improving transaction value in underperforming regions like Northwest.

## Customer Purchase Pattern Analysis for Report



This analysis provides insights into the purchase behavior of customers, focusing on key metrics such as Order Count, Total Spent, and Average Order Value. It helps identify trends in customer spending and highlights the performance of different customer types (Corporate, Wholesale, Retail).

**Grain**

The dataset represents customer purchase patterns for January 2024. Each row corresponds to a unique customer's purchase data during this period.

**Key Measures**

* **Order Count:** Total number of orders placed by a customer during the month.
* **Total Spent:** Total amount spent by the customer across all orders.
* **Total Units:** Total number of units purchased.
* **Avg Order Value:** Average value of each order, calculated as Total Spent divided by Order Count.
* **Previous Month Spend:** The customer's spending in the previous month (not available for January 2024 in this dataset).
* **Month Over Month Growth:** Percentage change in spending compared to the previous month (set to 0 due to missing data for previous months).

**Tracking Elements**

* **Customer Name and Customer Type:** Identifies individual customers and categorizes them into Corporate, Wholesale, or Retail types.
* **Year and Month:** Represents the time of the purchases.

**Insights and Observations**

1. **Customer Segments Performance:**
   * **Corporate Customers:**
     + "Corporate Cycles Inc" and "Global Riders Inc" are top spenders, each contributing $341,698.35 and $301,198.5, respectively, with high Avg Order Values of $9,491.62 and $10,039.95. o **Wholesale Customers:**
     + "Bikes Plus LLC" leads the Wholesale category, with Total Spent of $341,698.35 and an Avg Order Value of $9,491.62.
     + "Cycle World LLC" demonstrates consistent spending with $301,198.5, indicating significant purchasing power. o **Retail Customers:**
     + "John Smith" and "Mary Johnson" are the top Retail spenders, each contributing $100,739.55, with Avg Order Values of $2,798.32.
     + "Robert Williams" and "Lisa Brown" exhibit smaller purchase volumes, with Total Spent of $60,239.7 and $36,409.82, respectively.
2. **Average Order Value Trends:**
   * Wholesale customers have the highest Avg Order Value, exceeding $10,000 in some cases, indicating bulk purchases. o Retail customers have significantly lower Avg Order Values, averaging between $2,000 and $2,800, reflecting smaller, more frequent transactions.
3. **Order Volume Analysis:**
   * Corporate and Wholesale customers place fewer but higher-value orders, typically around 26–32 orders per month.
   * Retail customers place smaller orders, averaging 18–32 orders.
4. **Customer Loyalty Indicators:**
   * Repeat customers like "Cycle World LLC" and "Global Riders Inc" demonstrate consistent purchase behavior, indicative of strong customer loyalty.
5. **Month Over Month Growth:**
   * Growth metrics are currently unavailable due to the lack of Previous Month Spend data. However, tracking these metrics in subsequent months will provide valuable insights into customer spending trends over time.

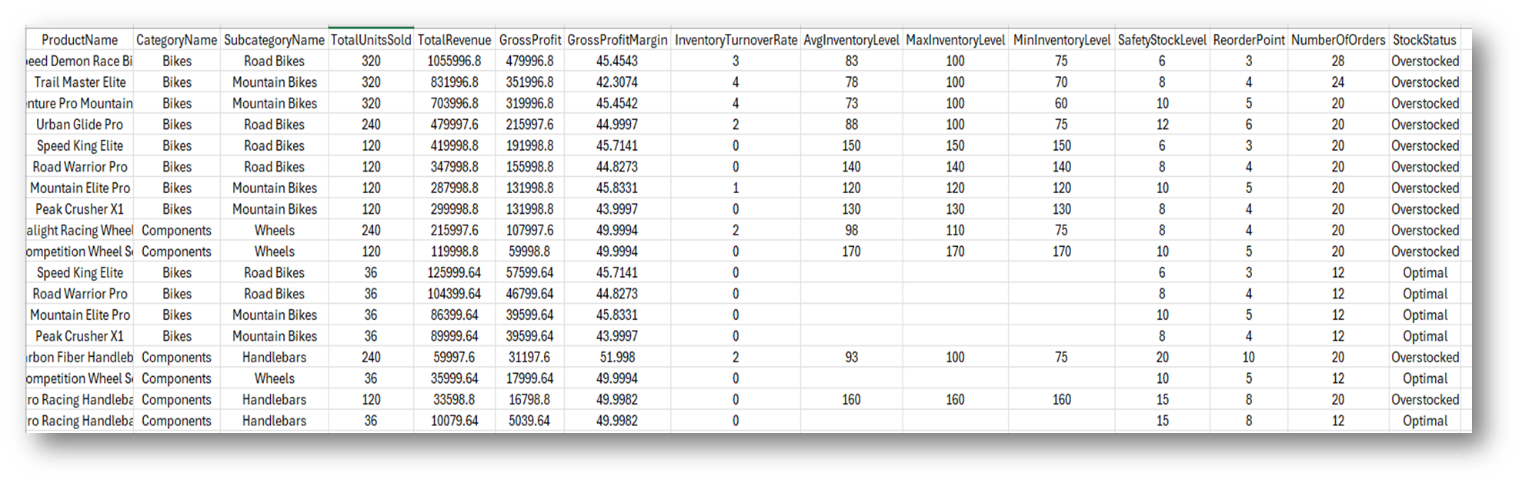
**Actionable Insights**

1. **Focus on High-Spending Segments:**
   * Develop loyalty programs or exclusive offers for high-spending Corporate and Wholesale customers like "Corporate Cycles Inc" and "Cycle World LLC."
2. **Increase Retail Customer Spending:**
   * Introduce promotions or bundled deals to encourage higher Avg Order Values for Retail customers like "Robert Williams" and "Lisa Brown."
3. **Track Month Over Month Trends:**
   * Capture Previous Month Spend data to analyze growth trends and identify seasonal or behavioral patterns in customer purchases.
4. **Enhance Engagement with Loyal Customers:**
   * Strengthen relationships with repeat customers by offering incentives or early access to new products.

**Summary**

Corporate and Wholesale customers dominate spending and Avg Order Value metrics, making them vital for revenue generation. Retail customers present opportunities for growth through targeted promotions. The absence of month-over-month data limits trend analysis but highlights the need for continuous tracking of customer behavior over time. Leveraging these insights will improve customer segmentation strategies and enhance overall revenue performance.

**Product Profitability and Inventory Efficiency**



This analysis evaluates product-level profitability and inventory efficiency across categories and subcategories. Key metrics such as Total Revenue, Gross Profit, Gross Profit Margin, Inventory Turnover Rate, and Stock Status provide insights into product performance and inventory management.

**Grain**

The dataset represents profitability and inventory efficiency at the product level. Each row corresponds to a unique product, detailing its sales performance and stock metrics.

**Key Measures**

* **Total Units Sold:** The total quantity of units sold for the product.
* **Total Revenue:** Revenue generated from the sales of the product.
* **Gross Profit:** The total profit earned after accounting for costs.
* **Gross Profit Margin:** Percentage of profit relative to revenue.
* **Inventory Turnover Rate:** Frequency of inventory replenishment calculated as Total Unit Sold divided by Avg Inventory Level.
* **Avg Inventory Level:** The average stock level maintained for the product.
* **Safety Stock Level:** Minimum stock required to prevent stockouts.
* **Reorder Point:** Inventory level at which replenishment should be triggered.
* **Number of Orders:** Total number of orders processed for the product.
* **Stock Status:** Indicates whether the product is "Overstocked" or "Optimal."

**Insights and Observations**

1. **Top-Performing Products:**
   * **"Speed Demon Race Bike"** leads in Total Revenue ($1,055,996.8) and Gross Profit ($479,996.8) with a Gross Profit Margin of 45.45%. However, its Inventory Turnover Rate is only 3, indicating slower movement despite high revenue.
   * **"Trail Master Elite"** and **"Adventure Pro Mountain Bike"** also demonstrate strong revenue and profit, with Gross Profit Margins of 42.31% and 45.45%, respectively.
2. **High-Margin Products:**
   * **"Ultralight Racing Wheel Set"** and **"Carbon Fiber Handlebar"** in the Components category achieve Gross Profit Margins of 49.99% and 51.99%, respectively, reflecting their profitability.
   * Both products have efficient Inventory TurnoverRates of 2 and are flagged as "Optimal" in Stock Status.
3. **Underperforming Products:**
   * **"Peak Crusher X1"** and **"Mountain Elite Pro"** generate moderate revenue and profit but have lower Inventory TurnoverRates (1). Both products are flagged as "Overstocked."
4. **Inventory Efficiency:**
   * **Optimal Stock Levels:** Products like **"Competition Wheel Set"** and **"Ultralight Racing Wheel Set"** maintain efficient stock levels with high profitability.
   * **Overstocking:** Most products in the Bikes category, such as **"Speed King Elite"** and **"Road Warrior Pro,"** are overstocked, indicating potential for inventory optimization.
5. **Category and Subcategory Trends:**
   * **Bikes:** Dominates in Total Revenue and Gross Profit but struggles with overstocking and low turnover rates. The Road Bikes and Mountain Bikes subcategories show consistent performance. o **Components:** Balances profitability and inventory efficiency, with products achieving high Gross Profit Margins and optimal stock levels.

**Actionable Insights**

1. **Optimize Overstocked Products:**
   * Reassess inventory levels for products like **"Speed Demon Race Bike"** and **"Trail Master Elite"** to align with demand, reducing holding costs.
2. **Promote High-Margin Products:**
   * Increase focus on products like **"Ultralight Racing Wheel Set"** and **"Carbon Fiber Handlebar"** to capitalize on their profitability.
3. **Improve Turnover Rates:**
   * Implement marketing campaigns or bundling strategies to accelerate sales for slow-moving products like **"Peak Crusher X1"** and **"Mountain Elite Pro."**
4. **Enhance Replenishment Strategies:**
   * Adjust reorder points and safety stock levels for high-demand products to prevent overstocking and ensure steady supply.

**Summary**

The analysis identifies **"Speed Demon Race Bike"** as the highest revenue generator, while Components products such as **"Ultralight Racing Wheel Set"** lead in profitability and inventory efficiency. Overstocking in the Bikes category highlights the need for inventory optimization. By promoting high-margin products, improving turnover rates, and streamlining replenishment strategies, the overall profitability and efficiency of the product portfolio can be enhanced.

**Dimension Tables and Hierarchies**

* **Product Hierarchy:** 
  + Product -> Subcategory -> Category o Product details include cost, price, specifications o Category and Subcategory include descriptions and groupings
* **Customer Hierarchy:** 
  + Customer -> CustomerType
  + Customer includes contact and location details o CustomerType defines discounts and payment terms
* **Geographic Hierarchy:** 
  + Territory: Group -> Division -> Region -> Country o Includes sales targets and cost centers o Managed by territory managers
* **Employee Hierarchy:** 
  + Employee -> Department o Employee includes contact and role information o Department includes budget and management structure

**Key Design Considerations**

* **Surrogate Keys:** 
  + Used in all dimension tables (e.g., ProductKey, CustomerKey) o Auto-incrementing integers for optimal performance
  + Maintains referential integrity
* **Natural Keys:** 
  + Preserved in dimension tables (e.g., ProductID, CustomerID) o Enables business user recognition o Supports data integration processes
* **Slowly Changing Dimensions (SCDs):** 
  + **Type 1: Simple updates for corrections** 
    - Customer contact information
    - Product descriptions o **Type 2: Historical tracking for**
    - Product prices and costs
    - Customer addresses
    - Employee roles
* **Conformed Dimensions:** 
  + **Date: Used across all fact tables**
  + **Product:** Shared between Sales, Purchasing, and Inventory o **Employee:** Used in both Sales and Purchasing o **Location:** Standardized across warehouse and territory

**Operational Considerations**

* **Performance:** 
  + Indexed foreign keys in fact tables o Partitioning strategy for large fact tables o Optimized date dimension joins

* **Data Quality:** 
  + Referential integrity enforced o Business rules for measures o Status tracking across transactions
* **Integration:** 
  + Regular dimension updates o Transaction processing schedule o History preservation strategy

# Challenges

* **Data Complexity and Volume**:
  + Requires careful integration of dimensions like Product, Customer, and Employee into a unified data warehouse.
  + Managing increasing complexity and data volume.
* **Data Quality**:
  + Maintaining accuracy across diverse sources is critical. o Robust validation mechanisms and integrity checks are necessary to avoid discrepancies.
* **Performance Optimization**:
  + Snowflake schema introduces performance challenges.
  + Efficient query execution relies on proper indexing and partitioning for large fact tables.
* **Integration and Compatibility**:
  + Seamless integration of ETL processes and compatibility with BI tools require a strong framework.
* **Historical Data Tracking**:
  + Requires thoughtful schema design for Slowly Changing Dimensions (SCDs).
  + Ensures proper version management of historical changes.
* **Scalability and Flexibility**:
  + Schema must accommodate growth and new requirements.
  + Allows for easy additions of dimensions or fact tables.
* **Operational Challenges**:
  + Managing ETL processes for consistent updates. o Requires ongoing monitoring and maintenance to address operational hurdles.

# Deployment Strategies

* **Incremental Implementation**: o Deploy the warehouse in phases, starting with critical data like sales transactions.
  + Gradually integrate other dimensions to minimize initial complexity and risk.
* **Robust ETL Processes**:
  + Implement automated workflows for efficient data extraction, transformation, and loading.
  + Schedule regular updates to maintain data freshness.
* **Performance Tuning**: o Optimize queries using indexing and partitioning for large fact tables.
  + Refine SQL for frequent reporting needs.
* **Data Quality Assurance**:
  + Enforce constraints and validation rules to maintain high data quality.
  + Conduct regular audits to ensure accuracy and consistency.
* **Scalable Infrastructure**:
  + Use scalable RDBMS solutions, such as SQL Server, to handle growth.
  + Ensure high availability and fault tolerance.
* **Integration with BI Tools**:
  + Connect the warehouse to BI tools for analytics and reporting.
  + Provide user-friendly dashboards for stakeholders.
* **Documentation and Training**:
  + Maintain comprehensive documentation for the data model and workflows.
  + Train users and administrators for smooth operations and troubleshooting.
* **Monitoring and Maintenance**:
  + Deploy tools to monitor performance and integrity. o Conduct periodic reviews to align the system with evolving business needs.

# Considerations and Future Expansion

**Scalability**

* Design the model to accommodate future growth in data volume

**Performance**

* Balance normalization with query performance requirements
* Implement appropriate indexing, especially on foreign key columns in the fact table
* Consider partitioning the Sales fact table for large volumes of data

**Flexibility**

* Allow for potential additions of new data elements or relationships
* The current design supports easy addition of new dimensions or fact tables

**Data Quality**

* Implement checks and constraints to ensure data accuracy and consistency
* Consider implementing a system to track data lineage

**Future Expansion**

* Integration with ETL (Extract, Transform, Load) processes
* Connection to business intelligence and reporting tools
* Potential expansion to include additional data sources or business areas
* Consider adding a Customer dimension for customer-centric analysis
* Implement a full-fledged Date dimension to enhance time-based analysis capabilities

# Conclusion

The implemented data warehouse schema represents a comprehensive and robust solution designed to meet the organization's analytical and reporting needs. Built on a foundation of three interconnected fact tables - Sales, Purchasing, and Inventory - along with well-structured dimension hierarchies, the model achieves an optimal balance between normalization and denormalization. This balance ensures both data integrity and query performance, making it highly effective for complex analytical requirements.

The design's key strength lies in its integrated approach to handling core business processes. Through careful implementation of surrogate and natural keys, thoughtfully structured hierarchies, and conformed dimensions, the warehouse enables seamless analysis across different business aspects. The model efficiently supports various analytical needs, from sales performance tracking and inventory management to customer behavior analysis and supplier performance monitoring, all while maintaining data consistency and accuracy.

This data warehouse implementation establishes a strong foundation for the organization's future growth and analytical capabilities. Its flexible and scalable architecture can accommodate increasing data volumes and evolving business requirements without compromising performance or data integrity. By providing a single, reliable source of truth for business intelligence, the warehouse empowers stakeholders to make data-driven decisions with confidence. As the organization continues to grow, this robust data architecture will remain a crucial asset, driving operational efficiency and strategic decision-making while maintaining its adaptability to future business needs and technological advancements.

**Final Code:**

-- Create Database

CREATE DATABASE RetailDW;

GO

USE RetailDW;

GO

-- Create Dimension Tables

CREATE TABLE DimDate (

DateKey INT PRIMARY KEY,

FullDate DATE,

Year INT,

Quarter INT,

Month INT,

MonthName VARCHAR(10),

DayOfWeek INT,

DayName VARCHAR(10),

FiscalYear INT,

IsHoliday BIT

);

CREATE TABLE DimProductCategory (

CategoryKey INT IDENTITY(1,1) PRIMARY KEY,

CategoryID VARCHAR(10),

CategoryName VARCHAR(50),

Description TEXT,

ImageURL VARCHAR(200)

);

CREATE TABLE DimProductSubcategory (

SubcategoryKey INT IDENTITY(1,1) PRIMARY KEY,

CategoryKey INT FOREIGN KEY REFERENCES DimProductCategory(CategoryKey),

SubcategoryID VARCHAR(10),

SubcategoryName VARCHAR(50),

Description TEXT,

ImageURL VARCHAR(200)

);

CREATE TABLE DimProduct (

ProductKey INT IDENTITY(1,1) PRIMARY KEY,

SubcategoryKey INT FOREIGN KEY REFERENCES

DimProductSubcategory(SubcategoryKey),

ProductID VARCHAR(20),

ProductName VARCHAR(100),

StandardCost DECIMAL(18,2),

ListPrice DECIMAL(18,2),

Color VARCHAR(20),

Size VARCHAR(10),

Weight DECIMAL(8,2),

UnitMeasure VARCHAR(10),

ProductLine VARCHAR(20),

Style VARCHAR(20),

ModelName VARCHAR(50),

Description TEXT,

SafetyStockLevel INT,

ReorderPoint INT,

DaysToManufacture INT,

ProductStatus VARCHAR(20),

IsDiscontinued BIT,

DiscontinuedDate DATE

);

CREATE TABLE DimCustomerType (

CustomerTypeKey INT IDENTITY(1,1) PRIMARY KEY,

TypeName VARCHAR(50),

Description TEXT,

DiscountPercent DECIMAL(5,2),

MinimumOrderAmount DECIMAL(18,2),

PaymentTerms VARCHAR(50)

);

CREATE TABLE DimCustomer (

CustomerKey INT IDENTITY(1,1) PRIMARY KEY,

CustomerTypeKey INT FOREIGN KEY REFERENCES

DimCustomerType(CustomerTypeKey),

CustomerID VARCHAR(20),

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100),

Phone VARCHAR(20),

AddressLine1 VARCHAR(100),

City VARCHAR(50),

StateProvince VARCHAR(50),

PostalCode VARCHAR(20),

Country VARCHAR(50),

CreditLimit DECIMAL(18,2),

AccountOpenedDate DATE,

CustomerStatus VARCHAR(20)

);

-- Insert sample data into DimProductCategory

INSERT INTO DimProductCategory (CategoryID, CategoryName, Description) VALUES

('CAT-001', 'Bikes', 'All types of bicycles'),

('CAT-002', 'Components', 'Bicycle parts and components'),

('CAT-003', 'Clothing', 'Cycling apparel and accessories'),

('CAT-004', 'Accessories', 'Bike accessories and gear');

-- Insert sample data into DimProductSubcategory

INSERT INTO DimProductSubcategory (CategoryKey, SubcategoryID, SubcategoryName,

Description) VALUES

(1, 'SUB-001', 'Mountain Bikes', 'Off-road bicycles'),

(1, 'SUB-002', 'Road Bikes', 'Performance road bicycles'),

(1, 'SUB-003', 'Hybrid Bikes', 'Multi-purpose bicycles'),

(2, 'SUB-004', 'Handlebars', 'Bicycle handlebars'),

(2, 'SUB-005', 'Wheels', 'Bicycle wheels and rims'),

(3, 'SUB-006', 'Jerseys', 'Cycling jerseys'),

(3, 'SUB-007', 'Shorts', 'Cycling shorts'),

(4, 'SUB-008', 'Helmets', 'Protective headgear'),

(4, 'SUB-009', 'Lights', 'Bicycle lighting');

-- Insert sample data into DimProduct (abbreviated sample)

INSERT INTO DimProduct (SubcategoryKey, ProductID, ProductName, StandardCost,

ListPrice, Color, Size, Weight, UnitMeasure, ProductLine, Style, ModelName, SafetyStockLevel, ReorderPoint, DaysToManufacture, ProductStatus, IsDiscontinued) VALUES

-- Mountain Bikes

(1, 'BK-M001', 'Adventure Pro Mountain Bike', 1200.00, 2199.99, 'Red', 'M', 12.5, 'EA',

'Mountain', 'Pro', 'Adventure 2024', 10, 5, 3, 'Active', 0),

(1, 'BK-M002', 'Trail Master Elite', 1500.00, 2599.99, 'Blue', 'L', 11.8, 'EA', 'Mountain', 'Elite',

'Trail 2024', 8, 4, 4, 'Active', 0),

-- Road Bikes

(2, 'BK-R001', 'Speed Demon Race Bike', 1800.00, 3299.99, 'Black', 'M', 8.2, 'EA', 'Road', 'Race', 'Speed 2024', 6, 3, 5, 'Active', 0),

(2, 'BK-R002', 'Urban Glide Pro', 1100.00, 1999.99, 'White', 'L', 9.0, 'EA', 'Road', 'Sport', 'Glide 2024', 12, 6, 3, 'Active', 0),

-- Components

(4, 'CP-H001', 'Carbon Fiber Handlebar', 120.00, 249.99, 'Black', 'STD', 0.3, 'EA', 'Pro', 'Racing', 'CF-2024', 20, 10, 1, 'Active', 0),

(5, 'CP-W001', 'Ultralight Racing Wheel Set', 450.00, 899.99, 'Black', '700c', 1.6, 'SET', 'Elite',

'Racing', 'UL-2024', 8, 4, 2, 'Active', 0);

-- Insert sample data into DimCustomerType

INSERT INTO DimCustomerType (TypeName, Description, DiscountPercent,

MinimumOrderAmount, PaymentTerms) VALUES

('Retail', 'Individual retail customers', 0.00, 0.00, 'Due on receipt'),

('Wholesale', 'Wholesale business customers', 10.00, 1000.00, 'Net 30'),

('Corporate', 'Large corporate accounts', 15.00, 5000.00, 'Net 45'),

('VIP', 'High-value individual customers', 5.00, 0.00, 'Due on receipt');

-- Insert sample data into DimCustomer

INSERT INTO DimCustomer (CustomerTypeKey, CustomerID, FirstName, LastName, Email,

Phone, AddressLine1, City, StateProvince, PostalCode, Country, CreditLimit,

AccountOpenedDate, CustomerStatus) VALUES

(1, 'CUST-R001', 'John', 'Smith', 'john.smith@email.com', '555-0101', '123 Main St', 'Seattle',

'WA', '98101', 'USA', 1000.00, '2023-01-15', 'Active'),

(1, 'CUST-R002', 'Mary', 'Johnson', 'mary.j@email.com', '555-0102', '456 Oak Ave', 'Portland',

'OR', '97201', 'USA', 1000.00, '2023-02-01', 'Active'),

(2, 'CUST-W001', 'Bikes Plus', 'LLC', 'orders@bikesplus.com', '555-0201', '789 Business Park',

'Denver', 'CO', '80202', 'USA', 10000.00, '2023-01-01', 'Active'),

(3, 'CUST-C001', 'Corporate Cycles', 'Inc', 'procurement@corpcycles.com', '555-0301', '321

Enterprise Way', 'Chicago', 'IL', '60601', 'USA', 50000.00, '2023-03-01', 'Active');

-- Generate DateDimension for 2024

DECLARE @StartDate DATE = '2024-01-01';

DECLARE @EndDate DATE = '2024-12-31';

WHILE @StartDate <= @EndDate

BEGIN

INSERT INTO DimDate (

DateKey, FullDate,

Year,

Quarter,

Month,

MonthName,

DayOfWeek,

DayName,

FiscalYear,

IsHoliday

)

VALUES (

CONVERT(INT, CONVERT(VARCHAR, @StartDate, 112)),

@StartDate,

YEAR(@StartDate),

DATEPART(QUARTER, @StartDate),

MONTH(@StartDate),

DATENAME(MONTH, @StartDate),

DATEPART(WEEKDAY, @StartDate),

DATENAME(WEEKDAY, @StartDate),

YEAR(@StartDate),

CASE

WHEN (MONTH(@StartDate) = 1 AND DAY(@StartDate) = 1)

OR (MONTH(@StartDate) = 12 AND DAY(@StartDate) = 25) THEN 1

ELSE 0

END

);

SET @StartDate = DATEADD(DAY, 1, @StartDate);

END;

*-- Create remaining dimension tables*

CREATE TABLE DimDepartment (

DepartmentKey INT IDENTITY(1,1) PRIMARY KEY,

DepartmentName VARCHAR(50),

GroupName VARCHAR(50),

CostCenter VARCHAR(20),

BudgetAmount DECIMAL(18,2),

ManagerEmployeeKey INT

);

CREATE TABLE DimEmployee (

EmployeeKey INT IDENTITY(1,1) PRIMARY KEY,

DepartmentKey INT FOREIGN KEY REFERENCES DimDepartment(DepartmentKey),

EmployeeID VARCHAR(20),

FirstName VARCHAR(50),

LastName VARCHAR(50),

Title VARCHAR(50),

UPN VARCHAR(100),

HireDate DATE,

Email VARCHAR(100),

Phone VARCHAR(20),

Address VARCHAR(100),

City VARCHAR(50),

CurrentFlag BIT

);

*-- Add ManagerEmployeeKey foreign key after DimEmployee creation*

ALTER TABLE DimDepartment

ADD CONSTRAINT FK\_Department\_Manager

FOREIGN KEY (ManagerEmployeeKey) REFERENCES DimEmployee(EmployeeKey);

CREATE TABLE DimSupplier (

SupplierKey INT IDENTITY(1,1) PRIMARY KEY,

SupplierID VARCHAR(20),

CompanyName VARCHAR(100),

ContactName VARCHAR(100),

Email VARCHAR(100),

Phone VARCHAR(20),

Address VARCHAR(100),

City VARCHAR(50),

Country VARCHAR(50),

CreditRating INT,

PreferredVendorStatus BIT

);

CREATE TABLE DimWarehouse (

WarehouseKey INT IDENTITY(1,1) PRIMARY KEY,

WarehouseName VARCHAR(50),

Location VARCHAR(50),

Country VARCHAR(50),

Capacity INT,

Address VARCHAR(100),

ManagerEmployeeKey INT FOREIGN KEY REFERENCES DimEmployee(EmployeeKey),

IsAutomated BIT,

TemperatureControlled BIT

);

CREATE TABLE DimTerritory (

SalesTerritoryKey INT IDENTITY(1,1) PRIMARY KEY,

Region VARCHAR(50),

Country VARCHAR(50),

[Group] VARCHAR(50),

Division VARCHAR(50),

ManagerEmployeeKey INT FOREIGN KEY REFERENCES DimEmployee(EmployeeKey),

CostCenter VARCHAR(20),

SalesTarget DECIMAL(18,2)

);

CREATE TABLE DimShipMethod (

ShipMethodKey INT IDENTITY(1,1) PRIMARY KEY,

MethodName VARCHAR(50),

BaseRate DECIMAL(10,2),

EstimatedDays INT,

CarrierName VARCHAR(50),

TrackingAvailable BIT,

MaxWeight DECIMAL(10,2),

ActiveFlag BIT

);

*-- Insert sample data into DimDepartment*

INSERT INTO DimDepartment (DepartmentName, GroupName, CostCenter, BudgetAmount)

VALUES

('Sales', 'Revenue', 'CC-001', 1000000.00),

('Marketing', 'Revenue', 'CC-002', 500000.00),

('Operations', 'Operations', 'CC-003', 750000.00),

('Customer Service', 'Operations', 'CC-004', 300000.00),

('Finance', 'Administrative', 'CC-005', 400000.00),

('Human Resources', 'Administrative', 'CC-006', 250000.00),

('Warehouse', 'Operations', 'CC-007', 600000.00),

('IT', 'Administrative', 'CC-008', 450000.00);

*-- Insert sample data into DimEmployee*

INSERT INTO DimEmployee (DepartmentKey, EmployeeID, FirstName, LastName, Title,

UPN, HireDate, Email, Phone, Address, City, CurrentFlag) VALUES

(1, 'EMP001', 'James', 'Wilson', 'Sales Director', 'james.wilson', '2020-01-15',

'j.wilson@company.com', '555-0001', '123 Business Ave', 'Seattle', 1),

(1, 'EMP002', 'Sarah', 'Johnson', 'Sales Manager', 'sarah.johnson', '2020-02-01',

's.johnson@company.com', '555-0002', '456 Commerce St', 'Portland', 1),

(2, 'EMP003', 'Michael', 'Brown', 'Marketing Director', 'michael.brown', '2020-03-15',

'm.brown@company.com', '555-0003', '789 Market St', 'San Francisco', 1),

(3, 'EMP004', 'Emily', 'Davis', 'Operations Manager', 'emily.davis', '2020-04-01',

'e.davis@company.com', '555-0004', '321 Industrial Way', 'Chicago', 1),

(4, 'EMP005', 'Robert', 'Taylor', 'Customer Service Manager', 'robert.taylor', '2020-05-15',

'r.taylor@company.com', '555-0005', '654 Service Rd', 'Denver', 1),

(5, 'EMP006', 'Lisa', 'Anderson', 'Finance Director', 'lisa.anderson', '2020-06-01',

'l.anderson@company.com', '555-0006', '987 Financial Center', 'New York', 1),

(7, 'EMP007', 'David', 'Martinez', 'Warehouse Manager', 'david.martinez', '2020-07-15',

'd.martinez@company.com', '555-0007', '147 Warehouse Blvd', 'Dallas', 1),

(8, 'EMP008', 'Jennifer', 'Garcia', 'IT Director', 'jennifer.garcia', '2020-08-01',

'j.garcia@company.com', '555-0008', '258 Tech Park', 'Austin', 1);

*-- Update DimDepartment with ManagerEmployeeKey*

UPDATE DimDepartment SET ManagerEmployeeKey = 1 WHERE DepartmentName = 'Sales';

UPDATE DimDepartment SET ManagerEmployeeKey = 3 WHERE DepartmentName =

'Marketing';

UPDATE DimDepartment SET ManagerEmployeeKey = 4 WHERE DepartmentName =

'Operations';

UPDATE DimDepartment SET ManagerEmployeeKey = 5 WHERE DepartmentName =

'Customer Service';

UPDATE DimDepartment SET ManagerEmployeeKey = 6 WHERE DepartmentName =

'Finance';

UPDATE DimDepartment SET ManagerEmployeeKey = 7 WHERE DepartmentName = 'Warehouse';

UPDATE DimDepartment SET ManagerEmployeeKey = 8 WHERE DepartmentName = 'IT';

*-- Insert sample data into DimSupplier*

INSERT INTO DimSupplier (SupplierID, CompanyName, ContactName, Email, Phone,

Address, City, Country, CreditRating, PreferredVendorStatus) VALUES

('SUP001', 'Global Bikes Parts Co.', 'John Smith', 'j.smith@globalbikes.com', '555-1001', '123 Supplier Way', 'Detroit', 'USA', 5, 1),

('SUP002', 'Elite Components Ltd.', 'Mary Johnson', 'm.johnson@elitecomp.com', '555-1002',

'456 Parts Ave', 'Chicago', 'USA', 4, 1),

('SUP003', 'Asian Bicycle Supply', 'David Lee', 'd.lee@asianbike.com', '555-1003', '789

Manufacturing Rd', 'Shanghai', 'China', 4, 1),

('SUP004', 'European Cycling Gear', 'Hans Mueller', 'h.mueller@eurocycle.com', '555-1004', '321

Bike Street', 'Hamburg', 'Germany', 5, 1),

('SUP005', 'Pacific Rim Supplies', 'James Wong', 'j.wong@pacificrim.com', '555-1005', '654

Trading Center', 'Singapore', 'Singapore', 3, 0);

*-- Insert sample data into DimWarehouse*

INSERT INTO DimWarehouse (WarehouseName, Location, Country, Capacity, Address,

ManagerEmployeeKey, IsAutomated, TemperatureControlled) VALUES

('West Coast DC', 'Seattle', 'USA', 100000, '123 Harbor Way', 7, 1, 1),

('East Coast DC', 'New Jersey', 'USA', 120000, '456 Port Street', 7, 1, 1),

('Central DC', 'Dallas', 'USA', 150000, '789 Logistics Park', 7, 1, 1),

('European DC', 'Rotterdam', 'Netherlands', 80000, '321 Euro Port', 7, 1, 1),

('Asian DC', 'Singapore', 'Singapore', 90000, '654 Trade Center', 7, 1, 1);

*-- Insert sample data into DimTerritory*

INSERT INTO DimTerritory (Region, Country, [Group], Division, ManagerEmployeeKey,

CostCenter, SalesTarget) VALUES

('Northwest', 'USA', 'North America', 'Domestic', 1, 'CC-S001', 2000000.00), ('Northeast', 'USA', 'North America', 'Domestic', 1, 'CC-S002', 2500000.00),

('Southwest', 'USA', 'North America', 'Domestic', 1, 'CC-S003', 1800000.00),

('Southeast', 'USA', 'North America', 'Domestic', 1, 'CC-S004', 2200000.00),

('Western Europe', 'Multiple', 'Europe', 'International', 2, 'CC-S005', 3000000.00),

('Asia Pacific', 'Multiple', 'Asia', 'International', 2, 'CC-S006', 2800000.00);

*-- Insert sample data into DimShipMethod*

INSERT INTO DimShipMethod (MethodName, BaseRate, EstimatedDays, CarrierName,

TrackingAvailable, MaxWeight, ActiveFlag) VALUES

('Standard Ground', 9.99, 5, 'Global Shipping Co.', 1, 150.00, 1),

('Express Ground', 14.99, 3, 'Global Shipping Co.', 1, 150.00, 1),

('Next Day Air', 29.99, 1, 'Global Shipping Co.', 1, 50.00, 1),

('International Economy', 49.99, 10, 'World Logistics', 1, 200.00, 1),

('International Express', 79.99, 5, 'World Logistics', 1, 150.00, 1),

('Local Delivery', 19.99, 1, 'City Courier', 1, 100.00, 1);

-- Create Fact Tables

CREATE TABLE FactSales (

SalesKey INT IDENTITY(1,1) PRIMARY KEY,

OrderDateKey INT FOREIGN KEY REFERENCES DimDate(DateKey),

DueDateKey INT FOREIGN KEY REFERENCES DimDate(DateKey),

ShipDateKey INT FOREIGN KEY REFERENCES DimDate(DateKey),

CustomerKey INT FOREIGN KEY REFERENCES DimCustomer(CustomerKey),

EmployeeKey INT FOREIGN KEY REFERENCES DimEmployee(EmployeeKey), ProductKey INT FOREIGN KEY REFERENCES DimProduct(ProductKey),

SalesTerritoryKey INT FOREIGN KEY REFERENCES DimTerritory(SalesTerritoryKey), ShipMethodKey INT FOREIGN KEY REFERENCES DimShipMethod(ShipMethodKey),

Quantity INT,

UnitPrice DECIMAL(18,2),

UnitCost DECIMAL(18,2),

TotalSales DECIMAL(18,2),

Cost DECIMAL(18,2),

Discount DECIMAL(18,2),

TaxAmount DECIMAL(18,2),

Freight DECIMAL(18,2),

SalesOrderNumber VARCHAR(20),

PONumber VARCHAR(20),

PaymentMethod VARCHAR(20),

Status VARCHAR(20)

);

CREATE TABLE FactPurchasing (

PurchaseKey INT IDENTITY(1,1) PRIMARY KEY,

OrderDateKey INT FOREIGN KEY REFERENCES DimDate(DateKey),

DueDateKey INT FOREIGN KEY REFERENCES DimDate(DateKey),

ReceivedDateKey INT FOREIGN KEY REFERENCES DimDate(DateKey),

SupplierKey INT FOREIGN KEY REFERENCES DimSupplier(SupplierKey),

ProductKey INT FOREIGN KEY REFERENCES DimProduct(ProductKey),

EmployeeKey INT FOREIGN KEY REFERENCES DimEmployee(EmployeeKey),

ShipMethodKey INT FOREIGN KEY REFERENCES DimShipMethod(ShipMethodKey),

WarehouseKey INT FOREIGN KEY REFERENCES DimWarehouse(WarehouseKey),

Quantity INT,

UnitCost DECIMAL(18,2),

TotalCost DECIMAL(18,2),

Freight DECIMAL(18,2),

TaxAmount DECIMAL(18,2),

PurchaseOrderNumber VARCHAR(20),

Status VARCHAR(20),

PaymentTerms VARCHAR(50),

QualityRating INT,

IsUrgent BIT

);

CREATE TABLE FactInventory (

InventoryKey INT IDENTITY(1,1) PRIMARY KEY,

DateKey INT FOREIGN KEY REFERENCES DimDate(DateKey),

ProductKey INT FOREIGN KEY REFERENCES DimProduct(ProductKey),

WarehouseKey INT FOREIGN KEY REFERENCES DimWarehouse(WarehouseKey),

Quantity INT,

UnitCost DECIMAL(18,2),

TotalValue DECIMAL(18,2),

TransactionType VARCHAR(20),

ReferenceNumber VARCHAR(20),

BatchNumber VARCHAR(20),

ExpiryDateKey INT FOREIGN KEY REFERENCES DimDate(DateKey),

ShelfLocation VARCHAR(20),

ReorderPoint INT,

MaximumStock INT,

StockOutDuration INT,

QualityStatus VARCHAR(20)

);

-- Insert sample data into FactSales

INSERT INTO FactSales (

OrderDateKey, DueDateKey, ShipDateKey, CustomerKey, EmployeeKey,

ProductKey, SalesTerritoryKey, ShipMethodKey, Quantity, UnitPrice,

UnitCost, TotalSales, Cost, Discount, TaxAmount, Freight,

SalesOrderNumber, PONumber, PaymentMethod, Status

)

SELECT

-- Date keys for January 2024

20240101, -- Order Date

20240105, -- Due Date

20240103, -- Ship Date

CustomerKey,

1, -- Employee Key

ProductKey,

1, -- Territory Key

1, -- Ship Method Key

2, -- Quantity

ListPrice,

StandardCost,

ListPrice \* 2, -- Total Sales

StandardCost \* 2, -- Cost

0.00, -- Discount

(ListPrice \* 2) \* 0.08, -- Tax Amount

15.99, -- Freight

'SO' + RIGHT('00000' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey) AS

VARCHAR), 5),

'PO' + RIGHT('00000' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey) AS

VARCHAR), 5),

'Credit Card',

'Completed'

FROM DimProduct p

CROSS JOIN DimCustomer c

WHERE p.ProductKey <= 3;

-- Additional sales records for different dates

INSERT INTO FactSales (

OrderDateKey, DueDateKey, ShipDateKey, CustomerKey, EmployeeKey,

ProductKey, SalesTerritoryKey, ShipMethodKey, Quantity, UnitPrice,

UnitCost, TotalSales, Cost, Discount, TaxAmount, Freight,

SalesOrderNumber, PONumber, PaymentMethod, Status

)

SELECT

20240115, -- Order Date

20240120, -- Due Date

20240117, -- Ship Date

CustomerKey,

2, -- Employee Key

ProductKey,

2, -- Territory Key

2, -- Ship Method Key

3, -- Quantity

ListPrice,

StandardCost,

ListPrice \* 3, -- Total Sales

StandardCost \* 3, -- Cost

ListPrice \* 0.05, -- Discount

(ListPrice \* 3) \* 0.08, -- Tax Amount

19.99, -- Freight

'SO' + RIGHT('00000' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey) + 100

AS VARCHAR), 5),

'PO' + RIGHT('00000' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey) + 100

AS VARCHAR), 5),

'Wire Transfer',

'Completed'

FROM DimProduct p

CROSS JOIN DimCustomer c

WHERE p.ProductKey <= 3;

-- Insert sample data into FactPurchasing

INSERT INTO FactPurchasing (

OrderDateKey, DueDateKey, ReceivedDateKey, SupplierKey, ProductKey,

EmployeeKey, ShipMethodKey, WarehouseKey, Quantity, UnitCost,

TotalCost, Freight, TaxAmount, PurchaseOrderNumber, Status,

PaymentTerms, QualityRating, IsUrgent

)

SELECT

20240105, -- Order Date

20240115, -- Due Date

20240112, -- Received Date

s.SupplierKey,

p.ProductKey,

4, -- Employee Key (Operations Manager)

1, -- Ship Method Key

1, -- Warehouse Key

50, -- Quantity

p.StandardCost,

p.StandardCost \* 50, -- Total Cost

99.99, -- Freight

(p.StandardCost \* 50) \* 0.08, -- Tax Amount

'PO' + RIGHT('00000' + CAST(ROW\_NUMBER() OVER (ORDER BY p.ProductKey) AS

VARCHAR), 5),

'Received',

'Net 30',

5,

0

FROM DimProduct p

CROSS JOIN DimSupplier s

WHERE p.ProductKey <= 3;

-- Insert sample data into FactInventory

INSERT INTO FactInventory (

DateKey, ProductKey, WarehouseKey, Quantity, UnitCost,

TotalValue, TransactionType, ReferenceNumber, BatchNumber,

ExpiryDateKey, ShelfLocation, ReorderPoint, MaximumStock,

StockOutDuration, QualityStatus

)

SELECT

20240101, -- Date

ProductKey,

1, -- Warehouse Key

100, -- Quantity

StandardCost,

StandardCost \* 100, -- Total Value

'Initial Stock',

'INV' + RIGHT('00000' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey) AS VARCHAR), 5),

'BATCH' + RIGHT('00000' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey)

AS VARCHAR), 5),

20241231, -- Expiry Date

'A' + RIGHT('00' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey) AS

VARCHAR), 2),

20,

200,

0,

'Good'

FROM DimProduct

WHERE ProductKey <= 6;

-- Add more inventory transactions

INSERT INTO FactInventory (

DateKey, ProductKey, WarehouseKey, Quantity, UnitCost,

TotalValue, TransactionType, ReferenceNumber, BatchNumber,

ExpiryDateKey, ShelfLocation, ReorderPoint, MaximumStock,

StockOutDuration, QualityStatus

)

SELECT

20240115, -- Date

ProductKey,

2, -- Different Warehouse

75, -- Different Quantity

StandardCost,

StandardCost \* 75, -- Total Value

'Stock Transfer',

'INV' + RIGHT('00000' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey) + 100 AS VARCHAR), 5),

'BATCH' + RIGHT('00000' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey) +

100 AS VARCHAR), 5),

20241231, -- Expiry Date

'B' + RIGHT('00' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey) AS

VARCHAR), 2),

20,

200,

0,

'Good'

FROM DimProduct

WHERE ProductKey <= 6;

*-- Create the view with explicit schema reference*

CREATE VIEW vw\_SalesSummary

AS

SELECT

CONVERT(VARCHAR(10), d.FullDate, 120) AS OrderDate, CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName, p.ProductName,

t.Region,

s.Quantity,

s.UnitPrice,

s.TotalSales,

s.Cost,

(s.TotalSales - s.Cost) AS Profit

FROM dbo.FactSales s

INNER JOIN dbo.DimDate d

ON s.OrderDateKey = d.DateKey

INNER JOIN dbo.DimCustomer c

ON s.CustomerKey = c.CustomerKey

INNER JOIN dbo.DimProduct p

ON s.ProductKey = p.ProductKey

INNER JOIN dbo.DimTerritory t

ON s.SalesTerritoryKey = t.SalesTerritoryKey;

GO

CREATE VIEW vw\_InventoryStatus AS

SELECT

d.FullDate,

p.ProductName,

w.WarehouseName,

i.Quantity,

i.UnitCost,

i.TotalValue,

i.QualityStatus

FROM FactInventory i

JOIN DimDate d ON i.DateKey = d.DateKey

JOIN DimProduct p ON i.ProductKey = p.ProductKey

JOIN DimWarehouse w ON i.WarehouseKey = w.WarehouseKey;

CREATE VIEW vw\_PurchasingSummary AS

SELECT

d.FullDate AS OrderDate,

s.CompanyName AS Supplier,

p.ProductName, fp.Quantity, fp.UnitCost, fp.TotalCost,

fp.Status,

w.WarehouseName FROM FactPurchasing fp

JOIN DimDate d ON fp.OrderDateKey = d.DateKey

JOIN DimSupplier s ON fp.SupplierKey = s.SupplierKey

JOIN DimProduct p ON fp.ProductKey = p.ProductKey

JOIN DimWarehouse w ON fp.WarehouseKey = w.WarehouseKey;

-- Create example indices for performance

CREATE NONCLUSTERED INDEX IX\_FactSales\_OrderDateKey ON

FactSales(OrderDateKey);

CREATE NONCLUSTERED INDEX IX\_FactSales\_ProductKey ON FactSales(ProductKey);

CREATE NONCLUSTERED INDEX IX\_FactSales\_CustomerKey ON FactSales(CustomerKey);

CREATE NONCLUSTERED INDEX IX\_FactInventory\_DateKey ON FactInventory(DateKey);

CREATE NONCLUSTERED INDEX IX\_FactInventory\_ProductKey ON

FactInventory(ProductKey);

CREATE NONCLUSTERED INDEX IX\_FactPurchasing\_OrderDateKey ON

FactPurchasing(OrderDateKey);

CREATE NONCLUSTERED INDEX IX\_FactPurchasing\_ProductKey ON

FactPurchasing(ProductKey);

INSERT INTO DimProduct (SubcategoryKey, ProductID, ProductName, StandardCost,

ListPrice, Color, Size, Weight, UnitMeasure, ProductLine, Style, ModelName, SafetyStockLevel,

ReorderPoint, DaysToManufacture, ProductStatus, IsDiscontinued) VALUES

-- More Mountain Bikes

(1, 'BK-M003', 'Mountain Elite Pro', 1300.00, 2399.99, 'Green', 'L', 12.2, 'EA', 'Mountain', 'Pro', 'Elite 2024', 10, 5, 3, 'Active', 0),

(1, 'BK-M004', 'Peak Crusher X1', 1400.00, 2499.99, 'Black', 'M', 11.5, 'EA', 'Mountain', 'Pro', 'Crusher 2024', 8, 4, 4, 'Active', 0),

-- More Road Bikes

(2, 'BK-R003', 'Road Warrior Pro', 1600.00, 2899.99, 'Silver', 'L', 8.5, 'EA', 'Road', 'Pro', 'Warrior 2024', 8, 4, 4, 'Active', 0),

(2, 'BK-R004', 'Speed King Elite', 1900.00, 3499.99, 'Red', 'M', 7.8, 'EA', 'Road', 'Elite',

'SpeedKing 2024', 6, 3, 5, 'Active', 0),

-- More Components

(4, 'CP-H002', 'Pro Racing Handlebar', 140.00, 279.99, 'Silver', 'STD', 0.35, 'EA', 'Pro', 'Racing', 'PR-2024', 15, 8, 1, 'Active', 0),

(5, 'CP-W002', 'Competition Wheel Set', 500.00, 999.99, 'Black', '700c', 1.8, 'SET', 'Pro', 'Racing', 'CW-2024', 10, 5, 2, 'Active', 0);

INSERT INTO DimCustomer (CustomerTypeKey, CustomerID, FirstName, LastName, Email,

Phone, AddressLine1, City, StateProvince, PostalCode, Country, CreditLimit,

AccountOpenedDate, CustomerStatus) VALUES

(1, 'CUST-R003', 'Robert', 'Williams', 'r.williams@email.com', '555-0103', '789 Pine St', 'San

Francisco', 'CA', '94101', 'USA', 1000.00, '2023-03-01', 'Active'),

(1, 'CUST-R004', 'Lisa', 'Brown', 'l.brown@email.com', '555-0104', '321 Cedar Ave', 'Los

Angeles', 'CA', '90001', 'USA', 1000.00, '2023-03-15', 'Active'),

(2, 'CUST-W002', 'Cycle World', 'LLC', 'orders@cycleworld.com', '555-0202', '456 Market St',

'Boston', 'MA', '02101', 'USA', 15000.00, '2023-02-01', 'Active'),

(3, 'CUST-C002', 'Global Riders', 'Inc', 'orders@globalriders.com', '555-0302', '789 Corporate Blvd', 'Houston', 'TX', '77001', 'USA', 40000.00, '2023-04-01', 'Active');

INSERT INTO FactSales (

OrderDateKey, DueDateKey, ShipDateKey, CustomerKey, EmployeeKey,

ProductKey, SalesTerritoryKey, ShipMethodKey, Quantity, UnitPrice,

UnitCost, TotalSales, Cost, Discount, TaxAmount, Freight,

SalesOrderNumber, PONumber, PaymentMethod, Status

)

SELECT

20240120, -- Order Date

20240125, -- Due Date

20240122, -- Ship Date

CustomerKey,

(CustomerKey % 2) + 1, -- Alternate between employee 1 and 2

ProductKey,

(ProductKey % 6) + 1, -- Distribute across territories

(CustomerKey % 3) + 1, -- Alternate between shipping methods

CASE WHEN CustomerTypeKey IN (2,3) THEN 5 ELSE 1 END, -- Higher quantity for business customers

ListPrice,

StandardCost,

ListPrice \* CASE WHEN CustomerTypeKey IN (2,3) THEN 5 ELSE 1 END, -- Total Sales StandardCost \* CASE WHEN CustomerTypeKey IN (2,3) THEN 5 ELSE 1 END, -- Cost CASE WHEN CustomerTypeKey IN (2,3) THEN ListPrice \* 0.1 ELSE 0 END, -- Discount for business customers

(ListPrice \* CASE WHEN CustomerTypeKey IN (2,3) THEN 5 ELSE 1 END) \* 0.08, -- Tax

Amount

24.99, -- Freight

'SO' + RIGHT('00000' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey) + 200

AS VARCHAR), 5),

'PO' + RIGHT('00000' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey) + 200

AS VARCHAR), 5),

CASE CustomerTypeKey

WHEN 1 THEN 'Credit Card'

WHEN 2 THEN 'Net 30'

ELSE 'Wire Transfer'

END,

'Completed'

FROM DimProduct p

CROSS JOIN DimCustomer c;

-- Add more inventory transactions

INSERT INTO FactInventory (

DateKey, ProductKey, WarehouseKey, Quantity, UnitCost,

TotalValue, TransactionType, ReferenceNumber, BatchNumber,

ExpiryDateKey, ShelfLocation, ReorderPoint, MaximumStock,

StockOutDuration, QualityStatus

)

SELECT

20240120, -- Date

ProductKey,

(ProductKey % 5) + 1, -- Distribute across warehouses

50 + (ProductKey \* 10), -- Varying quantities

StandardCost,

StandardCost \* (50 + (ProductKey \* 10)), -- Total Value

'Stock Update',

'INV' + RIGHT('00000' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey) + 200

AS VARCHAR), 5),

'BATCH' + RIGHT('00000' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey) +

200 AS VARCHAR), 5),

20241231, -- Expiry Date

'C' + RIGHT('00' + CAST(ROW\_NUMBER() OVER (ORDER BY ProductKey) AS

VARCHAR), 2),

20,

200,

0,

'Good'

FROM DimProduct

WHERE ProductKey <= 12;

WITH CustomerYearlyStats AS (

SELECT

c.CustomerKey,

c.FirstName + ' ' + c.LastName AS CustomerName, ct.TypeName AS CustomerType,

d.Year,

d.Month,

COUNT(DISTINCT s.SalesOrderNumber) AS OrderCount,

SUM(s.TotalSales) AS TotalSpent,

SUM(s.Quantity) AS TotalUnits,

AVG(s.TotalSales) AS AvgOrderValue

FROM FactSales s

JOIN DimCustomer c ON s.CustomerKey = c.CustomerKey

JOIN DimCustomerType ct ON c.CustomerTypeKey = ct.CustomerTypeKey

JOIN DimDate d ON s.OrderDateKey = d.DateKey

JOIN DimProduct p ON s.ProductKey = p.ProductKey

GROUP BY

c.CustomerKey,

c.FirstName + ' ' + c.LastName, ct.TypeName, d.Year,

d.Month

)

SELECT

CustomerName,

CustomerType,

Year,

Month,

OrderCount,

TotalSpent, TotalUnits,

AvgOrderValue,

LAG(TotalSpent) OVER (PARTITION BY CustomerKey ORDER BY Year, Month) AS

PreviousMonthSpend,

CASE

WHEN LAG(TotalSpent) OVER (PARTITION BY CustomerKey ORDER BY Year, Month) > 0

THEN ((TotalSpent - LAG(TotalSpent) OVER (PARTITION BY CustomerKey ORDER BY Year, Month)) /

LAG(TotalSpent) OVER (PARTITION BY CustomerKey ORDER BY Year, Month)) \* 100

ELSE 0

END AS MonthOverMonthGrowth

FROM CustomerYearlyStats

ORDER BY TotalSpent DESC;

WITH TerritorySales AS (

SELECT

t.SalesTerritoryKey,

t.Region,

t.Country,

t.[Group] AS RegionGroup,

e.FirstName + ' ' + e.LastName AS TerritoryManager, d.Year,

d.Month,

COUNT(DISTINCT s.SalesOrderNumber) AS NumberOfOrders,

COUNT(DISTINCT s.CustomerKey) AS NumberOfCustomers,

SUM(s.Quantity) AS TotalUnits,

SUM(s.TotalSales) AS TotalRevenue,

SUM(s.TotalSales - s.Cost) AS GrossProfit,

AVG(s.TotalSales) AS AverageOrderValue

FROM FactSales s

JOIN DimTerritory t ON s.SalesTerritoryKey = t.SalesTerritoryKey

JOIN DimEmployee e ON t.ManagerEmployeeKey = e.EmployeeKey

JOIN DimDate d ON s.OrderDateKey = d.DateKey

GROUP BY

t.SalesTerritoryKey,

t.Region,

t.Country,

t.[Group],

e.FirstName + ' ' + e.LastName,

d.Year,

d.Month

)

SELECT

Region,

Country,

RegionGroup,

TerritoryManager,

Year,

Month,

NumberOfOrders,

NumberOfCustomers,

TotalUnits,

TotalRevenue,

GrossProfit,

CASE

WHEN TotalRevenue > 0

THEN (GrossProfit / TotalRevenue) \* 100

ELSE 0

END AS ProfitMargin,

AverageOrderValue,

DENSE\_RANK() OVER (ORDER BY TotalRevenue DESC) AS RevenueRank,

DENSE\_RANK() OVER (ORDER BY GrossProfit DESC) AS ProfitRank,

DENSE\_RANK() OVER (ORDER BY NumberOfCustomers DESC) AS CustomerRank,

LAG(TotalRevenue) OVER (PARTITION BY Region ORDER BY Year, Month) AS

PreviousMonthRevenue,

CASE

WHEN LAG(TotalRevenue) OVER (PARTITION BY Region ORDER BY Year, Month) >

0

THEN ((TotalRevenue - LAG(TotalRevenue) OVER (PARTITION BY Region ORDER BY Year, Month)) /

LAG(TotalRevenue) OVER (PARTITION BY Region ORDER BY Year, Month)) \* 100

ELSE 0

END AS MonthOverMonthGrowth

FROM TerritorySales

ORDER BY TotalRevenue DESC;

WITH ProductMetrics AS (

SELECT

p.ProductKey,

p.ProductName, pc.CategoryName, ps.SubcategoryName,

SUM(s.Quantity) AS TotalUnitsSold,

SUM(s.TotalSales) AS TotalRevenue,

SUM(s.Cost) AS TotalCost,

AVG(i.Quantity) AS AvgInventoryLevel,

MAX(i.Quantity) AS MaxInventoryLevel, MIN(i.Quantity) AS MinInventoryLevel, p.SafetyStockLevel,

p.ReorderPoint,

COUNT(DISTINCT s.SalesOrderNumber) AS NumberOfOrders

FROM DimProduct p

JOIN DimProductSubcategory ps ON p.SubcategoryKey = ps.SubcategoryKey

JOIN DimProductCategory pc ON ps.CategoryKey = pc.CategoryKey

LEFT JOIN FactSales s ON p.ProductKey = s.ProductKey

LEFT JOIN FactInventory i ON p.ProductKey = i.ProductKey

GROUP BY

p.ProductKey,

p.ProductName, pc.CategoryName, ps.SubcategoryName, p.SafetyStockLevel,

p.ReorderPoint

)

SELECT

ProductName,

CategoryName,

SubcategoryName,

TotalUnitsSold,

TotalRevenue,

TotalRevenue - TotalCost AS GrossProfit,

CASE WHEN TotalRevenue > 0

THEN ((TotalRevenue - TotalCost) / TotalRevenue) \* 100

ELSE 0

END AS GrossProfitMargin,

CASE WHEN AvgInventoryLevel > 0

THEN (TotalUnitsSold / AvgInventoryLevel)

ELSE 0

END AS InventoryTurnoverRate,

AvgInventoryLevel,

MaxInventoryLevel,

MinInventoryLevel,

SafetyStockLevel,

ReorderPoint,

NumberOfOrders,

CASE

WHEN AvgInventoryLevel < ReorderPoint THEN 'Critical Low'

WHEN AvgInventoryLevel < SafetyStockLevel THEN 'Low'

WHEN AvgInventoryLevel > (SafetyStockLevel \* 2) THEN 'Overstocked'

ELSE 'Optimal'

END AS StockStatus

FROM ProductMetrics

ORDER BY GrossProfit DESC;