

Executive Summary

Data Analyst: Kiet Tuan

Purpose: The purpose of this project is to design and develop a comprehensive Sales Performance Dashboard for a fictional company. This project simulates a real-world scenario where a stakeholder has requested an interactive and insightful dashboard that provides an in-depth analysis of the company's sales data. By leveraging data extracted from the company's large data warehouse, this project aims to showcase my ability to perform data wrangling, analysis, and visualization to support informed decision-making.

Project goals:

Create an interactive Sales performance (filter by year/Product category) dashboard that includes the following information:

Calculate metrics:

- Total Sales, order
- Average monthly sales, order
- Freight costs to sales ratio
- Average spending per customer

Create bar graphs showing:

- Total Sales from product category
- Total sales from product subcategory

Create line graphs showing:

- Sales trend (with trend line)
- Accumulate Order count trend

A map shows location of sales.

Scope / Major Project Activities:

| Activity | Description |
|---|--|
| 1. Inspect the dataset and load to SQL database | -Assess the dataset -Choose appropriate tables -Mapping tables |
| 2. Data wrangling | -Data structuring, Data integration, Data Enrichment |
| 3. Data analysis | -Calculate metrics -Apply sales moving average -Calculate accumulate order count |
| 4. Data visualization | -Create an interactive dashboard using Tableau. |

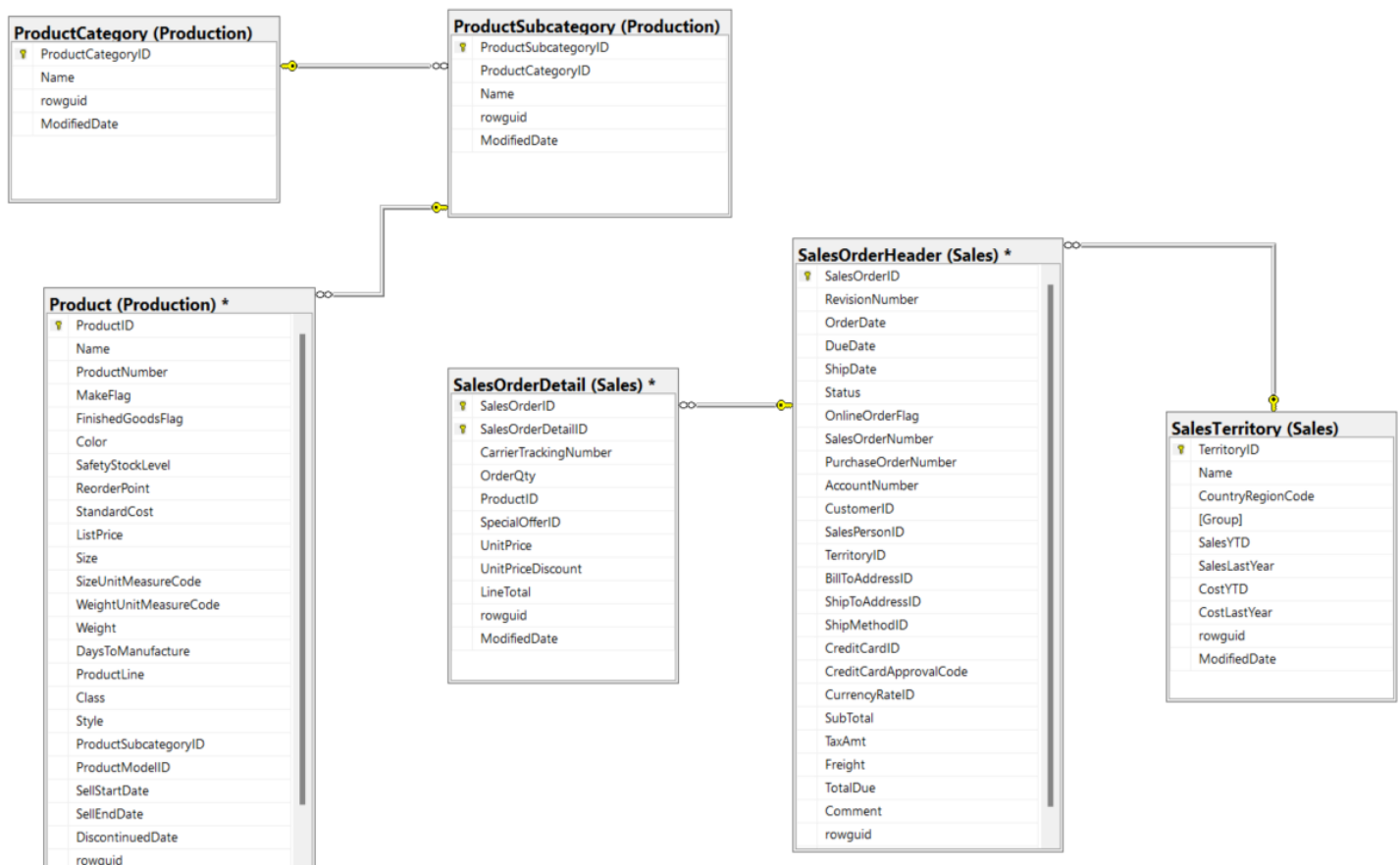
This project does not include: This project is not responsible for collecting primary sources data (surveying, observing, testing, etc.). All the data sources used are secondary datasets.

Data sources used and assessment

In this project I will use the AdventureWorks2019 dataset from Microsoft SQL Server. You can access the data [here](#). This dataset is about a fictional ecommerce company containing various columns such as Sales, IDs, Date, etc. The data is in bak format so I will need to restore and load the bak database into SQL Server studio. Upon assessing, I discover:

- Most of the data is stored in long format
- The data is organized in various schemas (HumanResource, Person, Production, Purchasing, Sales).
- Some columns contain many null values
- The rest of the data is in good format

In this project scope, I have identified the tables will be used and created a relational table diagram in SQL Server.



Data wrangling & Analysis

In the first step, I will prepare the data model that I will need to perform analysis and visualization. The data models will be stored in SQL temp tables for queries optimization as the data are relatively large and I want to break down queries to simple steps.

Data frame for the metrics and the line graphs

```
--Create Sales temp table
CREATE TABLE #Sales
(SalesOrderID int,
OrderDate Date,
OrderCount int,
OrderCountGrowth int,
SubTotal int,
SalesmovingAverage int,
Freight int,
CustomerID int
)

--Insert existing values to #Sales temp table
INSERT INTO #Sales
(
    SalesOrderID,
    OrderDate,
    SubTotal,
    Freight,
    CustomerID
)
SELECT
    SalesOrderID,
    OrderDate,
    SubTotal,
    Freight,
    CustomerID

FROM
    [Sales].[SalesOrderHeader]

SELECT* FROM #Sales
```

```
--Populate OrderQty column in #Sales temp table
UPDATE #Sales
SET OrderCount = X.CountOrder
FROM (
    SELECT
        SalesOrderID,
        COUNT(DISTINCT SalesOrderDetailID ) AS CountOrder
    FROM
        [Sales].[SalesOrderDetail]
    GROUP BY
        SalesOrderID
) X
WHERE #Sales.SalesOrderID = X.SalesOrderID

--Populate OrderCountGrowth accumulate column
UPDATE #Sales
SET OrderCountGrowth = X.AccumulateOrder
FROM
(
    SELECT
        SalesOrderID,
        COUNT(DISTINCT SalesOrderDetailID ) AS CountOrder,
        Sum(COUNT(DISTINCT SalesOrderDetailID )) OVER (ORDER BY SalesOrderID) As AccumulateOrder
    FROM
        [Sales].[SalesOrderDetail]
    GROUP BY SalesOrderID) X
WHERE #Sales.SalesOrderID = X.SalesOrderID

--Populate SalesMovingAverage Column
UPDATE #Sales
SET SalesmovingAverage = X.SalesMovingAverage
FROM
    (SELECT
        SalesOrderID,
        OrderDate,
        SubTotal,
        AVG(SubTotal) OVER (ORDER BY OrderDate ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS
SalesMovingAverage
    FROM
        [Sales].[SalesOrderHeader]) X
WHERE #Sales.SalesOrderID = X.SalesOrderID
SELECT* FROM #Sales
```

The data results should look like this with 31465 rows

| | SalesOrderID | OrderDate | OrderCount | OrderCountGrowth | SubTotal | SalesmovingAverage | Freight | CustomerID |
|---|--------------|------------|------------|------------------|----------|--------------------|---------|------------|
| 1 | 43659 | 2011-05-31 | 12 | 12 | 20566 | 20566 | 616 | 29825 |
| 2 | 43660 | 2011-05-31 | 2 | 14 | 1294 | 10930 | 39 | 29672 |
| 3 | 43661 | 2011-05-31 | 15 | 29 | 32726 | 18195 | 986 | 29734 |
| 4 | 43662 | 2011-05-31 | 22 | 51 | 28833 | 20855 | 867 | 29994 |
| 5 | 43663 | 2011-05-31 | 1 | 52 | 419 | 16768 | 13 | 29565 |
| 6 | 43664 | 2011-05-31 | 8 | 60 | 24433 | 18045 | 733 | 29898 |
| 7 | 43665 | 2011-05-31 | 10 | 70 | 14353 | 17518 | 430 | 29580 |
| 8 | 43666 | 2011-05-31 | 6 | 76 | 5056 | 15302 | 152 | 30052 |
| 9 | 43667 | 2011-05-31 | 4 | 80 | 6107 | 15990 | 183 | 29974 |

Then I copy the data into Excel file for Tableau visualization. The data is stored in Excel_Data.xlsx (Sales tab)

Data frame for Product Category and Subcategory

```
--Create Category_Subcategory temp table
CREATE TABLE #Category_Subcategory
(
    SalesOrderID int,
    OrderDate Date,
    ProductName nvarchar(50),
    ProductSubcategory nvarchar(50),
    ProductCategory nvarchar(50),
    LineTotal int
)

--Insert existing data to #Category_Subcategory temp table
INSERT INTO #Category_Subcategory
(
    SalesOrderID,
    ProductName,
    ProductSubcategory,
    ProductCategory,
    LineTotal
)
SELECT
    SalesOrderID,
    B.Name AS ProductName,
    C.Name AS ProductSubcategory,
    D.Name AS ProductCategory,
    LineTotal
FROM [Sales].[SalesOrderDetail] A INNER JOIN Production.Product B ON
    A.ProductID = B.ProductID INNER JOIN Production.ProductSubcategory C ON
    B.ProductSubcategoryID = C.ProductSubcategoryID INNER JOIN
    Production.ProductCategory D ON C.ProductCategoryID = D.ProductCategoryID

SELECT* FROM #Category_Subcategory
ORDER BY 1

--Populate OrderDate column
UPDATE #Category_Subcategory
SET OrderDate = X.OrderDate
FROM (
    SELECT
        SalesOrderID,
        OrderDate
    FROM
        [Sales].[SalesOrderHeader]) X
WHERE #Category_Subcategory.SalesOrderID = X.SalesOrderID

SELECT* FROM #Category_Subcategory
ORDER BY 1
```

The final result should look like this with 121317 rows. I copy the data to Excel_Data.xlsx, "Category_Subcategory" tab.

| | SalesOrderID | OrderDate | ProductName | ProductSubcategory | ProductCategory | LineTotal |
|----|--------------|------------|-----------------------------|--------------------|-----------------|-----------|
| 1 | 43659 | 2011-05-31 | Mountain-100 Black, 42 | Mountain Bikes | Bikes | 2024 |
| 2 | 43659 | 2011-05-31 | Mountain-100 Black, 44 | Mountain Bikes | Bikes | 6074 |
| 3 | 43659 | 2011-05-31 | Mountain-100 Black, 48 | Mountain Bikes | Bikes | 2024 |
| 4 | 43659 | 2011-05-31 | Mountain-100 Silver, 38 | Mountain Bikes | Bikes | 2039 |
| 5 | 43659 | 2011-05-31 | Mountain-100 Silver, 42 | Mountain Bikes | Bikes | 2039 |
| 6 | 43659 | 2011-05-31 | Mountain-100 Silver, 44 | Mountain Bikes | Bikes | 4079 |
| 7 | 43659 | 2011-05-31 | Mountain-100 Silver, 48 | Mountain Bikes | Bikes | 2039 |
| 8 | 43659 | 2011-05-31 | Long-Sleeve Logo Jersey, M | Jerseys | Clothing | 86 |
| 9 | 43659 | 2011-05-31 | Long-Sleeve Logo Jersey, XL | Jerseys | Clothing | 28 |
| 10 | 43659 | 2011-05-31 | Mountain Bike Socks, M | Socks | Clothing | 34 |

Data frame for Sales location map

```
--Create Sales_location temp table
CREATE TABLE #Sales_Location
(
    SalesOrderID int,
    OrderDate Date,
    SubTotal int,
    RegionName nvarchar(50),
    CountryRegionCode nvarchar(10),
    CountryGroup Nvarchar(50)
);

--Populate existing data to #Sales_location temp table
INSERT INTO #Sales_Location
(
    SalesOrderID,
    OrderDate,
    SubTotal,
    RegionName,
    CountryRegionCode,
    CountryGroup
)

SELECT
    A.SalesOrderID,
    A.OrderDate,
    A.SubTotal,
    B.Name,
    B.CountryRegionCode,
    B.[Group]
FROM
    [Sales].[SalesOrderHeader] A INNER JOIN [Sales].[SalesTerritory] B ON
        A.TerritoryID = B.TerritoryID

SELECT*FROM #Sales_Location
```


The final results should look like this with 31465 rows. Data is stored in Excel_Data.xlsx, "Sales_location" tab



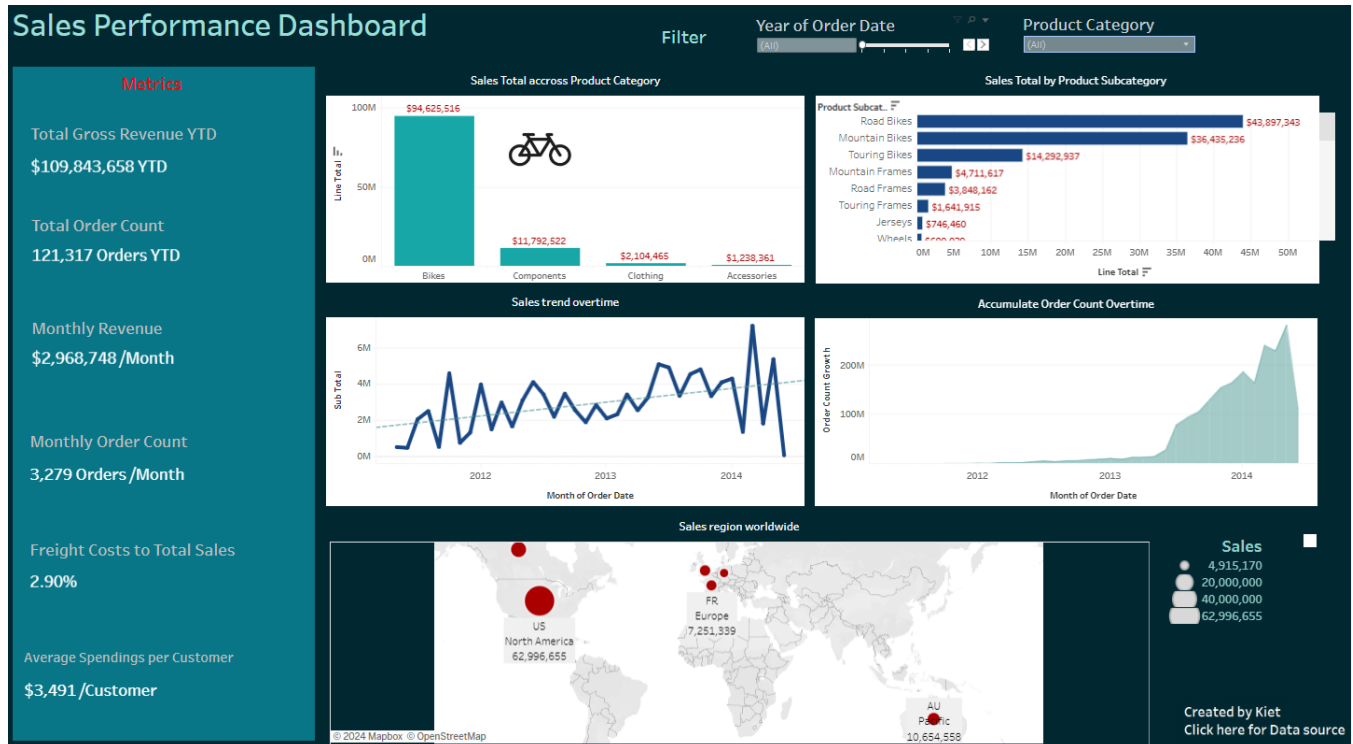
The screenshot shows a SQL query results window with a tab labeled "Results". The table displayed has 7 columns: SalesOrderID, OrderDate, SubTotal, RegionName, CountryRegionCode, and CountryGroup. The first 8 rows of data are shown, with the first row highlighted. The data is as follows:

| | SalesOrderID | OrderDate | SubTotal | RegionName | CountryRegionCode | CountryGroup |
|---|--------------|------------|----------|------------|-------------------|---------------|
| 1 | 43659 | 2011-05-31 | 20566 | Southeast | US | North America |
| 2 | 43660 | 2011-05-31 | 1294 | Southeast | US | North America |
| 3 | 43661 | 2011-05-31 | 32726 | Canada | CA | North America |
| 4 | 43662 | 2011-05-31 | 28833 | Canada | CA | North America |
| 5 | 43663 | 2011-05-31 | 419 | Southwest | US | North America |
| 6 | 43664 | 2011-05-31 | 24433 | Northwest | US | North America |
| 7 | 43665 | 2011-05-31 | 14353 | Northwest | US | North America |
| 8 | 43666 | 2011-05-31 | 5056 | Southwest | US | North America |

The sales metrics will be calculated in Tableau by creating calculated field.

Data Visualization

You can access the Sales performance dashboard [here](#)



Observations:

- The company achieved approximately \$110 million in sales from 2011 to 2014, with a total of 121,317 orders.
- Sales revenue peaked in 2013.
- The most popular products are bikes, with road bikes generating around \$44 million and mountain bikes \$36.5 million.
- In the accessories category, helmets are the top sellers, followed by bike racks and tires and tubes.
- For clothing, jerseys generate the most income at approximately \$746.5k, with shorts in second place at \$411k.
- Mountain frames and road frames are the best sellers in the components category.
- Sales are primarily distributed in North America, with the US being the largest market (approx. \$63 million). The Australia-Pacific region generates around \$10.6 million, followed by key European markets (France, Denmark, Great Britain).