DATAMITES INTERNSHIP

PURCHASE PATTERN ANALYTICS

Market Basket Analysis (MBA) – Company Project Report

PROJECT TEAM ID: **PTID-CDA-DEC-25-1033**

PROJECT ID: **CDACL005-Purchase Pattern Analytics**

PROJECT BY: **PRIYANKA C METI**

DATABASE MANAGEMENT SYSTEM: **SQL SERVER**

VISUALIZATION TOOL: **PYTHON, APRIORI ALGORITHM, POWER BI**

TABLE OF CONTENTS

INTRODUCTION:

In the modern retail environment, understanding customer purchasing behavior is essential for improving sales performance, enhancing customer satisfaction, and optimizing product strategies. Purchase Pattern Analytics enables organizations to analyze transactional data and identify relationships between products purchased together.

This project applies **Market Basket Analysis (MBA)** using **Python, SQL, and Power BI** to uncover hidden product associations, customer behavior patterns, and actionable business insights. The Apriori algorithm is used to identify frequent itemsets and association rules that support cross-selling and sales optimization strategies.

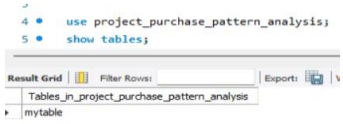
PROBLEM STATEMENT:

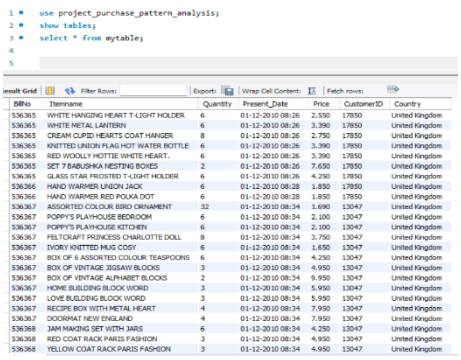
The objective of this project is to leverage Market Basket Analysis on retail transaction data to uncover relationships between products. By performing exploratory data analysis, addressing data quality issues, implementing visualization techniques, and applying the Apriori algorithm, the goal is to extract insights that improve sales strategies and customer satisfaction.

(Note: The SQL queries are mention in the colour – SQL QUERIES)

- **ACCESSED SQL SERVER DATABASE:**

```
use project_purchase_pattern_analysis;
show tables;
select * from mytable;
```





It shows the available database as **'project_ purchase_pattern_analysis'** and the table present within the database as **'mytable'.**

- # Project Overview & Roadmap

SQL-Based Exploratory Data Analysis & Business Queries

Database Initialization and Data Review

The analysis begins by selecting the project database and reviewing the available tables. The transaction table is explored to understand its structure, column names, and data types. This step ensures familiarity with the dataset before performing analytical operations.
A full table preview and schema description are performed to validate data completeness and field definitions.

```
DESCRIBE mytable;
```

It shows the available **fields, datatypes of the fields and also it shows whether there is null values present in the data This query is used to examine the structure of the transaction** This query is used to examine the structure of the transaction table. It provides details about

```
use project_purchase_pattern_analysis;
show tables;
select * from mytable;
DESCRIBE mytable;
```

esult Grid | Filter Rows: | Export: | Wrap Cell Con

Field	Type	Null	Key	Default	Extra
BillNo	varchar(7)	NO		NULL	
Itemname	varchar(36)	YES		NULL	
Quantity	int(11)	YES		NULL	
Present_Date	varchar(16)	YES		NULL	
Price	decimal(9,3)	YES		NULL	
CustomerID	varchar(255)	YES		NULL	
Country	varchar(20)	YES		NULL	

each column, including column names, data types, and whether null values are allowed**.**

Business Purpose:
Understanding the table schema ensures correct interpretation of fields such as transaction ID, product name, quantity, price, customer ID, and date before performing analysis.

Why it is Important:

- **Validates data types for accurate calculations**
- **Helps identify columns that may contain missing values**
- **Supports effective data cleaning and preprocessing**

STEP 1: EXPLORATORY DATA ANALYSIS (EDA)

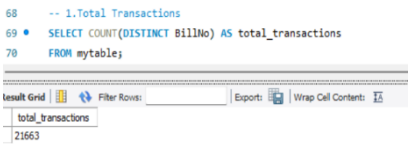
Total Transactions

The total number of unique transactions is calculated using distinct bill numbers.

Purpose:
To understand overall transaction volume and customer purchase activity.

Business Value:
This metric represents how many purchase events occurred and serves as a base KPI for sales analysis.

```
SELECT COUNT (DISTINCT BillNo) AS total_transactions
FROM mytable;
```



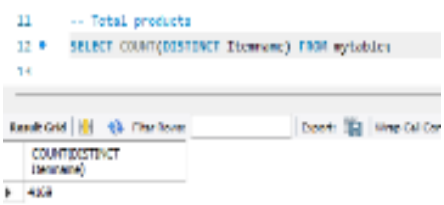
Total Products

The total number of unique products is calculated using distinct item names.

Purpose:
To identify product diversity within the dataset.

Business Value:
Helps assess inventory breadth and product variety offered to customers.

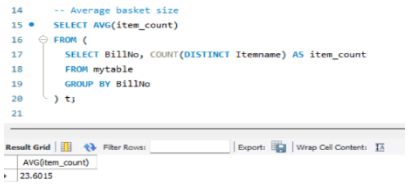
```
SELECT DISTINCT Routes FROM supply_chain_table;
```



Average Basket Size

The average number of distinct products per transaction is calculate

```
SELECT AVG(item_count) FROM
(SELECT BillNo, COUNT (DISTINCT Itemname) AS item_count
 FROM mytable
GROUP BY BillNo
) t;
```



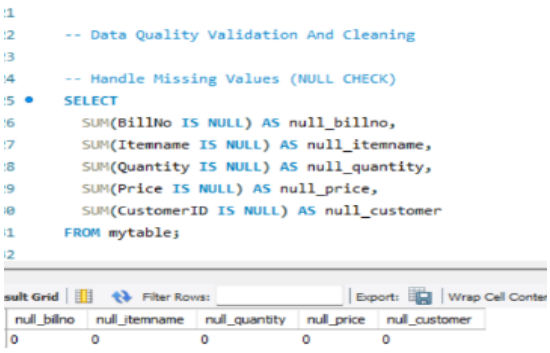
DATA QUALITY VALIDATION AND CLEANING

Before performing business analysis, data quality checks are conducted to ensure accuracy.

Missing Value Check

Null values are identified across critical columns such as Bill Number, Item Name, Quantity, Price, and Customer ID.

```
SELECT
 SUM(BillNo IS NULL) AS null_billno,
 SUM(Itemname IS NULL) AS null_itemname,
 SUM(Quantity IS NULL) AS null_quantity,
 SUM(Price IS NULL) AS null_price,
 SUM(CustomerID IS NULL) AS null_customer
FROM mytable;
```



- **Cancelled Transaction Removal**

  Cancelled invoices (identified by bill numbers starting with “C”) are excluded**.**

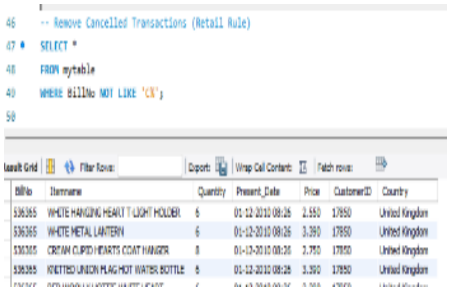
  ## Purpose:

  To remove non-sales transactions from analysis**.**

  ## Business Value:

  Ensures revenue and basket metrics reflect only completed purchases.

```
SELECT  * FROM mytable
WHERE BillNo NOT LIKE 'C%'; distinct;
```



- **Date Cleaning**

  Transaction dates stored as text are converted into proper date format**.**

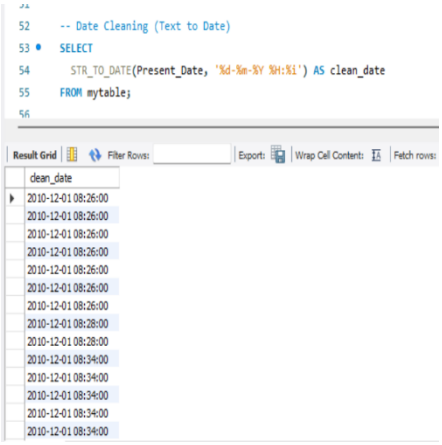
  ## Purpose

  To enable time-based analysis.

  ## Business Value:

  Allows accurate daily, monthly, and trend-based reporting.

```
SELECT
STR_TO_DATE(Present_Date, '%d-%m-%Y %H:%i') AS
clean_date
FROM mytable;
```



- **Final Clean Dataset Creation:**

  final cleaned dataset is created by filtering:
  Positive quantity
  Positive price
  Non-null item names
  Non-cancelled transactions

- **Purpose:**
  To create a trusted analytical dataset.

- **Business Value:**
  Forms a reliable foundation for Power BI dashboards and Python analysis.

```
SELECT DISTINCT
BillNo,
Itemname
FROM mytable
WHERE Quantity > 0
AND Price > 0
AND Itemname IS NOT NULL
AND BillNo NOT LIKE 'C%';
```

```sql
-- FINAL CLEAN DATASET
SELECT DISTINCT
  BillNo,
  Itemname
FROM mytable
WHERE Quantity > 0
  AND Price > 0
  AND Itemname IS NOT NULL
  AND BillNo NOT LIKE 'C%';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

BillNo	Itemname
536365	CREAM CUPID HEARTS COAT HANGER
536365	GLASS STAR FROSTED T-LIGHT HOLDER
536365	KNITTED UNION FLAG HOT WATER BOTTLE
536365	RED WOOLLY HOTTIE WHITE HEART.
536365	SET 7 BABUSHKA NESTING BOXES
536365	WHITE HANGING HEART T-LIGHT HOLDER
536365	WHITE METAL LANTERN
536366	HAND WARMER RED POLKA DOT
536366	HAND WARMER UNION JACK
536367	ASSORTED COLOUR BIRD ORNAMENT
536367	BOX OF 6 ASSORTED COLOUR TEASPOONS
536367	BOX OF VINTAGE ALPHABET BLOCKS
536367	BOX OF VINTAGE JIGSAW BLOCKS
536367	DOORMAT NEW ENGLAND
536367	FELTCRAFT PRINCESS CHARLOTTE DOLL

STEP 2: BUSINESS QUERIES & KEY METRICS

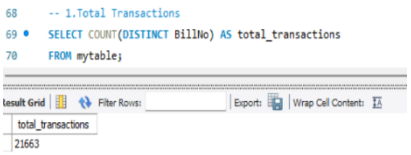
Total Transactions

The final transaction count is recalculated from the clean dataset.

Business Meaning:

Represents actual sales events after data cleansing.

```
SELECT COUNT(DISTINCT BillNo) AS total_transactions
FROM mytable;
```



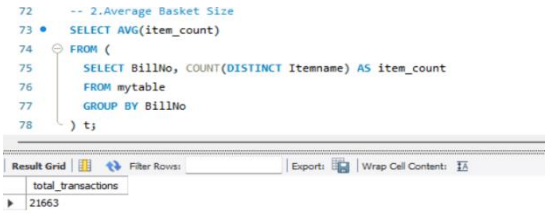
Average Basket Size

The average number of items per transaction is recalculated using only valid records.

Business Meaning:

Indicates customer purchasing depth and cross-sell potential.

```
SELECT AVG(item_count)
FROM (
SELECT BillNo, COUNT(DISTINCT Itemname) AS item_count
FROM mytable
GROUP BY BillNo
) t;
```



Total Revenue

Total revenue is calculated as the sum of quantity multiplied by price.

Business Meaning:

Represents total business sales value.

```
SELECT
SUM(Quantity * Price) AS total_revenue
FROM mytable
WHERE Quantity > 0 AND Price > 0;
```

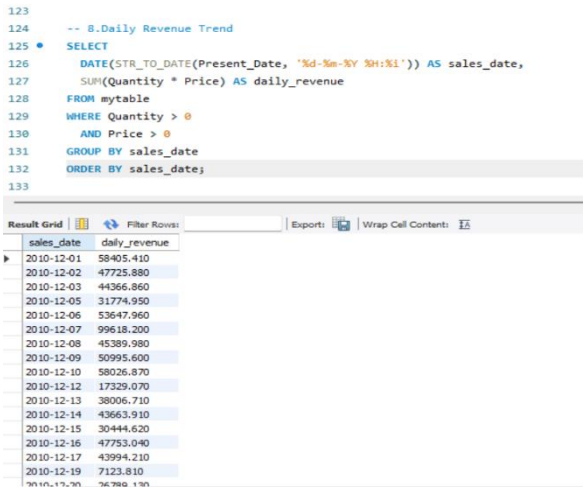
Daily Sales Trend

Revenue is aggregated by transaction date.

Business Meaning:

Helps identify sales patterns, peak days, and demand fluctuations.

```
SELECT
DATE(STR_TO_DATE(Present_Date, '%d-%m-%Y %H:%i')) AS sales_date,
SUM(Quantity * Price) AS daily_revenue
FROM mytable
GROUP BY sales_date;
```



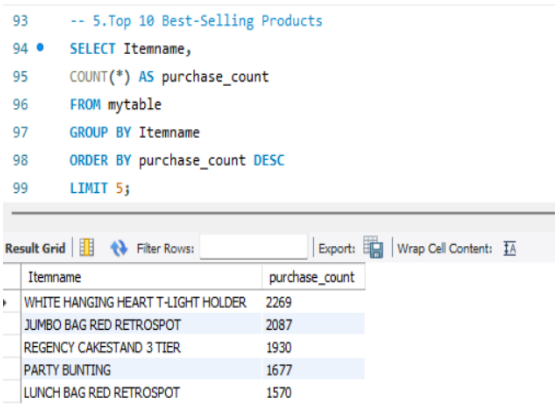
Top 10 Best-Selling Products

Products are ranked based on purchase frequency.

Business Meaning:

Identifies high-demand products and supports inventory and promotion decisions.

```
SELECT Itemname,
COUNT(*) AS purchase_count
FROM mytable
GROUP BY Itemname
ORDER BY purchase_count DESC
LIMIT 5;
```



```
93    -- 5.Top 10 Best-Selling Products
94    SELECT Itemname,
95    COUNT(*) AS purchase_count
96    FROM mytable
97    GROUP BY Itemname
98    ORDER BY purchase_count DESC
99    LIMIT 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Itemname	purchase_count
WHITE HANGING HEART T-LIGHT HOLDER	2269
JUMBO BAG RED RETROSPOT	2087
REGENCY CAKESTAND 3 TIER	1930
PARTY BUNTING	1677
LUNCH BAG RED RETROSPOT	1570

Revenue by Country:

Revenue contribution is analyzed at the country level.

Business Meaning:

Provides geographic insights for regional performance and market analysis.

```
SELECT  Country , SUM(Quantity * Price) AS  country_revenue
FROM mytable
WHERE Quantity > 0
AND Price > 0
GROUP BY Country
ORDER BY country_revenue DESC;
```

```sql
-- 7.Revenue by Country
SELECT
  Country,
  SUM(Quantity * Price) AS country_revenue
FROM mytable
WHERE Quantity > 0
  AND Price > 0
GROUP BY Country
ORDER BY country_revenue DESC;
```

Result Grid | Filter Rows: | Export: | Wrap

Country	country_revenue
United Kingdom	9025222.084
Netherlands	285446.340
Germany	228867.140
France	209715.110
Australia	138521.310
Spain	61577.110
Switzerland	57089.900
Belgium	41196.340
Sweden	38378.330
Japan	37416.370
Norway	36165.440
Portugal	33747.100
Singapore	21279.290
Italy	17483.240
Hong Kong	15691.800
Austria	10198.680
Israel	8135.260

Top Customers by Purchases

Customers are ranked based on purchase frequency within a defined range.

Business Meaning:

Identifies high-value and repeat customers for loyalty and retention strategies.

```sql
SELECT
CustomerID,
COUNT(BillNo) AS total_purchases
FROM mytable
WHERE CustomerID IS NOT NULL
AND CustomerID <> ''
AND BillNo BETWEEN 536000 AND 537000
GROUP BY CustomerID
ORDER BY total_purchases DESC
LIMIT 10;
```

```sql
-- 9.Top Customers by Purchases
SELECT
  CustomerID,
  COUNT(BillNo) AS total_purchases
FROM mytable
WHERE CustomerID IS NOT NULL
  AND CustomerID <> ''
  AND BillNo BETWEEN 536000 AND 537000
GROUP BY CustomerID
ORDER BY total_purchases DESC
LIMIT 10;
```

Result Grid | Filter Rows: | Export: | Wrap Ce

CustomerID	total_purchases
17850	297
15574	121
17841	94
17968	85
17920	81
18041	81
14573	76
15061	73
12433	73
14729	71

Single-Item vs Multi-Item Transactions

Transactions are reviewed to distinguish between single-product and multi-product purchases.

Business Meaning:

Supports Market Basket Analysis by understanding customer buying complexity.

```
SELECT
BillNo,
Itemname
FROM mytable
WHERE Quantity > 0;
```

```
146     -- 10.Single-Item vs Multi-Item Transactions
147 •   SELECT
148       BillNo,
149       Itemname
150     FROM mytable
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

BillNo	Itemname
536365	WHITE HANGING HEART T-LIGHT HOLDER
536365	WHITE METAL LANTERN
536365	CREAM CUPID HEARTS COAT HANGER
536365	KNITTED UNION FLAG HOT WATER BOTTLE
536365	RED WOOLLY HOTTIE WHITE HEART.
536365	SET 7 BABUSHKA NESTING BOXES
536365	GLASS STAR FROSTED T-LIGHT HOLDER
536366	HAND WARMER UNION JACK
536366	HAND WARMER RED POLKA DOT
536367	ASSORTED COLOUR BIRD ORNAMENT
536367	POPPY'S PLAYHOUSE BEDROOM
536367	POPPY'S PLAYHOUSE KITCHEN
536367	FELTCRAFT PRINCESS CHARLOTTE DOLL
536367	IVORY KNITTED MUG COSY
536367	BOX OF 6 ASSORTED COLOUR TEASPOONS
536367	BOX OF VINTAGE JIGSAW BLOCKS
536367	BOX OF VINTAGE ALPHABET BLOCKS
536367	HOME BUILDING BLOCK WORD
536367	LOVE BUILDING BLOCK WORD
536367	RECIPE BOX WITH METAL HEART
536367	DOORMAT NEW ENGLAND
536368	JAM MAKING SET WITH JARS
536368	RED COAT RACK PARIS FASHION
536368	YELLOW COAT RACK PARIS FASHION

mytable 17 ×

Python Analysis & Apriori Algorithm Implementation

After completing SQL-based exploratory analysis and data cleaning, Python was used to perform Market Basket Analysis using the Apriori algorithm. This phase focused on preparing transaction data, identifying product co-occurrence patterns, and generating inputs for association rule mining.

Data Loading and Initial Inspection

The cleaned transaction dataset was loaded into Python using Pandas. Initial inspection was performed to validate the dataset structure, column names, and data types.

Data Type Validation and Column Standardization

Column names were standardized to improve clarity and consistency. Invoice numbers were converted to string format to support transaction-level grouping. Quantity fields were validated to ensure numerical consistency.
This ensured uniform data representation across the dataset and avoided issues during aggregation and transformation**.**

Data Cleaning and Filtering

To maintain analytical accuracy, the following data cleaning steps were applied:

- Records with missing invoice numbers or product descriptions were removed
- Cancelled transactions were excluded
- Only transactions with positive quantity values were retained

These steps ensured that only valid and completed sales transactions were included in the analysis.

One-Hot Encoding of Transactions

The transaction baskets were converted into a binary (one-hot encoded) matrix, where:

- Rows represent individual transactions
- Columns represent products
- Values indicate whether a product was purchased in a transaction

This transformation is a mandatory input requirement for the Apriori algorithm.

Product Frequency Filtering

Product frequency was analyzed to identify commonly purchased items. Products with very low purchase frequency were removed to reduce noise and computational complexity. Only products purchased at least a minimum threshold number of times were retained. This resulted in a reduced and optimized dataset while preserving meaningful purchasing patterns.

After completing SQL-based exploratory analysis and data cleaning, Python was used to perform advanced analytical processing and Market Basket Analysis. The primary objective of this phase was to identify relationships between products purchased together and uncover hidden purchase patterns that cannot be identified using simple descriptive

Purpose of Python Analysis

Python was chosen for this phase due to its strong support for data manipulation and machine learning algorithms. It enables efficient transformation of transaction data into a format suitable for Market Basket Analysis and supports the implementation of the Apriori algorithm.

The Python analysis focused on:

- Preparing transaction-level data for association analysis
- Identifying frequently purchased product combinations
- Generating association rules for cross-selling insights

Data Preparation for Apriori

Before applying the Apriori algorithm, the cleaned transaction dataset was transformed into a basket format. Each transaction was represented as a collection of products purchased together. The data was then converted into a binary matrix, where each row represents a transaction and each column represents a product.
This transformation ensured that the dataset met the input requirements of the Apriori algorithm.

Apriori Algorithm Implementation

The Apriori algorithm was applied to identify frequent itemsets based on product co-occurrence within transactions. The algorithm uses support as a key metric to determine how frequently a product combination appears across all transactions.
After identifying frequent itemsets, association rules were generated using confidence and lift metrics:

- Support measures how often a product combination occurs
- Confidence indicates the likelihood of purchasing one product given another
- Lift evaluates the strength of the association beyond random chance

Output of Apriori Analysis

The Apriori analysis produced:

- A list of frequent itemsets representing commonly purchased product combinations
- Association rules highlighting strong relationships between products

These outputs provide clear insights into customer buying behavior and product affinities**.**

Business Value of Apriori Insights

The results of the Apriori analysis support several business use cases:

- Identification of product bundling opportunities
- Improved cross-selling and upselling strategies

- # IMPORTING DATA IN POWERBI:

After completing SQL-based exploratory analysis and Python-based Market Basket Analysis, Power BI was used to visualize key insights and present business findings in an interactive and user-friendly format. The objective of the Power BI dashboard is to enable stakeholders to quickly understand sales performance, customer behavior, product contribution, and geographic trends.

Purpose of Using Power BI

Power BI was selected as the visualization tool due to its ability to:

- Present complex data insights in an intuitive manner
- Enable interactive filtering and drill-down analysis
- Support data-driven decision-making for business users

The dashboard acts as a decision-support system rather than a technical report.

Data Source for Power BI

The Power BI dashboard was built using the cleaned transaction dataset, which was prepared through:

- SQL-based data validation and cleaning
- Python-based preprocessing for advanced analysis

Only valid and completed transactions were used to ensure accuracy and reliability.

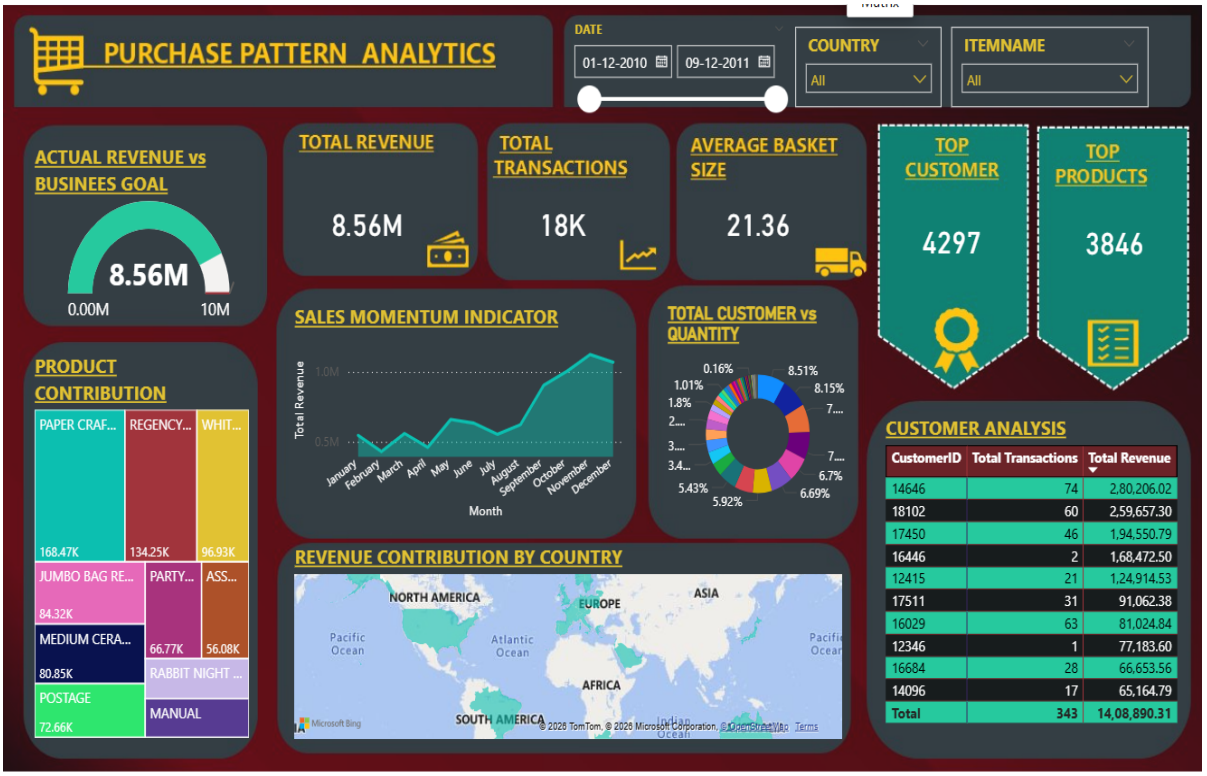
DAX MEASURES:

DAX is called as DATA ANALYSIS EXPRESSIONS and it is used for data analysis and calculations after the data has been loaded into Power BI or Excel. It is used to create measures, calculated columns, and perform complex aggregations.

Exploring New Columns with DAX Measures

To support the insights shown in the dashboard, additional DAX measures were created to calculate key business metrics such as **Total Revenue, Total Transactions, Average Basket Size, Product Contribution, and Customer Performance**.
These measures enable real-time calculations across filters and slicers, ensuring that KPIs, sales trends, product-wise analysis, customer insights, and country-level revenue update dynamically. The use of DAX ensures accuracy, consistency, and clarity in presenting business performance through the dashboard visuals.

- **DASHBOARD**



Dashboard Consists of:

- Displays overall sales performance using **Total Revenue, Total Transactions, and Average Basket Size**
- Compares **Actual Revenue vs Business Goal** to track target achievement
- Shows **monthly sales trends** using the Sales Momentum Indicator
- Highlights **top-performing products** through Product Contribution treemap
- Identifies **Top Customers and Top Products** for focused business strategies
- Analyzes **customer purchase quantity distribution** using donut chart
- Visualizes **country-wise revenue contribution** through interactive map
- Provides **Customer Analysis table** to identify high-value customers
- Includes **Date, Country, and Item Name slicers** for interactive analysis
- Supports data-driven decisions for **sales, marketing, and customer retention**

Insights & Business Recommendations

Key Insights

- Customer purchasing behavior indicates a strong tendency toward multi-item transactions, highlighting opportunities for cross-selling.
- Sales performance demonstrates identifiable trends over time, supporting demand forecasting and planning.
- Revenue is distributed across a broad range of products, indicating a balanced and low-risk product portfolio.
- A segment of repeat customers contributes a significant share of transactions and revenue.
- Market Basket Analysis reveals consistent product pairings, confirming meaningful purchase relationships.
- Geographic analysis highlights both high-performing regions and markets with growth potential.

Business Recommendations

- Introduce bundled offers and promotions based on frequently purchased product combinations.
- Apply targeted cross-sell strategies to increase average basket size and transaction value.
- Align marketing and promotional activities with peak sales periods to maximize impact.
- Develop loyalty programs to retain high-value and repeat customers.
- Use product contribution insights to optimize inventory planning and shelf placement.
- Focus expansion and targeted campaigns on underperforming regions to drive balanced growth.

CONCLUSION:

- Successfully analyzed retail transaction data to understand customer purchasing behavior and sales patterns.
- Ensured data accuracy and reliability through structured exploratory analysis and validation.
- Identified meaningful product associations using Python-based Market Basket Analysis (Apriori).
- Delivered an interactive Power BI dashboard to visualize revenue, transactions, basket size, products, customers, and geography.
- Enabled data-driven decision-making for cross-selling, product bundling, customer targeting, and regional growth.