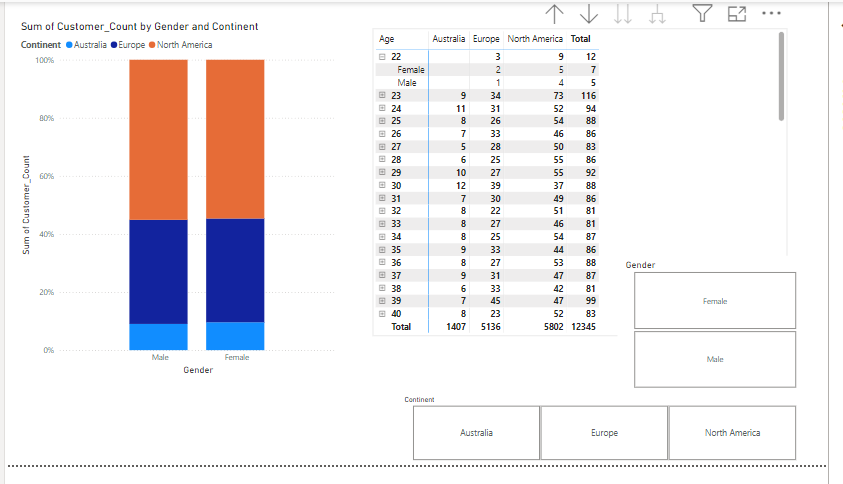
**1.DEMOGRAPHIC DISTRIBUTION**

SELECT Gender, City, State, Country, Continent, EXTRACT(YEAR FROM CURRENT\_DATE) - EXTRACT(YEAR FROM Birthday) AS Age, COUNT(\*) AS Customer\_Count FROM processed\_customers GROUP BY Gender, City, State, Country, Continent, Age ORDER BY Gender, Age;



2.PURCHASE PATTERNS

SELECT

CustomerKey,

COUNT(Order\_Number) AS Total\_Purchases,

AVG(Unit\_Price\_USD \* Quantity) AS Average\_Order\_Value,

SUM(Unit\_Price\_USD \* Quantity) AS Total\_Spend

FROM

Processed\_sales

JOIN

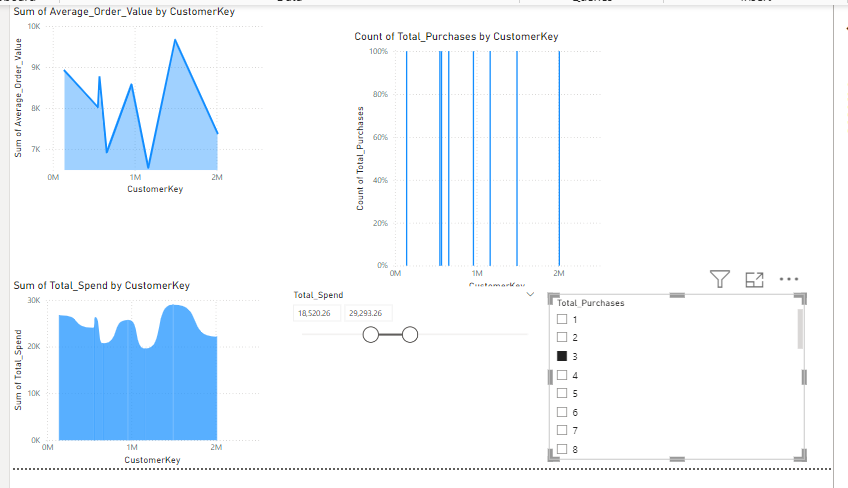
processed\_products ON processed\_sales.ProductKey = processed\_products.ProductKey

GROUP BY

CustomerKey

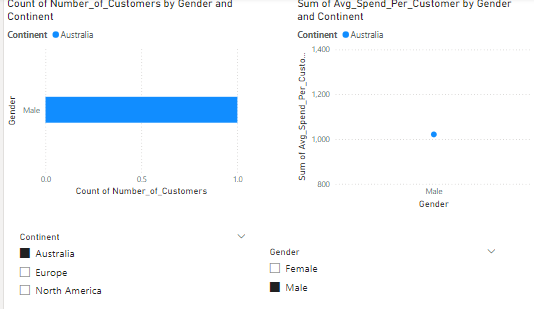
ORDER BY

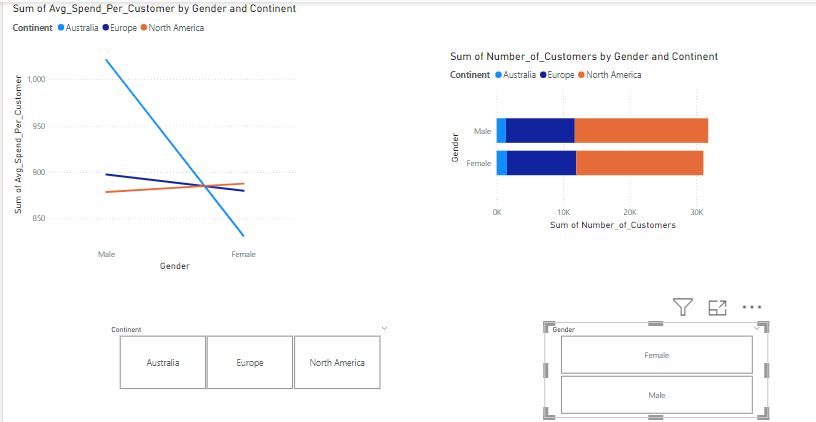
Total\_Spend DESC;



3.Customer Segmentation

SELECT Gender, Continent, COUNT(\*) AS Number\_of\_Customers, AVG(Unit\_Price\_USD \* Quantity) AS Avg\_Spend\_Per\_Customer FROM Customers JOIN Sales ON Customers.CustomerKey = Sales.CustomerKey JOIN Products ON Sales.ProductKey = Products.ProductKey GROUP BY Gender, Continent ORDER BY Avg\_Spend\_Per\_Customer DESC;





4.Overall Sales Performance

SELECT

EXTRACT(YEAR FROM Order\_Date) AS Year,

EXTRACT(MONTH FROM Order\_Date) AS Month,

SUM(Unit\_Price\_USD \* Quantity) AS Total\_Sales

FROM

processed\_sales

JOIN

processed\_products ON processed\_sales.ProductKey = processed\_products.ProductKey

GROUP BY

Year, Month

ORDER BY

Year, Month;



5. Sales by Product

SELECT

Product\_Name,

SUM(Quantity) AS Total\_Quantity\_Sold,

SUM(Unit\_Price\_USD \* Quantity) AS Total\_Revenue

FROM

processed\_sales

JOIN

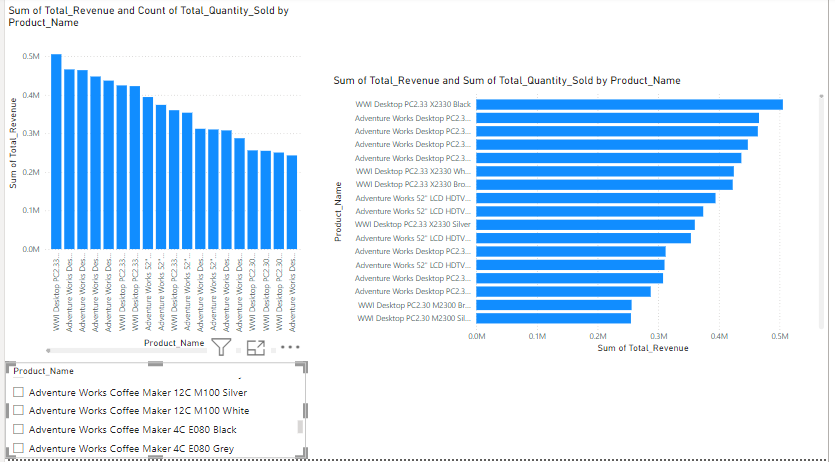
processed\_products ON processed\_sales.ProductKey = processed\_products.ProductKey

GROUP BY

Product\_Name

ORDER BY

Total\_Revenue DESC;



6. Sales by Store

SELECT

ps.StoreKey,

SUM(pp.Unit\_Price\_USD \* ps.Quantity) AS Total\_Sales,

COUNT(DISTINCT ps.Order\_Number) AS Number\_of\_Orders

FROM

processed\_sales ps

JOIN

processed\_stores pst ON ps.StoreKey = pst.StoreKey

JOIN

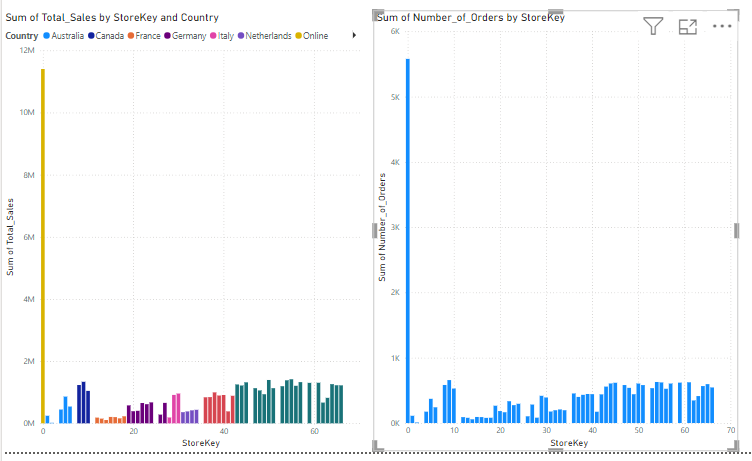
processed\_products pp ON ps.ProductKey = pp.ProductKey

GROUP BY

ps.StoreKey

ORDER BY

Total\_Sales DESC;



7. Sales by Currency

SELECT

Currency\_Code,

SUM(Unit\_Price\_USD \* Quantity) AS Total\_Sales

FROM

processed\_sales

JOIN

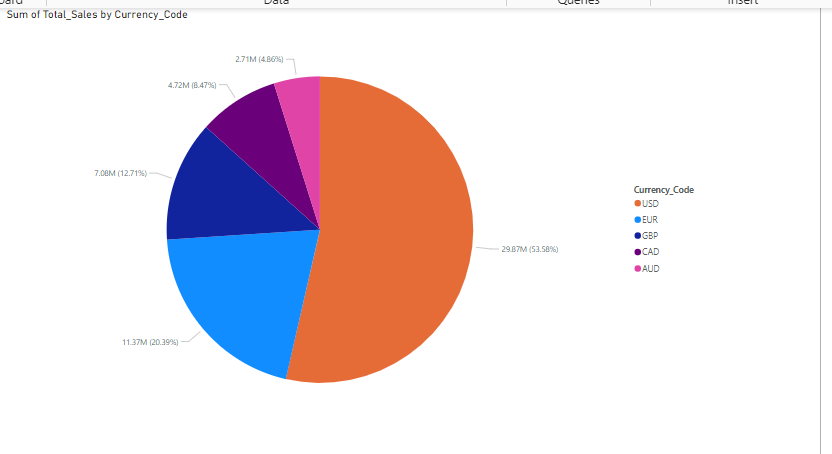
processed\_products ON processed\_sales.ProductKey = processed\_products.ProductKey

GROUP BY

Currency\_Code

ORDER BY

Total\_Sales DESC;



8. Product Popularity

SELECT

Product\_Name,

COUNT(\*) AS Number\_of\_Sales,

SUM(Quantity) AS Total\_Quantity\_Sold

FROM

processed\_sales

JOIN

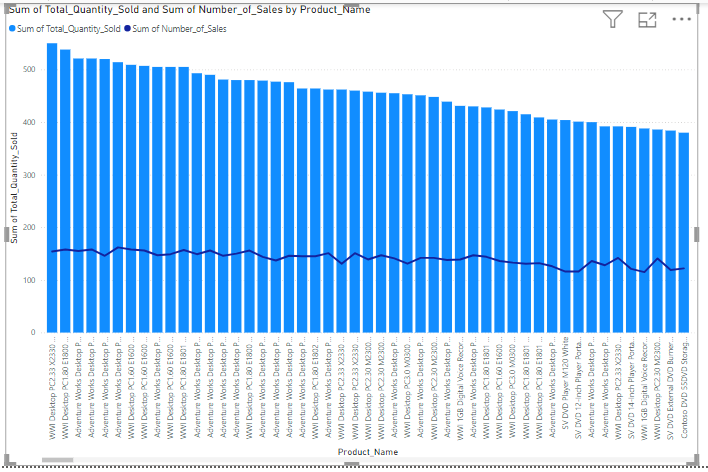
processed\_products ON processed\_sales.ProductKey = processed\_products.ProductKey

GROUP BY

Product\_Name

ORDER BY

Total\_Quantity\_Sold DESC;



9. Profitability Analysis

SELECT

Product\_Name,

Unit\_Price\_USD - Unit\_Cost\_USD AS Profit\_Margin,

SUM(Unit\_Price\_USD \* Quantity) AS Total\_Revenue

FROM

processed\_sales

JOIN

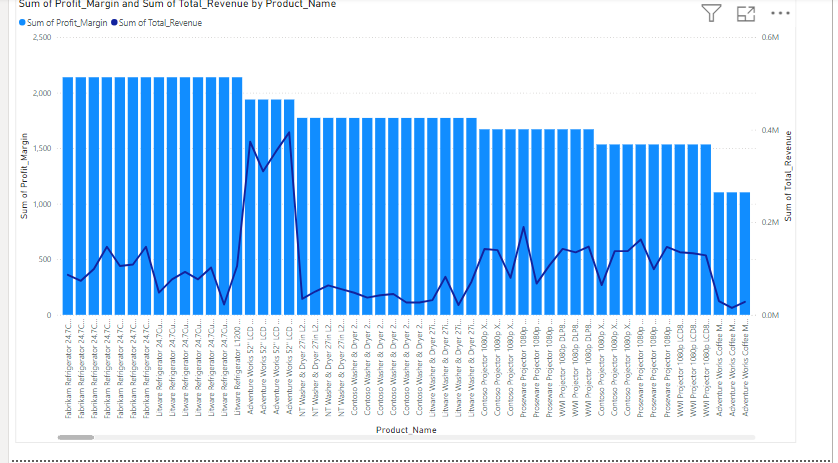
processed\_products ON processed\_sales.ProductKey = processed\_products.ProductKey

GROUP BY

Product\_Name, Unit\_Price\_USD, Unit\_Cost\_USD

ORDER BY

Profit\_Margin DESC;



**10.store performance**

SELECT ps.StoreKey, ps.Country, ps.State, ps.Square\_Meters, SUM(pp.Unit\_Price\_USD \* psales.Quantity) AS Total\_Sales, EXTRACT(YEAR FROM CURRENT\_DATE) - EXTRACT(YEAR FROM ps.Open\_Date) AS Store\_Age FROM processed\_stores ps LEFT JOIN processed\_sales psales ON ps.StoreKey = psales.StoreKey JOIN processed\_products pp ON psales.ProductKey = pp.ProductKey GROUP BY ps.StoreKey, ps.Country, ps.State, ps.Square\_Meters, ps.Open\_Date ORDER BY Total\_Sales DESC;

