**[AdventureWorks-SalesAnalysis](https://github.com/Future-Analyst/AdventureWorks-SalesAnalysis) Sql Doc**

**KPI’s**

-- Total Revenue

SELECT

ROUND(SUM(ft.OrderQuantity \* pr.ProductPrice),0) AS total\_revenue

FROM

FactTable ft

JOIN

AdventureWorks\_Products pr ON ft.ProductKey = pr.ProductKey;



-- Total Transaction

SELECT

COUNT(\*) AS total\_transaction

FROM

FactTable;



-- Total Quantity Sold

SELECT

SUM(OrderQuantity) AS total\_quantity

FROM

FactTable;



-- Total Returned Quantity

SELECT

SUM(ReturnQuantity) AS total\_returned\_qty

FROM

AdventureWorks\_Returns;



-- Total Cost

SELECT

ROUND(SUM(ft.OrderQuantity \* pr.ProductCost),0) AS total\_cost

FROM

FactTable ft

JOIN

AdventureWorks\_Products pr ON ft.ProductKey = pr.ProductKey;



-- Profit Margin

SELECT

(ROUND(SUM(ft.OrderQuantity \* pr.ProductPrice),0) - ROUND(SUM(ft.OrderQuantity \* pr.ProductCost),0)) AS profit\_margin

FROM

FactTable ft

JOIN

AdventureWorks\_Products pr ON ft.ProductKey = pr.ProductKey;



-- Highest Ordered Product

SELECT TOP 1

pr.ProductName,

SUM(OrderQuantity) AS total\_quantity\_ordered

FROM

FactTable ft

JOIN

AdventureWorks\_Products pr ON ft.ProductKey = pr.ProductKey

GROUP BY

pr.ProductName

ORDER BY

total\_quantity\_ordered DESC;



----- Product with Highest Profit

SELECT TOP 1

pr.ProductName,

ROUND(SUM(ft.OrderQuantity \* pr.ProductPrice),0) - ROUND(SUM(ft.OrderQuantity \* pr.ProductCost),0) AS profit\_margin

FROM

FactTable ft

JOIN

AdventureWorks\_Products pr ON ft.ProductKey = pr.ProductKey

GROUP BY

pr.ProductName

ORDER BY

profit\_margin DESC;



--- Top 3 product category with Highest profit

SELECT

pc.CategoryName,

ROUND(SUM(ft.OrderQuantity \* pr.ProductPrice),0) - ROUND(SUM(ft.OrderQuantity \* pr.ProductCost),0) AS profit\_margin

FROM

FactTable ft

JOIN

AdventureWorks\_Products pr ON ft.ProductKey = pr.ProductKey

JOIN

AdventureWorks\_Product\_Subcategories ps ON pr.ProductSubcategoryKey = ps.ProductCategoryKey

JOIN

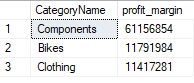
AdventureWorks\_Product\_Categories pc ON ps.ProductCategoryKey = pc.ProductCategoryKey

GROUP BY

pc.CategoryName

ORDER BY

profit\_margin DESC;



-- Profit Margin by Product Color

SELECT

pr.ProductColor,

ROUND(SUM(ft.OrderQuantity \* pr.ProductPrice),0) - ROUND(SUM(ft.OrderQuantity \* pr.ProductCost),0) AS profit\_margin

FROM

FactTable ft

JOIN

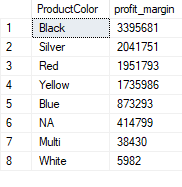
AdventureWorks\_Products pr ON ft.ProductKey = pr.ProductKey

GROUP BY

pr.ProductColor

ORDER BY

profit\_margin DESC;



-- TOp 10 Best Selling Products

SELECT TOP 10

pr.ProductName,

(SUM(re.ReturnQuantity) / NULLIF(SUM(ft.OrderQuantity), 0)) \* 100 AS ReturnRate,

COUNT(\*) AS total\_transactions,

COUNT(DISTINCT ft.CustomerKey) AS total\_customers,

SUM(ft.OrderQuantity) AS quantity\_sold

FROM

FactTable ft

JOIN

AdventureWorks\_Returns re ON ft.ProductKey = re.ProductKey

JOIN

AdventureWorks\_Products pr ON ft.ProductKey = pr.ProductKey

GROUP BY

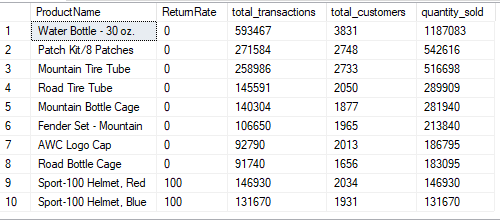
pr.ProductName

ORDER BY

quantity\_sold DESC,

total\_transactions DESC,

total\_customers DESC;



--------------------------------------Returned Analysis Query----------------------------------

-- Total Returned Transactions

SELECT

COUNT(\*) AS total\_return\_transactions

FROM

AdventureWorks\_Returns

-- Total Returned Quantity

SELECT

SUM(ReturnQuantity) AS total\_returned\_quantity

FROM

AdventureWorks\_Returns;



-- Total Refund

SELECT

COALESCE(SUM(re.ReturnQuantity \* pr.ProductPrice), 0) AS total\_refund

FROM

AdventureWorks\_Returns re

JOIN

AdventureWorks\_Products pr ON re.ProductKey = pr.ProductKey;



-- Returned Transaction Rate

WITH total\_transaction AS (

SELECT

COUNT(\*) AS TotalTransactions

FROM

FactTable

),

total\_return AS (

SELECT

COUNT(\*) AS TotalReturns

FROM

AdventureWorks\_Returns

)

SELECT

ROUND(COALESCE(tr.TotalReturns \* 100.0 / NULLIF(tt.TotalTransactions, 0), 0), 2) AS ReturnedTransactionRate

FROM

total\_transaction tt, total\_return tr;



-- Total Returned Quantity

SELECT

SUM(ReturnQuantity) AS total\_returned\_quanityt

FROM

AdventureWorks\_Returns;



-- Total Returned Rate

WITH total\_sold\_quantity AS (

SELECT

SUM(OrderQuantity) AS TotalQuantitySold

FROM

FactTable

),

total\_return\_quantity AS (

SELECT

SUM(ReturnQuantity) AS TotalQuantity

FROM

AdventureWorks\_Returns

)

SELECT

ROUND(

COALESCE(tr.TotalQuantity \* 100.0 / NULLIF(tt.TotalQuantitySold, 0), 0),

2

) AS ReturnedQuantityRate

FROM

total\_sold\_quantity tt,

total\_return\_quantity tr;



-- Total Refund Transaction

WITH Refunds AS (

SELECT

SUM(fr.ReturnQuantity \* p.ProductPrice) AS TotalRefund

FROM

AdventureWorks\_Returns fr

JOIN

AdventureWorks\_Products p ON fr.ProductKey = p.ProductKey

)

SELECT

COALESCE(TotalRefund, 0) AS TransactionRefund

FROM

Refunds;



-- Total Refund rate

WITH total\_revenue AS (

SELECT

SUM(ft.OrderQuantity \* pr.ProductPrice) AS total\_revenue --

FROM

FactTable ft

JOIN

AdventureWorks\_Products pr ON ft.ProductKey = pr.ProductKey

),

total\_refund AS (

SELECT

SUM(fr.ReturnQuantity \* p.ProductPrice) AS TotalRefund

FROM

AdventureWorks\_Returns fr

JOIN

AdventureWorks\_Products p ON fr.ProductKey = p.ProductKey -- Adjust the join condition based on your schema

)

SELECT

ROUND(

COALESCE(tr.TotalRefund / NULLIF(trv.total\_revenue, 0), 0),

2

) AS RefundRate

FROM

total\_revenue trv,

total\_refund tr;



-- Product with Highest Returns

SELECT TOP 1

pr.ProductName,

SUM(re.ReturnQuantity) AS total\_returned\_quantity

FROM

AdventureWorks\_Returns re

JOIN

AdventureWorks\_Products pr ON re.ProductKey = pr.ProductKey

GROUP BY

pr.ProductName

ORDER BY

total\_returned\_quantity DESC;



-- category with Highest Returned Quantity

SELECT TOP 1

pc.CategoryName,

SUM(re.ReturnQuantity) AS total\_returned\_quantity

FROM

AdventureWorks\_Returns re

JOIN

AdventureWorks\_Products pr ON re.ProductKey = pr.ProductKey

JOIN

AdventureWorks\_Product\_Subcategories ps ON pr.ProductSubcategoryKey = ps.ProductSubcategoryKey

JOIN

AdventureWorks\_Product\_Categories pc ON ps.ProductCategoryKey = pc.ProductCategoryKey

GROUP BY

pc.CategoryName

ORDER BY

total\_returned\_quantity DESC;



-- Country with Highest returns

SELECT TOP 1

te.Country,

SUM(re.ReturnQuantity) AS total\_returned\_quantity

FROM

AdventureWorks\_Returns re

JOIN

AdventureWorks\_Territories te ON re.TerritoryKey = te.SalesTerritoryKey

GROUP BY

te.Country

ORDER BY

total\_returned\_quantity DESC;

