--1. Total Sales by City

SELECT

City,

SUM(Sales) AS TotalSales

FROM

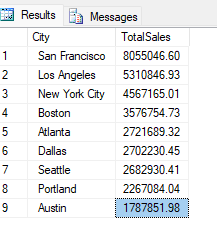
Orders

GROUP BY

City

ORDER BY

TotalSales DESC;



--2. Top-Selling Products

WITH RankedProducts AS (

SELECT

Product,

SUM(QuantityOrdered) AS TotalQuantity,

ROW\_NUMBER() OVER (ORDER BY SUM(QuantityOrdered) DESC) AS RowNum

FROM

Orders

GROUP BY

Product

)

SELECT

Product,

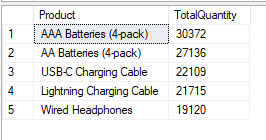
TotalQuantity

FROM

RankedProducts

WHERE

RowNum <= 5;



--3. Monthly Revenue Trends

SELECT

Month,

SUM(Sales) AS MonthlyRevenue

FROM

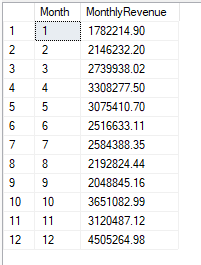
Orders

GROUP BY

Month

ORDER BY

Month;



--4. Total Sales by Product Category

SELECT

ProductCategory,

SUM(Sales) AS TotalSales

FROM

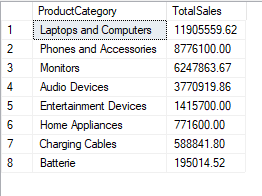
Orders

GROUP BY

ProductCategory

ORDER BY

TotalSales DESC;



--5. Hourly Sales Distribution

SELECT

Hour,

SUM(Sales) AS TotalSales

FROM

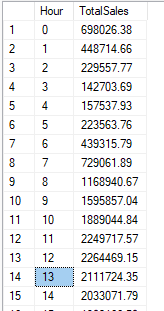
Orders

GROUP BY

Hour

ORDER BY

Hour;



--6. Cumulative Sales by City

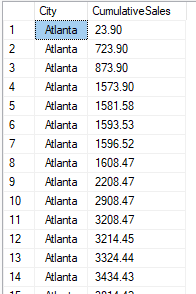
SELECT

City,

SUM(Sales) OVER (PARTITION BY City ORDER BY OrderDate) AS CumulativeSales

FROM

Orders;



--7. Ranking Products by Sales

SELECT

Product,

SUM(Sales) AS TotalSales,

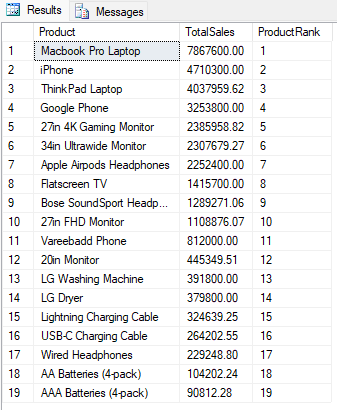
RANK() OVER (ORDER BY SUM(Sales) DESC) AS ProductRank

FROM

Orders

GROUP BY

Product;



-------------Top Products by Sales in Each City

WITH CityProductSales AS (

SELECT

City,

Product,

SUM(Sales) AS TotalSales,

RANK() OVER (PARTITION BY City ORDER BY SUM(Sales) DESC) AS ProductRank

FROM

Orders

GROUP BY

City, Product

)

SELECT

City,

Product,

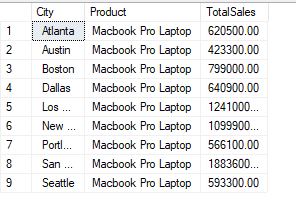
TotalSales

FROM

CityProductSales

WHERE

ProductRank = 1;



-------------------------- Products Contributing to 80% of Sales

WITH TotalSales AS (

SELECT

Product,

SUM(Sales) AS ProductSales

FROM

Orders

GROUP BY

Product

),

CumulativeSales AS (

SELECT

Product,

ProductSales,

SUM(ProductSales) OVER (ORDER BY ProductSales DESC) AS RunningTotal,

SUM(ProductSales) OVER () AS TotalSales

FROM

TotalSales

)

SELECT

Product,

ProductSales,

RunningTotal,

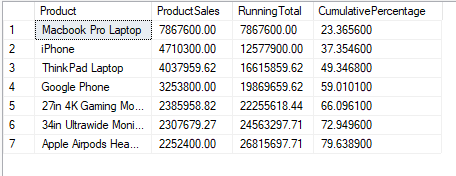
(RunningTotal / TotalSales) \* 100 AS CumulativePercentage

FROM

CumulativeSales

WHERE

(RunningTotal / TotalSales) <= 0.8;



------------------------------------Filter Cities with Sales Above a Threshold and Specific Product Demand

SELECT

City,

SUM(Sales) AS TotalSales,

SUM(CASE WHEN Product = 'USB-C Charging Cable' THEN QuantityOrdered ELSE 0 END) AS USBCCableQuantity

FROM

Orders

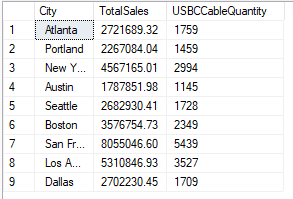
GROUP BY

City

HAVING

SUM(Sales) > 10000

AND SUM(CASE WHEN Product = 'USB-C Charging Cable' THEN QuantityOrdered ELSE 0 END) > 50;



----------Fast Query (Using JOIN and Aggregation)------------

SELECT

O.[Sr],

O.[OrderID],

O.[ProductCategory],

O.[Product],

O.[QuantityOrdered],

O.[PriceEach],

O.[OrderDate],

O.[PurchaseAddress],

O.[Month],

O.[Sales],

O.[City],

O.[Hour],

O.[TimeOfDay]

FROM

[Health\_Food\_Flask].[dbo].[Orders] O

INNER JOIN (

SELECT

[City],

MAX([Sales]) AS MaxSales

FROM

[Health\_Food\_Flask].[dbo].[Orders]

GROUP BY

[City]

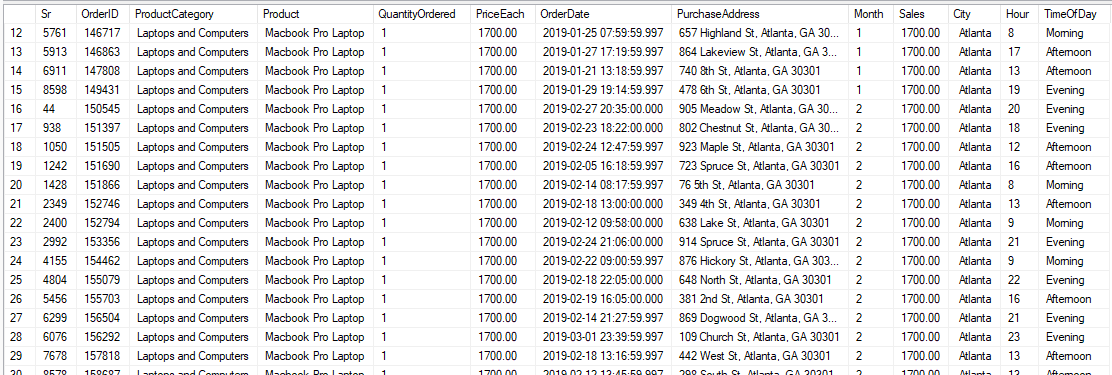
) AS SubQuery ON

O.[City] = SubQuery.[City] AND

O.[Sales] = SubQuery.MaxSales

ORDER BY

O.[City];



--------------------Slow Query (Using HAVING with Correlated Subquery)-----------------

SELECT

[Sr],

[OrderID],

[ProductCategory],

[Product],

[QuantityOrdered],

[PriceEach],

[OrderDate],

[PurchaseAddress],

[Month],

[Sales],

[City],

[Hour],

[TimeOfDay]

FROM

[Health\_Food\_Flask].[dbo].[Orders] O1

GROUP BY

[Sr],

[OrderID],

[ProductCategory],

[Product],

[QuantityOrdered],

[PriceEach],

[OrderDate],

[PurchaseAddress],

[Month],

[Sales],

[City],

[Hour],

[TimeOfDay]

HAVING

[Sales] = (

SELECT

MAX([Sales])

FROM

[Health\_Food\_Flask].[dbo].[Orders] O2

WHERE

O2.[City] = O1.[City]

)

ORDER BY

[City];

