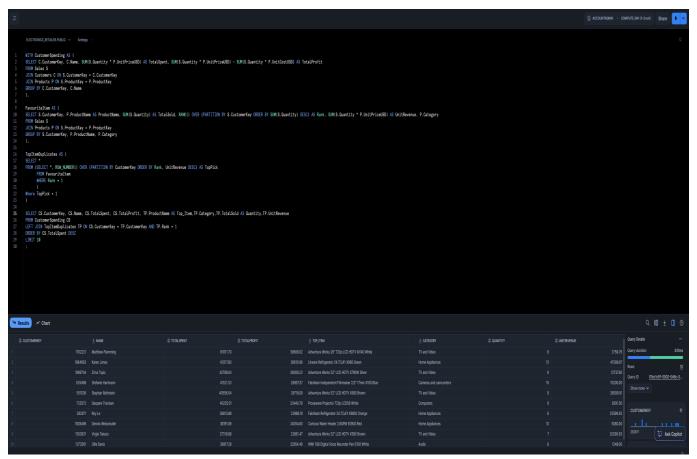Logan Samples

05/02/2025

Query Examples

Top Customers

This query uses 3 temporary tables to find the top 10 customers, how much they spent, profit made from their transactions, item most bought, the quantity of that item purchased, how much total was spent buying those items and the category of the item. The first temporary table uses aggregate functions SUM to find the revenue and profit from each customer by joining the sales, customers, and products tables.
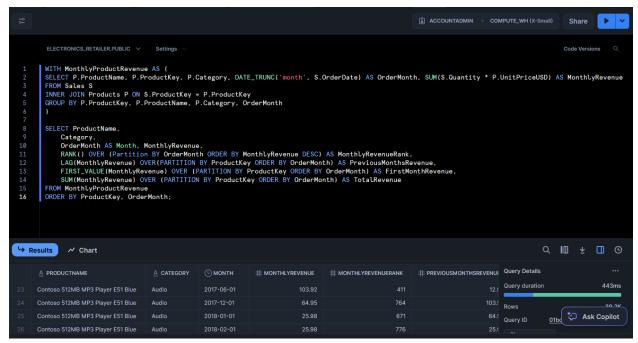
The second temporary table uses the window function RANK() to rank the products an individual customer has bought, not across all customers. Each customer will have a ranking of products based on how many units of that product they purchased. There are issues with this

method. If there is a tie with most purchased products, there will be multiple rank 1 products. This led me to create a 3rd table.

The top item duplicates table uses a sub-query along with another window function to sort out which items tied at the top spot had the most revenue. This window function goes through each customer's rank 1 products and puts the product with the most revenue at row number 1.

The query itself combines these temporary tables to find our desired information by joining tables with the customer key. AND clause is used to only join the tables by products that have rank 1.

Monthly Product Revenue

I wanted to find the performance of products on a monthly basis. A temporary table was used to aggregate monthly revenue for each product. The first window function in this query ranks how revenue each product produced for each month. The LAG function was used to go through each product's monthly revenue and give the previous month's value. The FIRST_VALUE function gives the oldest monthly revenue for each product. The SUM window function calculates the running total for revenue of each product.

Product Performance


```sql
SELECT P.ProductName, P.Category, S.OrderDate, SUM(S.Quantity) AS TotalSold, SUM(S.Quantity * P.UnitPriceUSD) AS UnitRevenue, SUM(S.Quantity * P.UnitPriceUSD - S.Quantity * P.UnitCostUSD) AS UnitProfit, C.Country, C.State
FROM Sales S
INNER JOIN Products P ON S.ProductKey = P.ProductKey
INNER JOIN Customers C ON S.CustomerKey = C.CustomerKey
WHERE YEAR(S.OrderDate) = 2016
GROUP BY P.ProductName, P.Category, S.OrderDate, C.Country, C.State
ORDER BY S.OrderDate, P.Category, P.ProductName
;
```

This query selects quantity sold, profit, and revenue for each product in 2016 per location and time. This query is useful for tracking these attributes over time and location.

Other Examples Below

ACCOUNTADMIN • COMPUTE_WH (X-Small)   Share

ELECTRONICS_RETAILER.PUBLIC   Settings   Code Versions

```sql
SELECT P.ProductKey, S.Quantity, P.UnitCostUSD, P.UnitPriceUSD
FROM Sales S
INNER JOIN Prodcuts P ON S.ProductKey = P.ProductKey
INNER JOIN Stores ST ON S.StoreKey = ST.StoreKey
WHERE ST.COUNTRY != 'Online'
ORDER BY ProductKey;
```

Results   Chart

	SALESMONTH	REVENUE
1	2016-01-01	649918.78
2	2016-02-01	891098.30
3	2016-03-01	338407.36
4	2016-04-01	110591.63

Query Details

Query duration   1.4s

Rows

Query ID   01bc

Ask Copilot

```sql
1   SELECT P.ProductKey, S.Quantity, P.UnitCostUSD, P.UnitPriceUSD
2   FROM Sales S
3   INNER JOIN Prodcuts P ON S.ProductKey = P.ProductKey
4   INNER JOIN Stores ST ON S.StoreKey = ST.StoreKey
5   WHERE ST.COUNTRY = 'Online'
6   ORDER BY ProductKey;
```

↳ Results   ∿ Chart   🔍 ▮▯ ⬇ ◧ ⏱

	🕐 SALESMONTH	# REVENUE
1	2016-01-01	649918.78
2	2016-02-01	891098.30
3	2016-03-01	338407.36
4	2016-04-01	110591.63

Query Details   ...

Query duration   1.4s

Rows

Query ID   01b...   💬 Ask Copilot