SQL MINI PROJECT

This is subject to review, all opinions and questions are welcome

Question 1:

Provide the top 10 customers (full name) by revenue, the country they shipped to, the cities and their revenue (orderqty * unitprice).

```
SELECT TOP 10
    CONCAT(c.FirstName, ' ', c.LastName) AS FullName,
    a.CountryRegion AS ShippedCountry,
    a.City AS ShippedCity,
    SUM(sod.OrderQty * sod.UnitPrice) AS Revenue
FROM SalesLT.Customer c
JOIN SalesLT.SalesOrderHeader soh ON c.CustomerID = soh.CustomerID
JOIN SalesLT.CustomerAddress ca ON c.CustomerID = ca.CustomerID
JOIN SalesLT.Address a ON ca.AddressID = a.AddressID
JOIN SalesLT.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID
GROUP BY c.CustomerID, c.FirstName, c.LastName, a.CountryRegion, a.City
ORDER BY Revenue DESC;
```

Question 2:

Create 4 distinct Customer segments using the total Revenue (orderqty * unitprice) by customer. List the customer details (ID, Company Name), Revenue, and the segment the customer belongs to.

```
SELECT c.CustomerID, c.CompanyName, SUM(sod.OrderQty * sod.UnitPrice) AS Revenue,
    CASE
        WHEN SUM(sod.OrderQty * sod.UnitPrice) >= 10000 THEN 'High Revenue'
        WHEN SUM(sod.OrderQty * sod.UnitPrice) >= 5000 THEN 'Medium Revenue'
        WHEN SUM(sod.OrderQty * sod.UnitPrice) >= 1000 THEN 'Low Revenue'
        ELSE 'Very Low Revenue'
    END AS CustomerSegment
FROM SalesLT.Customer c
```

```sql
JOIN SalesLT.SalesOrderHeader soh ON c.CustomerID = soh.CustomerID

JOIN SalesLT.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID

GROUP BY c.CustomerID, c.CompanyName

ORDER BY Revenue DESC;
```

Question 3:

What products with their respective categories did our customers buy on our last day of business? List the CustomerID, Product ID, Product Name, Category Name, and Order Date.

```sql
SELECT soh.CustomerID, sod.ProductID, p.Name AS ProductName, pc.Name AS CategoryName, soh.OrderDate

FROM SalesLT.SalesOrderHeader soh

JOIN SalesLT.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID

JOIN SalesLT.Product p ON sod.ProductID = p.ProductID

JOIN SalesLT.ProductCategory pc ON p.ProductCategoryID = pc.ProductCategoryID

WHERE soh.OrderDate = (SELECT MAX(OrderDate) FROM SalesLT.SalesOrderHeader);
```

Question 4:

Create a View called customer segments that stores the details (id, name, revenue) for customers and their segment (from Question 2).

```sql
CREATE VIEW customersegment AS

SELECT c.CustomerID, c.CompanyName, SUM(sod.OrderQty * sod.UnitPrice) AS Revenue,

  CASE

    WHEN SUM(sod.OrderQty * sod.UnitPrice) >= 10000 THEN 'High Revenue'

    WHEN SUM(sod.OrderQty * sod.UnitPrice) >= 5000 THEN 'Medium Revenue'

    WHEN SUM(sod.OrderQty * sod.UnitPrice) >= 1000 THEN 'Low Revenue'

    ELSE 'Very Low Revenue'

  END AS CustomerSegment

FROM SalesLT.Customer c

JOIN SalesLT.SalesOrderHeader soh ON c.CustomerID = soh.CustomerID

JOIN SalesLT.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID

GROUP BY c.CustomerID, c.CompanyName;
```

Question 5:

What are the top 3 selling products (include product name) in each category (include category name) by revenue? Tip: Use ranknum.

```
SELECT CategoryName, ProductName, Revenue

FROM (

    SELECT pc.Name AS CategoryName, p.Name AS ProductName,

           SUM(sod.OrderQty * sod.UnitPrice) AS Revenue,

           RANK() OVER (PARTITION BY pc.Name ORDER BY SUM(sod.OrderQty * sod.UnitPrice) DESC) AS RankNum

    FROM SalesLT.SalesOrderDetail sod

    JOIN SalesLT.Product p ON sod.ProductID = p.ProductID

    JOIN SalesLT.ProductCategory pc ON p.ProductCategoryID = pc.ProductCategoryID

    GROUP BY pc.Name, p.Name

) ranked

WHERE RankNum <= 3;
```