OMSBA 5145

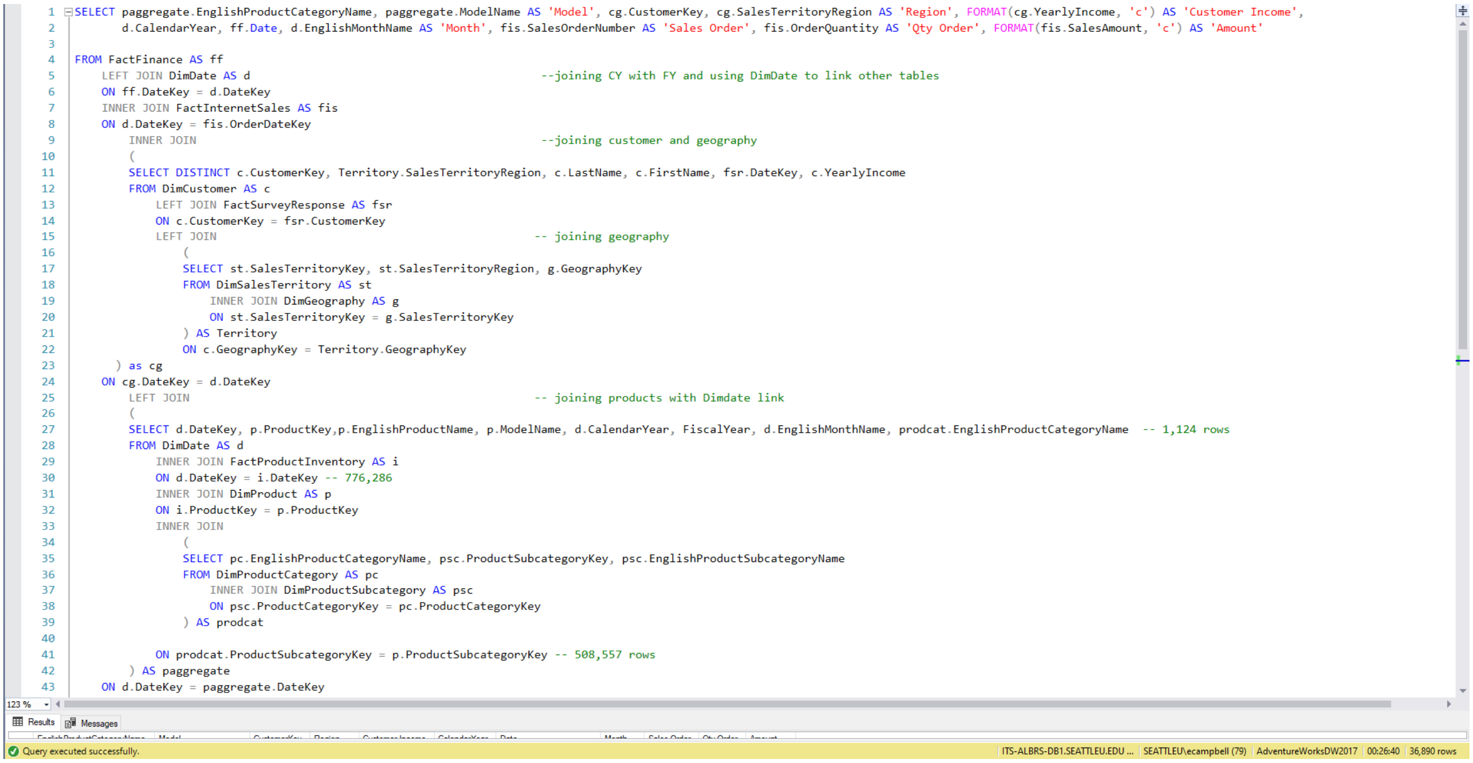
DATA TRANSLATION CHALLENGE

TECHNICAL PAPER ADVENTURE WORKS 2017

By

Campbell, Erik

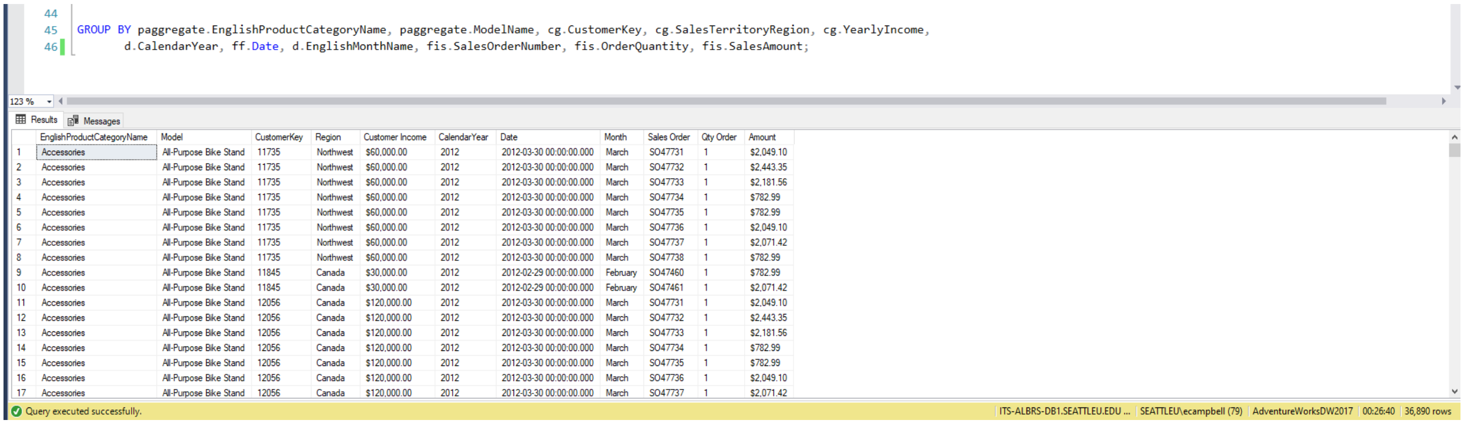
The first business requirement was to evaluate internet sales and provide a query that would present the following the following columns: Category, Model, Customer Key, Region, Income, Calendar year, Fiscal year, Month, Sales Order Number, Sales Order Quantity and Sales Amount. There were several tables that needed to be joined and subqueries performed to get the variables requested. I determined that date was one variable common to all the tables and could be used to create linking tables. Thus, I decided to use FactFinance and DimDate in the FROM Clause and perform complex joins and subqueries from that root table in the FROM Clause. I joined the calendar year and fiscal year using date keys and then joined internet sales on a foreign date key. My next steps were joining this data with a subquery that grabbed data from tables containing geographical information, in turn could be used to link with customer data. This new data set then was left-joined with a subquery consisting of fields from three product-related tables. The global query resulted in 36,890 rows, but the query speed was suboptimal at 26 minutes to run during high server traffic times. The global query be further evaluated to examine if the sequence and/or order of subqueries in relation to the global query, or strategically placing more subqueries in the WHERE clause of the subqueries result in faster query response. I recommend that this query be run outside conventional business hours when there is less demand on the server. Or,



```sql
SELECT paggregate.EnglishProductCategoryName, paggregate.ModelName AS 'Model', cg.CustomerKey, cg.SalesTerritoryRegion AS 'Region', FORMAT(cg.YearlyIncome, 'c') AS 'Customer Income',
       d.CalendarYear, ff.Date, d.EnglishMonthName AS 'Month', fis.SalesOrderNumber AS 'Sales Order', fis.OrderQuantity AS 'Qty Order', FORMAT(fis.SalesAmount, 'c') AS 'Amount'

FROM FactFinance AS ff
    LEFT JOIN DimDate AS d                                        --joining CY with FY and using DimDate to link other tables
    ON ff.DateKey = d.DateKey
    INNER JOIN FactInternetSales AS fis
    ON d.DateKey = fis.OrderDateKey
        INNER JOIN                                                --joining customer and geography
        (
        SELECT DISTINCT c.CustomerKey, Territory.SalesTerritoryRegion, c.LastName, c.FirstName, fsr.DateKey, c.YearlyIncome
        FROM DimCustomer AS c
            LEFT JOIN FactSurveyResponse AS fsr
            ON c.CustomerKey = fsr.CustomerKey
            LEFT JOIN                                            -- joining geography
                (
                SELECT st.SalesTerritoryKey, st.SalesTerritoryRegion, g.GeographyKey
                FROM DimSalesTerritory AS st
                    INNER JOIN DimGeography AS g
                    ON st.SalesTerritoryKey = g.SalesTerritoryKey
                ) AS Territory
                ON c.GeographyKey = Territory.GeographyKey
      ) as cg
    ON cg.DateKey = d.DateKey
        LEFT JOIN                                                -- joining products with Dimdate link
        (
        SELECT d.DateKey, p.ProductKey,p.EnglishProductName, p.ModelName, d.CalendarYear, FiscalYear, d.EnglishMonthName, prodcat.EnglishProductCategoryName  -- 1,124 rows
        FROM DimDate AS d
            INNER JOIN FactProductInventory AS i
            ON d.DateKey = i.DateKey -- 776,286
            INNER JOIN DimProduct AS p
            ON i.ProductKey = p.ProductKey
            INNER JOIN
                (
                SELECT pc.EnglishProductCategoryName, psc.ProductSubcategoryKey, psc.EnglishProductSubcategoryName
                FROM DimProductCategory AS pc
                    INNER JOIN DimProductSubcategory AS psc
                    ON psc.ProductCategoryKey = pc.ProductCategoryKey
                ) AS prodcat

            ON prodcat.ProductSubcategoryKey = p.ProductSubcategoryKey -- 508,557 rows
        ) AS paggregate
    ON d.DateKey = paggregate.DateKey
```

Query executed successfully.

ITS-ALBRS-DB1.SEATTLEU.EDU ... | SEATTLEU\ecampbell (79) | AdventureWorksDW2017 | 00:26:40 | 36,890 rows



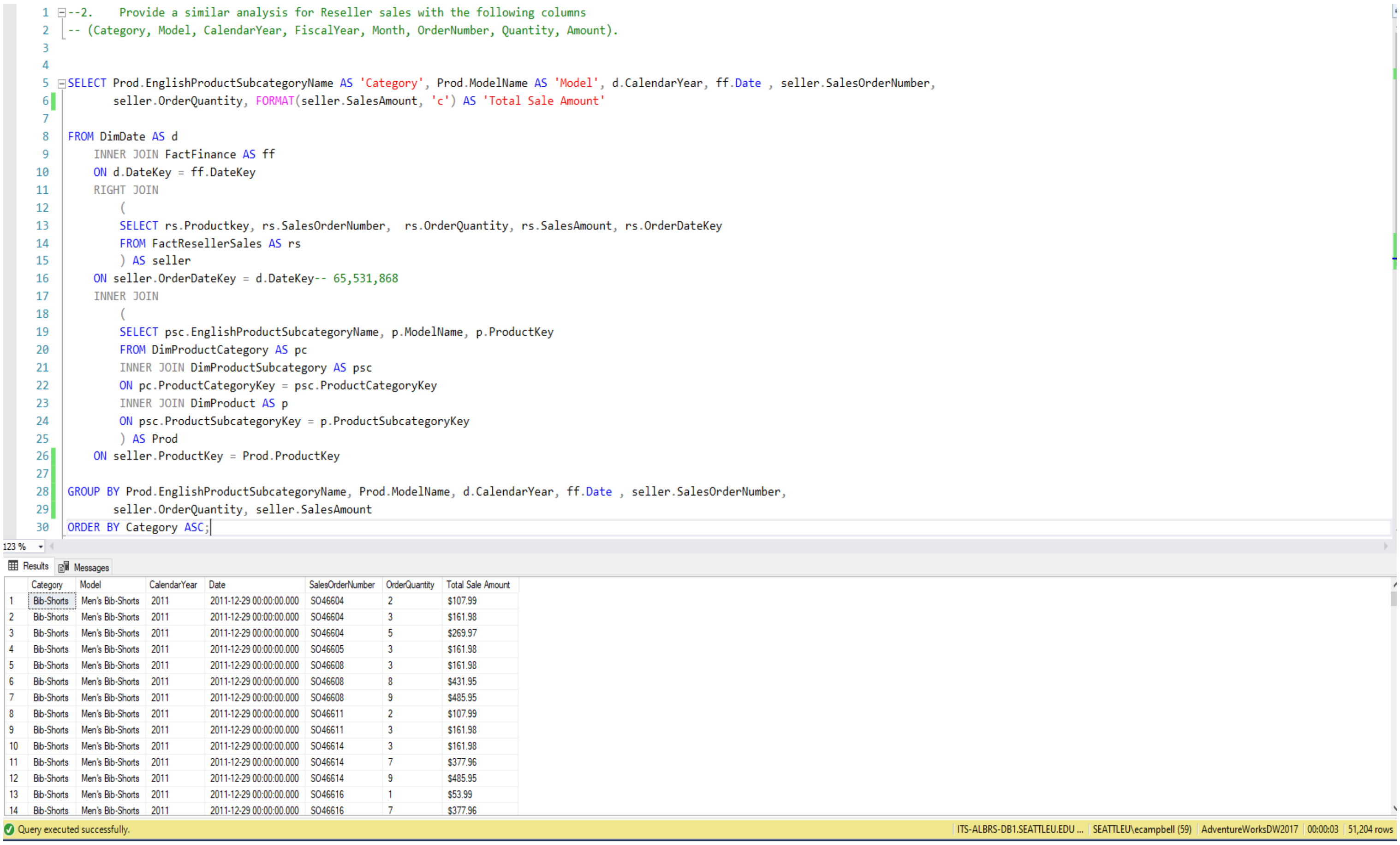
```sql

GROUP BY paggregate.EnglishProductCategoryName, paggregate.ModelName, cg.CustomerKey, cg.SalesTerritoryRegion, cg.YearlyIncome,
       d.CalendarYear, ff.Date, d.EnglishMonthName, fis.SalesOrderNumber, fis.OrderQuantity, fis.SalesAmount;
```

	EnglishProductCategoryName	Model	CustomerKey	Region	Customer Income	CalendarYear	Date	Month	Sales Order	Qty Order	Amount
1	Accessories	All-Purpose Bike Stand	11735	Northwest	$60,000.00	2012	2012-03-30 00:00:00.000	March	SO47731	1	$2,049.10
2	Accessories	All-Purpose Bike Stand	11735	Northwest	$60,000.00	2012	2012-03-30 00:00:00.000	March	SO47732	1	$2,443.35
3	Accessories	All-Purpose Bike Stand	11735	Northwest	$60,000.00	2012	2012-03-30 00:00:00.000	March	SO47733	1	$2,181.56
4	Accessories	All-Purpose Bike Stand	11735	Northwest	$60,000.00	2012	2012-03-30 00:00:00.000	March	SO47734	1	$782.99
5	Accessories	All-Purpose Bike Stand	11735	Northwest	$60,000.00	2012	2012-03-30 00:00:00.000	March	SO47735	1	$782.99
6	Accessories	All-Purpose Bike Stand	11735	Northwest	$60,000.00	2012	2012-03-30 00:00:00.000	March	SO47736	1	$2,049.10
7	Accessories	All-Purpose Bike Stand	11735	Northwest	$60,000.00	2012	2012-03-30 00:00:00.000	March	SO47737	1	$2,071.42
8	Accessories	All-Purpose Bike Stand	11735	Northwest	$60,000.00	2012	2012-03-30 00:00:00.000	March	SO47738	1	$782.99
9	Accessories	All-Purpose Bike Stand	11845	Canada	$30,000.00	2012	2012-02-29 00:00:00.000	February	SO47460	1	$782.99
10	Accessories	All-Purpose Bike Stand	11845	Canada	$30,000.00	2012	2012-02-29 00:00:00.000	February	SO47461	1	$2,071.42
11	Accessories	All-Purpose Bike Stand	12056	Canada	$120,000.00	2012	2012-03-30 00:00:00.000	March	SO47731	1	$2,049.10
12	Accessories	All-Purpose Bike Stand	12056	Canada	$120,000.00	2012	2012-03-30 00:00:00.000	March	SO47732	1	$2,443.35
13	Accessories	All-Purpose Bike Stand	12056	Canada	$120,000.00	2012	2012-03-30 00:00:00.000	March	SO47733	1	$2,181.56
14	Accessories	All-Purpose Bike Stand	12056	Canada	$120,000.00	2012	2012-03-30 00:00:00.000	March	SO47734	1	$782.99
15	Accessories	All-Purpose Bike Stand	12056	Canada	$120,000.00	2012	2012-03-30 00:00:00.000	March	SO47735	1	$782.99
16	Accessories	All-Purpose Bike Stand	12056	Canada	$120,000.00	2012	2012-03-30 00:00:00.000	March	SO47736	1	$2,049.10
17	Accessories	All-Purpose Bike Stand	12056	Canada	$120,000.00	2012	2012-03-30 00:00:00.000	March	SO47737	1	$2,071.42

Query executed successfully.

ITS-ALBRS-DB1.SEATTLEU.EDU ... | SEATTLEU\ecampbell (79) | AdventureWorksDW2017 | 00:26:40 | 36,890 rows

The Second business requirement was then to evaluate resellers while performing a similar analysis to internet sales; and ensure to include the following variables in the query results: Category, Model, Calendar Year, Fiscal Year, Month, Sales Orde Number, Sales Order Quantity, and Sales Order Amount. Once again, the Dimdate table and its date-key could prove useful in linking tables and I decided again to place the subqueries in the FROM Clause. The DimDate and Fact Finance tables were joined as a root linked table using the date-key and then joined with reseller table and applicable fields. Then, I inner-joined dates and reseller information with a sub-query selecting business requisite fields. Recognizing that business requirements required no customer or geographic data, I then grouped requisite business requirements query resulted in 51,204 rows in approximately 3 seconds. Analyst experimented with casting fiscal date values from Fact Finance table to year unsuccessfully, thus the whole fiscal date is present instead of just fiscal year.

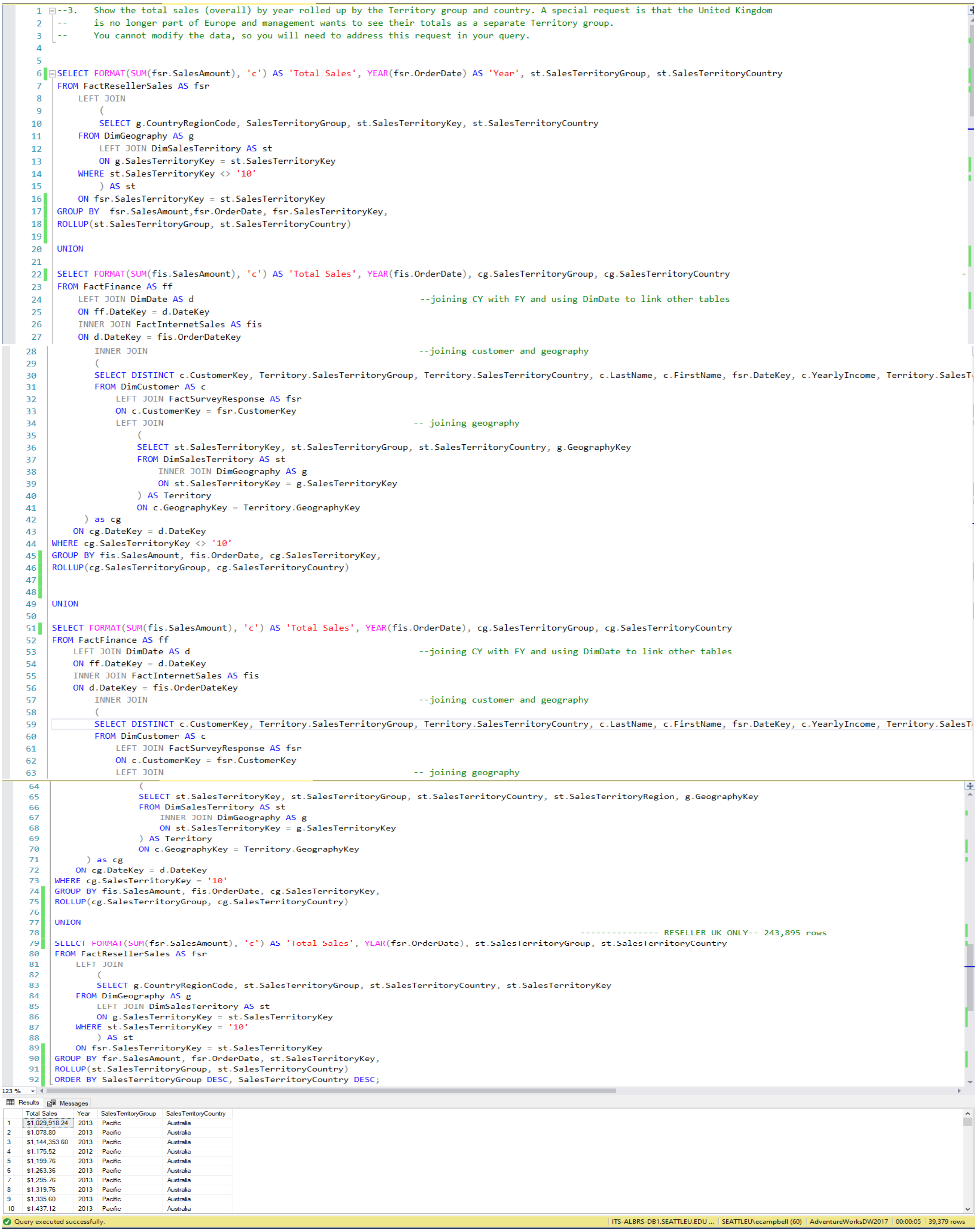


```sql
--2.    Provide a similar analysis for Reseller sales with the following columns
-- (Category, Model, CalendarYear, FiscalYear, Month, OrderNumber, Quantity, Amount).


SELECT Prod.EnglishProductSubcategoryName AS 'Category', Prod.ModelName AS 'Model', d.CalendarYear, ff.Date , seller.SalesOrderNumber,
       seller.OrderQuantity, FORMAT(seller.SalesAmount, 'c') AS 'Total Sale Amount'

FROM DimDate AS d
    INNER JOIN FactFinance AS ff
    ON d.DateKey = ff.DateKey
    RIGHT JOIN
        (
        SELECT rs.Productkey, rs.SalesOrderNumber,  rs.OrderQuantity, rs.SalesAmount, rs.OrderDateKey
        FROM FactResellerSales AS rs
        ) AS seller
    ON seller.OrderDateKey = d.DateKey-- 65,531,868
    INNER JOIN
        (
        SELECT psc.EnglishProductSubcategoryName, p.ModelName, p.ProductKey
        FROM DimProductCategory AS pc
        INNER JOIN DimProductSubcategory AS psc
        ON pc.ProductCategoryKey = psc.ProductCategoryKey
        INNER JOIN DimProduct AS p
        ON psc.ProductSubcategoryKey = p.ProductSubcategoryKey
        ) AS Prod
    ON seller.ProductKey = Prod.ProductKey

GROUP BY Prod.EnglishProductSubcategoryName, Prod.ModelName, d.CalendarYear, ff.Date , seller.SalesOrderNumber,
       seller.OrderQuantity, seller.SalesAmount
ORDER BY Category ASC;
```

	Category	Model	CalendarYear	Date	SalesOrderNumber	OrderQuantity	Total Sale Amount
1	Bib-Shorts	Men's Bib-Shorts	2011	2011-12-29 00:00:00.000	SO46604	2	$107.99
2	Bib-Shorts	Men's Bib-Shorts	2011	2011-12-29 00:00:00.000	SO46604	3	$161.98
3	Bib-Shorts	Men's Bib-Shorts	2011	2011-12-29 00:00:00.000	SO46604	5	$269.97
4	Bib-Shorts	Men's Bib-Shorts	2011	2011-12-29 00:00:00.000	SO46605	3	$161.98
5	Bib-Shorts	Men's Bib-Shorts	2011	2011-12-29 00:00:00.000	SO46608	3	$161.98
6	Bib-Shorts	Men's Bib-Shorts	2011	2011-12-29 00:00:00.000	SO46608	8	$431.95
7	Bib-Shorts	Men's Bib-Shorts	2011	2011-12-29 00:00:00.000	SO46608	9	$485.95
8	Bib-Shorts	Men's Bib-Shorts	2011	2011-12-29 00:00:00.000	SO46611	2	$107.99
9	Bib-Shorts	Men's Bib-Shorts	2011	2011-12-29 00:00:00.000	SO46611	3	$161.98
10	Bib-Shorts	Men's Bib-Shorts	2011	2011-12-29 00:00:00.000	SO46614	3	$161.98
11	Bib-Shorts	Men's Bib-Shorts	2011	2011-12-29 00:00:00.000	SO46614	7	$377.96
12	Bib-Shorts	Men's Bib-Shorts	2011	2011-12-29 00:00:00.000	SO46614	9	$485.95
13	Bib-Shorts	Men's Bib-Shorts	2011	2011-12-29 00:00:00.000	SO46616	1	$53.99
14	Bib-Shorts	Men's Bib-Shorts	2011	2011-12-29 00:00:00.000	SO46616	7	$377.96

```
Query executed successfully.    ITS-ALBRS-DB1.SEATTLEU.EDU ... | SEATTLEU\ecampbell (59) | AdventureWorksDW2017 | 00:00:03 | 51,204 rows
```

Next business requirement led to the examination of total sales by year being rolled up by Territory Group and Country fields, but with UK being segregated from rollup and not being able to change data types. Inability to change data and wanting UK by itself had me thinking UNION. Thus, the aggregation of data was completed by using a union of internet sales and reseller sales that did not include UK data; and then doing the same for just UK data. I created queries for reseller sales and internet sales not excluding UK, and then just the UK. Then a union was used to aggregate UK Data from internet sales and reseller sales. A total of 39,379 rows resulted repeatedly in less than 10 seconds, but there are too many as the sales are not summed for a total using the Group By and Rollup by Sales Territory and Country. Analyst will have to ask supplementary questions about business requirement. A snapshot of code used is below to examine what went right and what analyst missing.

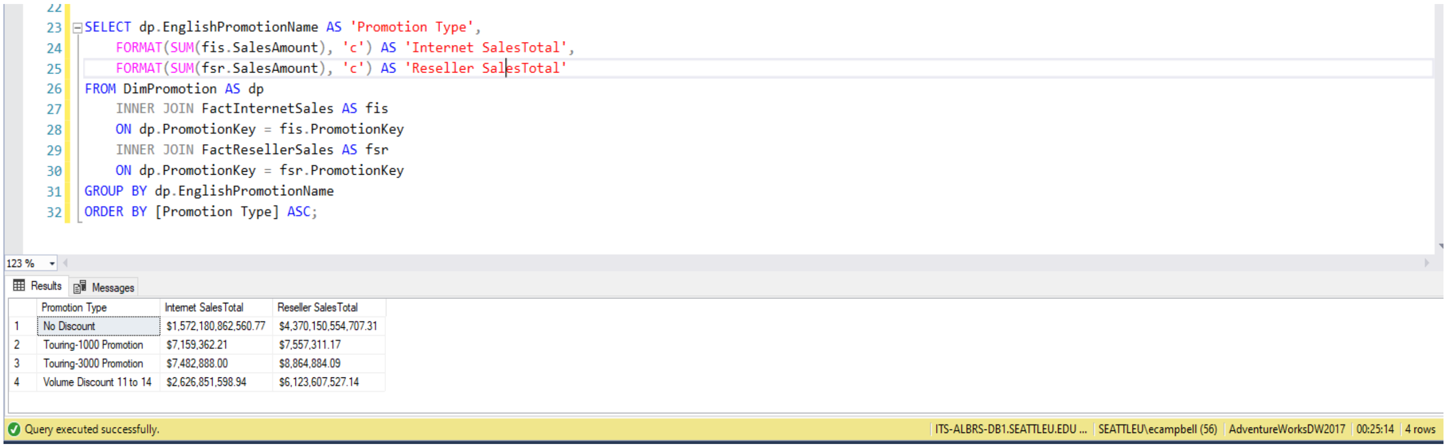


```sql
--3.   Show the total sales (overall) by year rolled up by the Territory group and country. A special request is that the United Kingdom
--     is no longer part of Europe and management wants to see their totals as a separate Territory group.
--     You cannot modify the data, so you will need to address this request in your query.


SELECT FORMAT(SUM(fsr.SalesAmount), 'c') AS 'Total Sales', YEAR(fsr.OrderDate) AS 'Year', st.SalesTerritoryGroup, st.SalesTerritoryCountry
FROM FactResellerSales AS fsr
    LEFT JOIN
        (
        SELECT g.CountryRegionCode, SalesTerritoryGroup, st.SalesTerritoryKey, st.SalesTerritoryCountry
    FROM DimGeography AS g
        LEFT JOIN DimSalesTerritory AS st
        ON g.SalesTerritoryKey = st.SalesTerritoryKey
    WHERE st.SalesTerritoryKey <> '10'
        ) AS st
    ON fsr.SalesTerritoryKey = st.SalesTerritoryKey
GROUP BY  fsr.SalesAmount,fsr.OrderDate, fsr.SalesTerritoryKey,
ROLLUP(st.SalesTerritoryGroup, st.SalesTerritoryCountry)

UNION

SELECT FORMAT(SUM(fis.SalesAmount), 'c') AS 'Total Sales', YEAR(fis.OrderDate), cg.SalesTerritoryGroup, cg.SalesTerritoryCountry
FROM FactFinance AS ff
    LEFT JOIN DimDate AS d                                           --joining CY with FY and using DimDate to link other tables
    ON ff.DateKey = d.DateKey
    INNER JOIN FactInternetSales AS fis
    ON d.DateKey = fis.OrderDateKey
        INNER JOIN                                                   --joining customer and geography
        (
        SELECT DISTINCT c.CustomerKey, Territory.SalesTerritoryGroup, Territory.SalesTerritoryCountry, c.LastName, c.FirstName, fsr.DateKey, c.YearlyIncome, Territory.SalesT
        FROM DimCustomer AS c
            LEFT JOIN FactSurveyResponse AS fsr
            ON c.CustomerKey = fsr.CustomerKey
            LEFT JOIN                                                -- joining geography
                (
                SELECT st.SalesTerritoryKey, st.SalesTerritoryGroup, st.SalesTerritoryCountry, g.GeographyKey
                FROM DimSalesTerritory AS st
                    INNER JOIN DimGeography AS g
                    ON st.SalesTerritoryKey = g.SalesTerritoryKey
                ) AS Territory
                ON c.GeographyKey = Territory.GeographyKey
      ) as cg
    ON cg.DateKey = d.DateKey
WHERE cg.SalesTerritoryKey <> '10'
GROUP BY fis.SalesAmount, fis.OrderDate, cg.SalesTerritoryKey,
ROLLUP(cg.SalesTerritoryGroup, cg.SalesTerritoryCountry)


UNION

SELECT FORMAT(SUM(fis.SalesAmount), 'c') AS 'Total Sales', YEAR(fis.OrderDate), cg.SalesTerritoryGroup, cg.SalesTerritoryCountry
FROM FactFinance AS ff
    LEFT JOIN DimDate AS d                                           --joining CY with FY and using DimDate to link other tables
    ON ff.DateKey = d.DateKey
    INNER JOIN FactInternetSales AS fis
    ON d.DateKey = fis.OrderDateKey
        INNER JOIN                                                   --joining customer and geography
        (
        SELECT DISTINCT c.CustomerKey, Territory.SalesTerritoryGroup, Territory.SalesTerritoryCountry, c.LastName, c.FirstName, fsr.DateKey, c.YearlyIncome, Territory.SalesT
        FROM DimCustomer AS c
            LEFT JOIN FactSurveyResponse AS fsr
            ON c.CustomerKey = fsr.CustomerKey
            LEFT JOIN                                                -- joining geography
                (
                SELECT st.SalesTerritoryKey, st.SalesTerritoryGroup, st.SalesTerritoryCountry, st.SalesTerritoryRegion, g.GeographyKey
                FROM DimSalesTerritory AS st
                    INNER JOIN DimGeography AS g
                    ON st.SalesTerritoryKey = g.SalesTerritoryKey
                ) AS Territory
                ON c.GeographyKey = Territory.GeographyKey
      ) as cg
    ON cg.DateKey = d.DateKey
WHERE cg.SalesTerritoryKey = '10'
GROUP BY fis.SalesAmount, fis.OrderDate, cg.SalesTerritoryKey,
ROLLUP(cg.SalesTerritoryGroup, cg.SalesTerritoryCountry)

UNION
                                                                                               --------------- RESELLER UK ONLY-- 243,895 rows
SELECT FORMAT(SUM(fsr.SalesAmount), 'c') AS 'Total Sales', YEAR(fsr.OrderDate), st.SalesTerritoryGroup, st.SalesTerritoryCountry
FROM FactResellerSales AS fsr
    LEFT JOIN
        (
        SELECT g.CountryRegionCode, st.SalesTerritoryGroup, st.SalesTerritoryCountry, st.SalesTerritoryKey
    FROM DimGeography AS g
        LEFT JOIN DimSalesTerritory AS st
        ON g.SalesTerritoryKey = st.SalesTerritoryKey
    WHERE st.SalesTerritoryKey = '10'
        ) AS st
    ON fsr.SalesTerritoryKey = st.SalesTerritoryKey
GROUP BY fsr.SalesAmount, fsr.OrderDate, st.SalesTerritoryKey,
ROLLUP(st.SalesTerritoryGroup, st.SalesTerritoryCountry)
ORDER BY SalesTerritoryGroup DESC, SalesTerritoryCountry DESC;
```

	Total Sales	Year	SalesTerritoryGroup	SalesTerritoryCountry
1	$1,029,918.24	2013	Pacific	Australia
2	$1,078.80	2013	Pacific	Australia
3	$1,144,353.60	2013	Pacific	Australia
4	$1,175.52	2012	Pacific	Australia
5	$1,199.76	2013	Pacific	Australia
6	$1,263.36	2013	Pacific	Australia
7	$1,295.76	2013	Pacific	Australia
8	$1,319.76	2013	Pacific	Australia
9	$1,335.60	2013	Pacific	Australia
10	$1,437.12	2013	Pacific	Australia

```
Query executed successfully.    ITS-ALBRS-DB1.SEATTLEU.EDU ... | SEATTLEU\ecampbell (60) | AdventureWorksDW2017 | 00:00:05 | 39,379 rows
```

The next business requirement was to evaluate internet sales and sales done by resellers by promotion and product. I wanted to look at promotional type in relation to sales made by our resellers and internet. Tables used were Promotion, internet sales and reseller sales and no complex joins completed. Emphasis of EDA was evaluating both internet sales and reseller sales under normal operations without a discount and with a promotion or discount. The query produced a result with 4 rows and gave us a view of sales with no discount those with promotions, but took a bit of time to run at 25 minutes. As stated earlier, performing a subquery or performing query outside of normal business hours might be prudent. Further EDA might would be to add fields from the Sales Reason and Internet sales Reason, products, customers, and geography tables and evaluate for seasonal and geographic trends in relation to promotion, but inventory forecasting, and management associated with trends.

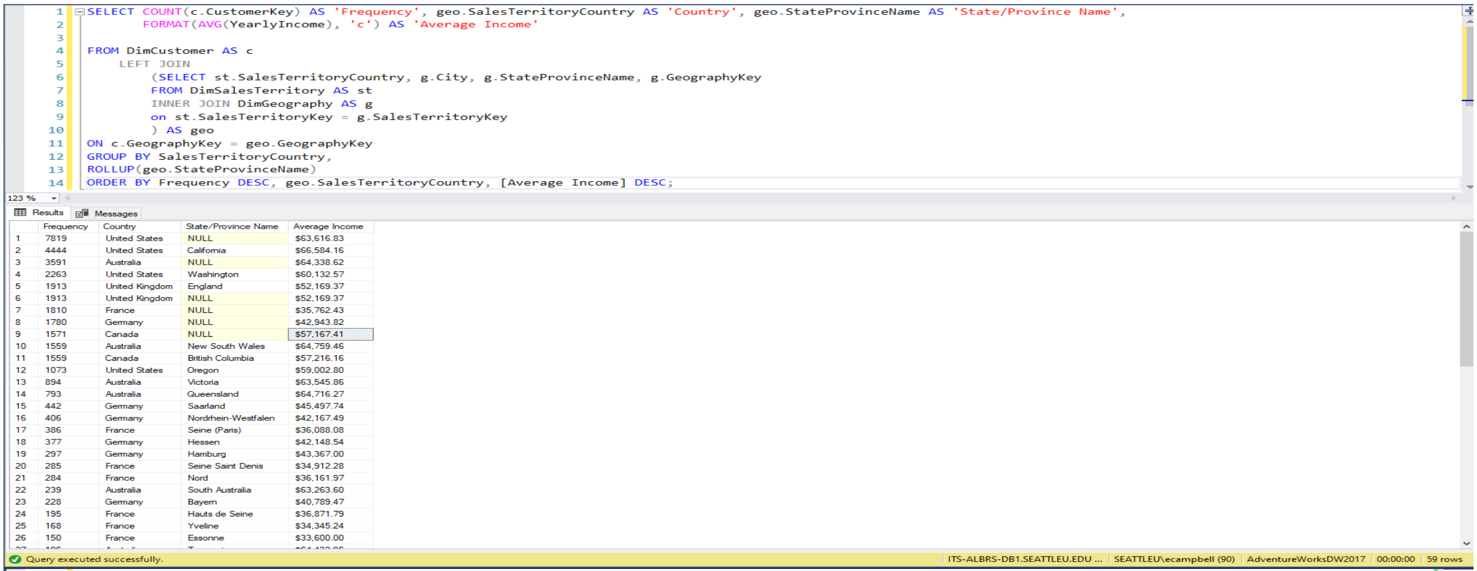


```sql
SELECT dp.EnglishPromotionName AS 'Promotion Type',
    FORMAT(SUM(fis.SalesAmount), 'c') AS 'Internet SalesTotal',
    FORMAT(SUM(fsr.SalesAmount), 'c') AS 'Reseller SalesTotal'
FROM DimPromotion AS dp
    INNER JOIN FactInternetSales AS fis
    ON dp.PromotionKey = fis.PromotionKey
    INNER JOIN FactResellerSales AS fsr
    ON dp.PromotionKey = fsr.PromotionKey
GROUP BY dp.EnglishPromotionName
ORDER BY [Promotion Type] ASC;
```

	Promotion Type	Internet SalesTotal	Reseller SalesTotal
1	No Discount	$1,572,180,862,560.77	$4,370,150,554,707.31
2	Touring-1000 Promotion	$7,159,362.21	$7,557,311.17
3	Touring-3000 Promotion	$7,482,888.00	$8,864,884.09
4	Volume Discount 11 to 14	$2,626,851,598.94	$6,123,607,527.14

```
Query executed successfully.    ITS-ALBRS-DB1.SEATTLEU.EDU ... | SEATTLEU\ecampbell (56) | AdventureWorksDW2017 | 00:25:14 | 4 rows
```

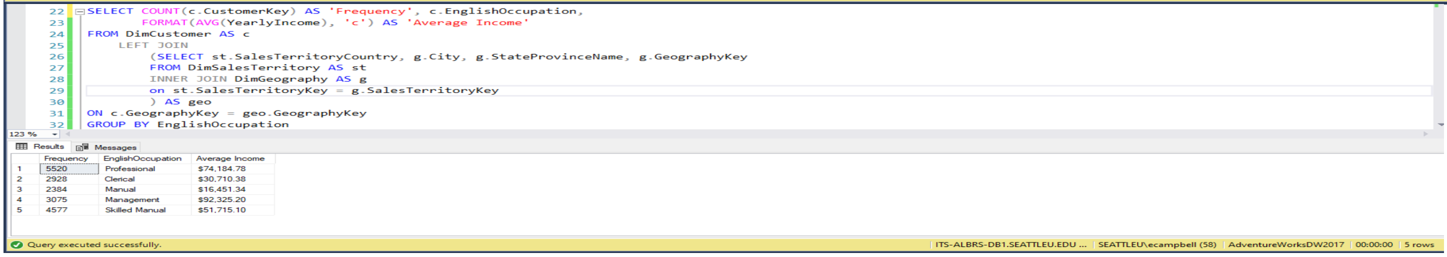
The final business opportunity was evaluating the firm's customer demographics. The customer table(s) were joined with tables relating to geography and general summary statistics was performed. Summary statistics provide a glimpse into who target market and provide further research opportunities to explore methods of effectively reaching repeat customers with discretionary income. Thus, a query focusing on identifying customer clusters by geography, education, their income, and occupation was developed.



```sql
SELECT COUNT(c.CustomerKey) AS 'Frequency', geo.SalesTerritoryCountry AS 'Country', geo.StateProvinceName AS 'State/Province Name',
       FORMAT(AVG(YearlyIncome), 'c') AS 'Average Income'

FROM DimCustomer AS c
    LEFT JOIN
        (SELECT st.SalesTerritoryCountry, g.City, g.StateProvinceName, g.GeographyKey
        FROM DimSalesTerritory AS st
        INNER JOIN DimGeography AS g
        on st.SalesTerritoryKey = g.SalesTerritoryKey
        ) AS geo
ON c.GeographyKey = geo.GeographyKey
GROUP BY SalesTerritoryCountry,
ROLLUP(geo.StateProvinceName)
ORDER BY Frequency DESC, geo.SalesTerritoryCountry, [Average Income] DESC;
```

	Frequency	Country	State/Province Name	Average Income
1	7819	United States	NULL	$63,616.83
2	4444	United States	California	$66,584.16
3	3591	Australia	NULL	$64,338.62
4	2263	United States	Washington	$60,132.57
5	1913	United Kingdom	England	$52,169.37
6	1913	United Kingdom	NULL	$52,169.37
7	1810	France	NULL	$35,762.43
8	1780	Germany	NULL	$42,943.82
9	1571	Canada	NULL	$57,167.41
10	1559	Australia	New South Wales	$64,759.46
11	1559	Canada	British Columbia	$57,216.16
12	1073	United States	Oregon	$59,002.80
13	894	Australia	Victoria	$63,545.86
14	793	Australia	Queensland	$64,716.27
15	442	Germany	Saarland	$45,497.74
16	406	Germany	Nordrhein-Westfalen	$42,167.49
17	386	France	Seine (Paris)	$36,088.08
18	377	Germany	Hessen	$42,148.54
19	297	Germany	Hamburg	$43,367.00
20	285	France	Seine Saint Denis	$34,912.28
21	284	France	Nord	$36,161.97
22	239	Australia	South Australia	$63,263.60
23	228	Germany	Bayern	$40,789.47
24	195	France	Hauts de Seine	$36,871.79
25	168	France	Yveline	$34,345.24
26	150	France	Essonne	$33,600.00

```
Query executed successfully.    ITS-ALBRS-DB1.SEATTLEU.EDU ... | SEATTLEU\ecampbell (90) | AdventureWorksDW2017 | 00:00:00 | 59 rows
```



```sql
SELECT COUNT(c.CustomerKey) AS 'Frequency', c.EnglishOccupation,
       FORMAT(AVG(YearlyIncome), 'c') AS 'Average Income'
FROM DimCustomer AS c
    LEFT JOIN
        (SELECT st.SalesTerritoryCountry, g.City, g.StateProvinceName, g.GeographyKey
        FROM DimSalesTerritory AS st
        INNER JOIN DimGeography AS g
        on st.SalesTerritoryKey = g.SalesTerritoryKey
        ) AS geo
ON c.GeographyKey = geo.GeographyKey
GROUP BY EnglishOccupation
```

	Frequency	EnglishOccupation	Average Income
1	5520	Professional	$74,184.78
2	2928	Clerical	$30,710.38
3	2384	Manual	$16,451.34
4	3075	Management	$92,325.20
5	4577	Skilled Manual	$51,715.10

```
Query executed successfully.    ITS-ALBRS-DB1.SEATTLEU.EDU ... | SEATTLEU\ecampbell (58) | AdventureWorksDW2017 | 00:00:00 | 5 rows
```