Gabe Boucaud

OMSBA 5140

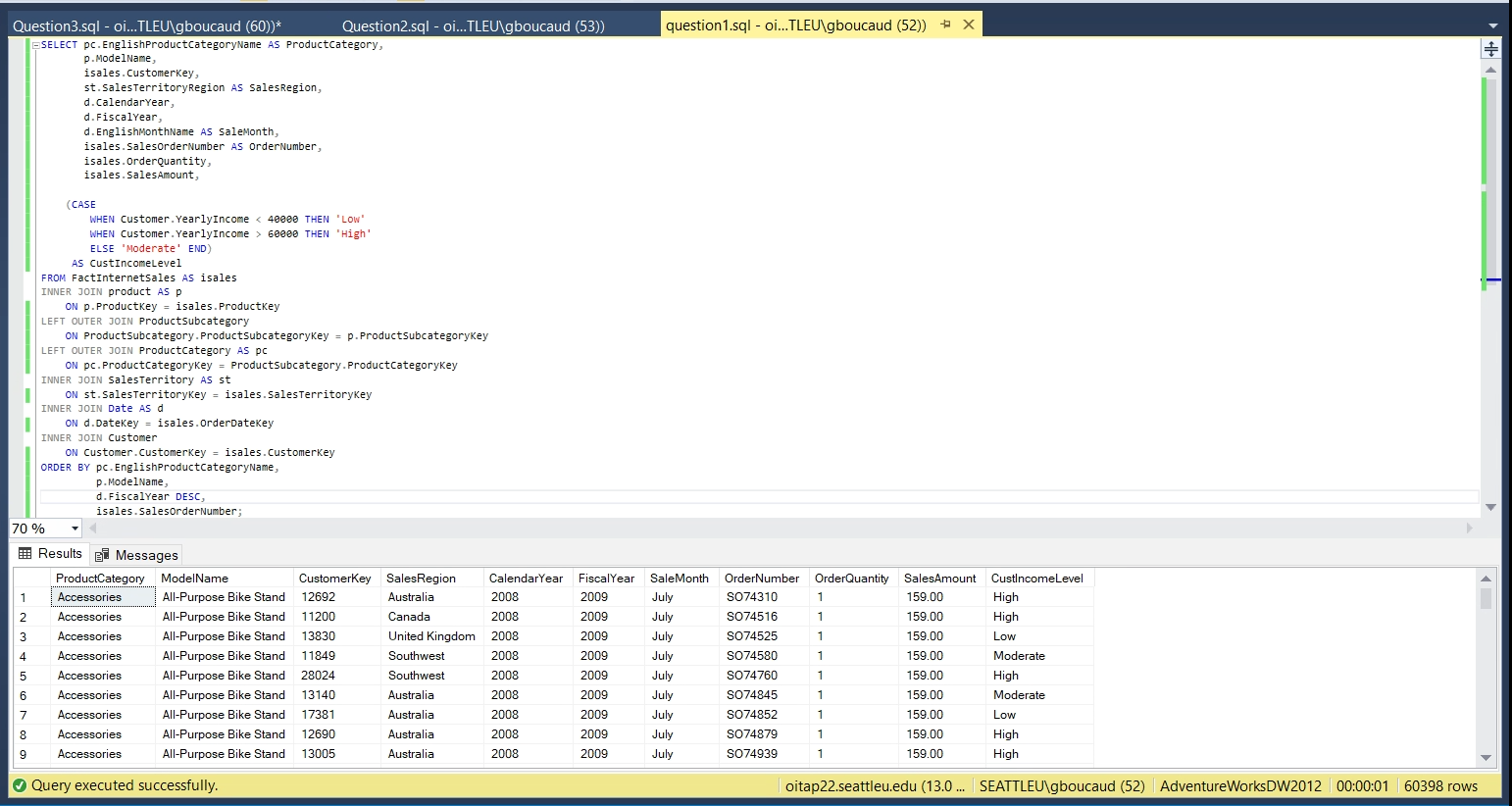
Snodgrass

Data Translation Challenge Technical Report

**Technical Report**

**Provide a detailed list of Internet sales with the following columns for the financial analyst team to review (Category, Model, CustomerKey, Region, IncomeGroup, CalendarYear, FiscalYear, Month, OrderNumber, Quantity, and Amount). Income group should categorize the people based on "Low" being less than 40,000, "High" being greater than 60,000, and the rest will be "Moderate".**

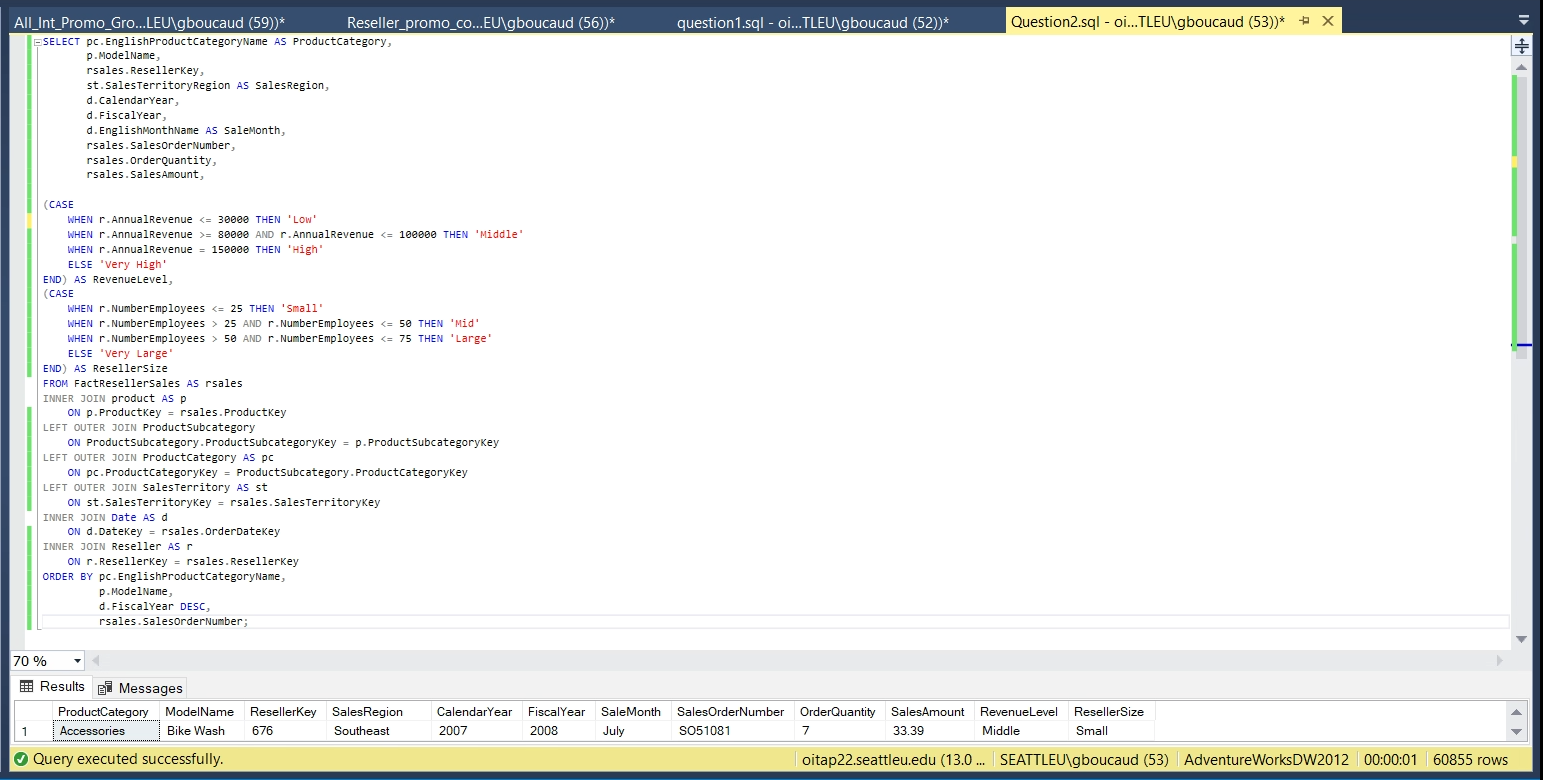
To gain more insight I also tried ordering this query by the SalesAmount and CustomerIncomeLevel, but neither of these variables gave substantial evidence that customers with high income placed more expensive orders.



* To extract the relevant results for this query I was required to extract information from the Internet Sales, Product, Product Subcategory, Product Category, Sales Territory, Date and Customer tables.
* To account for products that may not have been assigned a Subcategory or Category I used a “Left Join”, but for all other tables used an “Inner Join” because there cannot be orders without an assigned product, date, and customer key.
* I used a CASE function to properly assign customers to their respective income levels from the customer table.

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

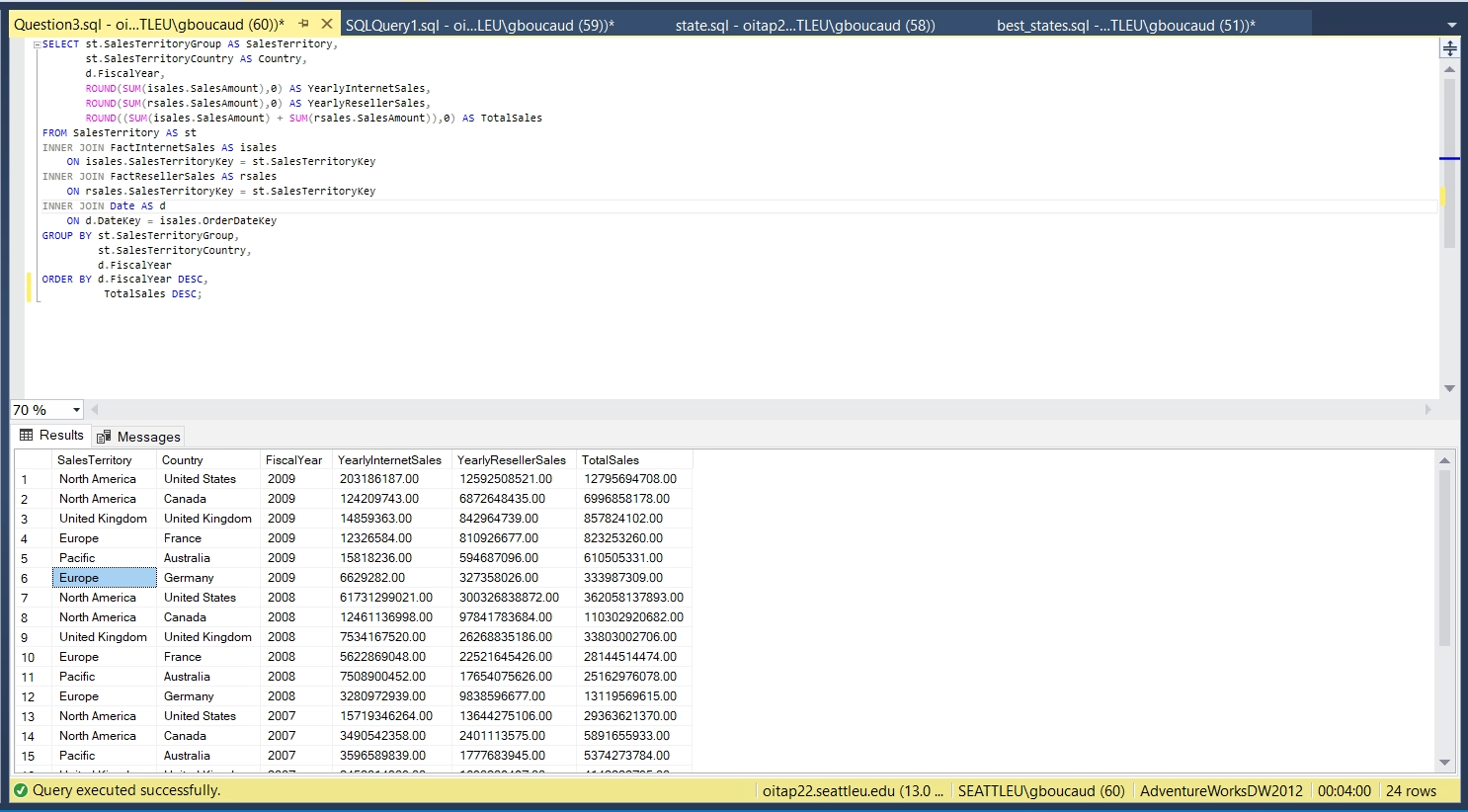
Similar to the above query not much insight can be gained from looking at individual orders.



* I took the same approach to the above query to extract similar data on the sales of our resellers.
* To show the difference in reseller company size and revenue level of our resellers I used the CASE function to assign these categories to every reseller.

**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.**

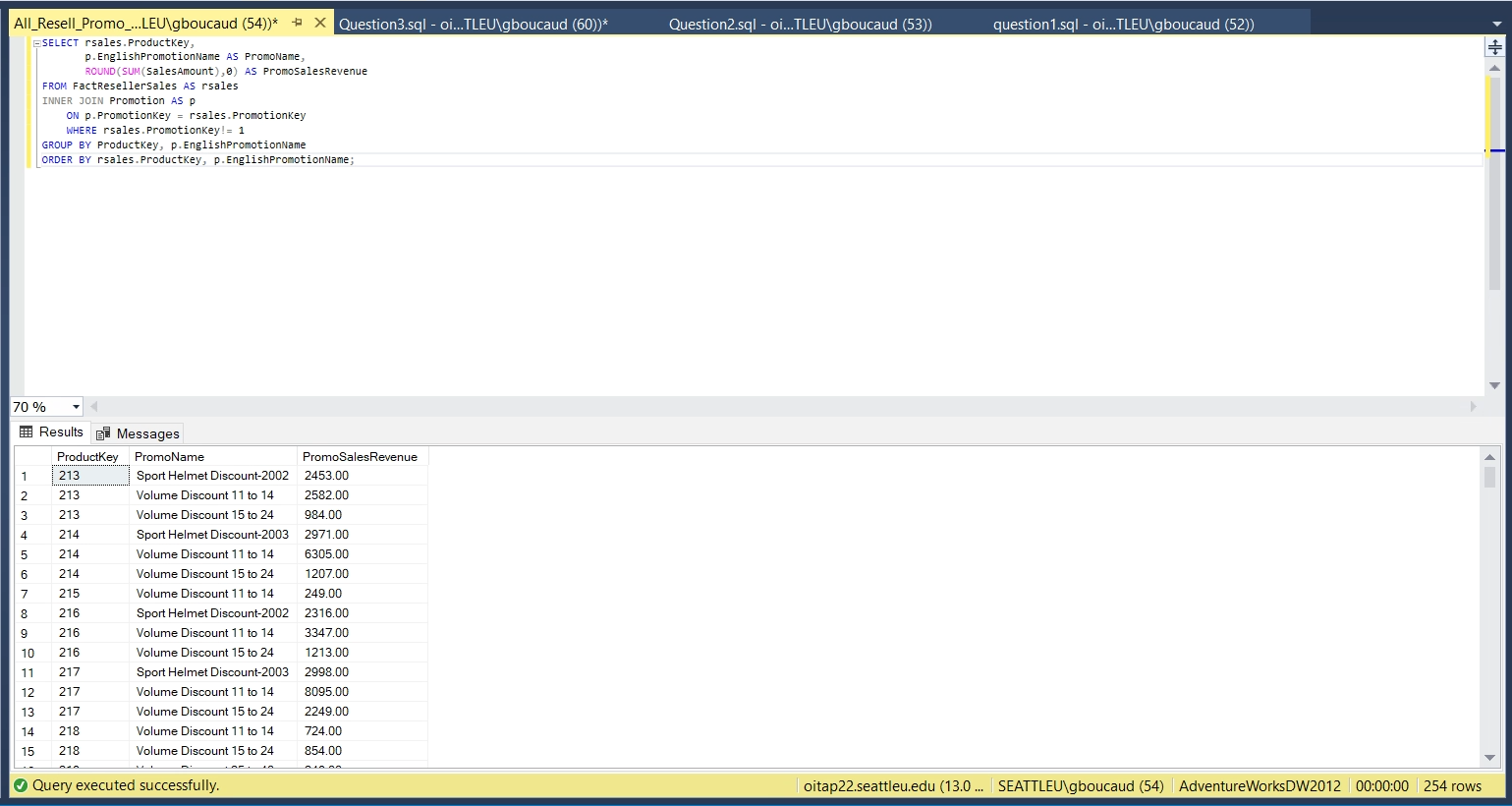
Here we see that the US and Canada have the most total sales for each year. In 2006 & 2007 Australia was the third most popular region and since then the United Kingdom has surpassed them in total sales.



* Simply joining all existing sales data for each sales territory and grouping them by their territory, country, and fiscal year.
* The list is ordered by year and then total sales so finance can easily break-up the information by fiscal year.

**Promotion Analysis**

The query below shows all products and reseller promotions that have been assigned to them along with the sales revenue for each promotion



--Average reseller sales for promotion items

SELECT rsales.ProductKey,

promo.EnglishPromotionName AS PromoName,

prod.EnglishProductName AS ProductName,

prod.ModelName,

prodcat.EnglishProductCategoryName AS CategoryName,

(CASE

WHEN DATEDIFF(month, promo.StartDate, promo.EndDate) = 0 THEN 1

ELSE DATEDIFF(month, promo.StartDate, promo.EndDate)

END) AS MonthsOfPromo,

--Finding the number of months each promo went on for. If it was less than a month listed as a single month

ROUND(SUM(rsales.OrderQuantity), 0) AS TotalQuantSold, --total order quantity of each product for every promotion

promo.DiscountPct AS PercentDiscount,

ROUND(SUM(rsales.SalesAmount), 0) AS PromoRevenue, --total product sales of each promotion for each product

(SUM(rsales.OrderQuantity) /

(CASE

WHEN DATEDIFF(month, promo.StartDate, promo.EndDate) = 0 THEN 1

ELSE DATEDIFF(month, promo.StartDate, promo.EndDate)

END)) AS MonthlyQuantSold,

--Average monthly order quantity of each product under each promotion

(ROUND(SUM(rsales.SalesAmount) /

(CASE

WHEN DATEDIFF(month, promo.StartDate, promo.EndDate) = 0 THEN 1

ELSE DATEDIFF(month, promo.StartDate, promo.EndDate)

END), 0)) AS MonthlyPromoRevenue

FROM FactResellerSales AS rsales

--Average monthly sales of each product under each promotion

INNER JOIN Promotion AS promo

ON promo.PromotionKey = rsales.PromotionKey

INNER JOIN product AS prod

ON prod.ProductKey = rsales.ProductKey

LEFT OUTER JOIN ProductSubcategory

ON ProductSubcategory.ProductCategoryKey = prod.ProductSubcategoryKey

LEFT OUTER JOIN ProductCategory AS prodcat

ON prodcat.ProductCategoryKey = ProductSubcategory.ProductCategoryKey

--Joins to get all columns in SELECT

WHERE (rsales.PromotionKey != 1) --Filtering out the orders that have no promotional discounts

GROUP BY rsales.ProductKey,

promo.EnglishPromotionName,

prod.EnglishProductName,

promo.DiscountPct,

prod.ModelName,

prodcat.EnglishProductCategoryName,

DATEDIFF(month, promo.StartDate, promo.EndDate)

HAVING

(SUM(rsales.SalesAmount) /

(CASE

WHEN DATEDIFF(month, promo.StartDate, promo.EndDate) = 0 THEN 1

ELSE DATEDIFF(month, promo.StartDate, promo.EndDate)

END)) >

--Only show the products and promotions that generated more avg monthly orders than products without promotions in subquery below

(SELECT (nondiscount.TotalSales / nondiscount.Months) AS NonDiscountMonthlyRevenue

FROM

((SELECT rsales2.ProductKey, --Select ProductKey to group by a similar key value as above

(SUM(rsales2.SalesAmount)) AS TotalSales, --Total order quantities for each product

DATEDIFF(month, promo2.StartDate, promo2.EndDate) AS Months --Total months for the No Discount promotion

FROM FactResellerSales AS rsales2

INNER JOIN product AS prod2

ON prod2.ProductKey = rsales2.ProductKey

INNER JOIN Promotion AS promo2

ON promo2.PromotionKey = rsales2.PromotionKey

WHERE (rsales2.PromotionKey = 1) --Only show products in the are "No Discount" promotion

AND (rsales2.ProductKey = rsales.ProductKey)

--Only show matching Products from the outer query

GROUP BY rsales2.ProductKey,

DATEDIFF(month, promo2.StartDate, promo2.EndDate))) AS nondiscount

WHERE (nondiscount.ProductKey = rsales.ProductKey)

--Group by, alias and match subquery with outer query

GROUP BY nondiscount.ProductKey,

nondiscount.TotalSales,

nondiscount.Months)

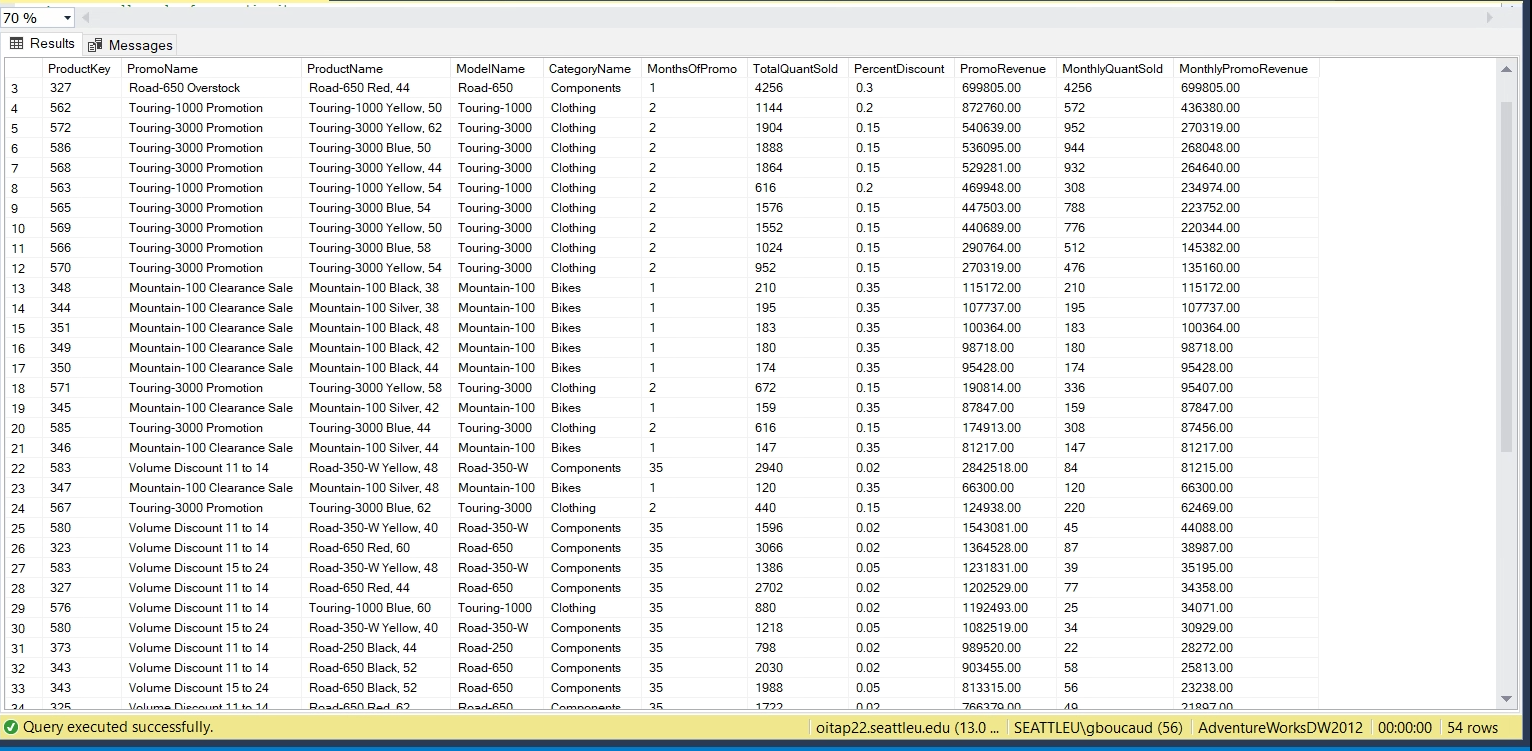
--Subquery to get the average monthly sales of products that have no promotions and discounts

ORDER BY rsales.ProductKey,

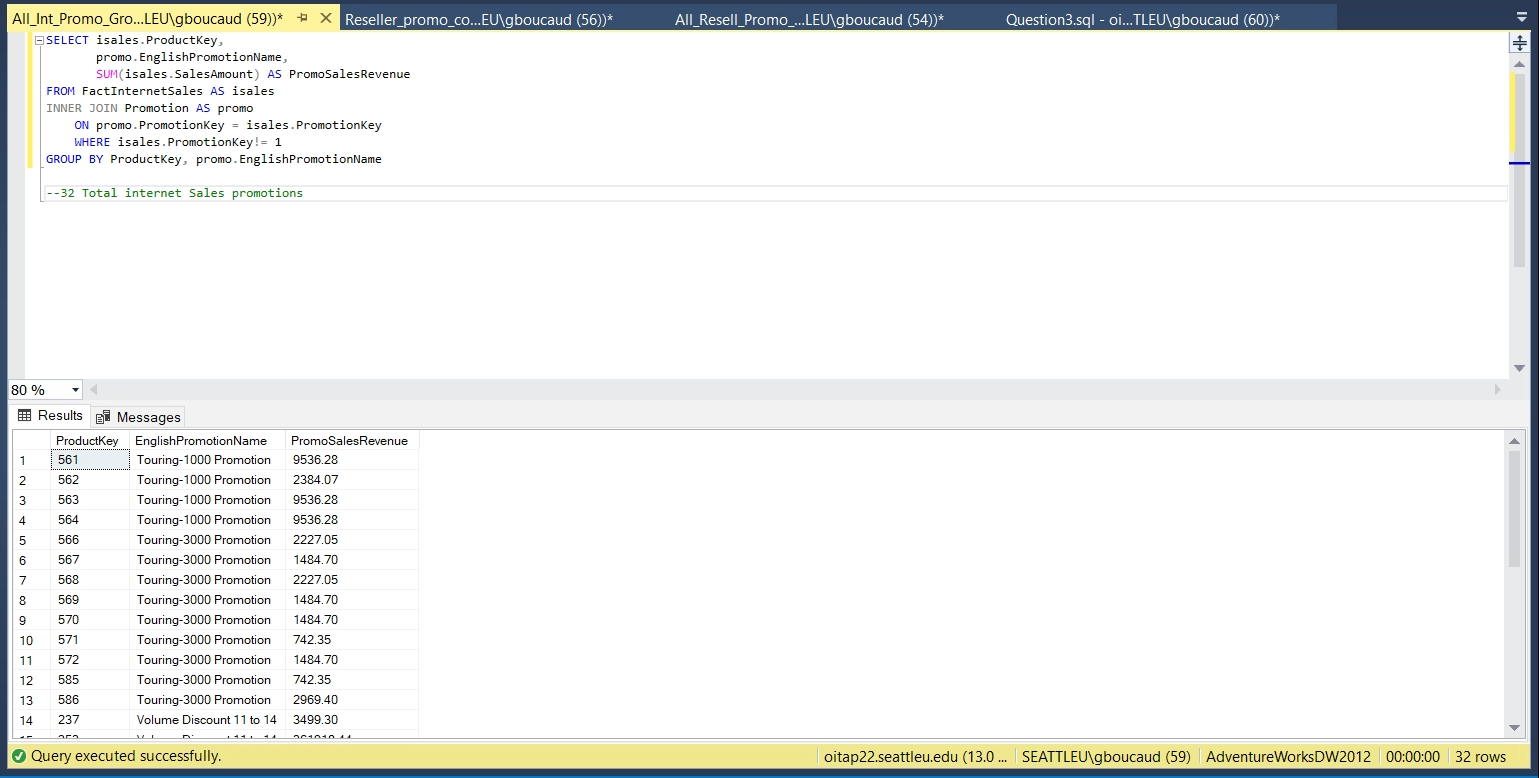
promo.EnglishPromotionName,

MonthlyPromoRevenue DESC;

This query shows us a list of all products and their promotions that generated an average monthly revenue greater than the average monthly revenue of the same products without the use of promotions. Notice that the above query of all products and their promotions extracted 254 rows while this only extracted 54 which is only 21% of all reseller promotions. This provides us with significant insight about the products their promotions that were and were not successful in driving revenue for our company. The promotions that were short and had significant discounts not the only promotions that were profitable as 20 of the 35-month promotions with discounts ranging from 2% to 5% produced greater monthly revenue with promotions than without promotions. From this we have an idea of which products and promotions will be effective going forward.



* In writing this query I wanted to create a standardized metric to measure revenue over the duration of each promotion which is why I created a column for the average monthly revenue for products purchased with promotions. I was able to easily compare this to our reseller sales that had no discounts or promotions because there was a promotion key and date range for products with and without promotions. Because a discounted product would have to sell a greater quantity to generate more revenue it was not necessary to compare this by the monthly quantity sold.
* In finding the length of a promotion the promotion table was very helpful in giving information about the start and end date of each promotion. Having a promotion key for “no discount” was also helpful as this had a date range that I was also able to find the difference between to calculate the average revenue for those products without promotions.
* To compare the same products and promotions for those with and without discounts it was important to match the product keys within my subquery.



SELECT isales.ProductKey,

promo.EnglishPromotionName AS PromoName,

prod.EnglishProductName AS ProductName,

prod.ModelName,

prodcat.EnglishProductCategoryName AS CategoryName,

(CASE

WHEN DATEDIFF(month, promo.StartDate, promo.EndDate) = 0 THEN 1

ELSE DATEDIFF(month, promo.StartDate, promo.EndDate)

END) AS MonthsOfPromo,

ROUND(SUM(isales.OrderQuantity), 0) AS TotalQuantSold,

promo.DiscountPct AS PercentDiscount,

ROUND(SUM(isales.SalesAmount), 0) AS PromoRevenue,

(SUM(isales.OrderQuantity) /

(CASE

WHEN DATEDIFF(month, promo.StartDate, promo.EndDate) = 0 THEN 1

ELSE DATEDIFF(month, promo.StartDate, promo.EndDate)

END)) AS MonthlyQuantSold,

(ROUND(SUM(isales.SalesAmount) /

(CASE

WHEN DATEDIFF(month, promo.StartDate, promo.EndDate) = 0 THEN 1

ELSE DATEDIFF(month, promo.StartDate, promo.EndDate)

END), 0)) AS MonthlyPromoRevenue

FROM FactInternetSales AS isales

INNER JOIN Promotion AS promo

ON promo.PromotionKey = isales.PromotionKey

INNER JOIN product AS prod

ON prod.ProductKey = isales.ProductKey

LEFT OUTER JOIN ProductSubcategory

ON ProductSubcategory.ProductCategoryKey = prod.ProductSubcategoryKey

LEFT OUTER JOIN ProductCategory AS prodcat

ON prodcat.ProductCategoryKey = ProductSubcategory.ProductCategoryKey

WHERE (isales.PromotionKey != 1)

GROUP BY isales.ProductKey,

promo.EnglishPromotionName,

prod.EnglishProductName,

promo.DiscountPct,

prod.ModelName,

prodcat.EnglishProductCategoryName,

DATEDIFF(month, promo.StartDate, promo.EndDate)

HAVING

(SUM(isales.SalesAmount) /

(CASE

WHEN DATEDIFF(month, promo.StartDate, promo.EndDate) = 0 THEN 1

ELSE DATEDIFF(month, promo.StartDate, promo.EndDate)

END)) >

(SELECT (nondiscount.TotalSales / nondiscount.Months) AS NonDiscountRevenue

FROM

((SELECT isales2.ProductKey,

(SUM(isales2.SalesAmount)) AS TotalSales,

DATEDIFF(month, promo2.StartDate, promo2.EndDate) AS Months

FROM FactInternetSales AS isales2

INNER JOIN product AS prod2

ON prod2.ProductKey = isales2.ProductKey

INNER JOIN Promotion AS promo2

ON promo2.PromotionKey = isales2.PromotionKey

WHERE (isales2.PromotionKey = 1)

AND (isales2.ProductKey = isales2.ProductKey)

GROUP BY isales2.ProductKey,

DATEDIFF(month, promo2.StartDate, promo2.EndDate))) AS nondiscount

WHERE (nondiscount.ProductKey = isales.ProductKey)

GROUP BY nondiscount.ProductKey,

nondiscount.TotalSales,

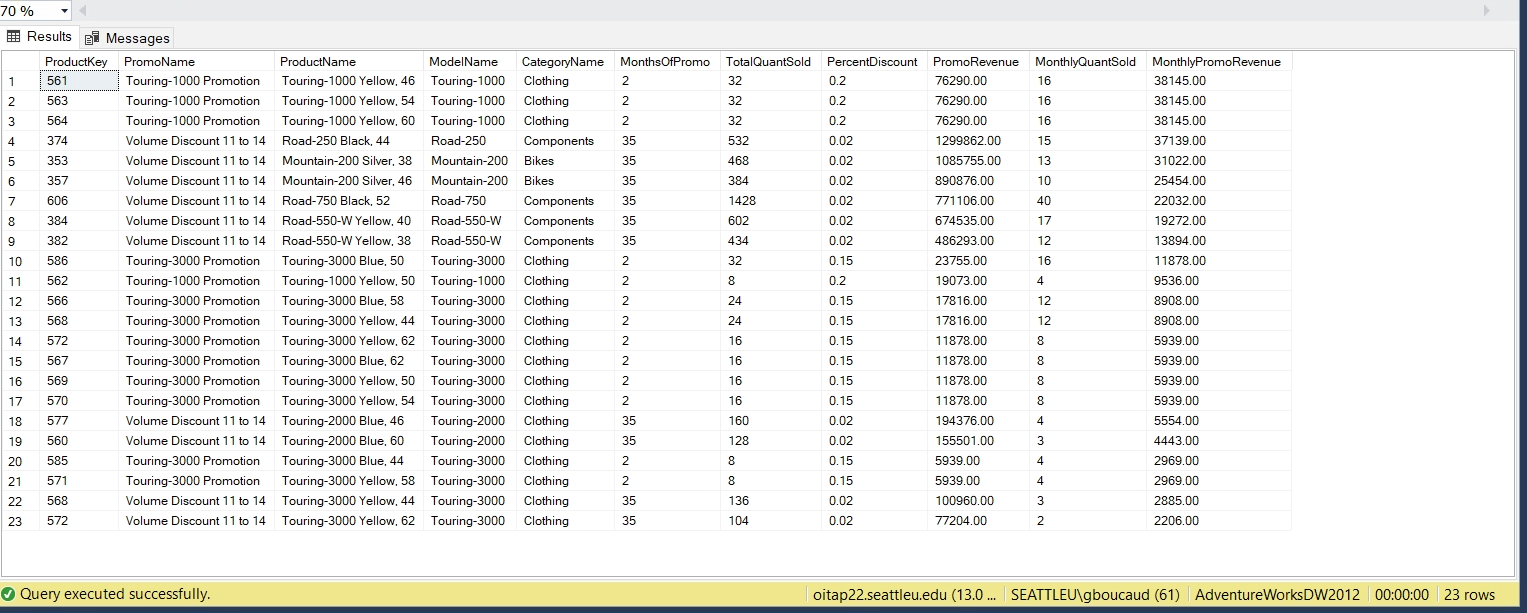
nondiscount.Months)

ORDER BY isales.ProductKey,

promo.EnglishPromotionName,

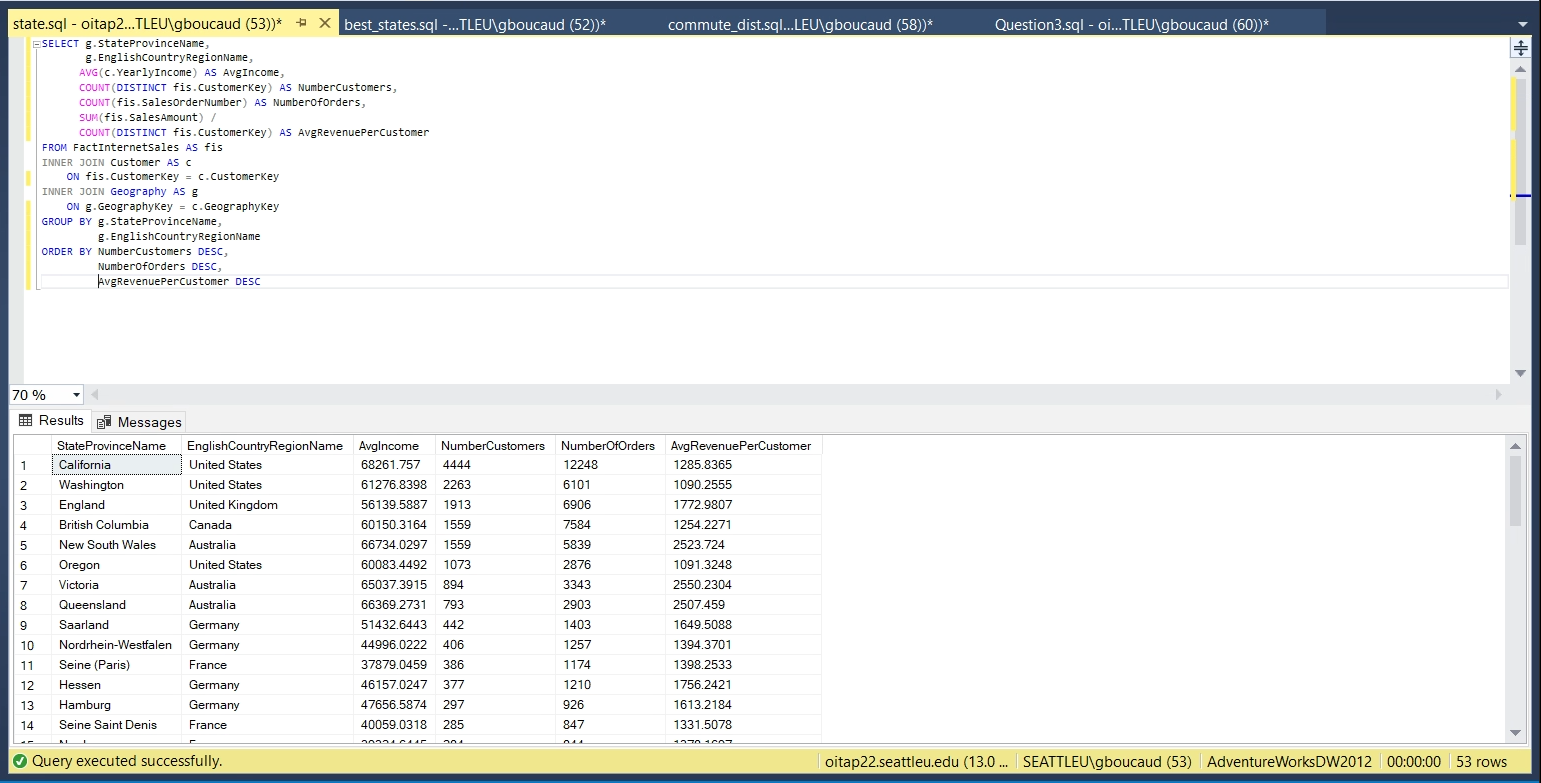
MonthlyPromoRevenue DESC;

I was able to conduct a similar analysis with our internet sales promotions and although there is a much smaller sample size of these promotions (32) there were still 9 ineffective promotions. With about 72% of our internet sales promotions driving higher average monthly revenue than the same products without promotions it would be smart to increase the amount of internet promotions we provide for our customers.



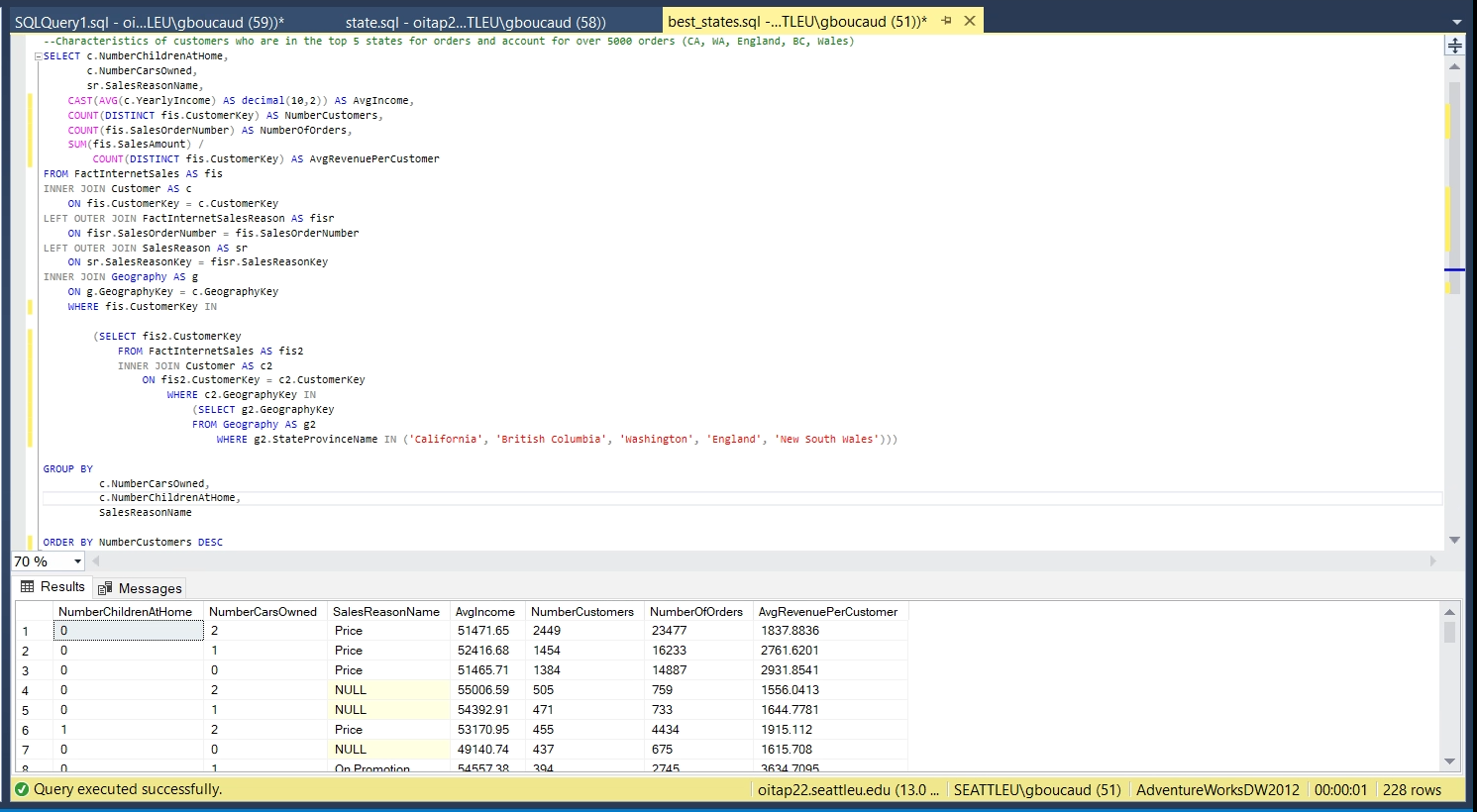
**Customer Analysis**

Here I wanted to get a general idea of the total number of customers and orders for each State/Province in our database was in addition to the average income in that state. Here we can see that California, Washington, England, British Columbia, and New South Wales clearly have the most customers and orders.



* I extracted the state/province names from the geography table
* Used expressions on data from the internet sales and orders of each state to obtain the number of customers, number of orders, and average revenue per customer in eachstate.

I was interested in the demographic makeup of the customers in the 5 states and provinces that contained the greatest number orders and customers. Getting a more in depth look of the customers in these states we can see that most customers had no children at home and had an average income lower than the average state incomes in the query above. The number of cars customers owned did not have much of an effect on the number of customers. I included the sales reasons to get an idea of why these customers chose to buy products from our company and I was surprised to see that promotions and advertisements where not very effective for these customers.



* Now that I knew which states had the most customers and number of orders, I wanted to look at the demographical makeup of the customers in the top performing states.
* I used the same expressions as the original query that created a list of each state, the average customer income, number of customers and number of orders.
* To make sure that all of the customer and order data in our list only contained information from the top 5 states I used a WHERE clause followed by a subquery to filter out any customers who were not in these states.

We see a similar pattern between number of children and average customer income if we only include data from next 5 states and provinces that have the most customers. It is important to note that the two German states/provinces of Saarland and Nordrehein-Westfalen have a lower average income than the averages shown in the query below.

