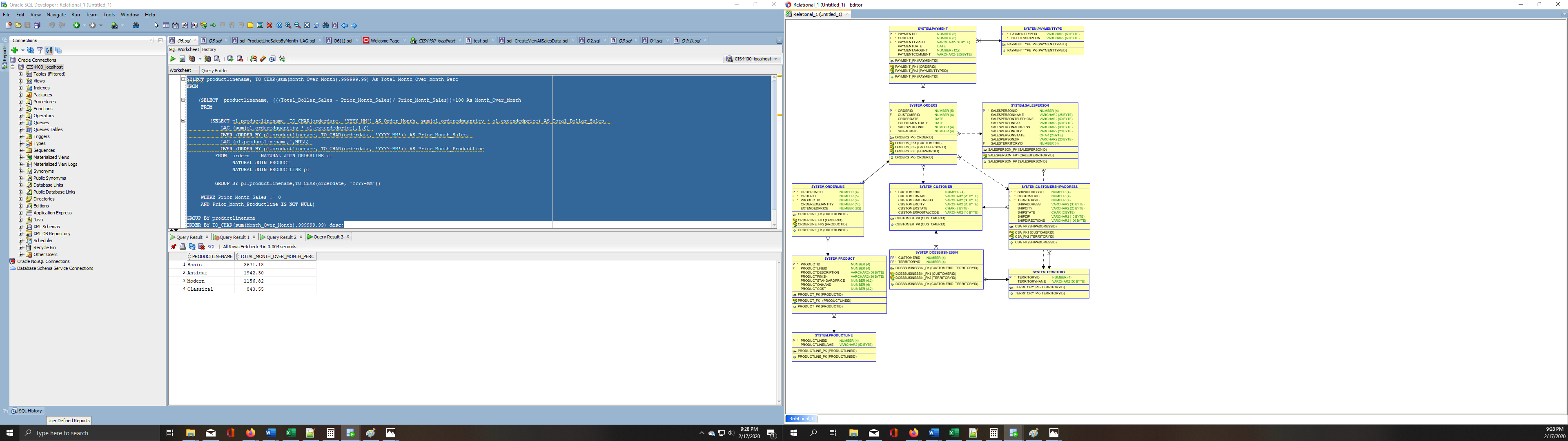
**ASSIGNMENT #1: ANALYTICAL SQL**

Richard Donald

CIS 4400 – CMWA

2/19/2020

**1)**



**2)**

SELECT cs.shipstate, TO\_CHAR(o.orderdate, 'YYYY-mm')AS Order\_Month\_Year,TO\_CHAR(SUM(ol.orderedquantity \* ol.extendedprice),'$9999999.99') AS Total\_Dollars

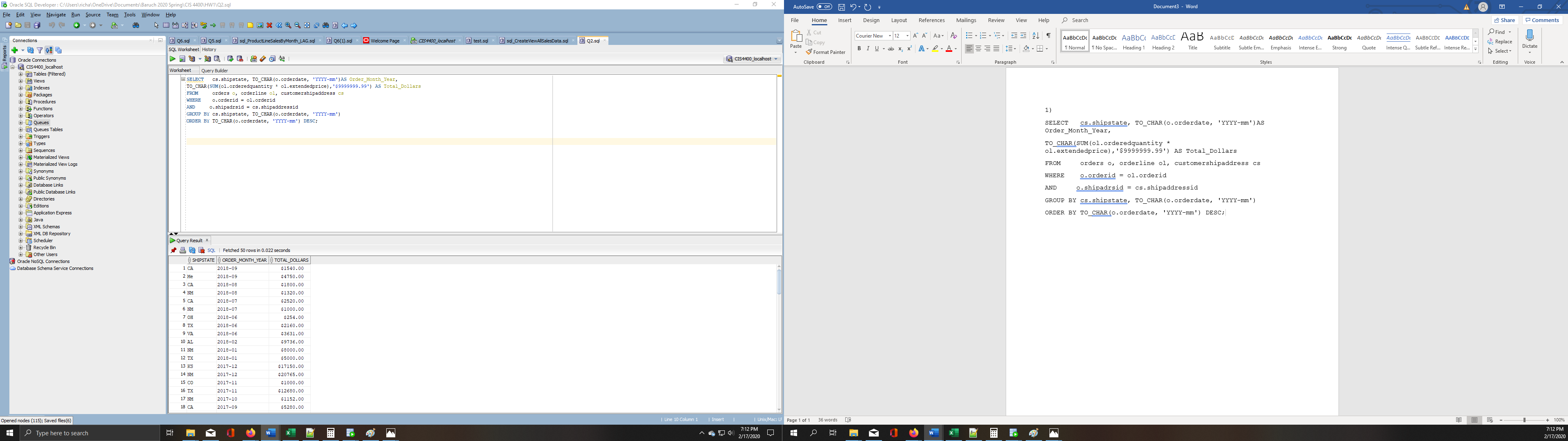
FROM orders o, orderline ol, customershipaddress cs

WHERE o.orderid = ol.orderid

AND o.shipadrsid = cs.shipaddressid

GROUP BY cs.shipstate, TO\_CHAR(o.orderdate, 'YYYY-mm')

ORDER BY TO\_CHAR(o.orderdate, 'YYYY-mm') DESC;



**3)**

SELECT c.customername, c.customeraddress, TO\_CHAR(SUM(ol.orderedquantity \* ol.extendedprice),'$99999.99') AS Total\_Dollars, RANK() OVER (ORDER BY SUM(ol.orderedquantity \* ol.extendedprice) DESC ) AS Order\_Rank

FROM product p, customer c, orderline ol, orders o

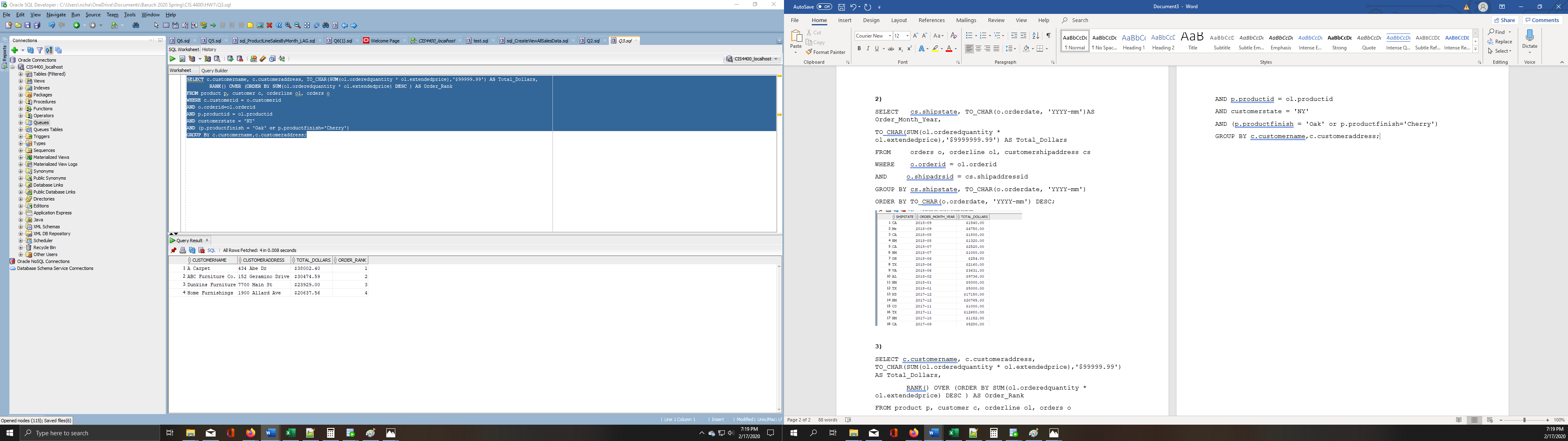
WHERE c.customerid = o.customerid

AND o.orderid=ol.orderid

AND p.productid = ol.productid

AND customerstate = 'NY'

AND (p.productfinish = 'Oak' or p.productfinish='Cherry')

GROUP BY c.customername,c.customeraddress;

**4)**

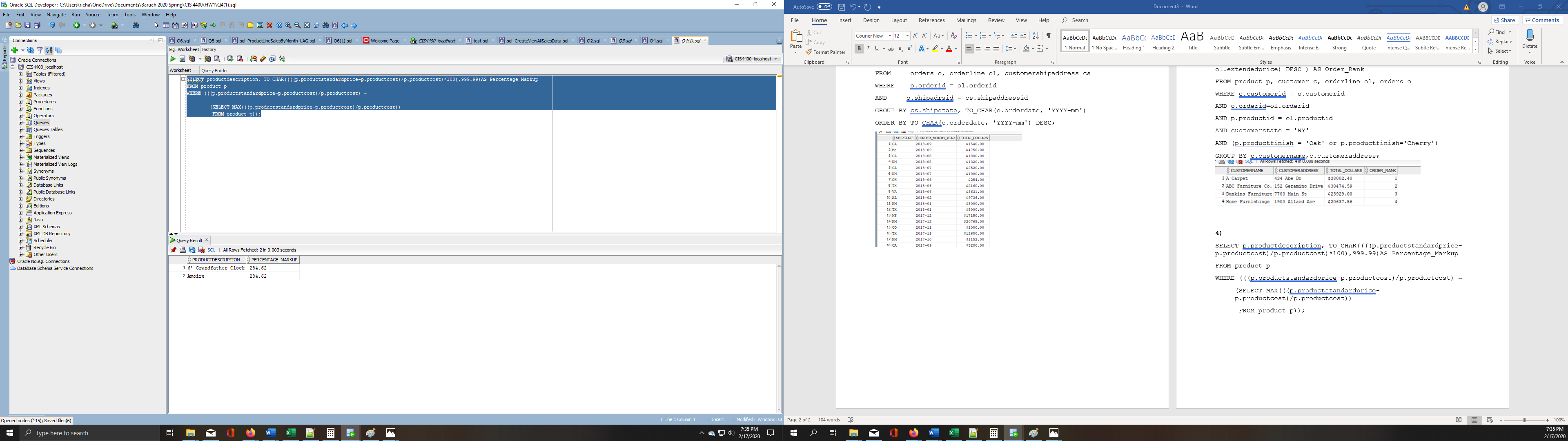
SELECT p.productdescription, TO\_CHAR((((p.productstandardprice- p.productcost)/p.productcost)\*100),999.99)AS Percentage\_Markup

FROM product p

WHERE (((p.productstandardprice-p.productcost)/p.productcost) =

(SELECT MAX(((p.productstandardprice-p.productcost)/p.productcost))

FROM product p));



**5)**

SELECT c.customername, TO\_CHAR(SUM(ol.orderedquantity \* ol.extendedprice) - SUM(ol.orderedquantity \* p.productcost),'$9999999.99') As Profit\_Generated,

TO\_CHAR((SUM(ol.orderedquantity \* ol.extendedprice) - SUM(ol.orderedquantity \* p.productcost)/SUM(ol.orderedquantity \* ol.extendedprice))/100,'9999.99') AS Total\_Gross\_Profit\_Margin,

RANK() OVER (ORDER BY (SUM(ol.orderedquantity \* ol.extendedprice) - SUM(ol.orderedquantity \* p.productcost)/SUM(ol.orderedquantity \* ol.extendedprice))/100 DESC) AS Customer\_Rank

FROM orderline ol , product p, customer c, orders o

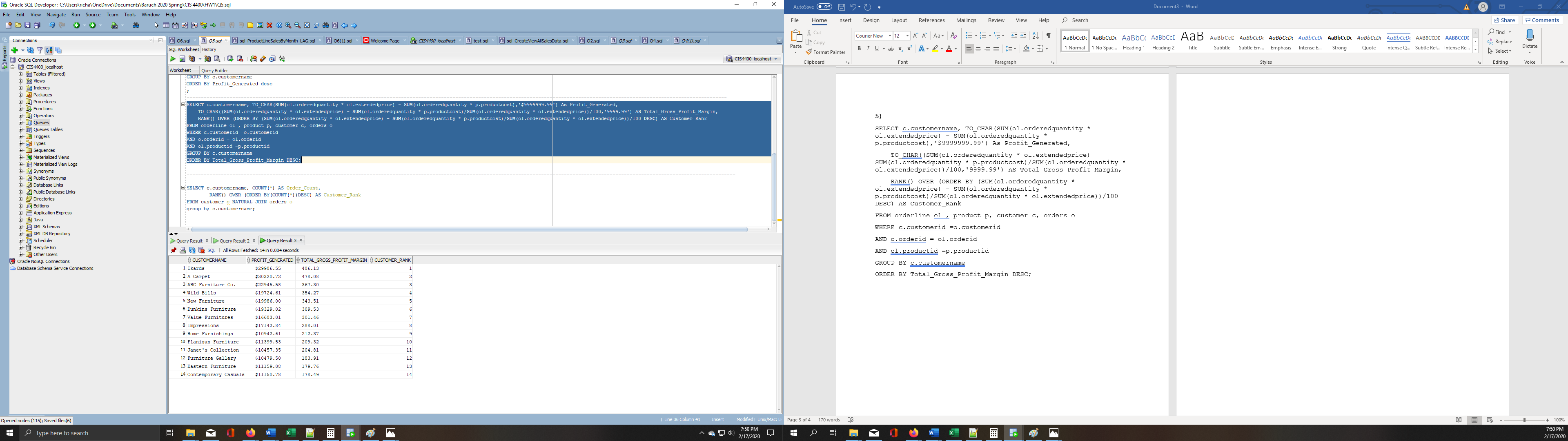
WHERE c.customerid = o.customerid

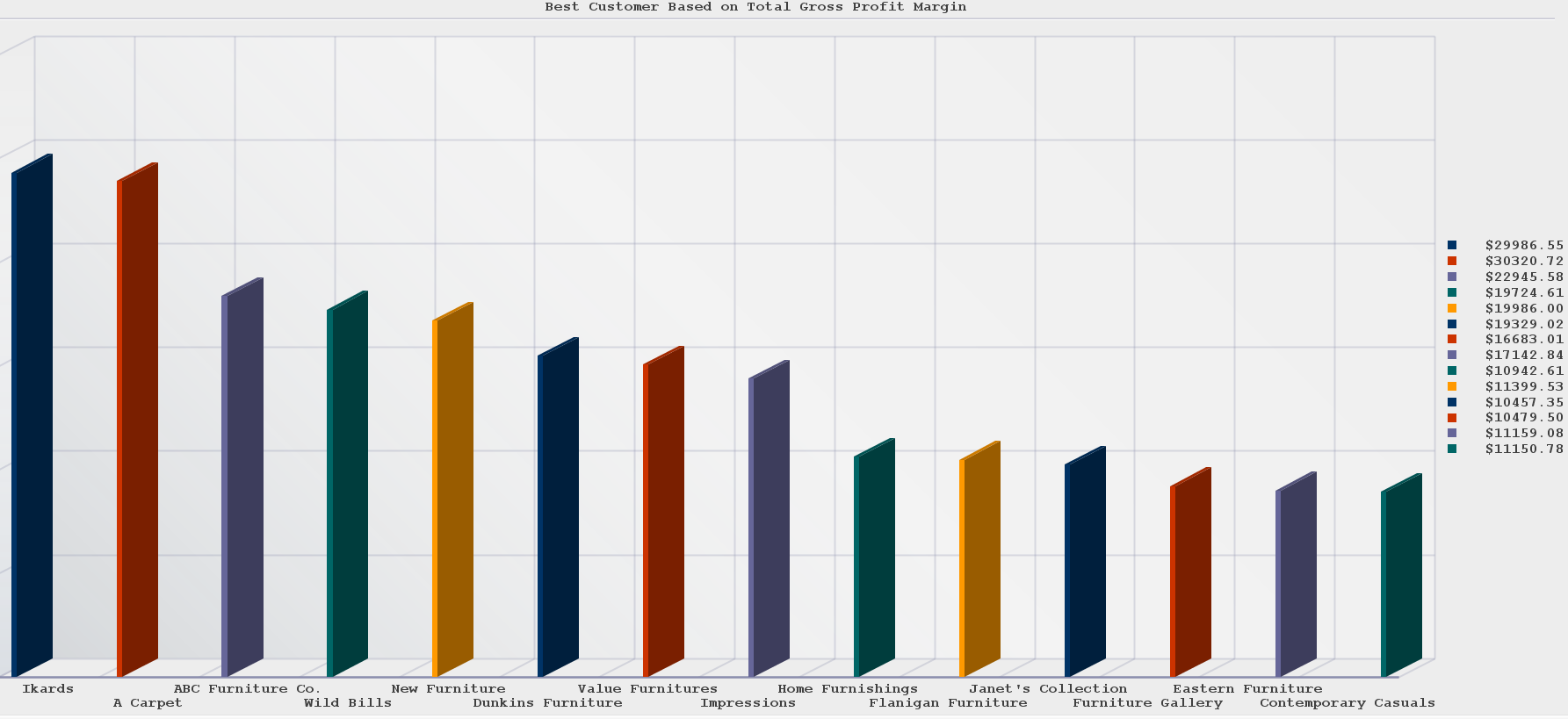
AND o.orderid = ol.orderid

AND ol.productid =p.productid

GROUP BY c.customername

ORDER BY Total\_Gross\_Profit\_Margin DESC;



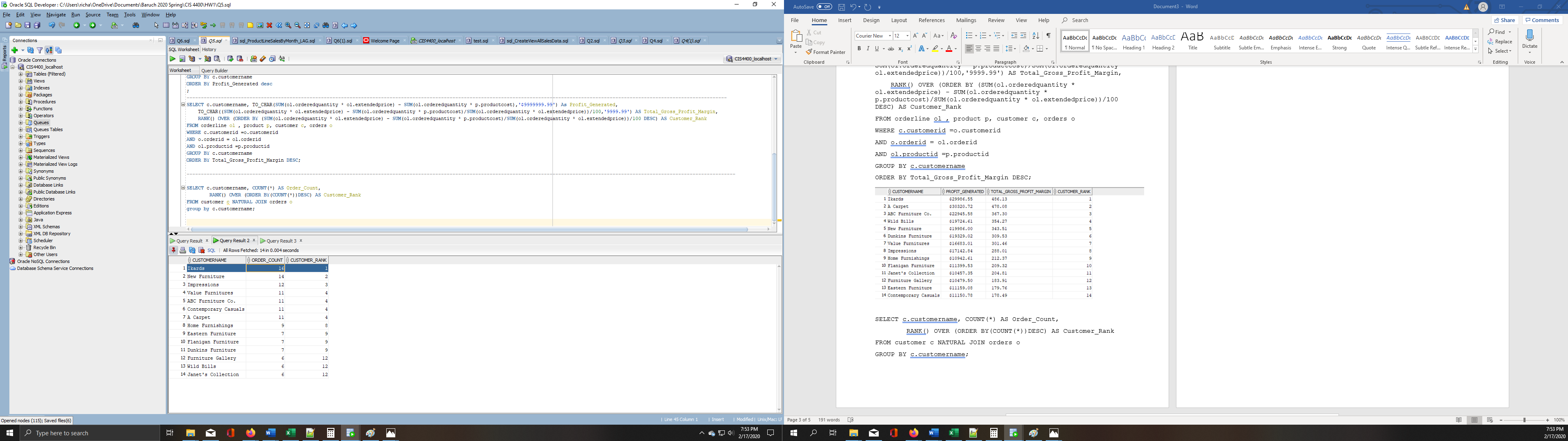


SELECT c.customername, COUNT(o.orderid) AS Order\_Count,

RANK() OVER (ORDER BY(COUNT(o.orderid))DESC) AS Customer\_Rank

FROM customer c NATURAL JOIN orders o

GROUP BY c.customername;

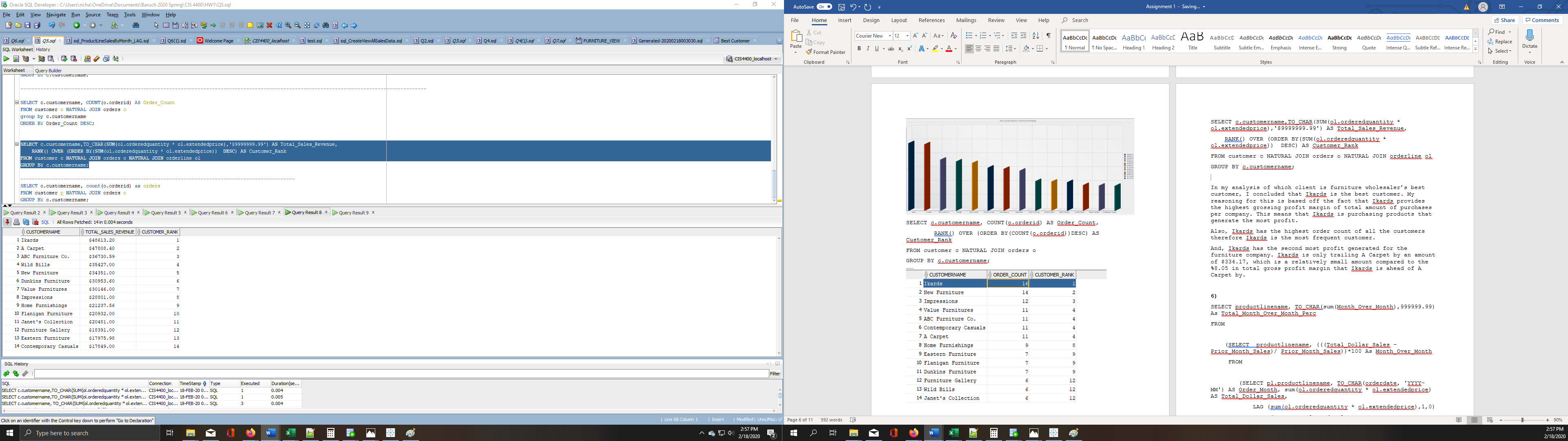


SELECT c.customername,TO\_CHAR(SUM(ol.orderedquantity \* ol.extendedprice),'$9999999.99') AS Total\_Sales\_Revenue,

RANK() OVER (ORDER BY(SUM(ol.orderedquantity \* ol.extendedprice)) DESC) AS Customer\_Rank

FROM customer c NATURAL JOIN orders o NATURAL JOIN orderline ol

GROUP BY c.customername;



In my analysis of which client is furniture wholesaler’s best customer, I concluded that Ikards is the best customer. My reasoning for this is based off the fact that Ikards provides the highest grossing profit margin of total amount of purchases per company. This means that Ikards is purchasing products that generate the most profit.

Also, Ikards has the highest order count of all the customers therefore Ikards is the most frequent customer.

And, Ikards has the second most profit generated for the furniture wholesaler. Ikards is only trailing A Carpet by an amount of $334.17, which is a relatively small amount compared to the %8.05 in total gross profit margin that Ikards is ahead of A Carpet by.

Additionally, Ikards has the highest total sales revenue per company.

Based off these three metrics (total sales, amount of orders, and profit margin) it is clear Ikards is the best customer

**6)**

SELECT productlinename, TO\_CHAR(sum(Month\_Over\_Month),999999.99) As Total\_Month\_Over\_Month\_Perc

FROM

(SELECT productlinename, (((Total\_Dollar\_Sales -Prior\_Month\_Sales)/ Prior\_Month\_Sales))\*100 As Month\_Over\_Month

FROM

(SELECT pl.productlinename, TO\_CHAR(orderdate, 'YYYY-MM') AS Order\_Month, sum(ol.orderedquantity \* ol.extendedprice) AS Total\_Dollar\_Sales,

LAG (sum(ol.orderedquantity \* ol.extendedprice),1,0)

OVER (ORDER BY pl.productlinename, TO\_CHAR(orderdate, 'YYYY-MM')) AS Prior\_Month\_Sales,

LAG (pl.productlinename,1,NULL)

OVER (ORDER BY pl.productlinename, TO\_CHAR(orderdate, 'YYYY-MM')) AS Prior\_Month\_ProductLine

FROM orders NATURAL JOIN ORDERLINE ol

NATURAL JOIN PRODUCT

NATURAL JOIN PRODUCTLINE pl

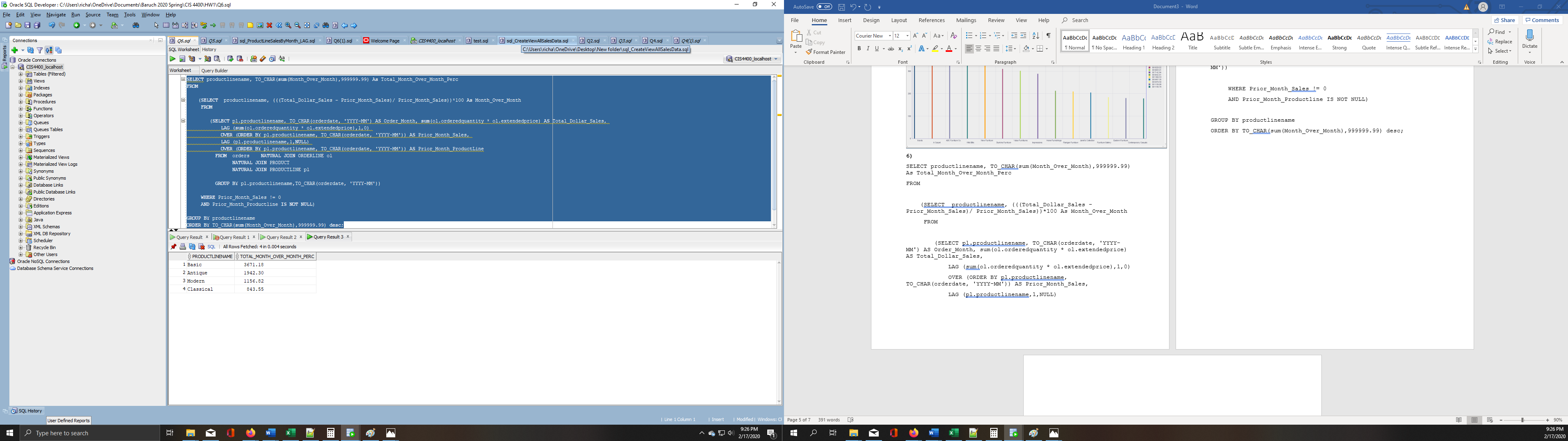
GROUP BY pl.productlinename,TO\_CHAR(orderdate, 'YYYY-MM'))

WHERE Prior\_Month\_Sales != 0

AND Prior\_Month\_Productline IS NOT NULL)

GROUP BY productlinename

ORDER BY TO\_CHAR(sum(Month\_Over\_Month),999999.99) desc;



‘Basic’ has the largest percentage of Month-Over-Month sales

**7)**

CREATE VIEW Furniture\_View AS

SELECT

o.orderid, o.customerid, o.orderdate, o.fulfillmentdate, o.shipadrsid, ol.orderlineid, ol.productid, ol.orderedquantity, ol.extendedprice,SUM(ol.orderedquantity \* ol.extendedprice) AS Total\_Sales, p.productlineid, p.productdescription, p.productfinish, p.productstandardprice, p.productonhand, p.productcost, pl.productlinename, cs.shipaddressid, cs.territoryid, cs.shipaddress, cs.shipcity,cs.shipstate, cs.shipzip, cs.shipdirections, sp.salespersonid, sp.salespersonname,sp.salespersontelephone, sp.salespersonfax, sp.salespersonaddress, sp.salespersoncity, sp.salespersonstate,sp.salespersonzip, sp.salesterritoryid, t.territoryname, c.customername, c.customeraddress, c.customercity, c.customerstate, c.customerpostalcode

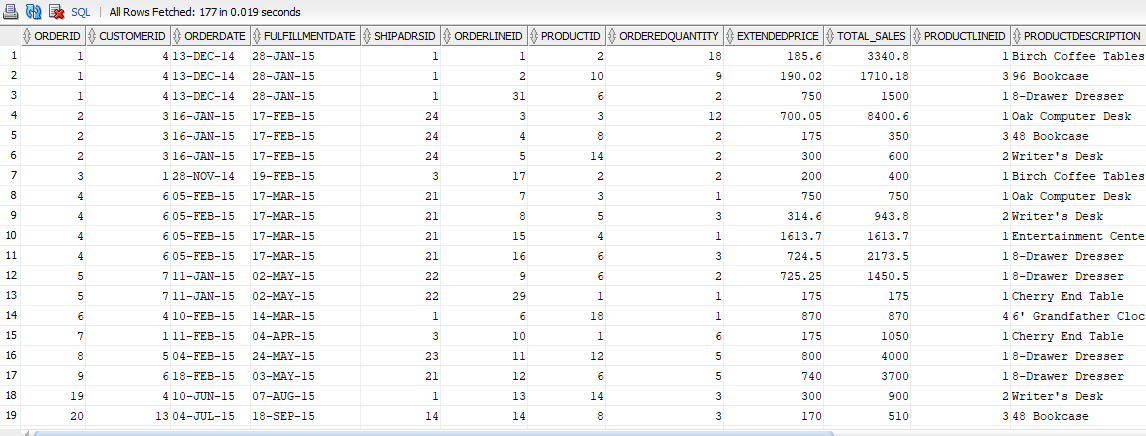
FROM

orderline ol INNER JOIN orders o ON ol.orderid = o.orderid INNER JOIN customer c ON o.customerid = c.customerid INNER JOIN customershipaddress cs ON o.customerid = cs.customerid AND o.shipadrsid = cs.shipaddressid LEFT OUTER JOIN salesperson sp ON cs.territoryid = sp.saleFursterritoryid AND o.salespersonid = sp.salespersonid INNER JOIN product p INNER JOIN productline pl ON p.productlineid = pl.productlineid ON ol.productid = p.productid INNER JOIN territory t ON cs.territoryid = t.territoryid

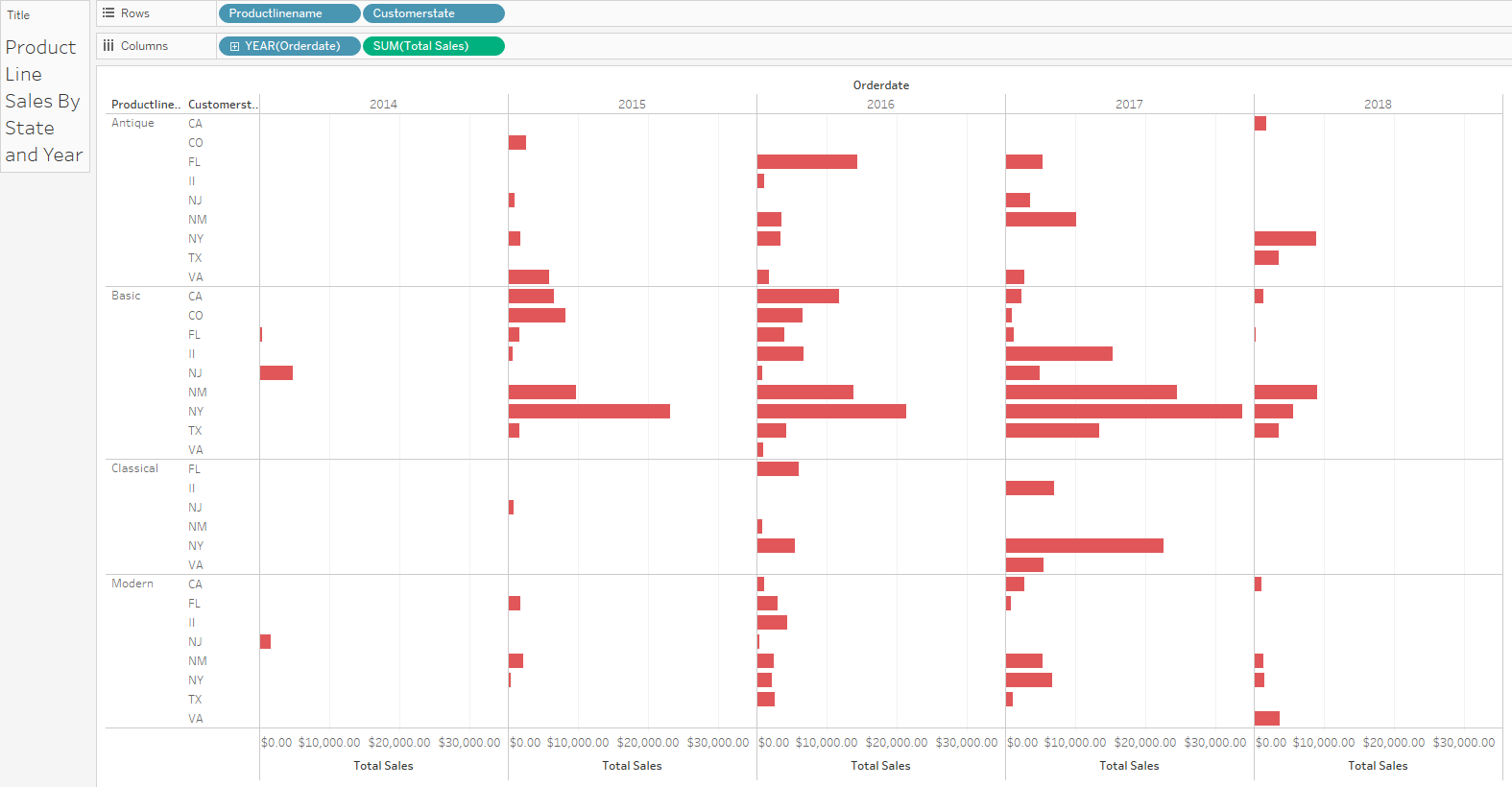
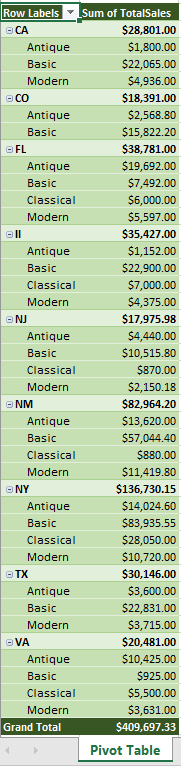
GROUP BY

o.orderid, o.customerid, o.orderdate, o.fulfillmentdate, o.shipadrsid, ol.orderlineid, ol.productid, ol.orderedquantity, ol.extendedprice, p.productlineid, p.productdescription, p.productfinish, p.productstandardprice, p.productonhand, p.productcost,pl.productlinename, cs.shipaddressid, cs.territoryid, cs.shipaddress, cs.shipcity, cs.shipstate, cs.shipzip, cs.shipdirections, sp.salespersonid, sp.salespersonname, sp.salespersontelephone, sp.salespersonfax, sp.salespersonaddress, sp.salespersoncity, sp.salespersonstate,sp.salespersonzip, sp.salesterritoryid, t.territoryname, c.customername, c.customeraddress, c.customercity, c.customerstate, c.customerpostalcode

ORDER BY o.orderid;

****

**8) 9)**

****

The entirety of this assignment was carried out on my own device (laptop computer) on which I was able to install the Oracle 12c database server and SQL Developer client software. I cannot even count the number of hours I spent on this assignment, it must have been somewhere between 24 and 48 in total. The most difficult part of the assignment was working with the unfamiliar functions such as LAG, I struggled with question 6 the most. I also had a hard time figuring out how to properly join tables.