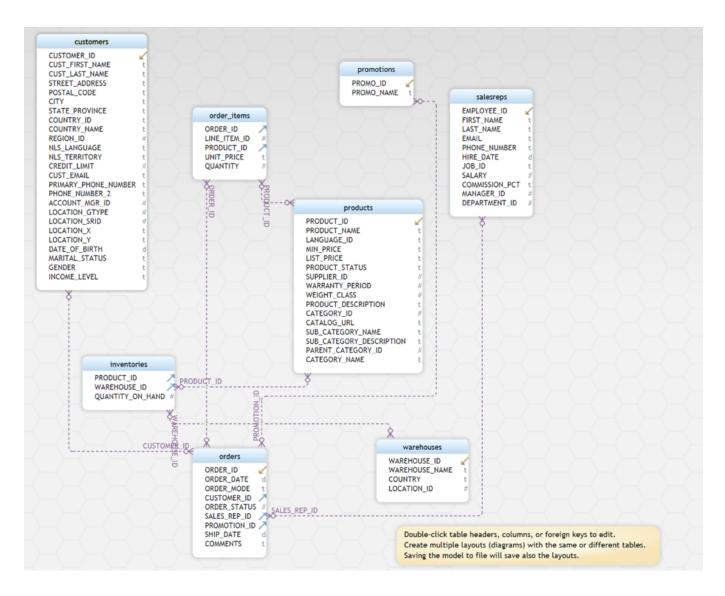
Lorena Vasquez

CIS 4400 Assignment 2

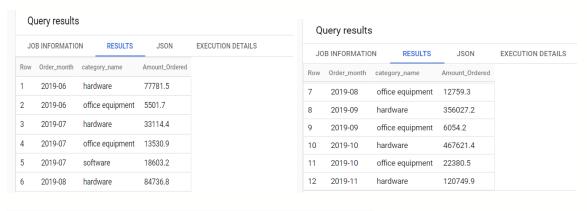


Question 3:

Write a SQL statement to show the total dollar amount sold to customers summarized by product category name and each month of each Year (YYYY-MM). Only include non-canceled orders that were shipped within 7 days of ordering

```
SELECT FORMAT_DATE("%Y-%m", order_date) AS Order_month,
category_name,
ROUND( SUM(unit_price * quantity), 2) AS Amount_Ordered
FROM `order_entry_dataset.customers`
INNER JOIN
`order_entry_dataset.orders` USING (customer_id)
INNER JOIN
`order_entry_dataset.order_items` USING (order_id)
inner join
`order_entry_dataset.products` USING (product_id)
Where ORDER_STATUS >= 4 AND DATE_DIFF(ship_date, order_date,
DAY) < 7
GROUP BY Order_month, CATEGORY_NAME
ORDER BY Order_month, CATEGORY_NAME;</pre>
```

Results:



Results per page: $50 \vee 1 - 50 \text{ of } 85$

Total of 85 results.

Question 4:

Write a SQL statement to show the total dollar amount sold summarized by Customer marital status and Year along with RANK. The largest sales by marital status by year should be ranked #1.

Results:

Row	Order_Year	marital_status	Amount_Sold	Ranking
1	2021	married	2998764.0	1
2	2021	single	2422835.5	2
3	2020	married	1878543.1	3
4	2019	married	1611426.8	4
5	2020	single	1120060.1	5
6	2019	single	522361.1	6
7	2022	married	65571.8	7

Total of 7 results.

Question 5:

Write a SQL statement to show the total dollar amount sold across product categories for all orderable products. Calculate the percentage contribution of each product category's sales to the overall total sales.

```
SELECT category name, Amount Ordered, SUM(Amount Ordered) OVER
() AS TOTAL, 100*Amount Ordered/SUM(Amount Ordered) OVER () AS
Percentage
FROM
( SELECT category name, ROUND( SUM(unit price * quantity), 2) AS
Amount Ordered
FROM
`order entry dataset.orders`
INNER JOIN
`order entry dataset.order items` USING (order id)
inner join
`order entry dataset.products` USING (product id)
WHERE SHIP DATE IS NOT null
GROUP BY CATEGORY NAME
ORDER BY CATEGORY NAME)
```

Result:

Row	category_name	Amount_Ordered	TOTAL	Percentage
1	hardware	6281185.8	10194690.0	61.612327594071033
2	office equipment	3179422.0	10194690.0	31.187039527440266
3	software	734082.2	10194690.0	7.2006328784887037

Total of 3 results.

Question 6:

Write a SQL statement to show the most profitable product over all orders. (unit price above Min Price). Only consider products that are available in the US or Canadian warehouses with list price over \$50.

```
SELECT (unit price - min price) AS profit , country , list price
, PRODUCT ID , PRODUCT NAME
FROM `order entry dataset.order items`
inner join
`order entry dataset.products` USING (product id)
inner join
`order entry dataset.inventories` USING (product id)
inner join
`order entry dataset.warehouses` USING (warehouse id)
WHERE country = ('US')
OR country = ('CA')
AND list price>50
GROUP BY country , profit , list price , PRODUCT ID ,
PRODUCT NAME
ORDER BY profit DESC
```


Result:

Row	profit	country	list_price	PRODUCT_ID	PRODUCT_NAME
1	193.6999999999982	US	3379.95	3003	Laptop 128/12/56/v90/110
2	193.6999999999982	CA	3379.95	3003	Laptop 128/12/56/v90/110
3	174.55000000000018	US	2625.0	2350	Desk - W/48
4	174.55000000000018	CA	2625.0	2350	Desk - W/48
5	173.69999999999982	US	3379.95	3003	Laptop 128/12/56/v90/110
6	173.69999999999982	CA	3379.95	3003	Laptop 128/12/56/v90/110
7	163.69999999999982	US	3379.95	3003	Laptop 128/12/56/v90/110
8	163.6999999999982	CA	3379.95	3003	Laptop 128/12/56/v90/110

Results per page: 50 ▼ 1 - 50 of 886

Question 7:

```
Which month (be sure to say from which year) had the largest
percentage increase in sales over the prior month? Justify your
rationale and show your SQL query (Hint: use the LAG function).
SELECT Order Month, MonthlySales, PriorMonthSales,
MonthlySales-PriorMonthSales AS MonthlySalesDifference ,
100*ABS((MonthlySales-PriorMonthSales))/ (PriorMonthSales) AS
PercentageChangeInSales
FROM (SELECT Order Month, MonthlySales,
LAG (MonthlySales, 1, 0)
OVER (ORDER BY Order Month) AS PriorMonthSales
(SELECT FORMAT DATE ("%Y-%m", DATE (order date)) AS Order Month
   , SUM(unit price * quantity) AS MonthlySales
FROM `order entry dataset.customers`
      INNER JOIN
      `order entry dataset.orders` USING (customer id)
      INNER JOIN
      `order entry dataset.order items` USING (order id)
GROUP BY Order Month))
WHERE PriorMonthSales <> 0
ORDER BY PercentageChangeInSales DESC ;
```

Results:

Row	Order_Month	MonthlySales	PriorMonthSales	MonthlySalesDifference	PercentageChangeInSales
1	2019-09	434015.5	97496.1	336519.4	345.16190904046414
2	2020-12	759097.49999999977	198009.70000000007	561087.7999999997	283.36379480399165
3	2020-05	338531.0999999998	130551.4	207979.69999999981	159.30867076109473
4	2019-07	390539.40000000008	152899.99999999997	237639.40000000011	155.4214519293657
5	2020-10	417839.49999999994	165608.7	252230.79999999993	152.30528347846453
6	2019-12	406763.10000000003	162071.9	244691.200000000004	150.9769429493947
7	2021-06	1038730.5999999999	426148.50000000012	612582.09999999974	143.74850550922966
8	2021-03	844924.00000000012	470307.500000000006	374616.500000000006	79.653524555742791
9	2021-09	255859.90000000002	142984.9	112875.00000000003	78.941902256811758
10	2019-08	97496.1	390539.40000000008	-293043.30000000005	75.035527785416775
11	2021-11	481332.89999999991	275400.6	205932.29999999993	74.775545151317729
12	2022-01	65571.8	245746.600000000003	-180174.80000000005	73.31731140939489

Results per page: 50 - 1 - 31 of 31

Total of 31 results.

Question 8:

Who is the "best" Sales Manager? Justify your rationale and back it up with queries and data. You may also wish to graph various data to support your justification. DO NOT just total up sales. Consider multiple factors and build a weighted model with SQL. Look at other tables beyond just orders.

a. BASED OFF THE AMOUNT SOLD

i. For this query, I wanted to see how much of the sum total did each manager contribute with. I did the sum of the "total sales", in this case it is (unit price x quantity). I then grouped it by the manager identification number only. Ranking made it easier to have an easy visualization that manager 147 had the highest sum out of all the managers.

Row	MANAGER_ID	Amount_Sold	Ranking
1	147	2512812.1	1
2	148	2412208.9	2
3	149	2149284.6	3
4	146	1649670.6	4
5	145	1647657.6	5

Total of 5 results.

b. Based on the commission percentage.

i. To begin with, I believed that having a higher commission percentage, would be a correlation with the manager who had the most amount sold, but in this case, that is not true. Manager 146 had the highest commission percentage, but from our previous query I know that manager 147 had a much higher amount sold. It could be that the commission percentage is based off of the amount sold, and he probably passed the amount sold to have a higher threshold of the commission percentage.

GROUP BY manager_ID , COMMISSION_PCT ORDER BY COMMISSION PCT DESC

Row	COMMISSION_PCT	MANAGER_ID	Amount_Sold				
1	0.35	146	364132.4				
2	0.3	145	236958.8				
3	0.3	146	533377.2	8	0.25	148	579827.4
4	0.3	149	302538.9	9	0.25	149	474381.7
5	0.25	145	430252.1	10	0.2	145	337528.0
6	0.25	146	752161.0	10	0.2	140	337320.0
7	0.25	147	268872.6	11	0.2	148	865761.0
Ω	0.25	1/18	570827 /	12	0.2	149	626344.2

Results per page: $50 \checkmark 1 - 19 \text{ of } 19$

Total of 19 results.

c. Based on the profit amount.

i. This idea came from one of the previous questions of the assignment. Here, I wanted to find the sum of the total profit of the product, I then found that out of all the managers, manager 147 had brought the most profit through sales. This also reinforces the idea from the first query, where this manager is also the manager who has the highest of the amount sold. So this query also works to help us identify who is the "best sales manager".

Row	MANAGER_ID	profitable
1	147	5552.5500000000038
2	149	3896.2000000000025
3	148	3821.0500000000056
4	146	3376.7250000000022
5	145	3087.8

Total of 5 results.

d. Based on the number of customers assisted.

i. For this one, I wanted to see how many customers each manager attended to. I calculated the sum of each manager, and found it surprising, that manager 148 assisted more customers than manager 147 but the amount sold is still greater for manager 147. Although

148 attended more customers, manager 147 was able to sell more in the amount of goods sold. It can also be predicted that if manager 147 had assisted more customers, they probably would of been able to product more amount sold and profit.

```
SELECT SUM (CUSTOMER ID) AS NUM CUSTOMERS, MANAGER ID,
     ROUND ( SUM (unit price * quantity), 2) AS Amount Sold,
     RANK() OVER (ORDER BY SUM(unit price * quantity) DESC) AS
Ranking
FROM `uplifted-valor-343403.order entry dataset.salesreps` AS SR
Inner Join `uplifted-valor-343403.order entry dataset.orders` AS
ORT
on SR.EMPLOYEE ID=ORT.SALES REP ID
INNER JOIN
`uplifted-valor-343403.order entry dataset.order items` USING
(order id)
INNER JOIN `uplifted-valor-343403.order entry dataset.customers`
USING (customer id)
GROUP BY manager ID
ORDER BY num customers DESC
```

JO	B INFORMATION	RESULTS	JSON	EXE
Row	NUM_CUSTOMERS	MANAGER_ID	Amount_Sold	Ranking
1	115653	148	2412208.9	2
2	113919	147	2512812.1	1
3	89997	149	2149284.6	3
4	73564	146	1649670.6	4
5	60439	145	1647657.6	5

Total of 5 results.

e. Based on the order comments.

- i. For this query, I initially wanted to form a WHERE clause
- ii. like this:
 - 1. Where COMMENTS = 'Canceled' (no success)
 - 2. WHERE comments LIKE 'Canceled' (no success)

iii.

The results below showed me the various different comments that the managers have received. Manager 147 had the most comments on customers canceling orders.

It could be a possible scenario where the manager is:

 Canceling orders to maintain the loyalty and customer satisfaction

```
SELECT manager_ID,COUNT(comments) AS Comm , comments
FROM `uplifted-valor-343403.order_entry_dataset.salesreps` AS SR
Inner Join `uplifted-valor-343403.order_entry_dataset.orders` AS
ORT
on SR.EMPLOYEE_ID=ORT.SALES_REP_ID
WHERE comments is not null
GROUP BY manager_ID , comments
ORDER BY comments DESC , manager_ID
```

Results

	mments
1 145 1 We	
	e must be cautions with this customer.
2 147 1 We	e must be cautions with this customer.
3 148 1 We	e must be cautions with this customer.
4 145 1 Wa	aiting on parts
5 147 1 Wa	aiting on parts
6 148 1 Wa	aiting on parts
7 149 1 Wa	aiting on part availability
8 149 1 Thi	nis order was on hold because customers's credit limit had been exceeded. Order will ship when payment is received
8 149 1 This or	order was on hold because customers's credit limit had been exceeded. Order will ship when payment is received
9 145 1 This or	order was disputed, but resolved; Customer doesn't like the colors and precision of the parts
10 149 1 This or wrong.	order was disputed and resolved. Customer claimed that container with shipment was damaged. FedEx's investigation proved this
11 148 1 This cu	customer found a better offer from one of our competitors. Will call back to renegotiate.
12 147 1 They w	want to reevaluate their terms agreement with the VP of Sales

45	146	1	Customer cancelled due to urgent budgeting issues. Must be cautious when dealing with them in the future. Since order shipped already we must discuss who would cover the shipping charges.
46	147	1	Customer canceled. Difficult to work with
47	145	2	Customer canceled.
48	146	2	Customer canceled.
49	147	4	Customer canceled.
50	148	2	Customer canceled.
51	149	2	Customer canceled.
52	148	1	Customer called to cancel. The warehouse was notified in time and the order didn't ship. They have a new VP of Sales and are shifting their sales model. Our VP of Sales should contact them.

Results per page: $50 \checkmark 1 - 50 \text{ of } 64$

Total of 64 results.

Final Answer for Question 8:

Ultimately through the different number of queries that I have constructed, I believe that the "best sales manager" would be manager 147. In numerous ways, they have shown through the number of factors like:

- Amount sold "Total Sales" by multiplying the unit price by the quantity
- Based on the profit that each manager brought the the company
- Based on the commission percentage
- Based on the number of customers assisted
- Based on the comments of the order

Question 9:

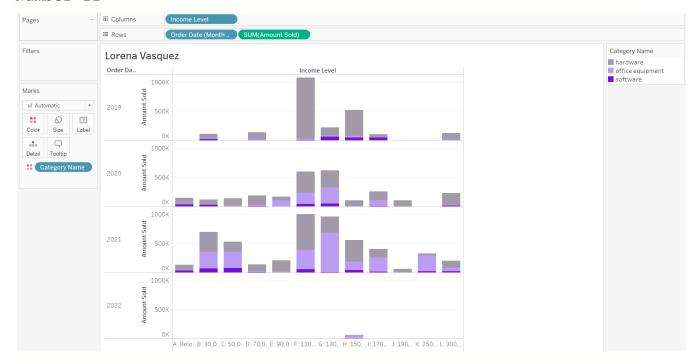
```
SELECT *
FROM `order_entry_dataset.salesreps` AS SR
Inner Join `order_entry_dataset.orders` AS ORT
on SR.EMPLOYEE_ID=ORT.SALES_REP_ID
INNER JOIN `order_entry_dataset.order_items` USING (order_id)
INNER JOIN `order_entry_dataset.promotions` AS p
on p.PROMO ID=ORT.PROMOTION ID
```

https://drive.google.com/file/d/1shwZ_BcLnGH14-E0hG1Z905XrPEdXMo
T/view?usp=sharing

Number 10.

Row Labels	Sum of Amount_Sold
□ China	9982969.2
hardware	4440473.7
office equipment	490842.8
software	18036.3
■ Germany	24870165.6
hardware	11813304.8
office equipment	1310738.1
software	97991
■ India	24397250.5
hardware	17153973
office equipment	424074
software	1659
■ Italy	86170076
hardware	42715556.4
office equipment	2987759.5
software	695550.1
Switzerland	81027471
hardware	44116805.6
office equipment	3926980.2
software	108763.5
∃Thailand	10430256
hardware	7728384
office equipment	29798.5
software	62587.8
■ United States of America	13078826797
hardware	5533609486
office equipment	802904461.2
software	121573919.2
Grand Total	22500638172

Number 11



At the end of the assignment answers, please tell me:

- 1. How many hours did you spend working on the assignment? About 10-15 hours
- 2. What was the most difficult part of completing the assignment?

The most difficult would be the coding, and having to use the various different platforms. It does give us a real life experience, as what we would be doing in school.

Resource:

https://www.mssqltips.com/sqlservertip/6150/get-column-attribute
s-for-all-sql-server-tables-that-match-search-criteria/
example, first page)

https://www.zentut.com/sql-tutorial/sql-aggregate-functions/
(First example, first page)

https://cloud.google.com/bigquery/docs/reference/standard-sql/da
te_functions#date_diff (first page)

https://www.mssqltips.com/sqlservertip/2508/sql-server-datediff-example/#:~:text=Instead%20of%20adding%20or%20subtracting,ORDER%20BY%20and%20HAVING%20clauses. (Third example, first page)

https://cloud.google.com/bigquery/docs/reference/standard-sql/nu
mbering functions#rank

http://holowczak.com/exploring-analytical-sql-with-google-bigque
ry/7/ (Page 7)

http://holowczak.com/getting-started-with-google-bigguery-on-goo
gle-cloud-platform/6/(Page 6)

http://holowczak.com/reverse-engineering-a-google-bigquery-schem a-with-dbschema/5/(Page 5)

http://holowczak.com/creating-a-service-account-and-key-file-for
-google-bigquery/3/(Page 3)