o How many hours did you spend working on the assignment?

I spent at least 4 hours a day for 3 days to complete this assignment, so 12+ hours total. I really love that it is structured in a way that I learned to approach a data set with dbSchema to see the data layout, manipulate and analyze on BigQuery, connect the dataset to Excel and Tableau using ODBC to create real-time reports and visualization.

o What was the most difficult part of completing the assignment?

The most difficult part is trying to refresh myself with windows function, join syntax. Besides, there are little things that I need to be super careful and filter data in order to not slice and dice irrelevant data.

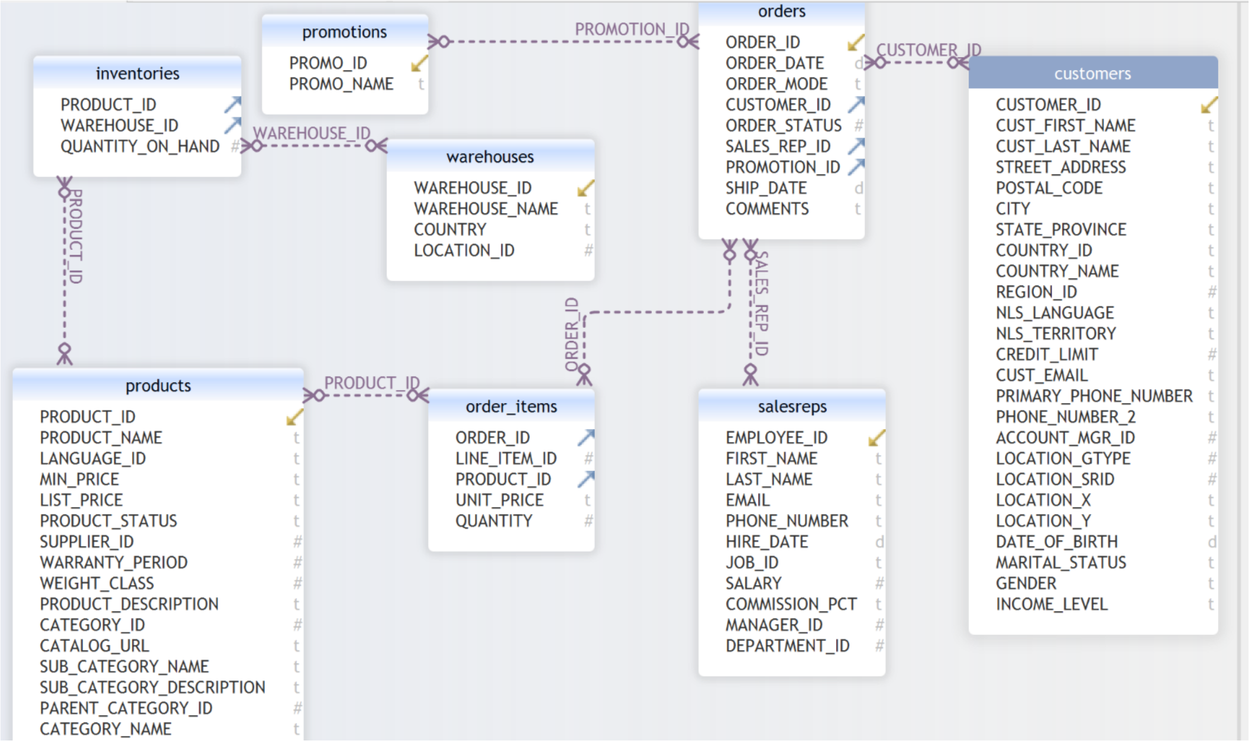
o Tell me one piece of advice to give to future CIS 9440 students about this assignment

- Little things matter. For example, after writing a long SQL script and run the result, you also need to check if the result makes sense and if there are any NULL data, you have to handle NULL values. You definitely do not want to see NULL in a report or your teammates would prefer a clean query so that it is easier for the next analysis step.
- If you know there are sets of data that repeatedly needed for other queries, you should box them in CTEs instead of a chunk of subquery. It benefits you in recalling them for other queries and increase query performance. Use subquery for special sets that you don’t use repeatedly.

1. Create tables

```
CREATE TABLE `order_entry_dataset.customers` AS
SELECT * FROM `handy-bonbon-
142723.order_entry_dataset.customers`

CREATE TABLE `order_entry_dataset.products` AS
SELECT * FROM `handy-bonbon-
142723.order_entry_dataset.products`;

CREATE TABLE `order_entry_dataset.orders` AS
SELECT * FROM `handy-bonbon-142723.order_entry_dataset.orders`;

CREATE TABLE `order_entry_dataset.order_items` AS
SELECT * FROM `handy-bonbon-
142723.order_entry_dataset.order_items`;

CREATE TABLE `order_entry_dataset.salesreps` AS
SELECT * FROM `handy-bonbon-
142723.order_entry_dataset.salesreps`;

CREATE TABLE `order_entry_dataset.warehouses` AS
SELECT * FROM `handy-bonbon-
142723.order_entry_dataset.warehouses`;

CREATE TABLE `order_entry_dataset.inventories` AS
SELECT * FROM `handy-bonbon-
142723.order_entry_dataset.inventories`;

CREATE TABLE `order_entry_dataset.promotions` AS
SELECT * FROM `handy-bonbon-
142723.order_entry_dataset.promotions`;
```

2. Database layout – created by dbSchema, connected with the help of GG service account



8. The “best” Sales Manager

Let's find out who is the best sale manager in 2021. Based on my experience in retail industry, a good sale manager is the one with a team of sale reps that increases sale over time, is able to keep customer coming back and has low sale returns. For these criteria, I form 3 KPIs to assess sale manager performance.

The first one is customer retention rate in 2021. The higher the customer retention rate, the higher the weighted score. It is calculated by the number of distinct customers from 2020 that continued to come back in 2021 *over* the total number of distinct customers in 2021. Specifically, I took a pool of all customers in 2020 and count the total number of distinctive customers. Then, I count the total number of distinctive customers in 2021 that were also our customer from 2020.

The second one is revenue churn rate, calculated by the dollar amount of return orders over total sale in 2021 for each department of a manager. The revenue churn rate is negatively correlated to weighted scores, therefore my data is calculated with negative rate to calculate the weighted score.

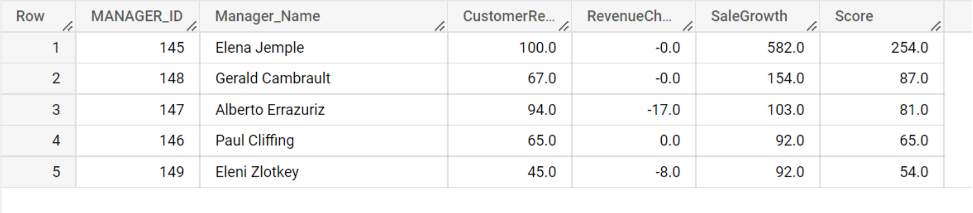
The third one is sale growth compared to 2020, calculated by the difference of sale between 2021 and 2020 over sale in 2020. The higher the sale growth rate, the higher the weighted score.

For me, I think customer loyalty is very important, therefore it is weighted **0.5**, sales growth is the next important, weighted **0.35**. The last one is return rate, weighted **0.15.**

Then, I formed a sum weighted criteria of 3 for 5 managers. Elena Jemple has the highest score.

```
/*1. CALCULATE KPI CustomerRetentionRate*/
/*1.1 COUNT CUSTOMER POOL IN 2020 */
WITH CustomerIn2020 AS (SELECT MANAGER_ID, COUNT(distinct CUSTOMER_ID) as Customer_Pool_In_2020
FROM `order_entry_dataset.salesreps` AS S RIGHT JOIN `order_entry_dataset.orders` AS O ON S.EMPLOYEE_ID=O.SALES_REP_ID
WHERE MANAGER_ID IS NOT NULL AND FORMAT_DATE("%Y", O.ORDER_DATE) = "2020"
GROUP BY MANAGER_ID),
/*1.2 COUNT NUMBER OF CUSTOMER IN 2020 THAT RETURN IN 2021*/
ReturnedCustomer AS
(SELECT MANAGER_ID, COUNT(DISTINCT O.CUSTOMER_ID) AS ReturnedCustomer2021
FROM `order_entry_dataset.salesreps` AS S RIGHT JOIN `order_entry_dataset.orders` AS O ON S.EMPLOYEE_ID=O.SALES_REP_ID
WHERE MANAGER_ID IS NOT NULL AND FORMAT_DATE("%Y", O.ORDER_DATE) = "2021" AND O.CUSTOMER_ID IN ( SELECT DISTINCT CUSTOMER_ID FROM `cis-9440-362403.order_entry_dataset.orders` WHERE ORDER_STATUS NOT IN (2, 3) AND  FORMAT_DATE("%Y", ORDER_DATE)="2020")
GROUP BY MANAGER_ID
ORDER BY MANAGER_ID),
/*1.3 CustomerRetentionRate*/
RetentionRate as
```

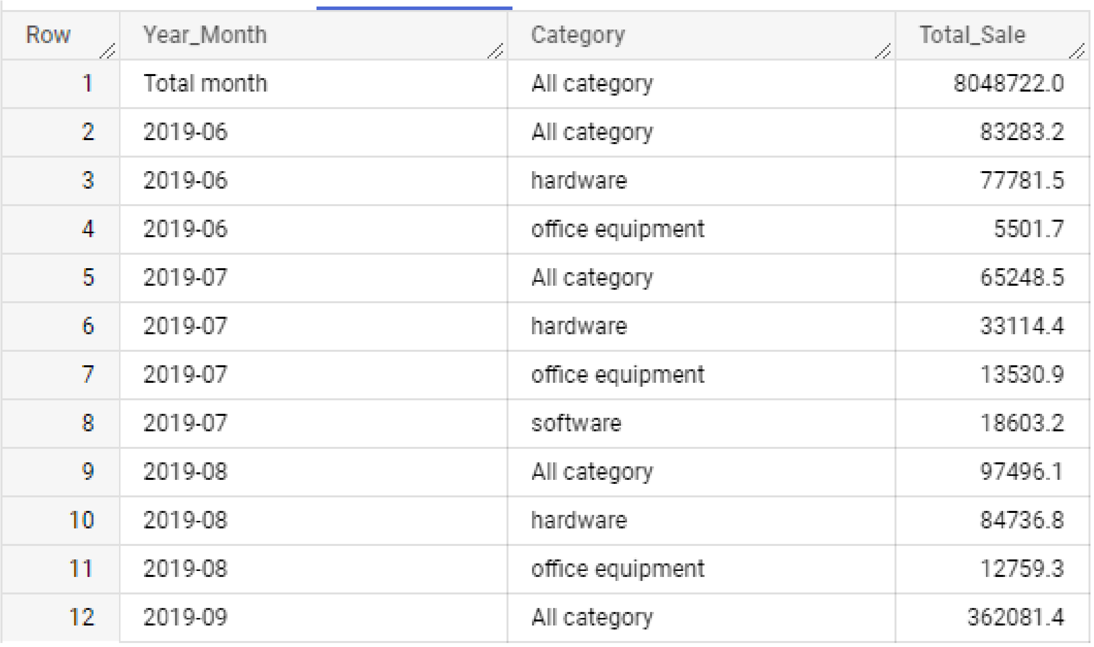
```
(SELECT MANAGER_ID,CustomerIn2020.Customer_Pool_In_2020, ReturnedCustomer.ReturnedCustomer2021, ROUND((100*ReturnedCustomer.ReturnedCustomer2021/CustomerIn2020.Customer_Pool_In_2020)) AS CustomerRetentionRate
FROM CustomerIn2020 JOIN ReturnedCustomer USING(MANAGER_ID)
ORDER BY MANAGER_ID),

/*2. CALCULATE KPI RevenueChurnRate*/
/*2.1 CALCULATE SALE BY ORDER*/
Orders AS
(SELECT O2.UNIT_PRICE*O2.QUANTITY AS Amount_Sold, ORDER_ID FROM
`cis-9440-362403.order_entry_dataset.orders` AS O1 LEFT JOIN `cis-9440-362403.order_entry_dataset.order_items` AS O2 USING(ORDER_ID)),
/*2.2 CALCULATE REVENUE CHURN*/
RevenueChurn AS
(SELECT MANAGER_ID, SUM(Orders.Amount_Sold) AS RevenueChurn
FROM `cis-9440-362403.order_entry_dataset.orders` AS O
JOIN `order_entry_dataset.salesreps` AS S ON O.SALES_REP_ID = S.EMPLOYEE_ID
JOIN Orders USING(ORDER_ID)
WHERE FORMAT_DATE("%Y", ORDER_DATE) = "2021" AND ORDER_STATUS=3
GROUP BY MANAGER_ID
ORDER BY MANAGER_ID),
TotalSaleIn2021 AS
(SELECT MANAGER_ID, SUM(Orders.Amount_Sold) AS Sale
FROM `cis-9440-362403.order_entry_dataset.orders` AS O
JOIN `order_entry_dataset.salesreps` AS S ON O.SALES_REP_ID = S.EMPLOYEE_ID
JOIN Orders USING(ORDER_ID)
WHERE FORMAT_DATE("%Y", ORDER_DATE) = "2021"
GROUP BY MANAGER_ID),
RevenueChurnRate AS
(SELECT MANAGER_ID, ROUND(-100*(RevenueChurn.RevenueChurn/TotalSaleIn2021.Sale)) AS RevenueChurnRate
FROM RevenueChurn JOIN TotalSaleIn2021 USING(MANAGER_ID)),

/*3. CALCULATE KPI YEARLY SALE GROWTH BY MANAGER: PercentageChangeInSale then rank descendant*/
/*3.1 FILTER YEARLY SALE BY MANAGER*/
Orders2 AS
(
  SELECT SUM(O2.UNIT_PRICE*O2.QUANTITY) AS YearlySale, FORMAT_DATE("%Y", ORDER_DATE) AS Year, S.MANAGER_ID AS MANAGER_ID
  FROM (`cis-9440-362403.order_entry_dataset.orders` AS O1
  LEFT JOIN `cis-9440-
362403.order_entry_dataset.order_items` AS O2
  USING(ORDER_ID)) LEFT JOIN `order_entry_dataset.salesreps` AS
S ON O1.SALES_REP_ID=S.EMPLOYEE_ID
  WHERE
(O1.ORDER_STATUS NOT IN (2,3)) and (FORMAT_DATE("%Y", ORDER_DATE
)) IN('2020','2021')
  GROUP BY MANAGER_ID,Year
  ORDER BY MANAGER_ID, Year),
/*3.2 USE LAG TO CALCULATE PERCENTAGE CHANGE IN SALE*/
YearlySaleGrowth AS
(SELECT manager_id,Year, ROUND(100*(YearlySale-
PriorYearlySales)/PriorYearlySales) as SaleGrowth
FROM
(
SELECT manager_id,Year, LAG(YearlySale, 1) OVER ( PARTITION BY M
ANAGER_ID ORDER BY YearlySale) as PriorYearlySales, YearlySale a
s YearlySale
FROM Orders2
ORDER BY Manager_id,Year
)
where year = '2021'),

/*4 FILTER MANAGER NAME*/
ManagerName as (SELECT DISTINCT EMPLOYEE_ID AS MANAGER_ID, CONCA
T(S.FIRST_NAME, " ",S.LAST_NAME) AS Manager_Name
FROM `order_entry_dataset.salesreps` AS S
WHERE S.JOB_ID ="SA_MAN")
/*5. FORM A SUM WEIGHTED MODEL*/
SELECT ManagerName.MANAGER_ID,ManagerName.Manager_Name, Retentio
nRate.CustomerRetentionRate, COALESCE(RevenueChurnRate.RevenueCh
urnRate,0) AS RevenueChurnRate, YearlySaleGrowth.SaleGrowth, ROU
ND((0.5*RetentionRate.CustomerRetentionRate + 0.15*COALESCE(Reve
nueChurnRate.RevenueChurnRate,0) + 0.35*YearlySaleGrowth.SaleGro
wth)) as Score
FROM ManagerName
LEFT JOIN YearlySaleGrowth USING(MANAGER_ID)
LEFT JOIN RetentionRate USING(MANAGER_ID)
LEFT JOIN RevenueChurnRate USING(MANAGER_ID)
ORDER BY score desc
/* Elena Jemple is the best sale manager*/
```



Row	MANAGER_ID	Manager_Name	CustomerRe...	RevenueCh...	SaleGrowth	Score
1	145	Elena Jemple	100.0	-0.0	582.0	254.0
2	148	Gerald Cambrault	67.0	-0.0	154.0	87.0
3	147	Alberto Errazuriz	94.0	-17.0	103.0	81.0
4	146	Paul Cliffing	65.0	0.0	92.0	65.0
5	149	Eleni Zlotkey	45.0	-8.0	92.0	54.0

3. Sale by product category name and each month of each Year

```
  /*Create CTE Order from order and items table. Assume Profits
only counts for products shipped within 7 days and are not cance
lled  */
WITH
  Orders AS (
  SELECT
    O2.UNIT_PRICE*O2.QUANTITY AS Amount_Sold,
    ORDER_ID,
    PRODUCT_ID,
    ORDER_DATE,
    FORMAT_DATE("%Y-%m", ORDER_DATE) AS Month_Year
  FROM
    `cis-9440-362403.order_entry_dataset.orders` AS O1
  LEFT JOIN
    `cis-9440-362403.order_entry_dataset.order_items` AS O2
  USING
    (ORDER_ID)
  WHERE
    (DATE_DIFF(SHIP_DATE, ORDER_DATE, DAY)<=7)
    AND (O1.ORDER_STATUS NOT IN (2,
        3)) )
SELECT
  COALESCE(Month_Year,"Total month")  AS Year_Month,

  COALESCE(P.CATEGORY_NAME,"All category") AS Category,
  ROUND(SUM(Amount_Sold), 2) AS Total_Sale
FROM
  Orders
LEFT JOIN
  `cis-9440-362403.order_entry_dataset.products` AS P
USING
  (PRODUCT_ID)
GROUP BY
  ROLLUP(Month_Year,
    P.CATEGORY_NAME)
ORDER BY
  Month_Year
```

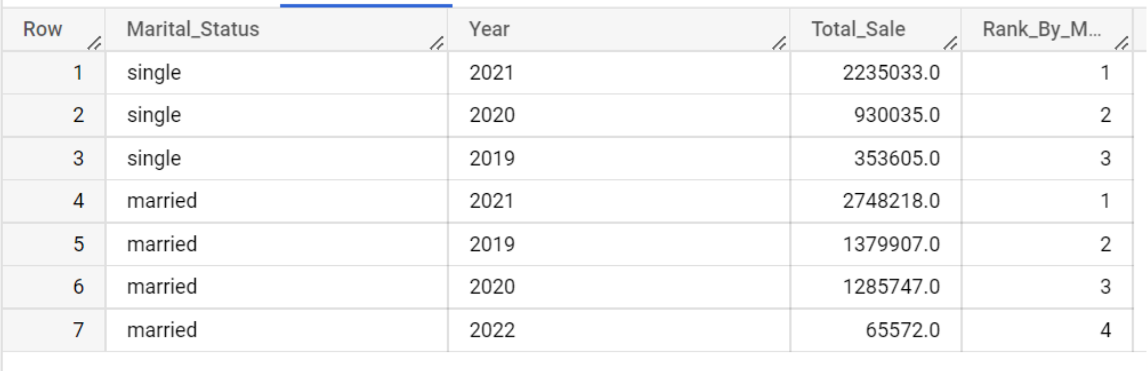


Row	Year_Month	Category	Total_Sale
1	Total month	All category	8048722.0
2	2019-06	All category	83283.2
3	2019-06	hardware	77781.5
4	2019-06	office equipment	5501.7
5	2019-07	All category	65248.5
6	2019-07	hardware	33114.4
7	2019-07	office equipment	13530.9
8	2019-07	software	18603.2
9	2019-08	All category	97496.1
10	2019-08	hardware	84736.8
11	2019-08	office equipment	12759.3
12	2019-09	All category	362081.4

Total Results 117

4. Sale by Customer marital status and Year and rank

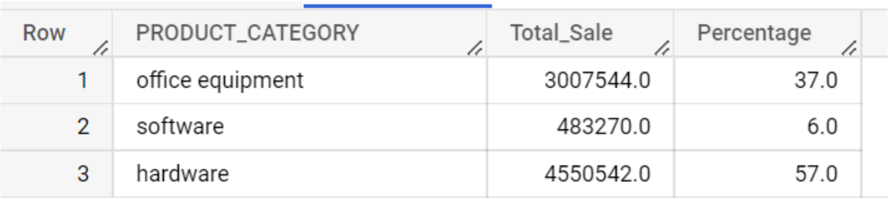
```
/*Create CTE Order from order and items table. Assume Profits onl
y counts for products not cancelled  */
WITH
  Orders AS (
  SELECT
    O2.UNIT_PRICE*O2.QUANTITY AS Amount_Sold,
    ORDER_ID,
    CUSTOMER_ID,
    PRODUCT_ID,
    ORDER_DATE
  FROM
    `cis-9440-362403.order_entry_dataset.orders` AS O1
  LEFT JOIN
    `cis-9440-362403.order_entry_dataset.order_items` AS O2
  USING
    (ORDER_ID)
  WHERE
    AND (O1.ORDER_STATUS NOT IN (2,
        3)) )
SELECT
  C.MARITAL_STATUS AS Marital_Status,
  FORMAT_DATE("%Y", ORDER_DATE) AS Year,
  ROUND(SUM(Orders.Amount_Sold)) AS Total_Sale,
  RANK() OVER (PARTITION BY C.MARITAL_STATUS ORDER BY SUM(Orders
.Amount_Sold) DESC) AS Rank_By_Marital_Status
FROM
  Orders
LEFT JOIN
  `cis-9440-362403.order_entry_dataset.customers` AS C
USING
  (CUSTOMER_ID)
GROUP BY
  Marital_Status,
  Year
```



Row	Marital_Status	Year	Total_Sale	Rank_By_M…
1	single	2021	2235033.0	1
2	single	2020	930035.0	2
3	single	2019	353605.0	3
4	married	2021	2748218.0	1
5	married	2019	1379907.0	2
6	married	2020	1285747.0	3
7	married	2022	65572.0	4

5. Sale by product categories for all orderable products. Percentage product category's sales to the overall total sales.

```
/*Create CTE Order from order and items table. Assume Profits only cou
nts for products not cancelled  */
WITH
  Orders AS (
  SELECT
    O2.UNIT_PRICE*O2.QUANTITY AS Amount_Sold,
    ORDER_ID,
    CUSTOMER_ID,
    PRODUCT_ID,
    ORDER_DATE
  FROM
    `cis-9440-362403.order_entry_dataset.orders` AS O1
  LEFT JOIN
    `cis-9440-362403.order_entry_dataset.order_items` AS O2
  USING
    (ORDER_ID)
  WHERE
    O1.ORDER_STATUS NOT IN (2,
        3))
Products AS (
  SELECT
    SUM(Amount_Sold) AS Total_Sale,
    P.CATEGORY_NAME AS PRODUCT_CATEGORY
  FROM
    Orders
  LEFT JOIN
    `cis-9440-362403.order_entry_dataset.products` AS P
  USING
    (PRODUCT_ID)
  WHERE
    P.PRODUCT_STATUS="orderable"
  GROUP BY
    P.CATEGORY_NAME)
SELECT
  PRODUCT_CATEGORY,
  ROUND(Total_Sale) AS Total_Sale,
  ROUND(100*Products.Total_Sale/SUM(Products.Total_Sale) OVER()) AS Pe
rcentage
FROM
  Products
```

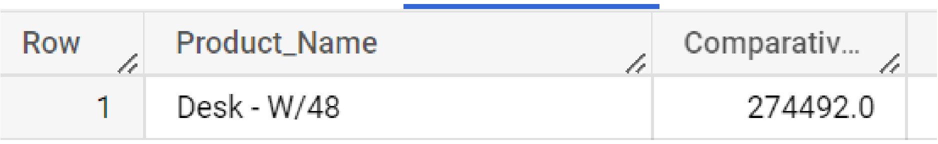


Row	PRODUCT_CATEGORY	Total_Sale	Percentage
1	office equipment	3007544.0	37.0
2	software	483270.0	6.0
3	hardware	4550542.0	57.0

6. The most profitable product overall orders. (unit price above Min Price). Only consider products that are available in the US or Canadian warehouses with list price over $50.

```
/*Create CTE Order from order and items table. Assume Profits on
ly counts for products not cancelled
Calculate current sales and projected minimum sales, calculated
at floor price  */
WITH
  Orders AS (
  SELECT
    O2.UNIT_PRICE*O2.QUANTITY AS Amount_Sold,
    O2.QUANTITY*P.MIN_PRICE AS Sale_At_Floor_Price,
    ORDER_ID,
    CUSTOMER_ID,
    PRODUCT_ID,
    ORDER_DATE
  FROM
    `cis-9440-362403.order_entry_dataset.orders` AS O1
  LEFT JOIN (`cis-9440-
362403.order_entry_dataset.order_items` AS O2
    LEFT JOIN
      `order_entry_dataset.products` AS P
    USING
      (PRODUCT_ID))
  USING
    (ORDER_ID)
  WHERE

   O1.ORDER_STATUS NOT IN (2,
        3) ),
  /*JOIN Products, Inventories, warehouse to FILTER country IN (
'US','CA')AND list_price>50*/
  Warehouse AS (
  SELECT
    PRODUCT_ID,
    PRODUCT_NAME
  FROM
    `order_entry_dataset.products`AS p
  LEFT JOIN (`order_entry_dataset.inventories` AS i
    LEFT JOIN
      `order_entry_dataset.warehouses` AS w
    USING
      (WAREHOUSE_ID))
  USING
    (PRODUCT_ID)
  WHERE
```

```
    w.COUNTRY IN ('US',
      'CA')
    AND p.LIST_PRICE>50 )
/* Assuming Minimum price is the price allowed to break even,pro
fits are the difference between actual sales and projected minim
um sale*/
SELECT
  Warehouse.PRODUCT_NAME AS Product_Name,
  ROUND(SUM(Orders.Amount_Sold-
Orders.Sale_At_Floor_Price)) AS Profit
FROM
  Orders
LEFT JOIN
  Warehouse
USING
  (PRODUCT_ID)
WHERE
  PRODUCT_NAME IS NOT NULL
GROUP BY
  Warehouse.PRODUCT_NAME
ORDER BY
  Profit DESC
LIMIT 1

/* The most profitable product are Desk - W/48.*/
```

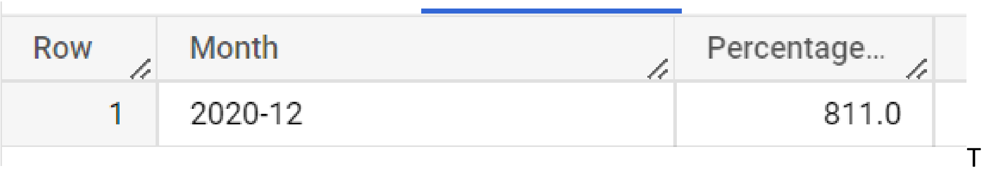


Row	Product_Name	Comparativ...
1	Desk - W/48	274492.0

7. The largest percentage increase in sales over the prior month - Provide rationale

December 2020 had the highest monthly sale growth where sales increased more than 8 times compared to November. I assumed it is due to holiday sale, end of year bonus and end of year promotions. December is when there are tons of promotions and two biggest holidays of the year, Christmas and New Year. Employees also get bonus, therefore there are a stronger buying power.

```
/*Create CTE Order from order and items table. Assume Profits on
ly counts for products cancelled  */
WITH Orders AS
(
  SELECT SUM(O2.UNIT_PRICE*O2.QUANTITY) AS MonthlySales, FORMAT_
DATE("%Y-%m", ORDER_DATE) AS Month_Year
  FROM `cis-9440-362403.order_entry_dataset.orders` AS O1
  LEFT JOIN `cis-9440-
362403.order_entry_dataset.order_items` AS O2
  USING(ORDER_ID)
  WHERE O1.ORDER_STATUS NOT IN (2,3)
  GROUP BY Month_year
  ORDER BY Month_year
)
SELECT Month, ROUND(100*(MonthlySales-
PriorMonthSales)/PriorMonthSales) as PercentageChangeInSale
FROM
(
SELECT Orders.Month_Year AS Month, LAG(Orders.MonthlySales, 1) O
VER (ORDER BY Orders.Month_Year) as PriorMonthSales, Orders.Mont
hlySales as MonthlySales
FROM Orders
ORDER BY Month
)
ORDER BY ROUND(100*(MonthlySales-
PriorMonthSales)/PriorMonthSales) DESC
LIMIT 1
```

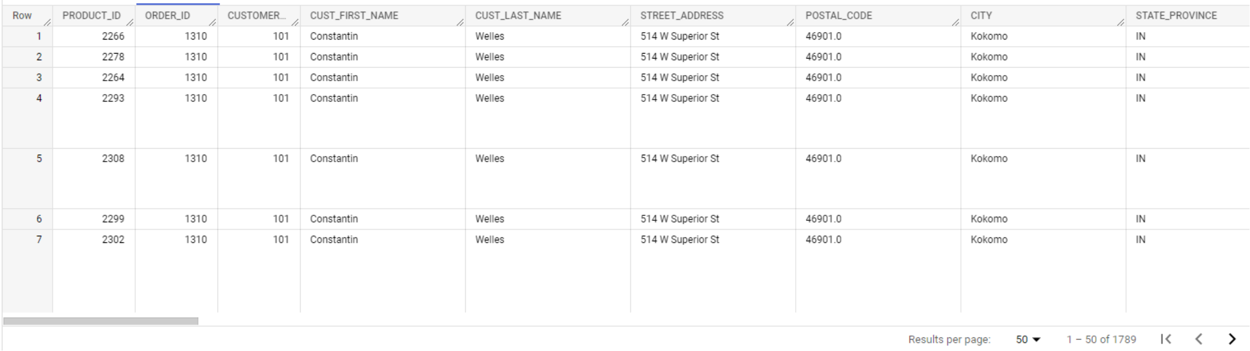


Row	Month	Percentage...
1	2020-12	811.0

Total results: 31

9. Join all of the columns in these tables: PRODUCTS, CUSTOMERS, ORDERS, ORDER_ITEMS, PROMOTIONS, and SALESREP

```
  SELECT
    *
  FROM `order_entry_dataset.customers` AS C
  JOIN (`order_entry_dataset.orders` AS O
    JOIN `order_entry_dataset.promotions` AS P ON O.PROMOTION_ID
 = P.PROMO_ID
    JOIN `order_entry_dataset.salesreps` AS S ON O.SALES_REP_ID=
 S.EMPLOYEE_ID) USING(CUSTOMER_ID)
  JOIN `order_entry_dataset.order_items` USING (ORDER_ID)
  JOIN `order_entry_dataset.products` USING (PRODUCT_ID)
```

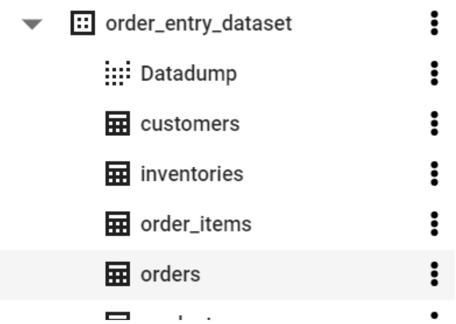


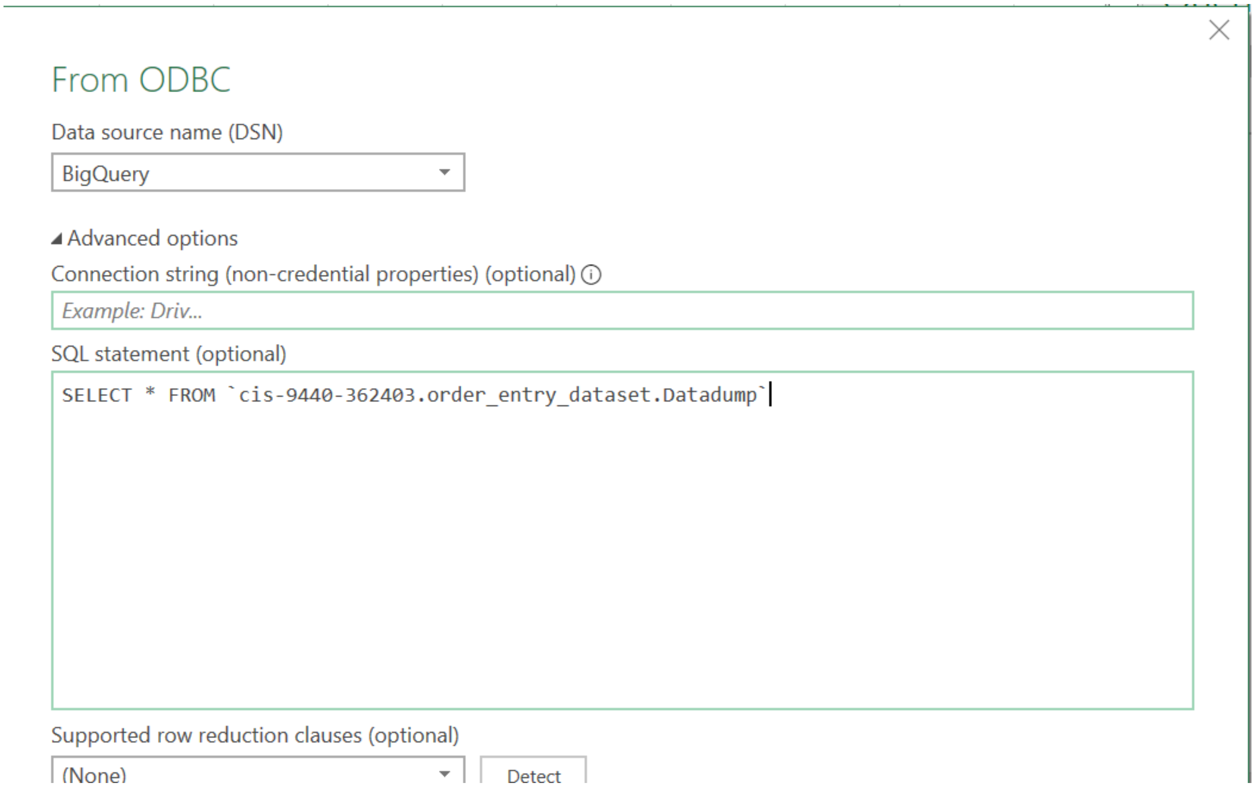
Row	PRODUCT_ID	ORDER_ID	CUSTOMER...	CUST_FIRST_NAME	CUST_LAST_NAME	STREET_ADDRESS	POSTAL_CODE	CITY	STATE_PROVINCE
1	2266	1310	101	Constantin	Welles	514 W Superior St	46901.0	Kokomo	IN
2	2278	1310	101	Constantin	Welles	514 W Superior St	46901.0	Kokomo	IN
3	2264	1310	101	Constantin	Welles	514 W Superior St	46901.0	Kokomo	IN
4	2293	1310	101	Constantin	Welles	514 W Superior St	46901.0	Kokomo	IN
5	2308	1310	101	Constantin	Welles	514 W Superior St	46901.0	Kokomo	IN
6	2299	1310	101	Constantin	Welles	514 W Superior St	46901.0	Kokomo	IN
7	2302	1310	101	Constantin	Welles	514 W Superior St	46901.0	Kokomo	IN

10. Pivot table that summarizes sales by product category and customer country.

Here I used Simba ODBC to connect BigQuery to Excel. BigQuery connector does not work for me.

I created view by running the question 9 query, save → save view and name the view as Datadump.





From ODBC

Data source name (DSN)

BigQuery

Advanced options

Connection string (non-credential properties) (optional)

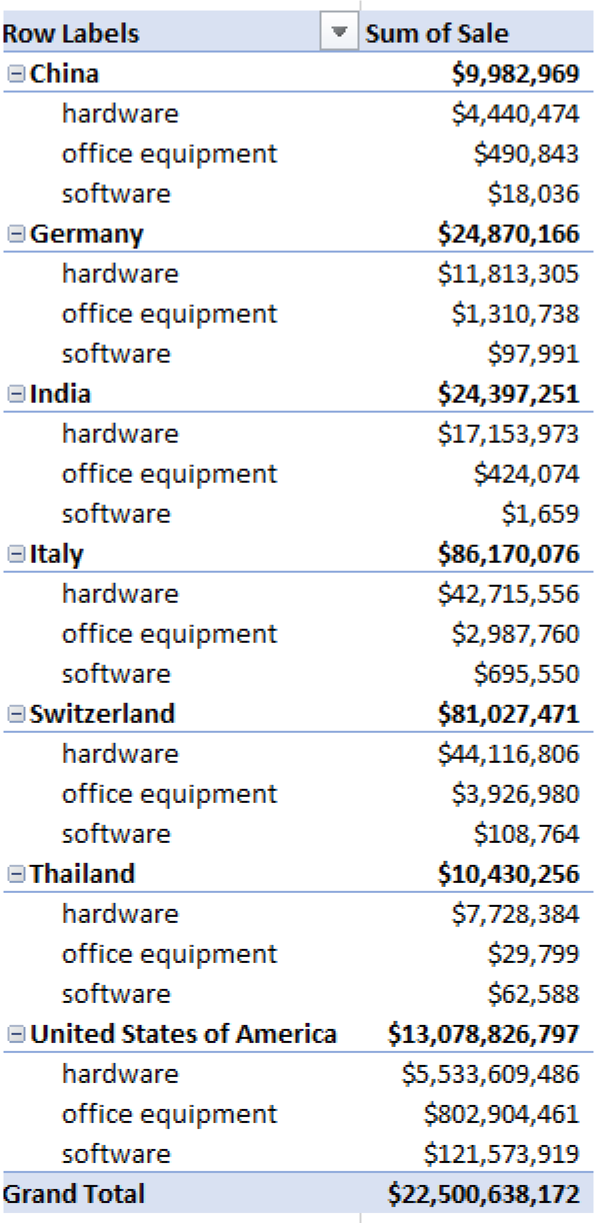
Example: Driv...

SQL statement (optional)

```
SELECT * FROM `cis-9440-362403.order_entry_dataset.Datadump`
```

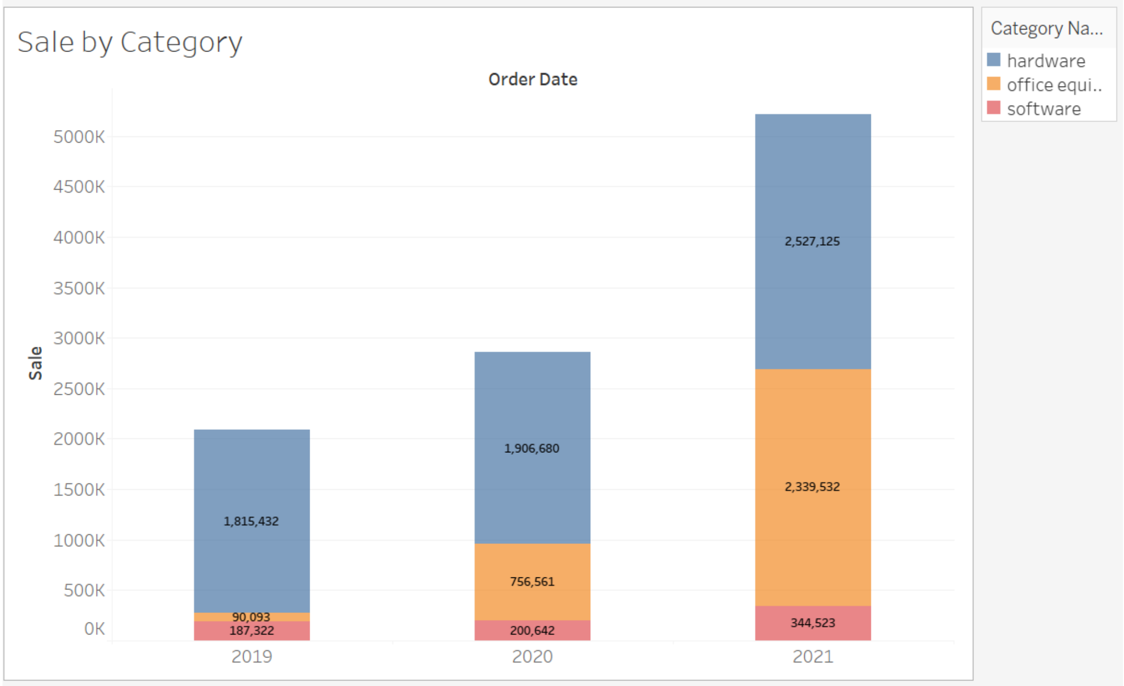
Supported row reduction clauses (optional)

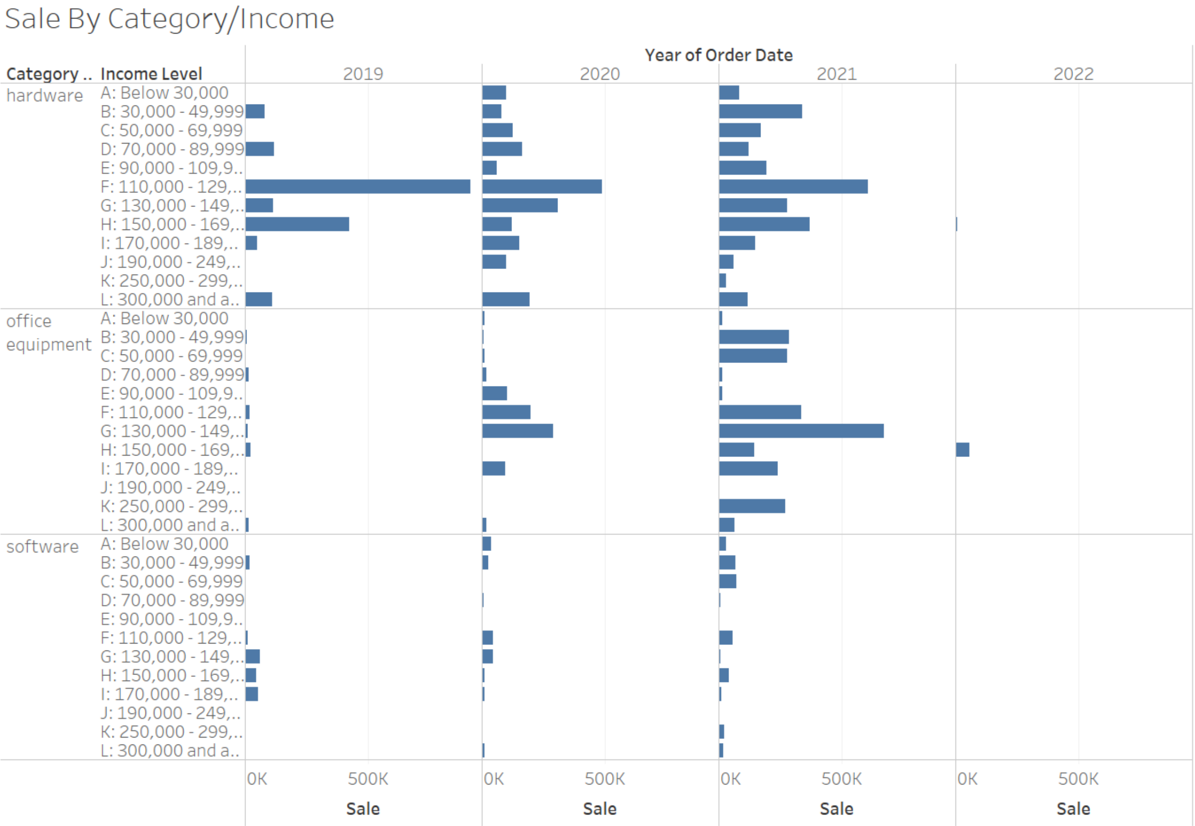
(None)



Row Labels	Sum of Sale
⊟China	**$9,982,969**
hardware	$4,440,474
office equipment	$490,843
software	$18,036
⊟Germany	**$24,870,166**
hardware	$11,813,305
office equipment	$1,310,738
software	$97,991
⊟India	**$24,397,251**
hardware	$17,153,973
office equipment	$424,074
software	$1,659
⊟Italy	**$86,170,076**
hardware	$42,715,556
office equipment	$2,987,760
software	$695,550
⊟Switzerland	**$81,027,471**
hardware	$44,116,806
office equipment	$3,926,980
software	$108,764
⊟Thailand	**$10,430,256**
hardware	$7,728,384
office equipment	$29,799
software	$62,588
⊟United States of America	**$13,078,826,797**
hardware	$5,533,609,486
office equipment	$802,904,461
software	$121,573,919
Grand Total	**$22,500,638,172**

11. Import all of the data from your VIEW into Tableau or Google Data Studio. Create an appropriate visualization from the resulting data set that summarizes the data according to total sales by product category and customer income level over time (e.g., months or years).





Sum of Sale for each Income Level broken down by Order Date Year vs. Category Name.

o