medium.com /@donhalogen/enhancing-business-insights-through-a-comprehensive-review-of-2019-sales-dynamics-cf719c5eed20

Unknown Title

don halogen ⋮⋮ 1/10/2024

Data Analysis of a Store Business’s 2019 Sales Performance I

don halogen

Executed by Alimi Ibrahim

Introduction

A data analysis challenge hosted by a Data Community requires participants to explore, clean, and analyse the provided dataset in order to generate data-driven recommendations.

Dataset Description

The dataset comprises twelve sets of uncleaned data, each representing a month in 2019, documenting the company’s product sales. It includes information such as the product’s name, order ID, the address from which the order was placed, the quantity of orders, and the cost per order.

Tools

I utilised Microsoft SQL (Azure Data Studio) to extract datasets from the provided database. Following that, I employed BigQuery for both dataset cleaning and analysis. Lastly, Tableau was utilised for visualising the analysed data.

Data Dictionary

a. Order ID — record of unique transactions

b. Product — product sold

c. Quantity — quantity of product sold

d. Price each — price sold for every transaction

e. Order date — date and time for each transaction

f. Purchase address — store location for every transaction

g. Total Cost / Revenue — Multiplication of Price each and Quantity

h. Month — Month for each transaction

Challenge Outline

i. Data Understanding; Obtain and explore the dataset

ii. Data Cleaning and Preparation for analysis

iii. Identify key features/columns relevant to sales analysis

iv. Calculate and present key sales metrics

v. Share your Insight and make recommendations

i. Data Understanding; Obtain and explore the dataset

The dataset was provided as a .bak file. I downloaded it onto my MacBook, traced the location of the downloaded file on terminal

```
ls /Users/mac/
```

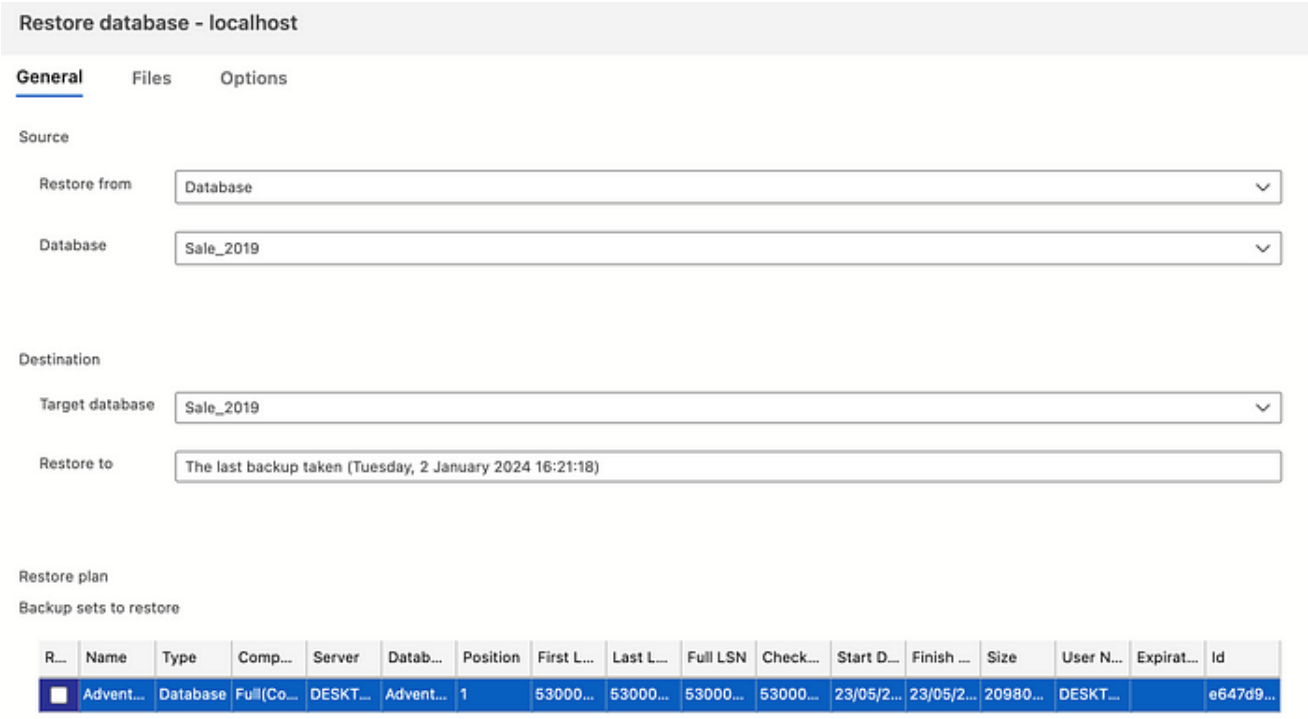
To verify the location of my .bak file

and then restored it in Microsoft SQL using Azure Data Studio by running the terminal functions

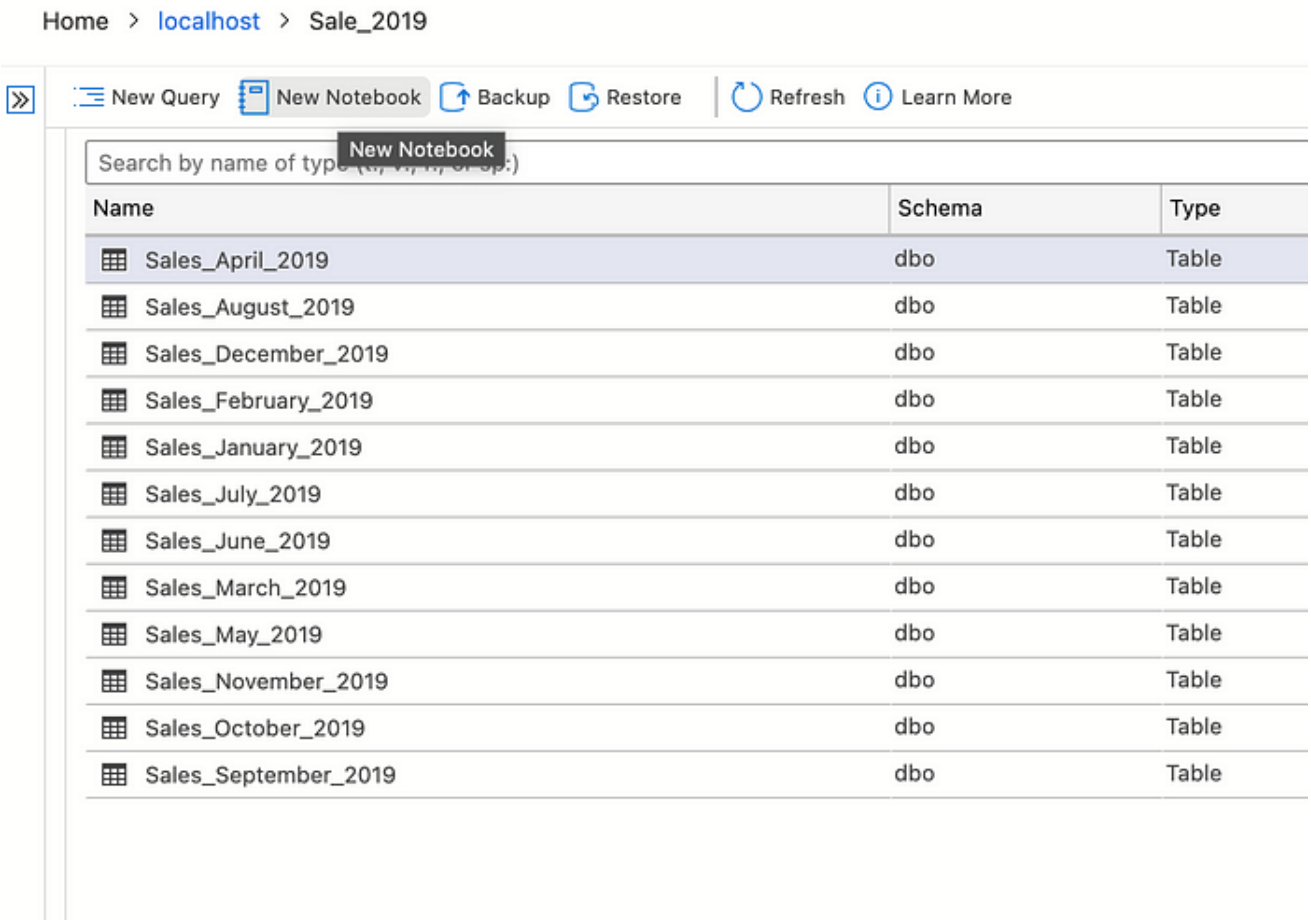
```
% sudo docker exec -it sql_server_test mkdir var/opt/mssql/backup
%  sudo docker cp Sale_2019.bak sql_server_test:/var/opt/mssql/backup
```

To Drop it in the Azure data studio MMSQL Folder

I used the restore function in the Azure Data Studio software to import and then



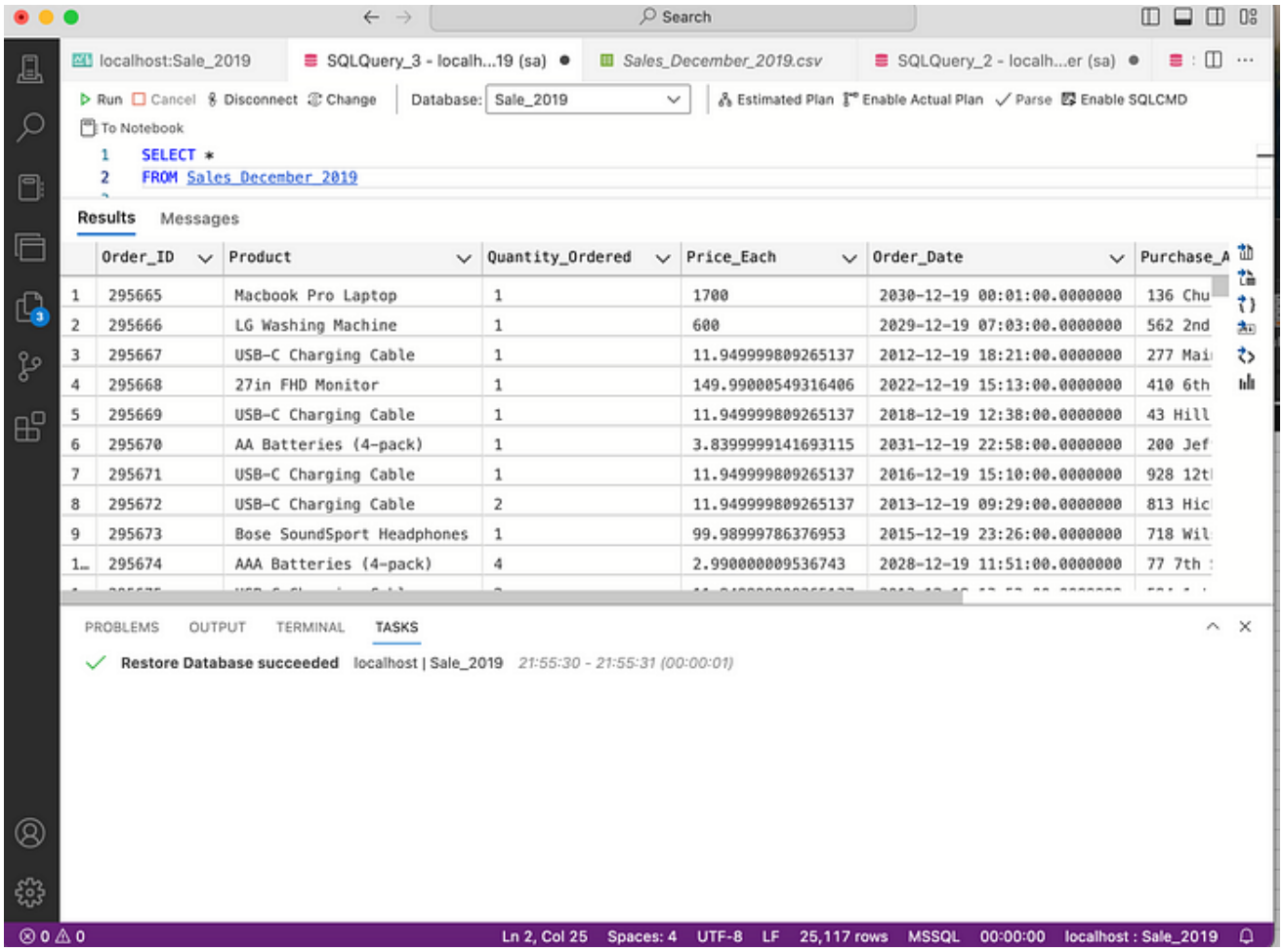
To restore the database



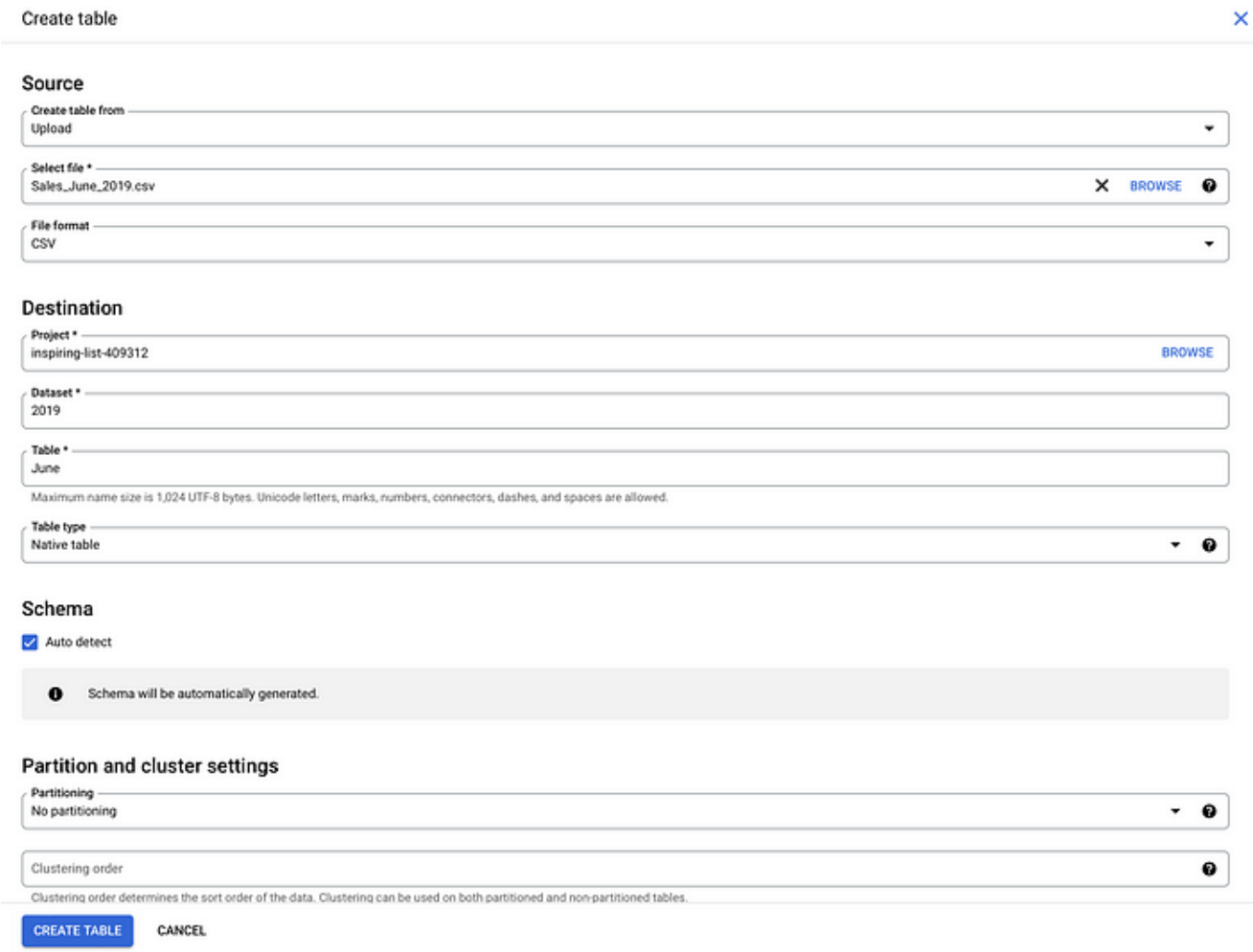
The twelve sales datasets provided for analysis

extracted them as .csv files which were later uploaded as tables to Big Query

SQL Code used to extract the tables

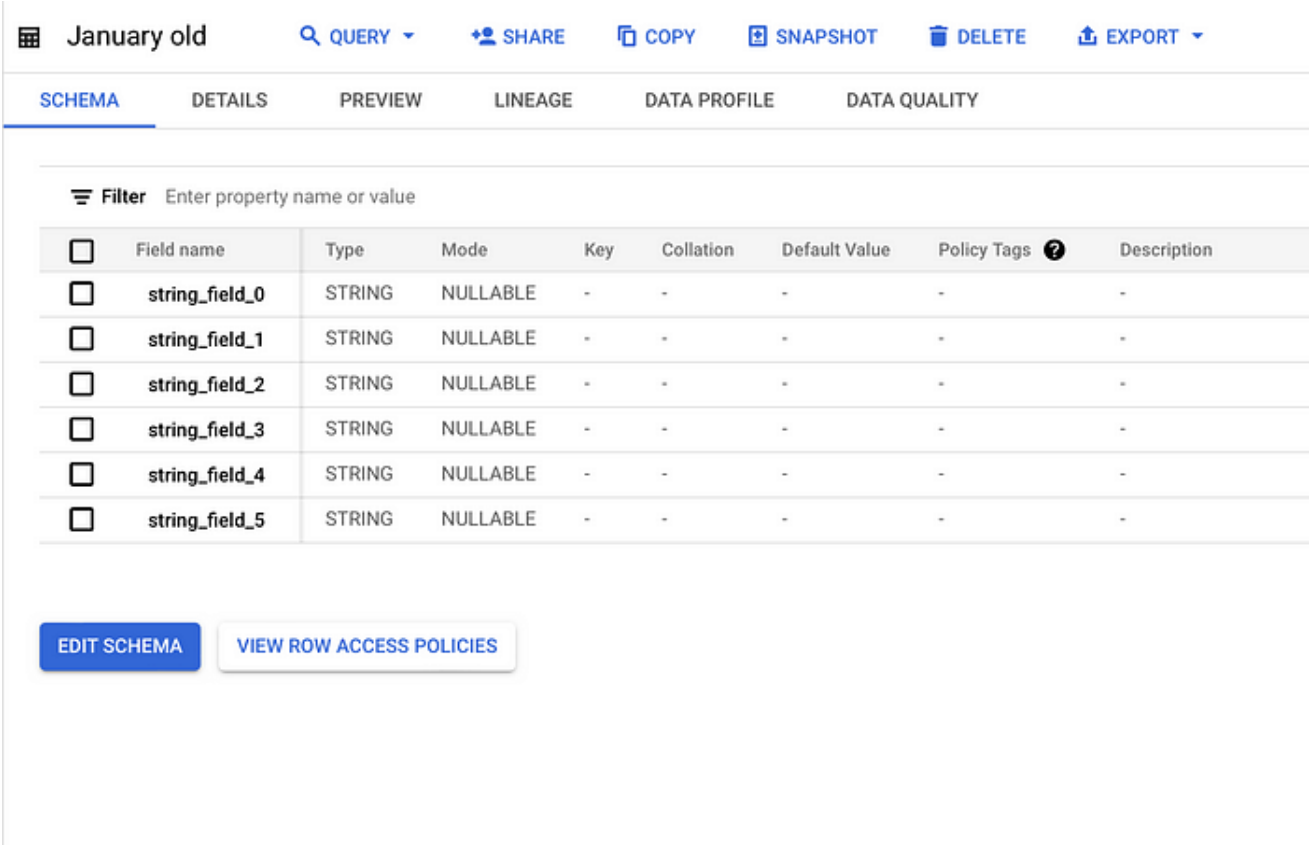


to extract the files as .csv



imported to BigQuery under a 2019 Dataset

The Schema of the datasets looked like this after import into BigQuery,



Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
string_field_0	STRING	NULLABLE	-	-	-	-	-
string_field_1	STRING	NULLABLE	-	-	-	-	-
string_field_2	STRING	NULLABLE	-	-	-	-	-
string_field_3	STRING	NULLABLE	-	-	-	-	-
string_field_4	STRING	NULLABLE	-	-	-	-	-
string_field_5	STRING	NULLABLE	-	-	-	-	-

the default schema of the raw data for month of january

and the previews showed that they were all structured the same way and exhibited similar issues

Row	string_field_0	string_field_1	string_field_2	string_field_3	string_field_4	string_field_5
1	141234	iPhone	1	700	2022-01-19 21:25:00.0000000	944 Walnut St, Boston, MA 022...
2	141336	iPhone	1	700	2009-01-19 18:23:00.0000000	811 Hickory St, Portland, OR 97...
3	141394	iPhone	1	700	2006-01-19 16:54:00.0000000	534 12th St, San Francisco, CA ...
4	141437	iPhone	1	700	2010-01-19 15:40:00.0000000	377 Meadow St, New York City,...
5	141457	iPhone	1	700	2009-01-19 22:11:00.0000000	820 Jackson St, Seattle, WA 98...
6	141458	iPhone	1	700	2029-01-19 01:12:00.0000000	497 Park St, San Francisco, CA ...
7	141460	iPhone	1	700	2026-01-19 17:29:00.0000000	855 2nd St, New York City, NY 1...
8	141472	iPhone	1	700	2027-01-19 07:05:00.0000000	853 9th St, San Francisco, CA 9...
9	141473	iPhone	1	700	2011-01-19 16:48:00.0000000	768 Maple St, San Francisco, C...
10	141476	iPhone	1	700	2006-01-19 15:38:00.0000000	295 Hickory St, New York City, ...
11	141504	iPhone	1	700	2013-01-19 13:37:00.0000000	855 11th St, Seattle, WA 98101
12	141517	iPhone	1	700	2013-01-19 14:14:00.0000000	326 Maple St, Los Angeles, CA ...
13	141520	iPhone	1	700	2015-01-19 11:24:00.0000000	23 Cherry St, Atlanta, GA 30301
14	141550	iPhone	1	700	2031-01-19 10:58:00.0000000	399 Church St, Boston, MA 022...
15	141582	iPhone	1	700	2015-01-19 01:45:00.0000000	416 9th St, San Francisco, CA 9...
16	141632	iPhone	1	700	2021-01-19 05:00:00.0000000	964 Hill St, Seattle, WA 98101
17	141663	iPhone	1	700	2001-01-19 11:01:00.0000000	738 8th St, San Francisco, CA 9...
18	141670	iPhone	1	700	2031-01-19 12:52:00.0000000	166 10th St, Atlanta, GA 30301
19	141702	iPhone	1	700	2027-01-19 18:33:00.0000000	776 10th St, Boston, MA 02215
20	141703	iPhone	1	700	2029-01-19 11:04:00.0000000	371 11th St, Portland, OR 97035
21	141732	iPhone	1	700	2001-01-19 06:13:00.0000000	446 Pine St, Atlanta, GA 30301
22	141738	iPhone	1	700	2014-01-19 20:53:00.0000000	183 Cherry St, Atlanta, GA 30301
23	141795	iPhone	1	700	2019-01-19 20:31:00.0000000	383 Jefferson St, San Francisc...
24	141886	iPhone	1	700	2010-01-19 06:37:00.0000000	94 7th St, New York City, NY 10...
25	141888	iPhone	1	700	2009-01-19 11:01:00.0000000	17 Spruce St, Los Angeles, CA ...
26	141961	iPhone	1	700	2012-01-19 17:39:00.0000000	512 Hickory St, Los Angeles, C...

The january.csv file preview

-
-
-
-
-
-

ii. Data Cleaning and Preparation for analysis

The process involved six straightforward steps to ensure comprehensive data cleaning. Initiation began by selecting a single table as a template, meticulously cleaning it, and leveraging it as a reference point for cleansing the remaining tables.

The first step done was to assign a column name to each column.

```sql
CREATE TABLE `inspiring-list-409312.2019`.Januaryy AS
SELECT
  string_field_0 AS Order_ID,
  string_field_1 AS Product,
  string_field_2 AS Quantity_Ordered,
  string_field_3 AS Price_each,
  string_field_4 AS Order_date,
  string_field_5 AS Purchase_address
FROM `inspiring-list-409312.2019.January old`
```

this created a new table with all the columns named

Januaryy QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW LINEAGE DATA PROFILE DATA QUALITY

Row	Order_ID	Product	Quantity_Ordered	Price_each	Order_date	Purchase_address
1	142421	iPhone	1	700	2006-01-19 14:23:00.0000000	242 5th St, San Francisco, CA 9...
2	143111	iPhone	1	700	2021-01-19 20:44:00.0000000	641 Washington St, Seattle, WA...
3	143998	iPhone	1	700	2024-01-19 11:12:00.0000000	232 Cherry St, Atlanta, GA 30301
4	144757	iPhone	1	700	2012-01-19 11:45:00.0000000	712 Cherry St, New York City, N...
5	146129	iPhone	1	700	2012-01-19 11:28:00.0000000	303 Walnut St, San Francisco, ...
6	148861	iPhone	1	700	2020-01-19 00:52:00.0000000	200 Cherry St, Los Angeles, CA ...
7	149500	iPhone	1	700	2031-01-19 19:27:00.0000000	691 4th St, San Francisco, CA 9...
8	141504	iPhone	1	700	2013-01-19 13:37:00.0000000	855 11th St, Seattle, WA 98101
9	141886	iPhone	1	700	2010-01-19 06:37:00.0000000	94 7th St, New York City, NY 10...
10	144114	iPhone	1	700	2002-01-19 19:19:00.0000000	426 14th St, San Francisco, CA ...
11	145704	iPhone	1	700	2021-01-19 15:25:00.0000000	671 Cherry St, New York City, N...
12	147289	iPhone	1	700	2012-01-19 18:47:00.0000000	44 Dogwood St, San Francisco,...
13	147577	iPhone	1	700	2017-01-19 20:02:00.0000000	378 Lake St, Seattle, WA 98101
14	148533	iPhone	1	700	2013-01-19 09:25:00.0000000	837 Maple St, Atlanta, GA 30301
15	149109	iPhone	1	700	2019-01-19 11:38:00.0000000	60 6th St, Boston, MA 02215
16	141738	iPhone	1	700	2014-01-19 20:53:00.0000000	183 Cherry St, Atlanta, GA 30301
17	143802	iPhone	1	700	2008-01-19 12:43:00.0000000	304 Lakeview St, San Francisc...
18	143966	iPhone	1	700	2018-01-19 22:26:00.0000000	653 Main St, Seattle, WA 98101

First 18 rows of the new table with named column

The next step was to convert the datatype in the columns

```sql
CREATE TABLE `inspiring-list-409312.2019`.January_3 AS
SELECT
  CASE
    WHEN Product = 'NULL' THEN NULL
    ELSE Product
  END AS Product_name,
  CASE
    WHEN Purchase_Address = 'NULL' THEN NULL
    ELSE SAFE_CAST(Purchase_Address AS STRING)
  END AS Purchase_Addres,
  CASE
    WHEN Order_ID = 'NULL' THEN NULL
    ELSE SAFE_CAST(Order_ID AS INT64)
  END AS Order_IDD,
  CASE
    WHEN Quantity_Ordered = 'NULL' THEN NULL
    ELSE SAFE_CAST(Quantity_Ordered AS INT64)
  END AS Quantity_Ordereds,
  CASE
    WHEN Price_Each = 'NULL' THEN NULL
    ELSE SAFE_CAST(Price_Each AS FLOAT64)
  END AS Price,
  Order_date AS date_ordered

FROM `inspiring-list-409312.2019.Januaryy`;
```

To convert some of data type from string to int64, float64

January_3 QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW LINEAGE DATA PROFILE DATA QUALITY

Filter Enter property name or value

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
Product_name	STRING	NULLABLE	-	-	-	-	-
Purchase_Addres	STRING	NULLABLE	-	-	-	-	-
Order_IDD	INTEGER	NULLABLE	-	-	-	-	-
Quantity_Ordereds	INTEGER	NULLABLE	-	-	-	-	-
Price	FLOAT	NULLABLE	-	-	-	-	-
date_ordered	STRING	NULLABLE	-	-	-	-	-

EDIT SCHEMA VIEW ROW ACCESS POLICIES

The schema after change of datatype

The third step involved removing the time part of the date column.

```
#Step 3

Create Table `inspiring-list-409312.2019`.January_4 as
SELECT
Product_name,
Purchase_Addres,
Order_IDD,
Quantity_Ordereds,
Price,
  SUBSTR(date_ordered, 1, 10) AS New_Date
  FROM `inspiring-list-409312.2019.January_3`
```

to select just the date part of the column as a new column

January_3 QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW LINEAGE DATA PROFILE DATA QUALITY

Row	Product_name	Purchase_Addres	Order_IDD	Quantity_Ordere	Price	date_ordered
67	iPhone	811 Hickory St, Portland, OR 97...	141336	1	700.0	2009-01-19 18:23:00.0000000
68	iPhone	2 Cherry St, San Francisco, CA ...	144271	1	700.0	2028-01-19 23:02:00.0000000
69	iPhone	956 Madison St, Los Angeles, ...	150335	1	700.0	2001-01-19 14:32:00.0000000
70	iPhone	712 Cherry St, New York City, N...	144757	1	700.0	2012-01-19 11:45:00.0000000
71	iPhone	283 Cedar St, Atlanta, GA 30301	144538	1	700.0	2016-01-19 18:05:00.0000000
72	iPhone	593 Chestnut St, New York City,...	147083	1	700.0	2008-01-19 13:16:00.0000000
73	iPhone	205 7th St, New York City, NY 1...	145709	1	700.0	2029-01-19 02:00:00.0000000
74	iPhone	953 Hickory St, Los Angeles, C...	144339	1	700.0	2021-01-19 14:33:00.0000000
75	iPhone	916 Pine St, New York City, NY ...	144819	1	700.0	2018-01-19 16:28:00.0000000
76	iPhone	46 Cedar St, Los Angeles, CA 9...	150092	1	700.0	2007-01-19 09:05:00.0000000
77	iPhone	916 Church St, Los Angeles, CA...	145995	1	700.0	2023-01-19 17:12:00.0000000
78	iPhone	754 Cherry St, Boston, MA 022...	147364	1	700.0	2021-01-19 18:40:00.0000000
79	iPhone	166 10th St, Atlanta, GA 30301	141670	1	700.0	2031-01-19 12:52:00.0000000
80	iPhone	904 14th St, New York City, NY ...	147377	1	700.0	2019-01-19 21:00:00.0000000
81	iPhone	785 6th St, San Francisco, CA 9...	143067	1	700.0	2020-01-19 09:23:00.0000000
82	iPhone	210 12th St, Seattle, WA 98101	145339	1	700.0	2028-01-19 09:06:00.0000000
83	iPhone	606 Hill St, New York City, NY 1...	148369	1	700.0	2010-01-19 20:27:00.0000000
84	iPhone	331 Hickory St, Boston, MA 02...	145176	1	700.0	2030-01-19 11:03:00.0000000

from this

January_4 QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW LINEAGE DATA PROFILE DATA QUALITY

Row	Product_name	Purchase_Addres	Order_IDD	Quantity_Ordere	Price	New_Date
51	iPhone	817 Lincoln St, Seattle, WA 981...	143547	1	700.0	2001-01-19
52	iPhone	446 Pine St, Atlanta, GA 30301	141732	1	700.0	2001-01-19
53	iPhone	761 Lakeview St, Dallas, TX 75...	148450	1	700.0	2001-01-19
54	iPhone	628 Lake St, New York City, NY ...	144804	1	700.0	2001-01-19
55	iPhone	956 Madison St, Los Angeles, ...	150335	1	700.0	2001-01-19
56	iPhone	281 Washington St, Boston, M...	145820	1	700.0	2001-01-19
57	iPhone	279 9th St, San Francisco, CA 9...	142914	1	700.0	2001-01-19
58	iPhone	442 Jackson St, Atlanta, GA 30...	149926	1	700.0	2001-01-19
59	iPhone	171 Madison St, Los Angeles, ...	143600	1	700.0	2001-01-19
60	iPhone	901 12th St, Seattle, WA 98101	146085	1	700.0	2001-01-19
61	iPhone	738 8th St, San Francisco, CA 9...	141663	1	700.0	2001-01-19
62	iPhone	926 Forest St, San Francisco, C...	148948	1	700.0	2001-01-19
63	iPhone	719 7th St, New York City, NY 1...	147592	1	700.0	2001-01-19
64	iPhone	956 Walnut St, Seattle, WA 981...	142350	1	700.0	2001-01-19
65	iPhone	462 Cedar St, Boston, MA 02215	143738	1	700.0	2001-01-19
66	LG Dryer	338 Main St, Boston, MA 02215	142264	1	600.0	2001-01-19
67	20in Monitor	625 Elm St, Boston, MA 02215	142112	1	109.989997...	2001-01-19
68	20in Monitor	191 13th St, Dallas, TX 75001	150038	1	109.989997...	2001-01-19
69	20in Monitor	654 6th St, Atlanta, GA 30301	144857	1	109.989997...	2001-01-19
70	20in Monitor	434 Sunset St, Atlanta, GA 303...	141647	1	109.989997...	2001-01-19
71	Google Phone	903 Lake St, Seattle, WA 98101	141789	1	600.0	2001-01-19
72	Google Phone	734 River St, New York City, NY ...	142134	1	600.0	2001-01-19
73	Google Phone	229 Elm St, New York City, NY 1...	147451	1	600.0	2001-01-19
74	Google Phone	979 9th St, Portland, OR 97035	143897	1	600.0	2001-01-19

to this

Step 4 involved taking just the city part of the purchase address, rounding up the price to 2 decimal places and re-arranging the date column, drop the order ID as it is irrelevant for the analysis, while dropping all the

null values in the table. Also created a new column as total_cost which is the order number multiplied by the price of the product ordered.

```sql
#Step 4
Create Table `inspiring-list-409312.2019`.Jan as
SELECT
Product_name AS Product,
TRIM(SPLIT(Purchase_Addres, ',')[OFFSET(1)]) AS Purchase_Address,
Quantity_Ordereds AS Quantity_Ordered,
ROUND(Price, 2) AS Price_Each,
ROUND(Quantity_Ordereds * Price , 2) AS Total_cost,
  FORMAT_DATE('%m-%d', (PARSE_DATE ('%Y-%m-%d', CONCAT(
    SUBSTR(New_Date, 1, 2),
    SUBSTR(New_Date, 9, 2),
    '-',
    SUBSTR(New_Date, 6, 2),
    '-',
    SUBSTR(New_Date, 3, 2)
  )))) AS Order_Date
FROM `inspiring-list-409312.2019.January_4`
WHERE New_Date IS NOT NULL AND New_Date NOT LIKE '%Order_date%' AND New_Date NOT LIKE '%NU-LL%'
AND New_date NOT LIKE '%Orte-_D-de%'
AND Product_name <> 'Product'AND Product_name IS NOT NULL
  AND Price IS NOT NULL AND Purchase_Addres <> 'Purchase Address'
  ORDER BY Product_name, Purchase_Addres
```

SQL Code to isolate the city, delete null and empty values, calculate total cost, and round the price. ordered by Product and address

SCHEMA DETAILS PREVIEW LINEAGE DATA PROFILE DATA QUALITY

Row	Product	Purchase_Address	Quantity_Ordere	Price_Each	Total_cost	Order_Date
1	20in Monitor	New York City	1	109.99	109.99	01-22
2	20in Monitor	Boston	1	109.99	109.99	01-21
3	20in Monitor	San Francisco	1	109.99	109.99	01-13
4	20in Monitor	Seattle	1	109.99	109.99	01-09
5	20in Monitor	Atlanta	1	109.99	109.99	01-26
6	20in Monitor	Seattle	1	109.99	109.99	01-13
7	20in Monitor	San Francisco	1	109.99	109.99	01-23
8	20in Monitor	Portland	1	109.99	109.99	01-24
9	20in Monitor	Dallas	1	109.99	109.99	01-09
10	20in Monitor	Austin	1	109.99	109.99	01-30
11	20in Monitor	San Francisco	1	109.99	109.99	01-04
12	20in Monitor	Los Angeles	1	109.99	109.99	01-15
13	20in Monitor	Atlanta	1	109.99	109.99	01-07
14	20in Monitor	Austin	1	109.99	109.99	01-22
15	20in Monitor	San Francisco	1	109.99	109.99	01-18
16	20in Monitor	New York City	1	109.99	109.99	01-20
17	20in Monitor	Los Angeles	1	109.99	109.99	01-25
18	20in Monitor	Austin	1	109.99	109.99	01-10
19	20in Monitor	Seattle	1	109.99	109.99	01-12
20	20in Monitor	Los Angeles	1	109.99	109.99	01-18
21	20in Monitor	Los Angeles	2	109.99	219.98	01-23
22	20in Monitor	Boston	1	109.99	109.99	01-15
23	20in Monitor	New York City	1	109.99	109.99	01-18
24	20in Monitor	Dallas	1	109.99	109.99	01-01
25	20in Monitor	Portland	1	109.99	109.99	01-15
26	20in Monitor	New York City	1	109.99	109.99	01-17
27	20in Monitor	Los Angeles	1	109.99	109.99	01-20
28	20in Monitor	Dallas	1	109.99	109.99	01-23
29	20in Monitor	Los Angeles	1	109.99	109.99	01-06

new look of the table after step 4

These 4 steps were done for all the datasets . The next step was to combine the twelve datasets as a single table.

```
#Step 5
CREATE TABLE `inspiring-list-409312.2019`.year_2019 AS

SELECT * FROM `inspiring-list-409312.2019.Jan`
UNION ALL
SELECT * FROM `inspiring-list-409312.2019.Feb`
UNION ALL
SELECT * FROM `inspiring-list-409312.2019.Mar`

UNION ALL
SELECT * FROM `inspiring-list-409312.2019.Apr`

UNION ALL
SELECT * FROM `inspiring-list-409312.2019.May`

UNION ALL
SELECT * FROM `inspiring-list-409312.2019.Jun`

UNION ALL
SELECT * FROM `inspiring-list-409312.2019.Jul`

UNION ALL
SELECT * FROM `inspiring-list-409312.2019.Aug`

UNION ALL
SELECT * FROM `inspiring-list-409312.2019.Sep`

UNION ALL
SELECT * FROM `inspiring-list-409312.2019.Oct`

UNION ALL
SELECT * FROM `inspiring-list-409312.2019.Nov`

UNION ALL
SELECT * FROM `inspiring-list-409312.2019.Dec`
```

Code to combine all the twelve tables as one

The last cleaning step was done to convert the date column to a ‘%b’ date datatype since we are only interested in the months and not specific date of orders.

```
#Step 6
CREATE TABLE `inspiring-list-409312.2019`.year_merged AS

SELECT
Product,
Purchase_Address,
Quantity_Ordered,
Price_Each,
Total_cost,

  FORMAT_DATE('%b', PARSE_DATE('%Y-%m-%d', CONCAT(EXTRACT(YEAR FROM CURRENT_DATE()), '-', Order_Date))) AS


 FROM `inspiring-list-409312.2019.year_2019`
```

Format date ‘%b”

SCHEMA DETAILS PREVIEW LINEAGE DATA PROFILE DATA QUALITY

Row	Product	Purchase_Address	Quantity_Ordere	Price_Each	Total_cost	month
1	Vareebadd Phone	Boston	1	400.0	400.0	Apr
2	Vareebadd Phone	Seattle	1	400.0	400.0	Apr
3	Vareebadd Phone	New York City	1	400.0	400.0	Apr
4	Vareebadd Phone	San Francisco	1	400.0	400.0	Apr
5	Vareebadd Phone	Dallas	1	400.0	400.0	Apr
6	Vareebadd Phone	New York City	1	400.0	400.0	Apr
7	Vareebadd Phone	Boston	1	400.0	400.0	Apr
8	Vareebadd Phone	Boston	1	400.0	400.0	Apr
9	Vareebadd Phone	Atlanta	1	400.0	400.0	Apr
10	Vareebadd Phone	New York City	1	400.0	400.0	Apr
11	Vareebadd Phone	Atlanta	1	400.0	400.0	Apr
12	Vareebadd Phone	Seattle	1	400.0	400.0	Apr
13	Vareebadd Phone	San Francisco	1	400.0	400.0	Apr
14	Vareebadd Phone	Dallas	1	400.0	400.0	Apr
15	Vareebadd Phone	Atlanta	1	400.0	400.0	Apr
16	Vareebadd Phone	Seattle	1	400.0	400.0	Apr
17	Vareebadd Phone	New York City	1	400.0	400.0	Apr
18	Vareebadd Phone	Boston	1	400.0	400.0	Apr
19	Vareebadd Phone	Portland	1	400.0	400.0	Apr
20	Vareebadd Phone	Dallas	1	400.0	400.0	Apr
21	Vareebadd Phone	Seattle	1	400.0	400.0	Apr
22	Vareebadd Phone	Seattle	1	400.0	400.0	Apr
23	Vareebadd Phone	Los Angeles	1	400.0	400.0	Apr
24	Vareebadd Phone	Los Angeles	1	400.0	400.0	Apr
25	Vareebadd Phone	San Francisco	1	400.0	400.0	Apr
26	Vareebadd Phone	Austin	1	400.0	400.0	Apr
27	Vareebadd Phone	Austin	1	400.0	400.0	Apr
28	Vareebadd Phone	Los Angeles	1	400.0	400.0	Apr
29	Vareebadd Phone	New York City	1	400.0	400.0	Apr
30	Vareebadd Phone	San Francisco	1	400.0	400.0	Apr
31	Vareebadd Phone	Seattle	1	400.0	400.0	Apr
32	Vareebadd Phone	Los Angeles	1	400.0	400.0	Apr
33	Vareebadd Phone	San Francisco	1	400.0	400.0	Apr
34	Vareebadd Phone	San Francisco	1	400.0	400.0	Apr

Results per page: 50 1 – 50 of 185950

Cleaned combined table with 185,950 columns

iii. Identify key features/columns relevant to sales analysis

The key columns identified after examining the data were

a. Purchase Address

b. The Product name

c. Number of Orders

d. The month and

e. The total cost

and the metrics that will be required from them to be able to generate insights and make recommendations are

- Total Revenue
- Total Revenue Per Month
- Total Revenue Per Product
- Total Revenue Per Purchase Address
- Average Revenue per Month
- Average Revenue Per Product
- Average Revenue Per Purchase Address
- Total Revenue per Month per Product
- Total Quantity sold per product
- Total Quantity sold per Purchase Address
- Total Quantity sold per Month
- Average Unit sale (number of orders) per Product
- Average Unit sale per Purchase address

iv. Calculate and present key sales metrics

The Calculations were done using BigQuery

-

```
SELECT
  ROUND(SUM(Total_cost), 2) AS total_revenue
FROM
  `inspiring-list-409312.2019.year_merged`;
```

Total Revenue for 2019

Row	total_revenue
1	34492035.97

total revenue

-

```sql
SELECT
  month,
  ROUND(SUM(Total_cost),2) AS total_revenue_per_month
FROM
  `inspiring-list-409312.2019.year_merged`
GROUP BY
  month
  ORDER BY total_revenue_per_month
```

to calculate the total revenue per month

Row	month	total_revenue_per_m
1	Jan	1822256.73
2	Sep	2097560.13
3	Feb	2202022.42
4	Aug	2244467.88
5	Jun	2577802.26
6	Jul	2647775.76
7	Mar	2807100.38
8	May	3152606.75
9	Nov	3199603.2
10	Apr	3390670.24
11	Oct	3736726.88
12	Dec	4613443.34

The revenue(total cost) per month which showed December as the highest performing month and January as the smallest.

-

```
SELECT
  Product,
  ROUND(SUM(Total_cost), 2) AS total_revenue_per_product
FROM
  `inspiring-list-409312.2019.year_merged`
GROUP BY
  Product

  Order by total_revenue_per_product
```

to calculate the total revenue per product

Row	Product	total_revenue_per_pr
1	AAA Batteries (4-pack)	92740.83
2	AA Batteries (4-pack)	106118.4
3	Wired Headphones	246478.43
4	USB-C Charging Cable	286501.25
5	Lightning Charging Cable	347094.15
6	LG Dryer	387600.0
7	LG Washing Machine	399600.0
8	20in Monitor	454148.71
9	Vareebadd Phone	827200.0
10	27in FHD Monitor	1132424.5
11	Bose SoundSport Headphones	1345565.43
12	Flatscreen TV	1445700.0
13	Apple Airpods Headphones	2349150.0
14	34in Ultrawide Monitor	2355558.01
15	27in 4K Gaming Monitor	2435097.56
16	Google Phone	3319200.0
17	ThinkPad Laptop	4129958.7
18	iPhone	4794300.0
19	Macbook Pro Laptop	8037600.0

Macbook Pro Laptop grossed the highest and AAA Batteries (4-pack) generated the lowest.

-

```sql
SELECT
  Purchase_Address,
  ROUND(SUM(Total_cost), 2) AS total_revenue_per_city
FROM
  `inspiring-list-409312.2019.year_merged`
GROUP BY
  Purchase_Address

  Order by total_revenue_per_city
```

total_revenue_per_city

Row	Purchase_Address	total_revenue_per_ci
1	Austin	1819581.75
2	Portland	2320490.61
3	Seattle	2747755.48
4	Dallas	2767975.4
5	Atlanta	2795498.58
6	Boston	3661642.01
7	New York City	4664317.43
8	Los Angeles	5452570.8
9	San Francisco	8262203.91

The Most Performing cities are New York, Los Angeles and San Francisco

-

```sql
SELECT
month,
ROUND(AVG(Total_cost), 2) AS average_revenue_per_month

FROM
  `inspiring-list-409312.2019.year_merged`

GROUP BY
 month
 Order by average_revenue_per_month
```

average revenue per month

Row	month	average_revenue_pe
1	Sep	180.5
2	Nov	182.07
3	Feb	183.88
4	Oct	184.24
5	Dec	184.66
6	Jul	185.25
7	Mar	185.25
8	Apr	185.5
9	Aug	187.65
10	Jan	187.69
11	Jun	190.19
12	May	190.31

•

```sql
SELECT
  Product,
 ROUND(AVG(Total_cost), 2) AS average_revenue_per_product
FROM
  `inspiring-list-409312.2019.year_merged`
GROUP BY
 Product
 Order by average_revenue_per_product
```

average revenue per product

Row	Product	average_revenue_per
1	AAA Batteries (4-pack)	4.49
2	AA Batteries (4-pack)	5.16
3	Wired Headphones	13.05
4	USB-C Charging Cable	13.08
5	Lightning Charging Cable	16.03
6	Bose SoundSport Headphones	100.98
7	20in Monitor	110.74
8	27in FHD Monitor	150.85
9	Apple Airpods Headphones	151.08
10	Flatscreen TV	301.19
11	34in Ultrawide Monitor	381.1
12	27in 4K Gaming Monitor	390.87
13	Vareebadd Phone	400.58
14	LG Washing Machine	600.0
15	LG Dryer	600.0
16	Google Phone	600.76
17	iPhone	700.72
18	ThinkPad Laptop	1000.47
19	Macbook Pro Laptop	1701.44

•

```sql
SELECT
  Purchase_Address,
 ROUND(AVG(Total_cost), 2) AS average_revenue_per_city
FROM
  `inspiring-list-409312.2019.year_merged`
GROUP BY
 Purchase_Address
 Order by average_revenue_per_city
```

average revenue per city

Row	Purchase_Address	average_revenue_per
1	Boston	183.69
2	Austin	183.7
3	Los Angeles	184.18
4	San Francisco	184.7
5	Portland	186.16
6	Seattle	186.52
7	Dallas	186.77
8	New York City	187.5
9	Atlanta	187.86

•

```sql
SELECT
  month,
  Product,
  ROUND(SUM(Total_cost),2) AS total_revenue_per_month_per_product
FROM
  `inspiring-list-409312.2019.year_merged`
GROUP BY
  month,
  Product
ORDER BY
CASE
    WHEN month = 'Jan' THEN 1
    WHEN month = 'Feb' THEN 2
    WHEN month = 'Mar' THEN 3
    WHEN month = 'Apr' THEN 4
    WHEN month = 'May' THEN 5
    WHEN month = 'Jun' THEN 6
    WHEN month = 'Jul' THEN 7
    WHEN month = 'Aug' THEN 8
    WHEN month = 'Sep' THEN 9
    WHEN month = 'Oct' THEN 10
    WHEN month = 'Nov' THEN 11
    WHEN month = 'Dec' THEN 12
    ELSE 99
  END
```

total revenue per month per product

Row	month	Product	total_revenue_per_m
1	Jan	Vareebadd Phone	50400.0
2	Jan	Macbook Pro Laptop	399500.0
3	Jan	Apple Airpods Headphones	122700.0
4	Jan	Flatscreen TV	73200.0
5	Jan	Google Phone	191400.0
6	Jan	LG Dryer	23400.0
7	Jan	LG Washing Machine	25200.0
8	Jan	iPhone	266700.0
9	Jan	AA Batteries (4-pack)	5468.16
10	Jan	27in FHD Monitor	63295.78
11	Jan	ThinkPad Laptop	218997.81
12	Jan	Wired Headphones	13009.15
13	Jan	USB-C Charging Cable	15379.65
14	Jan	Lightning Charging Cable	17267.25
15	Jan	AAA Batteries (4-pack)	4784.0
16	Jan	20in Monitor	23977.82
17	Jan	Bose SoundSport Headphones	66193.38
18	Jan	27in 4K Gaming Monitor	122066.87
19	Jan	34in Ultrawide Monitor	119316.86
20	Feb	Vareebadd Phone	51600.0
21	Feb	Macbook Pro Laptop	469200.0
22	Feb	Apple Airpods Headphones	151800.0
23	Feb	Flatscreen TV	93900.0

Load more

Results per page: 50 1 – 50 of 228

-

```
SELECT
  Product,
  SUM(Quantity_Ordered) AS total_quantity_sold
FROM
  `inspiring-list-409312.2019.year_merged`
GROUP BY
  Product
  Order by total_quantity_sold
```

total quantity sold per product

Row	Product	total_quantity_sold
1	LG Dryer	646
2	LG Washing Machine	666
3	Vareebadd Phone	2068
4	20in Monitor	4129
5	ThinkPad Laptop	4130
6	Macbook Pro Laptop	4728
7	Flatscreen TV	4819
8	Google Phone	5532
9	34in Ultrawide Monitor	6199
10	27in 4K Gaming Monitor	6244
11	iPhone	6849
12	27in FHD Monitor	7550
13	Bose SoundSport Headphones	13457
14	Apple Airpods Headphones	15661
15	Wired Headphones	20557
16	Lightning Charging Cable	23217
17	USB-C Charging Cable	23975
18	AA Batteries (4-pack)	27635
19	AAA Batteries (4-pack)	31017

```sql
SELECT
  Purchase_Address,
  SUM(Quantity_Ordered) AS total_quantity_sold
FROM
  `inspiring-list-409312.2019.year_merged`
GROUP BY
  Purchase_Address
  Order by total_quantity_sold
```

total quantity sold per purchase address

Row	Purchase_Address	total_quantity_sold
1	Austin	11153
2	Portland	14053
3	Seattle	16553
4	Atlanta	16602
5	Dallas	16730
6	Boston	22528
7	New York City	27932
8	Los Angeles	33289
9	San Francisco	50239

•

```sql
SELECT
  month,
  SUM(Quantity_Ordered) AS total_quantity_sold
FROM
  `inspiring-list-409312.2019.year_merged`
GROUP BY
  month
  Order by total_quantity_sold
```

total quantity sold per month

Row	month	total_quantity_sold
1	Jan	10903
2	Sep	13109
3	Aug	13448
4	Feb	13449
5	Jun	15253
6	Jul	16072
7	Mar	17005
8	May	18667
9	Nov	19798
10	Apr	20558
11	Oct	22703
12	Dec	28114

```sql
SELECT
  Product,
 ROUND(AVG(Quantity_Ordered), 2) AS average_unitsale_per_product
FROM
  `inspiring-list-409312.2019.year_merged`
GROUP BY
 Product
 Order by average_unitsale_per_product
```

average unit sale per product

Row	Product	average_unitsale_pe
1	Vareebadd Phone	1.0
2	Macbook Pro Laptop	1.0
3	Flatscreen TV	1.0
4	LG Washing Machine	1.0
5	Google Phone	1.0
6	LG Dryer	1.0
7	iPhone	1.0
8	ThinkPad Laptop	1.0
9	27in 4K Gaming Monitor	1.0
10	34in Ultrawide Monitor	1.0
11	Apple Airpods Headphones	1.01
12	27in FHD Monitor	1.01
13	20in Monitor	1.01
14	Bose SoundSport Headphones	1.01
15	Lightning Charging Cable	1.07
16	Wired Headphones	1.09
17	USB-C Charging Cable	1.09
18	AA Batteries (4-pack)	1.34
19	AAA Batteries (4-pack)	1.5

•

```sql
SELECT
  Purchase_Address,
 ROUND(AVG(Quantity_Ordered), 2) AS average_unitsale_per_city
FROM
  `inspiring-list-409312.2019.year_merged`
GROUP BY
 Purchase_Address
 Order by average_unitsale_per_city
```

average unit sale per purchase address

Row	Purchase_Address	average_unitsale_pe
1	Seattle	1.12
2	New York City	1.12
3	San Francisco	1.12
4	Atlanta	1.12
5	Los Angeles	1.12
6	Boston	1.13
7	Dallas	1.13
8	Portland	1.13
9	Austin	1.13

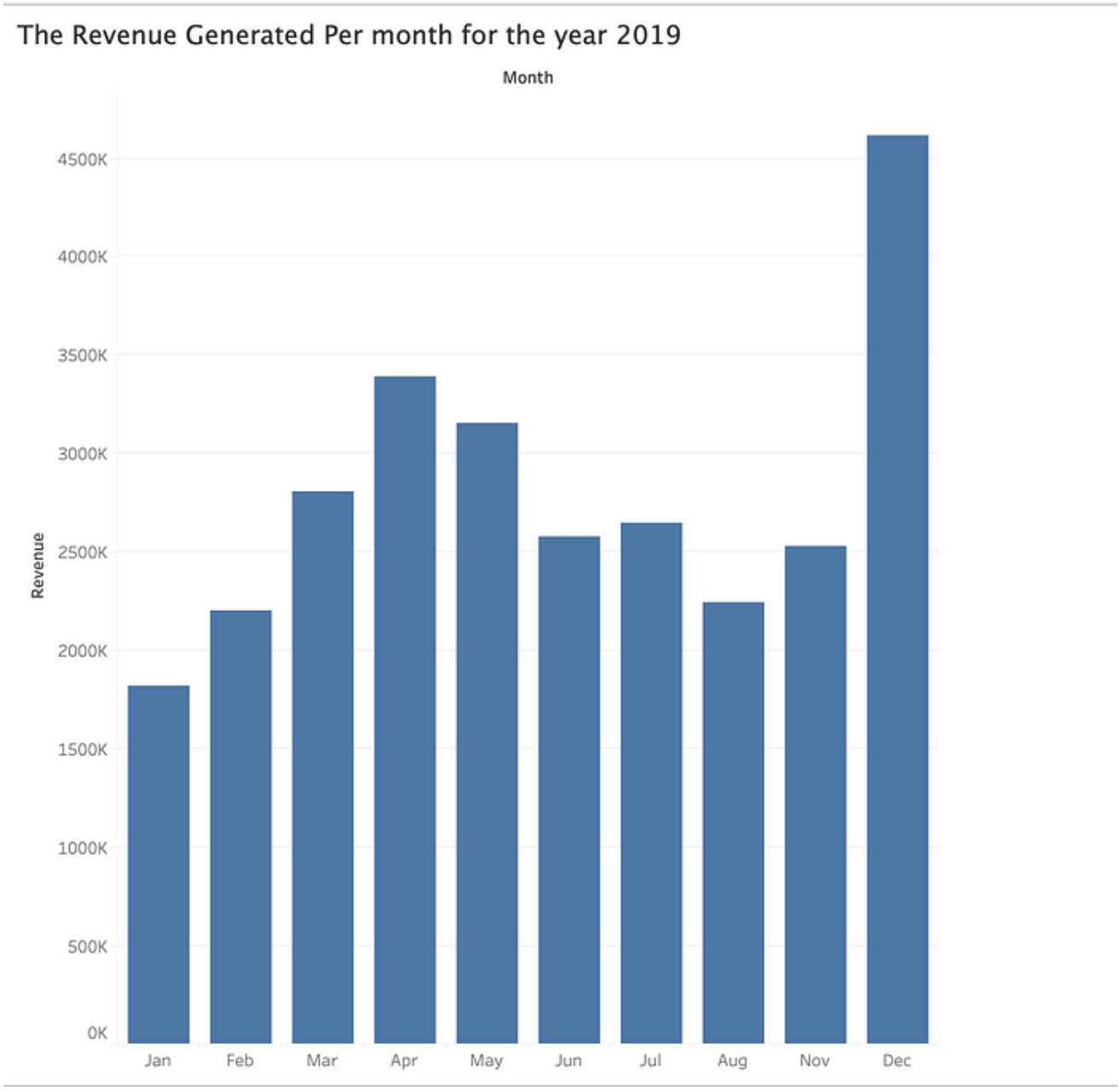
v. Share your Insight and make recommendations

Business Questions Asked

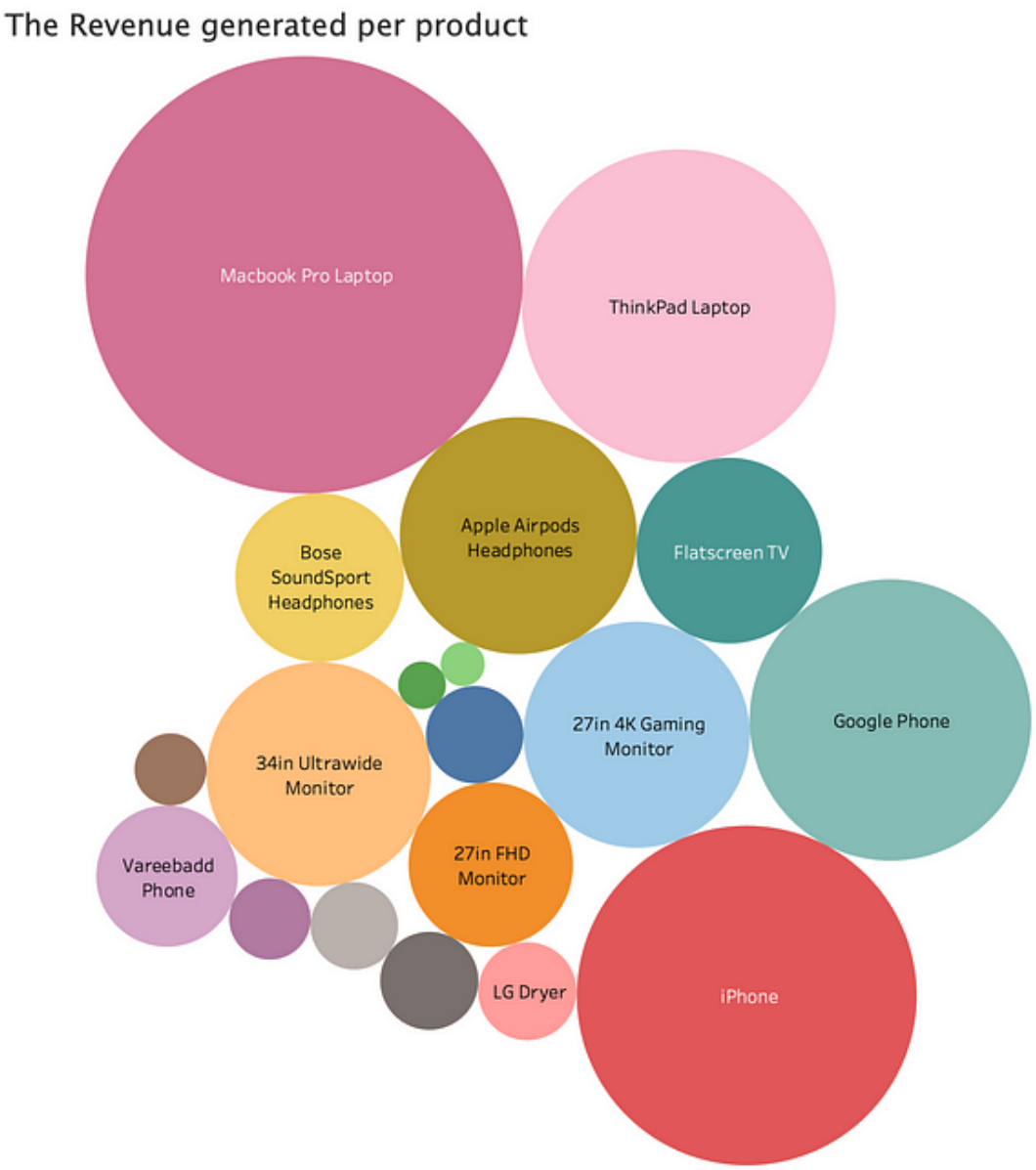
1. Uncover trends and patterns for the 2019 sales year
2. The Accountant reported that we made loss in the month of April, May, June and July as compared to other month. Is this true? What happened? Show monthly sales performance
3. The Assistant manager suggested that we should place more marketing attention on the following cities — Los Angeles, New York, Atlanta, San Francisco and Seattle as they seem to generate more revenue. From the result of your analysis, do you agree with this? Should we proceed with the suggestion?

Data-Driven Insights and Recommendation

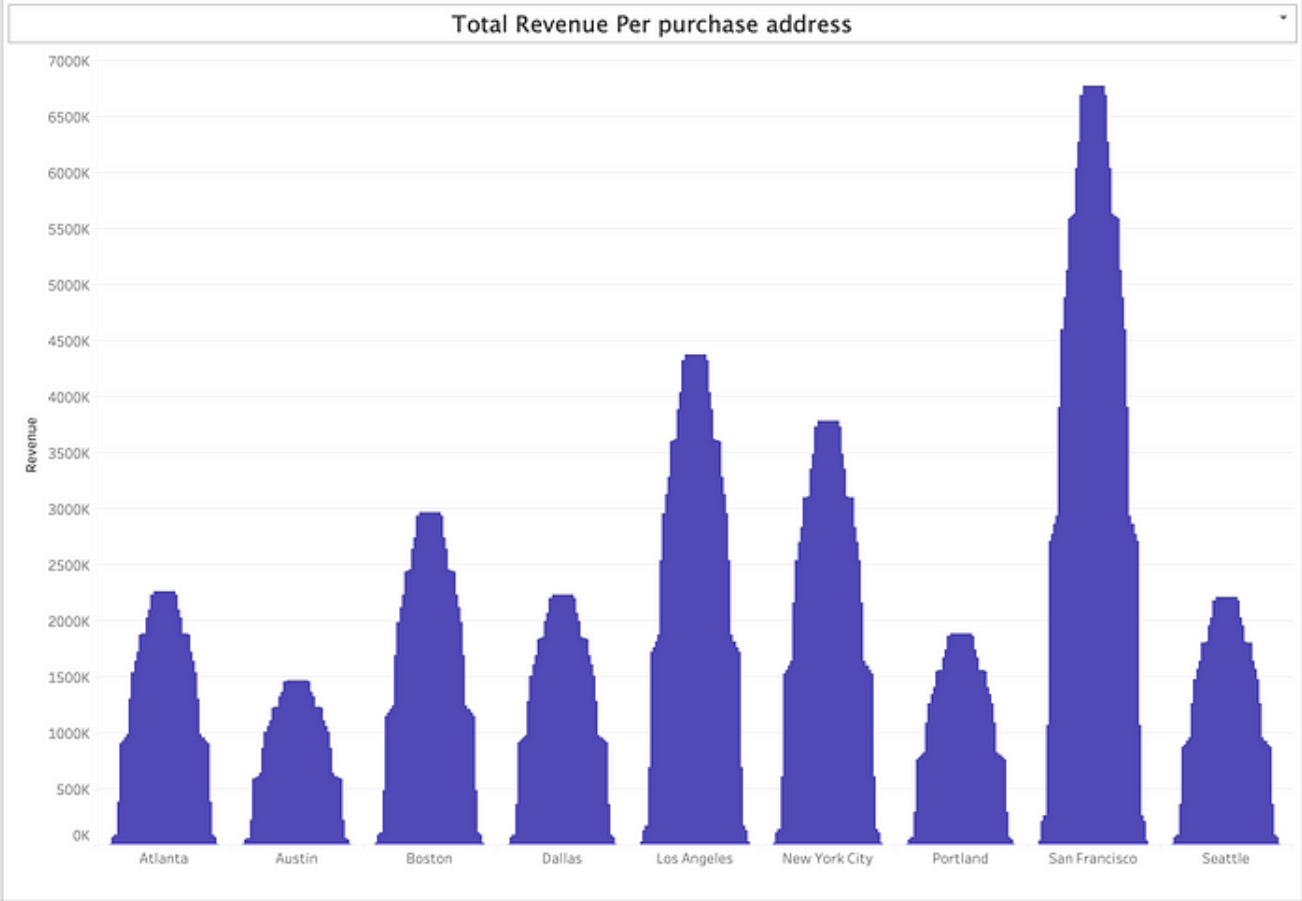
The Visualisations were done using the tableau public app



The Revenue chart per month showed December as the highest performing month and january as the lowest. Also, increase in revenue in the second and fourth quarter



The Macbook Pro Laptop, iPhone, ThinkPad Laptop and the Google Phone are on the top list of the products generating the most income for them.



Los Angeles, New York City, San Francisco are the stores where they generate the most revenue.

Total Revenue Per Month per Product

Product	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Nov	Dec
20in Monitor	23,978	27,058	35,857	43,226	37,507	35,417	35,967	28,707		62,584
27in 4K Gaming Monitor	122,067	166,526	186,805	220,734	211,375	175,885	188,365	171,596		336,951
27in FHD Monitor	63,296	71,245	91,194	110,393	103,343	85,044	91,044	73,645	107,843	144,290
34in Ultrawide Monitor	119,317	158,836	198,355	248,513	206,715	172,895	175,555	144,396		322,612
AA Batteries (4-pack)	5,468	6,662	8,509	10,833	9,155	8,049	7,953	7,020	1,382	14,300
AAA Batteries (4-pack)	4,784	5,896	7,412	8,788	8,752	6,464	7,071	6,043	75	12,681
Apple Airpods Headphones	122,700	151,800	198,300	227,850	204,750	175,050	183,000	151,350	226,950	311,400
Bose SoundSport Headph..	66,193	84,092	119,788	128,687	117,588	98,290	105,989	88,091		182,382
Flatscreen TV	73,200	93,900	108,600	138,000	119,700	110,100	119,700	99,300	138,000	199,800
Google Phone	191,400	228,600	277,800	348,000	288,000	234,600	246,600	216,600	295,800	429,000
iPhone	266,700	307,300	376,600	485,100	448,000	373,100	351,400	307,300	465,500	634,200
LG Dryer	23,400	22,800	29,400	46,800	45,600	25,800	33,000	27,600	33,600	51,600
LG Washing Machine	25,200	24,000	38,400	36,600	38,400	33,000	31,200	28,800	31,800	48,000
Lightning Charging Cable	17,267	22,410	28,001	35,476	31,021	25,221	27,074	21,573	12,872	46,151
Macbook Pro Laptop	399,500	469,200	644,300	771,800	790,500	605,200	625,600	508,300	748,000	1,093,100
ThinkPad Laptop	218,998	274,997	344,997	389,996	370,996	313,997	318,997	273,997	373,996	538,995
USB-C Charging Cable	15,380	19,765	23,231	27,115	24,713	20,016	22,096	17,471	22,514	38,838
Vareebadd Phone	50,400	51,600	69,600	88,000	74,000	62,000	58,400	57,200	70,800	113,600
Wired Headphones	13,009	15,335	19,951	24,759	22,493	17,673	18,764	15,479	576	32,961

Insight Summary

- The total revenue generated for 2019 is 34,492,035.97
- The total numbers of orders placed is 209,079
- The Sales revenue started slow, increased gradually till the second quarter, then dropped towards the end of the second quarter until it spiked back in the fourth quarter where it peaked
- The most sold product is the AAA Battery pack — 31, 017 and it generated the least revenue — 92,740.83
- The Macbook Pro Laptop generated the most revenue.

- December is the highest performing month generating 13.38% of the total revenue.
- The best performing cities are New York, San Francisco, and Los Angeles

Recommendations

1. To optimise performance and prioritise product categories for enhancement, the business should concentrate on the most successful segments. The Products that exhibited exceptional performance include . Therefore, it's recommended that they strategically expands its offerings within these categories while also enhancing the diversity of available products to cater to customer preferences.
2. Given the observed sales trend, focus marketing efforts during the Plan strategic promotions, campaigns, and product launches during these periods to leverage increased customer interest and buying behaviour.
3. Since the is the most sold but generates the least revenue, review the pricing strategy or explore ways to optimise the production cost or sourcing to increase profit margins. Simultaneously, ensure adequate inventory levels for high-demand products to meet customer needs promptly.
4. As New York, Los Angeles and San Francisco are the top-performing cities, allocate more resources towards targeted marketing campaigns in these locations. Tailor marketing strategies based on local preferences and behavior to enhance brand visibility and customer engagement.
5. Leverage the data on successful product categories and best-performing months to design loyalty programs or incentives for repeat customers. Offer personalized recommendations or exclusive deals to drive customer retention and increase lifetime value.
6. Continue leveraging data analytics to drive business decisions. Monitor sales trends, product performance, and geographical patterns regularly. Utilise predictive analytics to anticipate future trends and adjust strategies accordingly.
7. Explore collaborations with manufacturers or suppliers to develop exclusive products within the high-performing categories. Forge strategic alliances that can offer unique products, so as to stand out in the market.

Project Limitations

1. The dataset does not provide the cost price of the products to evaluate which product generates the most profit, and It lacks detail on customer demographics, specific product attributes, or detailed cost breakdowns, limiting a more comprehensive analysis.
2. The dataset does not include the cost of running the stores, and
3. The method of advertising wasn't specified, also which products are prioritised in the marketing adverts.
4. The dataset lacks detailed product-level information, such as variations within product categories or the introduction of new products, which could impact sales performance.
5. The data is focused on specific geographic regions, potentially overlooking broader market trends or global factors impacting the business.
6. The dataset represents a sales data from the year 2019. Changes in market trends, customer behaviour, or product offerings after 2019 is not captured, affecting the relevance of recommendations.

If you have recommendations, advise or tips, do not hesitate to send them across to me at donhalogen@gmail.com

Connect with me Here -