Data Analysis using SQL

The goal of data analysis is to give businesses actionable insights from the large amounts of data they receive every day. These insights can help organizations make decisions, solve problems, understand the needs of customers and identify future trends.

In this Project, we will gather insight from data we cleaned on Project 1 or Project 2, we will gather insight by performing table/data frame calculation. We will perform this process using SQL with MySQL Workbench.

We will be using several question guides to gather useful information from our data

Question guide list:

Question 1: What was the best time for sales?

Question 2: Which city sold the most product?

Question 3: What time should we display advertisements to maximize likelihood of customer's buying product?

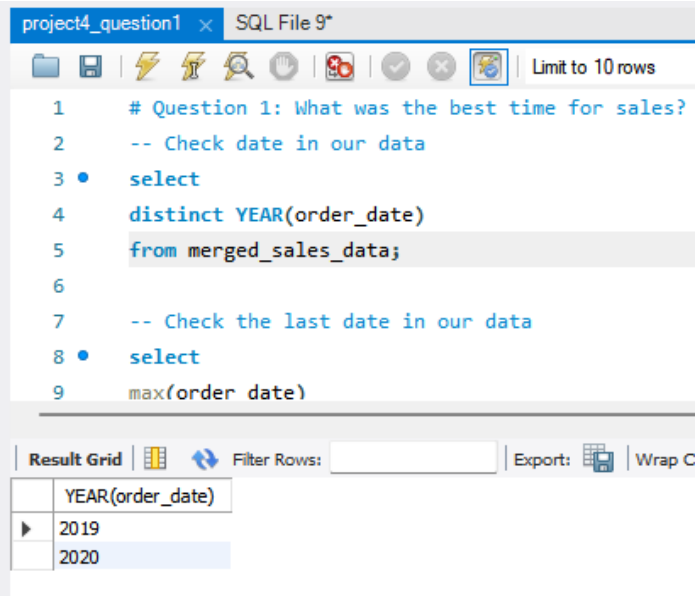
Question 4: What products are most often sold together?

Question 5: What product sold the most?

Question 6: Is there any correlation between quantity sold of product and product price?

Question 1: What was the best time for sales?

First, we need to find out is our data have any outliers.



We found our data have been contain 2019 and 2020 record.

```
-- Check the last date in our data
select
max(order_date)
from merged_sales_data;
```

Result Grid | Filter Rows: | Export:

max(order_date)
2020-01-01 05:13:00

But when we check the last date in the data, it seems 2020 data is just an outlier. Its better to exclude these outliers to perform consistency in our calculation.

```
-- Aggregate total_cost to find monthly sales
select
month,
round(sum(total_cost)) as monthly_total
from merged_sales_data
where order_date  BETWEEN '2019-01-01' AND '2020-01-01'
group by month
order by monthly_total desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

month	monthly_total
December	4608296
October	3734778
April	3389218
November	3197875
May	3150616
March	2804973
July	2646461
June	2576280

Lets calculate in quarter level too

```sql
-- Aggregate total_cost to find quarterly sales
select
quarter(order_date),
round(sum(total_cost)) as quarterly_total
from merged_sales_data
where order_date  BETWEEN '2019-01-01' AND '2020-01-01'
group by quarter(order_date)
order by quarterly_total desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

quarter(order_date)	quarterly_total
4	11540949
2	9116114
3	6982010
1	6817794

With this analysis we can answer the first question.

The best time for sales from consideration by sales data in 2019 is on quarter 4 and on December.

Question 2: Which city sold the most product?

```sql
# Question 2: Which city sold the most product?
-- Aggregate quantity_ordered by city
select
city,
sum(quantity_ordered) as total_unit_sold,
round(sum(total_cost)) as city_total_sales
from merged_sales_data
group by city
order by total_unit_sold desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

city	total_unit_sold	city_total_sales
San Francisco	50169	8254744
Los Angeles	33247	5448304
New York City	27903	4661867
Boston	22494	3658628
Dallas	16707	2765374
Atlanta	16584	2794199
Seattle	16534	2745046

With this analysis we can answer the second question.

City with the most sold product is San Fransisco with 50,169 units sold and sales about $8 million.

Question 3: What time should we display advertisements to maximize likelihood of customer's buying product?

```sql
# Question 3: What time should we display advertisements to maximize likelihood of customer's buying product?
-- Check count in hour level
select
hour(order_date) as order_hour,
count(*) as order_count
from
merged_sales_data
group by order_hour
order by order_count desc ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

order_hour	order_count
19	12886
12	12573
11	12392
18	12263
20	12218
13	12115
14	10965

Let’s try with interval 3 hours

```sql
-- Another count with interval 3 hours
select
CASE WHEN hour(order_date)>=0 and hour(order_date)<=2 THEN '12 AM - 3 AM'
        WHEN hour(order_date)>=3 and hour(order_date)<=5 THEN '3 AM - 6 AM'
        WHEN hour(order_date)>=6 and hour(order_date)<=8 THEN '6 AM - 9 AM'
        WHEN hour(order_date)>=9 and hour(order_date)<=11 THEN '9 AM - 12 PM'
        WHEN hour(order_date)>=12 and hour(order_date)<=14 THEN '12 PM - 3 PM'
        WHEN hour(order_date)>=15 and hour(order_date)<=17 THEN '3 PM - 6 PM'
        WHEN hour(order_date)>=18 and hour(order_date)<=20 THEN '6 PM - 8 PM'
        WHEN hour(order_date)>=21 and hour(order_date)<=23 THEN '9 PM - 12 AM'
    END as hour_interval,
count(*) as order_count
FROM
merged_sales_data
group by hour_interval
order by order_count DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

hour_interval	order_count
6 PM - 8 PM	37367
12 PM - 3 PM	35653
9 AM - 12 PM	32061

We will get more precise analysis with python since we can analysis peak in time series with visualization. With SQL we can manually make it in interval, but if we try calculate with 3 hours interval is still work bot less precision.

With this analysis we can answer the third question.

Best time should we display advertisements to maximize likelihood of customer's buying product is between 11 am - 1 pm and between 6 pm - 8 pm.

Question 4: What products are most often sold together?

To answer this question, we need to count combination of product with same Order ID

We will count in 2 level, 2 combination product and 3 combination products

2 product combination

```sql
# Question 4: What products are most often sold together?

## 2 products combination
-- using INNER JOIN
SELECT a.product AS original_SKU, b.product AS bought_with, count(a.order_id) as times_bought_together
FROM merged_sales_data AS a
INNER JOIN merged_sales_data AS b ON a.order_id = b.order_id
AND a.product < b.product
GROUP BY a.product,b.product
order by times_bought_together DESC;

## 3 products combination
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

original_SKU	bought_with	times_bought_together
iPhone	Lightning Charging Cable	1011
Google Phone	USB-C Charging Cable	997
iPhone	Wired Headphones	462

With this analysis we can answer the fourth question

2 Combination products which most often sold together is IPhone & Lightning Charging Cable.

3 product combination

In this part we will perform query using:

1. Subquery
2. Window function: row_number

Queries:

```sql
-- Find order ID with 3 different products
select
order_id from(
select
*,
row_number() over(partition by order_id order by order_id) as row_num
from merged_sales_data) as t1
where row_num =3;
```

```sql
-- Select all column with order id that contained 3 products
SELECT * FROM merged_sales_data
WHERE order_id IN (select order_id from( select *,
                    row_number() over(partition by order_id order by order_id) as row_num
                        from merged_sales_data) as t1
                        where row_num =3);
```

```sql
-- Combine 3 products listed into 1 rows using GROUP_CONCAT
-- from dataset we created above (all columns only contain order id with 3 products)
select
3Product_combination, count(3Product_combination)
from (select order_id, group_concat(product separator ",") as 3Product_combination
        from (
                SELECT * FROM merged_sales_data
                WHERE order_id IN (
                                select order_id from(
                                select *, row_number() over(partition by order_id order by order_id, product) as row_num
            -- Make sure row_number() order by order_id & product
            -- so we won't meet value like ABC, ACB, BCA in seperate rows
                                from merged_sales_data) as t1
                    where row_num =3)) t2
        group by order_id) as t3
group by 3Product_combination
order by count(3Product_combination) desc;
```

Result:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

3Product_combination	count(3Product_combination)
Google Phone,USB-C Charging Cable,Wired Headphones	80
iPhone,Lightning Charging Cable,Wired Headphones	57
Apple Airpods Headphones,iPhone,Lightning Charging Cable	43
USB-C Charging Cable,Vareebadd Phone,Wired Headphones	31
Bose SoundSport Headphones,Google Phone,USB-C Charging Cable	31

With this analysis we can answer the fourth question

3 Combination products which most often sold together is Google Phone, USB-C Charging Cable, Wired Headphones.

Question 5: What product sold the most?

```sql
SELECT
product,
SUM(quantity_ordered)
FROM
merged_sales_data
GROUP BY product
ORDER BY SUM(quantity_ordered) DESC
```

Result Grid | Filter Rows: | Export: | Wrap Cel

product	SUM(quantity_ordered)
AAA Batteries (4-pack)	30986
AA Batteries (4-pack)	27615
USB-C Charging Cable	23931
Lightning Charging Cable	23169
Wired Headphones	20524
Apple Airpods Headphones	15637
Bose SoundSport Headphones	13430

With this analysis we can answer the fifth question

Product that sold the most is AAA Batteries (4-pack)

Question 6: Is there any correlation between quantity sold of product and product price?

In SQL we can't do visualization, so we can find the correlation using statistic calculation method

```sql
# Question 6: Is there any correlation between quantity sold of product and product price?
-- Using statistic calculation of correlation
select  @ax := avg(quantity_ordered),
        @ay := avg(price_each),
        @div := (stddev_pop(quantity_ordered) * stddev_pop(price_each))
from merged_sales_data;

select sum( ( quantity_ordered - @ax ) * (price_each - @ay) ) / ((count(quantity_ordered) -1) * @div)
from merged_sales_data;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

sum((quantity_ordered - @ax) * (price_each - @ay)) / ((count(quantity_ordered) -1) * @div)
-0.1484223502645759

With this analysis we can answer the fifth question

There is negative correlation between quantity ordered and product price. But the result of correlation coefficient is quite small (-0.148), hard to say that 2 variables not correlate enough because when we

perform analysis using visualization in Python, there actually has quite big correlation with some inconsistency.

So, I conclude this correlation analysis with SQL not precise enough because we can’t detect inconsistency in some products.