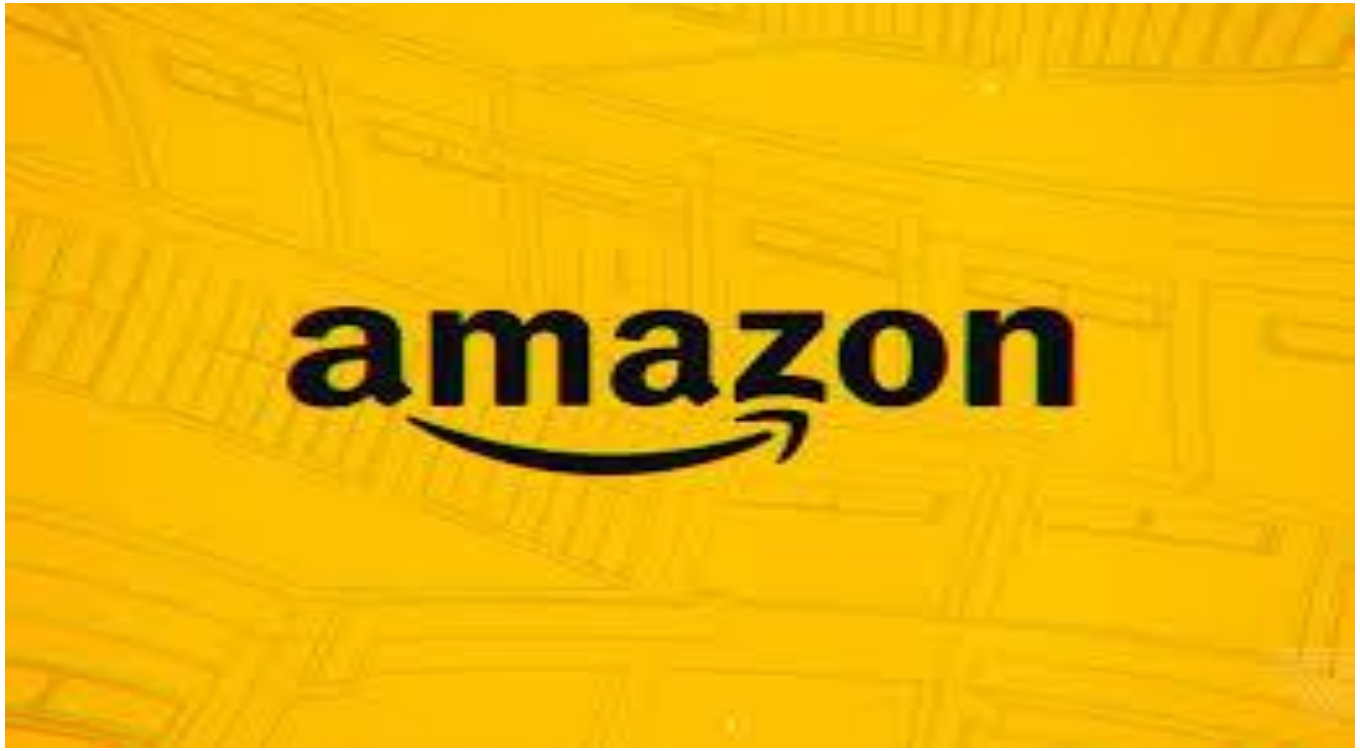


AMAZON SALES DATA

SQL PROJECT PRESENTATION

Summary of Amazon Sales Data

- The data consists of sales record of three cities/branch in Myanmar which are Naypyitaw, Yangon, Mandalay which took place in first quarter of year 2019 . The data consists of 1000 rows and 17 columns.

Objective of Project

- The major aim of this project is to gain insight into the sales data of Amazon to understand the different factors that affect sales of the different branches

Column	Description	Data Type
Invoice Id	Invoice of the sales made	Varchar(30)
Branch	Branch at which sales were made	Varchar(5)
City	The location of the branch	Varchar(30)
Customer Type	The type of the customer	Varchar(30)
Gender	Gender of the customer making purchase	Varchar(10)
Product Line	Product line of the product sold	Varchar(100)
Unit Price	The price of each product	Decimal(10,2)
Quantity	The amount of the product sold	Int
VAT	The amount of tax on the purchase	Float
Total	The total cost of the purchase	Decimal(10,2)
Date	The date on which the purchase was made	Date
Time	The time at which the purchase was made	Time
Payment Method	The total amount paid	Varchar(15)
Cogs	Cost Of Goods sold	Decimal(10,2)
Gross Margin Percentage	Gross margin percentage	Float
Gross Income	Gross Income	Decimal(10,2)
Rating	Rating	Decimal(3,1)

Step [1]: Created a database named Amazon in MS SQL SERVER

```
--Data Wrangling--
--Creating database and importing the data which is in the form of csv.file

create database amazon

use amazon
```

Step [2]: Checking null values and datatypes of columns of demo amazon table.

Note: as observe the datatype are incorrect and column names contain space which is syntactically incorrect, also table has no null values. This correction is done in EDA.

```sql
--Checking null values and datatypes of columns of demo amazon table.
SELECT
    TABLE_SCHEMA,
    TABLE_NAME,
    COLUMN_NAME,
    DATA_TYPE,
    IS_NULLABLE
FROM
    INFORMATION_SCHEMA.COLUMNS
ORDER BY
    TABLE_SCHEMA,
    TABLE_NAME,
    COLUMN_NAME
```

55 %

Results | Messages

	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	DATA_TYPE	IS_NULLABLE
1	dbo	Amazon	Branch	nvarchar	YES
2	dbo	Amazon	City	nvarchar	YES
3	dbo	Amazon	cogs	float	YES
4	dbo	Amazon	Customer_type	nvarchar	YES
5	dbo	Amazon	Date	date	YES
6	dbo	Amazon	day_name	varchar	YES
7	dbo	Amazon	Gender	nvarchar	YES
8	dbo	Amazon	gross_income	float	YES

Feature Engineering

In this step we are creating new columns named **timeofday**, **dayname**, **monthname** by extracting values from date and time column. This will help us to analyse and answer sales based on time-of-day (Morning, Afternoon, Evening), day-of-week (Sunday to Saturday) and month (Jan-March).

```sql
select time_of_day,day_name,month_name from amazon
```

161 %

Results | Messages

	time_of_day	day_name	month_name
1	Afternoon	Saturday	January
2	Morning	Friday	March
3	Afternoon	Sunday	March
4	Evening	Sunday	January
5	Morning	Friday	February
6	Evening	Monday	March
7	Afternoon	Monday	February
8	Morning	Sunday	February
9	Afternoon	Thursday	January
10	Afternoon	Wednes...	February
11	Evening	Wednes...	February
12	Afternoon	Saturday	March
13	Morning	Tuesday	February
14	Afternoon	Thursday	February
15	Evening	Friday	March
16	Afternoon	Tuesday	January
17	Morning	Monday	March
18	Morning	Tuesday	January

Query executed successfully.

```sql
--Now we are creating the new columns which decribed about the timeofday,dayname,monthname
--from the columns named time and date

alter table amazon add time_of_day varchar(15)

update amazon set time_of_day = case when datepart(hour,time) between 06 and 11 then 'Morning'
                                     when datepart(hour,time) between 12 and 17 then 'Afternoon'
                                     else 'Evening'
                                end

alter table amazon add day_name varchar(10)
update amazon set day_name = datename(weekday,date)
select day_name from amazon

alter table amazon add month_name varchar(10)

update amazon set month_name = datename(month,date)
select month_name from amazon
select time_of_day,day_name,month_name from amazon
```

Exploratory Data Analysis

Step [1]: Creating new table named **Amazon Sales** by adding correct column names, datatypes, constraints while copying values from demo table Amazon.

```sql
-- Exploratory data analysis
-- 1.Creating the new table amazon_sales by creating the columns as same as the table amazon for getting the values

create table amazon_sales
(invoice_id varchar(30) primary key not null,
branch varchar(5) not null,
city varchar(30) not null,
customer_type varchar(30) not null,
gender varchar(10) not null,
product_line varchar(100) not null,
unit_price decimal(10,2) not null,
quantity int not null,
vat float not null,
total decimal(10,2) not null,
date date not null,
time time not null,
payment_method varchar(20) not null,
cogs decimal(10,2) not null,
gross_margin_percentage float not null,
gross_income decimal(10,2) not null,
rating decimal(3,1) not null,
time_of_day varchar(15) not null,
day_name varchar(10) not null,
month_name varchar(10) not null)

insert into amazon_sales
select * from amazon
select* from amazon_sales
```

77 %

Results | Messages

	invoice_id	branch	city	customer_type	gender	product_line	unit_price	quantity	vat	total	date	time	payment_method	cogs	gross_margin_percentage	gross_income	rating	time_of_day
1	101-17-6199	A	Yangon	Normal	Male	Food and beverages	45.79	7	16.0265007019043	336.56	2019-03-13	19:44:00.0000000	Credit card	320.53	4.7619047164917	16.03	7.0	Evening
2	101-81-4070	C	Naypyitaw	Member	Female	Health and beauty	62.82	2	6.28200006484985	131.92	2019-01-17	12:36:00.0000000	Ewallet	125.64	4.7619047164917	6.28	4.9	Afternoon
3	102-06-2002	C	Naypyitaw	Member	Male	Sports and travel	25.25	5	6.3125	132.56	2019-03-20	17:52:00.0000000	Cash	126.25	4.7619047164917	6.31	6.1	Afternoon
4	102-77-2261	C	Naypyitaw	Member	Male	Health and beauty	65.31	7	22.8584995269775	480.03	2019-03-05	18:02:00.0000000	Credit card	457.17	4.7619047164917	22.86	4.2	Evening
5	105-10-6182	A	Yangon	Member	Male	Fashion accessories	21.48	2	2.14800000190735	45.11	2019-02-27	12:22:00.0000000	Ewallet	42.96	4.7619047164917	2.15	6.6	Afternoon
6	105-31-1824	A	Yangon	Member	Male	Sports and travel	69.52	7	24.3320007324219	510.97	2019-02-01	15:10:00.0000000	Credit card	486.64	4.7619047164917	24.33	8.5	Afternoon
7	106-35-6779	A	Yangon	Member	Male	Home and lifestyle	44.34	2	4.43400001525879	93.11	2019-03-27	11:26:00.0000000	Cash	88.68	4.7619047164917	4.43	5.8	Morning
8	109-28-2512	B	Mandalay	Member	Female	Fashion accessories	97.61	6	29.2830009460449	614.94	2019-01-07	15:01:00.0000000	Ewallet	585.66	4.7619047164917	29.28	9.9	Afternoon
9	109-86-4363	B	Mandalay	Member	Female	Sports and travel	60.08	7	21.0279998779297	441.59	2019-02-14	11:36:00.0000000	Credit card	420.56	4.7619047164917	21.03	4.5	Morning
10	110-05-6330	C	Naypyitaw	Normal	Female	Food and beverages	39.43	6	11.8290004730225	248.41	2019-03-25	20:18:00.0000000	Credit card	236.58	4.7619047164917	11.83	9.4	Evening
11	110-48-7033	B	Mandalay	Member	Male	Fashion accessories	32.62	4	6.52400016784668	137.00	2019-01-29	14:12:00.0000000	Cash	130.48	4.7619047164917	6.52	9.0	Afternoon

Step [2]: Checking size of table, unique values in columns.

```sql
--2.Checking the size of the table ,no of rows,unique values
select count(*) as total_columns from information_schema.columns
where table_name ='amazon_sales'
```

150 %

Results | Messages

	total_columns
1	20

```sql
select count(*) as total_rows from amazon_sales
```

150 %

Results | Messages

	total_rows
1	1000

```sql
create view count_unique_values as
(select count(distinct invoice_id) invoice_id, count(distinct branch) branch, count(distinct city) city, count(di
count(distinct gender) gender, count(distinct product_line) product_line, count(distinct unit_price) unit_price,
count(distinct vat) vat, count(distinct total) total, count(distinct date) date, count(distinct time) time, count
count(distinct cogs) cogs, count(distinct gross_margin_percentage) gross_margin_percentage, count(distinct gross_
count(distinct time_of_day) time_of_day, count(distinct day_name) day_name, count(distinct month_name) month_name

select * from count_unique_values
```

150 %

Results | Messages | Client Statistics

	invoice_id	branch	city	customertype	gender	product_line	unit_price	quantity	vat	total	date	time	payment_method	cogs	gross_margin_percentage	gross_income	rating	time_of_day	day_name	month_name
1	1000	3	3	2	2	6	943	10	990	990	89	506	3	990	1	883	61	3	7	3

Step [3]: Checking the unique values in each categorical column. There are 10 categorical columns **[invoice_id, branch, city, customer_type, gender, product_line, payment_method, time_of_day, day_name, month_name]**

branch
A
C
B

city
Yangon
Naypyitaw
Mandalay

time_of_day
Evening
Afternoon
Morning

month_name
March
January
February

payment_method
Credit card
Ewallet
Cash

gender
Male
Female

customer_type
Normal
Member

day_name
Wednesday
Thursday
Tuesday
Friday
Monday
Saturday
Sunday

product_line
Food and beverages
Health and beauty
Sports and travel
Fashion accessories
Home and lifestyle
Electronic accessories

Answering Business Questions

Q.1] What is the count of distinct cities in the dataset?

```sql
--1.What is the count of distinct cities in the dataset?
select count(distinct city) city from amazon_sales
```

150 %

Results | Messages | Client Statistics

	city
1	3

Q.2] For each branch, what is corresponding city?

```sql
-- 2.For each branch, what is the corresponding city?
select distinct branch ,city from amazon_sales
```

150 %

Results | Messages | Client Statistics

	branch	city
1	B	Mandalay
2	C	Naypyitaw
3	A	Yangon

Q.3] What is the count of distinct product lines in the dataset?

```sql
-- 3.What is the count of distinct product lines in the dataset?
select count(distinct product_line) product_line from amazon_sales
```

150 %

Results | Messages | Client Statistics

	product_line
1	6

Q.4] Which payment method occurs most frequently?

```sql
-- 4.Which payment method occurs most frequently?
select count(*) as occurance, payment_method from amazon_sales
group by payment_method
order by occurance
```

150 %

Results | Messages

	occurance	payment_method
1	311	Credit card
2	344	Cash
3	345	Ewallet

Q.5] Which product line has the highest sales?

```sql
-- 5.Which product line has the highest sales?
select  product_line,sum(quantity) as highest_sales from amazon_sales
group by product_line
order by highest_sales desc
```

150 %

Results | Messages

	product_line	highest_sales
1	Electronic accessories	971
2	Food and beverages	952
3	Sports and travel	920
4	Home and lifestyle	911
5	Fashion accessories	902
6	Health and beauty	854

Q.6] How much revenue is generated each month?

```sql
--6.How much revenue is generated each month?
select month_name,sum(total) as total_revenue from amazon_sales
group by month_name
order by total_revenue desc
```

50 %

Results | Messages

	month_name	total_revenue
1	January	116291.87
2	March	109455.54
3	February	97219.39

Q.7] Which product line generated highest revenue?

```sql
--7.In which month did the cost of goods sold reach its peak?
select top 1 month_name,sum(cogs) as cost_of_goods_sold from amazon_sales
group by month_name
order by cost_of_goods_sold desc
```

150 %

Results | Messages

	month_name	total_revenue
1	January	116291.87
2	March	109455.54
3	February	97219.39

Q.8] In which month cost of goods sold reach its peak?

```sql
--8.Which product line generated the highest revenue?
select product_line,sum(total) as revenue from amazon_sales
group by product_line
order by revenue desc
```

150 %

Results | Messages

	month_name	total_revenue
1	January	116291.87
2	March	109455.54
3	February	97219.39

Q.9] Which city has the highest revenue recorded?

```sql
--9.In which city was the highest revenue recorded?
select city,sum(total) as revenue from amazon_sales
group by city
order by revenue desc
```

50 %

Results | Messages

	city	revenue
1	Naypyitaw	110568.72
2	Yangon	106200.34
3	Mandalay	106197.74

Q.10] Which product line incurred the highest value added tax?

```sql
--10.Which product line incurred the highest Value Added Tax?
select product_line,max(vat) as highest_value from amazon_sales
group by product_line
order by highest_value desc
```

150 %

Results | Messages

	product_line	highest_value
1	Fashion accessories	49.6500015258789
2	Food and beverages	49.2599983215332
3	Home and lifestyle	48.75
4	Sports and travel	47.7200012207031
5	Health and beauty	45.25
6	Electronic accessories	44.8785018920898

Q.11] Which customer type occurs most frequently?

```sql
--11.Which customer type occurs most frequently?
select count(*) as occurance,customer_type from amazon_sales
group by customer_type
order by occurance desc
```

150 %

Results | Messages

	occurance	customer_type
1	501	Member
2	499	Normal

Q.12] For each product line, add a column indicating "Good" if its sales are above average, otherwise "Bad."

```sql
--11.For each product line, add a column indicating "Good" if its sales are above average, otherwise "Bad."
select product_line,case
                         when sum(total)>(select (sum(total)/count(distinct product_line)) from amazon_sales)
                         then 'good' else 'bad' end as performance
                        from amazon_sales
group by product_line

--select (sum(total)/count(distinct product_line)) from amazon_sales
```

150 %

Results | Messages

	product_line	performance
1	Fashion accessories	good
2	Health and beauty	bad
3	Electronic accessories	good
4	Food and beverages	good
5	Sports and travel	good
6	Home and lifestyle	good

Q.13] Which branch exceeded the average number of product sold?

```sql
--13.Identify the branch that exceeded the average number of products sold.

--select avg(quantity) from amazon_sales
select branch ,sum(quantity)from amazon_sales
group by branch
having sum(quantity)>(select sum(quantity)/count(distinct branch) from amazon_sales)
```

150 %

Results | Messages

	branch	(No column name)
1	A	1859

Q.14] Which product line is most frequently associated with each gender?

```sql
--14.Which product line is most frequently associated with each gender?

with jack as(select count(*) as depp,product_line,gender from amazon_sales
group by product_line,gender)

select max(depp),gender from jack
group by gender
```

150 %

Results | Messages

	(No column name)	gender
1	96	Female
2	88	Male

Q.15] What is the count of distinct customer types in the dataset?

```sql
--15.What is the count of distinct customer types in the dataset?
select count(distinct customer_type) as count_of_distinct_customer_type from amazon_sales
```

50 %

Results | Messages

	(No column name)	gender
1	96	Female
2	88	Male

Q.16] Calculate the average rating for each product line.

```sql
--16.Calculate the average rating for each product line.
select product_line,avg(rating)as avg_rating from amazon_sales
group by product_line
order by avg_rating desc
```

150 %

Results | Messages

	product_line	avg_rating
1	Food and beverages	7.113218
2	Fashion accessories	7.029213
3	Health and beauty	7.003289
4	Electronic accessories	6.924705
5	Sports and travel	6.916265
6	Home and lifestyle	6.837500

Q.17] Identify the customer type contributing the highest revenue.

```sql
--17.Identify the customer type contributing the highest revenue.
select top 1 customer_type,sum(total) as revenue from amazon_sales
group by customer_type
order by revenue desc
```

150 %

Results | Messages

	product_line	avg_rating
1	Food and beverages	7.113218
2	Fashion accessories	7.029213
3	Health and beauty	7.003289
4	Electronic accessories	6.924705
5	Sports and travel	6.916265
6	Home and lifestyle	6.837500

Q.18] Count the sales occurrences for each time of day on every weekday.

```sql
--15.Count the sales occurrences for each time of day on every weekday.
select count(quantity) as occrance,time_of_day,day_name from amazon_sales
group by time_of_day,day_name
order by  case day_name
                        when 'Sunday' then 1
                        when 'Monday' then 2
                        when 'Tuesday' then 3
                        when 'wednesday' then 4
                        when 'Thursday' then 5
                        when 'Friday' then 6
                        when 'Saturday' then 7
                        else 8
        end ,
case time_of_day when 'Morning' then 1
                 when 'Afternoon' then 2
                 when'Evening' then 3
                 else 4
end
```

150 %

Results | Messages

	occrance	time_of_day	day_name
1	22	Morning	Sunday
2	70	Afternoon	Sunday
3	41	Evening	Sunday
4	21	Morning	Monday
5	75	Afternoon	Monday
6	29	Evening	Monday

Q.19] Determine city with highest VAT percentage.

```sql
--19.Determine the city with the highest VAT percentage.
select city,max(vat) as vat_percentage from amazon_sales
group by city
order by vat_percentage
```

150 %

Results | Messages

	city	vat_percentage
1	Mandalay	48.689998626709
2	Yangon	49.4900016784668
3	Naypyitaw	49.6500015258789

Q.20] Identify the customer type with the highest VAT payments.

```sql
--20.Identify the customer type with the highest VAT payments.
select customer_type,max(vat) as highest_vat from amazon_sales
group by customer_type
order  by highest_vat desc
```

150 %

Results | Messages

	customer_type	highest_vat
1	Member	49.6500015258789
2	Normal	49.4900016784668

Q.21] What is the count of distinct payment methods in the dataset?

```
--20.What is the count of distinct payment methods in the dataset?
select count(distinct payment_method) as count_distinct_payment_methods from amazon_sales
```

150 %

Results | Messages

	count_distinct_payment_methods
1	3

Q.22] Examine distribution of gender within each branch.

```
--22.Examine the distribution of genders within each branch.
select  branch,gender,count(*) as gender_count from amazon_sales
group by branch ,gender
order by branch,gender
```

150 %

Results | Messages

	branch	gender	gender_count
1	A	Female	161
2	A	Male	179
3	B	Female	162
4	B	Male	170
5	C	Female	178
6	C	Male	150

Q.23] Determine predominant gender among customer.

```sql
--23.Determine the predominant gender among customers.
select gender,count(*)  as count from amazon_sales
group by gender
order by count(*) desc
```

150 %

Results | Messages

	gender	count
1	Female	501
2	Male	499

Q.24] Identify the day of the week with the highest average ratings.

```sql
--24.Identify the day of the week with the highest average ratings.
select day_name ,avg(rating) as avg_rating from amazon_sales
group by day_name
order by avg_rating desc
```

150 %

Results | Messages

	day_name	avg_rating
1	Monday	7.153600
2	Friday	7.076258
3	Sunday	7.011278
4	Tuesday	7.003164
5	Saturday	6.901829
6	Thursday	6.889855
7	Wednesday	6.805594

Q.25] Identify the time of day when customer provide most ratings.

```sql
--25.Identify the time of day when customers provide the most ratings.
select time_of_day ,count(rating) as rating_count from amazon_sales
group by time_of_day
order by rating_count desc
```

150 %

Results | Messages

	time_of_day	rating_count
1	Afternoon	528
2	Evening	281
3	Morning	191

Q.26] Determine the time of day with the highest customer ratings for each branch.

```sql
--26.Determine the time of day with the highest customer ratings for each branch.
with cte as(select branch,
                   max(rating) as highest_rating
                   from amazon_sales
            group by branch)

select time_of_day,branch,max(rating) as rating_count from amazon_sales
group by time_of_day,branch
having max(rating)=(select highest_rating from cte where cte.branch=amazon_sales.branch)
order by branch
```

150 %

Results | Messages

	time_of_day	branch	rating_count
1	Afternoon	A	10.0
2	Afternoon	B	10.0
3	Evening	B	10.0
4	Morning	B	10.0
5	Afternoon	C	10.0

Q.27]. Determine the day of the week with the highest average ratings for each branch.

```sql
--28.Determine the day of the week with the highest average ratings for each branch.
with cte as(
            select branch,day_name,avg(rating) as avg_rating from amazon_sales
            group by branch,day_name),

max_r as(select max(avg_rating) as avg_rat from cte
group by branch)

select branch,day_name,avg_rating from cte
where avg_rating in (select * from max_r)
order by branch
```

150 %

Results | Messages

	branch	day_name	avg_rating
1	A	Friday	7.312000
2	B	Monday	7.335897
3	C	Friday	7.278947

Key Findings

Product Analysis:

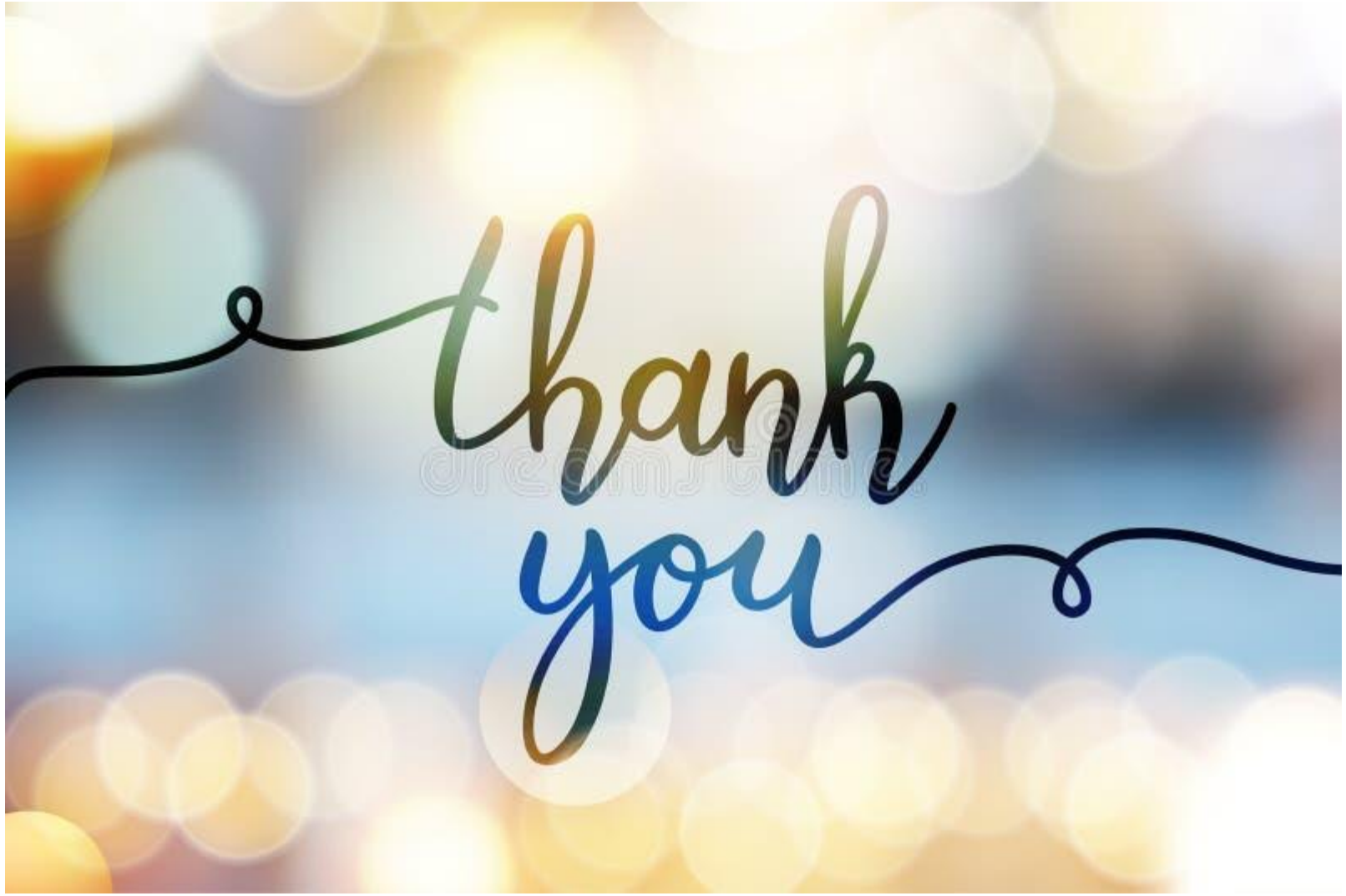
- Highest Sales Product Line: **Electronic Accessories (Units Sold:971)**
- Highest Revenue Product Line: **Food and Beverages ($ 56144.96)**
- Lowest Sales Product Line: **Health and Beauty (Unit Sold: 854)**
- Lowest Revenue Product Line: **Health and Beauty ($ 49193.84)**

Sales Analysis:

- Month With Highest Revenue: **January ($ 116292.11)**
- City & Branch With Highest Revenue: **Naypyitaw[C] ($ 110568.86)**
- Month With Lowest Revenue: **February ($ 97219.58)**
- City & Branch With Lowest Revenue: **Mandalay[B] ($ 106198.00)**
- Peak Sales Time Of Day: **Afternoon**

Customer Analysis:

- Most Predominant Gender: **Female**
- Most Predominant Customer Type: **Member**
- Highest Revenue Gender: **Female ($ 167883.26)**
- Highest Revenue Customer Type: **Member ($ 164223.81)**
- Most Popular Product Line (Male): **Health and Beauty**
- Most Popular Product Line (Female): **Fashion Accessories**
- Distribution Of Members Based On Gender: **Male(240) Female(261)**
- Sales Male: **2641 units**
- Sales Female: **2869 units**


thank
you