# Amazon Sales Data~

# SQL Project Presentation

~S8229

(Vaibhav Adakane)

# Overview 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

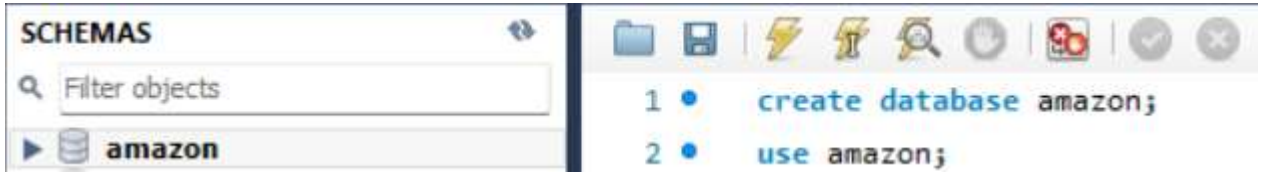
- The major aim of this project is to gain insight into the sales data of Amazon to understand the different factors that affect sales of the different branches

# Preview of Amazon Sales Data

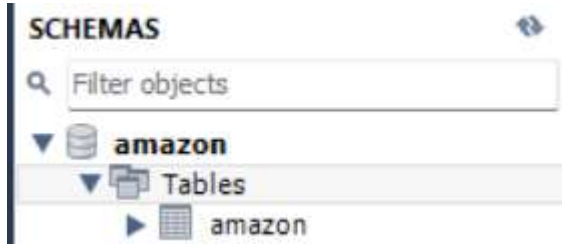
| Column | Description | Data Type |
|---|---|---|
| Invoice Id | Invoice of the sales made | Varchar(30) |
| Branch | Branch at which sales were made | Varchar(5) |
| City | The location of the branch | Varchar(30) |
| Customer Type | The type of the customer | Varchar(30) |
| Gender | Gender of the 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) |

# Data Wrangling

Step [1]: Created a database named Amazon in MySQL.




Step [2]: Importing data in the form of a demo table named Amazon using table data import wizard.


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

| Invoice ID | Branch | City | Customer type | Gender | Product line | Unit price | Quantity | Tax 5% | Total | Date | Time | Payment | cogs | gross margin percentage | gross income | Rating |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 750-67-8428 | A | Yangon | Member | Female | Health and beauty | 74.69 | 7 | 26.1415 | 548.9715 | 2019-01-05 | 13:08:00 | Ewallet | 522.83 | 4.761904762 | 26.1415 | 9.1 |
| 226-31-3081 | C | Naypyitaw | Normal | Female | Electronic accessories | 15.28 | 5 | 3.82 | 80.22 | 2019-03-08 | 10:29:00 | Cash | 76.4 | 4.761904762 | 3.82 | 9.6 |
| 631-41-3108 | A | Yangon | Normal | Male | Home and lifestyle | 46.33 | 7 | 16.2155 | 340.5255 | 2019-03-03 | 13:23:00 | Credit card | 324.31 | 4.761904762 | 16.2155 | 7.4 |
| 123-19-1176 | A | Yangon | Member | Male | Health and beauty | 58.22 | 8 | 23.288 | 489.048 | 2019-01-27 | 20:33:00 | Ewallet | 465.76 | 4.761904762 | 23.288 | 8.4 |
| 373-73-7910 | A | Yangon | Normal | Male | Sports and travel | 86.31 | 7 | 30.2085 | 634.3785 | 2019-02-08 | 10:37:00 | Ewallet | 604.17 | 4.761904762 | 30.2085 | 5.3 |
| 699-14-3026 | C | Naypyitaw | Normal | Male | Electronic accessories | 85.39 | 7 | 29.8865 | 627.6165 | 2019-03-25 | 18:30:00 | Ewallet | 597.73 | 4.761904762 | 29.8865 | 4.1 |
| 355-53-5943 | A | Yangon | Member | Female | Electronic accessories | 68.84 | 6 | 20.652 | 433.692 | 2019-02-25 | 14:36:00 | Ewallet | 413.04 | 4.761904762 | 20.652 | 5.8 |

## Step [3]: 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.

```
7 •  describe amazon;
```

Result Grid | Filter Rows: | Export:

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| Invoice ID | text | YES | | NULL | |
| Branch | text | YES | | NULL | |
| City | text | YES | | NULL | |
| Customer type | text | YES | | NULL | |
| Gender | text | YES | | NULL | |
| Product line | text | YES | | NULL | |
| Unit price | double | YES | | NULL | |
| Quantity | int | YES | | NULL | |
| Tax 5% | double | YES | | NULL | |
| Total | double | YES | | NULL | |
| Date | text | YES | | NULL | |
| Time | text | YES | | NULL | |
| Payment | text | YES | | NULL | |
| cogs | double | YES | | NULL | |
| gross margin ... | double | YES | | NULL | |
| gross income | double | YES | | NULL | |
| Rating | double | YES | | NULL | |

```
9 •  select count(*) as count_of_null_values from amazon
10   where null;
```

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

| count_of_null_values |
|---|
| 0 |

# 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).

```
81 •  select `invoice id`, date, time, time_of_day, day_name, month_name from amazon
82    limit 5
```

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

| invoice id | date | time | time_of_day | day_name | month_name |
|---|---|---|---|---|---|
| 750-67-8428 | 2019-01-05 | 13:08:00 | Afternoon | Saturday | January |
| 226-31-3081 | 2019-03-08 | 10:29:00 | Morning | Friday | March |
| 631-41-3108 | 2019-03-03 | 13:23:00 | Afternoon | Sunday | March |
| 123-19-1176 | 2019-01-27 | 20:33:00 | Evening | Sunday | January |
| 373-73-7910 | 2019-02-08 | 10:37:00 | Morning | Friday | February |

```
46 •  alter table amazon
47    add time_of_day varchar(15) not null;
48
49 •  update amazon set time_of_day =
50    case
51        when hour(time) between 06 and 11 then 'Morning'
52        when hour(time) between 12 and 17 then 'Afternoon'
53        else 'Evening'
54    end;
55
56 •  alter table amazon
57    add day_name varchar(10) not null;
58
59 •  update amazon set day_name =
60    (select  dayname(date));
61
62 •  alter table amazon
63    add month_name varchar(10) not null;
64
65 •  update amazon set month_name =
66    (select monthname(date));
```

# 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.

```
17 •  create table amazon_sales
18    (invoice_id varchar(30) primary key not null,
19    branch varchar(5) not null,
20    city varchar(30) not null,
21    customer_type varchar(30) not null,
22    gender varchar(10) not null,
23    product_line varchar(100) not null,
24    unit_price decimal(10,2) not null,
25    quantity int not null,
26    vat float not null,
27    total decimal(10,2) not null,
28    date date not null,
29    time time not null,
30    payment_method varchar(20) not null,
31    cogs decimal(10,2) not null,
32    gross_margin_percentage float not null,
33    gross_income decimal(10,2) not null,
34    rating decimal(3,1) not null,
35    time_of_day varchar(15) not null,
36    day_name varchar(10) not null,
37    month_name varchar(10) not null);
38
39 •  insert into amazon_sales
40    (select * from amazon);
```

```
42 •  describe amazon_sales;
```

Result Grid | Filter Rows: | Export:

| Field | Type | Null | Key | Default |
|---|---|---|---|---|
| invoice_id | varchar(30) | NO | PRI | NULL |
| branch | varchar(5) | NO | | NULL |
| city | varchar(30) | NO | | NULL |
| customer_type | varchar(30) | NO | | NULL |
| gender | varchar(10) | NO | | NULL |
| product_line | varchar(100) | NO | | NULL |
| unit_price | decimal(10,2) | NO | | NULL |
| quantity | int | NO | | NULL |
| vat | float | NO | | NULL |
| total | decimal(10,2) | NO | | NULL |
| date | date | NO | | NULL |
| time | time | NO | | NULL |
| payment_met... | varchar(20) | NO | | NULL |
| cogs | decimal(10,2) | NO | | NULL |
| gross_margin... | float | NO | | NULL |
| gross_income | decimal(10,2) | NO | | NULL |
| rating | decimal(3,1) | NO | | NULL |
| time_of_day | varchar(15) | NO | | NULL |
| day_name | varchar(10) | NO | | NULL |
| month_name | varchar(10) | NO | | NULL |

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

```
71 •  select count(*) as total_columns from information_schema.columns
72    where table_name ='amazon_sales';
```

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

| total_columns |
|---|
| 20 |

```
75 •  select count(*) as total_rows from amazon_sales;
```

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

| total_rows |
|---|
| 1000 |

```
77 •  select count(*) as null_values from amazon_sales where null;
```

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

| null_values |
|---|
| 0 |

```
86 •  select * from unique_values;
```

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

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1000 | 3 | 3 | 2 | 2 | 6 | 943 | 10 | 990 | 990 | 89 | 506 | 3 | 990 | 1 | 873 | 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?

```
5 •  select count(distinct(city)) from amazon_sales;
```

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

| count(distinct(city)) |
|---|
| 3 |

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

```
5 •  select distinct city, branch from amazon_sales;
```

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

| city | branch |
|---|---|
| Yangon | A |
| Naypyitaw | C |
| Mandalay | B |

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

```
8 •  select count(distinct(product_line)) from amazon_sales;
```

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

| count(distinct(product_line)) |
|---|
| 6 |

## Q.4] Which payment method occurs most frequently?

```
11 •  select payment_method, count(*) as occurance from amazon_sales
12    group by payment_method
13    order by occurance desc;
```

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

| payment_method | occurance |
|---|---|
| Ewallet | 345 |
| Cash | 344 |
| Credit card | 311 |

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

```
16 •  select product_line, sum(quantity) as total_sales from amazon_sales
17    group by product_line
18    order by total_sales desc;
```

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

| product_line | total_sales |
|---|---|
| Electronic accessories | 971 |
| Food and beverages | 952 |
| Sports and travel | 920 |
| Home and lifestyle | 911 |
| Fashion accessories | 902 |
| Health and beauty | 854 |

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

```
21 •  select month_name, sum(total) as monthly_revenue$ from amazon_sales
22    group by month_name
23    order by monthly_revenue$ desc;
```

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

| month_name | monthly_revenue$ |
|---|---|
| January | 116292.11 |
| March | 109455.74 |
| February | 97219.58 |

Q.7] Which product line generated highest revenue?

```
31 •  select product_line, sum(total) as total_revenue$ from amazon_sales
32    group by product_line
33    order by total_revenue$ desc;
```

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

| product_line | total_revenue$ |
|---|---|
| Food and beverages | 56144.96 |
| Sports and travel | 55123.00 |
| Electronic accessories | 54337.64 |
| Fashion accessories | 54306.03 |
| Home and lifestyle | 53861.96 |
| Health and beauty | 49193.84 |

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

```
26 •  select month_name, sum(cogs) as cost_of_goods_sold from amazon_sales
27    group by month_name
28    order by cost_of_goods_sold desc;
```

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

| month_name | cost_of_goods_sold |
|---|---|
| January | 110754.16 |
| March | 104243.34 |
| February | 92589.88 |

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

```
36 •  select city, sum(total) as revenue$ from amazon_sales
37    group by city
38    order by revenue$ desc;
```

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

| city | revenue$ |
|---|---|
| Naypyitaw | 110568.86 |
| Yangon | 106200.57 |
| Mandalay | 106198.00 |

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

```
41 •  select product_line, max(vat) highest_vat  from amazon_sales
42    group by product_line
43    order by highest_vat desc;
```

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

| product_line | highest_vat |
|---|---|
| Fashion accessories | 49.65 |
| Food and beverages | 49.26 |
| Home and lifestyle | 48.75 |
| Sports and travel | 47.72 |
| Health and beauty | 45.25 |
| Electronic accessories | 44.8785 |

Q.11] Which customer type occurs most frequently?

```
102 •  select customer_type, count(*) as count from amazon_sales
103    group by customer_type
104    order by count desc;
```

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

| customer_type | count |
|---|---|
| Member | 501 |
| Normal | 499 |

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

```
46 •  select product_line, sum(total) as revenue,
47    case
48        when sum(total) > (select sum(total)/count(distinct(product_line)) from amazon_sales) then 'Good'
49        else 'Bad'
50    end performance
51    from amazon_sales
52    group by product_line;
```

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

| product_line | revenue | performance |
|---|---|---|
| Food and beverages | 56144.96 | Good |
| Health and beauty | 49193.84 | Bad |
| Sports and travel | 55123.00 | Good |
| Fashion accessories | 54306.03 | Good |
| Home and lifestyle | 53861.96 | Good |
| Electronic accessories | 54337.64 | Good |

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

```
55 •  select branch, sum(quantity) as product_sold from amazon_sales
56    group by branch
57    having product_sold > (select sum(quantity)/count(distinct branch) as avg_quantity from amazon_sales);
```

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

| branch | product_sold |
|---|---|
| A | 1859 |

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

```
60 •  with new as
61    (select gender, product_line, count(*) as count from amazon_sales
62    group by gender, product_line),
63
64    max_count as
65    (select max(count) from new group by gender)
66
67    select * from new
68    where count in (select * from max_count) limit 2;
```

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

| gender | product_line | count |
|---|---|---|
| Male | Health and beauty | 88 |
| Female | Fashion accessories | 96 |

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

```
96 •  select count(distinct(customer_type)) as count_distinct_customer_type from amazon_sales;
```

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

| count_distinct_customer_type |
|---|
| 2 |

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

```
71 •  select product_line, avg(rating) as avg_rating from amazon_sales
72    group by product_line;
```

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

| product_line | avg_rating |
|---|---|
| Food and beverages | 7.11322 |
| Health and beauty | 7.00329 |
| Sports and travel | 6.91627 |
| Fashion accessories | 7.02921 |
| Home and lifestyle | 6.83750 |
| Electronic accessories | 6.92471 |

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

```
81 •  select customer_type, sum(total) as revenue from amazon_sales
82    group by customer_type
83    order by revenue desc;
```

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

| customer_type | revenue |
|---|---|
| Member | 164223.81 |
| Normal | 158743.62 |

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

```
75 •    select day_name, time_of_day, count(*) sales from amazon_sales
76      group by day_name, time_of_day
77      order by field(day_name, 'Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'),
78      field(time_of_day, 'Morning', 'Afternoon', 'Evening');
```

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

| day_name | time_of_day | sales |
|---|---|---|
| Sunday | Morning | 22 |
| Sunday | Afternoon | 70 |
| Sunday | Evening | 41 |
| Monday | Morning | 21 |
| Monday | Afternoon | 75 |
| Monday | Evening | 29 |
| Tuesday | Morning | 36 |
| Tuesday | Afternoon | 71 |
| Tuesday | Evening | 51 |
| Wednesday | Morning | 22 |
| Wednesday | Afternoon | 81 |
| Wednesday | Evening | 40 |
| Thursday | Morning | 33 |
| Thursday | Afternoon | 76 |
| Thursday | Evening | 29 |
| Friday | Morning | 29 |
| Friday | Afternoon | 74 |
| Friday | Evening | 36 |
| Saturday | Morning | 28 |
| Saturday | Afternoon | 81 |
| Saturday | Evening | 55 |

Q.19] Determine city with highest VAT percentage.

```
86 •  select city, max(vat) as vat_percentage from amazon_sales
87    group by city
88    order by vat_percentage desc;
```

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

| city | vat_percentage |
|---|---|
| Naypyitaw | 49.65 |
| Yangon | 49.49 |
| Mandalay | 48.69 |

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

```
91 •  select customer_type, max(vat) as vat_percentage from amazon_sales
92    group by customer_type
93    order by vat_percentage desc;
```

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

| customer_type | vat_percentage |
|---|---|
| Member | 49.65 |
| Normal | 49.49 |

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

```
99 •  select count(distinct(payment_method)) as count_distinct_payment from amazon_sales;
```

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

| count_distinct_payment |
|---|
| 3 |

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

```
117 •  select branch, gender, count(*) as count from amazon_sales
118    group by branch, gender
119    order by branch, gender;
```

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

| branch | gender | count |
|---|---|---|
| A | Female | 161 |
| A | Male | 179 |
| B | Female | 162 |
| B | Male | 170 |
| C | Female | 178 |
| C | Male | 150 |

Q.23] Determine predominant gender among customer.

```
112 •   select gender, count(*) as count from amazon_sales
113     group by gender
114     order by count desc;
```

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

| gender | count |
|---|---|
| Female | 501 |
| Male | 499 |

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

```
134 •   select day_name, avg(rating) as avg_rating from amazon_sales
135     group by day_name
136     order by avg_rating desc;
```

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

| day_name | avg_rating |
|---|---|
| Monday | 7.15360 |
| Friday | 7.07626 |
| Sunday | 7.01128 |
| Tuesday | 7.00316 |
| Saturday | 6.90183 |
| Thursday | 6.88986 |
| Wednesday | 6.80559 |

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

```
122 •  select time_of_day, count(rating) as rating_count from amazon_sales
123    group by time_of_day
124    order by rating_count desc;
```

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

| time_of_day | rating_count |
|---|---|
| Afternoon | 528 |
| Evening | 281 |
| Morning | 191 |

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

```
127 •  select branch, time_of_day, max(rating) highest_rating from amazon_sales
128    group by branch, time_of_day
129    having highest_rating = (select max(x.max) from (select branch, time_of_day, max(rating) max from amazon_sales
130    group by branch, time_of_day) as x where x.branch= amazon_sales.branch)
131    order by branch;
```

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

| branch | time_of_day | highest_rating |
|---|---|---|
| A | Afternoon | 10.0 |
| B | Afternoon | 10.0 |
| B | Evening | 10.0 |
| B | Morning | 10.0 |
| C | Afternoon | 10.0 |

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

```
139 •   with avg_rating as
140     (select branch, day_name, avg(rating) avg_rat from amazon_sales
141     group by branch, day_name),
142
143     max_rating as
144     (select max(avg_rat) from avg_rating group by branch)
145
146     select branch, day_name, avg_rat as highest_avg_rat from avg_rating where avg_rat in (select * from max_rating);
```

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

| branch | day_name | highest_avg_rat |
|---|---|---|
| A | Friday | 7.31200 |
| B | Monday | 7.33590 |
| C | Friday | 7.27895 |

# 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**
- Peak Sales Day Of Week: **Saturday**

## 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