# SUPER MARKET ANALYSIS USING SQL WITH EDA

## EDA ON SUPER MARKET ANALYSIS USING SQL

**STEP 1: -** Firstly, we use MYSQL.

**STEP 2: -** We will use our database.

**STEP 3: -** We will import our database.

**STEP 4: -** We will analyze our dataset.

**STEP 5: -** We will first study all the contents of the dataset.

# FIELDS OF DATASETS:

**Invoice ID:** Computer generated sales slip - invoice identification number.

**Branch:** Branch of supermarket - 3 branches are available identifies by A,B,C.

**City:** Location of supermarket.

**Customer Type:** Type of customer
-- member - The customer using membership card
-- Normal - the customer without membership card.

**Gender:** Gender type of customer.

**Product line:** General item categorization group - electronics, accessories, health & beauty, home & lifestyle, fashion accessories, sports & travel, food & beverages.

**Unit Price:** price of product.

**Quantity:** No.Of product purchased by customers.

**Tax %:** 5% tax fees for customer buying.

**Total:** Total price including tax.

**Date:** Date of purchase (record available from Jan 2019 to mar 2019).

**Time:** purchase time 10 am - 9 pm.

**Payment:** Payment used by customer for purchase - 3 payment method available - cash,Ewallet,credit card.

**cogs:** cost of goods sold.

**gross margin percentage:** gross margin %.

**gross income:** income.

**Rating:** customer satisfaction rating on their overall shopping experience - on a scale1 to 10.

# After the analysis we will solve some queries related to the dataset.

### -- 1. Display the first 5 rows from the dataset.

**Query:**

```
SELECT * FROM supermarket
LIMIT 5;
```

**Output:**

| 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 | 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 | 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 | 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 | 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 | Ewallet | 604.17 | 4.761904762 | 30.2085 | 5.3 |

### -- 2. Display the last 5 rows from the dataset.

**Query:**

```
SELECT  * FROM
 supermarket
ORDER BY 'InvoiceID' DESC
LIMIT 5;
```

**Output:**

| 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 | 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 | 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 | 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 | Ewallet | 465.76 | 4.761904762 | 23.288 | 8.4 |
| 373-73-7910 | A | Yangon | Normal | Male | orts and travel | 86.31 | 7 | 30.2085 | 634.3785 | 2019-02-08 | 10:37 | Ewallet | 604.17 | 4.761904762 | 30.2085 | 5.3 |

### -- 3. Display random 5 rows from the dataset.

**Query:**

```
SELECT  * FROM
supermarket
ORDER BY RAND()
LIMIT 5;
```

**Output:**

| 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 | 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 | 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 | 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 | 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 | Ewallet | 604.17 | 4.761904762 | 30.2085 | 5.3 |

**-- 4. Display count, min, max, avg, and std values for a column(gross income) in the dataset.**

**Query:**

```
SELECT
    COUNT(`gross income`),
    MIN(`gross income`),
    MAX(`gross income`),
    AVG(`gross income`),
    STD(`gross income`) FROM supermarket;
```

**Output:**

| COUNT(`gross income`) | MIN(`gross income`) | MAX(`gross income`) | AVG(`gross income`) | STD(`gross income`) |
|---|---|---|---|---|
| 1000 | 0.5085 | 49.65 | 15.379369000000002 | 11.702969603922716 |

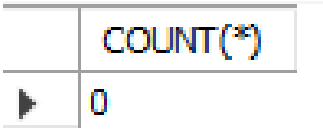
**-- 5. Find the number of missing values.**

**Query:**

```
SELECT COUNT(*)
FROM
    supermarket
```

```
WHERE
    branch IS NULL;
```

**Output:**



| | COUNT(*) |
|---|---|
| ▶ | 0 |

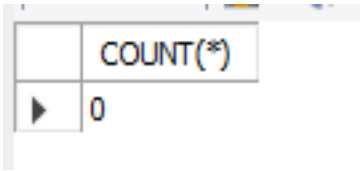
**-- there are no missing values in dataset as dataset is cleaned dataset**

**-- 6. Count the number of occurrences of each city.**

**Query:**

```
SELECT
    city, COUNT(*)
FROM
    supermarket
 group by city;
```

**Output:**



| | COUNT(*) |
|---|---|
| ▶ | 0 |

# Visualization:


**-- 7. Find the most frequently used payment method.**

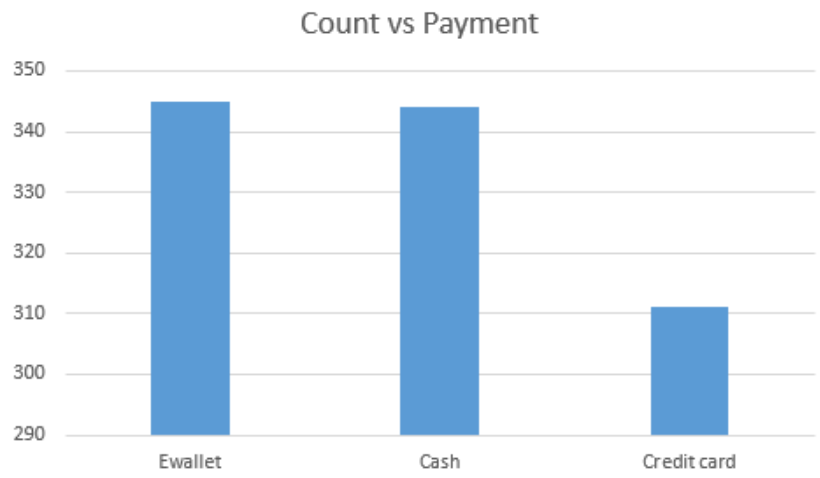
**Query:**

```
SELECT payment, COUNT(Payment)
FROM
    supermarket
GROUP BY Payment
ORDER BY COUNT(Payment) DESC;
```

**Output:**

| payment | count(Payment) |
|---|---|
| Ewallet | 345 |
| Cash | 344 |
| Credit card | 311 |

# Visualization:




**-- 8. Does The Cost of Goods Sold Affect The Ratings That The Customers Provide.**

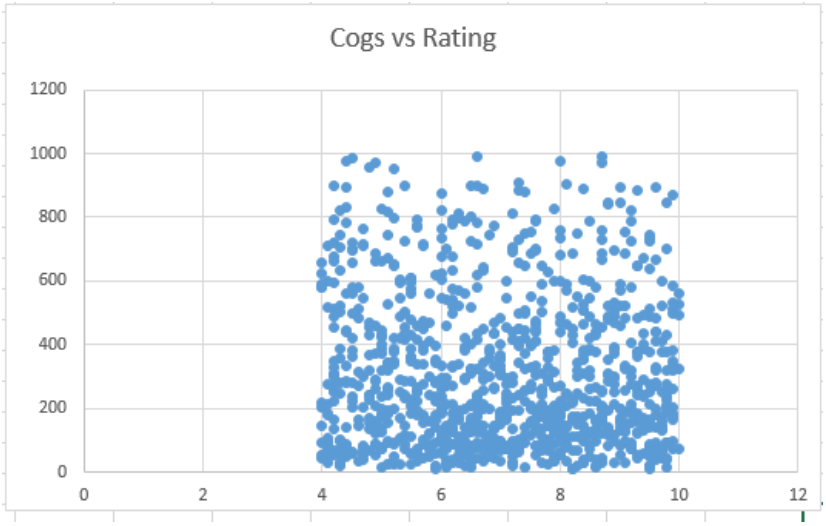
**Query:**

```
SELECT
    rating, cogs
FROM
    supermarket;
```

**Output:**

| rating | cogs |
|---|---|
| 9.1 | 522.83 |
| 9.6 | 76.4 |
| 7.4 | 324.31 |
| 8.4 | 465.76 |
| 5.3 | 604.17 |
| 4.1 | 597.73 |
| 5.8 | 413.04 |
| 8 | 735.6 |

# Visualization:



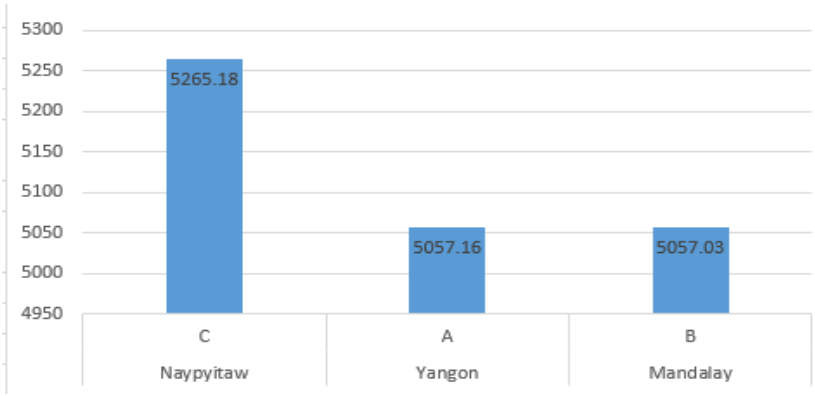

**-- 9. Find the most profitable branch as per gross income.**
**Query:**

```
SELECT
    city,
    branch,
    ROUND(SUM(`gross income`), 2) AS sum_gross_income
FROM
    supermarket
GROUP BY branch , city
ORDER BY sum_gross_income DESC;
```

**Output:**

| city | branch | sum_gross_income |
|---|---|---|
| Naypyitaw | C | 5265.18 |
| Yangon | A | 5057.16 |
| Mandalay | B | 5057.03 |

# Visualization:




**-- 10. Find the most used payment method city-wise.**

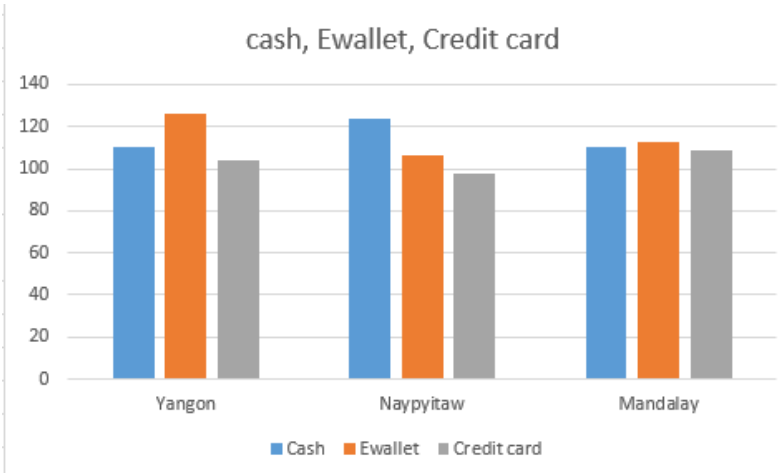
**Query:**

```
SELECT city,
    SUM(CASE WHEN Payment = 'cash' THEN 1 ELSE 0 END) AS 'cash',
    SUM(CASE WHEN Payment = 'Ewallet' THEN 1 ELSE 0 END) AS 'Ewallet',
    SUM(CASE WHEN Payment = 'credit card' THEN 1 ELSE 0 END) AS 'creadit card'
FROM
    supermarket
GROUP BY city;
```

**Output:**

| city | cash | Ewallet | creadit card |
|---|---|---|---|
| Yangon | 110 | 126 | 104 |
| Naypyitaw | 124 | 106 | 98 |
| Mandalay | 110 | 113 | 109 |

## Visualization:




**-- 11. Find the product line purchased in the highest quantity.**
**Query:**

```
SELECT
    `Product line`, SUM(quantity) AS quantity
FROM
    supermarket
GROUP BY `Product line`
ORDER BY quantity DESC;
```

**Output:**

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

# Visualization:


**-- 12. Display the daily sales by day of the week.**

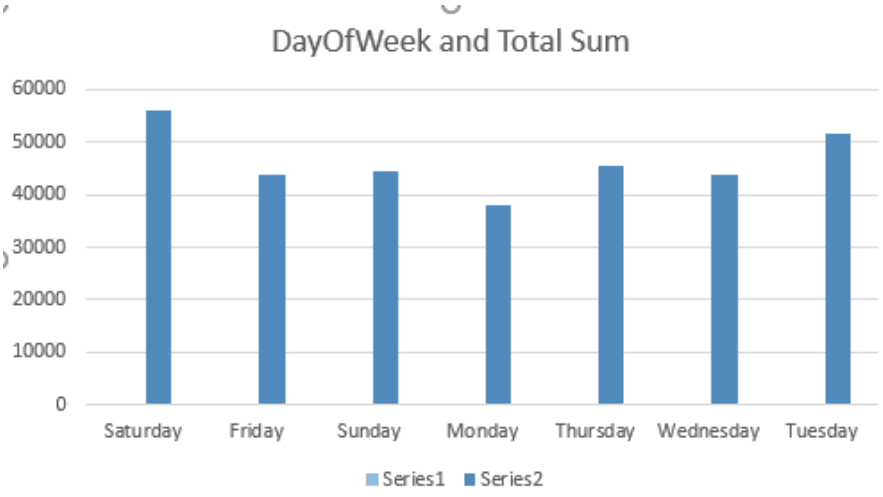
**Query:**

```
SELECT
    DAYNAME(Date), DAYOFWEEK(Date), ROUND(SUM(Total), 2)
FROM
    supermarket
GROUP BY DAYNAME(Date) , DAYOFWEEK(Date);
```

**Output:**

| DAYNAME(Date) | DAYOFWEEK(Date) | ROUND(SUM(Total), 2) |
|---|---|---|
| Saturday | 7 | 56120.81 |
| Friday | 6 | 43926.34 |
| Sunday | 1 | 44457.89 |
| Monday | 2 | 37899.08 |
| Thursday | 5 | 45349.25 |
| Wednesday | 4 | 43731.14 |
| Tuesday | 3 | 51482.25 |

# Visualization:




**-- 13. Find the month with the highest sales.**

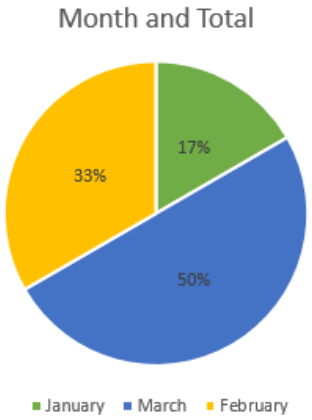
**Query:**

```
SELECT
    MONTHNAME(Date) AS name,
    MONTH(Date) AS month,
    SUM(total) AS total
FROM
    supermarket
GROUP BY name , month
ORDER BY total DESC;
```

**Output:**

| name | month | total |
|---|---|---|
| January | 1 | 116291.86800000005 |
| March | 3 | 109455.50700000004 |
| February | 2 | 97219.37399999997 |

## Visualization:



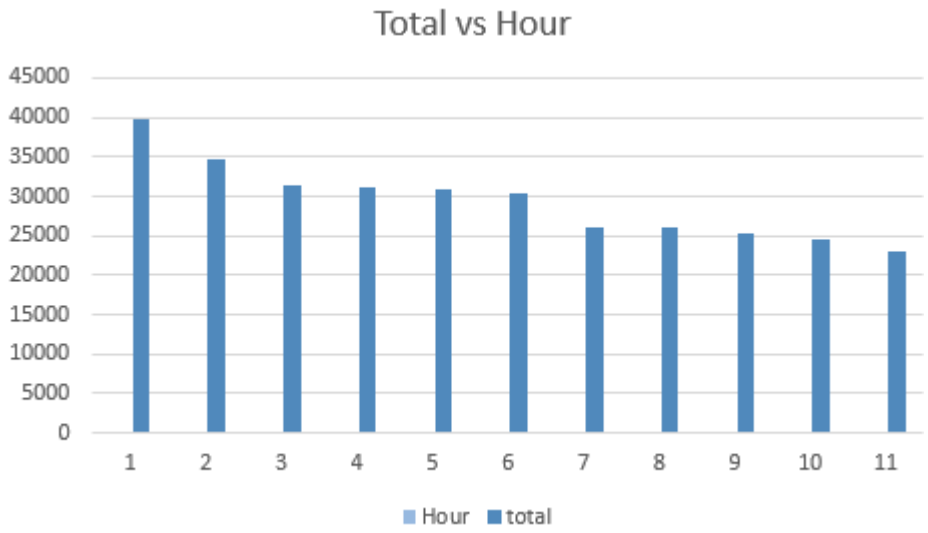

**-- 14. Find the time at which sales are highest.**
**Query:**

```
SELECT
    HOUR(time) AS hour, ROUND(SUM(total), 2) AS total
FROM
    supermarket
GROUP BY hour
ORDER BY total DESC;
```

**Output:**

| hour | total |
|---|---|
| 19 | 39699.51 |
| 13 | 34723.23 |
| 10 | 31421.48 |
| 15 | 31179.51 |
| 14 | 30828.4 |
| 11 | 30377.33 |
| 12 | 26065.88 |
| 18 | 26030.34 |
| 16 | 25226.32 |
| 17 | 24445.22 |
| 20 | 22969.53 |

## Visualization:




**-- 15. Which gender spends more on average**
**Query:**

```
SELECT
    gender, AVG(`gross income`)
FROM
    supermarket
GROUP BY gender;
```

**Output:**

| gender | AVG(`gross income`) |
|---|---|
| Female | 15.956936127744514 |
| Male | 14.7994869739479 |

## Visualization: