WALMART SALES EXPLORATORY DATA ANALYSIS

```sql
-- Exploratory Data Analysis
-- Viewing the table
SELECT * FROM walmart.sales;
```

invoice_id	branch	city	customer_type	gender	product_line	unit_price	quantity	tax_pct	total	date	time	payment	cogs	gross_margin_pct
101-17-6199	A	Yangon	Normal	Male	Food and beverages	46	7	16.0265	337	2019-03-13 00:00:00	19:44:00	Credit card	321	4.7619
101-81-4070	C	Naypyitaw	Member	Female	Health and beauty	63	2	6.282	132	2019-01-17 00:00:00	12:36:00	Ewallet	126	4.7619
102-06-2002	C	Naypyitaw	Member	Male	Sports and travel	25	5	6.3125	133	2019-03-20 00:00:00	17:52:00	Cash	126	4.7619
102-77-2261	C	Naypyitaw	Member	Male	Health and beauty	65	7	22.8585	480	2019-03-05 00:00:00	18:02:00	Credit card	457	4.7619
105-10-6182	A	Yangon	Member	Male	Fashion accessories	21	2	2.148	45	2019-02-27 00:00:00	12:22:00	Ewallet	43	4.7619
105-31-1824	A	Yangon	Member	Male	Sports and travel	70	7	24.332	511	2019-02-01 00:00:00	15:10:00	Credit card	487	4.7619
106-35-6779	A	Yangon	Member	Male	Home and lifestyle	44	2	4.434	93	2019-03-27 00:00:00	11:26:00	Cash	89	4.7619
109-28-2512	B	Mandalay	Member	Female	Fashion accessories	98	6	29.283	615	2019-01-07 00:00:00	15:01:00	Ewallet	586	4.7619
109-86-4363	B	Mandalay	Member	Female	Sports and travel	60	7	21.028	442	2019-02-14 00:00:00	11:36:00	Credit card	421	4.7619
110-05-6330	C	Naypyitaw	Normal	Female	Food and beverages	39	6	11.829	248	2019-03-25 00:00:00	20:18:00	Credit card	237	4.7619
110-48-7033	B	Mandalay	Member	Male	Fashion accessories	33	4	6.524	137	2019-01-29 00:00:00	14:12:00	Cash	130	4.7619

```sql
-- Feature Engineering
select time, (
case
    when time between '00:00:00' and '12:00:00' then 'Morning'
    when time between '12:01:00' and '16:00:00' then 'Afternoon'
    else 'Evening'
    end
) as time_of_day from sales;

alter table sales
add column time_of_day varchar(50);

update sales
```

duct_line	unit_price	quantity	tax_pct	total	date	time	payment	cogs	gross_margin_pct	gross_income	rating	time_of_day
l and beverages	46	7	16.0265	337	2019-03-13 00:00:00	19:44:00	Credit card	321	4.7619	16	7	Evening
th and beauty	63	2	6.282	132	2019-01-17 00:00:00	12:36:00	Ewallet	126	4.7619	6	4.9	Afternoon
ts and travel	25	5	6.3125	133	2019-03-20 00:00:00	17:52:00	Cash	126	4.7619	6	6.1	Evening
th and beauty	65	7	22.8585	480	2019-03-05 00:00:00	18:02:00	Credit card	457	4.7619	23	4.2	Evening

```sql
-- Add column 'day_name' --------------
----------------------------------------
SET SQL_SAFE_UPDATES = 0;
alter table sales
add column day_name varchar(50);

select date, dayname(DATE) from sales;

update sales
set day_name = (dayname(date));

-- Add column 'month_name' --------------
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

duct_line	unit_price	quantity	tax_pct	total	date	time	payment	cogs	gross_margin_pct	gross_income	rating	time_of_day	day_name
l and beverages	46	7	16.0265	337	2019-03-13 00:00:00	19:44:00	Credit card	321	4.7619	16	7	Evening	Wednesday
th and beauty	63	2	6.282	132	2019-01-17 00:00:00	12:36:00	Ewallet	126	4.7619	6	4.9	Afternoon	Thursday
ts and travel	25	5	6.3125	133	2019-03-20 00:00:00	17:52:00	Cash	126	4.7619	6	6.1	Evening	Wednesday
th and beauty	65	7	22.8585	480	2019-03-05 00:00:00	18:02:00	Credit card	457	4.7619	23	4.2	Evening	Tuesday

```sql
update sales
set day_name = (dayname(date));

-- Add column 'month_name' --------------
----------------------------------------

alter table sales
add column month_name varchar(50);

select date, monthname(date) from sales;

update sales
set month_name = monthname(date);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

duct_line	unit_price	quantity	tax_pct	total	date	time	payment	cogs	gross_margin_pct	gross_income	rating	time_of_day	day_name	month_name
l and beverages	46	7	16.0265	337	2019-03-13 00:00:00	19:44:00	Credit card	321	4.7619	16	7	Evening	Wednesday	March
th and beauty	63	2	6.282	132	2019-01-17 00:00:00	12:36:00	Ewallet	126	4.7619	6	4.9	Afternoon	Thursday	January
ts and travel	25	5	6.3125	133	2019-03-20 00:00:00	17:52:00	Cash	126	4.7619	6	6.1	Evening	Wednesday	March
th and beauty	65	7	22.8585	480	2019-03-05 00:00:00	18:02:00	Credit card	457	4.7619	23	4.2	Evening	Tuesday	March

```sql
-- What is the most selling product line ?

select product_line, count(product_line) as count from sales
group by product_line
order by count desc;
```

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

product_line	count
Fashion accessories	178
Food and beverages	174
Electronic accessories	170
Sports and travel	166
Home and lifestyle	160
Health and beauty	152

```sql
-- What product line had the largest revenue?

select product_line, sum(total) as sales from sales
group by product_line
order by sales desc
limit 1;
```

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

product_line	sales
Food and beverages	56153

```sql
-- >> Fetch each product line and add a column to those product line
-- showing "Good", "Bad". Good if its greater than average sales

select product_line,case
when avg_sale >= (select ROUND(AVG(total), 2) from sales) then 'Good'
else 'Bad'
end as performance
from (select product_line,round(avg(total), 2) as avg_sale
from sales
group by product_line
) as subquery;
```

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

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

```sql
-- >> Which branch sold more products than average product sold?

WITH qnt1 AS (
    SELECT branch, SUM(quantity) as qnt
    FROM sales
    GROUP BY branch
)
SELECT branch, SUM(quantity) as qnt FROM sales
GROUP BY branch
HAVING SUM(quantity) > (SELECT AVG(qnt) FROM qnt1)
```

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

branch	qnt
A	1859

```sql
-- >> What is the most common product line by gender?

SELECT gender, product_line, count(gender) as count from sales
group by gender, product_line
order by gender desc;
```

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

gender	product_line	count
Male	Electronic accessories	86
Male	Fashion accessories	82
Male	Food and beverages	84
Male	Health and beauty	88
Male	Home and lifestyle	81
Male	Sports and travel	78
Female	Electronic accessories	84
Female	Fashion accessories	96
Female	Food and beverages	90
Female	Health and beauty	64
Female	Home and lifestyle	79
Female	Sports and travel	88

```sql
-- >> What is the average rating of each product line?

SELECT product_line, ROUND(AVG(rating),2) AS avg_rating FROM sales
GROUP BY product_line;
```

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

product_line	avg_rating
Food and beverages	7.11
Health and beauty	7
Sports and travel	6.92
Fashion accessories	7.03
Home and lifestyle	6.84
Electronic accessories	6.92

```sql
-- >> What is the total revenue by month ?

select month_name, sum(total) as sales from sales
group by month_name;
```

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

month_name	sales
March	109463
January	116294
February	97213

```sql
-- >> Number of sales made in each time of the day per weekday ?

SELECT day_name, COUNT(invoice_id) as total_invoice, SUM(total) as sale FROM sales
GROUP BY day_name;
```

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

day_name	total_invoice	sale
Wednesday	143	43731
Thursday	138	45352
Tuesday	158	51486
Friday	139	43923
Monday	125	37905
Saturday	164	56118
Sunday	133	44455

```sql
-- >> What is the city with the largest revenue?

select city, sum(total) as sale from sales
group by city
order by sale desc;
```

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

city	sale
Naypyitaw	110569
Yangon	106202
Mandalay	106199

```sql
-- >> What is the most common customer type?
SELECT
    customer_type, count(customer_type) as count
from sales
group by customer_type
order by count desc;
```

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

customer_type	count
Member	501
Normal	499

```sql
SELECT
    customer_type, count(invoice_id) as count
from sales
group by customer_type
order by count desc;
```

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

customer_type	count
Member	501
Normal	499

```sql
-- >> What is the gender of most of the customers?
SELECT
    gender, COUNT(gender) as count
FROM
    sales
GROUP BY gender
ORDER BY count DESC;
```

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

gender	count
Female	501
Male	499

```sql
-- >> What is the gender distribution per branch?

SELECT branch,
       SUM(CASE WHEN gender = 'Male' THEN 1 ELSE 0 END) as male_count,
       SUM(CASE WHEN gender = 'Female' THEN 1 ELSE 0 END) as female_count
FROM sales
GROUP BY branch;
```

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

branch	male_count	female_count
A	179	161
C	150	178
B	170	162

```sql
-- >> Which of the customer types brings the most revenue?
SELECT
    customer_type, SUM(total) as total
FROM
sales
group by customer_type
order by total desc
LIMIT 1;
```

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

customer_type	total
Member	164230

```sql
-- >> Which time of the day do customers give most ratings per branch?
WITH rating_counts AS (
    SELECT branch, time_of_day, COUNT(rating) as count,
    RANK() OVER(PARTITION BY branch order by count(rating) desc) as ranking
    FROM sales
    GROUP BY branch, time_of_day)
select branch, time_of_day from rating_counts
where ranking = 1;
```

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

branch	time_of_day
A	Evening
B	Evening
C	Evening

```sql
-- >> Which day of the week has the best avg ratings?

SELECT day_name, ROUND(AVG(rating),2) as avg_rat
FROM sales
GROUP BY day_name
ORDER BY  avg_rat DESC;
```

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

day_name	avg_rat
Monday	7.15
Friday	7.08
Sunday	7.01
Tuesday	7
Saturday	6.9
Thursday	6.89
Wednesday	6.81

```sql
-- >> Which day of the week has the best average ratings per branch?
WITH avgdf as
(SELECT branch, day_name, ROUND(AVG(rating),2) as avg_rat,
RANK() OVER(PARTITION BY branch ORDER BY  ROUND(AVG(rating),2) DESC) AS avg_rank
FROM sales
GROUP BY branch, day_name)

SELECT branch, day_name, avg_rat from avgdf
WHERE avg_rank = 1;
```

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

branch	day_name	avg_rat
A	Friday	7.31
B	Monday	7.34
C	Friday	7.28