2. \*\*Feature Engineering:\*\*

CREATE DATABASE IF NOT EXISTS walmartSales;

CREATE TABLE IF NOT EXISTS sales(

invoice\_id VARCHAR(30) NOT NULL PRIMARY KEY,

branch VARCHAR(5) NOT NULL,

city VARCHAR(30) NOT NULL,

customer\_type VARCHAR(30) NOT NULL,

gender VARCHAR(30) NOT NULL,

product\_line VARCHAR(100) NOT NULL,

unit\_price DECIMAL(10,2) NOT NULL,

quantity INT NOT NULL,

tax\_pct FLOAT(6,4) NOT NULL,

total DECIMAL(12, 4) NOT NULL,

date DATETIME NOT NULL,

time TIME NOT NULL,

payment VARCHAR(15) NOT NULL,

cogs DECIMAL(10,2) NOT NULL,

gross\_margin\_pct FLOAT(11,9),

gross\_income DECIMAL(12, 4),

rating FLOAT(2, 1)

);

-- Data cleaning

SELECT

\*

FROM sales;

> 1. Add a new column named `time\_of\_day` to give insight of sales in the Morning, Afternoon and Evening. This will help answer the question on which part of the day most sales are made.

-- Add the time\_of\_day column

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(20);

set sql\_safe\_updates = 0;

> 2. Add a new column named `day\_name` that contains the extracted days of the week on which the given transaction took place (Mon, Tue, Wed, Thur, Fri). This will help answer the question on which week of the day each branch is busiest.

SELECT

date,

DAYNAME(date)

FROM sales;

ALTER TABLE sales ADD COLUMN day\_name VARCHAR(10);

UPDATE sales

SET day\_name = DAYNAME(date);

> 3. Add a new column named `month\_name` that contains the extracted months of the year on which the given transaction took place (Jan, Feb, Mar). Help determine which month of the year has the most sales and profit.

SELECT

date,

MONTHNAME(date)

FROM sales;

ALTER TABLE sales ADD COLUMN month\_name VARCHAR(10);

UPDATE sales

SET month\_name = MONTHNAME(date);

1. How many unique cities does the data have?

SELECT

DISTINCT city

FROM sales;

1. In which city is each branch?

SELECT

DISTINCT city,

branch

FROM sales;

###product

1. How many unique product lines does the data have?

SELECT

DISTINCT product\_line

FROM sales;

1. What is the most common payment method?
2. SELECT Payment, COUNT(\*) AS payment\_count
3. FROM sales
4. GROUP BY Payment
5. ORDER BY payment\_count DESC
6. LIMIT 1;
7. What is the most selling product line?

SELECT

SUM(quantity) as qty,

product\_line

FROM sales

GROUP BY product\_line

ORDER BY qty DESC;

1. What is the total revenue by month?

SELECT

month\_name AS month,

SUM(total) AS total\_revenue

FROM sales

GROUP BY month\_name

ORDER BY total\_revenue;

1. What month had the largest COGS?

SELECT

month\_name AS month,

SUM(cogs) AS cogs

FROM sales

GROUP BY month\_name

ORDER BY cogs;

1. What product line had the largest revenue?

SELECT

product\_line,

SUM(total) as total\_revenue

FROM sales

GROUP BY product\_line

ORDER BY total\_revenue DESC;

1. What is the city with the largest revenue?

SELECT

branch,

city,

SUM(total) AS total\_revenue

FROM sales

GROUP BY city, branch

ORDER BY total\_revenue;

1. What product line had the largest VAT?

SELECT

product\_line,

AVG(tax\_pct) as avg\_tax

FROM sales

GROUP BY product\_line

ORDER BY avg\_tax DESC;

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

SELECT

AVG(quantity) AS avg\_qnty

FROM sales;

SELECT

product\_line,

CASE

WHEN AVG(quantity) > 6 THEN "Good"

ELSE "Bad"

END AS remark

FROM sales

GROUP BY product\_line;

1. Which branch sold more products than average product sold?

SELECT

branch,

SUM(quantity) AS qnty

FROM sales

GROUP BY branch

HAVING SUM(quantity) > (SELECT AVG(quantity) FROM sales);

1. What is the most common product line by gender?

SELECT

gender,

product\_line,

COUNT(gender) AS total\_cnt

FROM sales

GROUP BY gender, product\_line

ORDER BY total\_cnt DESC;

1. What is the average rating of each product line?

SELECT

ROUND(AVG(rating), 2) as avg\_rating,

product\_line

FROM sales

GROUP BY product\_line

ORDER BY avg\_rating DESC;

**Sales**

1. Number of sales made in each time of the day per weekday

SELECT

time\_of\_day,

COUNT(\*) AS total\_sales

FROM sales

WHERE day\_name = "Sunday"

GROUP BY time\_of\_day

ORDER BY total\_sales DESC;v

1. Which of the customer types brings the most revenue?

SELECT

customer\_type,

SUM(total) AS total\_revenue

FROM sales

GROUP BY customer\_type

ORDER BY total\_revenue;

**customer**

1. How many unique customer types does the data have?

SELECT

DISTINCT customer\_type

FROM sales;

1. How many unique payment methods does the data have?

SELECT

DISTINCT payment

FROM sales;

1. What is the most common customer type?

SELECT

customer\_type,

count(\*) as count

FROM sales

GROUP BY customer\_type

ORDER BY count DESC;

1. Which customer type buys the most?

SELECT

customer\_type,

COUNT(\*)

FROM sales

GROUP BY customer\_type;

1. What is the gender of most of the customers?

SELECT

gender,

COUNT(\*) as gender\_cnt

FROM sales

GROUP BY gender

ORDER BY gender\_cnt DESC;

1. What is the gender distribution per branch?

SELECT

gender,

COUNT(\*) as gender\_cnt

FROM sales

WHERE branch = "C"

GROUP BY gender

ORDER BY gender\_cnt DESC;

1. Which time of the day do customers give most ratings?

SELECT

time\_of\_day,

AVG(rating) AS avg\_rating

FROM sales

GROUP BY time\_of\_day

ORDER BY avg\_rating DESC;

1. Which time of the day do customers give most ratings per branch?

SELECT

time\_of\_day,

AVG(rating) AS avg\_rating

FROM sales

WHERE branch = "A"

GROUP BY time\_of\_day

ORDER BY avg\_rating DESC;

1. Which day of the week has the best avg ratings?

SELECT

day\_name,

AVG(rating) AS avg\_rating

FROM sales

GROUP BY day\_name

ORDER BY avg\_rating DESC;

1. Which day of the week has the best average ratings per branch?

SELECT

day\_name,

COUNT(day\_name) total\_sales

FROM sales

WHERE branch = "C"

GROUP BY day\_name

ORDER BY total\_sales DESC;