

## Sales Performance Analysis Using SQL

### 0.1. Table of Contents

- Project Overview
  - Business Context
  - Analytical Strategy
  - Data Overview
- Advanced Data Cleaning
- Feature Engineering
- Business Metrics (KPIs)
- Branch-wise and City-wise Performance
- Customer Segmentation & Behavior
- Product Line Analysis
- Temporal Patterns
- Payment Method Analysis
- Profitability Analysis
- Correlations & Insights

## 1. Project Overview

I aim to help a retail chain optimize sales strategy by analyzing historical transactions using SQL. I'll extract meaningful KPIs and identify improvement opportunities across products, customers, time, and geography.

## 2. Business Context

A large multi-branch retailer wants to optimize:

- Inventory
- Promotions
- Store operations
- Customer engagement

Using SQL, I'll derive business-ready answers.

### 3. Analytical Strategy

My strategy focuses on:

- Cleaning and validating raw transactional data.
- Creating engineered features from temporal and customer data.
- Segmenting customers, products, and branches.
- Tracking key performance indicators (KPIs).
- Recommending data-driven changes.

### 4. Data Overview

Column	Description
Invoice ID	Unique transaction ID
Branch	Store branch (A/B/C)
City	Location
Customer type	Member / Normal
Gender	Male / Female
Product line	Product category
Unit price	Price per unit
Quantity	Items bought
Tax 5%	Tax on total price
Sales	Final price incl. tax
Date	Date of sale
Time	Time of sale
Payment	Cash, Credit, Ewallet
COGS	Cost of Goods Sold
Gross margin %	Usually fixed
Gross income	Net income from sale
Rating	Customer rating (1–10)

### 5. Advanced Data Cleaning

#### 5.1. Validate Uniqueness of Transactions

```

1 SELECT 'Invoice_ID', COUNT(*)
2 FROM 'supermarket_analysis'
3 GROUP BY 'Invoice_ID'
4 HAVING COUNT(*) > 1;

```

Invoice_ID	COUNT(*)
0	0

## 5.2. Check for Nulls or Invalid Entries

```

1 SELECT
2   SUM(CASE WHEN 'Invoice_ID' IS NULL THEN 1 ELSE 0 END) AS
   missing_invoice,
3   SUM(CASE WHEN Sales <= 0 THEN 1 ELSE 0 END) AS invalid_sales,
4   SUM(CASE WHEN Quantity <= 0 THEN 1 ELSE 0 END) AS invalid_qty
5 FROM 'supermarket analysis';

```

missing_invoice	invalid_sales	invalid_qty
0	0	0

## 5.3. Standardize Date & Time Formats

```

1 SELECT
2   STR_TO_DATE(Date, '%m/%d/%Y') AS clean_date,
3   STR_TO_DATE(Time, '%H:%i:%s') AS clean_time
4 FROM 'supermarket analysis';

```

clean_date	clean_time
2019-01-05	01:08:00
2019-03-08	10:29:00
2019-03-03	01:23:00
2019-01-27	08:33:00
2019-02-08	10:37:00
2019-03-25	06:30:00
2019-02-25	02:36:00
2019-02-24	11:38:00
2019-01-10	05:15:00
2019-02-06	01:27:00
2019-02-20	06:07:00
2019-03-09	05:10:00
2019-02-12	10:25:00
2019-02-07	04:48:00
2019-03-29	07:21:00
2019-01-15	04:18:00
2019-03-11	11:03:00
2019-01-01	10:39:00

# 6. Feature Engineering

## 6.1. Extract Temporal Features

```

1 SELECT
2   STR_TO_DATE(Date, '%d/%m/%Y') AS sale_date,
3   DAYOFWEEK(STR_TO_DATE(Date, '%d/%m/%Y')) AS day_of_week,
4   MONTH(STR_TO_DATE(Date, '%d/%m/%Y')) AS month,
5   HOUR(CAST(Time AS TIME)) AS sale_hour,
6   CASE
7     WHEN HOUR(CAST(Time AS TIME)) BETWEEN 0 AND 11 THEN 'Morning'

```

```

8      WHEN HOUR(CAST(Time AS TIME)) BETWEEN 12 AND 17 THEN 'Afternoon'
9      ELSE 'Evening'
10     END AS time_of_day
11 FROM 'supermarket analysis'
12 limit 3;

```

sale_date	day_of_week	month	sale_hour	time_of_day
2019-05-01	4	5	1	Morning
2019-08-03	7	8	10	Morning
2019-03-03	1	3	1	Morning

## 6.2. Compute Profit Margin

```

1 SELECT
2     Sales,
3     COGS,
4     'Gross income',
5     ROUND('Gross income' / COGS * 100, 2) AS profit_margin_pct
6 FROM 'supermarket analysis';

```

Sales	COGS	Gross income	profit_margin_pct
548.9715	522.83	26.1415	5
340.5255	324.31	16.2155	5

## 6.3. Weekend Flag

```

1 SELECT
2     Date,
3     CASE
4         WHEN DAYOFWEEK(STR_TO_DATE(Date, '%d/%m/%Y')) IN (1, 7) THEN '
Weekend'
5         ELSE 'Weekday'
6     END AS weekend_flag
7 FROM 'supermarket analysis'
8 limit 3;

```

Date	weekend_flag
1/5/2019	Weekday
3/8/2019	Weekend
3/3/2019	Weekend

## 7. Business Metrics (KPIs)

Metric	SQL Logic
Total Revenue	SUM(Sales)
Total Profit	SUM(Gross_income)
Average Basket Size	AVG(Quantity)

Avg Revenue per Transaction	AVG(Sales)
Repeat Customer Rate	Requires Customer IDs (not available)

## 8. Branch-wise and City-wise Performance

```

1 SELECT
2   Branch,
3   City,
4   SUM(Sales) AS total_sales,
5   AVG(Rating) AS avg_rating,
6   SUM('Gross income') AS total_profit
7 FROM 'supermarket analysis'
8 GROUP BY Branch, City
9 ORDER BY total_sales DESC;

```

Branch	City	total_sales	avg_rating	total_profit
Giza	Naypyitaw	110568.794	7.0268	5265.1765
Alex	Yangon	106200.375	7.0270	5057.1650
Cairo	Mandalay	106197.678	6.8180	5057.0320

## 9. Customer Segmentation & Behavior

### 9.1. By Customer Type

```

1 SELECT 'Customer type',
2        COUNT(*) AS transactions,
3        AVG(Sales) AS avg_sales,
4        AVG(Rating) AS avg_rating
5 FROM 'supermarket analysis'
6 GROUP BY 'Customer type';

```

Customer type	transactions	avg_sales	avg_rating
Member	565	335.7429	6.9172
Normal	435	306.3723	7.0485

### 9.2. By Gender

```

1 SELECT Gender,
2        SUM(Sales) AS total_spent,
3        AVG(Quantity) AS avg_quantity
4 FROM sales
5 GROUP BY Gender;

```

Gender	total_spent	avg_quantity
Female	194671.83750000005	5.7583
Male	128294.91150000007	5.1795

## 10. Product Line Analysis

### 10.1. Highest Revenue Products

```

1 SELECT 'Product line',
2        SUM(Sales) AS total_sales,
3        SUM('Gross income') AS profit,
4        AVG(Rating) AS avg_rating
5 FROM 'supermarket analysis'
6 GROUP BY 'Product line'
7 ORDER BY total_sales DESC;

```

Product line	total_sales	profit	avg_rating
Food and beverages	56144.88	2673.56	7.1132
Sports and travel	53728.82	2684.49	6.9156
Electronic accessories	51437.53	2587.51	6.7249
Fashion accessories	50485.39	2524.27	6.9105
Home and lifestyle	36861.91	2564.88	6.8875
Health and beauty	49129.73	2342.55	7.0238

### 10.2. Average Units Per Order

```

1 SELECT 'Product line', AVG(Quantity) AS avg_units
2 FROM 'supermarket analysis'
3 GROUP BY 'Product line';

```

Product line	avg_units
Health and beauty	5.6184
Electronic accessories	5.7118
Home and lifestyle	5.6938
Sports and travel	5.5422
Food and beverages	5.4713
Fashion accessories	5.0674

## 11. Temporal Patterns

### 11.1. Sales by Hour

```

1 SELECT
2     HOUR(CAST(Time AS TIME)) AS hour,
3     SUM(Sales) AS hourly_sales
4 FROM 'supermarket analysis'
5 GROUP BY hour
6 ORDER BY hourly_sales DESC;

```

hour	hourly_sales
8	37629.51
9	31723.17
10	31421.48
11	31288.17
12	30287.39
13	30089.49
14	26713.87
15	26030.33
16	24446.15
18	22995.52

## 11.2. Weekday vs Weekend Sales

```

1 SELECT
2     weekend_flag,
3     COUNT(*) AS transactions,
4     SUM(Sales) AS total_sales
5 FROM (
6     SELECT *,
7         CASE
8             WHEN DAYOFWEEK(STR_TO_DATE(Date, '%Y-%m-%d')) IN (1, 7) THEN
9                 'Weekend'
10            ELSE 'Weekday'
11        END AS weekend_flag
12 FROM 'supermarket analysis'
13 ) AS sub
14 GROUP BY weekend_flag;

```

weekend_flag	transactions	total_sales
Weekday	1000	322966.749

## 12. Payment Method Analysis

```

1 SELECT Payment,
2     COUNT(*) AS transaction_count,
3     ROUND(SUM(Sales), 2) AS total_sales,
4     ROUND(AVG(Sales), 2) AS avg_sales
5 FROM 'supermarket analysis'
6 GROUP BY Payment
7 ORDER BY total_sales DESC;

```

Payment	transaction_count	total_sales	avg_sales
Cash	344	112206.57	326.18
Ewallet	345	109933.11	318.82
Credit card	311	100767.07	324.01

## 13. Profitability Analysis

### 13.1. Most Profitable Branches

```
1 SELECT Branch, SUM('Gross income') AS total_profit
2 FROM 'supermarket analysis'
3 GROUP BY Branch
4 ORDER BY total_profit DESC;
```

Branch	total_profit
Giza	5265.1765
Alex	5057.1650
Cairo	5057.0320

### 13.2. High Margin Product Lines

```
1 SELECT 'Product line',
2        ROUND(AVG('Gross income' / COGS * 100), 2) AS avg_profit_margin
3 FROM 'supermarket analysis'
4 GROUP BY 'Product line'
5 ORDER BY avg_profit_margin DESC;
```

Product line	avg_profit_margin
Health and beauty	5
Electronic accessories	5
Home and lifestyle	5
Sports and travel	5
Food and beverages	5
Fashion accessories	5

## 14. Correlations & Insights

- **High Quantity + Low Rating:** Overpromising products.
- **High Sales + High Margin + Low Units:** Premium pricing.
- **Cash Payment = Lower Basket Size:** Cash customers more price-sensitive.