Retail Sales Performance and Profitability Investigation

SAHEED OLAYINKA ADEBIYI

DATA ANALYST

Problem Statement

This comprehensive analysis addresses some critical business questions raised by leadership regarding retail sales performance, profitability, customer behavior, and operational efficiency which are listed below

- **There is limited visibility into true product profitability, making it difficult to identify which items genuinely drive profit versus volume.**
- **Despite stable sales volume, the business is experiencing fluctuating revenue and inconsistent profit margins across multiple product categories.**
- **Customer behavior patterns and lifetime value are unclear, preventing effective identification of the most valuable customer segments.**
- **The organization lacks insight into seasonal and time-based sales trends, reducing its ability to optimize inventory and promotions.**
- **These gaps hinder data-driven decision-making, impacting pricing strategies, customer retention, and overall business performance.**

Objectives

To investigate fluctuations in revenue, customer behavior, and product profitability while identifying opportunities for improved margins, customer retention, and operational efficiency.

- **Identify whether our revenue growth is translating into real profit or being eroded by high costs**
- **Analyze the categories with high sales volume that also deliver strong profitability**
- **Study which customers drive the most revenue and repeat purchases.**
- **Examine and tailor promotions and product offerings to the right demographic groups.**
- **To determine the sales trend over time**
- **To determine whether customers typically make bulk purchases or small, frequent ones**

Overview of data

- **The data consist of 2,000 rows and 12 columns which is clean without duplicates and empty spaces except for creating new conditional columns**
- **Dataset main columns include: transaction_id, customer_id, gender, age, category, quantity, price_per_unit, total_sale, cogs, total_cost, sale_date, sale_time**

Implementation Outline

- The tool I used for the entire analysis process is Structured Query language (SQL)

KPI

```sql
-- Overview of the data (KPI)
SELECT
    COUNT(DISTINCT customer_id) AS total_customers,
    COUNT(*) AS total_transactions,
    ROUND(SUM(total_sale)) AS total_revenue,
    ROUND(SUM(total_cost)) AS total_costs,
    ROUND(SUM(total_sale - total_cost)) AS total_profit,
    ROUND((SUM(total_sale - total_cost) / SUM(total_sale)) * 100, 2) AS overall_profit_margin,
    ROUND(AVG(total_sale)) AS avg_transaction_value,
    ROUND(AVG(quantity)) AS avg_items_per_transaction
FROM retail_sales;
```

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

total_customers	total_transactions	total_revenue	total_costs	total_profit	overall_profit_margin	avg_transaction_value	avg_items_per_transaction
155	2000	911720	488717	423003	46.40	456	3

ANALYSIS

```
72
73      -- Calculating the overal profit and profit margin
74  •   SELECT  SUM(total_sale) AS total_revenue, SUM(total_cost) AS total_costs,
75      ROUND(SUM(total_sale - total_cost)) AS total_profit,
76      ROUND((SUM(total_sale - total_cost) / SUM(total_sale)) * 100, 2) AS profit_margin_percentage
77      FROM retail_sales;
78
```

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

total_revenue	total_costs	total_profit	profit_margin_percentage
911720	488717	423003	46.40

ANALYSIS

```
78
79      -- Determine the category of product that generates more profit
80      SELECT category, SUM(total_sale - total_cost) AS total_profit,
81      COUNT(*) AS transactions,
82      ROUND(SUM(total_sale)) AS total_revenue,
83      ROUND((SUM(total_sale - total_cost) / SUM(total_sale)) * 100, 2) AS profit_margin
84      FROM retail_sales
85      GROUP BY category
86      ORDER BY total_profit DESC;
```

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

category	total_profit	transactions	total_revenue	profit_margin
Beauty	145082	636	299410	48.46
Electronics	141893	684	313810	45.22
Clothing	136028	680	298500	45.57

ANALYSIS

```sql
-- Top most valuable customers
SELECT customer_id, gender, age,
COUNT(*) AS total_purchases,
ROUND(SUM(total_sale)) AS total_revenue,
ROUND(SUM(total_sale - total_cost)) AS total_profit,
ROUND(AVG(total_sale)) AS avg_transaction_value
FROM retail_sales
GROUP BY customer_id, gender, age
ORDER BY total_revenue DESC LIMIT 10;
```

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

customer_id	gender	age	total_purchases	total_revenue	total_profit	avg_transaction_value
5	Male	59	3	5000	2790	1667
3	Male	31	3	4500	2645	1500
87	Male	36	3	4100	2238	1367
4	Female	29	2	4000	2240	2000
83	Male	33	2	4000	-120	2000
5	Female	20	2	3200	1796	1600
3	Male	54	5	3125	1991	625
1	Male	41	3	3100	1868	1033
37	Male	22	2	3000	1670	1500
1	Female	30	2	2900	1725	1450

ANALYSIS

```
98      -- Determine the demographic
99      -- -- Revenue, total profit, transaction, revenue by gender
100 •   SELECT gender, COUNT(*) AS total_transactions,
101     ROUND(SUM(total_sale)) AS total_revenue,
102     ROUND(AVG(total_sale)) AS avg_transaction_value,
103     ROUND(SUM(total_sale - total_cost)) AS total_profit
104     FROM retail_sales
105     GROUP BY gender
106     ORDER BY total_revenue DESC;
107
```

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

gender	total_transactions	total_revenue	avg_transaction_value	total_profit
Female	1020	465400	456	198448
Male	980	446320	455	224555

ANALYSIS

```sql
-- Revenue, total profit, transaction, revenue by age group
SELECT
    CASE
        WHEN age < 25 THEN '18-24'
        WHEN age BETWEEN 25 AND 34 THEN '25-34'
        WHEN age BETWEEN 35 AND 44 THEN '35-44'
        WHEN age BETWEEN 45 AND 54 THEN '45-54'
        WHEN age >= 55 THEN '55+'
        ELSE 'Unknown'
    END AS age_group, COUNT(*) AS transactions, ROUND(SUM(total_sale)) AS revenue,
    ROUND(AVG(total_sale)) AS avg_transaction_value,
    ROUND(SUM(total_sale - total_cost)) AS total_profit FROM retail_sales
WHERE age IS NOT NULL
GROUP BY age_group
ORDER BY revenue DESC;
```

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

age_group	transactions	revenue	avg_transaction_value	total_profit
35-44	423	197085	466	95671
25-34	406	194090	478	82955
45-54	447	193395	433	93304
55+	427	177990	417	77285
18-24	297	149160	502	73788

ANALYSIS

```
124     -- Categories Product preference by age group
125 •   SELECT CASE
126             WHEN age < 25 THEN '18-24'
127             WHEN age BETWEEN 25 AND 34 THEN '25-34'
128             WHEN age BETWEEN 35 AND 44 THEN '35-44'
129             WHEN age BETWEEN 45 AND 54 THEN '45-54'
130             WHEN age >= 55 THEN '55+'
131         END AS age_group, category, COUNT(*) AS purchases, ROUND(SUM(total_sale)) AS revenue
132         FROM retail_sales WHERE age IS NOT NULL GROUP BY age_group, category ORDER BY age_group, revenue DESC;
```

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

age_group	category	purchases	revenue
18-24	Beauty	106	57720
18-24	Electronics	102	47170
18-24	Clothing	89	44270
25-34	Clothing	140	79180
25-34	Beauty	142	61090
25-34	Electronics	124	53820
35-44	Electronics	160	75285
35-44	Beauty	110	63710
35-44	Clothing	153	58090
45-54	Beauty	149	72450
45-54	Electronics	155	63405
45-54	Clothing	143	57540
55+	Electronics	143	74130
55+	Clothing	155	59420
55+	Beauty	129	44440

ANALYSIS

```sql
-- Category preference by gender
SELECT gender, category, COUNT(*) AS purchases,
ROUND(SUM(total_sale)) AS revenue
FROM retail_sales
GROUP BY gender, category
ORDER BY gender, revenue DESC;
```

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

gender	category	purchases	revenue
Female	Beauty	344	156690
Female	Clothing	336	155240
Female	Electronics	340	153470
Male	Electronics	344	160340
Male	Clothing	344	143260
Male	Beauty	292	142720

ANALYSIS

```sql
-- Sales trend by year
SELECT
    EXTRACT(YEAR FROM sale_date) AS year,
    COUNT(*) AS transactions,
    ROUND(SUM(total_sale)) AS revenue,
    ROUND(SUM(total_sale - total_cost)) AS profit,
    ROUND((SUM(total_sale - total_cost) / SUM(total_sale)) * 100, 2) AS profit_margin_percentage
FROM retail_sales
GROUP BY year
ORDER BY year;
```

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

year	transactions	revenue	profit	profit_margin_percentage
2024	1049	486175	222658	45.80
2025	951	425545	200345	47.08

ANALYSIS

```sql
-- Sales trend by month
SELECT
    EXTRACT(YEAR FROM sale_date) AS year,
    EXTRACT(MONTH FROM sale_date) AS month,
    DATE_FORMAT(sale_date, '%b') AS month_name,
    COUNT(*) AS transactions,
    ROUND(SUM(total_sale)) AS revenue,
    ROUND(SUM(total_sale - total_cost)) AS profit
FROM retail_sales
GROUP BY year, month, month_name
ORDER BY year, month;
```

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

year	month	month_name	transactions	revenue	profit
2024	1	Jan	57	22635	13780
2024	2	Feb	45	16110	8543
2024	3	Mar	47	24505	14657
2024	4	Apr	59	28705	16822
2024	5	May	52	24980	14089
2024	6	Jun	43	20700	12750
2024	7	Jul	42	22195	13020
2024	8	Aug	56	21195	11890
2024	9	Sep	129	61770	24978
2024	10	Oct	146	68235	19331
2024	11	Nov	146	68915	27930
2024	12	Dec	227	106230	44868
2025	1	Jan	60	23790	13963
2025	2	Feb	47	25170	14843
2025	3	Mar	52	20530	11937
2025	4	Apr	47	21925	13320
2025	5	May	60	27010	16142
2025	6	Jun	56	24555	14970
2025	7	Jul	84	35925	21197
2025	8	Aug	57	28270	16304
2025	9	Sep	146	67560	25662
2025	10	Oct	145	57880	17529
2025	11	Nov	126	57135	21738
2025	12	Dec	71	35795	12740

ANALYSIS

```
172     -- Sales Trend by day
173 •   SELECT
174         DAYNAME(sale_date) AS day_name,
175         DAYOFWEEK(sale_date) AS day_number,
176         COUNT(*) AS transactions,
177         ROUND(SUM(total_sale)) AS revenue,
178         ROUND(SUM(total_sale - total_cost), 2) AS profit
179     FROM retail_sales
180     GROUP BY day_name, day_number
181     ORDER BY day_number;
182
```

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

day_name	day_number	transactions	revenue	profit
Sunday	1	290	125015	65761
Monday	2	302	128590	55172
Tuesday	3	305	141285	
Wednesday	4	303	151290	
Thursday	5	277	146635	70558
Friday	6	247	98055	45861
Saturday	7	276	120850	58491

141285

ANALYSIS

```sql
-- Quantity of product by category
SELECT category,
ROUND(AVG(quantity)) AS avg_quantity,
MIN(quantity) AS min_quantity,
MAX(quantity) AS max_quantity,
COUNT(*) AS transactions
FROM retail_sales
GROUP BY category
ORDER BY avg_quantity DESC;
```

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

category	avg_quantity	min_quantity	max_quantity	transactions
Clothing	3	0	4	680
Beauty	3	0	4	636
Electronics	2	1	4	684

ANALYSIS

```sql
-- Sales by time period of the day and respective category
SELECT category, IF(HOUR(sale_time) BETWEEN 6 AND 11, 'Morning',
      IF(HOUR(sale_time) BETWEEN 12 AND 17, 'Afternoon',
       IF(HOUR(sale_time) BETWEEN 18 AND 21, 'Evening', 'Night'))) AS time_period,
    COUNT(*) AS transactions,
    ROUND(SUM(total_sale)) AS revenue,
    ROUND(SUM(total_sale - total_cost)) AS profit
FROM retail_sales
GROUP BY time_period, category
ORDER BY MIN(HOUR(sale_time));
```

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

category	time_period	transactions	revenue	profit
Clothing	Morning	216	93560	55659
Beauty	Morning	188	95010	57905
Electronics	Morning	157	71330	42323
Clothing	Afternoon	118	53425	22954
Beauty	Afternoon	113	54840	25999
Electronics	Afternoon	146	67615	34843
Electronics	Evening	299	143850	54220
Beauty	Evening	273	121170	49141
Clothing	Evening	283	126215	49124
Clothing	Night	63	25300	8291
Electronics	Night	82	31015	10507
Beauty	Night	62	28390	12037

ANALYSIS

```sql
-- Sales by time period of the day
SELECT  IF(HOUR(sale_time) BETWEEN 6 AND 11, 'Morning',
       IF(HOUR(sale_time) BETWEEN 12 AND 17, 'Afternoon',
        IF(HOUR(sale_time) BETWEEN 18 AND 21, 'Evening', 'Night'))) AS time_period,
    COUNT(*) AS transactions,
    ROUND(SUM(total_sale)) AS revenue,
    ROUND(SUM(total_sale - total_cost)) AS profit
FROM retail_sales
GROUP BY time_period
ORDER BY MIN(HOUR(sale_time));
```

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

time_period	transactions	revenue	profit
Morning	561	259900	155887
Afternoon	377	175880	83796
Evening	855	391235	152485
Night	207	84705	30835

Key Finding of Performance Dashboard

Key Findings Summary

- **Established clear investment priorities based on profit margins and transaction volumes**
- **Segmented customers by lifetime value and identified top revenue generators for retention programs**
- **Revealed distinct purchasing patterns across age groups and genders for targeted marketing**
- **Identified peak sales periods for optimized staffing and inventory management**
- **Analyzed quantity patterns to inform inventory planning and promotional strategies**

CONCLUSION

This comprehensive analysis has provided critical insights into six key business questions regarding retail sales performance, profitability, customer behavior, and operational efficiency.

Recommendation

- Increase marketing spend on high-margin, high-volume categories
- Develop win-back campaigns for one-time purchasers
- Implement personalized recommendations based on purchase history
- Adjust staffing levels to match peak sales periods (by day and hour)
- Optimize inventory stocking based on seasonal and monthly trends
- Schedule promotions during slower periods to balance revenue throughout the year
- Implement bulk discount strategies to increase average basket size


THANK
YOU