

Coffee Sales Analysis using SQL

Presented By -
Swapnil Palsapure



Project Overview

Objective :

- Analyze coffee sales data to derive insights and improve business strategies.

Goals :

- Understand transaction details such as IDs, quantities, and prices.
Segment data by store locations to identify performance variations.
- Calculate metrics like total sales, average prices, and popular products for informed decision-making.



Total Revenue

```
select
    SUM(transaction_qty*unit_price) as Total_Revenue
from
    coffee_details
```

total_revenue	🔒
numeric	
698812.33	



Total Revenue generated on each Store location

```
select store_location, SUM(transaction_qty*unit_price) as Total_Revenue  
from coffee_details  
group by 1  
order by Total_Revenue
```



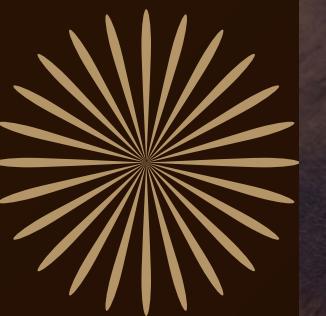
	store_location character varying	total_revenue numeric
1	Lower Manhattan	230057.25
2	Astoria	232243.91
3	Hell's Kitchen	236511.17



Month wise Revenue generated

```
select extract(Month from transaction_date) as Month_no,  
to_char(transaction_date,'Month') as Month,  
SUM(transaction_qty*unit_price) as Total_Revenue  
from coffee_details  
group by Month_no,Month
```

	month_no numeric	month text	total_revenue numeric
1	1	January	81677.74
2	2	February	76145.19
3	3	March	98834.68
4	4	April	118941.08
5	5	May	156727.76
6	6	June	166485.88





Product category wise revenue



```
select product_category,  
SUM(transaction_qty*unit_price) as Total_Revenue  
from coffee_details  
group by product_category  
order by Total_Revenue desc
```

	product_category character varying	total_revenue numeric
1	Coffee	269952.45
2	Tea	196405.95
3	Bakery	82315.64
4	Drinking Chocolate	72416.00
5	Coffee beans	40085.25
6	Branded	13607
7	Loose Tea	11213.60
8	Flavours	8408.8
9	Packaged Chocolate	4407.64





Product type wise revenue

```
select product_type,  
count(transaction_qty) as qty  
from coffee_details  
group by product_type  
order by qty desc
```

	product_type character varying	qty bigint
1	Brewed Chai tea	17183
2	Gourmet brewed coffee	16912
3	Barista Espresso	16403
4	Hot chocolate	11468
5	Brewed Black tea	11350
6	Brewed herbal tea	11245
7	Scone	10173
8	Organic brewed coffee	8489
9	Drip coffee	8477
10	Premium brewed coffee	8135
11	Pastry	6912
12	Biscotti	5711
13	Brewed Green tea	5671
14	Regular syrup	4979
15	Sugar free syrup	1811





Total Revenue percent of each product category

```
with cte as(
select product_category,
ROUND(SUM(transaction_qty*unit_price),2)/(select ROUND(SUM(transaction_qty*unit_price),2)
from coffee_details)
*100 as revenue
from coffee_details
group by product_category)

select product_category,
round(cte.revenue,2) as percent_total_revenue
from cte
order by percent_total_revenue asc
```

	product_category	percent_total_revenue
	character varying	numeric
1	Packaged Chocol...	0.63
2	Flavours	1.20
3	Loose Tea	1.60
4	Branded	1.95
5	Coffee beans	5.74
6	Drinking Chocolate	10.36
7	Bakery	11.78
8	Tea	28.11
9	Coffee	38.63





Total Revenue generated on Store location and product category basis

```
select store_location,product_category,  
SUM(transaction_qty*unit_price) as Total_Revenue  
from coffee_details  
group by 1,2  
order by Total_Revenue desc
```

	store_location character varying	product_category character varying	total_revenue numeric
1	Hell's Kitchen	Coffee	91222.65
2	Astoria	Coffee	89744.30
3	Lower Manhattan	Coffee	88985.50
4	Astoria	Tea	67839.90
5	Hell's Kitchen	Tea	64701.30
6	Lower Manhattan	Tea	63864.75
7	Lower Manhattan	Bakery	28328.94
8	Hell's Kitchen	Bakery	27386.95
9	Astoria	Bakery	26599.75
10	Astoria	Drinking Chocolate	26335.25
11	Hell's Kitchen	Drinking Chocolate	23586.25
12	Lower Manhattan	Drinking Chocolate	22494.50
13	Hell's Kitchen	Coffee beans	18635.10
14	Lower Manhattan	Coffee beans	11230.95
15	Astoria	Coffee beans	10219.20





Avg transaction on Store location and product type basis

```
select store_location,  
round(avg(transaction_qty*unit_price),2)  
as avg_transaction, product_type  
from coffee_details  
group by store_location,3  
order by avg_transaction desc
```

	store_location character varying	avg_transaction numeric	product_type character varying
1	Hell's Kitchen	57.32	Premium Beans
2	Lower Manhattan	33.60	Premium Beans
3	Astoria	31.59	Premium Beans
4	Hell's Kitchen	28.00	Clothing
5	Lower Manhattan	28.00	Clothing
6	Astoria	27.77	Clothing
7	Lower Manhattan	20.78	Organic Beans
8	Hell's Kitchen	20.42	Organic Beans
9	Astoria	20.18	Organic Beans
10	Lower Manhattan	19.09	Gourmet Beans





Store and Month-wise Revenue generated

```
select store_location, to_char(transaction_date, 'Month') as Month,  
SUM(transaction_qty*unit_price) as Total_Revenue  
from coffee_details  
group by 1, Month
```

	store_location character varying	month text	total_revenue numeric
1	Astoria	April	39477.61
2	Astoria	February	25105.34
3	Astoria	January	27313.66
4	Astoria	June	55083.11
5	Astoria	March	32835.43
6	Astoria	May	52428.76
7	Hell's Kitchen	April	40304.14
8	Hell's Kitchen	February	25719.80
9	Hell's Kitchen	January	27820.65
10	Hell's Kitchen	June	56957.08
11	Hell's Kitchen	March	33110.57
12	Hell's Kitchen	May	52598.93
13	Lower Manhattan	April	39159.33
14	Lower Manhattan	February	25320.05
15	Lower Manhattan	January	26543.43
16	Lower Manhattan	June	54445.69
17	Lower Manhattan	March	32888.68
18	Lower Manhattan	May	51700.07





Contribution percentage of tea and coffee in Total Revenue

```
WITH cte AS (
    SELECT SUM(transaction_qty * unit_price) AS total_revenue_overall
    FROM coffee_details
),
cte_2 AS (
    select product_category, total_revenue_overall, SUM(transaction_qty * unit_price) as revenue_category,
    round((sum(transaction_qty * unit_price)/(total_revenue_overall))*100, 2 )as percent_contribution from coffee_details,cte
    group by product_category,cte.total_revenue_overall
)
select * from cte_2
where product_category='Coffee' or product_category='Tea'
```

	product_category character varying 	total_revenue_overall numeric 	revenue_category numeric 	percent_contribution numeric 
1	Tea	698812.33	196405.95	28.11
2	Coffee	698812.33	269952.45	38.63





Cumulative sum over time

```
with cte as(
    select transaction_date,sum(transaction_qty*unit_price) as revenue
    from coffee_details
    group by 1
),
cte_2 as(
    select *, sum(revenue) over(order by transaction_date) as cumulative_sum from cte
)
select * from cte_2
```

	transaction_date date	revenue numeric	cumulative_sum numeric
1	2023-01-01	2508.20	2508.20
2	2023-01-02	2403.35	4911.55
3	2023-01-03	2565.00	7476.55
4	2023-01-04	2220.10	9696.65
5	2023-01-05	2418.85	12115.50
6	2023-01-06	2273.85	14389.35
7	2023-01-07	2619.65	17009.00
8	2023-01-08	2638.53	19647.53
9	2023-01-09	2676.61	22324.14
10	2023-01-10	2685.65	25009.79





Highest Avg sales amt per date

```
with cte as(
select product_type,
round(avg(transaction_qty*unit_price),2) as Total_Sales,
transaction_date
from coffee_details
group by product_type ,3
order by 2 desc),

cte_2 as(
select product_type, transaction_date, Total_Sales,
ROW_NUMBER()OVER(PARTITION BY transaction_date ORDER BY 2 desc)
as RowNo
from cte
)
select product_type, transaction_date, Total_Sales from cte_2
where RowNo<=1
```

	product_type character varying	transaction_date date	total_sales numeric
1	Hot chocolate	2023-01-01	5.91
2	Brewed herbal tea	2023-01-02	4.22
3	Organic brewed coffee	2023-01-03	4.81
4	Premium brewed coff...	2023-01-04	4.81
5	Organic brewed coffee	2023-01-05	4.46
6	Biscotti	2023-01-06	3.39
7	Drinking Chocolate	2023-01-07	6.40
8	Espresso Beans	2023-01-08	16.65
9	Drinking Chocolate	2023-01-09	11.94
10	Sugar free syrup	2023-01-10	1.24





*Thank
you*



[linkedin.com/in/contact-swapnilpalsapure/](https://www.linkedin.com/in/contact-swapnilpalsapure/)



github.com/Swapnil1A