

Let's Connect

sohamghosh7704@gmail.com

SQL DATA ANALYSIS PROJECT

Customer Revenue Insights from
E-commerce Dataset 2023 (Jan - Jun)

CREATED BY: SOHAM GHOSH

Data Analyst | MySQL • Excel

Published: June 2025



SKILLS APPLIED

1. MySQL Integration with DBeaver :

Imported and queried a structured dataset using MySQL database through the DBeaver SQL client for efficient query management and result visualization.

2. Data Import & Preprocessing :

Loaded CSV files into MySQL tables, handled data type conversions, and ensured clean schema structure for relational operations.

3. Data Aggregation Using GROUP BY & ORDER BY :

Performed aggregations like total revenue, sales count, and quantity sold, and sorted results to uncover trends and top performers.

4. Advanced Filtering Using Subqueries :

Applied subqueries to extract specific data insights such as top-selling products, high-revenue customers and revenue contribution for each product.

5. Analytical Queries with Window Functions :

Used RANK() and DENSE_RANK() to create leaderboards and rank entities (e.g., top stores per month), enabling powerful comparative insights.



Query 1: List all unique store locations.

```
select distinct store_location from c_sales;
```

c_sales 1 ×	
select distinct store_location Enter a SQL expression to filter results ▾	
⊙	A-Z store_location
1	Lower Manhattan
2	Hell's Kitchen
3	Astoria

Business Problem

Solved :

Identify all operational retail store locations for analysis or expansion strategy.

Count total no. of transactions : Query 2

Business Problem

Solved :

Understand the overall transaction volume to evaluate customer activity and store performance.

```
select count(*) as total_transactions from c_sales;
```

Results 1 ×	
select count(*) as total_trans Enter a SQL expression to filter results ▾	
⊙	total_transactions
1	149,116

Query 3 : Find the total quantity sold

```
select sum(transaction_qty) as total_quantity_sold  
from c_sales;
```

Results 1 x

select sum(transaction_qty) as total_quantity_sold f | Enter a SQL expression

total_quantity_sold
214,470

Business Problem Solved :

Measure total sales volume to assess product movement and inventory requirements.

Find the earliest and latest transaction date of order : Query 4

```
select min(transaction_date) earliest_date, max(transaction_date) latest_date  
from c_sales;
```

Results 1 x

select min(transaction_date) earliest_date, max(trai | Enter a SQL expression to filter results (use Ctrl+Space)

earliest_date	latest_date
2023-01-01	2023-06-30

Business Problem Solved :

Define the time range of the dataset for accurate time-series or trend analysis



Query 5 : Count how many products were sold per category.

```
select product_category, SUM(transaction_qty) AS total_quantity
from c_sales
group by product_category
order by total_quantity desc;
```

es 1 X

product_category, SUM(transaction_qty) AS total_quantity | Enter a SQL expression to filter results (

product_category	total_quantity
Coffee	89,250
Tea	69,737
Bakery	23,214
Drinking Chocolate	17,457
Flavours	10,511
Coffee beans	1,828
Loose Tea	1,210
Branded	776
Packaged Chocolate	487

Business Problem Solved :

Evaluate which product categories perform best in terms of units sold.

Find the total sales revenue (qty * unit_price) by store location : Query 6

```
select store_location,
concat('$ ', format(round(sum(transaction_qty * unit_price),2),2)) as total_sales
from c_sales
group by store_location
order by total_sales desc;
```

sales 1 X

select store_location, concat('\$ ', format(round(sum(transaction_qty * unit_price),2),2)) as total_sales | Enter a SQL expression to filter results (use Ctrl+Space)

store_location	total_sales
Hell's Kitchen	\$ 236,511.17
Astoria	\$ 232,243.91
Lower Manhattan	\$ 230,057.25

Business Problem Solved : Compare store performance based on revenue to allocate resources and plan marketing.





Query 7 : Top 5 Products by Revenue Generated

```
select product_type,  
concat('$ ', format(sum(transaction_qty * unit_price), 2)) as revenue  
from c_sales  
group by product_type  
order by sum(transaction_qty * unit_price) desc  
limit 5;
```

Business Problem Solved :

Identify the most profitable products to optimize inventory and marketing focus.

sales 1 ×	
select product_type, concat('\$ ', format(sum(transaction_qty * unit_price), 2)) as revenue from c_sales group by product_type order by sum(transaction_qty * unit_price) desc limit 5;	
product_type	revenue
Barista Espresso	\$ 91,406.20
Brewed Chai tea	\$ 77,081.95
Hot chocolate	\$ 72,416.00
Gourmet brewed coffee	\$ 70,034.60
Brewed Black tea	\$ 47,932.00

Daily Revenue Trend (February 2023) : Query 8

```
select transaction_date,  
concat('$ ', format(round(sum(transaction_qty * unit_price), 2), 2)) AS daily_revenue  
from c_sales  
where transaction_date like '2023-02-%'  
group by transaction_date  
order by transaction_date;
```

Business Problem Solved :

Analyze daily revenue patterns to detect seasonality or peak/off or peak days in a given month.

sales 1 ×	
select transaction_date, concat('\$ ', format(round(sum(transaction_qty * unit_price), 2), 2)) AS daily_revenue from c_sales where transaction_date like '2023-02-%' group by transaction_date order by transaction_date;	
transaction_date	daily_revenue
2023-02-01	\$ 2,466.30
2023-02-02	\$ 2,506.90
2023-02-03	\$ 2,591.45
2023-02-04	\$ 2,551.70
2023-02-05	\$ 2,304.70
2023-02-06	\$ 2,203.40
2023-02-07	\$ 2,434.55
2023-02-08	\$ 2,762.43
2023-02-09	\$ 2,610.63
2023-02-10	\$ 2,901.60





Query 9 : Top 10 transaction counts by hour for (time-based analysis)

```
select concat(lpad(extract(hour from transaction_time),2, '0'),' th') as hour,  
count(*) as transaction_count  
from c_sales  
group by hour  
order by transaction_count desc  
limit 10;
```

hour	transaction_count
10 th	18,545
09 th	17,764
08 th	17,654
07 th	13,428
11 th	9,766
16 th	9,093
15 th	8,979
14 th	8,933
17 th	8,745
13 th	8,714

Business Problem Solved :

Discover peak shopping hours to optimize staffing and promotional timing.

Top 3 Most Sold Product Category by quantity : Query 10

Business Problem Solved :

Understand customer preference trends to plan future stock and promotions.

```
select product_category, sum(transaction_qty) as total_quantity  
from c_sales  
group by product_category  
order by total_quantity desc  
limit 3;
```

product_category	total_quantity
1 Coffee	89,250
2 Tea	69,737
3 Bakery	23,214



Query 11 : Top 10 Product Type with highest grossing Average Unit Price

```
select product_type,  
concat('$ ', format(round(avg(unit_price),2),2)) as avg_unit_price  
from c_sales  
group by product_type  
order by avg(unit_price) desc  
limit 10;
```

Business Problem

Solved :

Identify premium products for pricing strategy, bundling, or customer targeting.

product_type	avg_unit_price
Premium Beans	\$ 34.03
Clothing	\$ 27.89
Organic Beans	\$ 20.29
Gourmet Beans	\$ 18.57
House blend Beans	\$ 18.00
Espresso Beans	\$ 17.43
Housewares	\$ 13.45
Drinking Chocolate	\$ 10.26
Green beans	\$ 10.00
Chai tea	\$ 9.71

Find the best - selling product in each store location (by quantity) : Query 12

```
with best as (  
  select store_location, product_detail, sum(transaction_qty) as total_quantity,  
         dense_rank() over(partition by store_location order by sum(transaction_qty) desc) as rnk  
  from c_sales  
  group by store_location ,product_detail  
)  
select store_location, product_detail, total_quantity  
from best  
where rnk = 1  
order by total_quantity desc;
```

store_location	product_detail	total_quantity
Hell's Kitchen	Ouro Brasileiro shot	1,854
Astoria	Dark chocolate Lg	1,755
Lower Manhattan	Peppermint Lg	1,582

Business Problem Solved : Pinpoint local customer preferences for store-specific merchandising.

Query 13 : Compare Weekday vs Weekend sales

```
select case
  when dayofweek(transaction_date) in (1, 7) then 'Weekend' -- 1 = Sunday, 7 = Saturday
  else 'Weekday'
end as day_type,
concat('$ ', format(round(sum(transaction_qty * unit_price),2),2)) AS total_revenue
from c_sales
group by day_type
order by total_revenue desc;
```

results 1 x

Select case when dayofweek(transaction_date) in (1, 7) then 'Weekend' else 'Weekday' end as day_type, concat('\$ ', format(round(sum(transaction_qty * unit_price),2),2)) AS total_revenue from c_sales group by day_type order by total_revenue desc;

day_type	total_revenue
Weekday	\$ 503,587.54
Weekend	\$ 195,224.79

Business Problem Solved : Understand customer behavior across weekdays and weekends to adjust store operations and marketing.

Find revenue contribution % of each product category : Query 14

```
select product_category,
format(round(sum(transaction_qty * unit_price),2),2) as category_revenue,
format(round(sum(transaction_qty * unit_price) /
(select sum(transaction_qty * unit_price) from c_sales) * 100, 2),2) as revenue_percentage
from c_sales
group by product_category;
```

sales 1 x

Select product_category, format(round(sum(transaction_qty * unit_price),2),2) as category_revenue, format(round(sum(transaction_qty * unit_price) / (select sum(transaction_qty * unit_price) from c_sales) * 100, 2),2) as revenue_percentage from c_sales group by product_category;

product_category	category_revenue	revenue_percentage
Coffee	269,952.45	38.63
Tea	196,405.95	28.11
Drinking Chocolate	72,416.00	10.36
Bakery	82,315.64	11.78
Flavours	8,408.80	1.20
Loose Tea	11,213.60	1.60
Coffee beans	40,085.25	5.74
Packaged Chocolate	4,407.64	0.63
Branded	13,607.00	1.95

Business Problem Solved : Assess how each product category contributes to overall revenue to prioritize business focus.

Query 15 : Identify Slowest and Fastest-moving products (by quantity sold).

Fastest - moving product by quantity sold :

```
select product_detail, sum(transaction_qty) as total_quantity
from c_sales
group by product_detail
order by total_quantity desc
limit 1;
```

product_detail	total_quantity
Earl Grey Rg	4,708

Slowest - moving product by quantity sold :

```
select product_detail, sum(transaction_qty) as total_quantity
from c_sales
group by product_detail
order by total_quantity
limit 1;
```

product_detail	total_quantity
Dark chocolate	118

Business Problem Solved : Optimize inventory by promoting slow movers and ensuring fast movers remain in stock.

Query 16 : Create a rank of stores based on monthly revenue using RANK() choosing a single month (Jun)

```
with monthly_revenue as (  
  select store_location,  
         extract(month from transaction_date) as month_num,  
         concat('$ ', format(round(sum(transaction_qty * unit_price),2),2)) as monthly_revenue  
  from c_sales  
  group by store_location, month_num  
)  
select store_location, month_num, monthly_revenue,  
       rank() over (partition by month_num order by monthly_revenue desc) as revenue_rank  
from monthly_revenue  
where month_num = 6  
order by month_num, revenue_rank;
```

sales 1 X

with monthly_revenue as (select store_location, ex | Enter a SQL expression to filter results (use Ctrl+Space)

store_location	month_num	monthly_revenue	revenue_rank
Hell's Kitchen	6	\$ 56,957.08	1
Astoria	6	\$ 55,083.11	2
Lower Manhattan	6	\$ 54,445.69	3

Business Problem Benchmark store performance monthly to recognize
Solved : high performers and support low performers.

-- THANK YOU --