Customer Order Analysis Using SQL

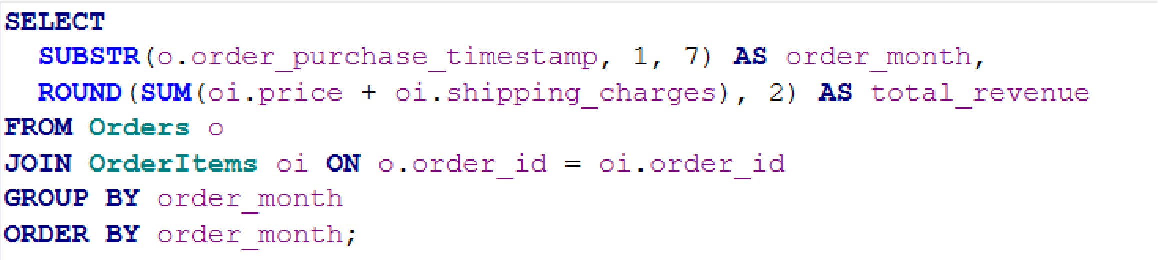
Github : https://github.com/DineshM1703

1. Introduction

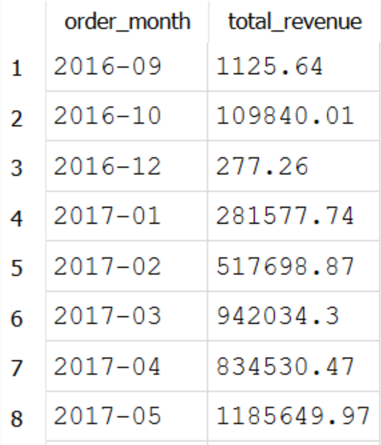
This project analyzes customer orders from an e-commerce platform using SQL. The dataset includes tables for customers, orders, order items, payments, and products. Insights were generated by querying a SQLite database built in DB Browser for SQLite.

2. Question & Query

Q1. What is the total revenue generated each month?



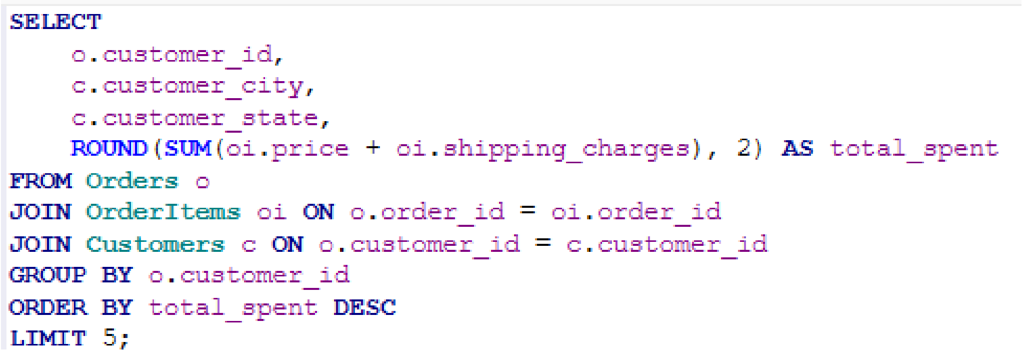
```
SELECT
  SUBSTR(o.order_purchase_timestamp, 1, 7) AS order_month,
  ROUND(SUM(oi.price + oi.shipping_charges), 2) AS total_revenue
FROM Orders o
JOIN OrderItems oi ON o.order_id = oi.order_id
GROUP BY order_month
ORDER BY order_month;
```



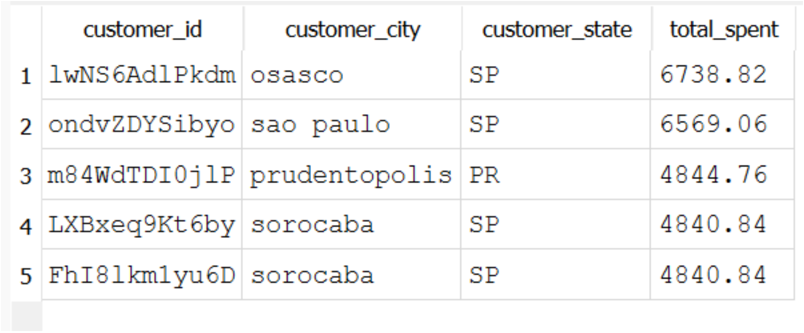
	order_month	total_revenue
1	2016-09	1125.64
2	2016-10	109840.01
3	2016-12	277.26
4	2017-01	281577.74
5	2017-02	517698.87
6	2017-03	942034.3
7	2017-04	834530.47
8	2017-05	1185649.97

This query calculates the total monthly revenue from all customer orders by summing the product prices and shipping charges. The result shows how much revenue the business earns month by month.

Q2. Who are the top 5 customers by total spending?



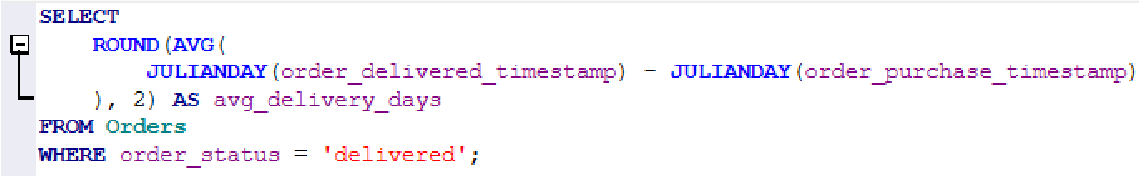
```
SELECT
    o.customer_id,
    c.customer_city,
    c.customer_state,
    ROUND(SUM(oi.price + oi.shipping_charges), 2) AS total_spent
FROM Orders o
JOIN OrderItems oi ON o.order_id = oi.order_id
JOIN Customers c ON o.customer_id = c.customer_id
GROUP BY o.customer_id
ORDER BY total_spent DESC
LIMIT 5;
```



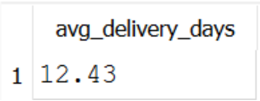
	customer_id	customer_city	customer_state	total_spent
1	lwNS6AdlPkdm	osasco	SP	6738.82
2	ondvZDYSibyo	sao paulo	SP	6569.06
3	m84WdTDI0jlP	prudentopolis	PR	4844.76
4	LXBxeq9Kt6by	sorocaba	SP	4840.84
5	FhI8lkm1yu6D	sorocaba	SP	4840.84

This query identifies the top 5 customers based on their total spending, including product price and shipping. These customers are likely VIPs and can be prioritized for loyalty programs, early product access, or targeted marketing campaigns.

Q3. How long does it take on average to deliver an order?



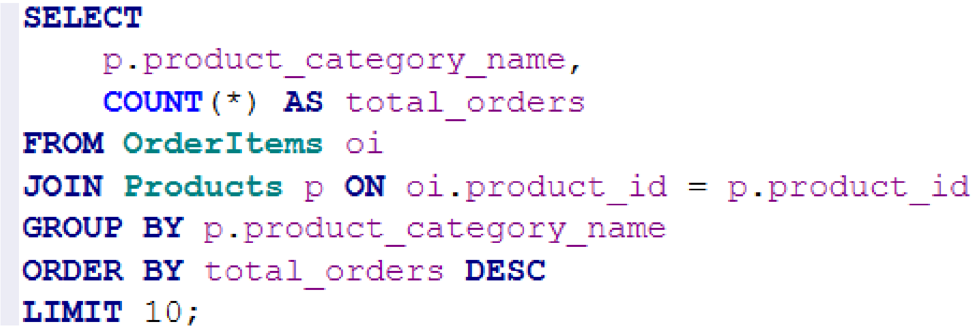
```
SELECT
    ROUND(AVG(
        JULIANDAY(order_delivered_timestamp) - JULIANDAY(order_purchase_timestamp)
    ), 2) AS avg_delivery_days
FROM Orders
WHERE order_status = 'delivered';
```



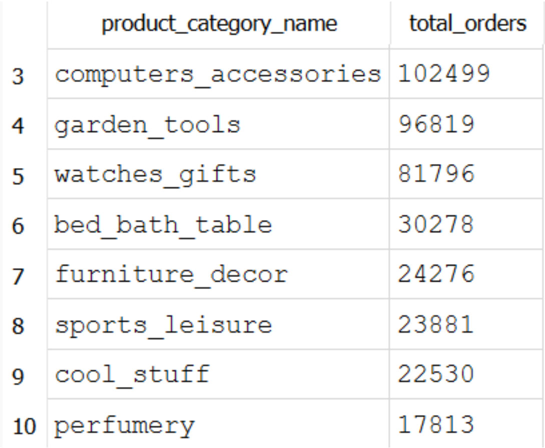
	avg_delivery_days
1	12.43

The average delivery time helps identify how efficient the order fulfillment process is. If the value is above 10 days, this might indicate shipping delays. Faster delivery improves customer satisfaction and repeat purchases.

Q4. Which product categories are ordered the most?



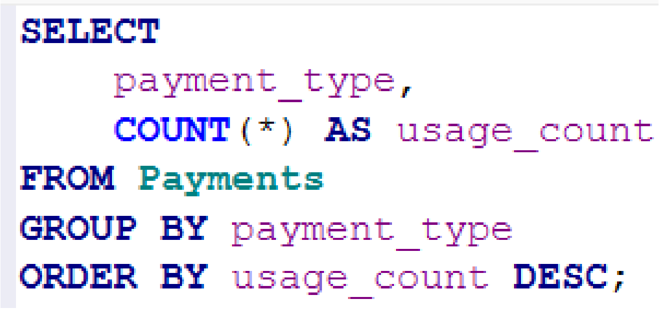
```
SELECT
    p.product_category_name,
    COUNT(*) AS total_orders
FROM OrderItems oi
JOIN Products p ON oi.product_id = p.product_id
GROUP BY p.product_category_name
ORDER BY total_orders DESC
LIMIT 10;
```



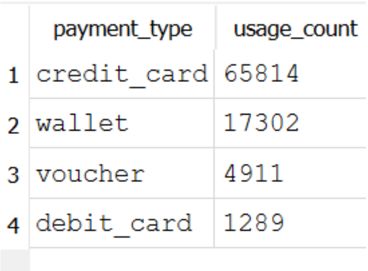
	product_category_name	total_orders
3	computers_accessories	102499
4	garden_tools	96819
5	watches_gifts	81796
6	bed_bath_table	30278
7	furniture_decor	24276
8	sports_leisure	23881
9	cool_stuff	22530
10	perfumery	17813

This query reveals the top 10 product categories based on the number of times they were ordered. These high-demand categories help businesses prioritize inventory, promotions, and supplier focus. For example, if "toys" or "watches_gifts" are at the top, they can be featured during campaigns.

Q5. Which payment types are used most by customers?



```
SELECT
    payment_type,
    COUNT(*) AS usage_count
FROM Payments
GROUP BY payment_type
ORDER BY usage_count DESC;
```

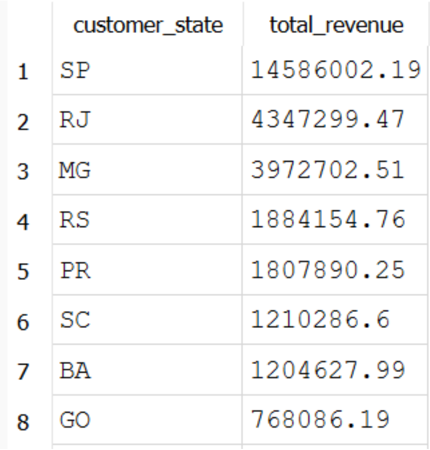


	payment_type	usage_count
1	credit_card	65814
2	wallet	17302
3	voucher	4911
4	debit_card	1289

The majority of customers prefer to pay using **credit cards**, followed by **voucher**, **debit card**, and other options. This insight can guide decisions about which payment methods to promote or improve. If most users use credit cards, you may consider partnerships or better fraud protection features.

Q6. Revenue by Customer State

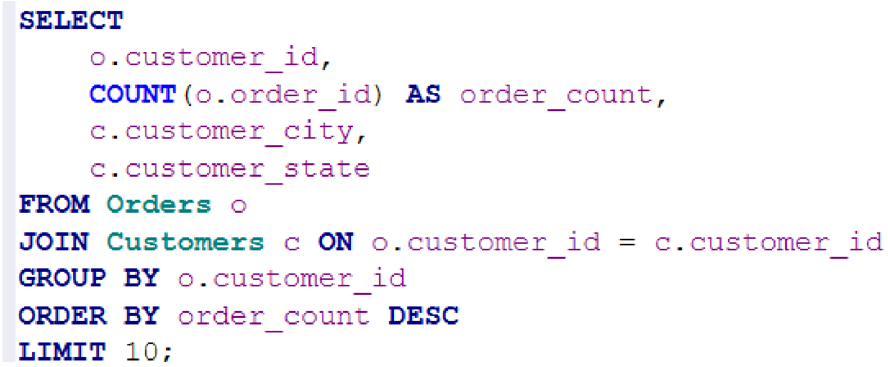
```
SELECT
    c.customer_state,
    ROUND(SUM(oi.price + oi.shipping_charges), 2) AS total_revenue
FROM Orders o
JOIN Customers c ON o.customer_id = c.customer_id
JOIN OrderItems oi ON o.order_id = oi.order_id
GROUP BY c.customer_state
ORDER BY total_revenue DESC
LIMIT 10;
```



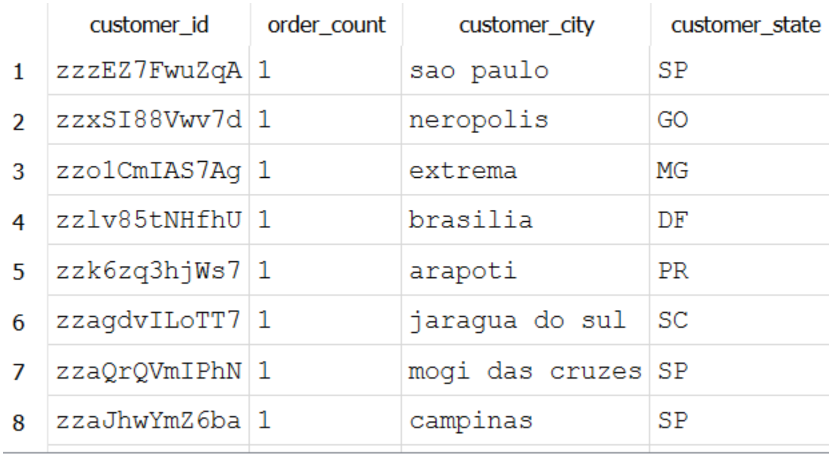
	customer_state	total_revenue
1	SP	14586002.19
2	RJ	4347299.47
3	MG	3972702.51
4	RS	1884154.76
5	PR	1807890.25
6	SC	1210286.6
7	BA	1204627.99
8	GO	768086.19

See which states contribute the most revenue. This helps target regional marketing and logistics improvements.

Q7. Customers with Most Orders



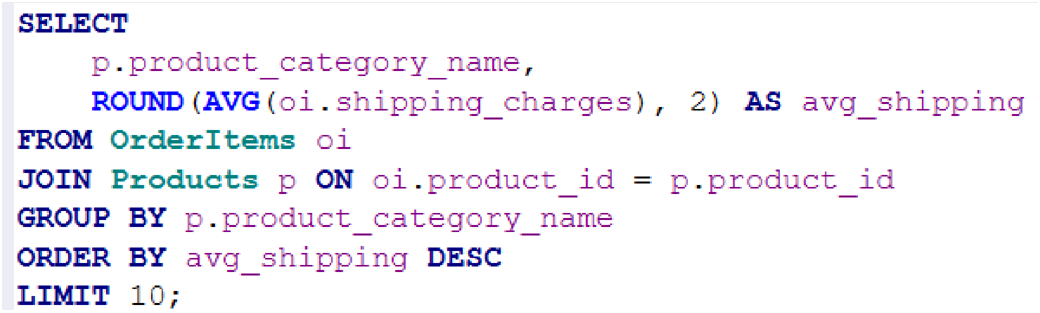
```
SELECT
    o.customer_id,
    COUNT(o.order_id) AS order_count,
    c.customer_city,
    c.customer_state
FROM Orders o
JOIN Customers c ON o.customer_id = c.customer_id
GROUP BY o.customer_id
ORDER BY order_count DESC
LIMIT 10;
```



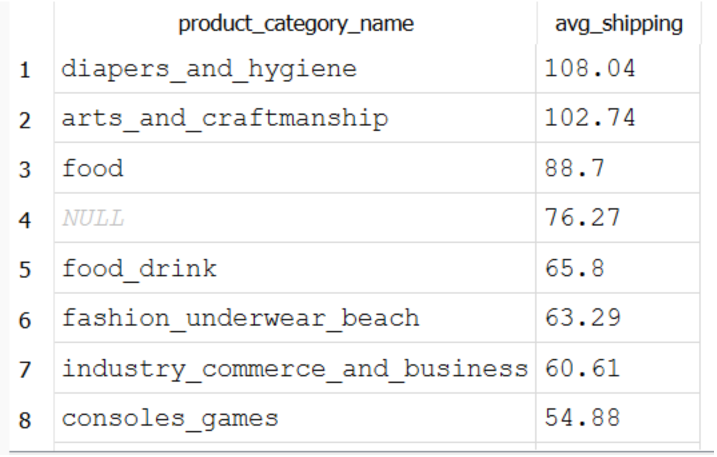
	customer_id	order_count	customer_city	customer_state
1	zzzEZ7FwuZqA	1	sao paulo	SP
2	zzxSI88Vwv7d	1	neropolis	GO
3	zzo1CmIAS7Ag	1	extrema	MG
4	zzlv85tNHfhU	1	brasilia	DF
5	zzk6zq3hjWs7	1	arapoti	PR
6	zzagdvILoTT7	1	jaragua do sul	SC
7	zzaQrQVmIPhN	1	mogi das cruzes	SP
8	zzaJhwYmZ6ba	1	campinas	SP

Shows the most loyal/repeat customers. Great for loyalty rewards and high-LTV targeting.

Q8. Average Shipping Fee per Category



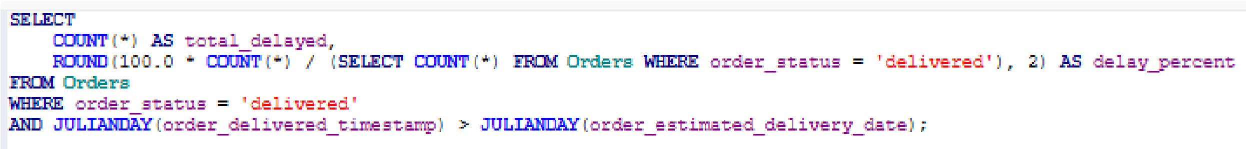
```
SELECT
    p.product_category_name,
    ROUND(AVG(oi.shipping_charges), 2) AS avg_shipping
FROM OrderItems oi
JOIN Products p ON oi.product_id = p.product_id
GROUP BY p.product_category_name
ORDER BY avg_shipping DESC
LIMIT 10;
```



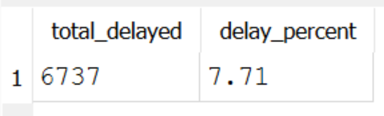
	product_category_name	avg_shipping
1	diapers_and_hygiene	108.04
2	arts_and_craftmanship	102.74
3	food	88.7
4	NULL	76.27
5	food_drink	65.8
6	fashion_underwear_beach	63.29
7	industry_commerce_and_business	60.61
8	consoles_games	54.88

High shipping fees could indicate bulky or heavy products. This insight helps with pricing and free-shipping decisions.

Q9. Delayed Deliveries vs Estimated



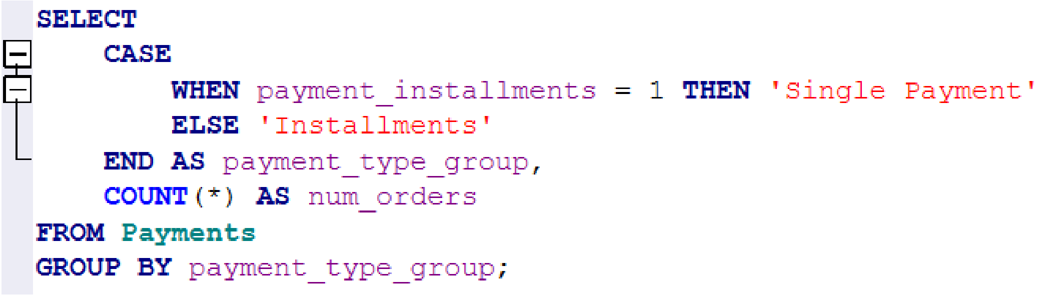
```
SELECT
    COUNT(*) AS total_delayed,
    ROUND(100.0 * COUNT(*) / (SELECT COUNT(*) FROM Orders WHERE order_status = 'delivered'), 2) AS delay_percent
FROM Orders
WHERE order_status = 'delivered'
AND JULIANDAY(order_delivered_timestamp) > JULIANDAY(order_estimated_delivery_date);
```



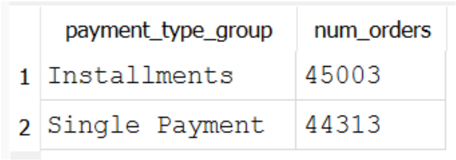
	total_delayed	delay_percent
1	6737	7.71

Find out what % of deliveries were late. This helps identify logistics efficiency and improve service level agreements (SLAs).

Q10. Single vs Multi-Installment Payments



```
SELECT
    CASE
        WHEN payment_installments = 1 THEN 'Single Payment'
        ELSE 'Installments'
    END AS payment_type_group,
    COUNT(*) AS num_orders
FROM Payments
GROUP BY payment_type_group;
```



	payment_type_group	num_orders
1	Installments	45003
2	Single Payment	44313

Shows how often customers use payment plans. Helpful for financial planning and offering buy-now-pay-later features.