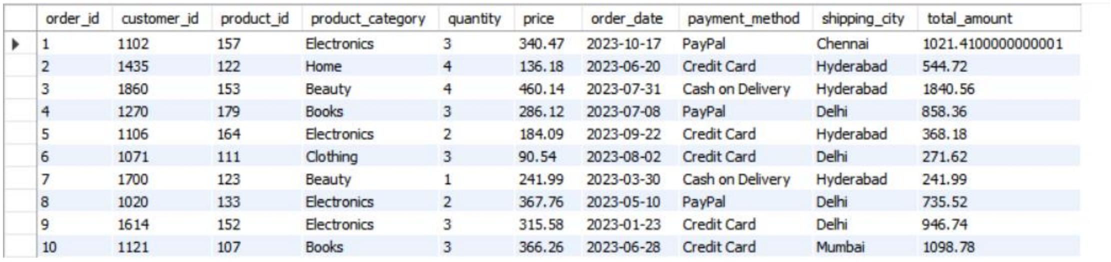
ELEVATE LABS
TASK-3

The E-Commerce dataset contains detailed information about customer orders, including product details, quantities, prices, payment methods, and shipping locations. This dataset has been imported into SQL Workbench for analysis. Using SQL queries, we aim to extract meaningful business insights such as sales trends, customer behavior, and product performance. The following commands demonstrate how SQL can be used to analyze and summarize this data effectively

1. SELECT * FROM ecommerce_dataset LIMIT 10;

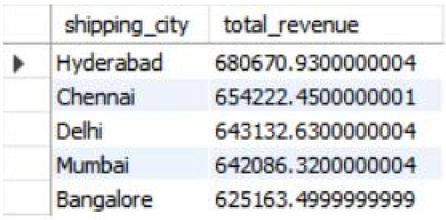


	order_id	customer_id	product_id	product_category	quantity	price	order_date	payment_method	shipping_city	total_amount
▶	1	1102	157	Electronics	3	340.47	2023-10-17	PayPal	Chennai	1021.4100000000001
	2	1435	122	Home	4	136.18	2023-06-20	Credit Card	Hyderabad	544.72
	3	1860	153	Beauty	4	460.14	2023-07-31	Cash on Delivery	Hyderabad	1840.56
	4	1270	179	Books	3	286.12	2023-07-08	PayPal	Delhi	858.36
	5	1106	164	Electronics	2	184.09	2023-09-22	Credit Card	Hyderabad	368.18
	6	1071	111	Clothing	3	90.54	2023-08-02	Credit Card	Delhi	271.62
	7	1700	123	Beauty	1	241.99	2023-03-30	Cash on Delivery	Hyderabad	241.99
	8	1020	133	Electronics	2	367.76	2023-05-10	PayPal	Delhi	735.52
	9	1614	152	Electronics	3	315.58	2023-01-23	Credit Card	Delhi	946.74
	10	1121	107	Books	3	366.26	2023-06-28	Credit Card	Mumbai	1098.78

2. Use SELECT, WHERE, ORDER BY, GROUP BY

Goal: Identify top cities by total revenue in the second half of the year.

```
SELECT shipping_city, SUM(total_amount) AS total_revenue
FROM ecommerce_dataset
WHERE order_date >= '2023-07-01'
GROUP BY shipping_city
ORDER BY total_revenue DESC;
```



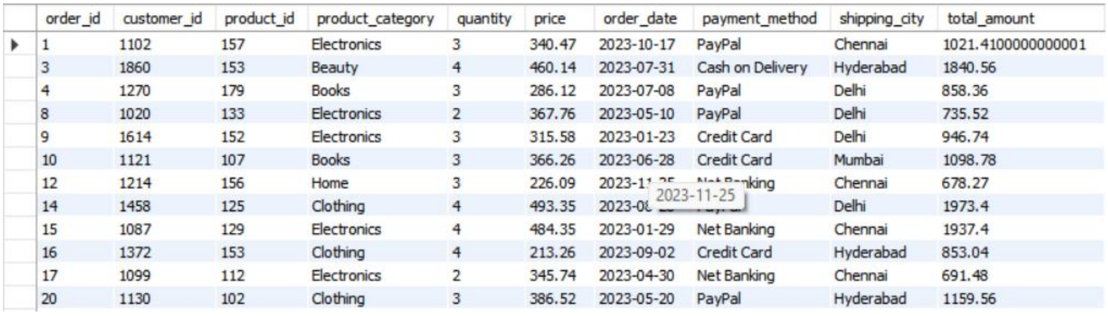
	shipping_city	total_revenue
▶	Hyderabad	680670.9300000004
	Chennai	654222.4500000001
	Delhi	643132.6300000004
	Mumbai	642086.3200000004
	Bangalore	625163.4999999999

2. Use subqueries

Goal:

Identify the top 5 most expensive products (by unit price) that have been ordered.

```
SELECT *
FROM ecommerce_dataset e1
WHERE total_amount > (
   SELECT AVG(total_amount)
   FROM ecommerce_dataset e2
   WHERE e2.shipping_city = e1.shipping_city
);
```

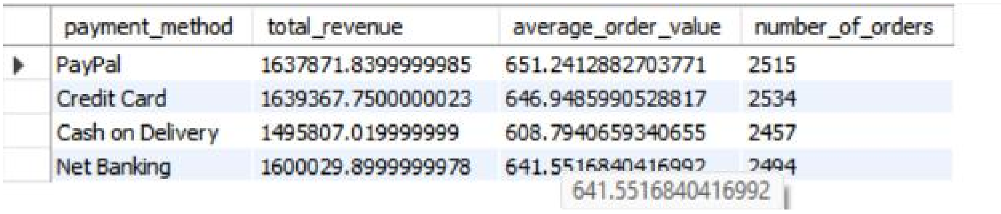


order_id	customer_id	product_id	product_category	quantity	price	order_date	payment_method	shipping_city	total_amount
1	1102	157	Electronics	3	340.47	2023-10-17	PayPal	Chennai	1021.4100000000001
3	1860	153	Beauty	4	460.14	2023-07-31	Cash on Delivery	Hyderabad	1840.56
4	1270	179	Books	3	286.12	2023-07-08	PayPal	Delhi	858.36
8	1020	133	Electronics	2	367.76	2023-05-10	PayPal	Delhi	735.52
9	1614	152	Electronics	3	315.58	2023-01-23	Credit Card	Delhi	946.74
10	1121	107	Books	3	366.26	2023-06-28	Credit Card	Mumbai	1098.78
12	1214	156	Home	3	226.09	2023-11-25	Net Banking	Chennai	678.27
14	1458	125	Clothing	4	493.35	2023-0[illegible]	[illegible]	Delhi	1973.4
15	1087	129	Electronics	4	484.35	2023-01-29	Net Banking	Chennai	1937.4
16	1372	153	Clothing	4	213.26	2023-09-02	Credit Card	Hyderabad	853.04
17	1099	112	Electronics	2	345.74	2023-04-30	Net Banking	Chennai	691.48
20	1130	102	Clothing	3	386.52	2023-05-20	PayPal	Hyderabad	1159.56

3. Aggregate Functions

Goal : Find the total revenue generated and average order value for each payment method.

```
SELECT payment_method,
       SUM(total_amount) AS total_revenue,
       AVG(total_amount) AS average_order_value,
       COUNT(order_id) AS number_of_orders
FROM ecommerce_dataset
GROUP BY payment_method;
```

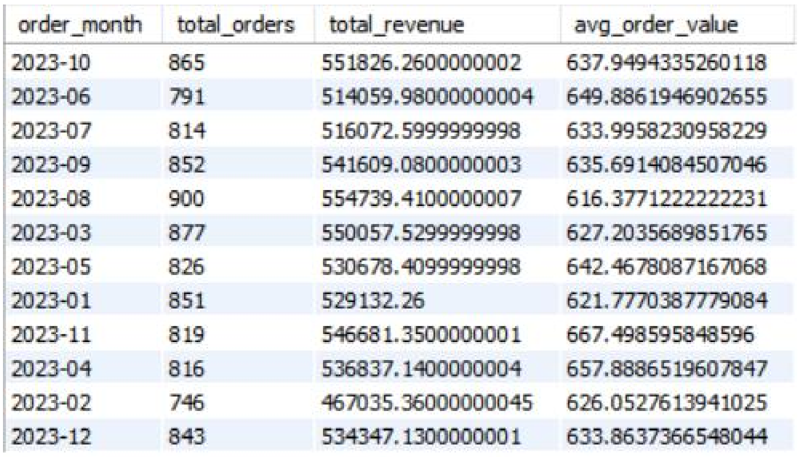


payment_method	total_revenue	average_order_value	number_of_orders
PayPal	1637871.8399999985	651.2412882703771	2515
Credit Card	1639367.7500000023	646.9485990528817	2534
Cash on Delivery	1495807.019999999	608.7940659340655	2457
Net Banking	1600029.8999999978	641.5516840416992	2494

4.Views

Goal : Analyze total sales and average order value by month to track business performance over time.

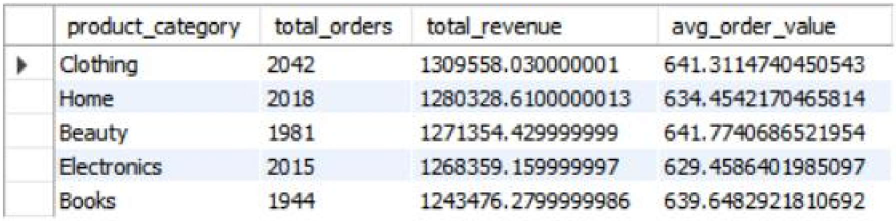
```
CREATE VIEW monthly_sales_summary AS
SELECT
   DATE_FORMAT(STR_TO_DATE(order_date, '%Y-%m-%d'), '%Y-%m')
AS order_month,
   COUNT(order_id) AS total_orders,
   SUM(total_amount) AS total_revenue,
   AVG(total_amount) AS avg_order_value
FROM ecommerce_dataset
GROUP BY order_month;
```



order_month	total_orders	total_revenue	avg_order_value
2023-10	865	551826.2600000002	637.9494335260118
2023-06	791	514059.98000000004	649.8861946902655
2023-07	814	516072.5999999998	633.9958230958229
2023-09	852	541609.0800000003	635.6914084507046
2023-08	900	554739.4100000007	616.3771222222231
2023-03	877	550057.5299999998	627.2035689851765
2023-05	826	530678.4099999998	642.4678087167068
2023-01	851	529132.26	621.7770387779084
2023-11	819	546681.3500000001	667.498595848596
2023-04	816	536837.1400000004	657.8886519607847
2023-02	746	467035.36000000045	626.0527613941025
2023-12	843	534347.1300000001	633.8637366548044

Goal : Identify which product categories generate the highest revenue to focus marketing and inventory.

```
CREATE VIEW category_revenue_summary AS
SELECT
    product_category,
    COUNT(order_id) AS total_orders,
    SUM(total_amount) AS total_revenue,
    AVG(total_amount) AS avg_order_value
FROM ecommerce_dataset
GROUP BY product_category
ORDER BY total_revenue DESC;
```



	product_category	total_orders	total_revenue	avg_order_value
▶	Clothing	2042	1309558.030000001	641.3114740450543
	Home	2018	1280328.6100000013	634.4542170465814
	Beauty	1981	1271354.429999999	641.7740686521954
	Electronics	2015	1268359.159999997	629.4586401985097
	Books	1944	1243476.2799999986	639.6482921810692