Ecommerce SQL Analysis — Professional Report

Generated: 2025-12-11 08:51:35 UTC

This report presents a professional-level analysis of a sample Ecommerce dataset using SQL. It includes data extraction queries, aggregations, joins, subqueries, views, index recommendations, visual charts, and actionable insights.

Database Schema

The database contains four tables: customers(id, name, country), products(id, name, price), orders(id, customer_id, order_date), order_items(id, order_id, product_id, qty).

1. Total Sales per Customer

This query aggregates total spending per customer using JOINs across orders, order_items, and products. Useful to identify high-value customers.

```
SQL:
SELECT c.name, c.country, SUM(p.price*oi.qty) AS total_spent FROM customers c JOIN orders o ON c.id=o.customer_id JOIN
```

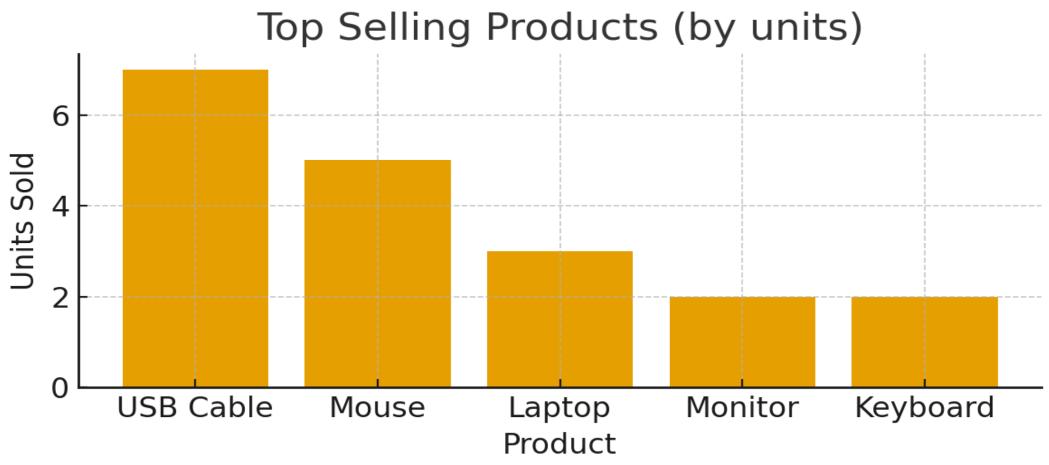
name	country	total_spent
Alice	USA	3995.0
Diana	UK	300.0
Carlos	USA	95.0
Eve	Germany	50.0
Bob	Canada	45.0

2. Top Selling Products

Shows units sold and revenue per product. Helps with inventory and marketing decisions.

```
SQL:
SELECT p.name, SUM(oi.qty) AS total_sold, SUM(p.price*oi.qty) AS revenue FROM products p JOIN order_items oi ON p.id=o
```

name	total_sold	revenue
USB Cable	7	70.0
Mouse	5	125.0
Laptop	3	3600.0
Monitor	2	600.0
Keyboard	2	90.0

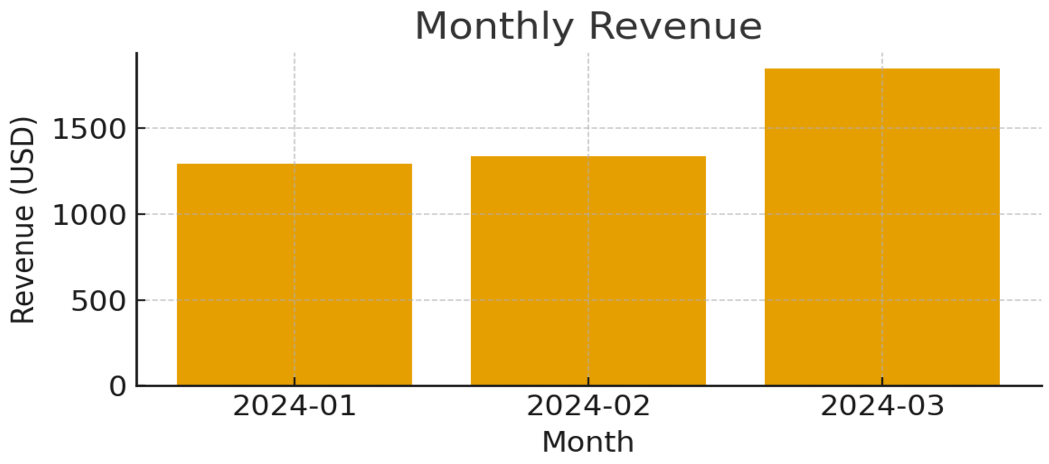


3. Monthly Revenue

Revenue grouped by year-month. Useful to observe seasonality and trends.

```
SQL:
SELECT * FROM monthly_revenue ORDER BY month;
```

month	revenue
2024-01	1295.0
2024-02	1340.0
2024-03	1850.0



4. Orders Above Average Value

Identifies orders whose total exceeds the average order value — useful for defining VIP purchase thresholds.

```
SQL:
SELECT o.id AS order_id, c.name AS customer, SUM(p.price*oi.qty) AS order_value, o.order_date FROM orders o JOIN custo
```

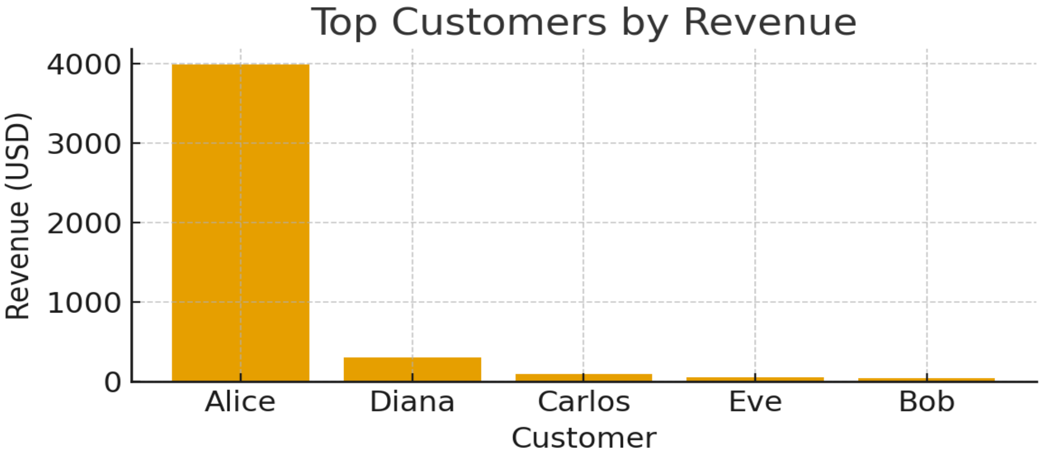
order_id	customer	order_value	order_date
1	Alice	1250.0	2024-01-10
3	Alice	1245.0	2024-02-01
7	Alice	1500.0	2024-03-22

5. Top Customers by Revenue

Highlights the top customers by lifetime revenue — target for retention & upsell.

```
SQL:
SELECT c.name, SUM(p.price*oi.qty) AS revenue FROM customers c JOIN orders o ON c.id=o.customer_id JOIN order_items oi
```

name	revenue
Alice	3995.0
Diana	300.0
Carlos	95.0
Eve	50.0
Bob	45.0



Recommendations & Next Steps

- Consider targeted promotions for top customers. - Reorder high-selling products to avoid stockouts. - Track monthly revenue and investigate sudden drops or spikes. - Create indexes on foreign keys (e.g., orders(customer_id)) for performance.

Appendix: Files included

```
- ecommerce_queries.sql
- total_sales.png
- top_products.png
- usa_customers.png
- orders_above_avg.png
- monthly_revenue.png
- ecommerce_report_professional.pdf
- README.md
```