Amazon Sales & Customer Analytics

Presented By:
Akash Mohan
Aditya Kumar
Kaushiki Sharma

Tools used: SQL, Python

Project Overview

This project analyzes an Amazon-like e-commerce dataset to extract business insights related to sales, customers, products, payments, and inventory using SQL.

Objectives

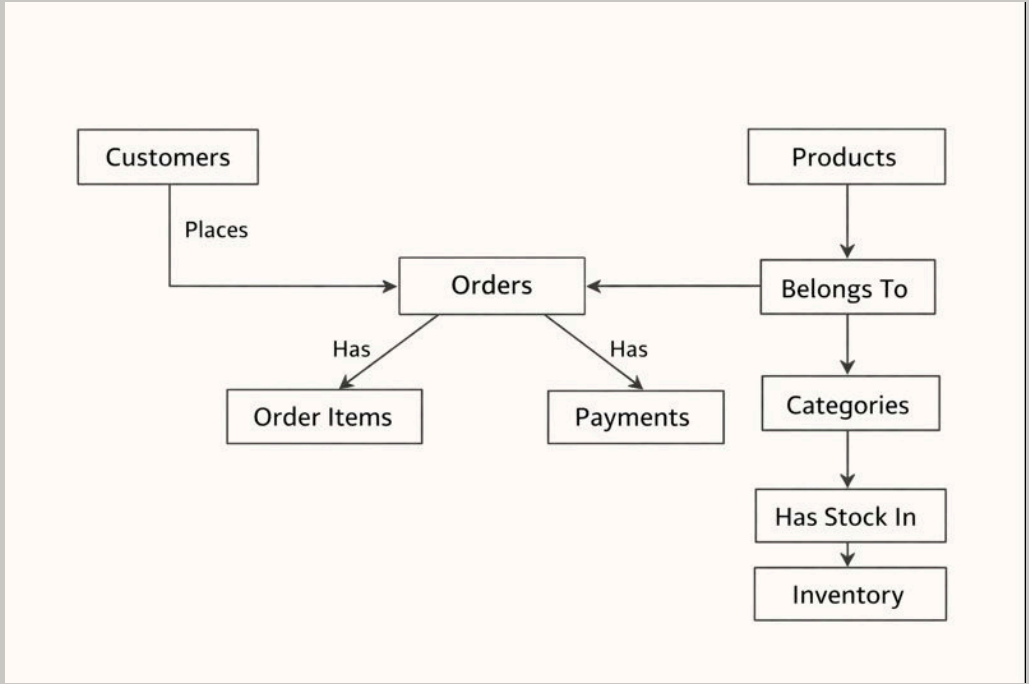
- Identify top products and categories
- Analyze sales and revenue trends
- Study customer behavior and value
- Monitor inventory and payments

Dataset Used

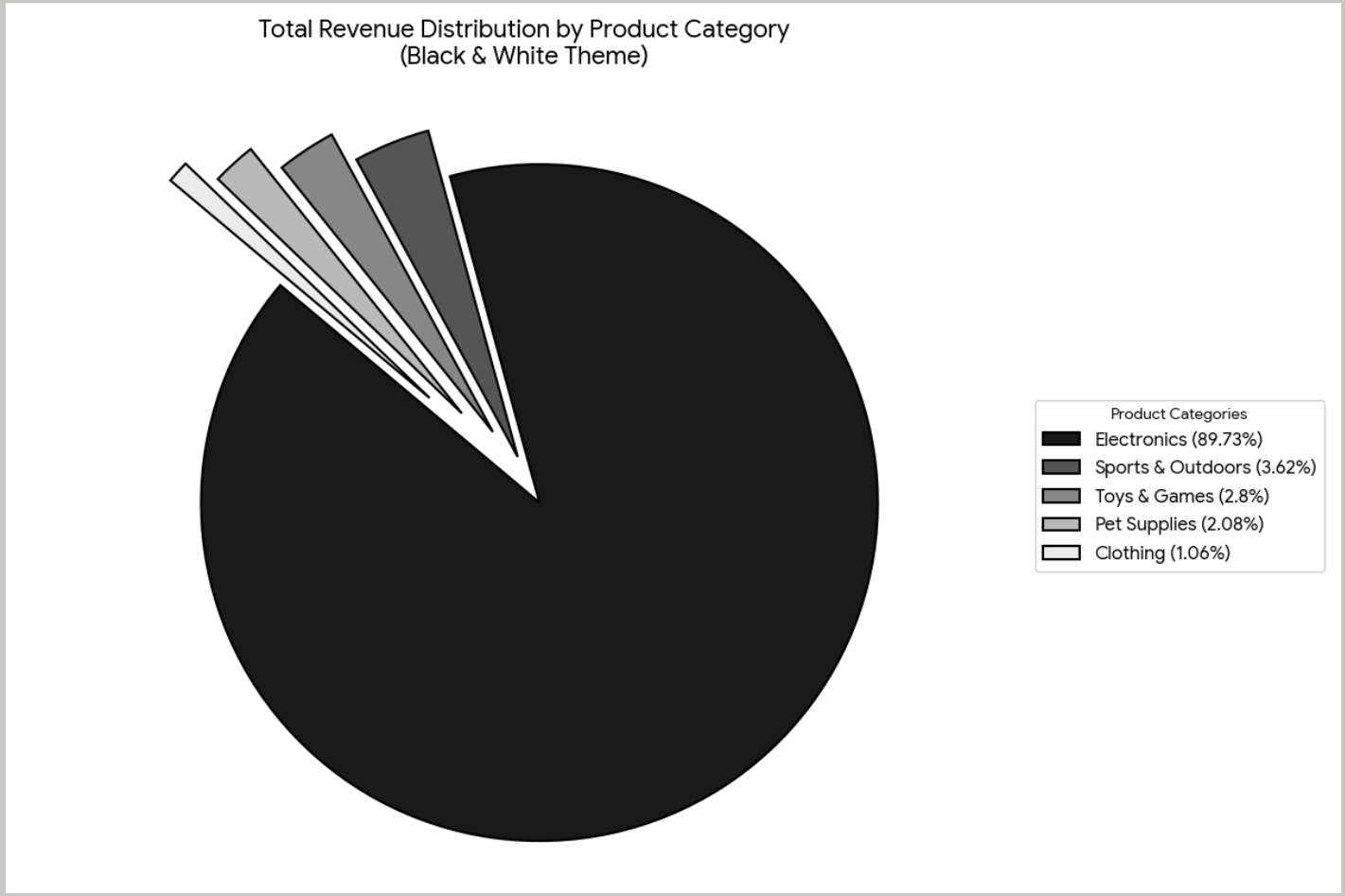
- Customers
- Orders
- Order Items
- Products & Categories
- Payments
- Shipping
- Inventory
- Sellers

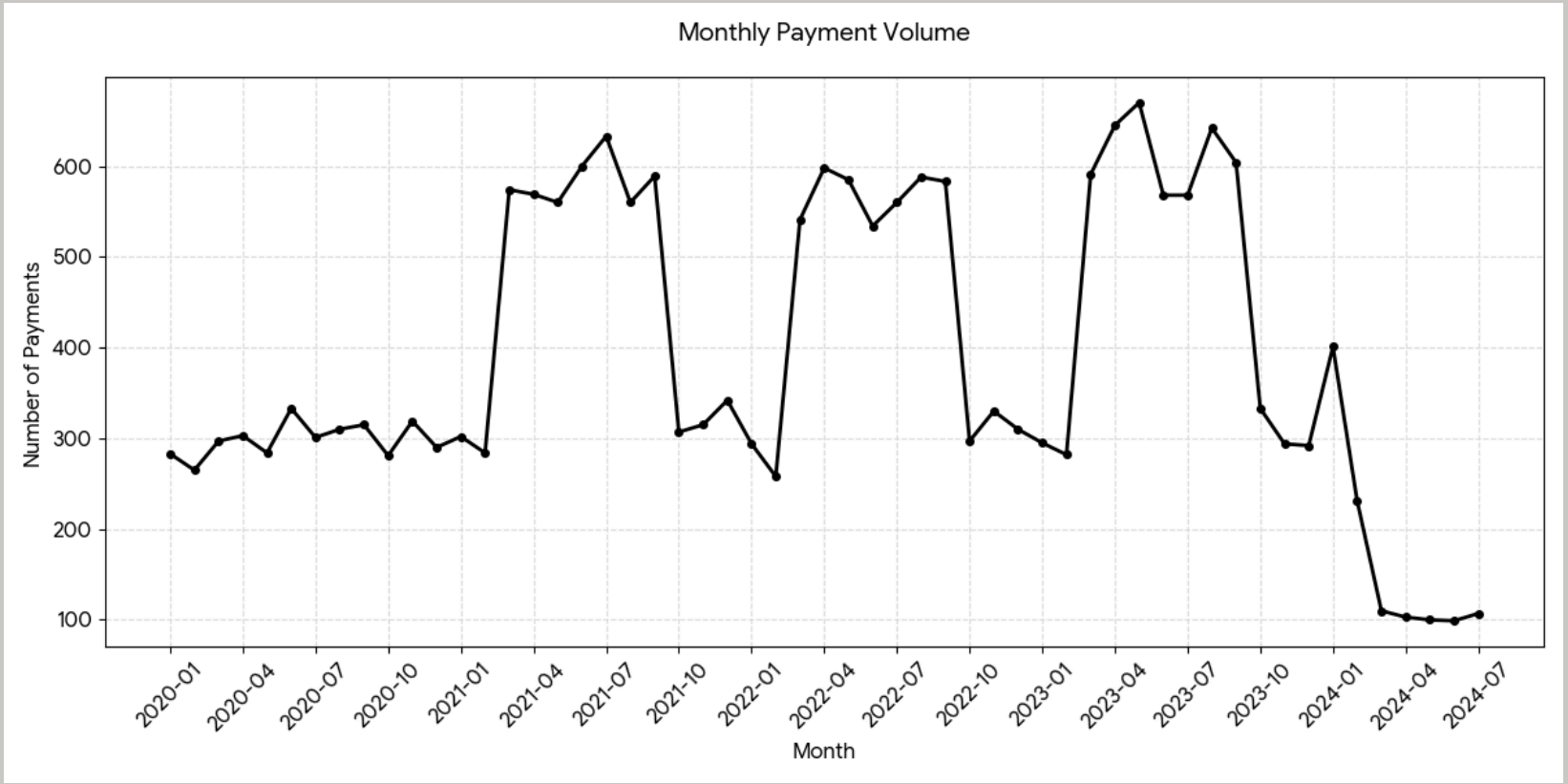
Data Relationships

- Customers → Orders
- Orders → Order Items & Payments
- Products → Categories & Inventory



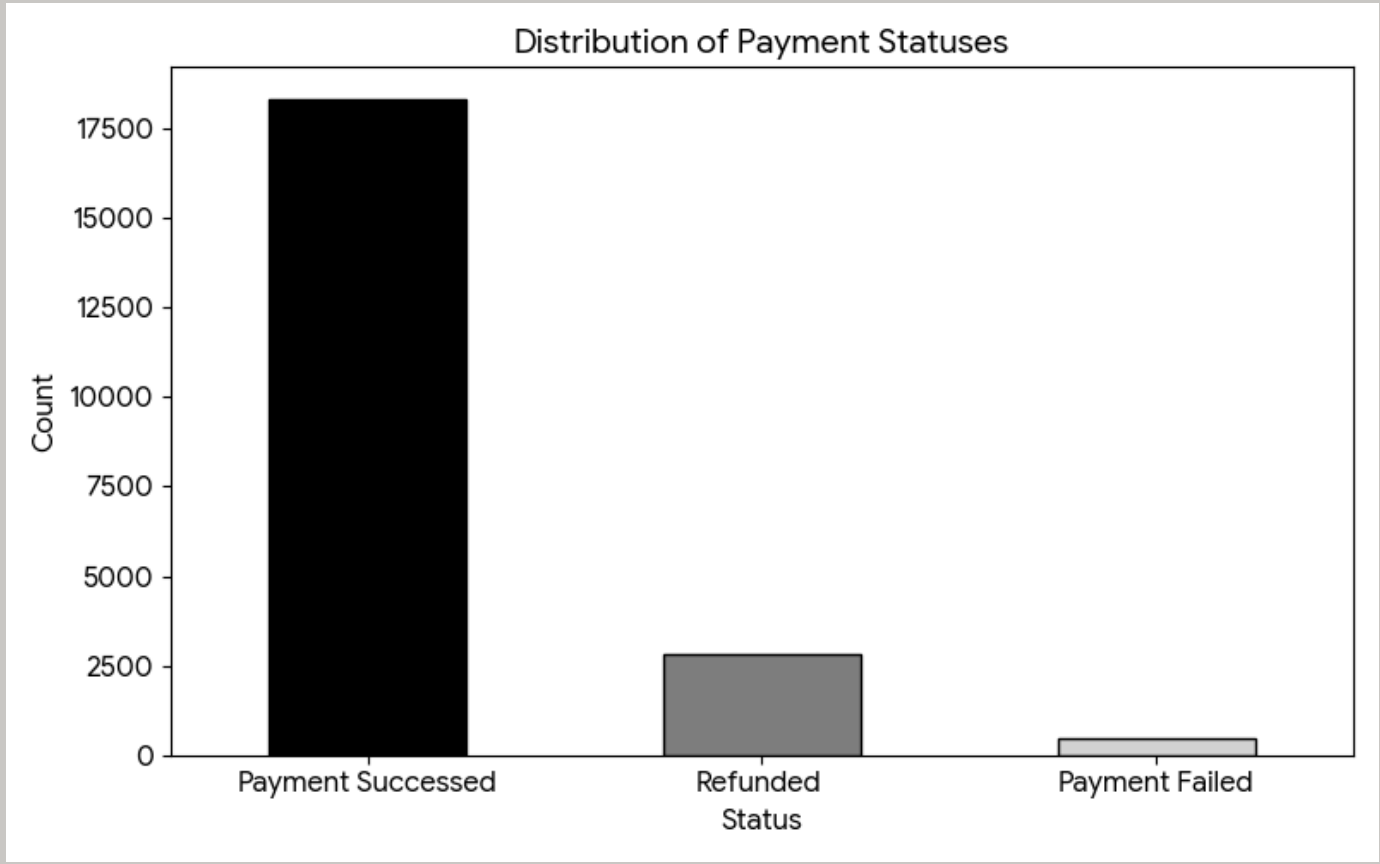
Pie Chart for Product Category





Monthly Payment Volume

This line chart displays the trend of transactions over time. It shows the total number of payments recorded each month, allowing you to identify seasonal peaks and growth in transaction volume.



Distribution of Payment Statuses

This bar chart breaks down the 21,629 records into their respective outcomes.

- Payment Successed: 18,301 transactions
- Refunded: 2,840 transactions
- Payment Failed: 488 transactions

Data Cleaning

- Handled missing values
- Removed duplicates
- Fixed data types
- Ensured table consistency

Renaming Column for Consistency

```sql
ALTER TABLE order_items
RENAME COLUMN `ï»¿order_item_id` TO order_item_id;

SELECT COLUMN_NAME
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'order_items';
```

This SQL command renames the column ï»¿order_item_id to order_item_id in the order_items table to fix encoding issues and maintain consistent column naming. The second query checks the updated column names using INFORMATION_SCHEMA.

Data Type Conversion for Date Columns

```sql
ALTER TABLE shipping
MODIFY COLUMN shipping_date DATE;

ALTER TABLE shipping
MODIFY COLUMN return_date DATE;
```

These queries modify the shipping_date and return_date columns in the shipping table to the DATE data type, enabling accurate date-based analysis and calculations.

**Query the top 10 products by total sales value, including product name,
total quantity sold, and total sales value.**

```sql
-- 1. Query the top 10 products by total sales value, including product name,
-- total quantity sold, and total sales value.
select p.product_name,
sum(o.quantity) as total_quantity_sold,
round(sum(quantity * price_per_unit),2) as total_sales from products p
join order_items o on p.product_id = o.product_id
group by p.product_name
order by total_sales desc limit 10;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

product_name	total_quantity_sold	total_sales
Apple iMac Pro	126	629998.74
Apple iMac 27-Inch Retina	129	232198.71
Canon EOS R5 Mirrorless Camera	57	222299.43
Apple iMac 24-Inch	146	189798.54
Apple MacBook Pro 16-inch	75	187499.25
Dell Alienware Aurora R13	71	177499.29

Find customers who have registered but never placed an order, listing
customer details and the time since their registration.

```sql
-- 5. Find customers who have registered but never placed an order, listing
-- customer details and the time since their registration.

SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    c.state
FROM customers AS c
LEFT JOIN orders AS o
    ON o.customer_id = c.customer_id
WHERE
    o.order_id IS NULL;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name	state
894	Ursula	Lee	Wisconsin
895	Samuel	Clark	Wisconsin
896	Patrick	Scott	Wisconsin
897	Mia	Davis	Wisconsin
898	Jake	Davis	Wisconsin
889	Tara	Sanchez	West Virginia
890	Rachel	Davis	West Virginia
891	Olivia	Adams	West Virginia
892	Leo	Smith	West Virginia

Conclusion

This project involved analyzing an e-commerce relational database using SQL to derive meaningful business insights from transactional data. By working with tables such as customers, orders, order_items, products, and categories, we implemented industry-standard SQL techniques including joins, aggregations, CTEs, and window functions to compute key metrics like top-selling products, category-wise revenue contribution, average order value, and state-level performance. Emphasis was placed on correct aggregation logic, clean query structure, and performance-oriented approaches, resulting in production-ready SQL solutions suitable for real-world data analytics and backend systems.


THANK YOU
FOR YOUR ATTENTION