E-Commerce Sales Analysis – SQL Project

1. Project Overview

The objective of this project is to analyze an e-commerce company's sales data using SQL.
The analysis helps in understanding customer behavior, sales trends, top-performing products, and revenue generation.

This project demonstrates practical usage of SQL concepts such as:

- Data filtering
- Aggregation
- Joins
- Subqueries
- Window functions

---

2. Tools & Technologies Used

- **Database:** MySQL
- **Language:** SQL
- **IDE:** MySQL Workbench
- **Visualization (optional):** Excel / Power BI
- **Documentation:** MS Word

---

3. Database Schema

Tables Used

1. **customers**
   - customer_id
   - customer_name
   - city
   - signup_date

2. **products**
    - product_id
    - product_name
    - category
    - price
3. **orders**
    - order_id
    - customer_id
    - order_date
4. **order_items**
    - order_item_id
    - order_id
    - product_id
    - quantity
    - price
5. **payments**
    - payment_id
    - order_id
    - payment_mode
    - payment_status

---

4. Business Questions & Analysis

1. What is the total sales revenue?

```
SELECT SUM(price * quantity) AS total_revenue

FROM order_items;
```

Insight:
This shows the overall revenue generated by the business.

---

2. Which cities generate the highest revenue?

```
SELECT c.city, SUM(oi.price * oi.quantity) AS total_revenue

FROM customers c

JOIN orders o ON c.customer_id = o.customer_id

JOIN order_items oi ON o.order_id = oi.order_id

GROUP BY c.city

ORDER BY total_revenue DESC;
```

Insight:
Cities with higher revenue indicate strong market demand.

---

3. Top 5 Best-Selling Products

```
SELECT p.product_name, SUM(oi.quantity) AS total_sold

FROM products p

JOIN order_items oi ON p.product_id = oi.product_id

GROUP BY p.product_name

ORDER BY total_sold DESC

LIMIT 5;
```

---

4. Top Customers by Spending

```
SELECT c.customer_name, SUM(oi.price * oi.quantity) AS total_spent

FROM customers c

JOIN orders o ON c.customer_id = o.customer_id

JOIN order_items oi ON o.order_id = oi.order_id

GROUP BY c.customer_name

ORDER BY total_spent DESC;
```

---

5. Monthly Sales Trend

```
SELECT MONTH(o.order_date) AS month,
    SUM(oi.price * oi.quantity) AS total_sales
FROM orders o
JOIN order_items oi ON o.order_id = oi.order_id
GROUP BY MONTH(o.order_date)
ORDER BY month;
```

---

5. Key Insights

- A small number of customers contribute to a large portion of revenue.
- Certain cities dominate total sales.
- Sales show monthly variation indicating seasonal trends.
- Few products generate most of the revenue (Pareto principle).

---

6. Conclusion

This project demonstrates how SQL can be used to:

- Analyze business performance
- Identify sales trends
- Support data-driven decision making

The analysis helps businesses improve marketing strategies, inventory planning, and customer targeting.