SQL Project Report – E-Commerce Data Analysis

Author: Pravin Gurung

Tools Used: MySQL Workbench, Excel, MS Word

This project demonstrates how SQL can be used to analyze real-world e-commerce data to uncover insights about customers, sales trends, product performance, and payment methods.

1. Objective

The main goal of this analysis is to extract valuable insights from an e-commerce database using SQL. The queries focus on customer behavior, sales trends, product categories, seller performance, and payment preferences.

2. Customer Analysis – Revenue by State

```
SELECT
    c.customer_state,
    COUNT(DISTINCT o.customer_id) AS total_customers,
    COUNT(o.order_id) AS total_orders,
    ROUND(SUM(oi.price), 2) AS total_revenue,
    ROUND(AVG(oi.price), 2) AS avg_order_value
FROM customers_dataset c
JOIN orders_dataset o ON c.customer_id = o.customer_id
JOIN order_items_dataset oi ON o.order_id = oi.order_id
WHERE o.order_status = 'delivered'
GROUP BY c.customer_state
ORDER BY total_revenue DESC;
```

Insight: Identifies states with the highest total revenue, customer engagement, and order volume.

3. Top Selling Product Categories

```
SELECT
    p.product_category_name,
    COUNT(oi.order_id) AS total_orders,
    ROUND(SUM(oi.price), 2) AS total_revenue,
    ROUND(AVG(oi.price), 2) AS avg_price
FROM products_dataset p
JOIN order_items_dataset oi ON p.product_id = oi.product_id
JOIN orders_dataset o ON oi.order_id = o.order_id
WHERE o.order_status = 'delivered'
GROUP BY p.product_category_name
ORDER BY total_revenue DESC
LIMIT 15;
```

Insight: Reveals top-performing product categories by total revenue and order volume.

4. Monthly Sales Trend

```
SELECT
    YEAR(order_purchase_timestamp) AS year,
    MONTH(order_purchase_timestamp) AS month,
    COUNT(o.order_id) AS order_count,
    ROUND(SUM(oi.price), 2) AS monthly_revenue
FROM orders_dataset o
JOIN order_items_dataset oi ON o.order_id = oi.order_id
WHERE o.order_status = 'delivered'
GROUP BY year, month
ORDER BY year, month;
```

Insight: Displays monthly sales and revenue trends, helping identify seasonal peaks.

5. Seller Analysis

```
SELECT
    seller_state,
    COUNT(seller_id) AS total_sellers
FROM sellers_dataset
GROUP BY seller_state
ORDER BY total_sellers DESC;
```

```
SELECT
    s.seller_state,
    COUNT(DISTINCT s.seller_id) AS total_sellers,
    COUNT(oi.order_id) AS total_orders
FROM sellers_dataset s
JOIN order_items_dataset oi ON s.seller_id = oi.seller_id
GROUP BY s.seller_state
ORDER BY total_orders DESC
LIMIT 15;
```

```
SELECT
    s.seller_state,
    COUNT(DISTINCT s.seller_id) AS total_sellers,
    COUNT(oi.order_id) AS total_orders,
    ROUND(SUM(oi.price), 2) AS total_revenue
FROM sellers_dataset s
JOIN order_items_dataset oi ON s.seller_id = oi.seller_id
GROUP BY s.seller_state
ORDER BY total_revenue DESC
LIMIT 15;
```

Insight: Evaluates seller distribution, performance, and contribution to total revenue.

6. Payment Method Market Share

```
SELECT
    payment_type,
    total_transactions,
    total_value,
    ROUND((total_transactions * 100.0 / (SELECT COUNT(*) FROM payments_dataset)), 2) AS transaction_share,
    ROUND((total_value * 100.0 / (SELECT SUM(payment_value) FROM payments_dataset)), 2) AS value_share
FROM (
    SELECT
        payment_type,
        COUNT(*) AS total_transactions,
        ROUND(SUM(payment_value), 2) AS total_value
    FROM payments_dataset
    GROUP BY payment_type
) AS payment_summary
ORDER BY value_share DESC;
```

Insight: Calculates the percentage share of each payment method by transaction count and total value.

7. Delivery Time and Review Analysis

```
SELECT
    c.customer_state,
    COUNT(o.order_id) AS total_orders,
    ROUND(AVG(DATEDIFF(o.order_delivered_customer_date, o.order_purchase_timestamp)), 1) AS avg_delivery_days,
    ROUND(AVG(r.review_score), 2) AS avg_rating
FROM orders_dataset o
JOIN customers_dataset c ON o.customer_id = c.customer_id
LEFT JOIN reviews_dataset r ON o.order_id = r.order_id
WHERE o.order_status = 'delivered'
GROUP BY c.customer_state
ORDER BY total_orders DESC;
```

Insight: Compares delivery time and customer satisfaction levels across states.

8. Customer Review Breakdown

```
SELECT
    review_score,
    COUNT(*) AS number_of_reviews,
    ROUND((COUNT(*) * 100.0 / (SELECT COUNT(*) FROM reviews_dataset)), 2) AS percentage
FROM reviews_dataset
GROUP BY review_score
ORDER BY review_score DESC;
```

Insight: Provides distribution of review scores, identifying overall customer satisfaction levels.

9. Testing Views

```
SELECT * FROM customer_analysis ORDER BY revenue DESC LIMIT 10;
SELECT * FROM product_analysis ORDER BY revenue DESC LIMIT 10;
SELECT * FROM monthly_sales ORDER BY year, month LIMIT 10;
```

Insight: Ensures analytical SQL views produce accurate aggregated data for dashboards.

10. Conclusion

This SQL project showcases the ability to derive actionable insights from complex e-commerce data. It demonstrates SQL proficiency in joins, aggregations, and analytical functions while connecting data from multiple tables. These insights can help businesses make data-driven decisions regarding customer engagement, product strategy, and operational efficiency.