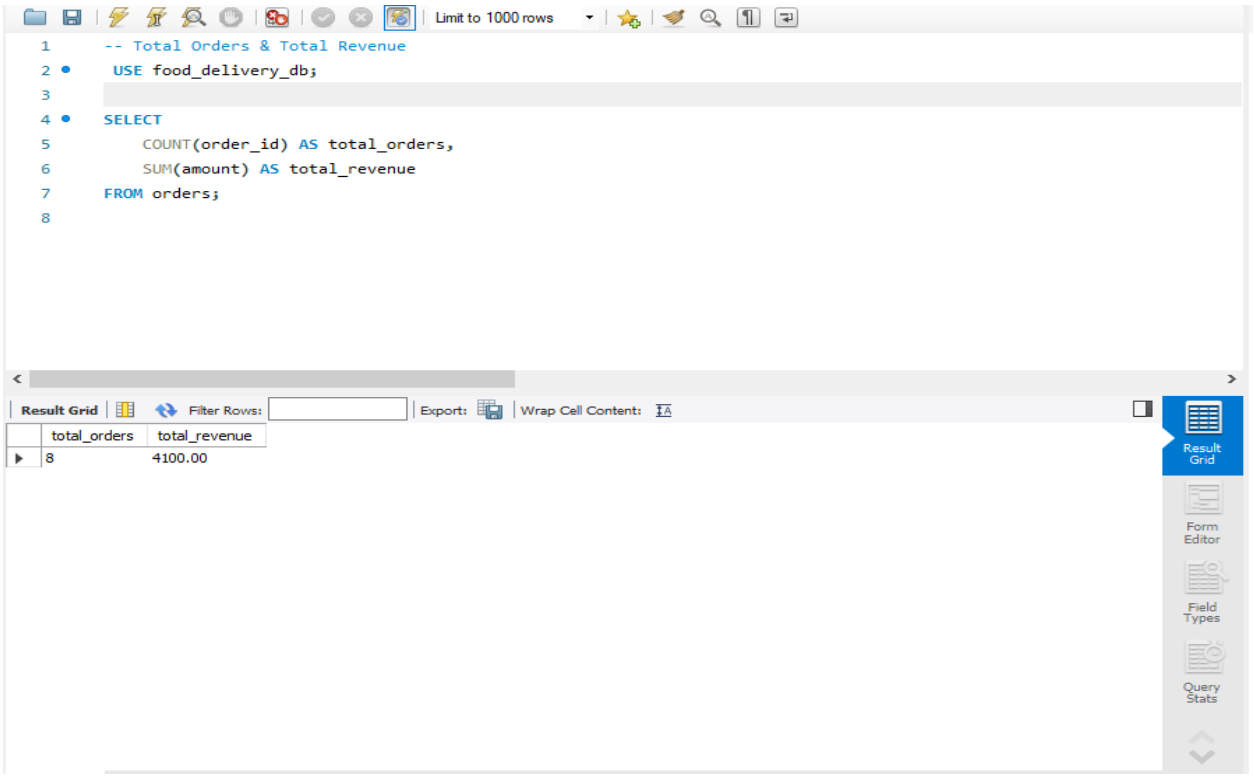
Online Food Delivery SQL Data Analysis

Project Objective

This project analyzes online food delivery data using SQL to understand customer behavior, order trends, revenue distribution, and repeat customers for business decision-making.

SQL Query Results & Analysis Screenshots

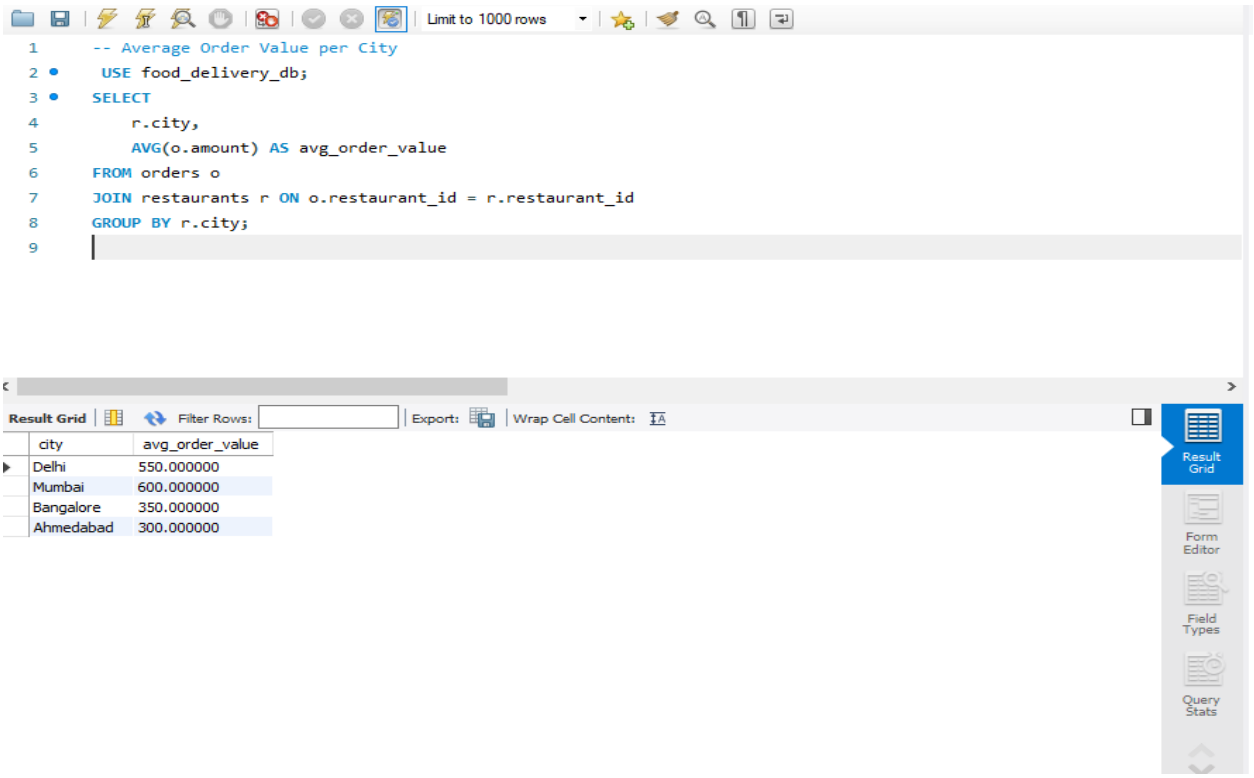


```sql
-- Total Orders & Total Revenue
 USE food_delivery_db;

SELECT
    COUNT(order_id) AS total_orders,
    SUM(amount) AS total_revenue
FROM orders;
```

Result Grid

total_orders	total_revenue
8	4100.00



```sql
-- Average Order Value per City
 USE food_delivery_db;
SELECT
    r.city,
    AVG(o.amount) AS avg_order_value
FROM orders o
JOIN restaurants r ON o.restaurant_id = r.restaurant_id
GROUP BY r.city;
```

Result Grid

city	avg_order_value
Delhi	550.000000
Mumbai	600.000000
Bangalore	350.000000
Ahmedabad	300.000000

```sql
-- Top Customers Analysis
 USE food_delivery_db;
SELECT
    c.customer_name,
    SUM(o.amount) AS total_spent
FROM customers c
JOIN orders o
    ON c.customer_id = o.customer_id
GROUP BY c.customer_name
ORDER BY total_spent DESC
LIMIT 5;
```

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

customer_name	total_spent
Riya Singh	1250.00
Aman Sharma	950.00
Neha Gupta	700.00
Pooja Mehta	550.00
Rahul Verma	350.00

```sql
--  Monthly Revenue Trend
 USE food_delivery_db;
SELECT
    DATE_FORMAT(order_date, '%Y-%m') AS month,
    SUM(amount) AS monthly_revenue
FROM orders
GROUP BY month
ORDER BY month;
```

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

month	monthly_revenue
2024-01	4100.00

```sql
-- Repeat Customers Analysis
 USE food_delivery_db;
SELECT
    c.customer_name,
    COUNT(o.order_id) AS total_orders
FROM customers c
JOIN orders o
    ON c.customer_id = o.customer_id
GROUP BY c.customer_name
HAVING COUNT(o.order_id) > 1;
```

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

customer_name	total_orders
Aman Sharma	2
Riya Singh	2

```sql
-- Customer who Spent the Most
 USE food_delivery_db;
SELECT
    c.customer_name,
    SUM(o.amount) AS total_spent
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_name
ORDER BY total_spent DESC
LIMIT 1;
```

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

customer_name	total_spent
Riya Singh	1250.00

```sql
-- Top Restaurants by Revenue
 USE food_delivery_db;
SELECT
    r.restaurant_name,
    SUM(o.amount) AS total_revenue
FROM restaurants r
JOIN orders o
    ON r.restaurant_id = o.restaurant_id
GROUP BY r.restaurant_name
ORDER BY total_revenue DESC
LIMIT 5;
```

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

restaurant_name	total_revenue
Food Villa	1800.00
Spicy Hub	1150.00
Tandoori Treat	500.00
South Delight	350.00
Street Eats	300.00

Result Grid
Form Editor
Field Types
Query Stats

Key Insights

- Mumbai city has the highest Average Order Value.
- Top customers contribute significantly to total revenue.
- Certain restaurants consistently generate higher revenue.
- Repeat customers indicate strong platform loyalty.
- Monthly revenue trends show stable performance.

Conclusion

This project demonstrates hands-on SQL skills including joins, aggregations, grouping, and real-world business insight extraction, making it suitable for Data Analyst and SQL Analyst roles.