Datapencil

SQL Project Report

Online Food Delivery

Advance Level

KODAVALLURI DINESH SAI

kodavalluridinesh60 @gmail.oom

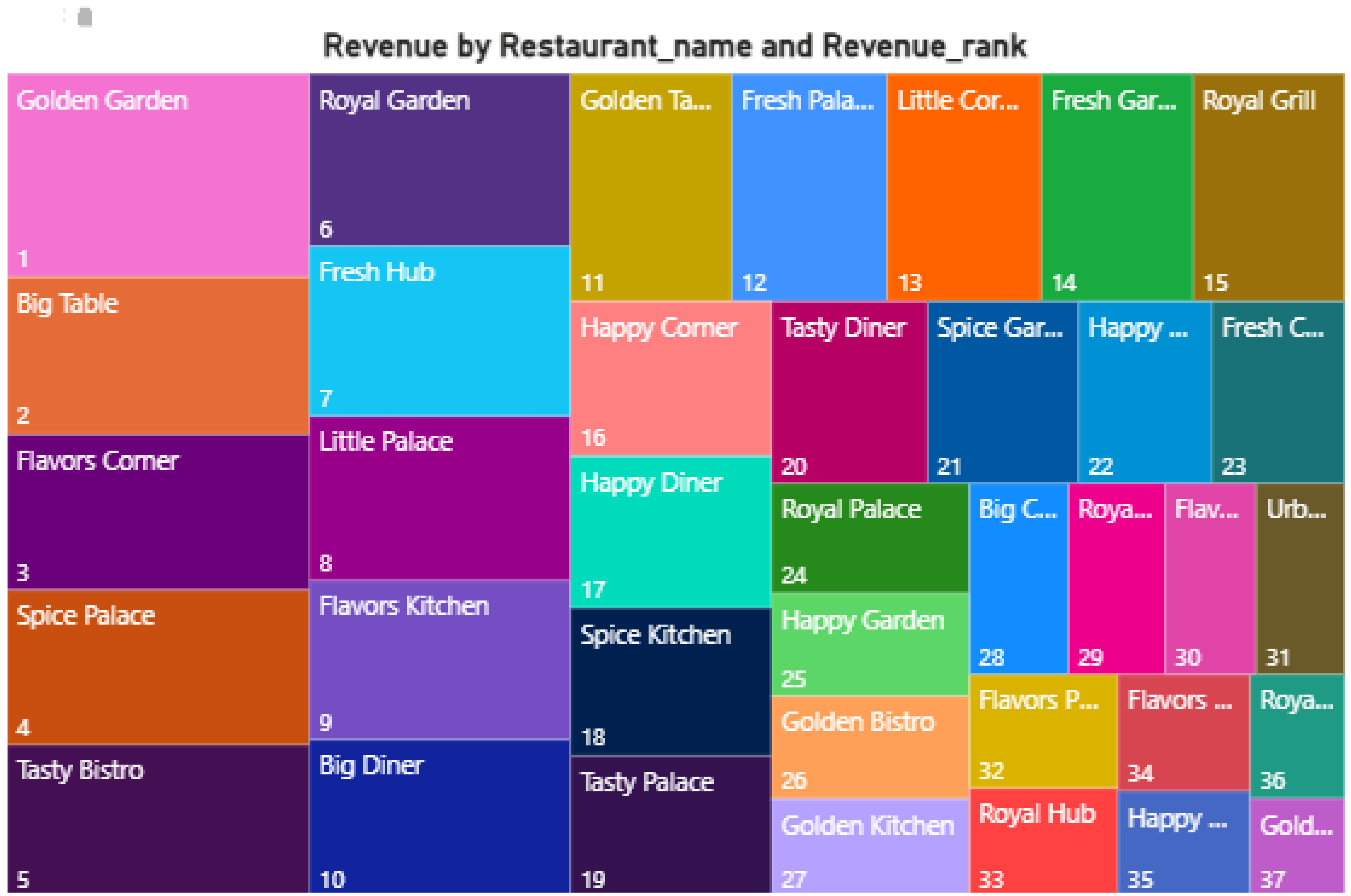
What the project is about

This Online Food Delivery Project is built around a database that stores information about customers, restaurants, menu items, and orders. Customers can browse restaurants, place orders, and choose items from the menu. Each order links the customer, restaurant, and menu items, along with quantity and price, which helps track revenue, order trends, and customer behavior. Using SQL queries, we can analyze the data to find top customers, popular items, revenue per restaurant, and customer activity. The project helps simulate how real-world food delivery platforms like Swiggy or Zomato manage and analyze their data for better decision-making.

Problem Statements

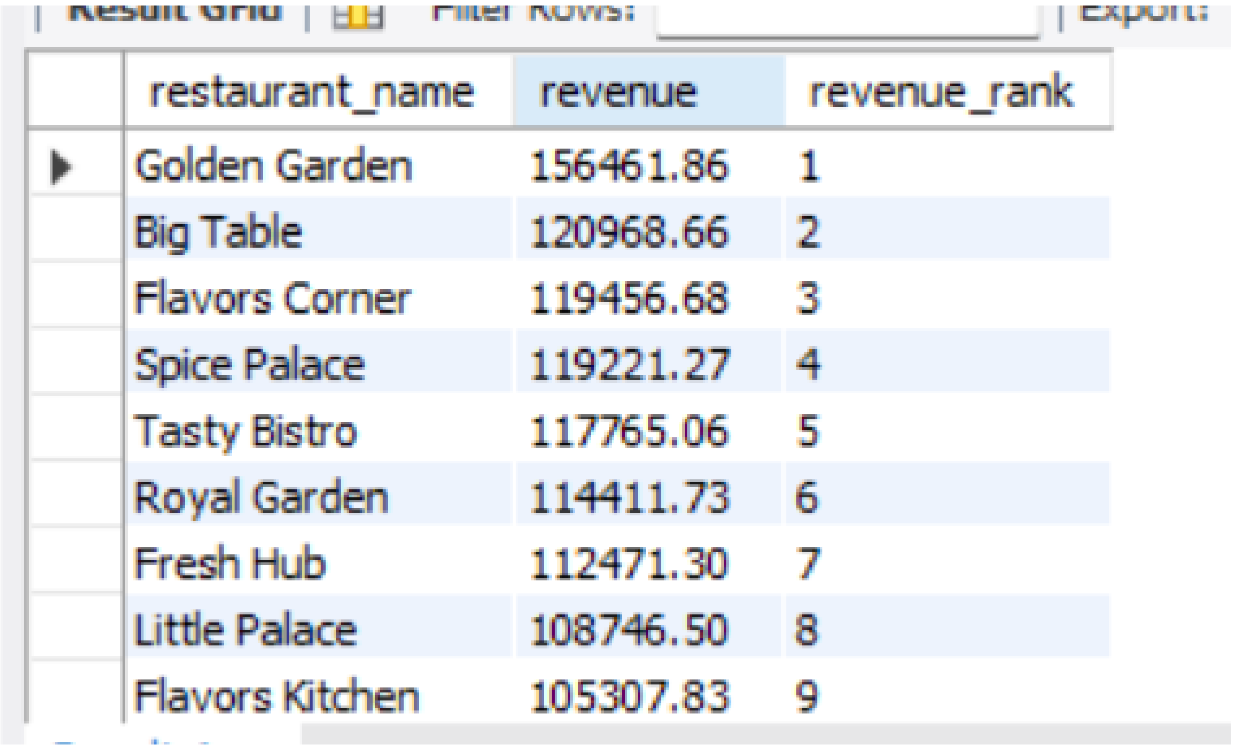
1. Which restaurants are generating the highest revenue and their rank?
2. Find all customers who placed orders from more than one restaurant?
3. Which customers placed more than 3 orders?
4. What is the average menu item price for each restaurant?
5. find the total amount spent by customers in each city?
6. find the 3 most frequently ordered menu items?
7. find the top 3 restaurants that generated the highest revenue?
8. Retrieve the top 5 most expensive menu items across all restaurants?
9. find out how many items were sold per day?
10. Analyze restaurant performance by calculating the total orders, total revenue, and average revenue per order for each restaurant?

Queries & Methods



Q1. Which restaurants are generating the highest revenue and their rank?

```
WITH restaurant_revenue AS (
SELECT r.restaurant_name,SUM(od.quantity * m.price) AS revenue
FROM restaurant r
JOIN orders o ON r.restaurant_id = o.restaurant_id
JOIN order_details od ON o.order_id = od.order_id
JOIN menu_item m ON od.item_id = m.item_id
GROUP BY r.restaurant_name
)SELECT restaurant_name,revenue,RANK() OVER (ORDER BY revenue DESC) AS revenue_rank
FROM restaurant_revenue;
```

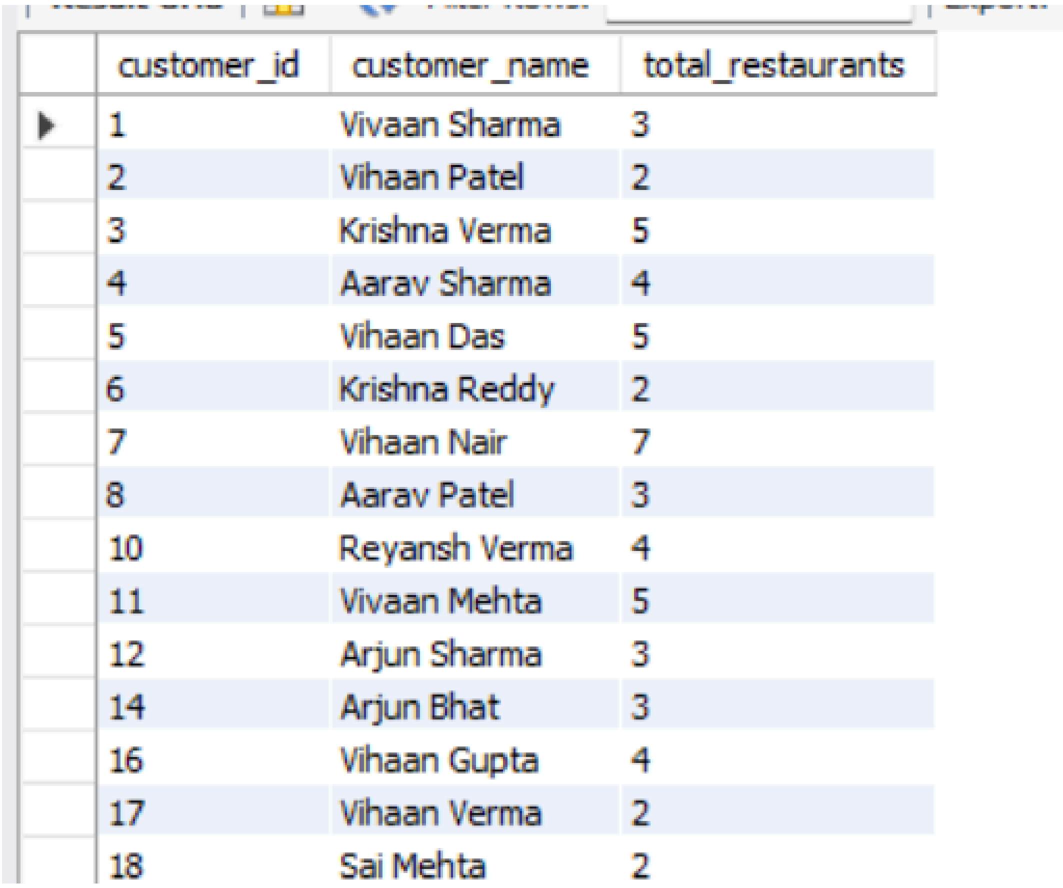


restaurant_name	revenue	revenue_rank
Golden Garden	156461.86	1
Big Table	120968.66	2
Flavors Corner	119456.68	3
Spice Palace	119221.27	4
Tasty Bistro	117765.06	5
Royal Garden	114411.73	6
Fresh Hub	112471.30	7
Little Palace	108746.50	8
Flavors Kitchen	105307.83	9

Queries & Methods

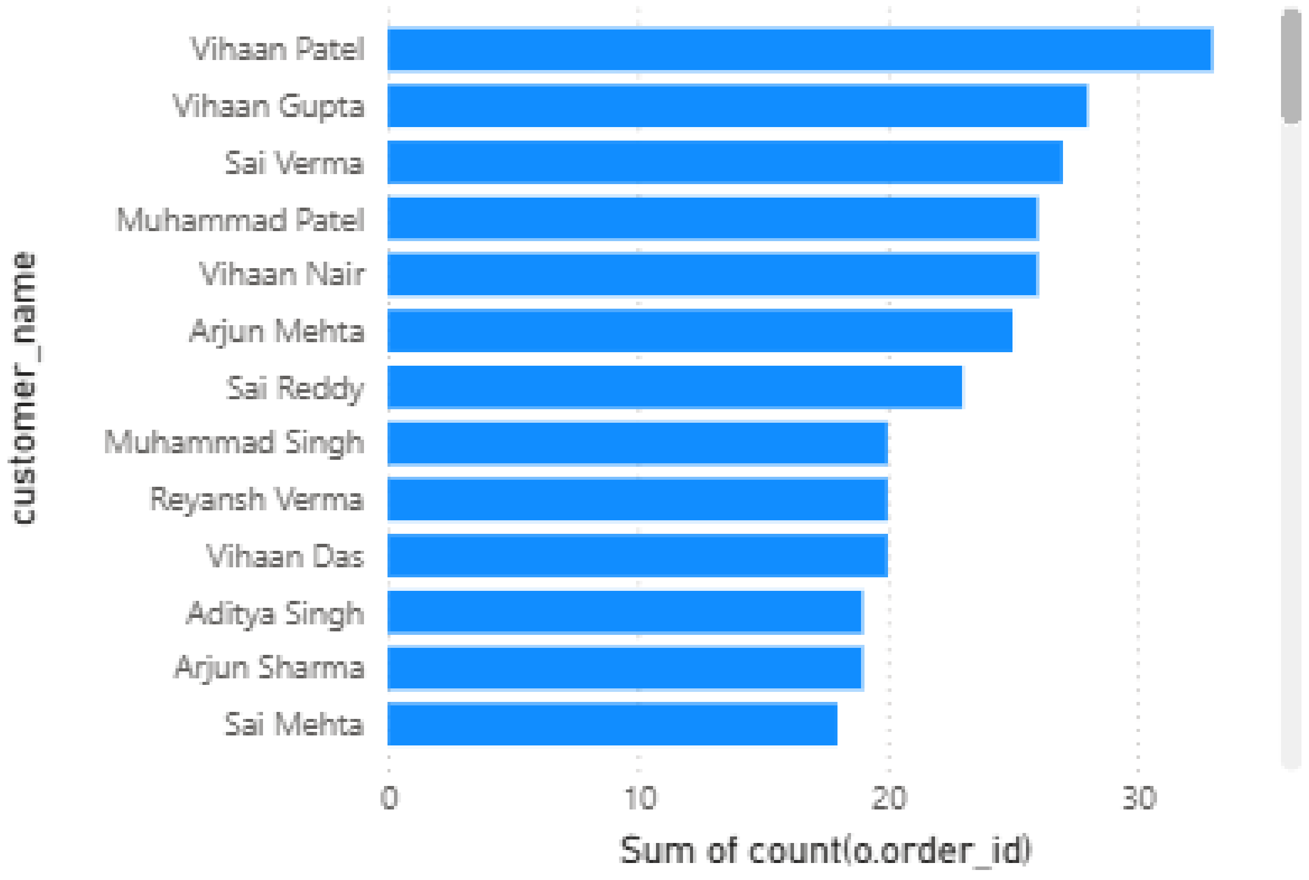
2. Find all customers who placed orders from more than one restaurant?

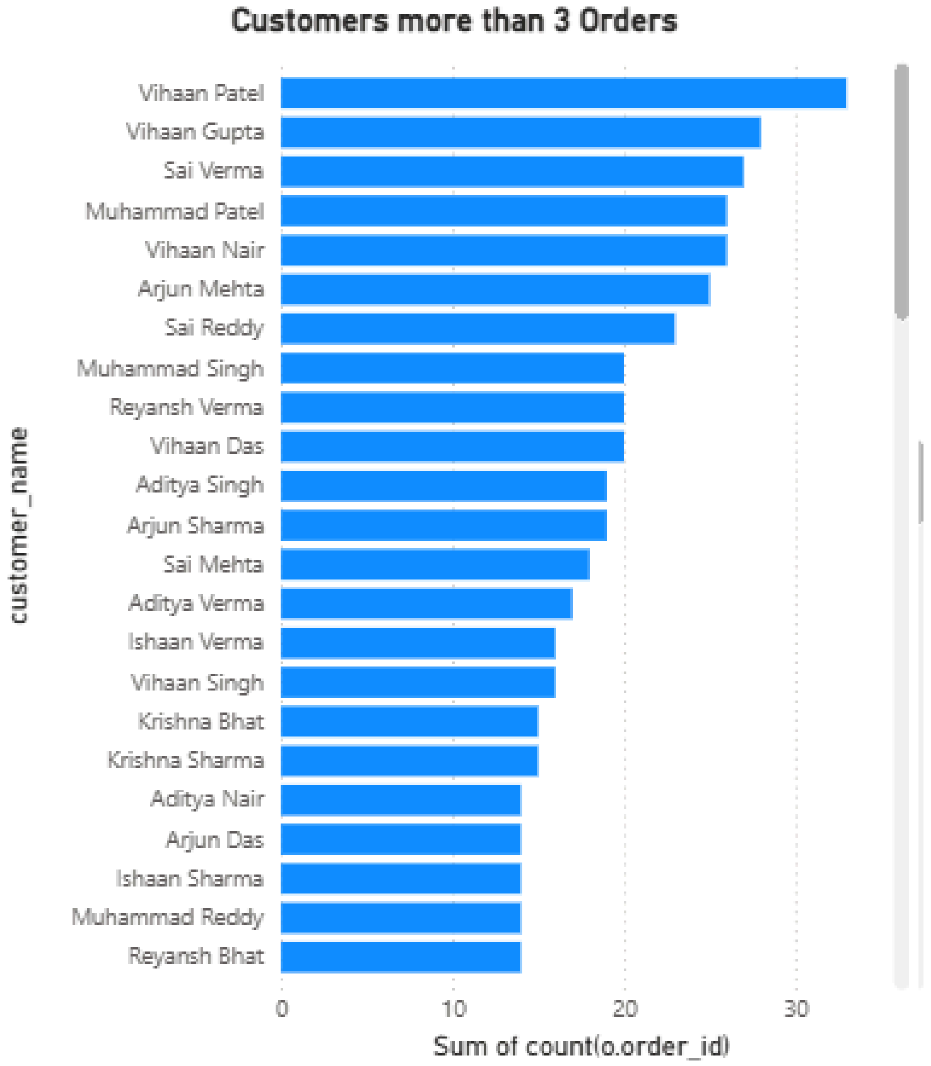
```
SELECT
c.customer_id,
c.customer_name,
COUNT(DISTINCT o.restaurant_id) AS total_restaurants
FROM customers c
JOIN orders o
ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.customer_name
HAVING COUNT(DISTINCT o.restaurant_id) > 1;
```



customer_id	customer_name	total_restaurants
1	Vivaan Sharma	3
2	Vihaan Patel	2
3	Krishna Verma	5
4	Aarav Sharma	4
5	Vihaan Das	5
6	Krishna Reddy	2
7	Vihaan Nair	7
8	Aarav Patel	3
10	Reyansh Verma	4
11	Vivaan Mehta	5
12	Arjun Sharma	3
14	Arjun Bhat	3
16	Vihaan Gupta	4
17	Vihaan Verma	2
18	Sai Mehta	2

Customers who placed Order more 1 Restaurant

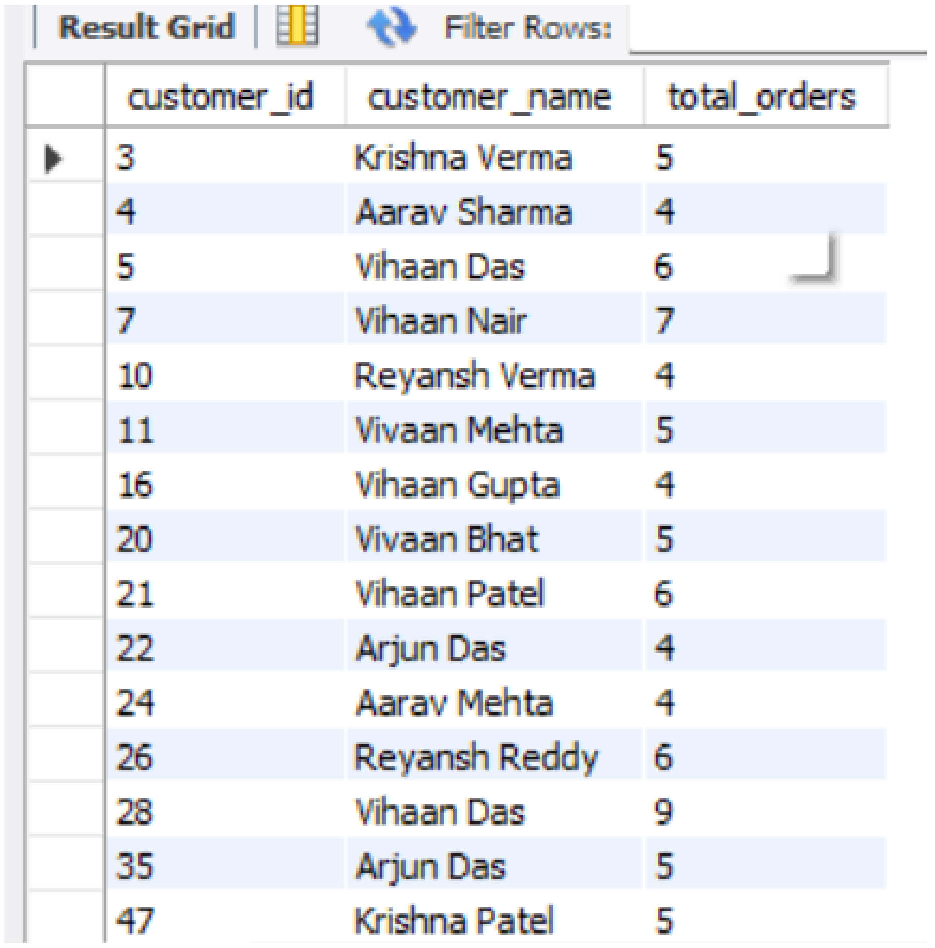




Queries & Methods

3. Which customers placed more than 3 orders?

```
SELECT
c.customer_id,
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_id, c.customer_name
HAVING COUNT(o.order_id) > 3;
```

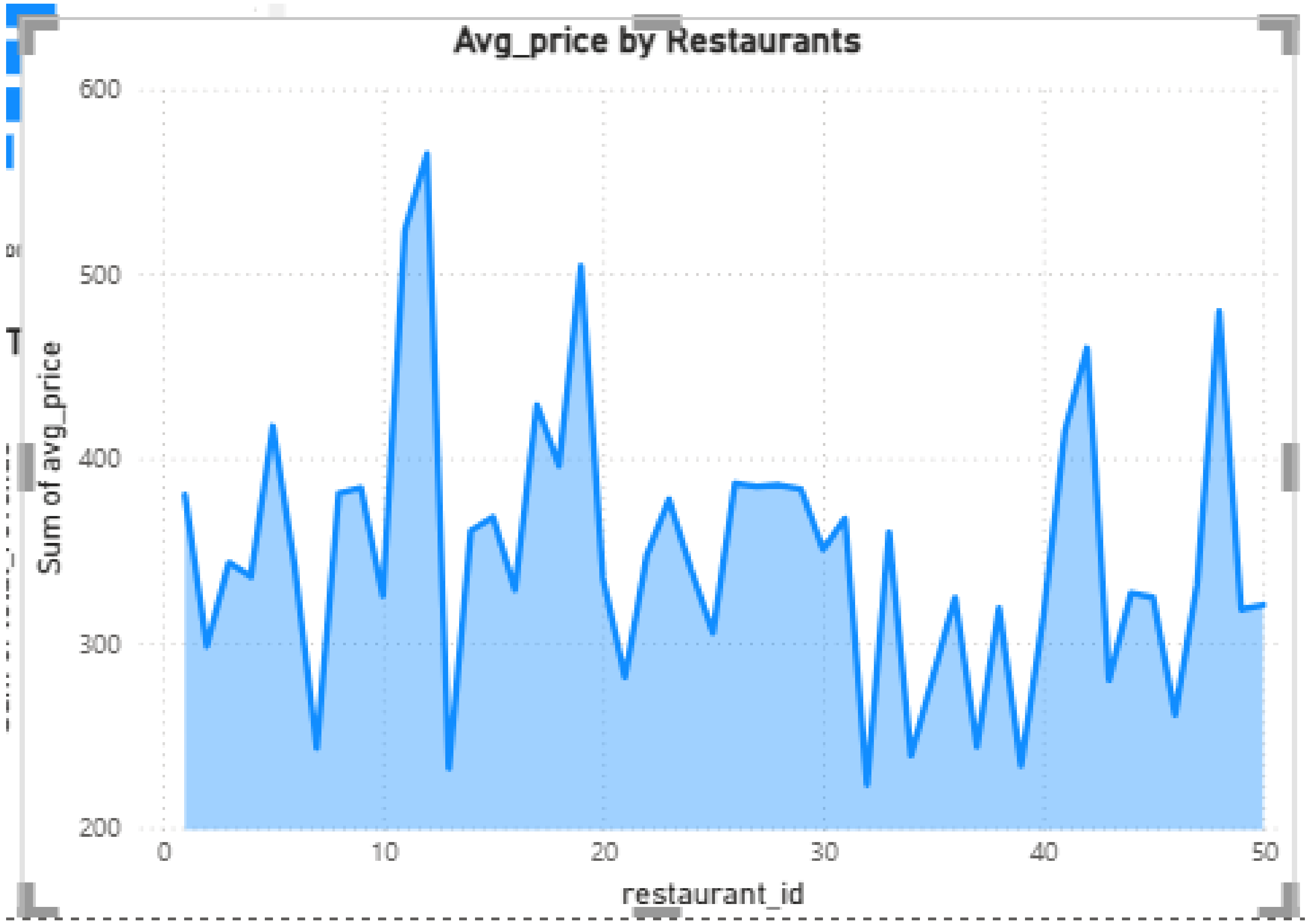


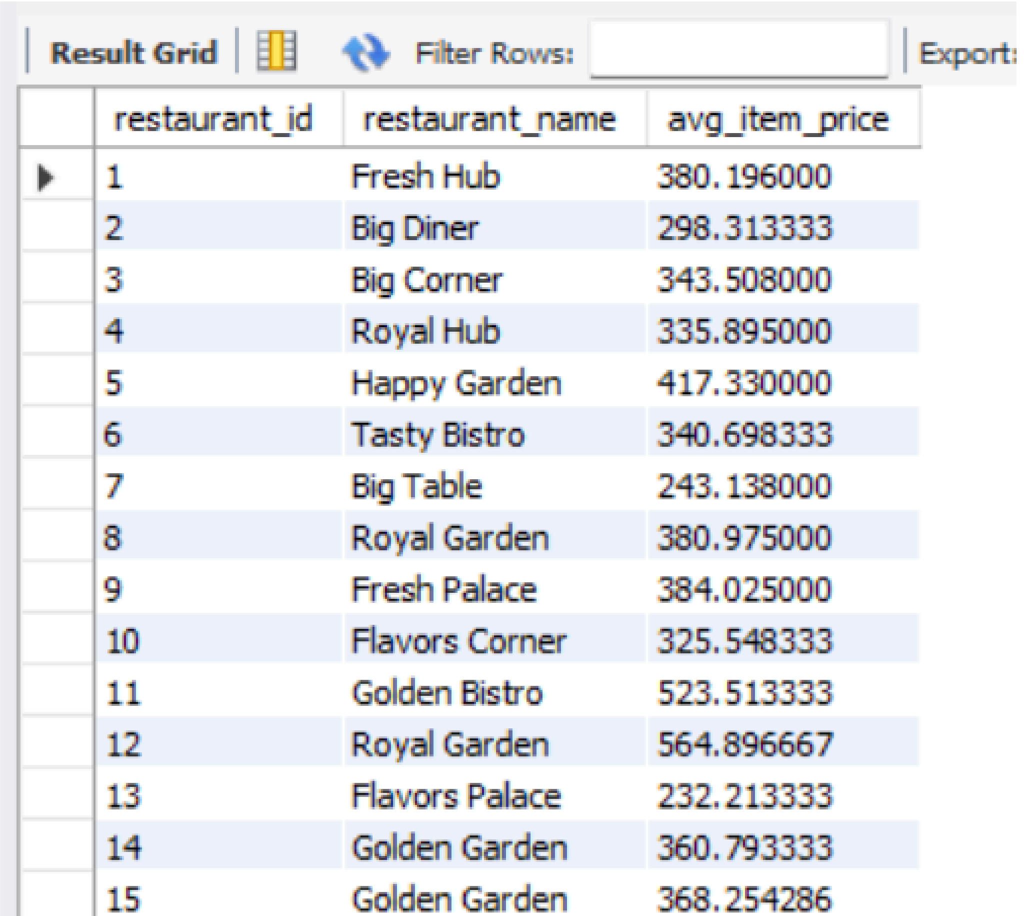
customer_id	customer_name	total_orders
3	Krishna Verma	5
4	Aarav Sharma	4
5	Vihaan Das	6
7	Vihaan Nair	7
10	Reyansh Verma	4
11	Vivaan Mehta	5
16	Vihaan Gupta	4
20	Vivaan Bhat	5
21	Vihaan Patel	6
22	Arjun Das	4
24	Aarav Mehta	4
26	Reyansh Reddy	6
28	Vihaan Das	9
35	Arjun Das	5
47	Krishna Patel	5

Queries & Methods

4. What is the average menu item price for each restaurant?

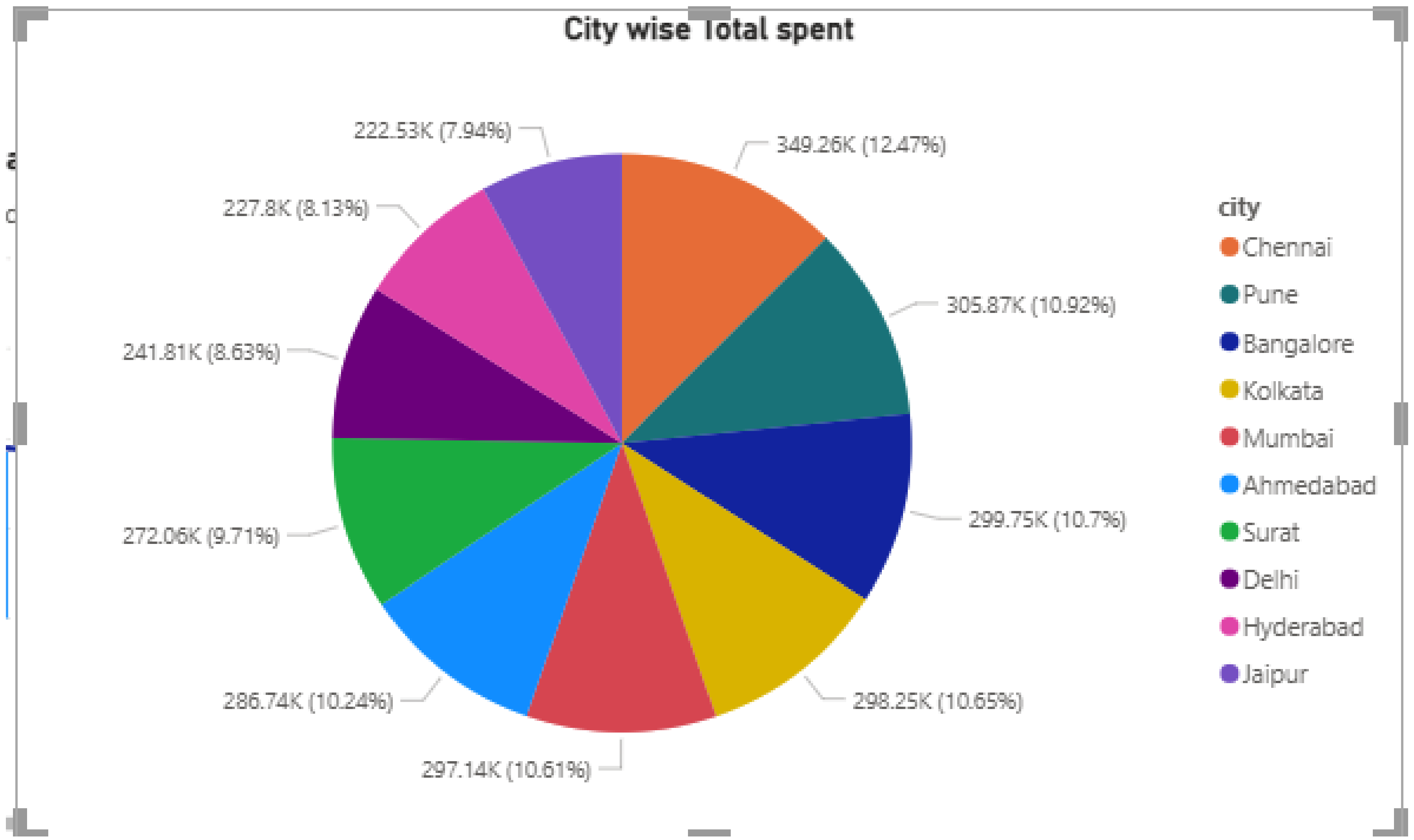
```
SELECT
r.restaurant_id,
r.restaurant_name,
AVG(m.price) AS avg_item_price
FROM restaurant r
JOIN menu_item m
ON r.restaurant_id = m.restaurant_id
GROUP BY r.restaurant_id, r.restaurant_name;
```





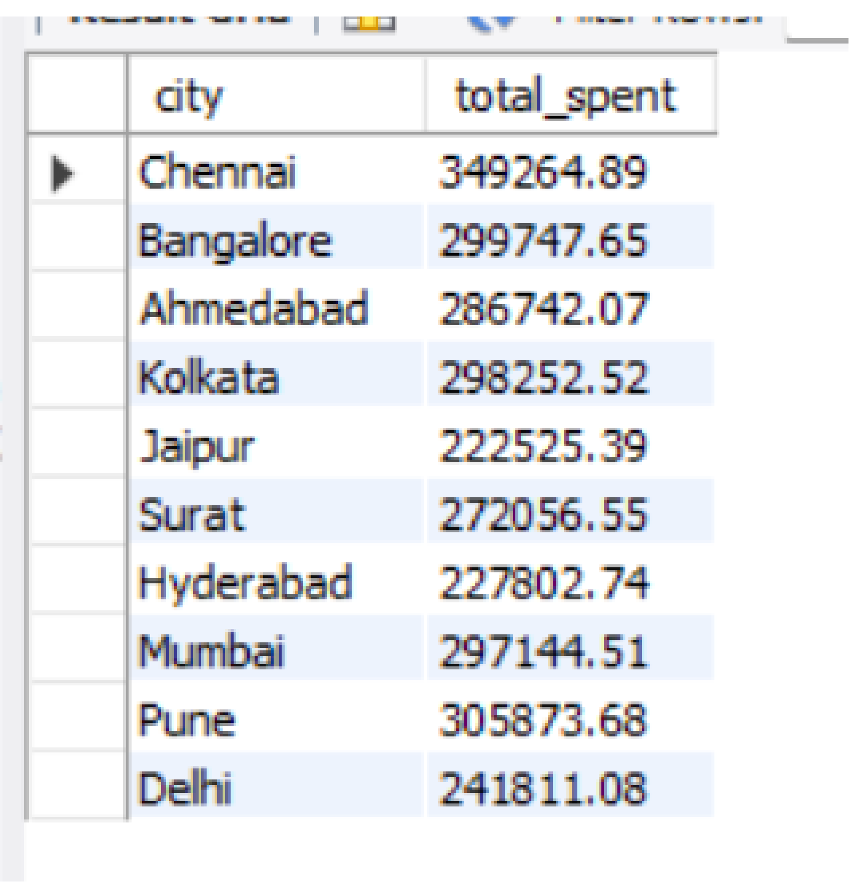
restaurant_id	restaurant_name	avg_item_price
1	Fresh Hub	380.196000
2	Big Diner	298.313333
3	Big Corner	343.508000
4	Royal Hub	335.895000
5	Happy Garden	417.330000
6	Tasty Bistro	340.698333
7	Big Table	243.138000
8	Royal Garden	380.975000
9	Fresh Palace	384.025000
10	Flavors Corner	325.548333
11	Golden Bistro	523.513333
12	Royal Garden	564.896667
13	Flavors Palace	232.213333
14	Golden Garden	360.793333
15	Golden Garden	368.254286

Queries & Methods



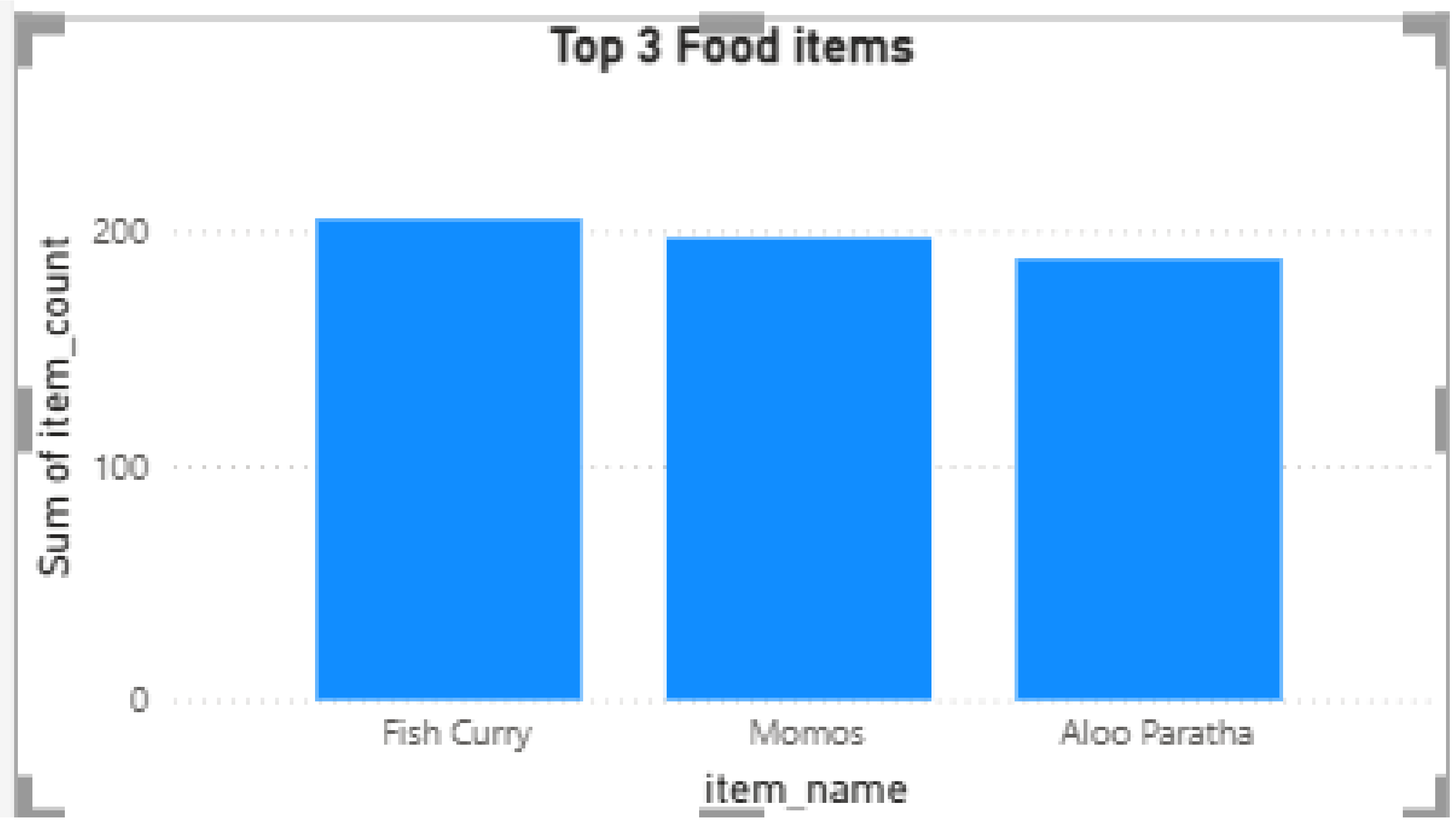
5. find the total amount spent by customers in each city?

```
SELECT
c.city,
SUM(m.price * od.quantity) AS total_spent
FROM customers c
JOIN orders o
ON c.customer_id = o.customer_id
JOIN order_details od
ON o.order_id = od.order_id
JOIN menu_item m
ON od.item_id = m.item_id
GROUP BY c.city;
```



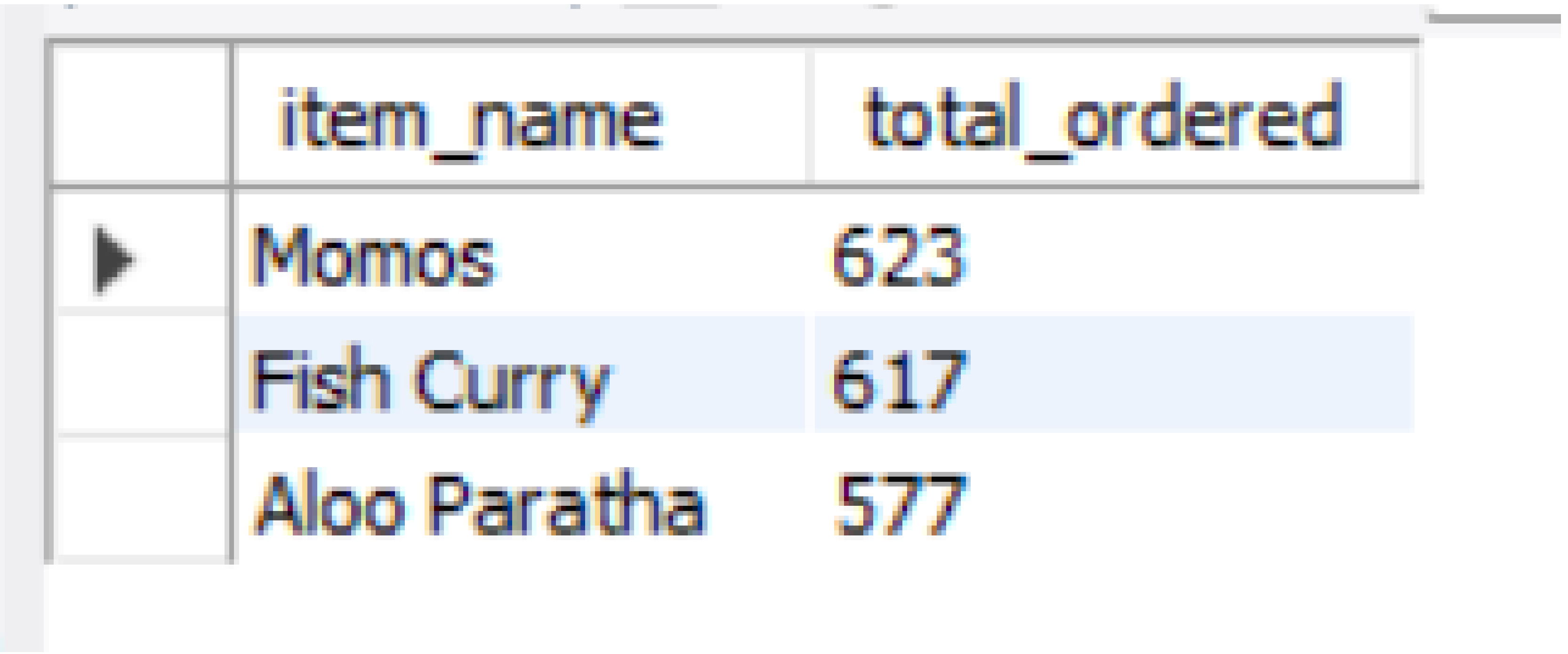
city	total_spent
Chennai	349264.89
Bangalore	299747.65
Ahmedabad	286742.07
Kolkata	298252.52
Jaipur	222525.39
Surat	272056.55
Hyderabad	227802.74
Mumbai	297144.51
Pune	305873.68
Delhi	241811.08

Queries & Methods



6. find the 3 most frequently ordered menu items?

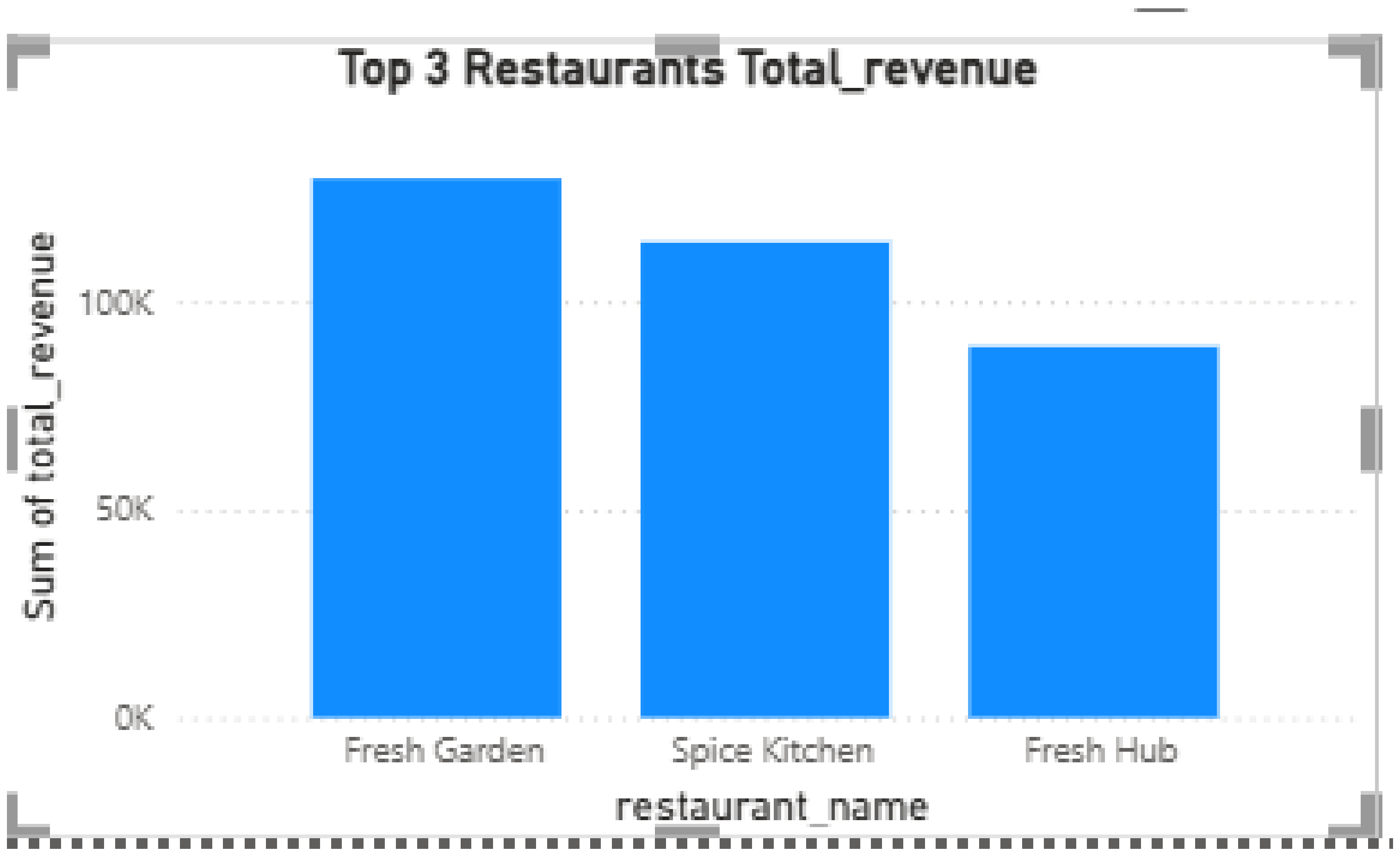
```
SELECT
m.item_name,
SUM(od.quantity) AS total_ordered
FROM order_details od
JOIN menu_item m
ON od.item_id = m.item_id
GROUP BY m.item_name
ORDER BY total_ordered DESC
LIMIT 3;
```

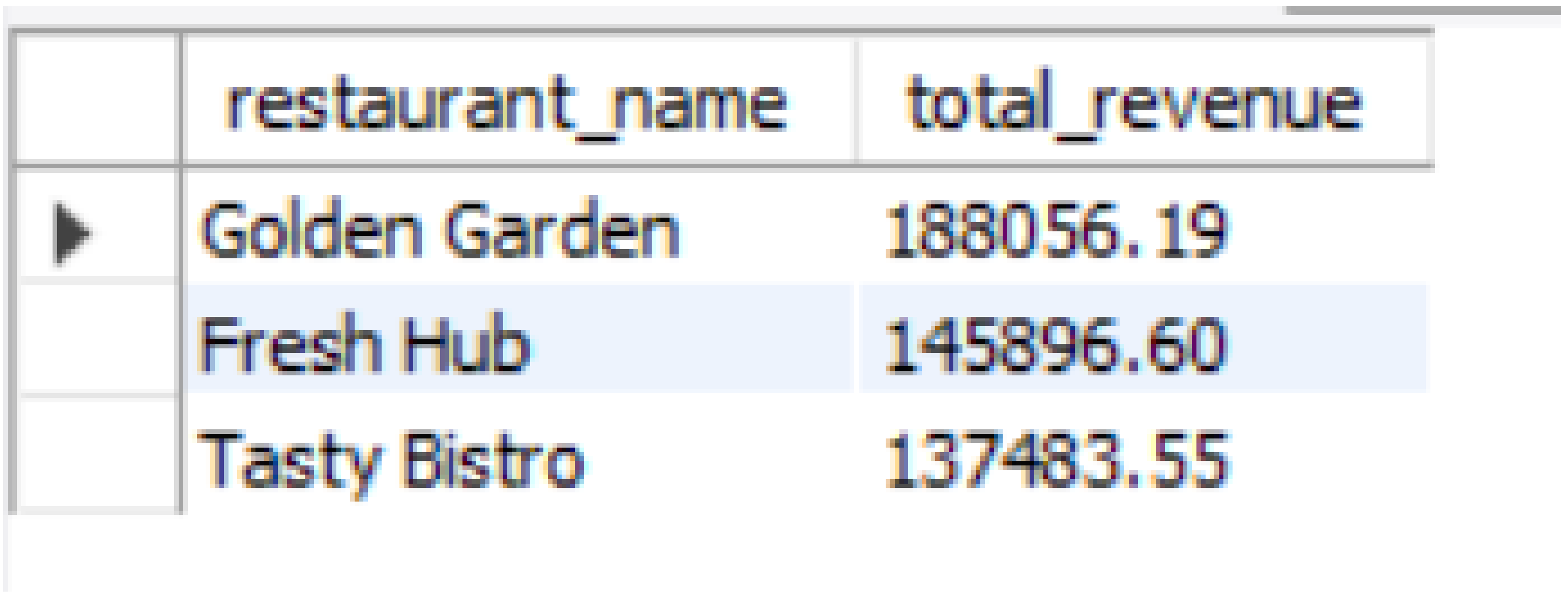


	item_name	total_ordered
▶	Momos	623
	Fish Curry	617
	Aloo Paratha	577

Queries & Methods

7. find the top 3 restaurants that generated the highest revenue?



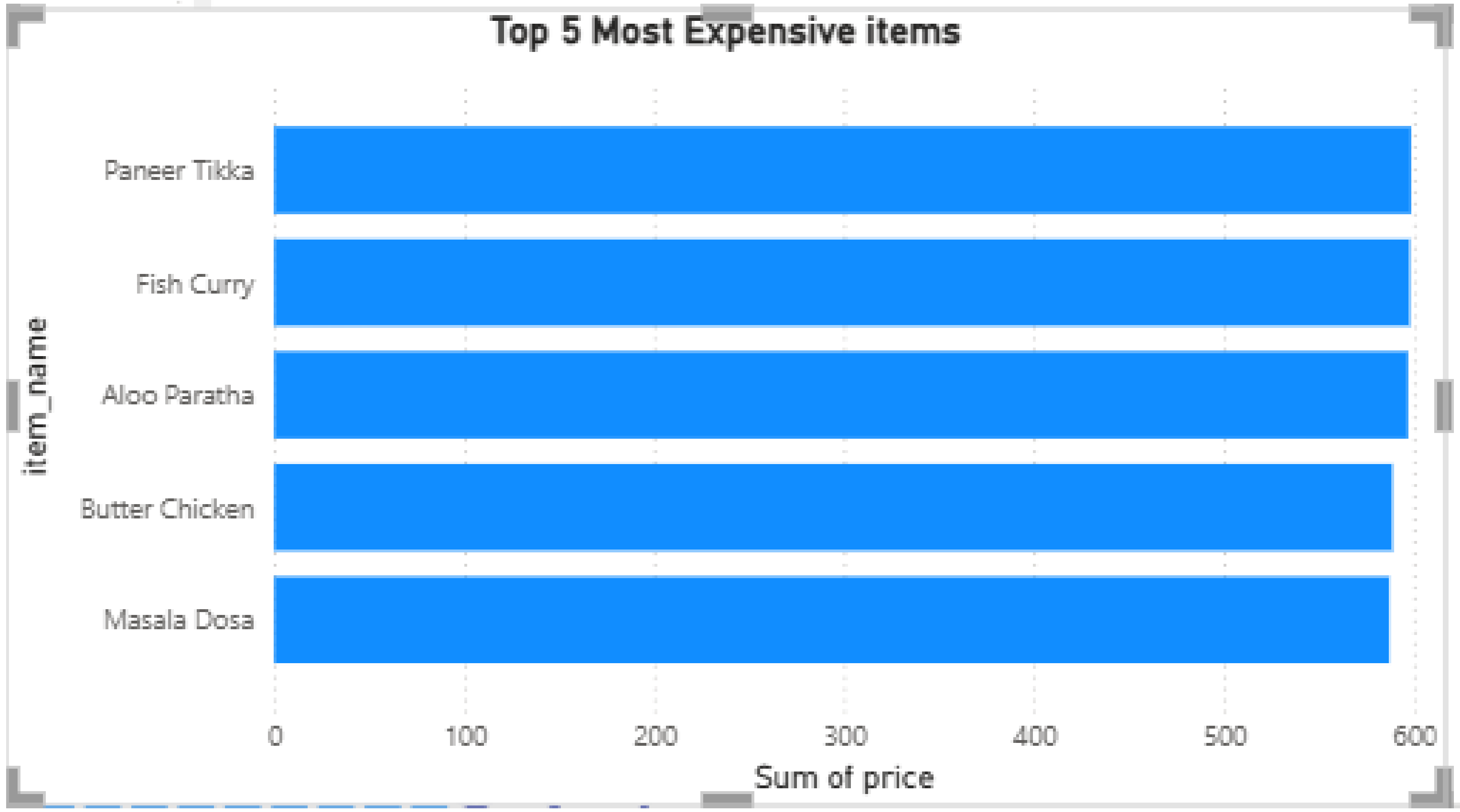


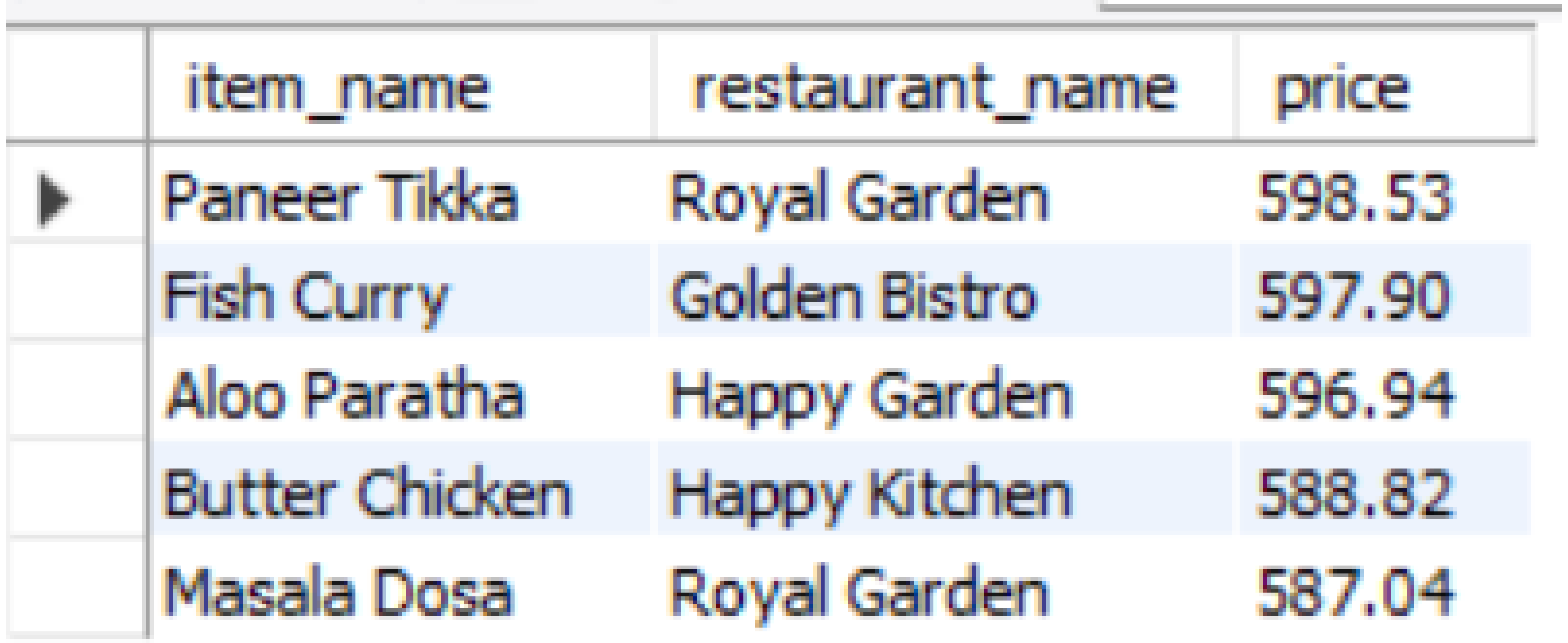
	restaurant_name	total_revenue
▶	Golden Garden	188056.19
	Fresh Hub	145896.60
	Tasty Bistro	137483.55

```
SELECT
r.restaurant_name,
SUM(od.quantity * m.price) AS total_revenue
FROM order_details od
JOIN menu_item m
ON od.item_id = m.item_id
JOIN restaurant r
ON m.restaurant_id = r.restaurant_id
GROUP BY r.restaurant_name
ORDER BY total_revenue DESC
LIMIT 3;
```

Queries & Methods

8. Retrieve the top 5 most expensive menu items across all restaurants?





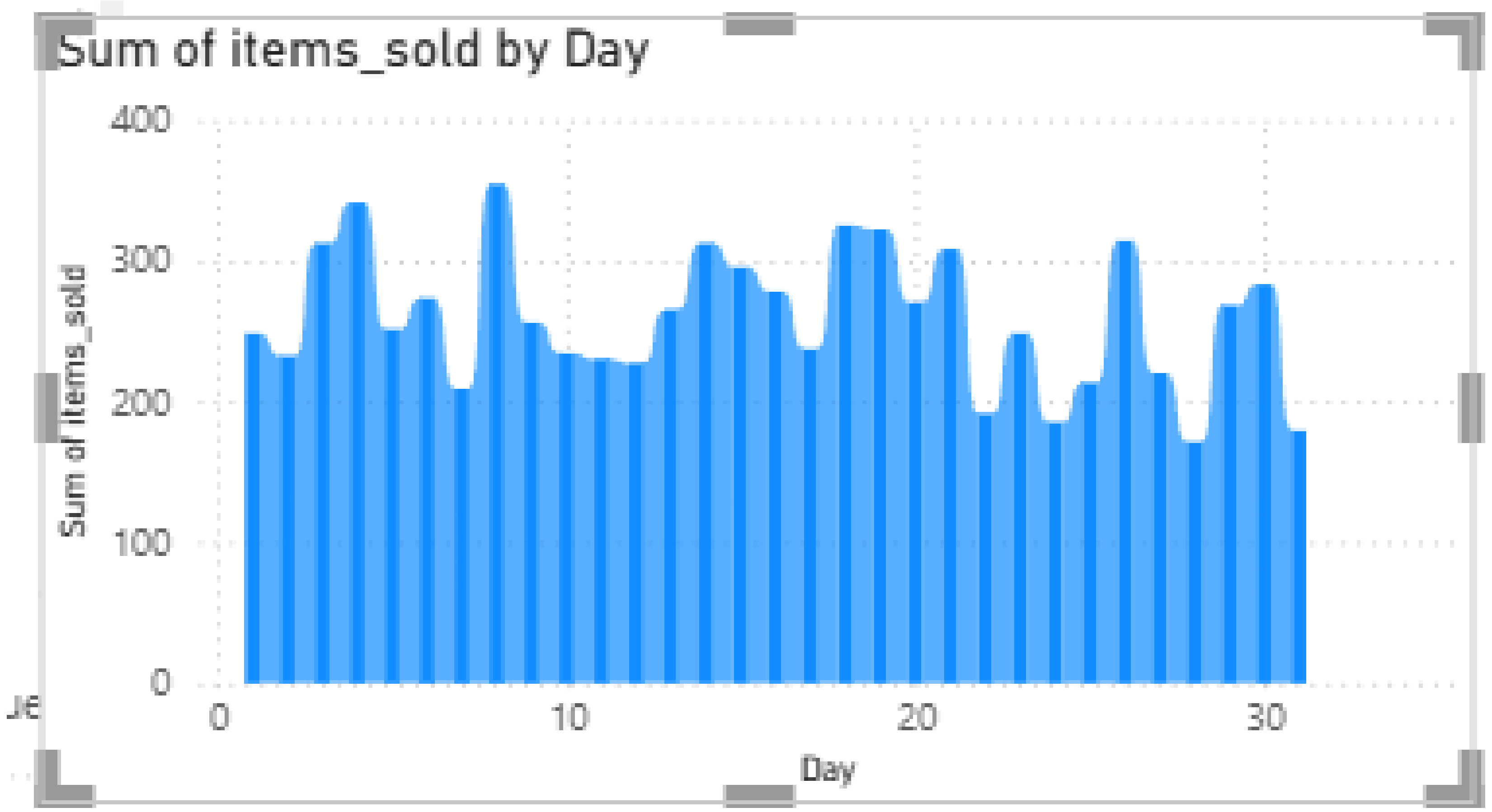
item_name	restaurant_name	price
Paneer Tikka	Royal Garden	598.53
Fish Curry	Golden Bistro	597.90
Aloo Paratha	Happy Garden	596.94
Butter Chicken	Happy Kitchen	588.82
Masala Dosa	Royal Garden	587.04

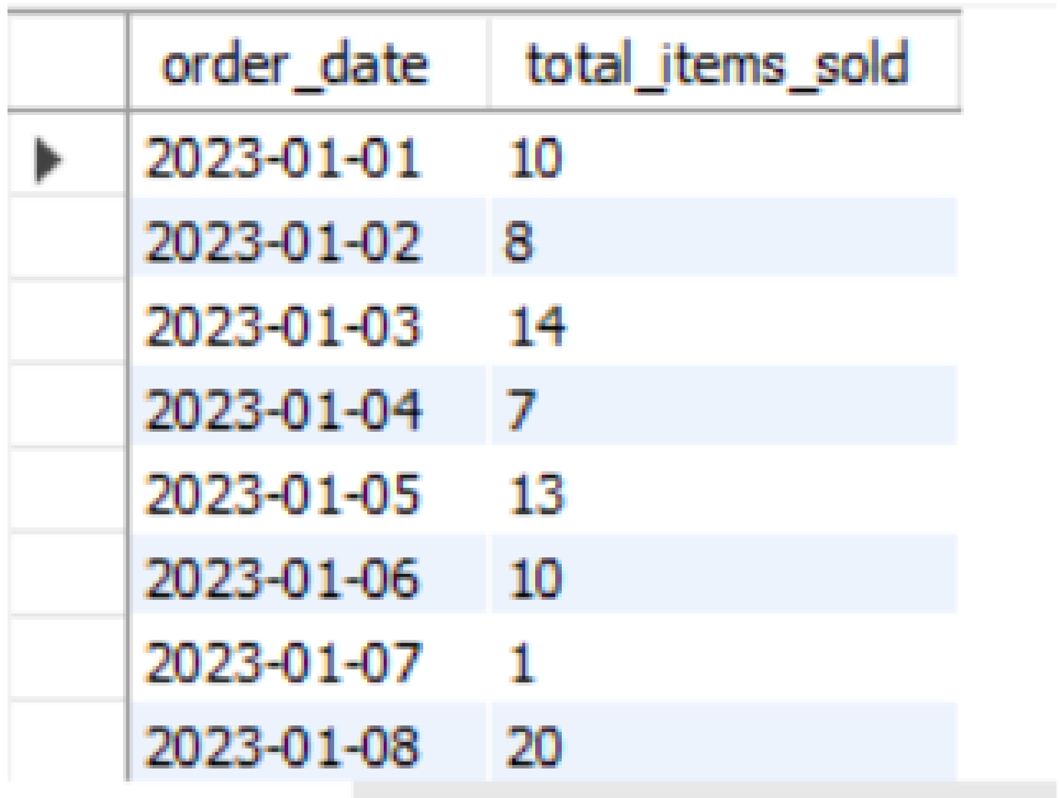
```
SELECT
m.item_name,
r.restaurant_name,
m.price
FROM menu_item m
JOIN restaurant r
ON m.restaurant_id = r.restaurant_id
ORDER BY m.price DESC
LIMIT 5;
```

Queries & Methods

9. find out how many items were sold per day?

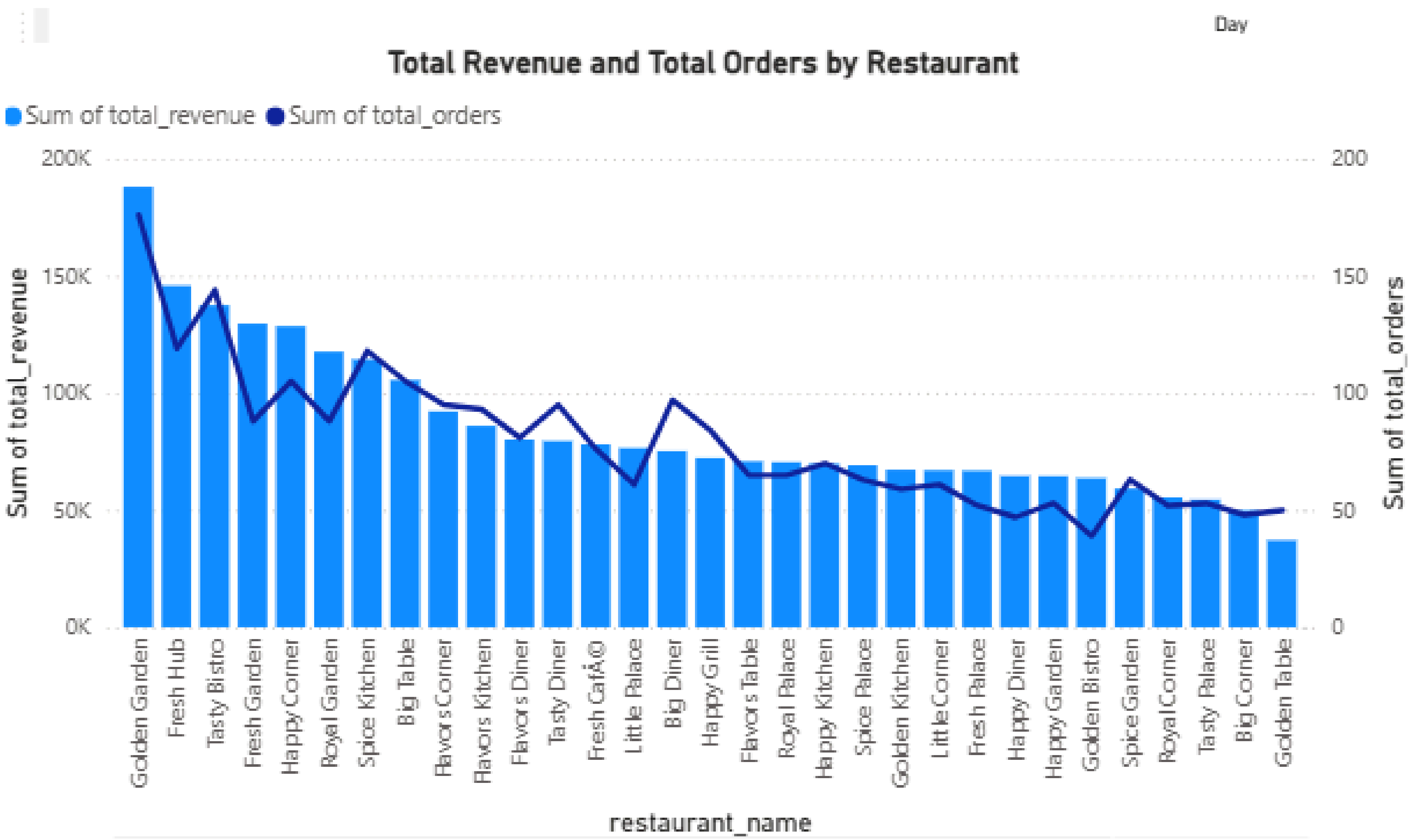
```
SELECT
o.order_date,
SUM(od.quantity) AS total_items_sold
FROM orders o
JOIN order_details od
ON o.order_id = od.order_id
GROUP BY o.order_date
ORDER BY o.order_date;
```

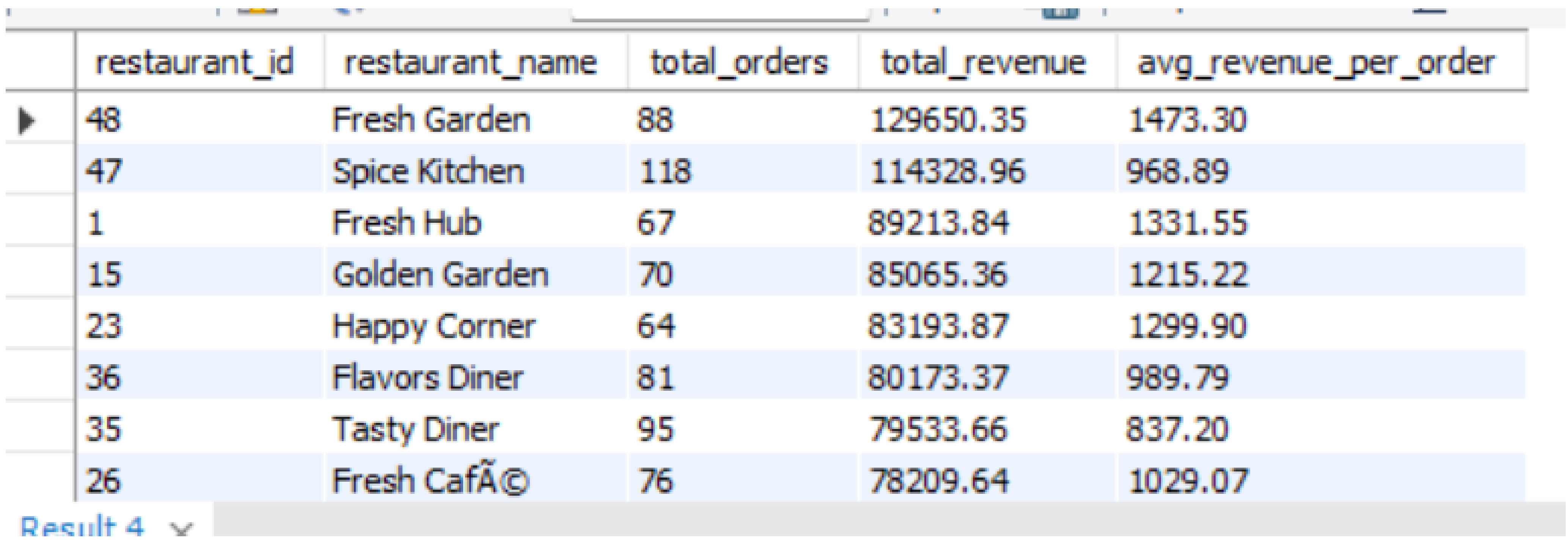




	order_date	total_items_sold
▶	2023-01-01	10
	2023-01-02	8
	2023-01-03	14
	2023-01-04	7
	2023-01-05	13
	2023-01-06	10
	2023-01-07	1
	2023-01-08	20

Queries & Methods



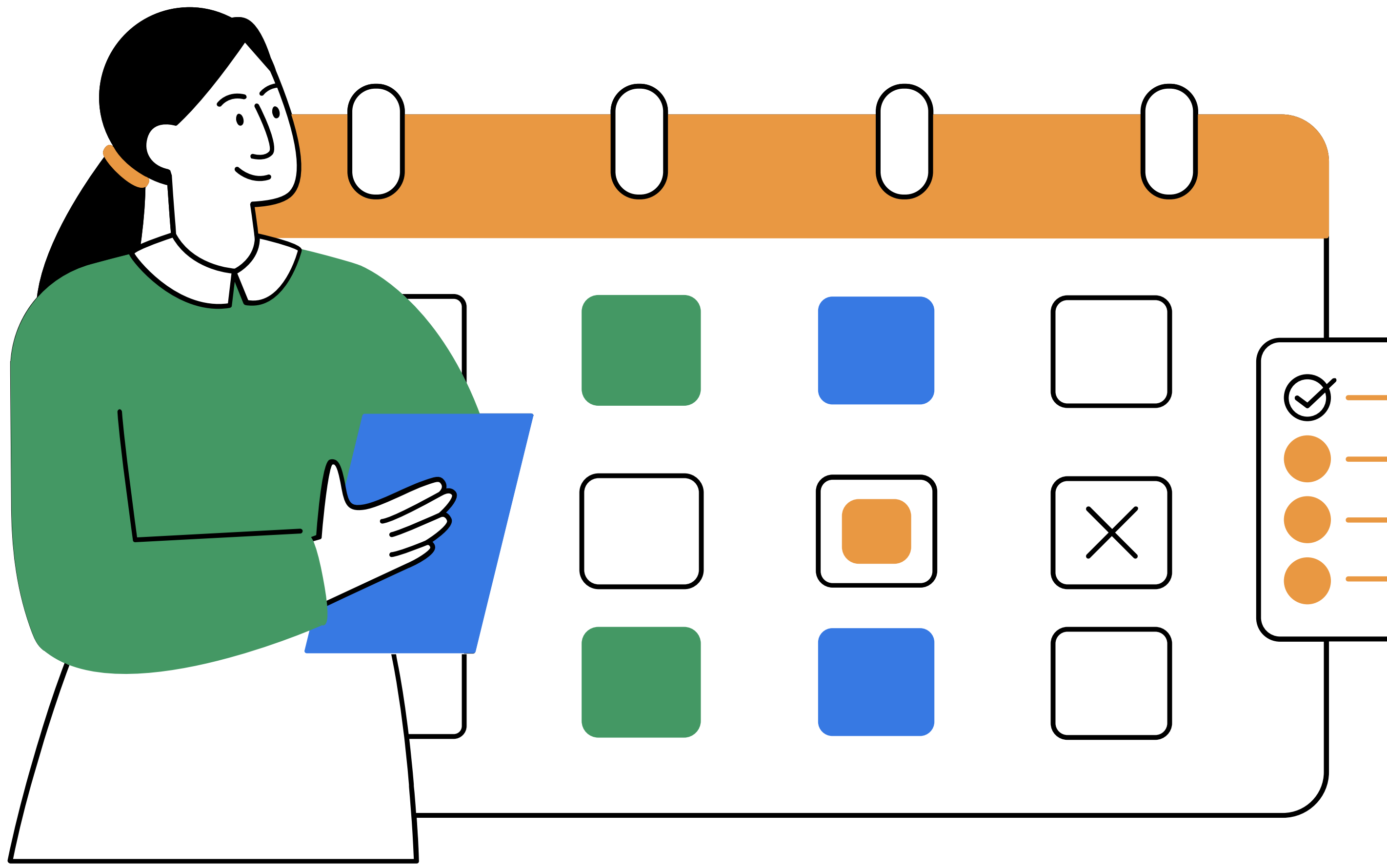


restaurant_id	restaurant_name	total_orders	total_revenue	avg_revenue_per_order
48	Fresh Garden	88	129650.35	1473.30
47	Spice Kitchen	118	114328.96	968.89
1	Fresh Hub	67	89213.84	1331.55
15	Golden Garden	70	85065.36	1215.22
23	Happy Corner	64	83193.87	1299.90
36	Flavors Diner	81	80173.37	989.79
35	Tasty Diner	95	79533.66	837.20
26	Fresh CafÃ©	76	78209.64	1029.07

10. Analyze restaurant performance by calculating the total orders, total revenue, and average revenue per order for each restaurant?

```
SELECT
r.restaurant_id,
r.restaurant_name,
COUNT(DISTINCT o.order_id) AS total_orders,
SUM(od.quantity * m.price) AS total_revenue,
ROUND(SUM(od.quantity * m.price) / COUNT(DISTINCT o.order_id), 2) AS avg_revenue_per_order
FROM restaurant r
JOIN menu_item m
ON r.restaurant_id = m.restaurant_id
JOIN order_details od
ON m.item_id = od.item_id
JOIN orders o
ON od.order_id = o.order_id
GROUP BY r.restaurant_id, r.restaurant_name
ORDER BY total_revenue DESC;
```

Insights & Storytelling



POINT1 . High Revenue ≠ High Orders

Some restaurants may have fewer orders but higher revenue, meaning they sell premium-priced items.

Others may have many small orders, showing popularity but lower average order value.

POINT2. Average Revenue per Order (ARPO)

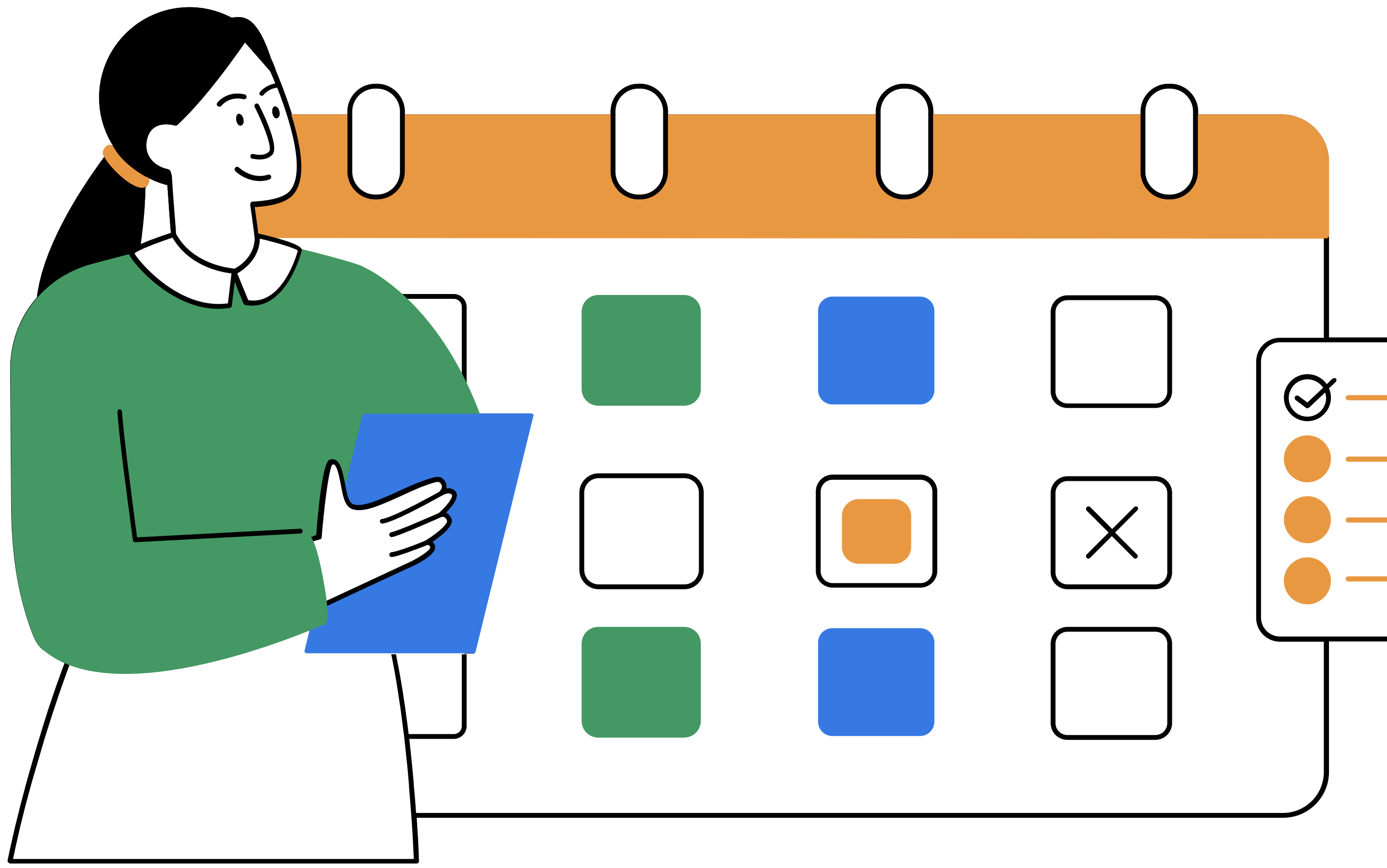
A higher ARPO indicates customers tend to spend more per visit/order, suggesting a premium positioning (fine dining, combo meals).

A lower ARPO might indicate budget-friendly restaurants that rely on order volume.

POINT3. Top Performers

The top 3 restaurants (by revenue) contribute disproportionately to overall sales, showing Pareto principle (80/20 rule) — a small % of restaurants generate most revenue.

Insights & Storytelling



POINT4. Operational Strategy

Restaurants with high total orders but low average revenue can focus on upselling (adding combos, drinks, desserts). Restaurants with low orders but high ARPO should focus on marketing/promotion to increase order frequency.

POINT5. City/Location Factor

If you break down by city, you might notice certain cities generate higher ARPO (maybe urban areas prefer premium restaurants).

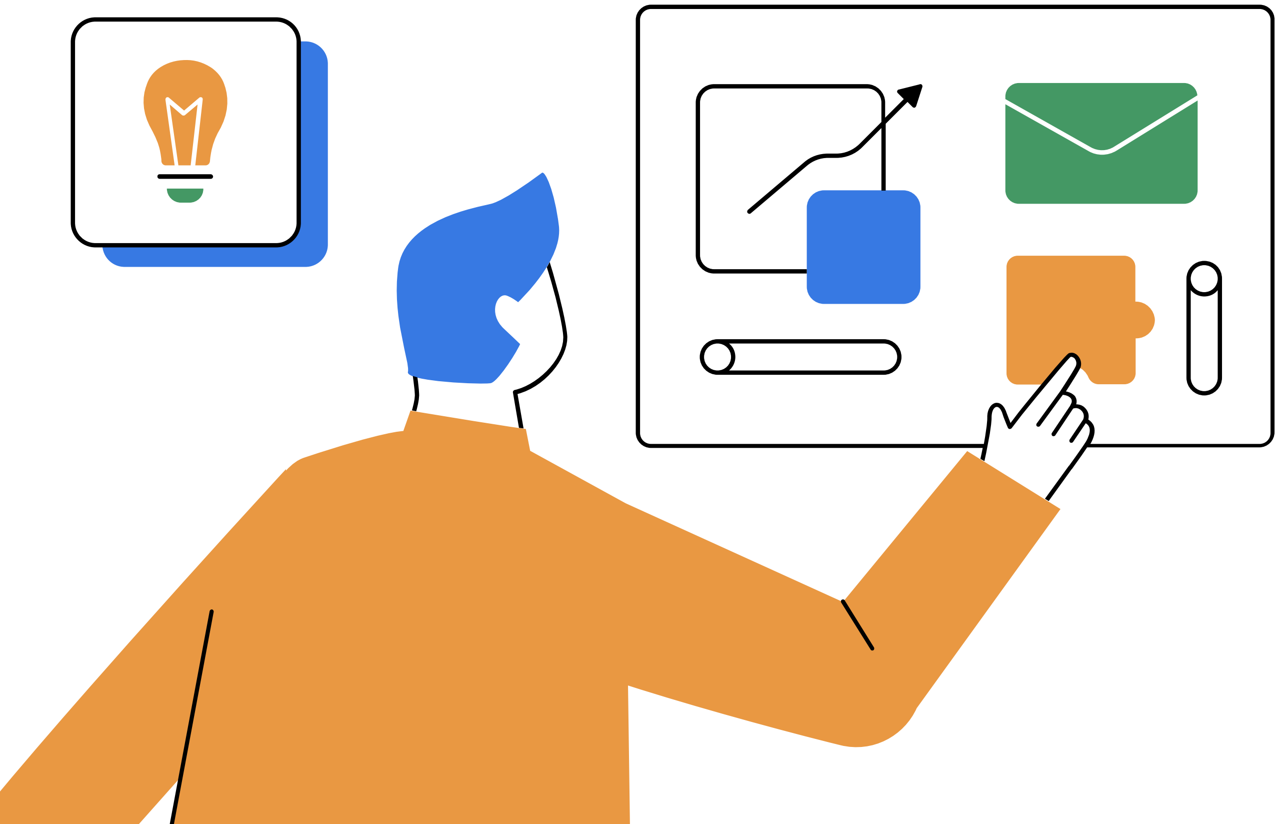
Conclusion & Recommendations

The analysis shows that a few top restaurants generate most of the revenue, while others contribute relatively less.

Some restaurants achieve high revenue due to order volume, while others rely on higher spending per order (premium pricing).

The average revenue per order (ARPO) highlights differences in customer spending patterns — budget restaurants focus on quantity, whereas premium ones focus on quality and pricing.

Overall, restaurant performance varies depending on menu pricing, customer base, and order frequency.



Datapencil

Thank You

● +123-456-7890

● hello@reallygreatsite.com