

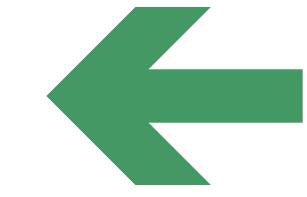
SQL Project Report

Online Food Delivery

Advance Level



Buragapalli Kamachari
Kamacharivarrier@gmail.com



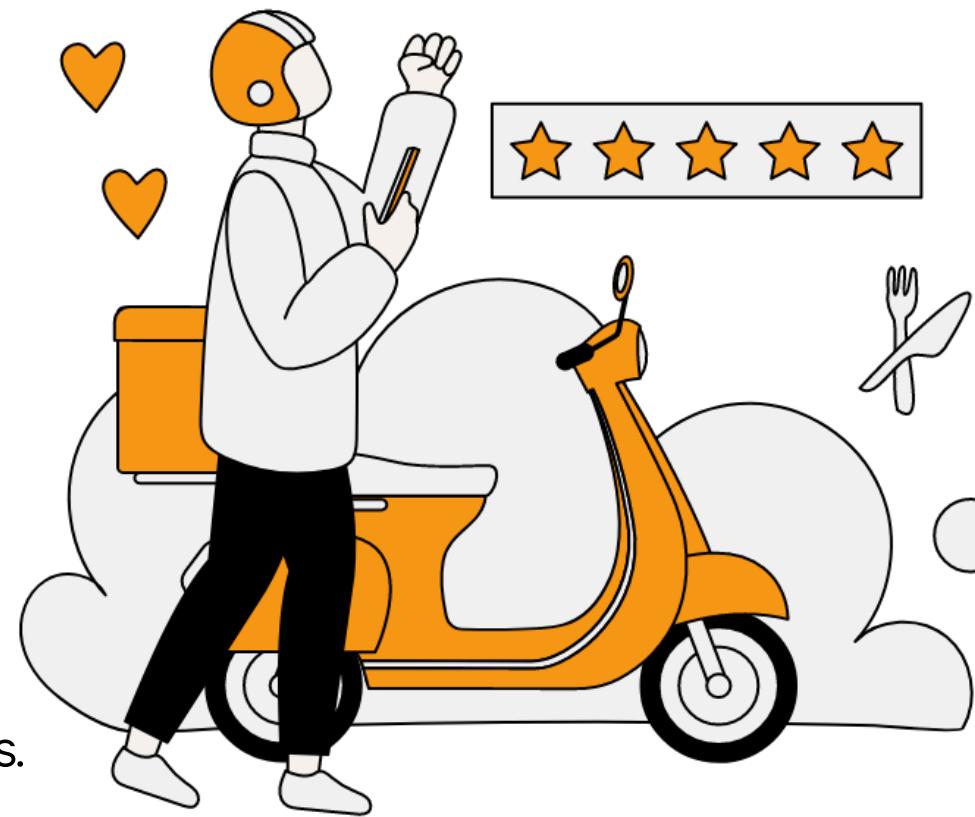
What the project is about



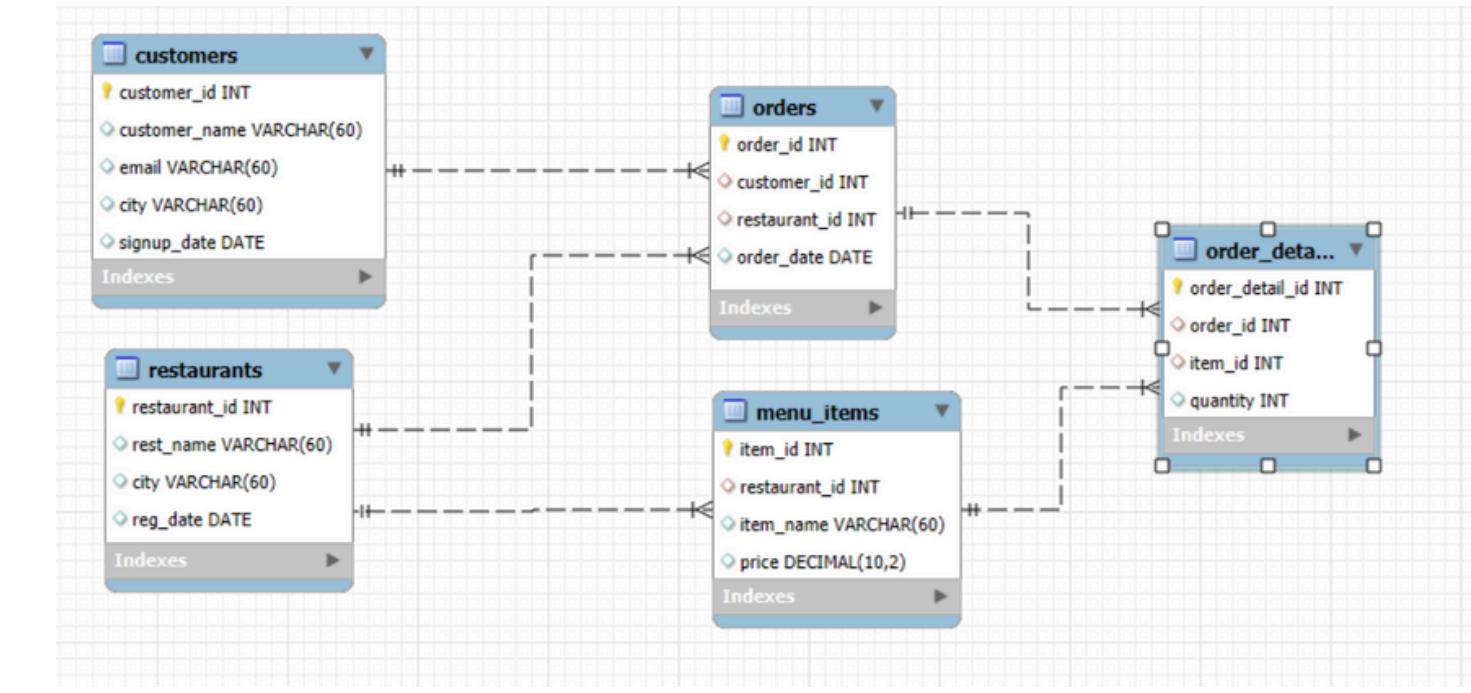
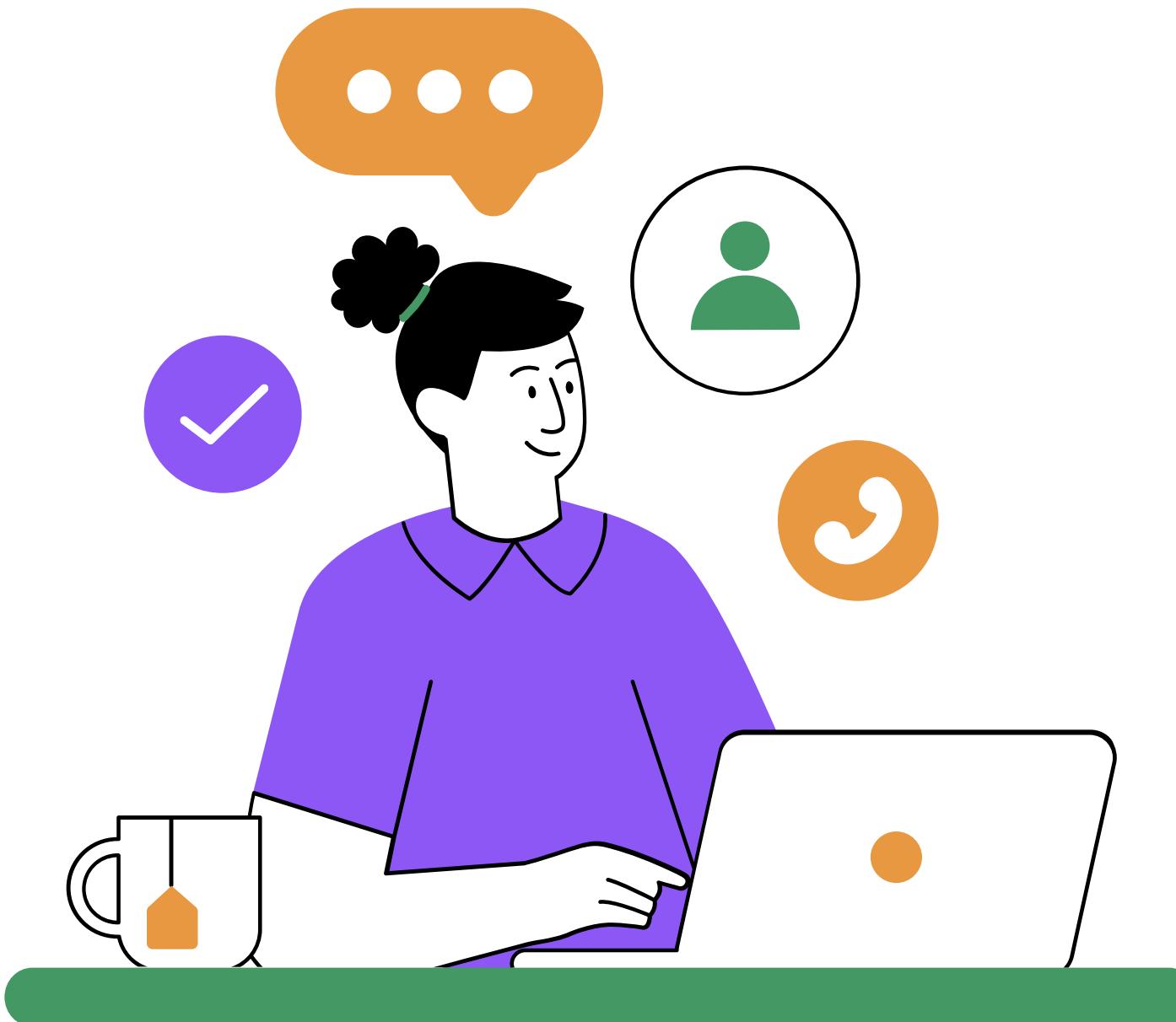
The Online Food Delivery Analytics Project analyzes customer behavior, restaurant performance, and revenue trends using SQL and visualization tools. Key focus areas include identifying top-performing restaurants, frequently ordered items, and customer spending patterns. Advanced SQL techniques like CTEs, views, temporary tables, and window functions were used for efficient data analysis. Visual dashboards (line, bar, and pie charts) were built to present insights clearly. The project demonstrates data analysis, business intelligence, and problem-solving skills relevant to real-world applications.

Problem Statements

1. Customer Segmentation by Signup Year – Classify customers into meaningful categories ("Early Bird," "Regular," "New") based on their signup date.
2. Identify Top Customers – Find customers with the highest number of orders to target loyalty and retention strategies.
3. Menu Price Benchmarking – Detect food items priced above the global average for competitive pricing insights.
4. Restaurant Menu Analysis – Highlight restaurants offering more items than the overall average, showing variety leadership.
5. Monthly Order Performance – Track order volume by month and identify months with more than 50 orders.
6. Restaurant Size Categorization – Classify restaurants as Small, Medium, or Large depending on their menu size.
7. Customer Ranking by Orders – Rank customers based on order frequency to understand engagement levels.
8. Top 3 Revenue-Generating Restaurants – Identify and store the top 3 restaurants by total revenue contribution.
9. Recent Order Tracking – Capture and analyze all orders placed within the last 7 days for trend monitoring.
10. Customer Spend Analysis – Build a view to measure total spend per customer and support customer lifetime value (CLV) analysis.



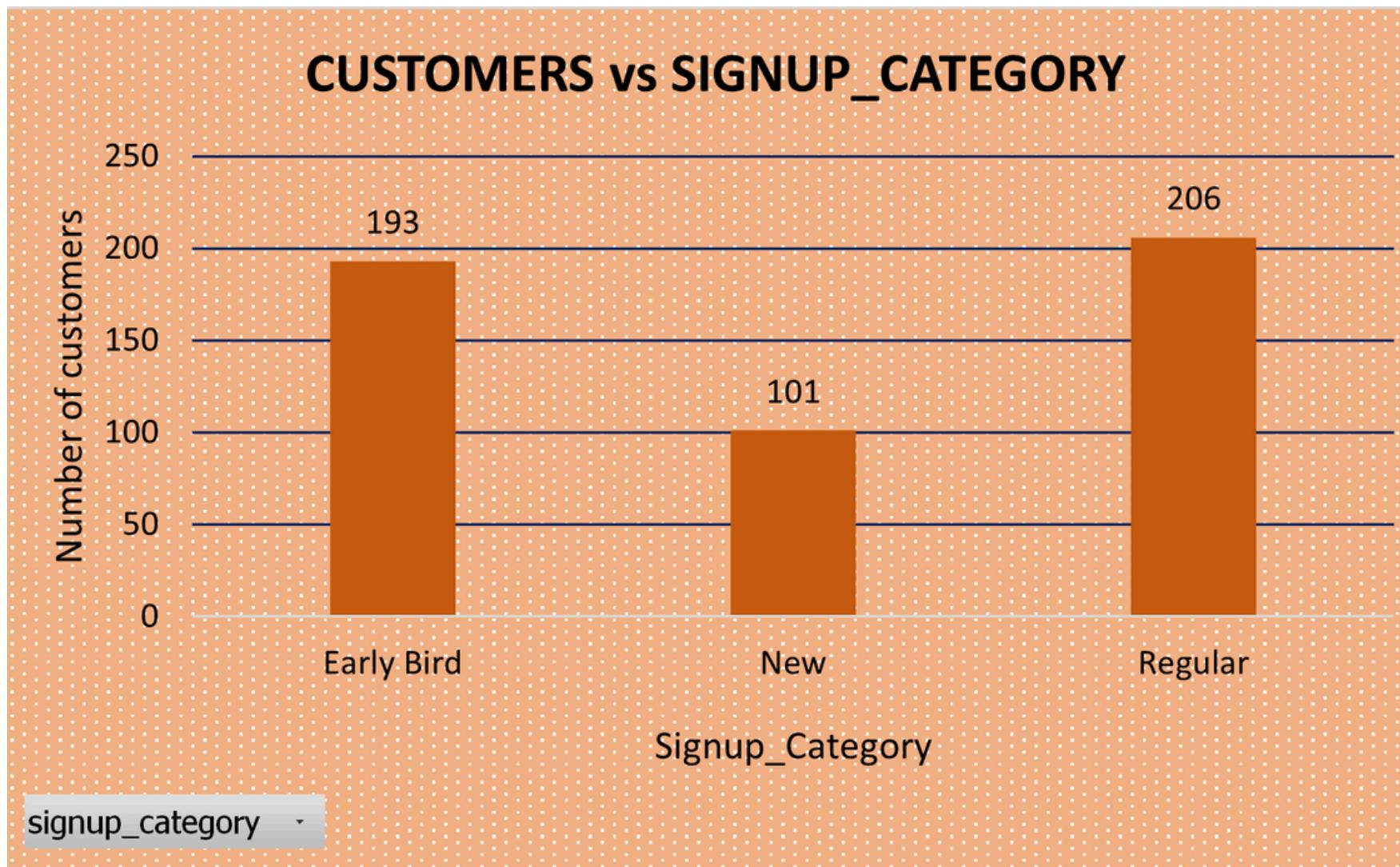
Dataset & Schema



- Represents an online food delivery system covering customers, restaurants, menu items, and orders.
- Organized into relational tables: Customers, Restaurants, Menu_Items, Orders, Order_Details.
- Captures key details such as customer profiles, restaurant offerings, pricing, and order transactions.
- Establishes primary–foreign key relationships for comprehensive sales and performance analysis.
- Serves as a realistic dataset for SQL-based analytics and actionable business insights.

Queries & Methods

Q1: Customer Signup Categories



```

SELECT customer_id, customer_name, signup_date,
CASE
    WHEN YEAR(signup_date) < 2024 THEN 'Early Bird'
    WHEN YEAR(signup_date) = 2024 THEN 'Regular'
    WHEN YEAR(signup_date) = 2025 THEN 'New'
    ELSE 'Future'
END AS signup_category
FROM customers;

```

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

	customer_id	customer_name	signup_date	signup_category
▶	1	Vivaan Sharma	2023-09-08	Early Bird
	2	Vihaan Patel	2024-11-23	Regular
	3	Krishna Verma	2024-03-08	Regular
	4	Aarav Sharma	2023-08-12	Early Bird
	5	Vihaan Das	2023-01-28	Early Bird
	6	Krishna Reddy	2024-03-05	Regular
	7	Vihaan Nair	2023-10-12	Early Bird
	8	Aarav Patel	2023-12-15	Early Bird
	9	Arjun Patel	2025-02-20	New
	10	Reyansh Verma	2024-01-25	Regular

Queries & Methods

Q2: Customers with Maximum Orders



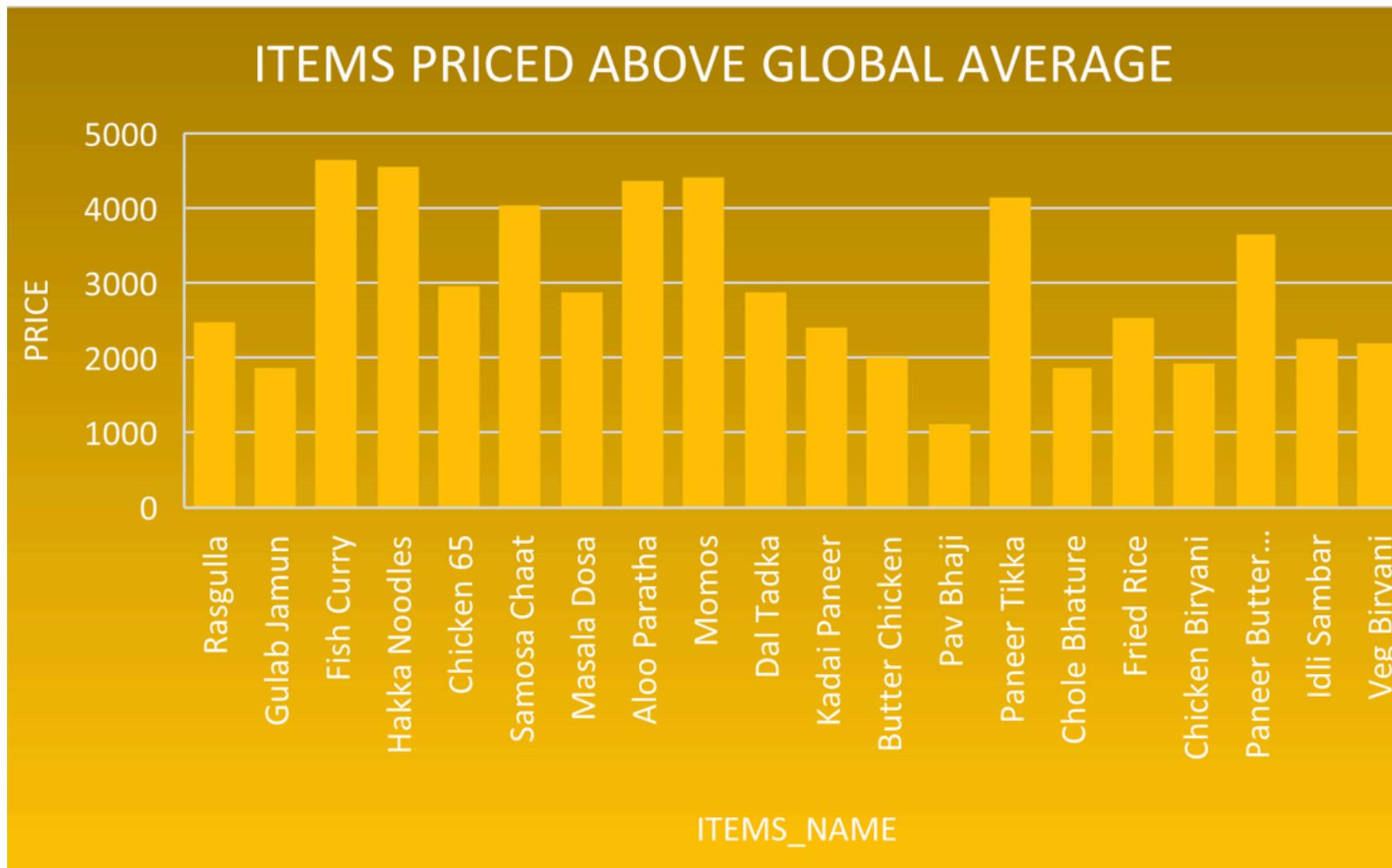
```
WITH customer_orders AS
(
    SELECT customer_id, COUNT(order_id) AS total_orders
    FROM orders
    GROUP BY customer_id
),
max_orders AS
(
    SELECT MAX(total_orders) AS max_order_count
    FROM customer_orders
)
SELECT c.customer_id, cu.customer_name, c.total_orders
FROM customer_orders c
JOIN max_orders m ON c.total_orders = m.max_order_count
JOIN customers cu ON c.customer_id = cu.customer_id;
```

Result Grid | Filter Rows: Export: Wrap

	customer_id	customer_name	total_orders
▶	336	Ishaan Sharma	10

Queries & Methods

Q3: Menu Items Priced Above Average



`SELECT`

```
item_id,  
item_name,  
price
```

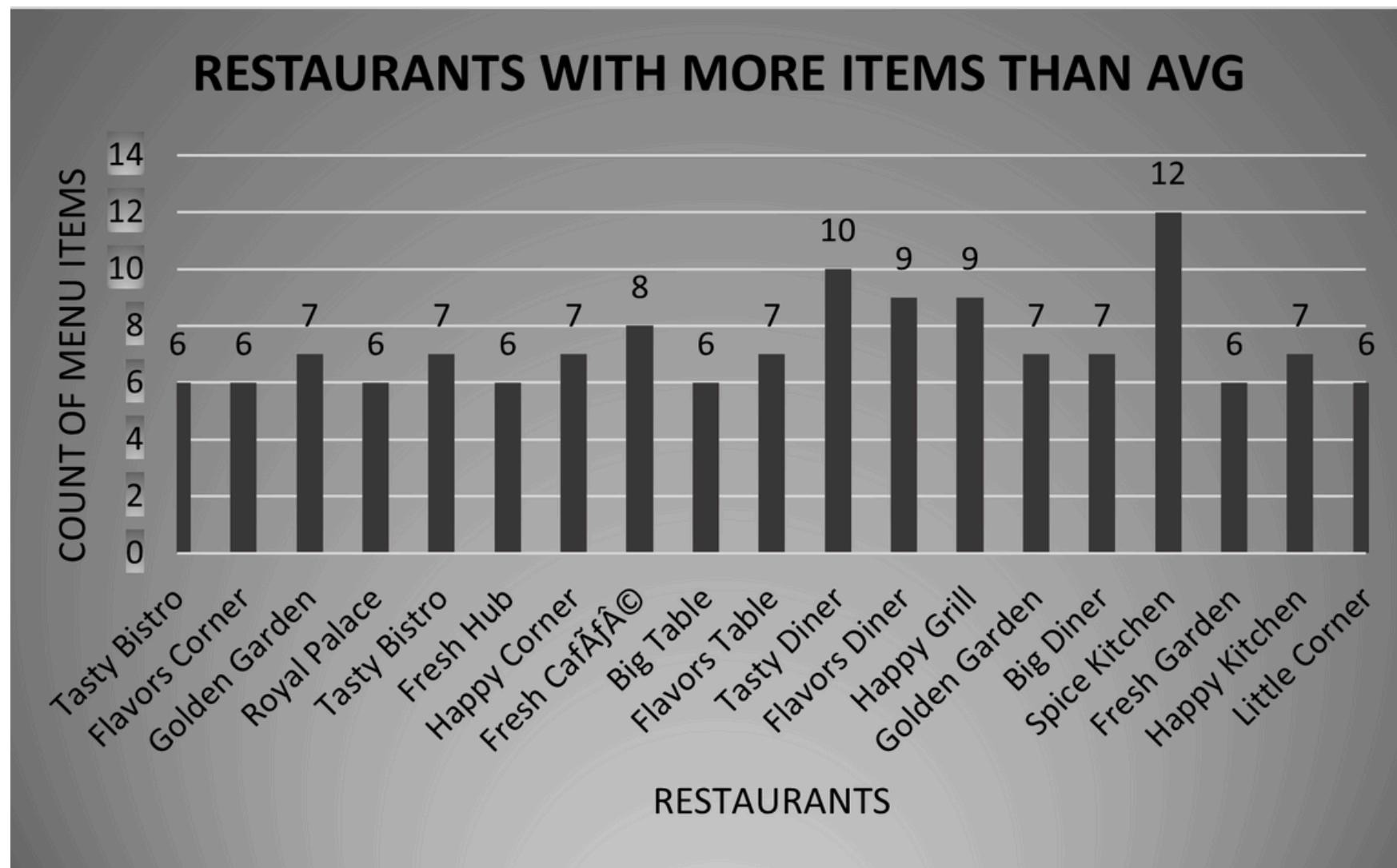
`FROM menu_items`

`WHERE price > (SELECT AVG(price) FROM menu_items);`

	item_id	item_name	price
▶	1	Rasgulla	347.79
	6	Gulab Jamun	565.55
	8	Fish Curry	597.90
	9	Fish Curry	537.12
	12	Hakka Noodles	383.34
	16	Rasgulla	477.21
	20	Chicken 65	477.49
	21	Samosa Chaat	433.81
	22	Masala Dosa	587.04

Queries & Methods

Q4: Restaurants With More Items Than Avg



```
WITH rest_item_count AS
(
    SELECT restaurant_id, COUNT(item_id) AS total_items
    FROM menu_items
    GROUP BY restaurant_id
)
SELECT r.restaurant_id, r.rest_name, ric.total_items
FROM rest_item_count ric
JOIN restaurants r ON ric.restaurant_id = r.restaurant_id
WHERE ric.total_items > (SELECT AVG(total_items) FROM rest_item_count);
```

Result Grid | Filter Rows:

	restaurant_id	rest_name	total_items
▶	6	Tasty Bistro	6
	10	Flavors Corner	6
	15	Golden Garden	7
	16	Royal Palace	6
	21	Tasty Bistro	7
	22	Fresh Hub	6
	23	Happy Corner	7
	26	Fresh CafÃ©©	8
	27	Big Table	6
	30	Flavors Table	7

Queries & Methods

Q5: Monthly Order Summary (>50 Orders)



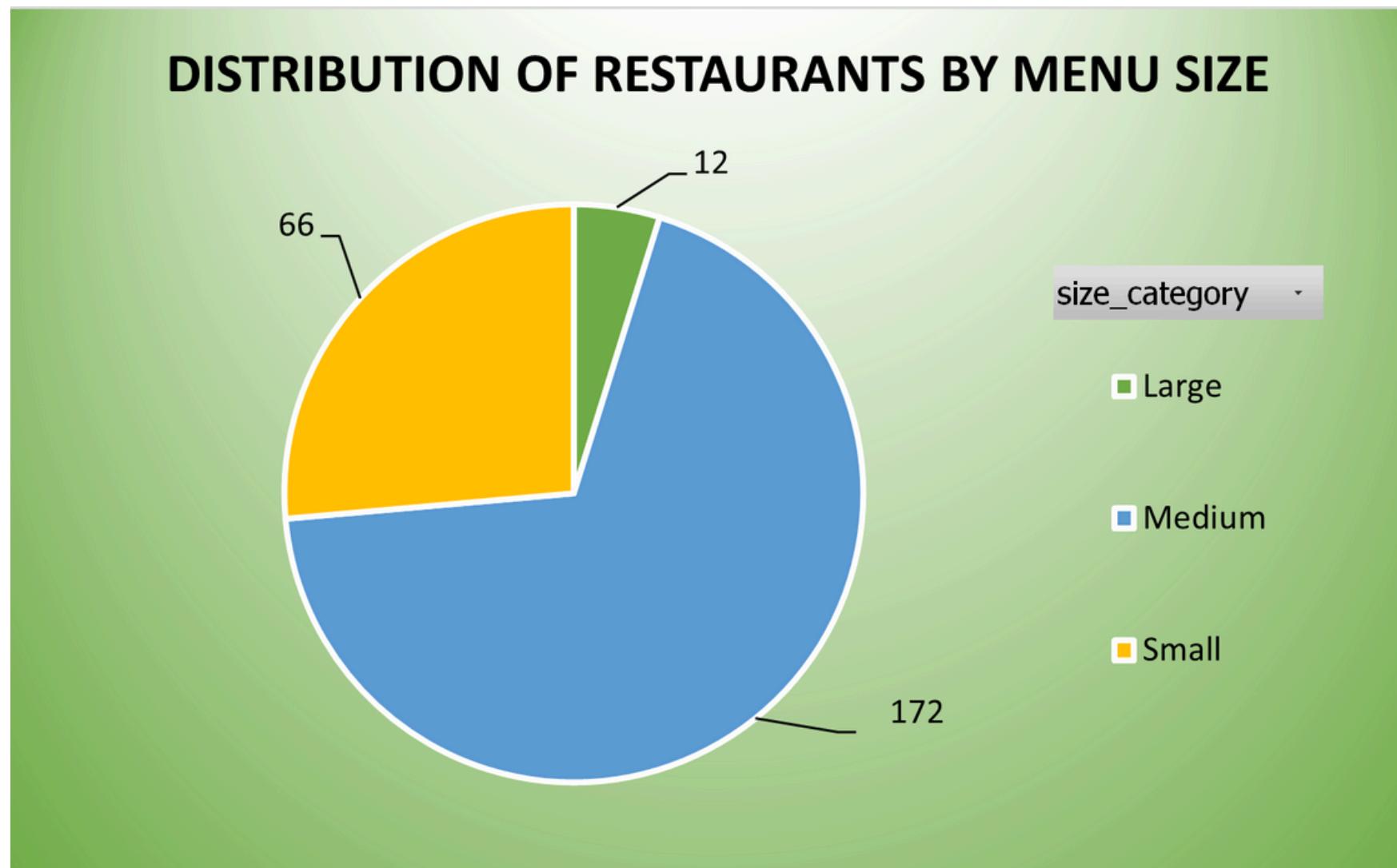
```
WITH monthly_orders AS
(
  SELECT DATE_FORMAT(order_date, '%Y-%m') AS order_month,
         COUNT(order_id) AS total_orders
    FROM orders
   GROUP BY DATE_FORMAT(order_date, '%Y-%m')
)
SELECT * FROM monthly_orders
 WHERE total_orders > 50
 ORDER BY order_month;
```

Result Grid |  Filter Rows: 

	order_month	total_orders
▶	2023-04	56
	2023-05	61
	2023-07	63
	2023-08	52
	2023-10	53
	2024-01	55
	2024-02	53
	2024-04	51
	2024-05	51
	2024-06	51
	2024-07	63

Queries & Methods

Q6: Restaurant Size Category



```

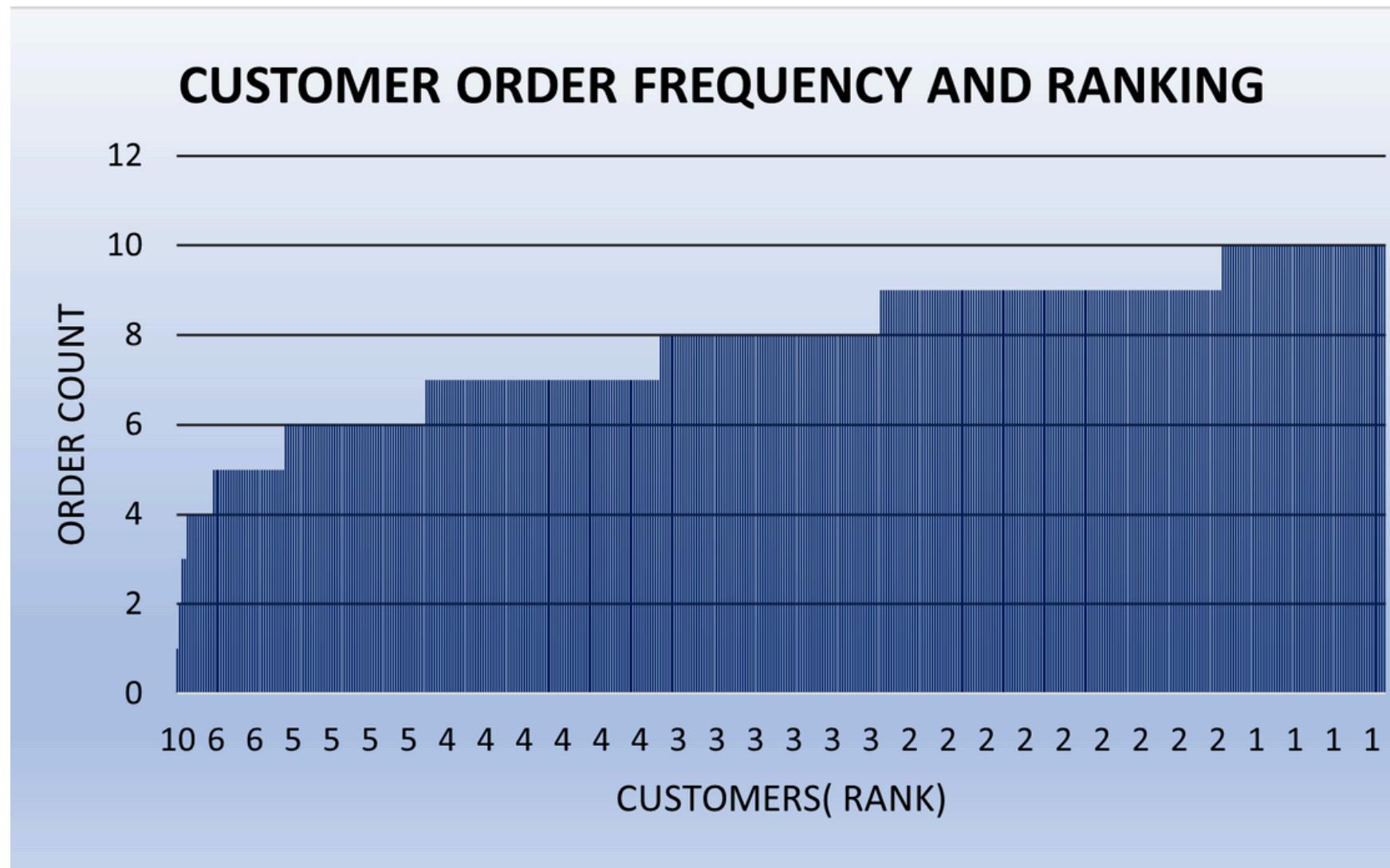
SELECT r.restaurant_id, r.rest_name, COUNT(m.item_id) AS total_items,
CASE
    WHEN COUNT(m.item_id) < 5 THEN 'Small'
    WHEN COUNT(m.item_id) BETWEEN 5 AND 10 THEN 'Medium'
    ELSE 'Large'
END AS size_category
FROM restaurants r
LEFT JOIN menu_items m ON r.restaurant_id = m.restaurant_id
GROUP BY r.restaurant_id, r.rest_name;
  
```

Result Grid |  Filter Rows: Export: 

	restaurant_id	rest_name	total_items	size_category
▶	1	Fresh Hub	5	Medium
	2	Big Diner	3	Small
	3	Big Corner	5	Medium
	4	Royal Hub	2	Small
	5	Happy Garden	4	Small
	6	Tasty Bistro	6	Medium
	7	Big Table	5	Medium
	8	Royal Garden	4	Small
	9	Fresh Palace	4	Small
	10	Flavors Corner	6	Medium

Queries & Methods

Q7: Orders per Customer (Ranked)

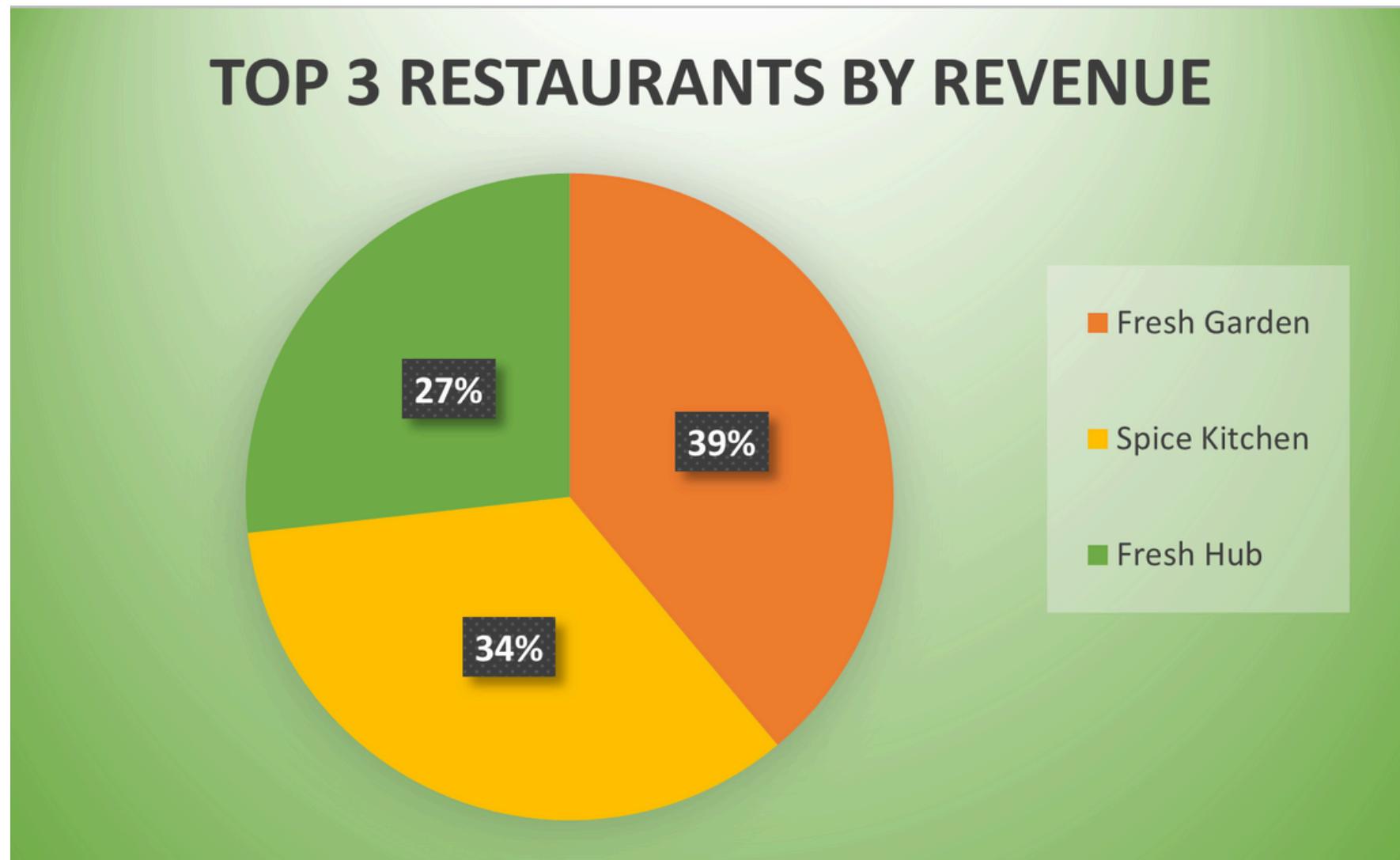


```
WITH customer_orders AS
(
    SELECT customer_id, COUNT(order_id) AS total_orders
    FROM orders
    GROUP BY customer_id
)
SELECT co.customer_id, c.customer_name, co.total_orders,
       RANK() OVER (ORDER BY co.total_orders DESC) AS order_rank
FROM customer_orders co
JOIN customers c ON c.customer_id = co.customer_id;
```

	customer_id	customer_name	total_orders	order_rank
▶	336	Ishaan Sharma	10	1
	28	Vihaan Das	9	2
	154	Aditya Reddy	8	3
	176	Vihaan Patel	8	3
	7	Vihaan Nair	7	5
	72	Arjun Nair	7	5
	120	Ishaan Singh	7	5
	135	Aditya Verma	7	5
	138	Vihaan Patel	7	5
	180	Vivaan Bhat	7	5

Queries & Methods

Q8: Top 3 Restaurants by Revenue



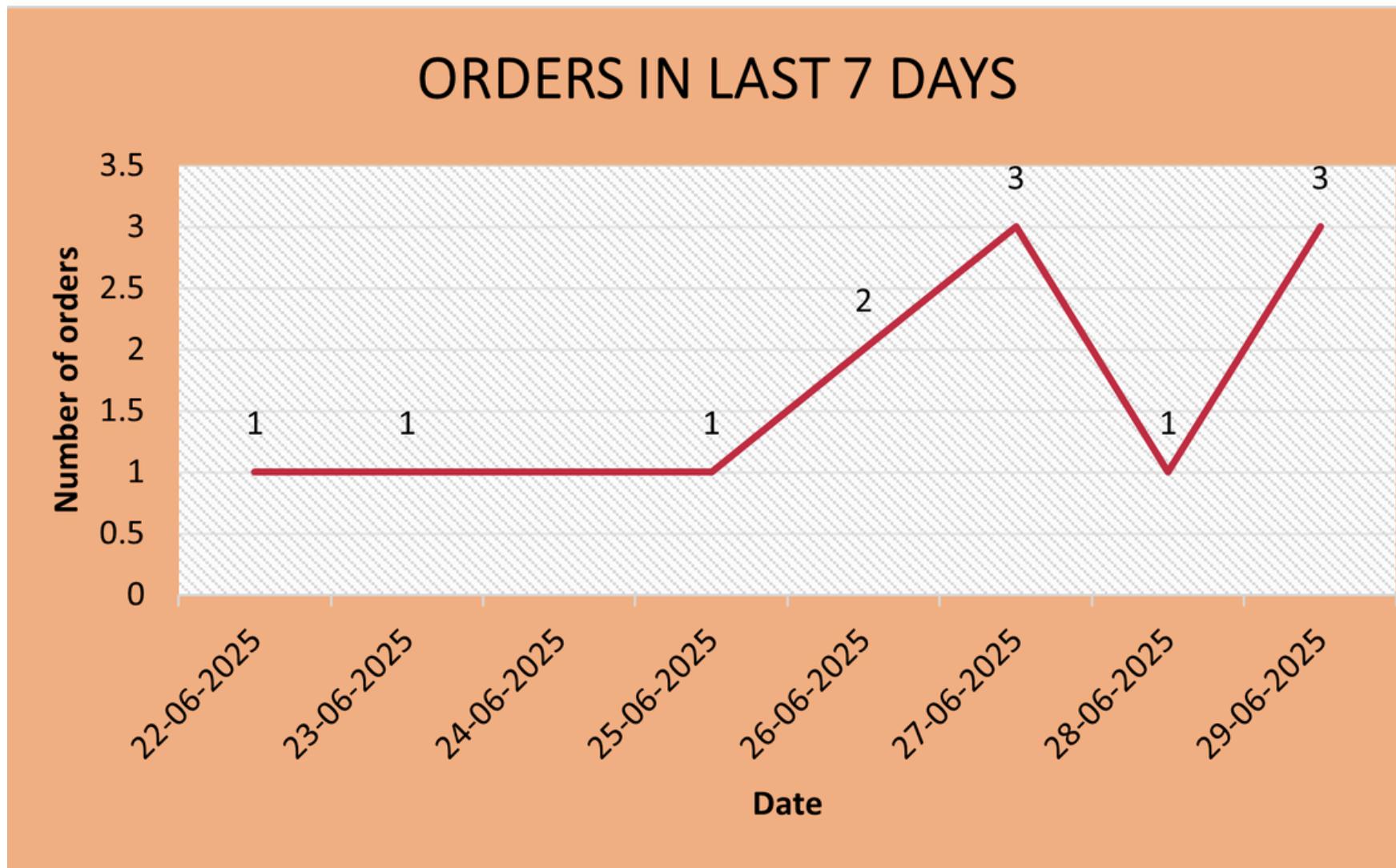
```
CREATE TEMPORARY TABLE top3_restaurants AS
SELECT r.restaurant_id, r.rest_name,
       SUM(mi.price * od.quantity) AS total_revenue
FROM restaurants r
JOIN menu_items mi ON r.restaurant_id = mi.restaurant_id
JOIN order_details od ON mi.item_id = od.item_id
GROUP BY r.restaurant_id, r.rest_name
ORDER BY total_revenue DESC
LIMIT 3;
-- Check result
SELECT * FROM top3_restaurants;
```

Result Grid | Filter Rows:

	restaurant_id	rest_name	total_revenue
▶	48	Fresh Garden	129650.35
	47	Spice Kitchen	114328.96
	1	Fresh Hub	89213.84

Queries & Methods

Q9: Orders in Last 7 Days



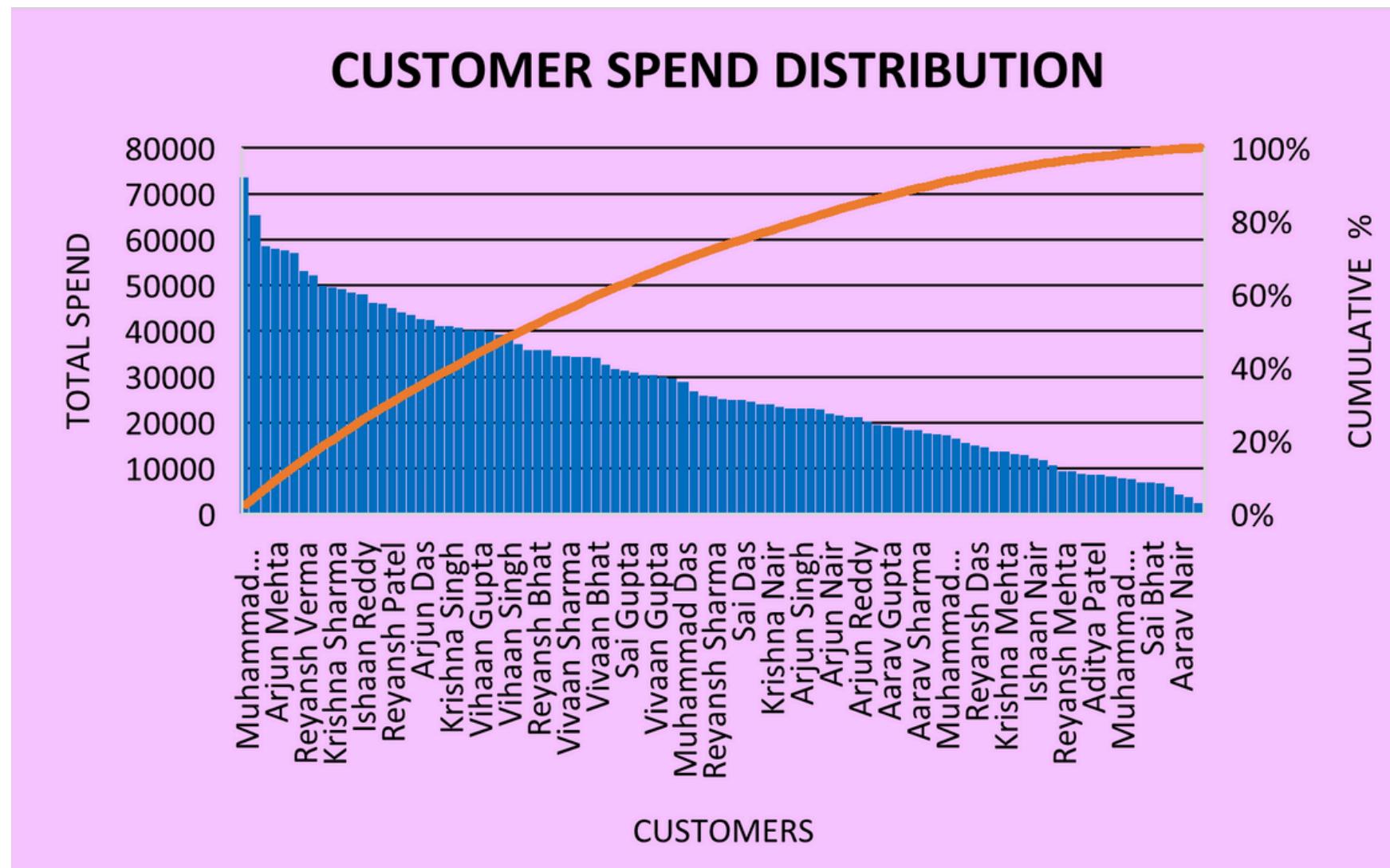
```
CREATE TEMPORARY TABLE last7_orders AS
SELECT *
FROM orders
WHERE order_date >= (SELECT DATE_SUB(max(order_date), INTERVAL 7 DAY)
FROM orders);
-- Check result
SELECT order_date, count(order_id) FROM last7_orders
GROUP BY order_date;
```

Result Grid |   Filter Rows: | Exp

	order_id	customer_id	restaurant_id	order_date
	226	37	34	2025-06-29
	518	20	26	2025-06-29
	1183	378	29	2025-06-29
	446	176	11	2025-06-28
	902	153	44	2025-06-27
	981	406	31	2025-06-27
	1283	102	24	2025-06-27
	261	138	47	2025-06-26
	700	216	42	2025-06-26
▶	426	86	41	2025-06-25

Queries & Methods

Q10: Customer Spend Distribution



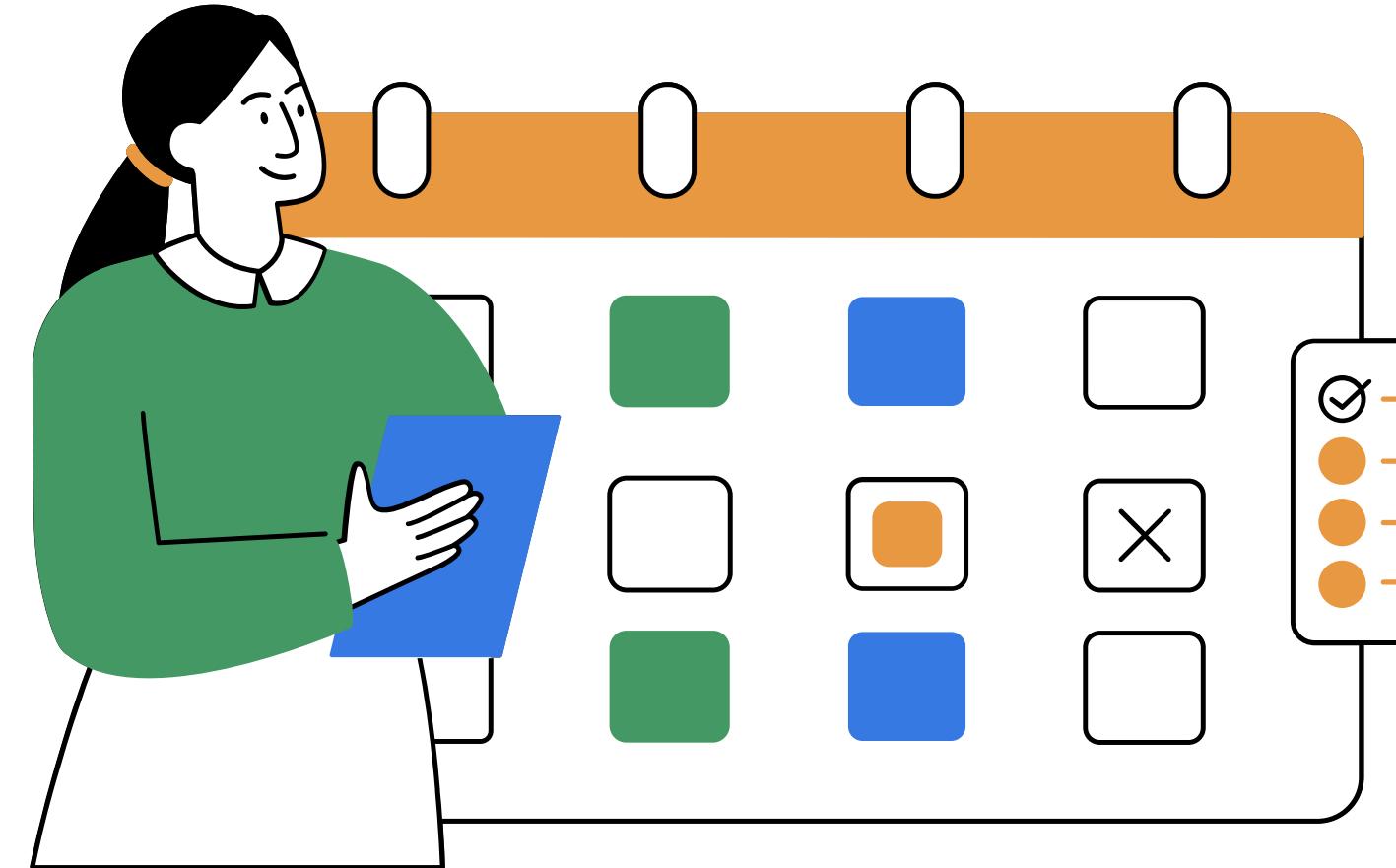
```

CREATE VIEW customer_spend AS
SELECT o.customer_id, c.customer_name,
       SUM(mi.price * od.quantity) AS total_spend
FROM orders o
JOIN order_details od ON o.order_id = od.order_id
JOIN menu_items mi ON od.item_id = mi.item_id
JOIN customers c ON o.customer_id = c.customer_id
GROUP BY o.customer_id, c.customer_name;
-- Check result
SELECT * FROM customer_spend;
    
```

Result Grid | Filter Rows: [] | Export: []

	customer_id	customer_name	total_spend
1		Vivaan Sharma	3036.44
2		Vihaan Patel	3550.25
3		Krishna Verma	4705.68
4		Aarav Sharma	13374.11
5		Vihaan Das	16394.11
6		Krishna Reddy	6155.66
7		Vihaan Nair	19481.05
8		Aarav Patel	14131.34
9		Arjun Patel	1721.34
10		Reyansh Verma	4143.11

Insights & Storytelling



Point 01

Customer Behavior Analysis – Segmentation revealed that most customers placed fewer than 3 orders, highlighting a low retention rate and the need for targeted loyalty programs.



Point 02

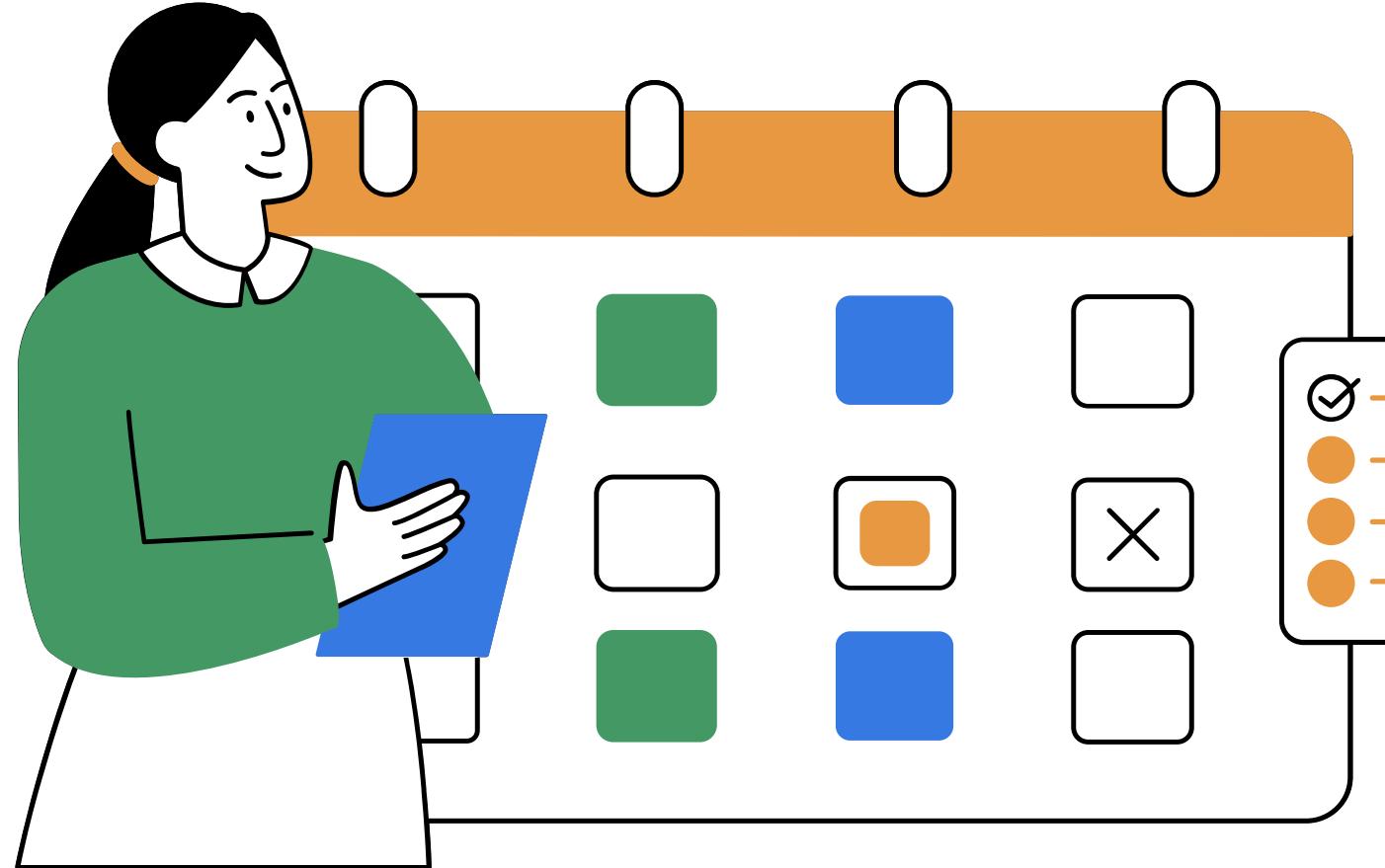
Restaurant Performance Tracking – Revenue ranking showed that top-performing restaurants contribute a significant share of sales, suggesting partnerships with these key players can maximize growth.



Point 03

Menu Optimization – Analysis identified high-demand vs. never-ordered items, enabling restaurants to refine their menus and focus on profitable offerings.

Insights & Storytelling



Point 04

Operational Efficiency – Order trends by city and time exposed regional demand spikes and monthly seasonality, helping optimize delivery capacity and promotional timing.



Point 05

Customer Spend Insights – Total spend per customer highlighted a small group of high-value customers, underscoring the importance of personalized offers to retain premium buyers.

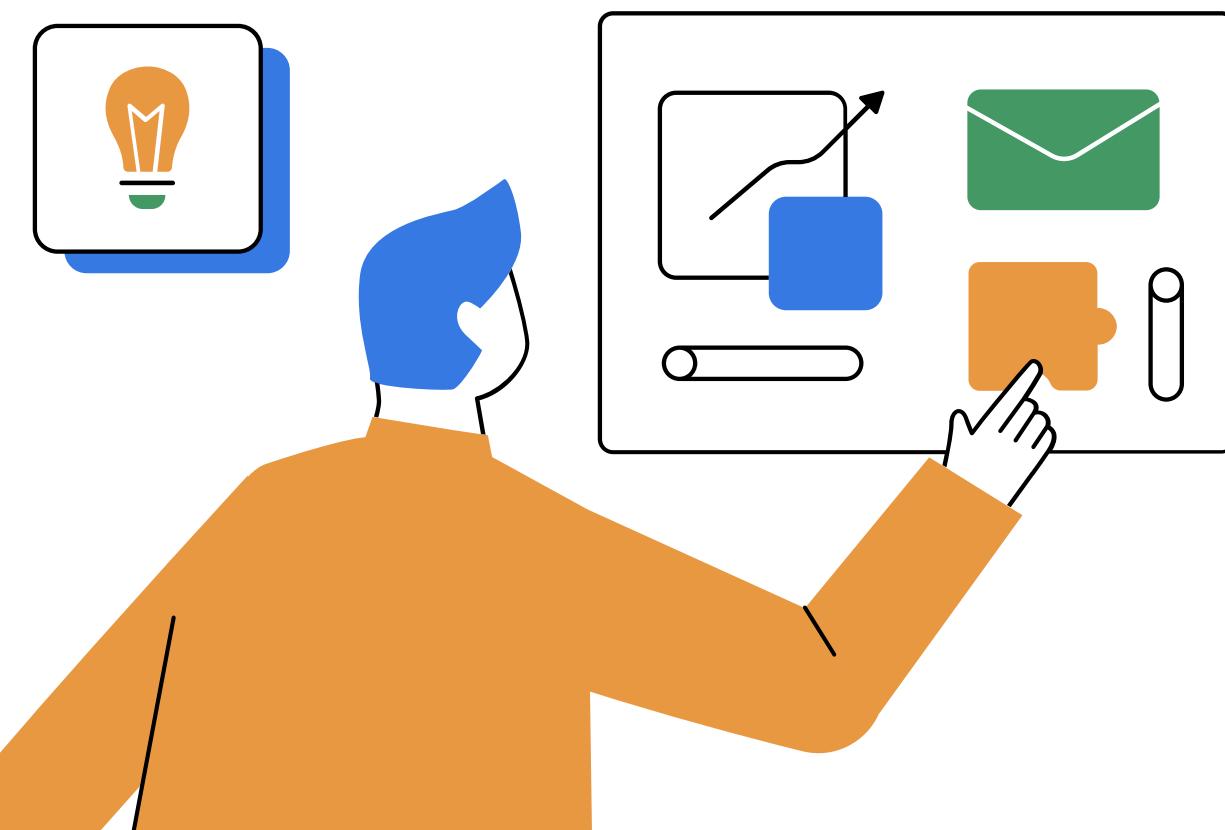


Point 06

Strategic Decision-Making – Overall, the project demonstrated how SQL-driven analytics, combined with visualization tools, can uncover actionable insights to improve customer retention, menu strategy, and restaurant partnerships.

Conclusion & Recommendations

This project on online food delivery analytics successfully demonstrated how SQL can be applied to extract meaningful insights from transactional datasets. By leveraging advanced techniques such as subqueries, CTEs, views, window functions, and temporary tables, we analyzed customer behavior, restaurant performance, pricing strategies, and order trends. The results highlight key areas such as top-performing restaurants, high-value customers, seasonal demand patterns, and opportunities for menu optimization. These insights can support better business decisions around customer retention, revenue growth, and operational efficiency. Overall, the project replicates real-world analytics scenarios and reflects the essential role of data-driven strategies in the food delivery industry.



Thank You



- +91 7893443499

- kamacharivarrier@gmail.com