SQL Project: Customer & Order Analysis

This project presents a comprehensive analysis of customer acquisition, order patterns, promotional impact, and growth strategies using SQL. The queries below were used to derive key business insights.

Top 3 Outlets by Cuisine (Without Using TOP or LIMIT)

```
WITH order_count AS (
    SELECT cuisine, Restaurant_id, COUNT(*) AS no_of_orders
    FROM orders
    GROUP BY cuisine, Restaurant_id
)
SELECT *
FROM (
    SELECT *, ROW_NUMBER() OVER(PARTITION BY cuisine ORDER BY no_of_orders DESC) AS rnk
    FROM order_count
) a
WHERE rnk <= 3;
```

Daily New Customer Count From Launch Date

```
WITH cte AS (
    SELECT customer_code, CAST(MIN(placed_at) AS DATE) AS first_order_date
    FROM orders
    GROUP BY customer_code
)
SELECT first_order_date, COUNT(*) AS new_cust_count
FROM cte
GROUP BY first_order_date
ORDER BY first_order_date;
```

Daily New Customer Count (Window Function Approach)

```
WITH customer_first_orders AS (
    SELECT customer_code, DATE(placed_at) AS order_date,
           ROW_NUMBER() OVER (PARTITION BY customer_code ORDER BY placed_at) AS rn
    FROM orders
)
SELECT order_date, COUNT(*) AS new_customers
FROM customer_first_orders
WHERE rn = 1
GROUP BY order_date
ORDER BY order_date;
```

Customers Acquired in Jan 2025 with Only One Order

```
SELECT customer_code, COUNT(*) AS no_of_orders
FROM orders
WHERE MONTH(Placed_at) = 1 AND YEAR(Placed_at) = 2025
  AND customer_code NOT IN (
    SELECT DISTINCT customer_code
    FROM orders
    WHERE MONTH(placed_at) != 1 AND YEAR(placed_at) = 2025
)
GROUP BY customer_code
HAVING COUNT(*) = 1;
```

Customers with No Order in Last 7 Days and First Order 1 Month Ago with Promo

```
WITH cte AS (
        SELECT customer_code, MIN(placed_at) AS first_order_date, MAX(placed_at) AS
latest_order_date
    FROM orders
    GROUP BY customer_code
)
SELECT cte.*, orders.promo_code_name
FROM cte
JOIN orders ON cte.customer_code = orders.customer_code AND cte.first_order_date =
orders.placed_at
WHERE cte.latest_order_date < DATE_SUB(CURDATE(), INTERVAL 7 DAY)
  AND cte.first_order_date < DATE_SUB(CURDATE(), INTERVAL 1 MONTH)
  AND orders.promo_code_name IS NOT NULL;
```

Trigger for Every Third Order (Personalized Communication)

```
WITH cte AS (
      SELECT *, ROW_NUMBER() OVER(PARTITION BY customer_code ORDER BY placed_at) AS
order_count
    FROM orders
)
SELECT *
FROM cte
WHERE order_count % 3 = 0 AND CAST(placed_at AS DATE) = CURDATE();
```

Customers Who Placed All Orders Using Promo Code

```
SELECT customer_code, COUNT(*) AS no_of_orders, COUNT(promo_code_name) AS promo_count
FROM orders
WHERE promo_code_name IS NOT NULL
GROUP BY customer_code
HAVING COUNT(*) > 1 AND COUNT(*) = promo_count;
```

Percentage of Organic Customers Acquired in Jan 2025

```
WITH cte AS (
    SELECT *, ROW_NUMBER() OVER(PARTITION BY customer_code ORDER BY placed_at) AS rn
    FROM orders
    WHERE MONTH(placed_at) = 1
)
SELECT COUNT(CASE WHEN rn = 1 AND promo_code_name IS NULL THEN customer_code END) *
100.0 / COUNT(DISTINCT customer_code) AS organic_count
FROM cte;
```