LARANA PIZZA

LARANA PIZZA

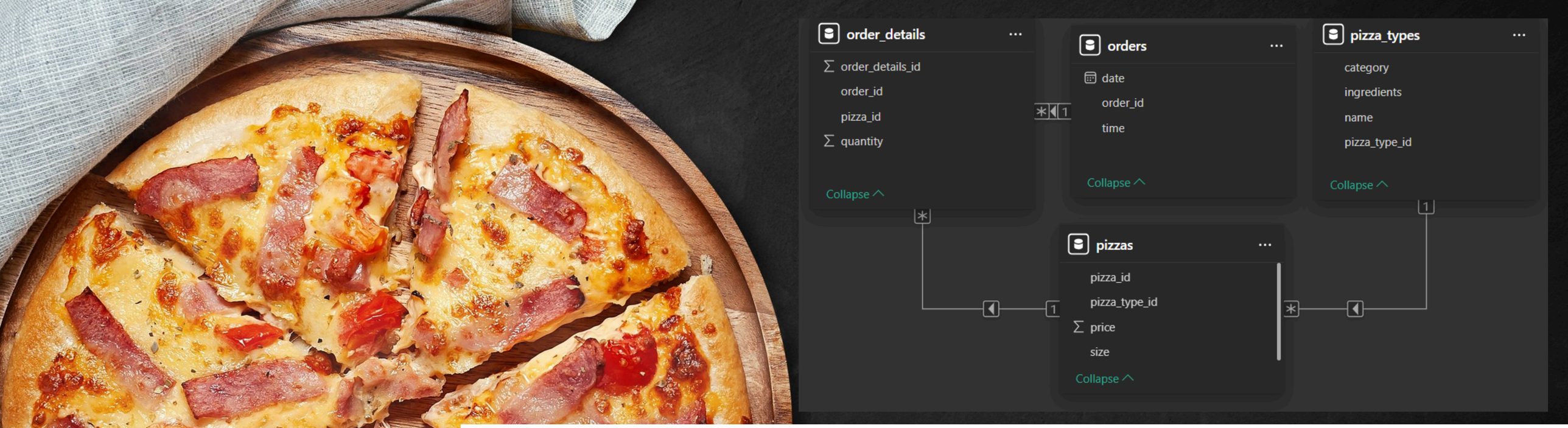
LARANA PIZZA

Hi, my name is Karnav Sood, and this project focuses on analyzing pizza sales data using SQL. In the fast-paced food industry, understanding sales trends and customer preferences is essential for business success. This project aims to extract valuable insights from a pizza sales dataset using SQL queries

Objective

The goal of this project is to analyse sales data to uncover trends, identify top-selling pizzas, and evaluate revenue distribution. Key objectives include:

- **Determining the total number of orders placed.**
- **Identifying the most popular pizza types and sizes.**
- **Analysing revenue trends and category-wise sales distribution.**
- **Understanding order patterns based on time and customer preferences.**



Pizza Sales Database Schema

- **orders**: Stores order date and time.
- **order_details**: Links orders with pizzas and tracks quantity.
- **pizzas**: Contains pizza prices, sizes, and type IDs.
- **pizza_types**: Defines pizza categories, ingredients, and names.

This schema helps analyze sales trends, popular pizzas, and revenue distribution efficiently.

--1 RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT COUNT(order_id) AS No_of_Orders
FROM orders;
```

Results | Messages

	No_of_Orders
1	21350

--2 CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT
    SUM(O.quantity * P.price) AS Total_Revenue
FROM
    order_details O
INNER JOIN
    pizzas P
ON
    O.pizza_id = P.pizza_id
```

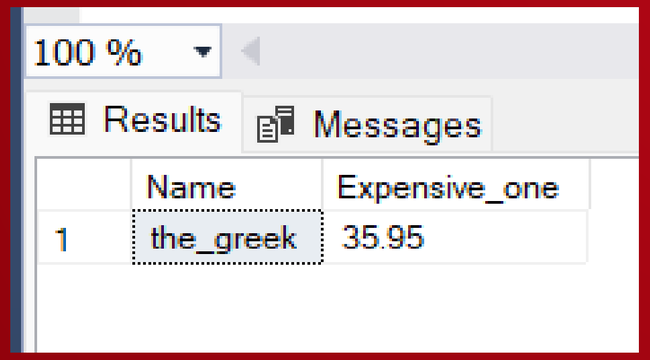
Results | Messages

	Total_Revenue
1	817860.049999994

LARANA PIZZA

--3 IDENTIFY THE HIGHEST-PRICED PIZZA.

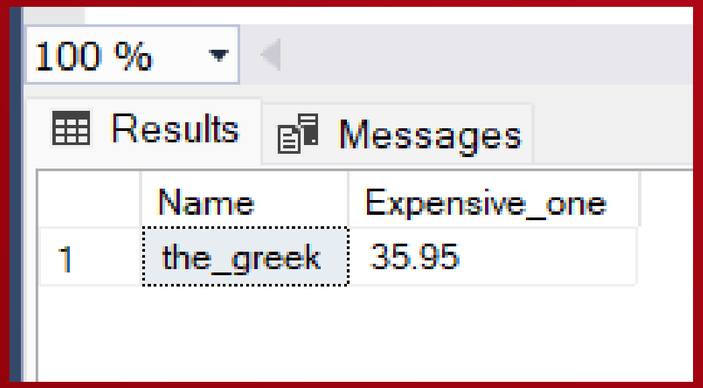
```
SELECT TOP 1
    pizza_type_id AS Name,
    MAX(price) AS Expensive_one
FROM
    pizzas
GROUP BY
    pizza_type_id
ORDER BY
    Expensive_one DESC;
```



	Name	Expensive_one
1	the_greek	35.95

--4 IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
SELECT TOP 1
    P.size,
    COUNT(P.size) AS Total_no_of_orders
FROM
    order_details O
INNER JOIN
    pizzas P
ON
    O.pizza_id = P.pizza_id
GROUP BY
    P.size
ORDER BY
    Total_no_of_orders DESC;
```



--5 LIST THE TOP 5 MOST ORDERED PIZZA NAME ALONG WITH THEIR QUANTITIES.

```sql
SELECT TOP 5
    A.name,
    SUM(O.quantity) AS Total_Orders
FROM
    order_details O
INNER JOIN
    (SELECT
        P1.name,
        P2.pizza_id
     FROM
        pizza_types P1
     INNER JOIN
        pizzas P2
     ON
        P1.pizza_type_id = P2.pizza_type_id
    ) AS A
ON
    A.pizza_id = O.pizza_id
GROUP BY
    A.name
ORDER BY
    Total_Orders DESC;
```

Results | Messages

	name	Total_Orders
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

LARANA PIZZA

--6 JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY
    pizza_types.category
ORDER BY
    quantity DESC;
```

Results | Messages

	category	quantity
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

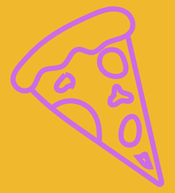
LARANA PIZZA

--7 DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    DATEPART(HOUR, O1.time) AS Hour_of_order,
    COUNT(O1.order_id) AS Count_of_order
FROM
    orders O1
GROUP BY
    DATEPART(HOUR, O1.time)
ORDER BY
    DATEPART(HOUR, O1.time);
```

Results | Messages

	Hour_of_order	Count_of_order
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009
12	20	1642
13	21	1198
14	22	663
15	23	28



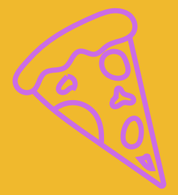
LARANA PIZZA

--8 JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```sql
SELECT
    category,
    COUNT(*) AS Total_Count
FROM
    pizza_types
GROUP BY
    category;
```

Results | Messages

	category	Total_Count
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9



--9 GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```sql
SELECT
    ROUND(AVG(Total_Quantity), 0) AS Avg_of_Quantity_order_per_day
FROM
    (SELECT
        O2.date,
        SUM(O1.quantity) AS Total_Quantity
     FROM
        order_details O1
     INNER JOIN
        orders O2
     ON
        O1.order_id = O2.order_id
     GROUP BY
        O2.date
    ) AS A;
```

Results | Messages

	Avg_of_Quantity_order_per_day
1	138

LARANA PIZZA

--9 GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```sql
SELECT
    ROUND(AVG(Total_Quantity), 0) AS Avg_of_Quantity_order_per_day
FROM
    (SELECT
        O2.date,
        SUM(O1.quantity) AS Total_Quantity
     FROM
        order_details O1
     INNER JOIN
        orders O2
     ON
        O1.order_id = O2.order_id
     GROUP BY
        O2.date
    ) AS A;
```

Results | Messages

	Avg_of_Quantity_order_per_day
1	138

LARANA PIZZA

--10 DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
SELECT TOP 3
    name,
    SUM(Total_Sales) AS Total_Revenue
FROM (
    SELECT
        P2.name,
        P.price * O.quantity AS Total_Sales
    FROM order_details O
    INNER JOIN pizzas P ON O.pizza_id = P.pizza_id
    INNER JOIN pizza_types P2 ON P.pizza_type_id = P2.pizza_type_id
) AS SalesData
GROUP BY name
ORDER BY Total_Revenue DESC;
```

Results | Messages

	name	Total_Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

LARANA PIZZA

--10 DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
SELECT TOP 3
    name,
    SUM(Total_Sales) AS Total_Revenue
FROM (
    SELECT
        P2.name,
        P.price * O.quantity AS Total_Sales
    FROM order_details O
    INNER JOIN pizzas P ON O.pizza_id = P.pizza_id
    INNER JOIN pizza_types P2 ON P.pizza_type_id = P2.pizza_type_id
) AS SalesData
GROUP BY name
ORDER BY Total_Revenue DESC;
```

Results | Messages

	name	Total_Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

LARANA PIZZA

--11 CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
SELECT P2.category,
    SUM(Total_Sales) AS Total_Revenue,
    (SUM(Total_Sales) /
        (SELECT SUM(O.quantity * P.price)
         FROM order_details O
         INNER JOIN pizzas P ON O.pizza_id = P.pizza_id)
    ) * 100 AS Percentage_Contribution
FROM
    pizza_types P2
INNER JOIN
    (SELECT
         P1.pizza_type_id,
         (O.quantity * P1.price) AS Total_Sales
     FROM
         order_details O
     INNER JOIN
         pizzas P1 ON O.pizza_id = P1.pizza_id
    ) AS A
ON
    A.pizza_type_id = P2.pizza_type_id
GROUP BY
    P2.category
ORDER BY Total_Revenue DESC;
```

Results | Messages

	category	Total_Revenue	Percentage_Contribution
1	Classic	220053.1	26.9059602556699
2	Supreme	208196.999999998	25.4563112600988
3	Chicken	195919.5	23.9551375568475
4	Veggie	193690.450000003	23.6825909273847

LARANA PIZZA

--12 ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
SELECT
    O1.date,
    SUM(A.Total_Revenue) AS Total_Revenue,
    SUM(SUM(A.Total_Revenue)) OVER (ORDER BY O1.date ASC) AS Cumulative_Sum
FROM orders O1
INNER JOIN (
    SELECT
        O.order_id,
        (O.quantity * P.price) AS Total_Revenue
    FROM order_details O
    INNER JOIN pizzas P ON O.pizza_id = P.pizza_id
) AS A ON O1.order_id = A.order_id
GROUP BY O1.date
ORDER BY O1.date;
```

100 %

Results | Messages

	date	Total_Revenue	Cumulative_Sum
1	2015-01-01 00:00:00.000	2713.85	2713.85
2	2015-01-02 00:00:00.000	2731.9	5445.75
3			8108.15
4	2015-01-04 00:00:00.000	1755.45	9863.6
5	2015-01-05 00:00:00.000	2065.95	11929.55
6	2015-01-06 00:00:00.000	2428.95	14358.5
7	2015-01-07 00:00:00.000	2202.2	16560.7
8	2015-01-08 00:00:00.000	2838.35	19399.05
9	2015-01-09 00:00:00.000	2127.35	21526.4
10	2015-01-10 00:00:00.000	2463.95	23990.35
11	2015-01-11 00:00:00.000	1872.3	25862.65
12	2015-01-12 00:00:00.000	1919.05	27781.7
13	2015-01-13 00:00:00.000	2049.6	29831.3
14	2015-01-14 00:00:00.000	2527.4	32358.7
15	2015-01-15 00:00:00.000	1984.8	34343.5
16	2015-01-16 00:00:00.000	2594.15	36937.65
17	2015-01-17 00:00:00.000	2064.1	39001.75
18	2015-01-18 00:00:00.000	1976.85	40978.6
19	2015-01-19 00:00:00.000	2387.15	43365.75
20	2015-01-20 00:00:00.000	2397.9	45763.65
21	2015-01-21 00:00:00.000	2040.55	47804.2
22	2015-01-22 00:00:00.000	2496.7	50300.9
23	2015-01-23 00:00:00.000	2423.7	52724.6
24	2015-01-24 00:00:00.000	2289.25	55013.85
25	2015-01-25 00:00:00.000	1617.55	56631.4

Click to select the whole row

LARANA PIZZA

--13 DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```sql
SELECT * FROM (
    SELECT
        A.category,
        A.name,
        SUM(O.quantity * A.price) AS Total_revenue,
        ROW_NUMBER() OVER (
            PARTITION BY A.category
            ORDER BY SUM(O.quantity * A.price) DESC
        ) AS Rn
    FROM order_details O
    INNER JOIN (
        SELECT
            P2.pizza_id,
            P1.category,
            P1.name,
            P2.price
        FROM pizza_types P1
        INNER JOIN pizzas P2
        ON P1.pizza_type_id = P2.pizza_type_id
    ) AS A
    ON O.pizza_id = A.pizza_id
    GROUP BY A.category, A.name
) AS B
WHERE Rn <= 3;
```

100 %

Results | Messages

	category	name	Total_revenue	Rn
1	Chicken	The Thai Chicken Pizza	43434.25	1
2	Chicken	The Barbecue Chicken Pizza	42768	2
3	Chicken	The California Chicken Pizza	41409.5	3
4	Classic	The Classic Deluxe Pizza	38180.5	1
5	Classic	The Hawaiian Pizza	32273.25	2
6	Classic	The Pepperoni Pizza	30161.75	3
7	Supreme	The Spicy Italian Pizza	34831.25	1
8	Supreme	The Italian Supreme Pizza	33476.75	2
9	Supreme	The Sicilian Pizza	30940.5	3
10	Veggie	The Four Cheese Pizza	32265.7000000007	1
11	Veggie	The Mexicana Pizza	26780.75	2
12	Veggie	The Five Cheese Pizza	26066.5	3