SQL QUERY

PIZZA SALES

WELCOME TO 🍕 SQL QUERY

🚀 Unlocking the Secrets of Pizza Sales with SQL! 🍕📊
Delighted to share key insights from our SQL analysis of Pizza's sales data! Here's what we discovered:

✅ **Order Insights:** Total number of orders and the most popular pizza sizes ordered.
💰 **Revenue Drivers:** Top revenue-generating pizzas and their category-wise contributions.
🏆 **Top Performers:** Most ordered pizzas and their quantities.
📈 **Trends:** Revenue patterns over time and distribution of orders by hour.

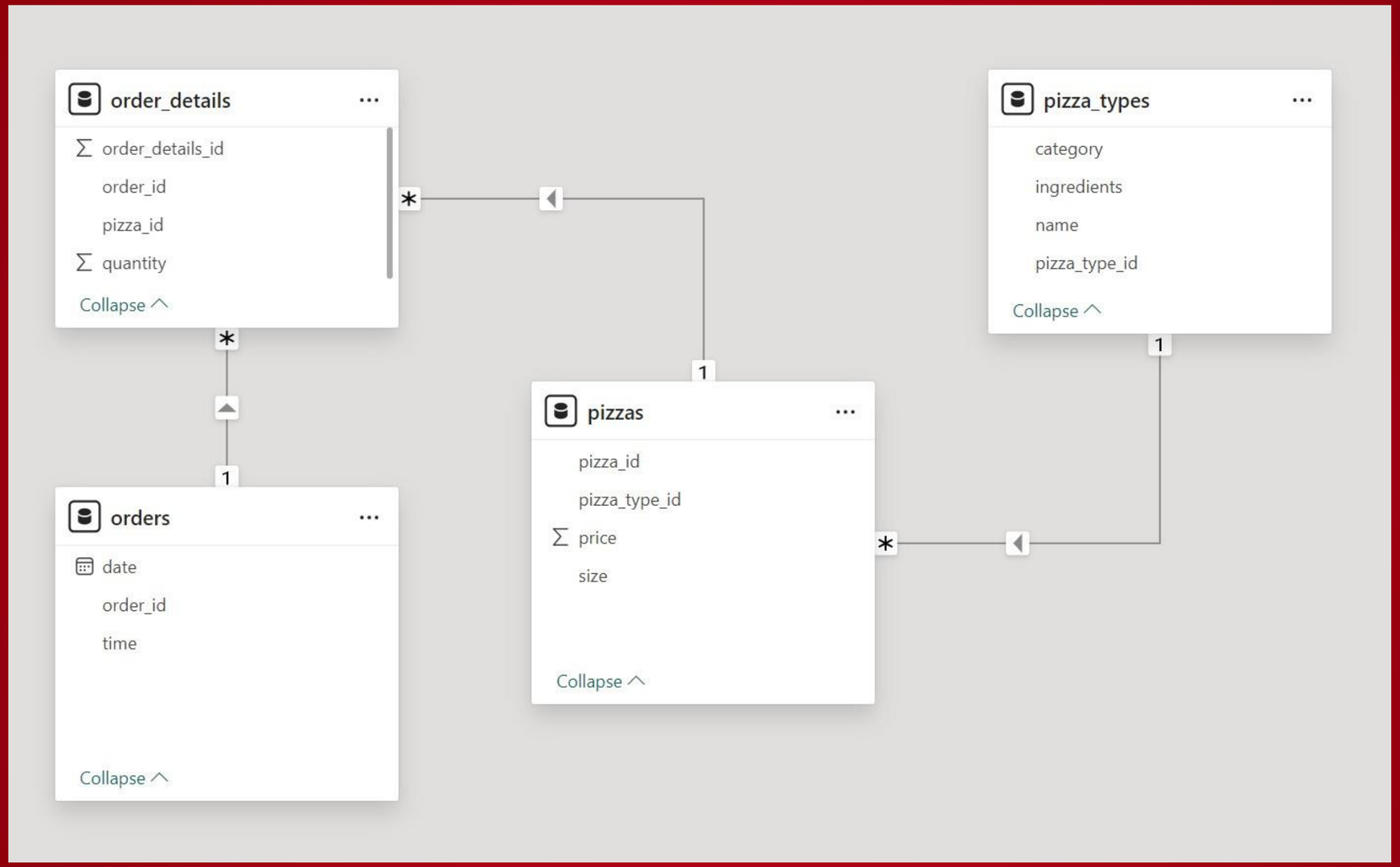
This project showcases the power of data-driven decisions in understanding customer preferences and optimizing business strategies.

MISSION

SQL QUERY

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.
- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.
- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

DATABASE SCHEMA DIAGRAM



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```sql
SELECT
    COUNT(order_id) AS Total_Order
FROM
    pizza.orders;
```

SQL QUERY

Result Grid

	Total_Order
▶	21350

OUTPUT

TOTAL NUMBER OF ODERS

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```sql
SELECT
    *
FROM
    pizza.pizzas;

SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS Total_Sales
FROM
    pizza.order_details
        JOIN
    pizza.pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

SQL QUERY

Result Grid

	Total_Sales
▶	817860.05

OUTPUT

TOTAL REVENUE

IDENTIFY THE HIGHEST-PRICED PIZZA.

```sql
SELECT
  t.name,
  p.price
FROM
  pizza.pizzas p
  JOIN pizza.pizza_types t ON p.pizza_type_id = t.pizza_type_id
ORDER BY
  p.price DESC
LIMIT
  1;
```

SQL QUERY

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

OUTPUT

HIGHEST PRICE

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
SELECT
    p.size, COUNT(od.order_details_id) order_count
FROM
    pizza.order_details od
        JOIN
    pizza.pizzas p ON od.Pizza_ID = p.pizza_id
GROUP BY p.size
ORDER BY order_count DESC;
```

SQL QUERY

MOST COMMON SIZE PIZZA ORDERED

Result Grid | Filter Ro

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

OUTPUT

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

```sql
SELECT
    pt.name, SUM(od.quantity) AS Quantity
FROM
    pizza.pizza_types pt
        JOIN
    pizza.pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    pizza.order_details od ON p.pizza_id = od.Pizza_ID
GROUP BY pt.name
ORDER BY Quantity DESC
LIMIT 5;
```

SQL QUERY

Result Grid | Filter Rows:

name	Quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

OUTPUT

5 MOST ORDERED PIZZA

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

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

SQL QUERY

TOTAL QUANTITY BY CATEGORY

Result Grid | Filter

category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

OUTPUT

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(order_time) AS Hour, COUNT(Order_ID) AS Order_Count
FROM
    pizza.orders
GROUP BY HOUR(order_time);
```

SQL QUERY

Result Grid | Filter

Hour	Order_Count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

OUTPUT

DISTRIBUTION OF ORDERS BY HOUR

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

```sql
SELECT
    category, COUNT(name) Category_Count
FROM
    pizza.pizza_types
GROUP BY category
ORDER BY Category_Count DESC;
```

SQL QUERY

Result Grid | Filter Rows:

category	Category_Count
Supreme	9
Veggie	9
Classic	8
Chicken	6

OUTPUT

CATEGORY-WISE DISTRIBUTION

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

```sql
SELECT
    ROUND(AVG(Quantity), 0) AS Avg_Pizza_Ordered_Per_Day
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS Quantity
    FROM
        pizza.orders o
    JOIN pizza.order_details od ON o.Order_ID = od.order_id
    GROUP BY o.order_date) AS Order_Qty;
```

SQL QUERY

Result Grid | Filter Rows:

	Avg_Pizza_Ordered_Per_Day
▶	138

OUTPUT

AVERAGE ORDERED PER DAY

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

```sql
SELECT
    pt.name, SUM(od.quantity * p.price) Total_Revenue
FROM
    pizza.pizza_types pt
        JOIN
    pizza.pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    pizza.order_details od ON p.Pizza_ID = od.Pizza_ID
GROUP BY pt.name
ORDER BY Total_Revenue DESC
LIMIT 3;
```

SQL QUERY

3 MOST ORDERED PIZZA

Result Grid | Filter Rows:

name	Total_Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

OUTPUT

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

```sql
SELECT
    pt.category,
    ROUND(SUM(od.quantity * p.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                            2) AS Revenue
                FROM
                    pizza.order_details
                        JOIN
                    pizza.pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
            2) AS revenue
FROM
    pizza.pizza_types pt
        JOIN
    pizza.pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    pizza.order_details od ON p.pizza_id = od.Pizza_ID
GROUP BY pt.category
ORDER BY Revenue DESC;
```

SQL QUERY

Result Grid | Filter

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

OUTPUT

PERCENTAGE BY CATEGORY

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
SELECT
  order_date,
  Sum(revenue) OVER (
    ORDER BY
      order_date
  ) AS Cum_Revenue
FROM
  (
    SELECT
      o.order_date AS Order_Date,
      Sum(od.quantity * p.price) AS Revenue
    FROM
      pizza.orders o
      JOIN pizza.order_details od ON o.order_id = od.order_id
      JOIN pizza.pizzas p ON od.pizza_id = p.pizza_id
    GROUP BY
      o.order_date
  ) AS Sales;
```

SQL QUERY

Result Grid Filter Rows:

order_date	Cum_Revenue
2015-01-01	2713.850000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006

OUTPUT

CUMULATIVE REVENUE OVER TIME

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

```sql
select
  category,
  Name,
  Revenue
from
  (
    select
      Category,
      Name,
      Revenue,
      rank() over(
        partition by category
        order by
          Revenue desc
      ) as rn
    from
      (
        select
          pt.category as Category,
          pt.name as Name,
          sum(
            (od.quantity) * p.price
          ) as Revenue
        from
          pizza.pizza_types pt
          join pizza.pizzas p on pt.pizza_type_id = p.pizza_type_id
          join pizza.order_details od on p.pizza_id = od.Pizza_ID
        group by
          category,
          name
      ) as A
  ) as B
where
  rn <= 3;
```

SQL QUERY

3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

Result Grid | Filter Rows: | Export:

category	Name	Revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

OUTPUT


THANK YOU!