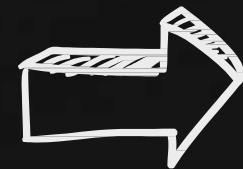




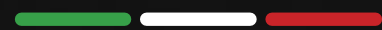
PIZZA SALES ANALYSIS

LET'S GO



BY MOULI KANNAN S S

Welcome To The Pizza-Sales Analysis



Hi folks,
My name is **Mouli Kannan S S**. I am excited to share my latest project on **Pizza Sales Analysis using SQL**.

Through this project, I was able to uncover valuable insights and trends within the data.

I hope you find this analysis as insightful and engaging as I did.
Thank you for taking the time to explore my work!



Learn More



Schema of Pizza Sales

Tables

Pizzas

This table contains columns such as pizza_id, pizza_type_id, price and size of the pizzas.

Pizza_types

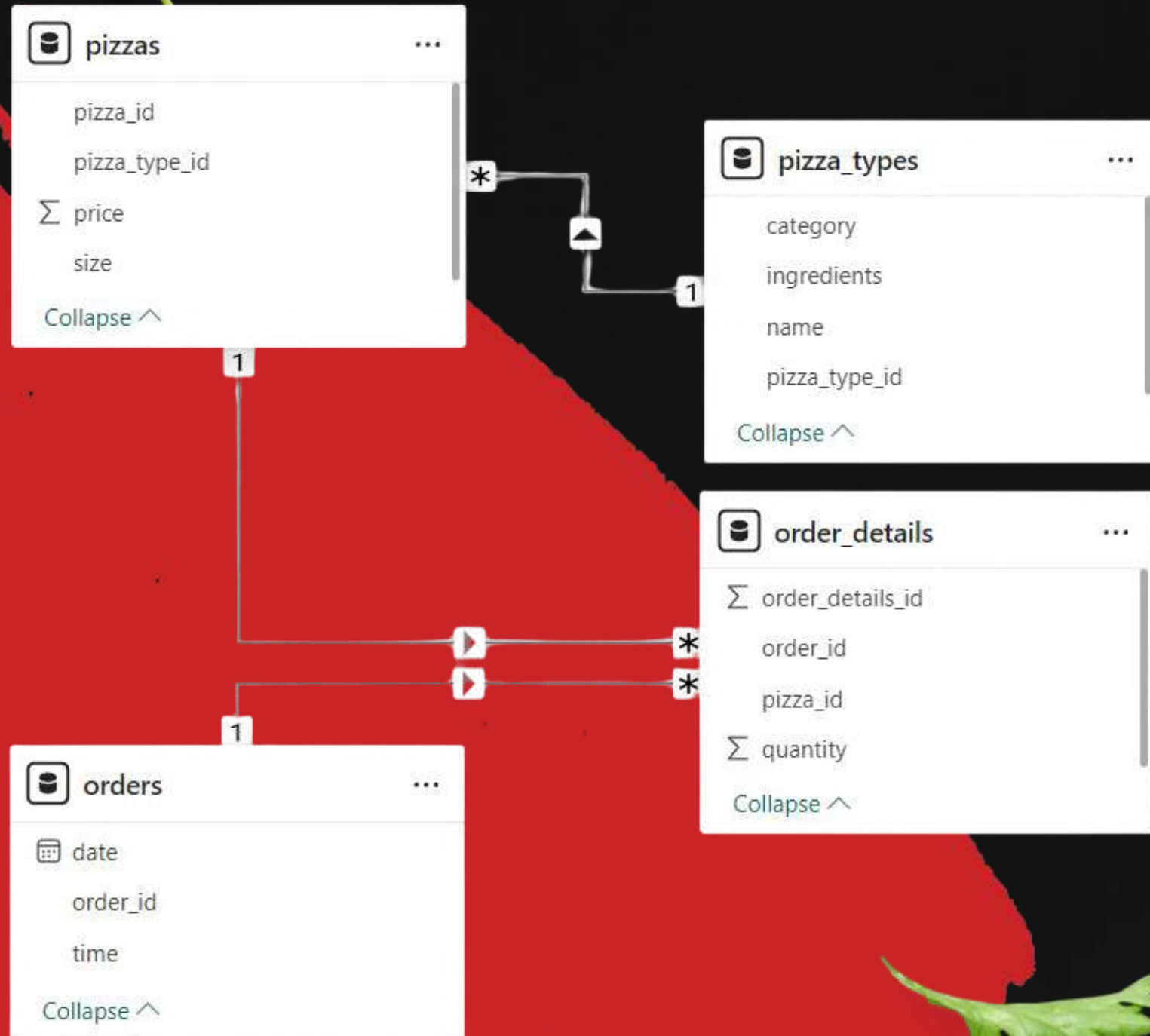
This table contains columns such as category, ingredients, name and pizza_type_id of the pizzas.

Order_details

This table contains columns such as order_details_id, order_id, pizza_id, quantity of the pizzas.

Orders

This table contains the date, order_id and time of the pizza orders.



Questions



Basic:

- Q) Retrieve the total number of orders placed.
- Q) Calculate the total revenue generated from pizza sales.
- Q) Identify the highest-priced pizza.
- Q) Identify the most common pizza size ordered.
- Q) List the top 5 most ordered pizza types along with their quantities.

Intermediate:

- Q) Join the necessary tables to find the total quantity of each pizza category ordered.
- Q) Determine the distribution of orders by hour of the day.
- Q) Join relevant tables to find the category-wise distribution of pizzas.
- Q) Group the orders by date and calculate the average number of pizzas ordered per day.
- Q) Determine the top 3 most ordered pizza types based on revenue.

Advanced:

- Q) Calculate the percentage contribution of each pizza type to total revenue.
- Q) Analyze the cumulative revenue generated over time.
- Q) Determine the top 3 most ordered pizza types based on revenue for each pizza category.

LET'S START

Q) Retrieve the total number of orders placed.

Query :

```
-- Q) Retrieve the total number of orders placed.  
Select count(order_id) as total_sales from pizzasales.orders;
```

Result :

Result Grid	
	total_sales
▶	21350

Q) Calculate the total revenue generated from pizza sales.

Query :

```
-- Q) Calculate the total revenue generated from pizza sales.  
SELECT  
ROUND(SUM(orders_details.quantity * pizzas.price),2)as total_revenue  
FROM orders_details JOIN pizzas  
ON orders_details.pizza_id = pizzas.pizza_id;
```

Result :

Result Grid			
	total_revenue		
▶	817860.05		





Q) Identify the highest-priced pizza.

Query :

```
-- Q) Calculate the total revenue generated from pizza sales.
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
          2) AS total_revenue
FROM
    orders_details
    JOIN
    pizzas ON orders_details.pizza_id = pizzas.pizza_id;
```

Result :

Result Grid			Filter Rows:
	highest_priced_pizza	price	
▶	The Greek Pizza	35.95	





Q) Identify the most common pizza size ordered.

Query :

```
-- Identify the most common pizza size ordered.  
SELECT  
    pizzas.size,  
    COUNT(orders_details.order_details_id) AS Total_Orders_Count  
FROM  
    orders_details  
    JOIN  
    pizzas ON orders_details.pizza_id = pizzas.pizza_id  
GROUP BY pizzas.size  
ORDER BY Total_Orders_Count DESC;
```

Result :

	size	Total_Quantity
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



Q) List the top 5 most ordered pizza types along with their quantities.

Query :

```
-- List the top 5 most ordered pizza types along with their quantities.
SELECT
    pizza_types.name AS Pizza_names,
    SUM(orders_details.quantity) AS Quantity
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY Pizza_names
ORDER BY Quantity DESC
LIMIT 5;
```

Result :

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

Q) Join the necessary tables to find the total quantity of each pizza category ordered.

Query :

```
-- Join the necessary tables to find the total quantity of each pizza category ordered. p1.category
SELECT
  p1.category AS pizza_category,
  SUM(p3.quantity) AS total_quantity
FROM
  pizza_types p1
  JOIN
  pizzas p2 ON p1.pizza_type_id = p2.pizza_type_id
  JOIN
  orders_details p3 ON p3.pizza_id = p2.pizza_id
GROUP BY pizza_category
ORDER BY total_quantity DESC;
```

Result :

pizza_category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q) Determine the distribution of orders by hour of the day.

Query :

```
-- Determine the distribution of orders by hour of the day.  
SELECT  
    HOUR(order_time) AS Hour_of_the_Day,  
    COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY Hour_of_the_Day;
```

Result :

Hour_of_the_Day	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

Q) Join relevant tables to find the category-wise distribution of pizzas.

Query :

```
-- Join relevant tables to find the category-wise distribution of pizzas.  
SELECT  
    category, COUNT(pizza_types.pizza_type_id)  
FROM  
    pizza_types  
GROUP BY category;
```

Result :

category	Distribution
Chicken	6
Classic	8
Supreme	9
Veggie	9

Q) Group the orders by date and calculate the average number of pizzas ordered per day.



Query :

```
-- Group the orders by date and calculate the average number of pizzas ordered per day.
SELECT
    ROUND(AVG(quantity), 0) AS Daily_avg_pizzas
FROM
    (SELECT
        o1.order_date, SUM(o2.quantity) AS quantity
    FROM
        orders o1
    JOIN orders_details o2 ON o1.order_id = o2.order_id
    GROUP BY o1.order_date) AS order_quantity
```

Result :

Daily_avg_pizzas
138

Q) Determine the top 3 most ordered pizza types based on revenue.

Query :

```
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
SELECT name, category, revenue from
(select category, name, revenue,
rank() over (partition by category order by revenue desc) as rn
from
(SELECT
  p1.name,
  p1.category,
  SUM(o1.quantity * p2.price) AS revenue
FROM
  pizza_types p1
  JOIN
  pizzas p2 ON p1.pizza_type_id = p2.pizza_type_id
  JOIN
  orders_details o1 ON o1.pizza_id = p2.pizza_id
GROUP BY p1.name , p1.category) as a ) as b
where rn<=3 ;
```

Result :

Pizza_names	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Q) Calculate the percentage contribution of each pizza type to total revenue.

Query :

```
-- Calculate the percentage contribution of each pizza type to total revenue.
SELECT
  p1.pizza_type_id AS Pizza_types,
  ROUND(SUM(p1.price * o1.quantity) / (SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
      2) AS total_revenue
    FROM
      orders_details
      JOIN
        pizzas ON orders_details.pizza_id = pizzas.pizza_id) * 100,
    2) AS revenue
FROM
  pizzas p1
  JOIN
    orders_details o1 ON p1.pizza_id = o1.pizza_id
GROUP BY p1.pizza_type_id;
```

Result :

Pizza_types	revenue
hawaiian	3.95
classic_dlx	4.67
five_cheese	3.19
ital_supr	4.09
mexicana	3.27
thai_dkn	5.31
...	...



Q) Analyze the cumulative revenue generated over time.

Query :

```
-- Analyze the cumulative revenue generated over time.
Select dates, round(sum(revenue) over (order by dates),2) as cum_revenue
from
  (SELECT
    o1.order_date AS dates,
    SUM(o2.quantity * p1.price) AS revenue
  FROM
    orders o1
    JOIN
    orders_details o2 ON o1.order_id = o2.order_id
    JOIN
    pizzas p1 ON p1.pizza_id = o2.pizza_id
  GROUP BY dates) as sales;
```

Result :

dates	cum_revenue
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6
2015-01-19	43365.75
2015-01-20	45763.65
2015-01-21	47804.2
2015-01-22	50300.9
2015-01-23	52724.6
2015-01-24	55013.85
2015-01-25	56631.4
2015-01-26	58515.8
2015-01-27	61043.85
2015-01-28	63059.85
2015-01-29	65105.15
2015-01-30	67375.45
2015-01-31	69793.3
2015-02-01	72982.5
2015-02-02	75311.1

Q) Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Query :

```
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
SELECT name, category, revenue from
(select category, name, revenue,
rank() over (partition by category order by revenue desc) as rn
from
(SELECT
    p1.name,
    p1.category,
    SUM(o1.quantity * p2.price) AS revenue
FROM
    pizza_types p1
    JOIN
    pizzas p2 ON p1.pizza_type_id = p2.pizza_type_id
    JOIN
    orders_details o1 ON o1.pizza_id = p2.pizza_id
GROUP BY p1.name , p1.category) as a ) as b
where rn<=3 ;
```

Result :

name	category	revenue
The Thai Chicken Pizza	Chicken	43434.25
The Barbecue Chicken Pizza	Chicken	42768
The California Chicken Pizza	Chicken	41409.5
The Classic Deluxe Pizza	Classic	38180.5
The Hawaiian Pizza	Classic	32273.25
The Pepperoni Pizza	Classic	30161.75
The Spicy Italian Pizza	Supreme	34831.25
The Italian Supreme Pizza	Supreme	33476.75
The Sicilian Pizza	Supreme	30940.5
The Four Cheese Pizza	Veggie	32265.700000000065
The Mexicana Pizza	Veggie	26780.75
The Five Cheese Pizza	Veggie	26066.5

Thank You

— — —

Thank you for taking the time to review my project on Pizza Sales Analysis using SQL. I hope you found the insights as interesting and valuable as I did.

Feel free to connect with me on [LinkedIn](#). I look forward to connecting with you!

