MAKE MORE DOUGH WITH THE RIGHT PIZZA QUERY!

SQL PROJECT ON PIZZA SALES ANALYSIS

ABOUT PIZZA PROJECT

The goal of this project is to create a pizza sales analysis system using SQL.By organizing and querying sales data, we can generate insights that help businesses understand customer preferences, optimize pricing, and improve sales performance.

ABOUT DATA

ORDER DETAILS

Contains information about individual pizza order including quantity and price

PIZZAS

Stores details about different pizza variation such as pizza type and size

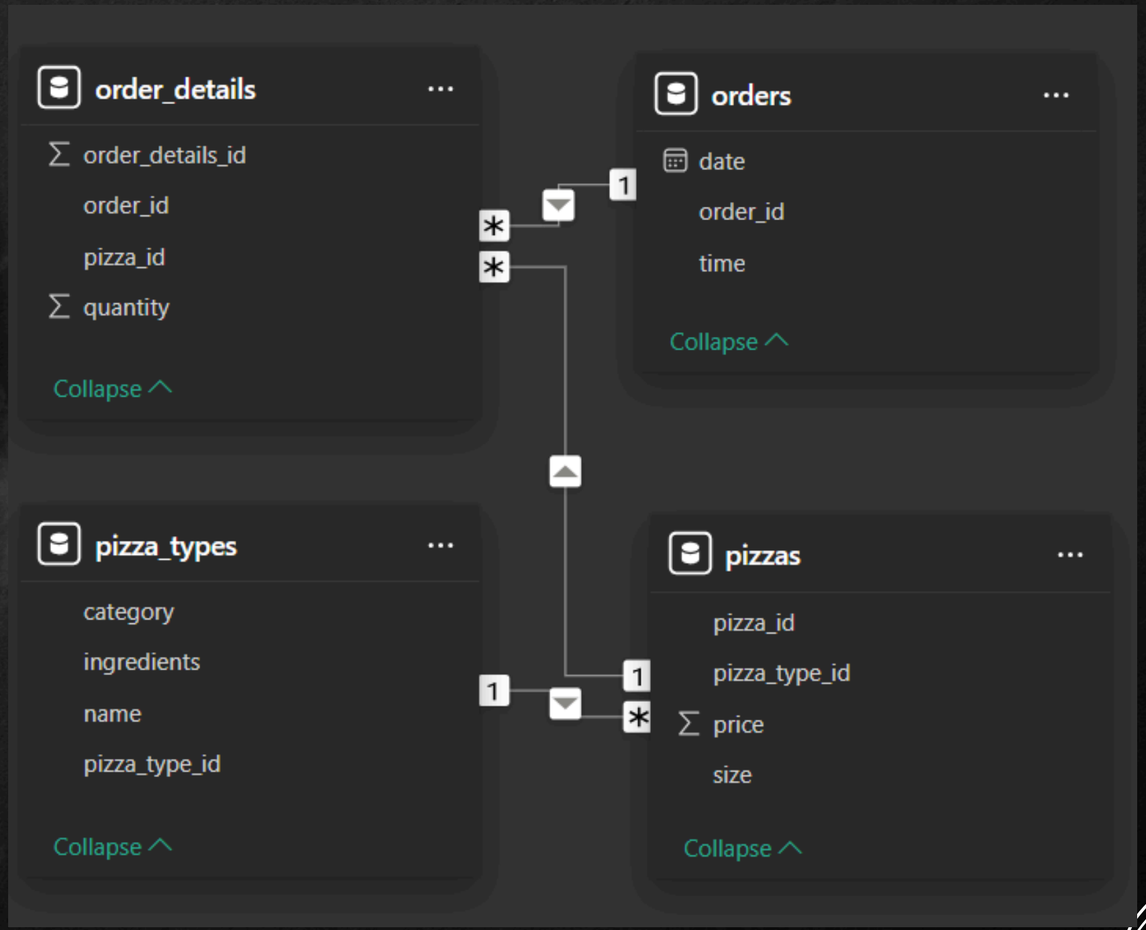
ORDERS

Captures order specific time such as order date and time

PIZZA TYPES

Provides information about different pizza type including ingredienyts and category

ABOUT SCHEMA



1.LIST OF MOST COMMON PIZZA SIZE ORDERED

```sql
SELECT
    pizzas.size, SUM(order_details.quantity) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY SUM(order_details.quantity) DESC
```

	size	order_count
▶	L	18956
	M	15635
	S	14403
	XL	552
	XXL	28

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

```sql
SELECT
    (pizza_types.name) AS pizza_name,
    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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_name
ORDER BY quantity DESC
LIMIT 5;
```

pizza_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

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

```sql
SELECT
    ROUND(AVG(quantity), 2) AS average
FROM
    (SELECT
        orders.Order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.Order_id = order_details.order_id
    GROUP BY orders.Order_date) AS order_quantity
```

	average
▶	138.47

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

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
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.name
ORDER BY revenue DESC
LIMIT 3
```

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

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

```
select (pizza_types.category) as category,
round(sum(order_details.quantity*pizzas.price) / (select ROUND(sum(order_details.quantity * pizzas.price),2)
AS total_sales from order_details
join pizzas on order_details.pizza_id=pizzas.pizza_id)*100,2) as revenue
from pizza_types join pizzas on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details on pizzas.pizza_id=order_details.pizza_id group by category order by revenue desc
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

8.ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
select order_date,sum(revenue) over(order by Order_date) as cum_revenue from
(select orders.Order_date,sum(order_details.quantity*pizzas.price) as revenue from orders join
order_details on orders.Order_id=order_details.order_id join pizzas on
pizzas.pizza_id=order_details.pizza_id group by orders.Order_date) as sales
```

order_date	cum_revenue
2015-01-01	2713.8500000000004
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

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

```
select category,name,revenue,rank() over(partition by category order by revenue desc) as Rank_no from
(select pizza_types.category,pizza_types.name,sum(order_details.quantity*pizzas.price) as revenue
from pizza_types join pizzas on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details on pizzas.pizza_id=order_details.pizza_id group by pizza_types.category,pizza_types.name)as a
```

category	name	revenue	Rank_no
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Chicken	The Southwest Chicken Pizza	34705.75	4
Chicken	The Chicken Alfredo Pizza	16900.25	5
Chicken	The Chicken Pesto Pizza	16701.75	6
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Classic	The Greek Pizza	28454.100000000013	4

ONLY A FIRST FEW ROWS OUTPUTS ARE ATTACHED FOR THE 6 AND 7 QUERIES

FINDINGS

- BEST SELLING PIZZAS

Classic deluxe pizza type and category chicken generated higher revenue

- ORDER SIZE PREFERENCE

Large-sized pizzas were the most popular, followed by medium,small while XL,XXL sized pizzas had lower demand.

- ORDER CATEGORY PREFERENCE

Classic pizzas followed by supreme, chicken had higher revenue while veggie category had lower revenue

CONCLUSION

With data-driven decision-making, businesses can enhance sales performance, improve customer satisfaction, and maximize profits. This project demonstrates how SQL can transform raw data into actionable insights, helping businesses make smarter, more profitable choices.

THANK YOU!