

PIZZA SALES ANALYSIS AND INSIGHTS

SALES REPORT

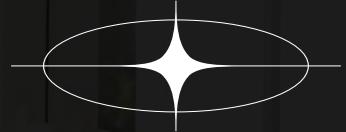
presented by
Suryakant



ABOUT ME

I am a data enthusiast with a strong foundation in SQL and database management. My expertise lies in analyzing complex datasets and solving business-driven queries to extract actionable insights. For this project, I utilized advanced SQL techniques to address a wide range of analytical questions, showcasing my ability to manipulate and interpret relational data effectively.

The work presented here demonstrates my problem-solving approach, from basic queries like calculating total revenue and identifying popular pizza categories to advanced analytics such as revenue contribution percentages and category-specific insights. My passion for database querying drives me to explore efficient and innovative solutions to uncover hidden patterns in data.



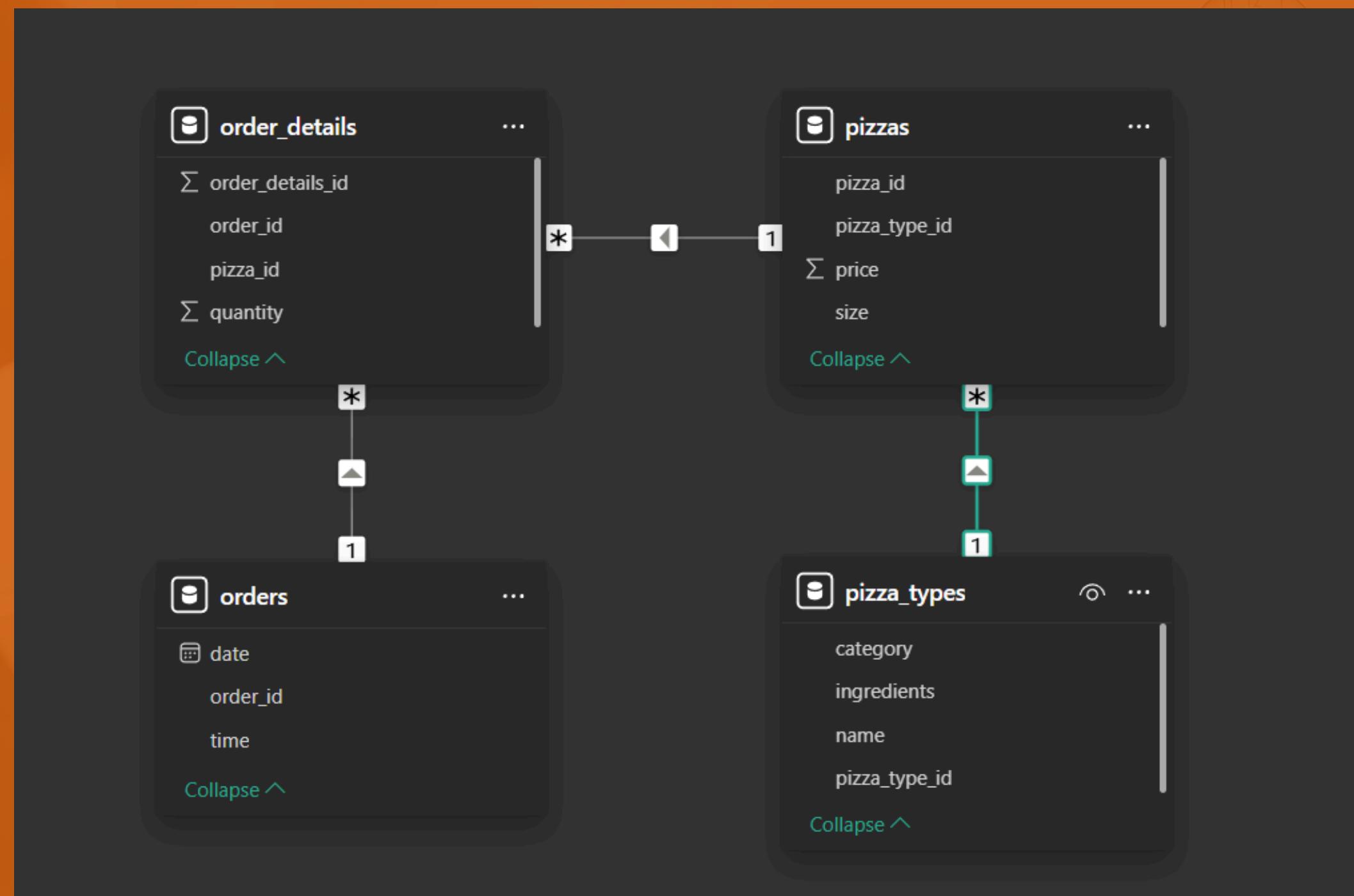
INTRODUCTION

In this analysis, we aim to derive actionable insights from pizza sales data by exploring basic, intermediate, and advanced queries. The goal is to understand customer preferences, revenue distribution, and category-specific performance. By processing and analyzing the given datasets (orders.csv, order_details.csv, pizzas.csv, and pizza_types), we provide a comprehensive evaluation of sales performance and customer behavior.



presented by
Suryakant

PIZZA SALES DATABASE SCHEMA OVERVIEW



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT count(*) AS Total_orders FROM orders;
```

Result:

Total Orders Placed: 21,350

This indicates that there have been a total of 21,350 orders placed as recorded in the orders table of the database.



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT sum(a.price * b.quantity) AS Total_sales FROM pizzas AS  
a JOIN orders_details AS b ON a.pizza_id = b.pizza_id;
```

This query is used to calculate the total revenue generated from pizza sales by multiplying the price of each pizza with the quantity ordered and then summing those values.

IDENTIFY THE HIGHEST-PRICED PIZZA.

```
-- Identify the highest-priced pizza.
```

```
SELECT a.name, max(b.price) AS pizza_price  
FROM pizza_types AS a  
JOIN pizzas AS b ON a.pizza_type_id = b.pizza_type_id  
GROUP BY a.name  
ORDER BY pizza_price DESC  
LIMIT 1;
```

Result Grid		
	name	pizza_price
▶	The Greek Pizza	35.95

This query retrieves the highest-priced pizza from the pizzas table, grouped by pizza type.

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
-- Identify the most common pizza size ordered.  
select a.size, count(b.quantity) as no_of_pizzas  
from orders_details as b join pizzas as a on a.pizza_id=b.pizza_id  
group by a.size order by no_of_pizzas desc;
```

Result Grid | Filter

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

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

```
SELECT pizza_types.name,sum(orders_details.quantity)as TOTAL_QTY  
FROM orders_details  
JOIN pizzas ON pizzas.pizza_id=orders_details.pizza_id  
JOIN pizza_types ON pizza_types.pizza_type_id= pizzas.pizza_type_id  
group by pizza_types.name  
order by TOTAL_QTY DESC  
LIMIT 5;
```

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

This query is used to retrieve the top 5 most ordered pizza types along with their total quantities ordered.

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA

```
select pizza_types.category ,sum(orders_details.quantity) as Total_qty_by_category  
from pizza_types join pizzas on pizzas.pizza_type_id = pizza_types.pizza_type_id  
join orders_details on orders_details.pizza_id=pizzas.pizza_id  
group by pizza_types.category  
order by Total_qty_by_category desc;
```

Result Grid		
	category	Total_qty_by_category
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

This provides the total quantity ordered for each pizza category, sorted by the most popular category.

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

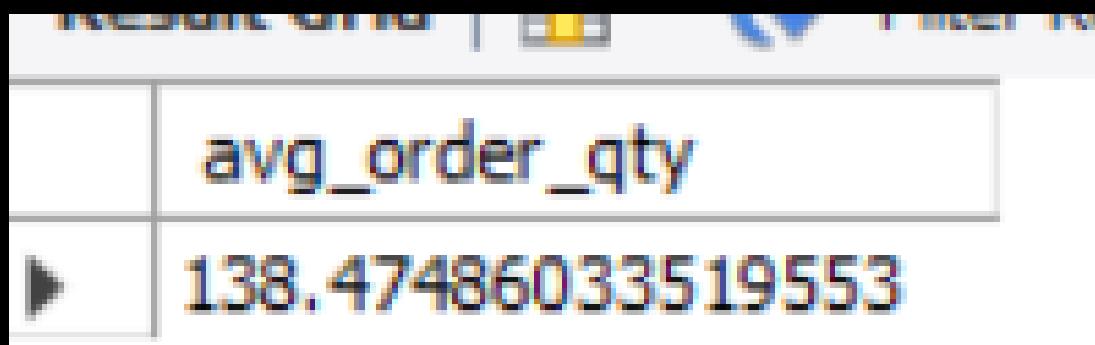
```
-- Determine the distribution of orders by hour of the day.  
select hour(order_time), count(order_id) as orders  
from orders  
group by hour(order_time)  
order by orders desc
```

	hour(order_time)	orders
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1472
	15	1468

This query is used to determine the distribution of orders by the hour of the day, i.e., how many orders were placed during each hour.

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

```
-- Group the orders by date and calculate the average number of pizzas ordered per day.  
select avg(total_orders) as avg_order_qty from  
(SELECT DATE(order_date) AS order_date, sum(orders_details.quantity) AS total_orders  
FROM orders join orders_details on orders_details.order_id=orders.order_id  
GROUP BY orders.order_date) as qty;
```



A screenshot of a MySQL command-line interface. The command entered is the same as the one above, calculating the average number of pizzas ordered per day. The result is displayed in a table with two rows. The first row is a header labeled 'avg_order_qty'. The second row contains the value '138.47486033519553'.

	avg_order_qty
▶	138.47486033519553

This query is used to calculate the average number of pizzas ordered per day.

DISTRIBUTION OF PIZZA TYPES BY CATEGORY

```
SELECT category, COUNT(name) AS total_count  
FROM pizza_types  
GROUP BY category  
ORDER BY total_count DESC;
```

category	total_count
Supreme	9
Veggie	9
Classic	8
Chicken	6

This query calculates the total count of pizza types for each category and orders the results in descending order of the count.

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

```
-- Determine the top 3 most ordered pizza types based on revenue.  
select pizza_types.name,sum(pizzas.price * orders_details.quantity) as revenue  
from orders_details  
join pizzas on pizzas.pizza_id = orders_details.pizza_id  
join pizza_types on pizza_types.pizza_type_id=pizzas.pizza_type_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

This query calculates the top 3 most ordered pizza types based on revenue.

TOP 3 PIZZA CATEGORIES BY REVENUE AND THEIR CONTRIBUTION TO TOTAL SALES

```
SELECT
    pizza_types.category,
    SUM(pizzas.price * orders_details.quantity) AS revenue,
    ((SUM(pizzas.price * orders_details.quantity) * 100.0) /
    (SELECT SUM(pizzas.price * orders_details.quantity)
     FROM orders_details
     JOIN pizzas ON pizzas.pizza_id = orders_details.pizza_id)) AS contribution_percentage
FROM orders_details
JOIN pizzas ON pizzas.pizza_id = orders_details.pizza_id
JOIN pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category
ORDER BY revenue DESC
LIMIT 3;
```

	category	revenue	contribution_percentage
▶	Classic	220053.1000000001	26.9059602556699
	Supreme	208196.99999999822	25.45631126009884
	Chicken	195919.5	23.955137556847493

This query calculates the top 3 pizza categories based on revenue, along with their contribution percentage to the total revenue.

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

```
-- 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) as ranks
from
(select pizza_types.name, pizza_types.category, sum(pizzas.price*orders_details.quantity) as revenue
from pizzas
join orders_details on orders_details.pizza_id = pizzas.pizza_id
join pizza_types on pizza_types.pizza_type_id = pizzas.pizza_type_id
group by pizza_types.category, pizza_types.name
order by revenue) as a
```

	category	name	revenue	ranks
▶	Chicken	The Chicken Pesto Pizza	16701.75	1
	Chicken	The Chicken Alfredo Pizza	16900.25	2
	Chicken	The Southwest Chicken Pizza	34705.75	3
	Chicken	The California Chicken Pizza	41409.5	34705.75
	Chicken	The Barbecue Chicken Pizza	42768	5

This query calculates the top 3 most ordered pizza types based on revenue for each pizza category, using ranking.

THANK YOU!

