

### Purpose and Overview of Analysis

The purpose of this analysis is to gain insights into the sales performance of the pizza store. By examining the data, the aim is to identify trends, uncover opportunities for growth, and make informed business decisions to enhance our operational efficiency and increase revenue.



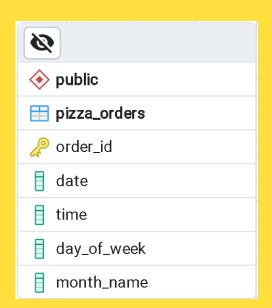
#### **Dataset Structure**



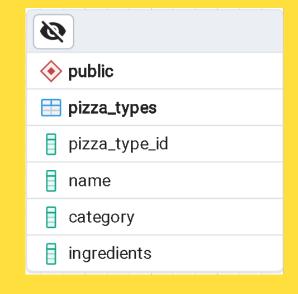
#### order\_details



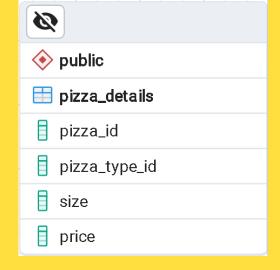
#### pizza\_orders



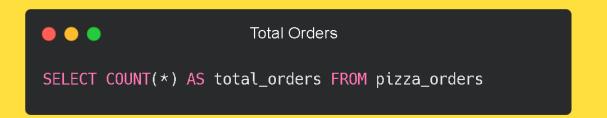
#### pizza\_types



#### pizza\_details



### 1. How many total orders have been placed?



	total_orders bigint	
1	21350	



## 2. How many orders were placed on each date?



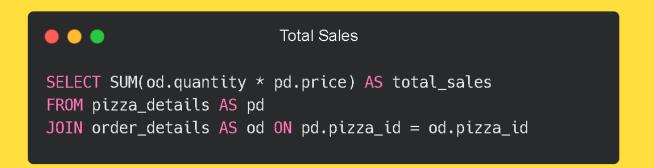
Order Frequency

SELECT date, day\_of\_week, COUNT(\*) AS order\_count FROM pizza\_orders
GROUP BY date, day\_of\_week ORDER BY order\_count DESC



	date date	day_of_week character varying (10)	order_count bigint
1	2022-11-27	Sunday	115
2	2022-11-26	Saturday	113
3	2022-10-15	Saturday	107
4	2022-07-04	Monday	105
5	2022-05-15	Sunday	94
6	2022-07-03	Sunday	93
7	2022-02-01	Tuesday	87
8	2022-10-01	Saturday	84
9	2022-07-17	Sunday	82
10	2022-08-14	Sunday	81
11	2022-09-11	Sunday	77
12	2022-12-18	Sunday	76

## 3. What is the total sales generated from all orders?



	total_sales numeric	
1	817860.05	



## 4. What is the average value of an order?

```
WITH order_values AS (
    SELECT po.order_id, SUM(od.quantity * pd.price) AS
order_value
    FROM pizza_orders po
    JOIN order_details od ON po.order_id = od.order_id
    JOIN pizza_details pd ON od.pizza_id = pd.pizza_id
    GROUP BY po.order_id
)
SELECT ROUND(AVG(order_value),2) AS average_order_value
FROM order_values
```

	average_order_value numeric	
1	38.31	



## 5. What is the daily revenue generated?

```
Daily Revenue

SELECT po.date, SUM(od.quantity * pd.price) AS
daily_revenue
FROM pizza_orders po
JOIN order_details od ON po.order_id = od.order_id
JOIN pizza_details pd ON od.pizza_id = pd.pizza_id
GROUP BY po.date
ORDER BY po.date;
```



	date date	daily_revenue numeric
1	2022-01-01	2713.85
2	2022-01-02	2731.90
3	2022-01-03	2662.40
4	2022-01-04	1755.45
5	2022-01-05	2065.95
6	2022-01-06	2428.95
7	2022-01-07	2202.20
8	2022-01-08	2838.35
9	2022-01-09	2127.35
10	2022-01-10	2463.95

# 6. What is the revenue generated from each type of pizza?

name

```
SELECT pt.name, SUM(od.quantity * pd.price) AS
total_revenue
FROM pizza_types pt
JOIN pizza_details pd ON pt.pizza_type_id =
pd.pizza_type_id
JOIN order_details od ON pd.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY total_revenue DESC
```

	character varying (100)	numeric
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768.00
3	The California Chicken Pizza	41409.50
4	The Classic Deluxe Pizza	38180.50
5	The Spicy Italian Pizza	34831.25
6	The Southwest Chicken Pizza	34705.75
7	The Italian Supreme Pizza	33476.75
8	The Hawaiian Pizza	32273.25
9	The Four Cheese Pizza	32265.70
10	The Sicilian Pizza	30940.50
11	The Pepperoni Pizza	30161.75

total revenue



## 7. Rank pizzas based on their popularity

```
SELECT name,

COUNT(order_details_id) AS order_count,

RANK() OVER (ORDER BY COUNT(order_details_id)

DESC) AS popularity_rank

FROM pizza_types

JOIN pizza_details ON pizza_types.pizza_type_id = pizza_details.pizza_type_id

JOIN order_details ON pizza_details.pizza_id = order_details.pizza_id

GROUP BY name
```



name character varying (100)	order_count bigint	popularity_rank bigint
The Classic Deluxe Pizza	2416	1
The Barbecue Chicken Pizza	2372	2
The Hawaiian Pizza	2370	3
The Pepperoni Pizza	2369	4
The Thai Chicken Pizza	2315	5
The California Chicken Pizza	2302	6
The Spicy Italian Pizza	1887	7
The Sicilian Pizza	1887	7
The Southwest Chicken Pizza	1885	9
The Four Cheese Pizza	1850	10
The Italian Supreme Pizza	1849	11

## 8. Compare sales between weekdays and weekends

```
Compare Sales

SELECT

CASE

WHEN day_of_week IN ('Saturday', 'Sunday') THEN 'Weekend'

ELSE 'Weekday'

END AS day_type,

SUM(od.quantity * pd.price) AS total_sales

FROM pizza_orders po

JOIN order_details od ON po.order_id = od.order_id

JOIN pizza_details pd ON od.pizza_id = pd.pizza_id

GROUP BY day_type
```

	day_type text	total_sales numeric
1	Weekend	259602.40
2	Weekday	558257.65



# 9. Compare the sales and revenue generated by different pizza sizes



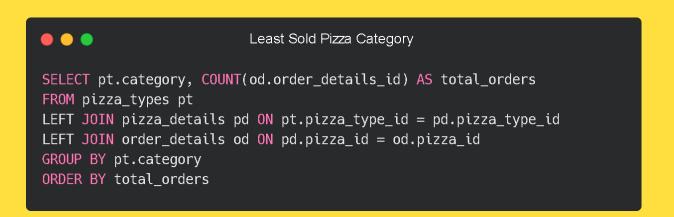
Pizza Size Analysis

SELECT size, COUNT(od.quantity) AS total\_pizzas\_sold,
SUM(od.quantity \* pd.price) AS total\_revenue
FROM order\_details od
JOIN pizza\_details pd ON od.pizza\_id = pd.pizza\_id
GROUP BY size

	size character varying (10)	total_pizzas_sold bigint	total_revenue numeric
1	S	14137	178076.50
2	XXL	28	1006.60
3	XL	544	14076.00
4	М	15385	249382.25
5	L	18526	375318.70



### 10. Which category of pizza is least selling?



	category character varying (50)	total_orders bigint
1	Chicken	10815
2	Veggie	11449
3	Supreme	11777
4	Classic	14579



#### Conclusion

- •Average Order Value: The average order value provides insight into typical customer spending, which is around \$38.31.
- •Sales Comparison: Total sales on weekdays are almost double those on weekends.
- •Pizza Size Analysis: Large-sized pizzas contribute the most to revenue, indicating a strong customer preference for this size. On the other hand, extra-large pizzas are the least popular.
- •Popular Pizza Types: Thai Chicken pizza is among the top sellers, suggesting a strong customer preference for this flavor.
- •Least Sold Categories: Chicken pizzas have the lowest sales among all categories.