



# PIZZA SALES REPORTS

02 May, 2024



# INTRODUCTION

Pizza sales analysis using SQL allows businesses to gain valuable insights from their sales data through querying and data manipulation. By leveraging SQL's powerful capabilities, such as filtering, aggregating, and joining tables, businesses can analyze trends, track sales performance, and identify customer preferences. This analysis helps in understanding which pizza types are most popular, identifying peak sales periods, and evaluating the effectiveness of promotions. By extracting and analyzing key data points from sales transactions, businesses can make data-driven decisions to optimize their operations, improve customer satisfaction, and increase profitability.

# AGENDA

01

Retrive the total number of order placed

02

calculate total revenue generate from pizza sales

03

dentify the highest price pizza

04

identify the most common pizza size ordered.

05

list the top 5 most ordered pizza types along with their quantitites

06

join the neccesary tables to find the total quantity of each pizza category

07

detremine the distribution of orders by hour of the day

08

join relavent tables to find the category wise disribution of pizzas

09

Group the orders by date and calculate the average number of pizza order per day

10

determine the top 3 pizza types based on revenue

# AGENDA

- 11** calculate the percentage contribution of each pizza type to total revenue
- 12** Analyze the cumulative revenue generated over time
- 13** determine the top 3 most ordered pizza type based on revenue for each pizza category

# EXECUTIVE SUMMARY

The pizza sales analysis provides key insights into sales performance, focusing on the following aspects:

1. Pizzas:
2. Pizza Types:
3. Order Time:
4. Order Details:

By analyzing these factors, businesses can identify peak sales periods, optimize their menu offerings, and tailor promotions to specific customer preferences, ultimately boosting sales and customer satisfaction.

## **1.Retrieve the total number of order placed**

```
SELECT  
    COUNT(order_id) AS TOTAL_ORDER  
FROM  
    orders
```

## **OUTPUT**

	TOTAL_ORDER
▶	21350



## 2.Calculate total revenue generate from pizza sales

```
SELECT
    SUM(pizzas.price * order_details.quantity) AS TOTAL_REVENUE
FROM
    pizzas
    JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
```

## OUTPUT

TOTAL_REVENUE
817860.049999993



### **3. Identify the highest price pizza**

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

### **OUTPUT**

	name	price
▶	The Greek Pizza	35.95

#### **4. Identify the most common pizza size ordered.**

```
SELECT
    pizzas.size,
    COUNT(order_details.order_detail_id) AS order_count
FROM
    pizzas
    JOIN
        order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

#### **OUTPUT**

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



## **5.List the top 5 most ordered pizza types along with their quantitites**

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS total_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_types.name
ORDER BY total_quantity DESC
LIMIT 5;
```

## **OUTPUT**

	<b>name</b>	<b>total_quantity</b>
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



## **6.Join the necessary tables to find the total quantity of each pizza category**

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_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_types.category;
```

## **OUTPUT**

	category	total_quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050



## **7.Determine the distribution of orders by hour of the day**

```
SELECT  
    HOUR(order_time) AS hour, COUNT(order_id)  
FROM  
    orders  
GROUP BY hour;
```

## **OUTPUT**

	hour	count(order_id)
▶	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



## 8.Join relevant tables to find the category wise distribution of pizzas

```
SELECT
    category, COUNT(pizza_type_id) AS total_count
FROM
    pizza_types
GROUP BY category;
```

## OUTPUT

	category	total_count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



## **9. Group the orders by date and calculate the average number of pizza order per day.**

```
SELECT
    ROUND(AVG(quantity), 0) AS avg_quantity
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_per_day;
```

## **OUTPUT**

	avg_quantity
▶	138



## **10.Determine the top 3 pizza types based on revenue**

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS total_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 total_revenue DESC
```

## **OUTPUT**

	name	total_revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5



## 11.calculate the percentage contribution of each pizza type to total revenue

```
SELECT
pizza_types.name,
round((SUM(order_details.quantity * pizzas.price) / (SELECT
ROUND(SUM(pizzas.price * order_details.quantity),
2)
FROM
pizzas
JOIN
order_details ON pizzas.pizza_id = order_details.pizza_id)) * 100 ,3)AS Per_revenue
FROM
pizzas
JOIN
pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
JOIN
order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY name
```

### OUTPUT

	category	Per_revenue
▶	Classic	26.906
	Veggie	23.683
	Supreme	25.456
	Chicken	23.955



## **12.Analyze the cumulative revenue generated over time**

```
select order_date , round(sum(revenue) over (order by order_date),0) as cumu_revenue  
from  
(select orders.order_date as order_date,sum(order_details.quantity*pizzas.price) as revenue  
from order_details  
join pizzas  
on order_details.pizza_id=pizzas.pizza_id  
join orders  
on orders.order_id=order_details.order_id  
group by orders.order_date) as rev_table
```

### **OUTPUT**

	order_date	cumu_revenue
▶	2015-01-01	2714
	2015-01-02	5446
	2015-01-03	8108
	2015-01-04	9864
	2015-01-05	11930



### **13.Determine the top 3 most ordered pizza type based on revenue for each pizza category.**

```
select category,name , revenue from  
(select category, name , revenue, rank () over (partition by category order by revenue desc) as rn  
from  
(select pizza_types.category, pizza_types.name, sum(order_details.quantity * pizzas.price) as revenue  
from pizzas join pizza_types  
on pizzas.pizza_type_id = pizza_types.pizza_type_id  
join order_details  
on pizzas.pizza_id= order_details.pizza_id  
group by  pizza_types.category,pizza_types.name ) as cate_revenue ) as b  
where rn<=3
```

### **OUTPUT**

	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

# THANK YOU

02 May, 2024