

# PIZZA SALES REPORT

NEERAJ KUMAR

## Special Pizza

Slice into Happiness. Irresistible Pizzas,  
Delivered to Your Doorstep!

35%  
OFF



123-456-7890  
www.reallygreatsite.com  
123 Anywhere St., Any City



ORDER NOW



# INTRODUCTION

This Pizza Sales Report provides a comprehensive analysis of pizza sales, with a focus on understanding customer preferences, revenue generation, and order trends. The report is designed to answer key questions about pizza sales performance through the following analyses:

- Total Orders Placed: Identifying the complete number of orders during the period.
- Revenue Calculation: Determining the total revenue generated from pizza sales, offering insights into the overall financial performance.
- Pricing Analysis: Identifying the highest-priced pizza on the menu and analyzing its contribution to overall sales.
- Order Trends:
  - Discovering the most commonly ordered pizza size, which provides insights into customer preferences.
  - Listing the top 5 most popular pizza types based on order quantities, helping to understand the best-selling items.

- 
- **Category and Distribution Analysis:**
  - Analyzing category-wise pizza distribution to understand how different pizza types perform within their categories.
  - Breaking down order patterns by time, including hourly distribution, to understand peak ordering times.
  - **Daily Trends:** Grouping orders by date to calculate the average number of pizzas sold per day, helping to identify any fluctuations in daily demand.
  - **Revenue Contribution:**
  - Identifying the top 3 most-ordered pizza types based on revenue.
  - Calculating the percentage contribution of each pizza type to the total revenue, providing insights into which products drive the most income.
  - **Cumulative Revenue Analysis:** Tracking how the revenue accumulates over time, offering a view of long-term financial performance.
- 

# Retrieve the total number of orders placed.

```
SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    orders;
```

## OUTPUT

	total_orders
▶	21350

# Calculate the total revenue generated from pizza sales.

```
SELECT  
    ROUND(SUM(order_details.quantity * pizzas.price),  
          0) AS total_sales  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

## OUTPUT

	total_sales
▶	817860

# Identify the highest-priced 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

# Identify the most common pizza size ordered.

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_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



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

```
SELECT
    pizza_types.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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

## OUTPUT

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

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

```
SELECT
    pizza_types.category,
    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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

## OUTPUT

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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

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

## OUTPUT

HOUR(order_time)	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	78



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

```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category
```

## OUTPUT

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Join relevant tables to find the category-wise distribution of the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
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;
```

## OUTPUT

	avg_pizza_ordered_per_day
▶	138

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

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.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;
```

## OUTPUT

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



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

```
SELECT
    pizza_types.category,
    (SUM(order_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
    FROM
        order_details
        JOIN
            pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100 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.category
ORDER BY revenue DESC;
```

## OUTPUT

category	revenue
Classic	26.90596025566967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

# 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 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 sales;
```

## OUTPUT

order_date	cum_revenue
2015-01-01	2713.85000000000004
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	23990.3500000000002
2015-01-11	25862.65
2015-01-12	27781.7

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

```
select 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 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.category, pizza_types.name) as a) as b
where rn<=3;
```

## OUTPUT

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



# CONCLUSION

This report is designed to offer a complete picture of pizza sales, helping to make informed decisions about product offerings, pricing strategies, and customer engagement.

The image features a dark maroon background with several overlapping, semi-transparent hexagonal shapes of varying sizes and orientations. These shapes create a layered, geometric effect. Scattered across the background are several small, solid maroon hexagons. The text "THANK YOU" is centered in a bold, white, sans-serif font.

**THANK YOU**