



Abhishek-dev-1

PIZZASALES REPORT

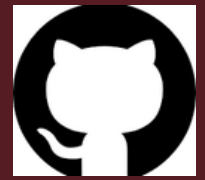


INTRODUCTION

Welcome to my Sales Report Presentation. In today's session, I will thoroughly examine our sales performance, showcasing our achievements, challenges, and strategic insights observed from the table. This presentation goes beyond numbers, analyzing and showcasing the efficacy of our sales strategies, and paving the way for future successes.

EXECUTIVE SUMMARY

In our pizza sales analysis, we utilized SQL queries to gain insights into various aspects of sales performance. We began by assessing the overall sales performance, calculating the total sales amount, the total number of pizzas sold, and the average sale amount per transaction. This provided a clear picture of revenue and transaction trends. We further examined sales trends over time by aggregating total sales by month and analyzing the number of pizzas sold each day. This helped in identifying any seasonal patterns or fluctuations in daily sales. Additionally, we evaluated sales by pizza type, summarizing total sales and quantities for each type to determine which pizzas were most popular and profitable. Finally, we identified the top-performing pizzas by listing the top 5 best-selling types based on the number of units sold. This comprehensive analysis enabled us to understand sales trends, peak periods, and product preferences, providing valuable insights for business strategy and decision-making.



Abhishek-dev-1

total number of orders placed

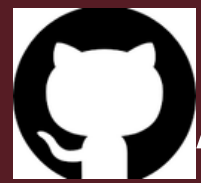
```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

| Result Grid | |
|-------------|--------------|
| | total_orders |
| ▶ | 21350 |

total revenue generated from pizza sales

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

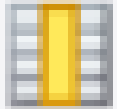

| Result Grid | |
|-------------|-------------|
| | total_sales |
| ▶ | 817860.05 |

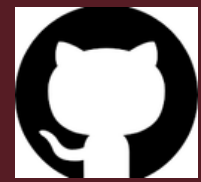


Abhishek-dev-1

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
```

| Result Grid   Filter Rows | | |
|---|-----------------|-------|
| | name | price |
| ▶ | The Greek Pizza | 35.95 |



Abhishek-dev-1

most common pizza size ordered

```
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
    JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count desc
LIMIT 1
```

| Result Grid | | | Filter Rows: |
|-------------|------|-------------|--------------|
| | size | order_count | |
| ▶ | L | 18526 | |



top 5 most ordered pizza types along with their quantities

```
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5
```

| Result Grid | | | Filter Rows: |
|-------------|----------------------------|----------|--------------|
| | 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 | |

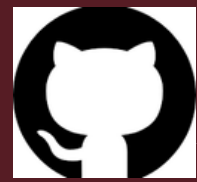


Abhishek-dev-1

total quantity of each pizza category ordered.

```
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity desc
```



| Result Grid | | | Filter Rows: |
|-------------|----------|----------|--------------|
| | category | quantity | |
| ▶ | Classic | 14888 | |
| | Supreme | 11987 | |
| | Veggie | 11649 | |
| | Chicken | 11050 | |

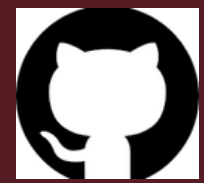


Abhishek-dev-1

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)
ORDER BY order_count DESC;
```

| Result Grid   Filter Rows: <input type="text"/> | | |
|---|------------------|-------------|
| | HOUR(order_time) | order_count |
| ▶ | 12 | 2520 |
| | 13 | 2455 |
| | 18 | 2399 |
| | 17 | 2336 |
| | 19 | 2009 |
| | 16 | 1920 |
| | 20 | 1642 |
| | 14 | 1472 |
| | 15 | 1468 |
| | 11 | 1231 |
| | 21 | 1198 |
| | 22 | 663 |
| | 23 | 28 |
| | 10 | 8 |
| | 9 | 1 |



Abhishek-dev-1

category-wise distribution of pizzas

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

| Result Grid | | | Filter Rows: |
|-------------|----------|-------------|--------------|
| | category | COUNT(name) | |
| ▶ | Chicken | 6 | |
| | Classic | 8 | |
| | Supreme | 9 | |
| | Veggie | 9 | |

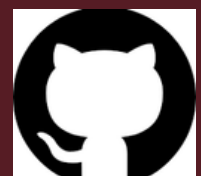


Abhishek-dev-1

Group orders by date and average number of pizzas ordered per day

```
SELECT
    round( AVG(quantity),0) as avg_quantity
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```



| Result Grid | |
|-------------|--------------|
| | avg_quantity |
| ▶ | 138 |

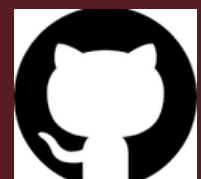


Abhishek-dev-1

top 3 most ordered pizza types based on revenue

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

| Result Grid   Filter Rows: <input type="text"/> | | |
|---|------------------------------|----------|
| | name | revenue |
| ▶ | The Thai Chicken Pizza | 43434.25 |
| | The Barbecue Chicken Pizza | 42768 |
| | The California Chicken Pizza | 41409.5 |



Abhishek-dev-1

percentage contribution of each pizza type to total revenue

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

| Result Grid | | | Filter |
|-------------|----------|---------|--------|
| | category | revenue | |
| ▶ | Classic | 26.91 | |
| | Supreme | 25.46 | |
| | Chicken | 23.96 | |
| | Veggie | 23.68 | |

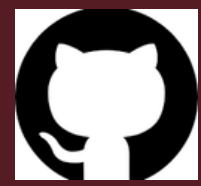


Abhishek-dev-1

cumulative revenue generated over time

```
select order_date, sum(revenue) over(order by order_date) as cum_revenue
from (select orders.order_date,
sum(orders_details.quantity*pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id=pizzas.pizza_id
join orders on orders.order_id=orders_details.order_id
group by orders.order_date) as sales
order by cum_revenue desc limit 5;
```

| Result Grid | | | Filter Rows: |
|-------------|------------|-------------|--------------|
| | order_date | cum_revenue | |
| ▶ | 2015-12-31 | 817860.05 | |
| | 2015-12-30 | 814944.05 | |
| | 2015-12-29 | 813606.25 | |
| | 2015-12-28 | 812253 | |
| | 2015-12-27 | 810615.8 | |



Abhishek-dev-1

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((orders_details.quantity)*pizzas.price) as revenue from
pizza_types join pizzas on pizza_types.pizza_type_id=pizzas.pizza_type_id
join orders_details on orders_details.pizza_id=pizzas.pizza_id group by
pizza_types.category,pizza_types.name) as a) as b where rn<=3
order by revenue limit 5;
```

| Result Grid | | | Filter Rows: |
|-------------|-----------------------|--------------------|--------------|
| | name | revenue | |
| ▶ | The Five Cheese Pizza | 26066.5 | |
| | The Mexicana Pizza | 26780.75 | |
| | The Pepperoni Pizza | 30161.75 | |
| | The Sicilian Pizza | 30940.5 | |
| | The Four Cheese Pizza | 32265.700000000065 | |

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