



# PIZZA SALES REPORT

# INTRODUCTION

Hey everyone, welcome to our pizza sales analysis project .In the fast-paced world of the food and beverage industry, it's super important to understand sales trends and what our customers really want. This project is all about diving into our pizza sales data to uncover insights that can help us make smarter decisions and boost the business performance.



Calculate the total revenue generated from pizza sales.



```
• SELECT
  ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS total_revenue
FROM
  order_details
  JOIN
  pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid	
	total_revenue
▶	168036.1



-- Identify the highest-priced pizza.

```
• SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types,
    pizzas
WHERE
    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



Identify the most common pizza size ordered.

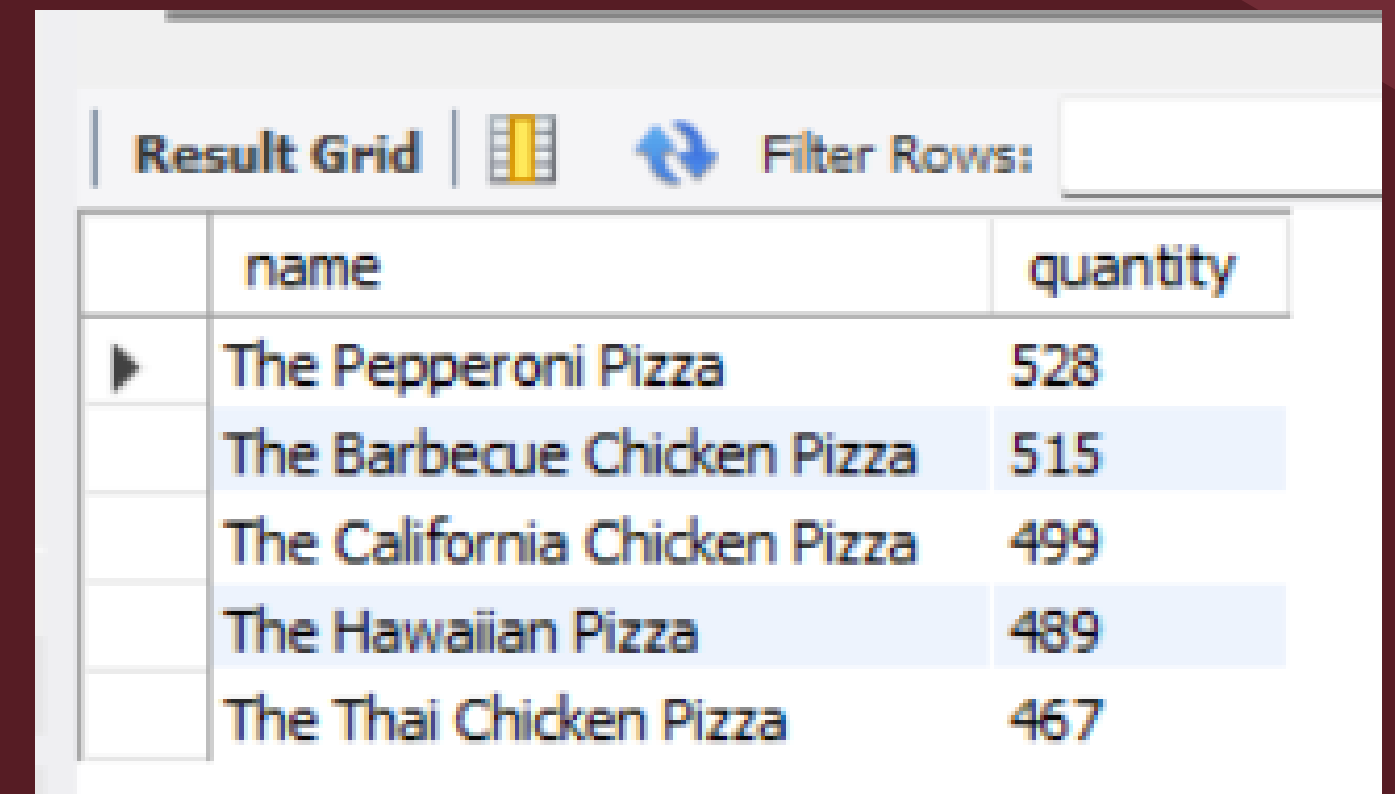
- ```
select pizzas.size, count(order_details.pizza_id) from pizzas
join order_details on
pizzas.pizza_id=order_details.pizza_id
group by pizzas.size;
```

| Result Grid |      |                               | Filter Rows: |
|-------------|------|-------------------------------|--------------|
|             | size | count(order_details.pizza_id) |              |
| ▶           | M    | 3080                          |              |
|             | L    | 3862                          |              |
|             | S    | 2954                          |              |
|             | XL   | 105                           |              |
|             | XXL  | 6                             |              |



-- 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;
```





The screenshot shows a 'Result Grid' window with a toolbar containing a grid icon, a refresh icon, and a 'Filter Rows' input field. The table below displays the results of the SQL query, listing the top 5 most ordered pizza types by quantity in descending order.

|   | name                         | quantity |
|---|------------------------------|----------|
| ▶ | The Pepperoni Pizza          | 528      |
|   | The Barbecue Chicken Pizza   | 515      |
|   | The California Chicken Pizza | 499      |
|   | The Hawaiian Pizza           | 489      |
|   | The Thai Chicken Pizza       | 467      |



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

```
select hour(order_time),count(order_id) from orders group by hour(order_time);
```

| Result Grid     Filter Rows: <input type="text"/> |                  |                 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|-----------------|
|                                                                                                                                                                                                                         | hour(order_time) | 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               |



-- 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;
```

Result Grid			Filter Rows
	category	quantity	
▶	Classic	3025	
	Veggie	2465	
	Supreme	2454	
	Chicken	2254	





-- Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT
    ROUND(AVG(quantity), 2)
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 order_date) AS order_quantity;
```

Result Grid		Filter Rows:
	ROUND(AVG(quantity), 2)	
▶	137.81	



-- 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;
```

Result Grid			Filter Rows:
	name	revenue	
▶	The Barbecue Chicken Pizza	9098.25	
	The California Chicken Pizza	8694.25	
	The Thai Chicken Pizza	8486.25	



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

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

Result Grid			Filter Rows
	category	total_sales	
▶	Classic	26.5	
	Veggie	24.45	
	Supreme	25.28	
	Chicken	23.78	



-- Analyze the cumulative revenue generated over time.

```
• SELECT order_date,
      round(sum(
revenue) over(ORDER BY order_date ),
      2) AS cum_rev from
      (SELECT orders.order_date,
      sum(order_details.quantity*pizzas.price) AS revenue
      FROM orders
      JOIN order_details on
orders.order_id=order_details.order_id
      JOIN pizzas
      ON
pizzas.pizza_id=order_details.pizza_id
      GROUP BY orders.order_date) AS sales;
```

Result Grid		Filter Rows:
	order_date	cum_rev
▶	2015-01-01	2713.85
	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	23990.35
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.3
	2015-01-14	32358.7
	2015-01-15	34343.5
	2015-01-16	36937.65
	2015-01-17	39001.75
	2015-01-18	40978.6

Result Grid		Filter Rows:
	order_date	cum_rev
	2015-01-01	2713.85
	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	23990.35
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.3
	2015-01-14	32358.7
	2015-01-15	34343.5
	2015-01-16	36937.65
	2015-01-17	39001.75
	2015-01-18	40978.6



-- 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 limit 3;
```

Result Grid			Filter Rows:
	name	revenue	
▶	The Barbecue Chicken Pizza	9098.25	
	The California Chicken Pizza	8694.25	
	The Classic Deluxe Pizza	7057.5	