

Delicious Pizza for Everyone!

# PIZZA BOXCAR

SQL



The background is a solid light orange color. In the corners, there are decorative illustrations: top-left shows a slice of pizza with toppings; top-center shows a whole pizza; top-right shows a cluster of orange dots; bottom-left shows a green leafy plant; bottom-right shows a slice of pizza with toppings. The word "Hello!" is centered in a large, bold, dark green font.

# Hello!

"Hello, I'm Pranav Golande, and I've been immersing myself in the world of SQL. Today, I'm excited to share how I've applied my SQL skills to analyze a dataset on pizza sales. By querying this data, I've sought answers to key questions, unlocking insights that can inform decision-making in the pizza industry. Let's delve into how SQL has empowered me to extract valuable information from this dataset."



# Chef Pranav

## Questions

- 1) Retrieve the total number of orders placed.
- 2) Calculate the total revenue generated from pizza sales.
- 3) Identify the highest-priced pizza.
- 4) Identify the most common pizza size ordered.
- 5) Calculate the percentage contribution of each pizza type to total revenue.
- 6) Analyze the cumulative revenue generated over time.







# Retrieve the total number of orders placed.



```
-- Retrieve the total number of orders placed.  
  
select count(order_id) from orders;
```

| Result Grid |                 |  |  | F |
|-------------|-----------------|---|---|---|
|             | count(order_id) |   |   |   |
| ▶           | 21350           |   |   |   |



# Calculate the total revenue generated from pizza sales.



```
select  
round(sum(orders_detail.quantity * pizzas.price),2) as total_sales  
from orders_detail join pizzas  
on pizzas.pizza_id = orders_detail.pizza_id
```

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

# 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;
```

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





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



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









# 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;
```



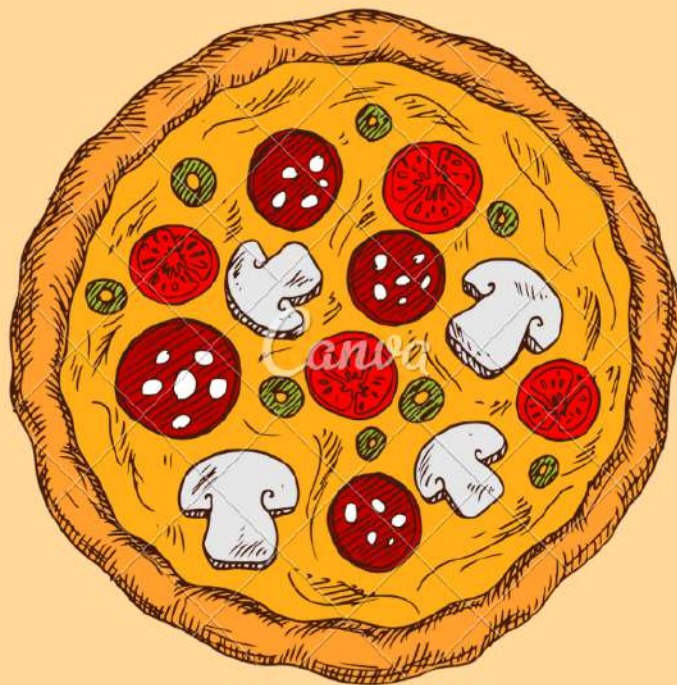
# Call us for Order



Phone Number:  
+7875883085

Mail:  
pranavgolande007@gmail.com

Github:  
<https://github.com/bytemaster69>



PRANAV GOLANDE

**THANK  
YOU**

