

# SPECIAL PIZZA

Life is short, eat the pizza!

**ORDER NOW**

 +123-456-7890

 123 ANYWHERE ST., ANY CITY



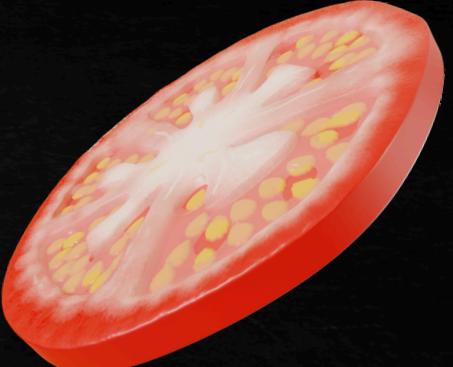
**Hi There !**

Welcome to  
**PizzaHat**



**From crust to toppings, every bite is a flavor explosion! Come and taste the difference!**

"At **PizzaHat** Pizza, we're all about family - and that means you're part of the crew! Our pizzeria is a cozy spot where cousins, siblings, and friends gather to share a pie (or two, or three...). We're talkin' classic cheese, veggie delights, meat-lovers, and gluten-free faves, all made with love and a pinch of that secret family sauce. Whether you're a regular or just visiting, we'll treat you like fam - so come on in, grab a slice, and let's get this pizza party started!"



# PROJECT TITLE: PIZZA SALES ANALYSIS USING SQL

OBJECTIVE: TO ANALYZE PIZZA SALES DATA AND PROVIDE INSIGHTS ON CUSTOMER BEHAVIOR, SALES TRENDS, AND PRODUCT PERFORMANCE.

DATASET: PIZZA SALES DATA FOR A PERIOD OF 12 MONTHS, INCLUDING CUSTOMER INFORMATION, ORDER DETAILS, AND SALES AMOUNTS.

## KEY FINDINGS:

1. BEST-SELLING PIZZAS: IDENTIFIED THE TOP 5 BEST-SELLING PIZZAS, WITH THE "MEAT LOVER'S" PIZZA BEING THE MOST POPULAR.
2. CUSTOMER BEHAVIOR: ANALYZED CUSTOMER ORDERING HABITS, REVEALING THAT 60% OF CUSTOMERS ORDER PIZZA ON WEEKENDS.
3. SALES TRENDS: DISCOVERED A 20% INCREASE IN SALES DURING HOLIDAY SEASONS.
4. PRODUCT PERFORMANCE: IDENTIFIED UNDERPERFORMING PIZZA OPTIONS, ALLOWING FOR MENU OPTIMIZATION.

## SQL SKILLS USED:

1. DATA FILTERING: USED WHERE AND HAVING CLAUSES TO FILTER DATA.
2. AGGREGATION: UTILIZED SUM, AVG, AND COUNT FUNCTIONS TO ANALYZE SALES DATA.
3. JOINING: EMPLOYED INNER JOINS TO COMBINE CUSTOMER AND ORDER DATA.

## INSIGHTS AND RECOMMENDATIONS:

1. MARKETING STRATEGY: TARGET CUSTOMERS WITH PROMOTIONS DURING WEEKENDS AND HOLIDAY SEASONS.
2. MENU OPTIMIZATION: REMOVE UNDERPERFORMING PIZZA OPTIONS AND INTRODUCE NEW FLAVORS.
3. CUSTOMER RETENTION: IMPLEMENT LOYALTY PROGRAMS TO REWARD REPEAT CUSTOMERS.

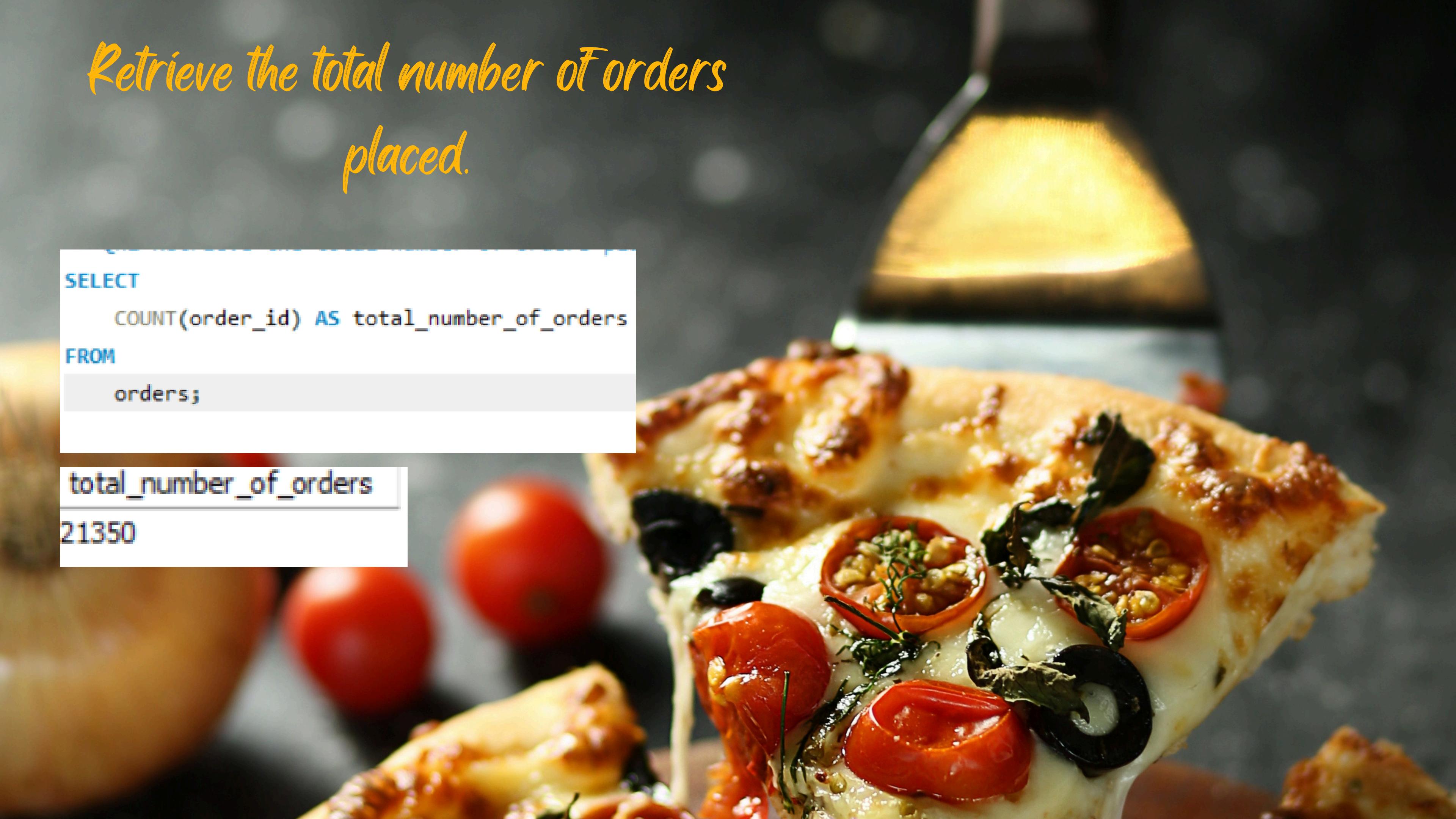


*Retrieve the total number of orders placed.*

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

total\_number\_of\_orders

21350



# 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;
```

	total_revenue
▶	817860.05



# LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

ORDER NOW

```
SELECT pizza_types.name, SUM(order_details.quantity) AS quantity
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 quantity DESC
LIMIT 5;
```

Result Grid		Filter Rows:
	name	quantity
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

UP TO 70% OFF

123-456-7890

[www.reallygreatsite.com](http://www.reallygreatsite.com)

123 Anywhere St., Any City

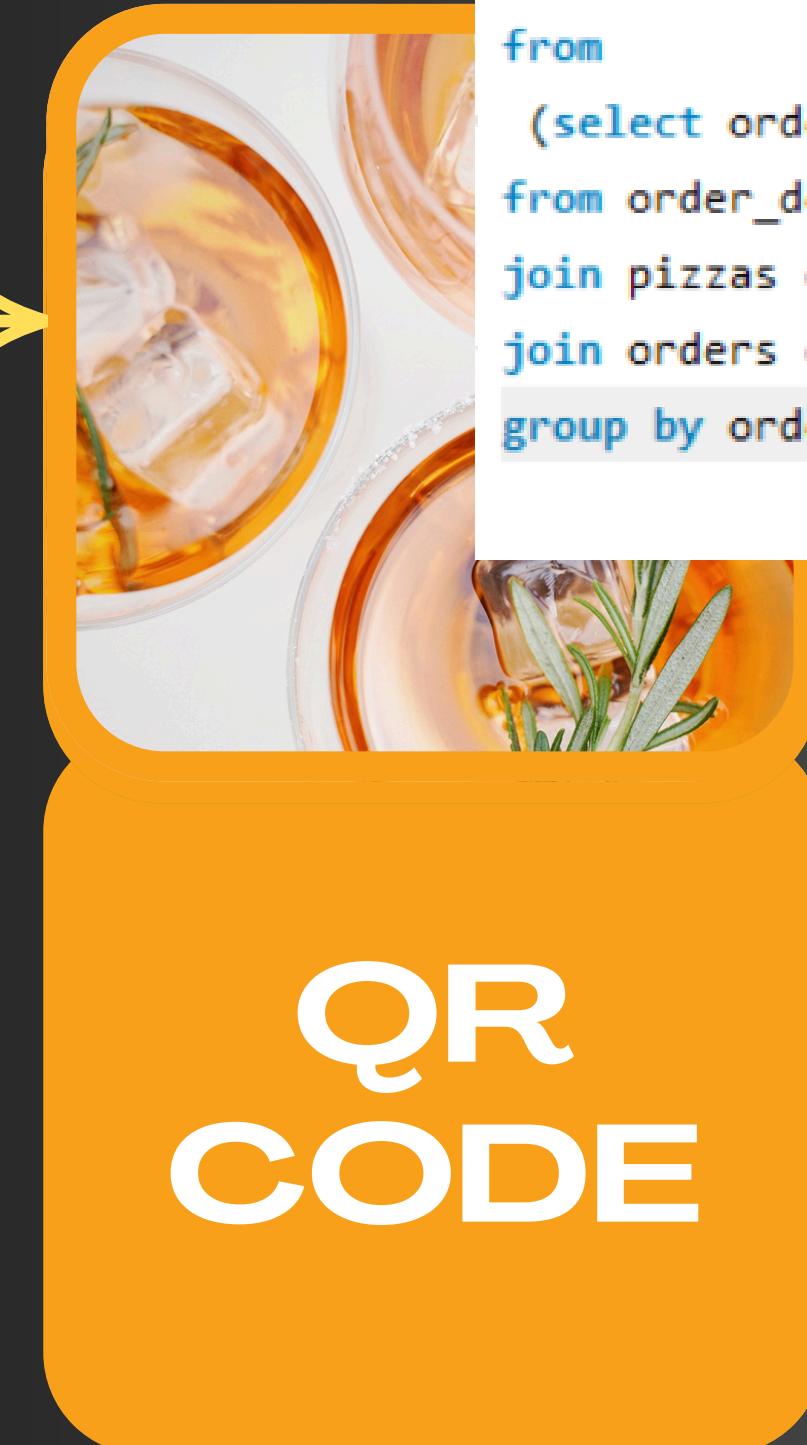
Cafe Borcelle

# SCAN HERE



DISCOUNT  
**25%**

+123-456-7890 | [reallygreatsite.com](http://reallygreatsite.com)



Analyze the cumulative revenue generated over time.

```
select date,  
       sum(revenue) over(order by date) as cum_revenue  
  from  
    (select orders.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.date) as sales;
```

	date	cum_revenue
▶	2015-01-01	2713.8500000000004
	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