



# KRITIKA KOUNDAL

R E S T A U R A N T

*Specialist in Italian Food*





# ABOUT US

We are a restaurant that focuses on delivering delicious foods, good quality, hygiene, and healthy menu combinations.  
— "Kritika Koundal".

Retrieve the total number of  
order placed

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

	total_orders
▶	11343

# Calculate the total number revenue generator from pizza size

```
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
```

total_sales
76273.8

# 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  
ORDER BY order_count DESC;
```

	size	order_count
▶	L	1777
	M	1395
	S	1324
	XL	46
	XXL	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  
LIMIT 5;
```

	category	quantity
▶	Classic	1387
	Veggie	1081
	Supreme	1080
	Chicken	1076

# Determine the distribution of orders by hour of the day

```
SHOW COLUMNS FROM orders;  
  
SELECT * FROM sales_pizza_project.orders;  
  
SELECT  
    HOUR(CONCAT(date, ' ', time)) AS hour,  
    COUNT(order_id) AS order_count  
FROM orders  
GROUP BY  
    HOUR(CONCAT(date, ' ', time));
```

	hour	order_count
	11	629
	12	1356
	13	1263
	14	833
	15	782
	16	1006
	17	1302
	18	1254
	19	1055
	20	879
	21	630
	22	334
	23	14
	10	6

# list of top 5 most ordered pizza type 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;
```

name	quantity
The California Chicken Pizza	249
The Hawaiian Pizza	245
The Barbecue Chicken Pizza	242
The Pepperoni Pizza	239
The Thai Chicken Pizza	231

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

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

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

# Group the order by date and calculate the average number of pizzas ordered per day

```
SELECT
    round(avg(total_quantity),0) AS avg_pizza_ordered_per_day
FROM(SELECT orders.date, SUM(order_details.quantity) AS total_quantity
      FROM orders
      JOIN order_details ON orders.order_id = order_details.order_id
     GROUP BY orders.date
     LIMIT 0 , 1000) AS order_quantity;
```

Result Grid | Filter Rows:

	avg_pizza_ordered_per_day
	70

# Determine the top3 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
1	The California Chicken Pizza	4382.75
2	The Barbecue Chicken Pizza	4237.5
3	The Thai Chicken Pizza	4217.25

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

```
SELECT pizza_types.category,
       (SUM(order_details.quantity * pizzas.price) / total_sales.total_revenue) * 100 AS percentage_of_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
JOIN ( SELECT SUM(order_details.quantity * pizzas.price) AS total_revenue FROM order_details
       JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id ) AS total_sales
GROUP BY pizza_types.category, total_sales.total_revenue
ORDER BY percentage_of_total_sales DESC;
```

category	percentage_of_total_sales
Classic	26.739784827817736
Chicken	25.08200718988702
Supreme	24.51923202987139
Veggie	23.658975952424132

# Analyze the cumulative revenue generated over time

```
SELECT date,  
       SUM(revenue) OVER (ORDER BY date) AS cum_revenue  
FROM (  
    SELECT o.date,  
           SUM(od.quantity * p.price) AS revenue  
    FROM order_details od  
    JOIN pizzas p ON od.pizza_id = p.pizza_id  
    JOIN orders o ON o.order_id = od.order_id  
    GROUP BY o.date  
) AS sales;
```

date	cum_revenue
2015-01-01	1249.6000000000001
2015-01-02	2712.05
2015-01-03	3942.600000000004
2015-01-04	4740.700000000001
2015-01-05	5783.800000000001
2015-01-06	7021.350000000001
2015-01-07	7795.500000000001
2015-01-08	8921.150000000001
2015-01-09	9959.900000000001
2015-01-10	11266.100000000002
2015-01-11	12375.500000000002
2015-01-12	13318.000000000002
2015-01-13	14102.700000000003
2015-01-14	15366.850000000002
2015-01-15	16325.300000000003
2015-01-16	17614.750000000004
2015-01-17	18807.000000000004
2015-01-18	19911.400000000005
2015-01-19	21232.650000000005
2015-01-20	22217.150000000005

# 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 ranking
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 sales_data) as b
where ranking <= 3 ;
```

name	revenue
The California Chicken Pizza	4382.75
The Barbecue Chicken Pizza	4237.5
The Thai Chicken Pizza	4217.25
The Hawaiian Pizza	3268
The Classic Deluxe Pizza	3193.5
The Pepperoni Pizza	2982
The Italian Supreme Pizza	3055.5
The Spicy Italian Pizza	2968.5
The Sicilian Pizza	2723
The Four Cheese Pizza	2810.0999999999967
The Five Cheese Pizza	2553
The Mexicana Pizza	2412.25



KRITIKA KOUNDAL  
R E S T A U R A N T

# CONTACT US

Linkedin: - [www.linkedin.com/in/kritika-koundal-b69552312](https://www.linkedin.com/in/kritika-koundal-b69552312)

GitHub ID :- <https://github.com/kritikakoundal?tab=repositories>





KRITIKA KOUNDAL  
R E S T A U R A N T

THANK YOU

