

# SPECIAL MENU

CHEESE AND SPICY PIZZA



click on given link

SHOP NOW

[WWW.REALLYGREATSITE.COM](http://WWW.REALLYGREATSITE.COM)



# Hello!

We are a pizza gift serving make very delicious pizza via the food luxurious way. we are ready to go around to deliver and serve pizza for you lovers!

# PIZZA MENU



**The Barbecue Chicken Pizza**

**The California Chicken Pizza**

**The Chicken Alfredo Pizza**

**The Chicken Pesto Pizza**

**The Southwest Chicken Pizza**

**The Thai Chicken Pizza**

**The Big Meat Pizza**

**The Classic Deluxe Pizza**

**The Hawaiian Pizza**

**The Napolitana Pizza**

**The Pepperoni, Mushroom, and Peppers Pizza**

**The Greek Pizza**

**The Brie Carre Pizza**

**The Calabrese Pizza**

**The Sicilian Pizza**

**The Soppressata Pizza**

**The Spicy Italian Pizza**

**The Spinach Supreme Pizza**

**The Five Cheese Pizza**

**The Four Cheese Pizza**

**The Green Garden Pizza**



# Chef Nail Train



hi, i am the main chef and owner of adri pizza gift serving . pizza gift is family- owned secret recipe pizza that has been passed down for generations with incredible delicacy!

# Retrieve the total number of orders placed.

```
-- Retrieve the total number of orders placed.
```

```
SELECT count(order_id) as total_orders from orders;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	count(order_id)			
▶	21350			

# Calculate the total revenue generated from pizza sales.

```
-- Calculate the total revenue generated from pizza sales.
```

```
• 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
```

Result Grid



Filter Rows:

	total_sale
▶	54672.65

# Identify the highest-priced pizza.

```
1 -- Identify the highest-priced pizza.  
2  
3 • SELECT  
4     pizza_types.name, pizzas.price  
5 FROM  
6     pizza_types  
7         JOIN  
8     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
9 ORDER BY pizzas.price DESC  
10 LIMIT 1;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

# Identify the most common pizza size ordered.

```
-- 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;
```

Result Grid | Filter Rows:

	size	order_count
▶	L	1246
	M	1013
	S	961
	XL	40

# List the top 5 most ordered pizza types along with their quantities.

```
1  -- List the top 5 most ordered pizza types
2  -- along with their quantities.
3
4 • SELECT
5      pizza_types.name, SUM(order_details.quantity) AS quantity
6  FROM
7      pizza_types
8      JOIN
9      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10     JOIN
11     order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY quantity DESC
14 LIMIT 5;
```

	name	quantity
▶	The Pepperoni Pizza	194
	The Thai Chicken Pizza	162
	The Barbecue Chicken Pizza	160
	The California Chicken Pizza	160
	The Classic Deluxe Pizza	152

# Join the necessary tables to find the total quantity of each pizza category ordered.

```
-- 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;
```

	<b>category</b>	<b>quantity</b>
▶	Classic	984
	Supreme	818
	Veggie	800
	Chicken	723

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

```
1 -- Determine the distribution of orders by hour of the day.  
2  
3 • SELECT  
4     HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
5 FROM  
6     orders  
7 GROUP BY HOUR(order_time);
```

	hour(order_time)	count(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468

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

```
1 -- Join relevant tables to find the
2 -- category-wise distribution of pizzas.
3
4 • SELECT category, count(name) FROM pizza_types
5 group by category
```

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```
1 -- Group the orders by date and calculate the average
2 -- number of pizzas ordered per day.
3
4 • SELECT
5     ROUND(AVG(quantity), 0)
6 FROM
7     (SELECT
8         orders.order_date, SUM(order_details.quantity) AS quantity
9     FROM
10    orders
11   JOIN order_details ON orders.order_id = order_details.order_id
12   GROUP BY orders.order_date) AS order_quantity;
```

	round(avg(quantity), 0)
▶	139

# Determine the top 3 most ordered pizza types based on revenue.

```
1  -- Determine the top 3 most ordered pizza types based on revenue.  
2  
3 • SELECT  
4      pizza_types.name,  
5      SUM(order_details.quantity * pizzas.price) AS revenue  
6  FROM  
7      pizza_types  
8      JOIN  
9      pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
10     JOIN  
11     order_details ON order_details.pizza_id = pizzas.pizza_id  
12   GROUP BY pizza_types.name  
13 ORDER BY revenue DESC  
14 LIMIT 3;
```

name	revenue
The Thai Chicken Pizza	2969.5
The Barbecue Chicken Pizza	2804
The California Chicken Pizza	2740

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

```
1  -- Calculate the percentage contribution of each
2  -- pizza type to total revenue.
3
4 • SELECT
5   pizza_types.category,
6   ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
7     ROUND(SUM(order_details.quantity * pizzas.price),
8       2) AS total_sales
9
10    FROM
11      order_details
12      JOIN
13        pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
14      2) AS revenue
15
16  FROM
17    pizza_types
18    JOIN
19      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
20    JOIN
21      order_details ON order_details.pizza_id = pizzas.pizza_id
22
23  GROUP BY pizza_types.category
24
25  ORDER BY revenue DESC;
```

category	revenue
Classic	26.49
Supreme	25.67
Veggie	24.54
Chicken	23.29

# Analyze the cumulative revenue generated over time.

```
1  -- Analyze the cumulative revenue generated over time.  
2  
3 • SELECT order_date,  
4   sum(revenue) over(order by order_date) as cum_revenue  
5   FROM  
6   (SELECT orders.order_date,  
7    SUM(order_details.quantity * pizzas.price) as revenue  
8    FROM order_details join pizzas  
9    on order_details.pizza_id = pizzas.pizza_id  
10   join orders  
11   on orders.order_id = order_details.order_id  
12   group by orders.order_date) as sales;
```

order_date	cum_revenue
2015-01-01	2713.850000000004
2015-01-02	45.75
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

# Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
1 -- Determine the top 3 most ordered pizza types
2 -- based on revenue for each pizza category.
3
4 • SELECT name, revenue FROM
5   (SELECT category, name, revenue,
6    rank() over(partition by category order by revenue desc) as rn
7   FROM
8   (SELECT pizza_types.category, pizza_types.name,
9    SUM((order_details.quantity) * pizzas.price) as revenue
10   FROM pizza_types join pizzas
11   on pizza_types.pizza_type_id = pizzas.pizza_type_id
12   join order_details
13   on order_details.pizza_id = pizzas.pizza_id
14   group by pizza_types.category, pizza_types.name) as a) as b
15 WHERE rn <= 3;
```

name	revenue
The Thai Chicken Pizza	2969.5
The Barbecue Chicken Pizza	2804
The California Chicken Pizza	2740
The Pepperoni Pizza	2438.75
The Classic Deluxe Pizza	2367

# **Thank You!**