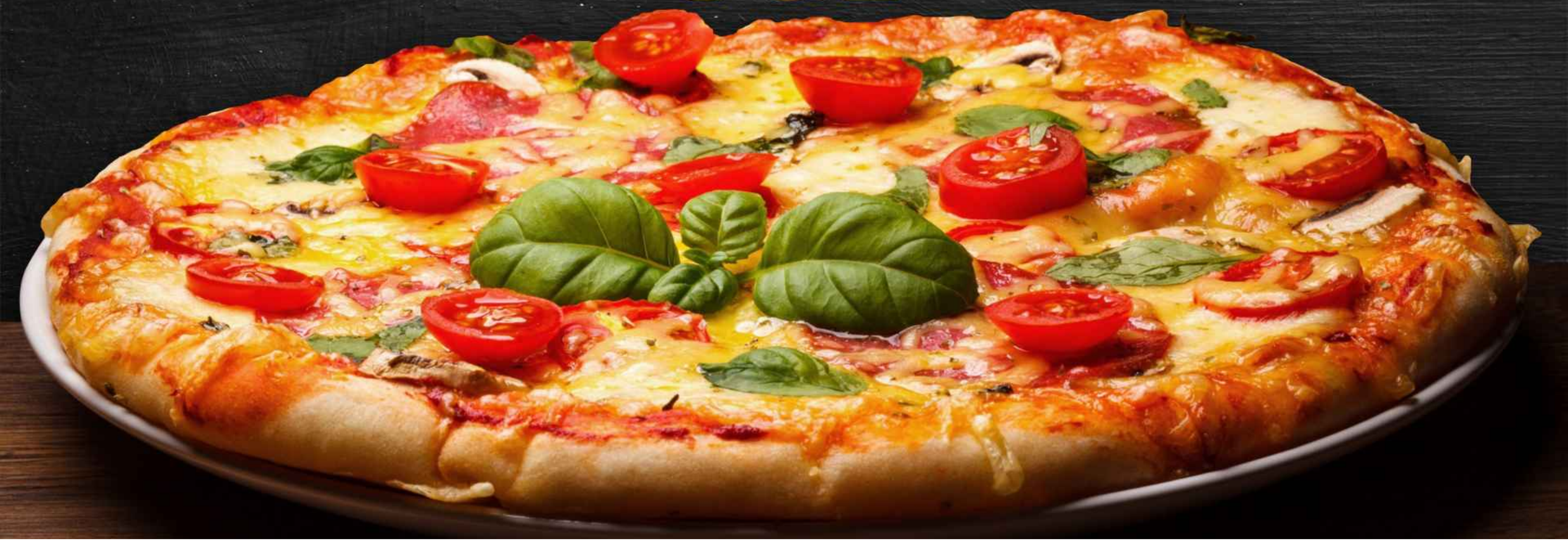
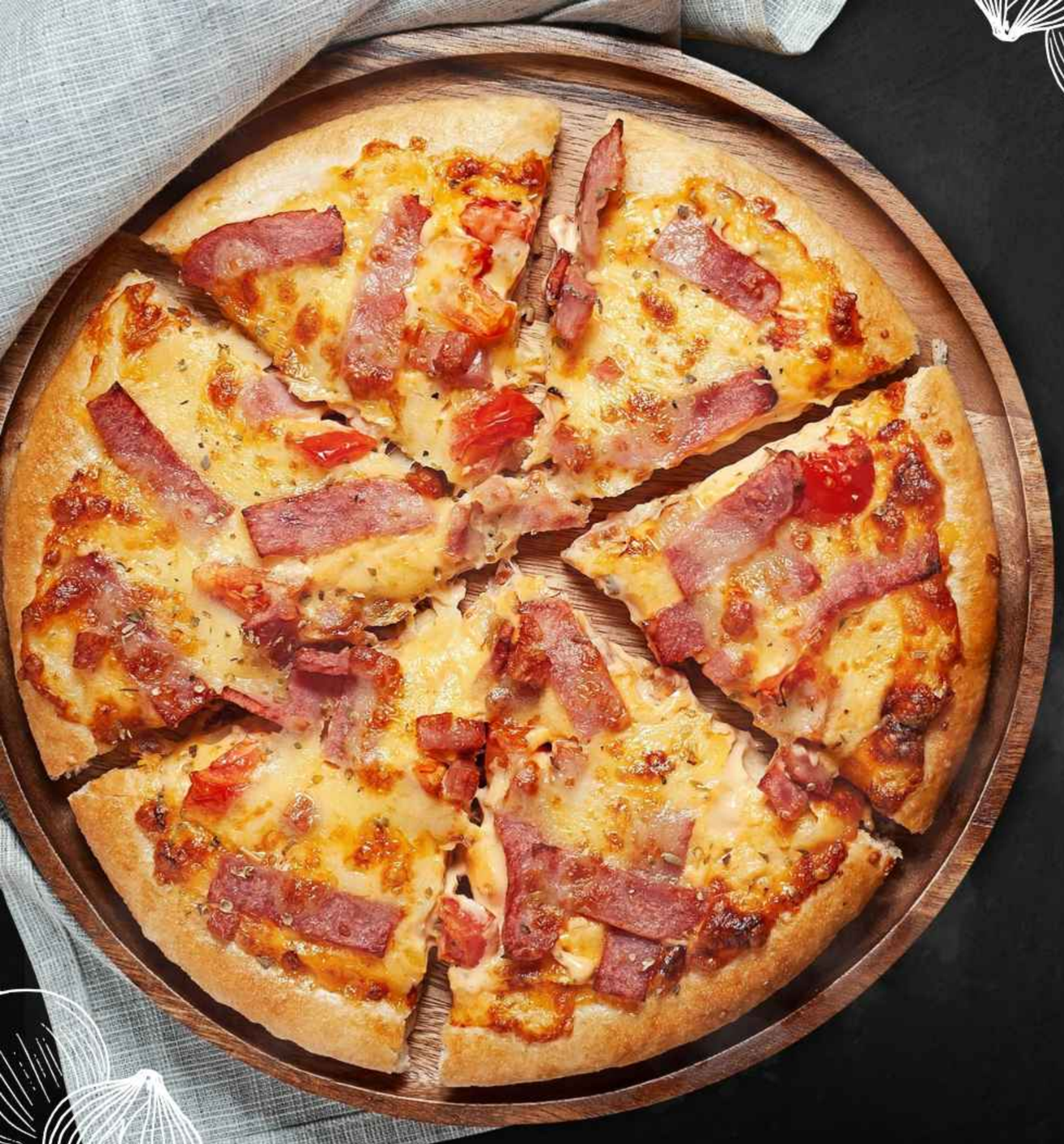
PIZZA SALES ANALYSIS USING SQL



HELLO EVERYONE

My name is Bala Murugan , and i completed a pizza sales analysis using sql . I utilized SQL queries to explore data and gain insights that could enhance sales and operations. This project showcases my SQL skills and my ability wo work with data to support business decisions.



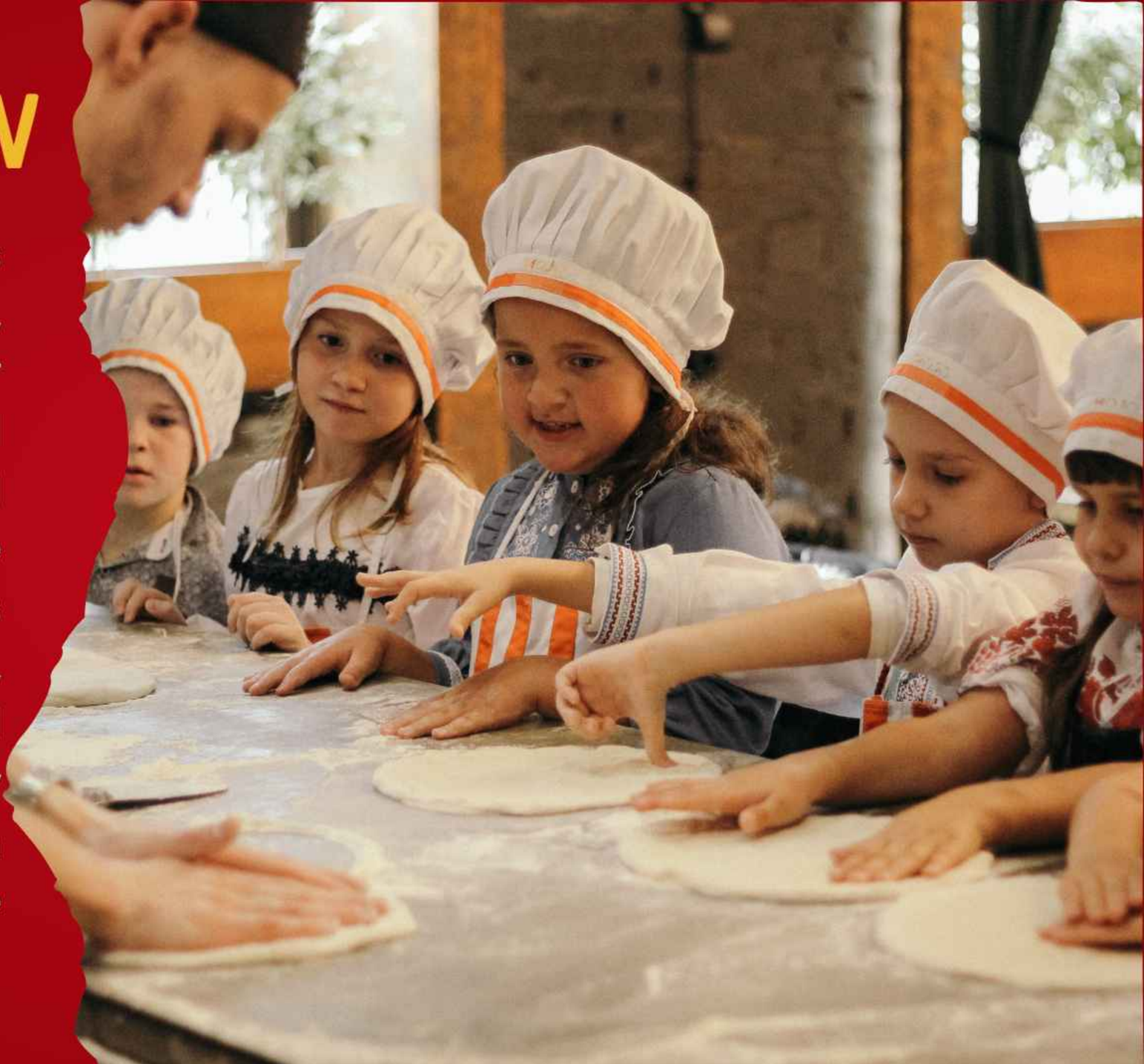
INTRODUCTION

- This presentation focuses on analyzing pizza sales data using SQL to uncover key business insights.
-

- Key Objectives:
  - Identify top-selling pizzas.
  - Understand customer preferences and patterns.
  - Analyze sales trends over time.

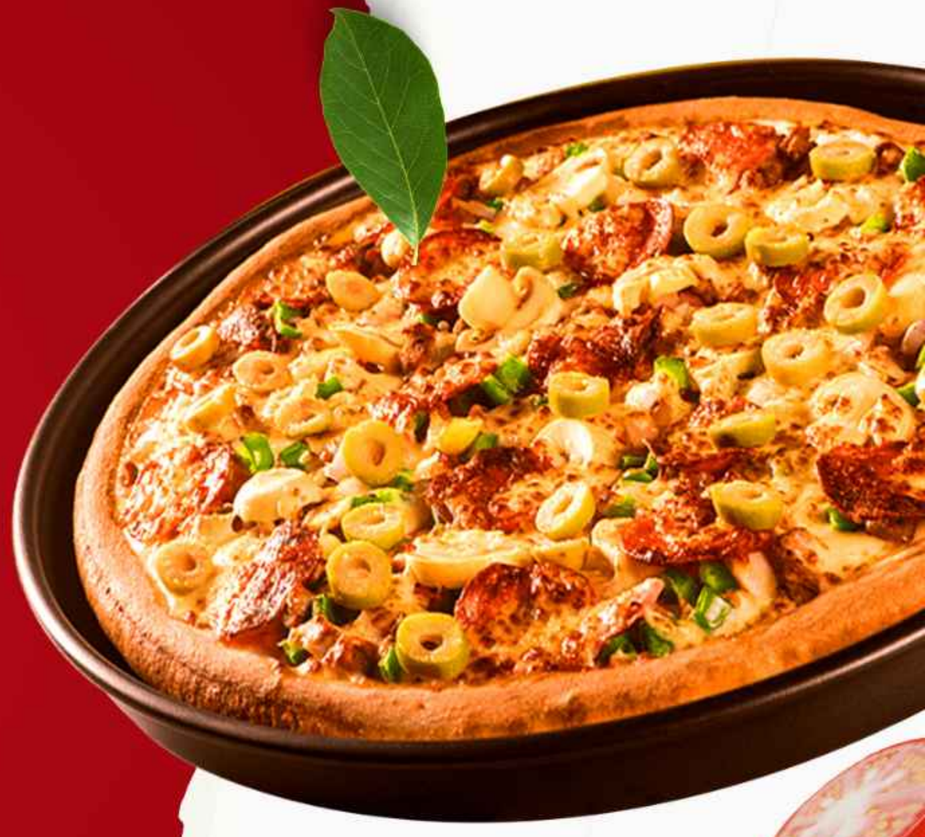
DATA OVERVIEW

This analysis identifies factors infulencing purchasing behaviour, such as seasonal trends , popular pizza types, peak sales time and hour. it also examines regional taste differences to refine marketing strategies. The findings will guide menu enhancements, new product introductions, and inventory optimization, ultimately boosting customer satisfaction and sales growth in the competitive food industry.



RETRIVE THE TOTAL NUMBER OF ORDERS PLACED

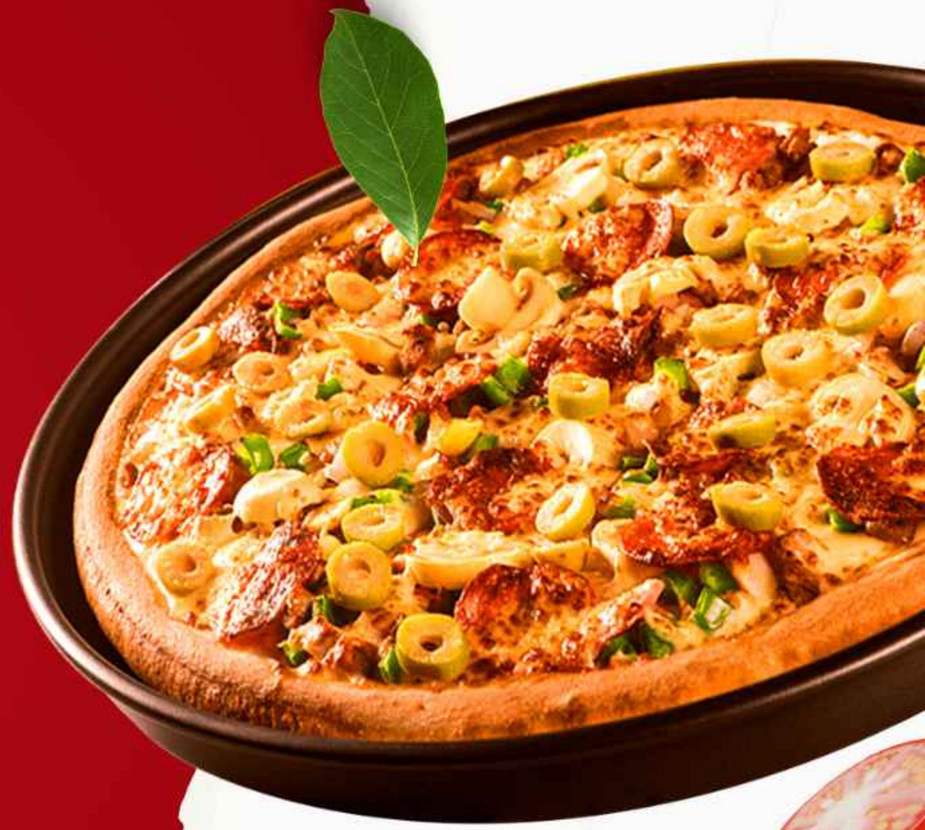
```sql
SELECT
    COUNT(order_id) AS Total_orders
FROM
    Orders;
```



	Total_orders
▶	4042

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

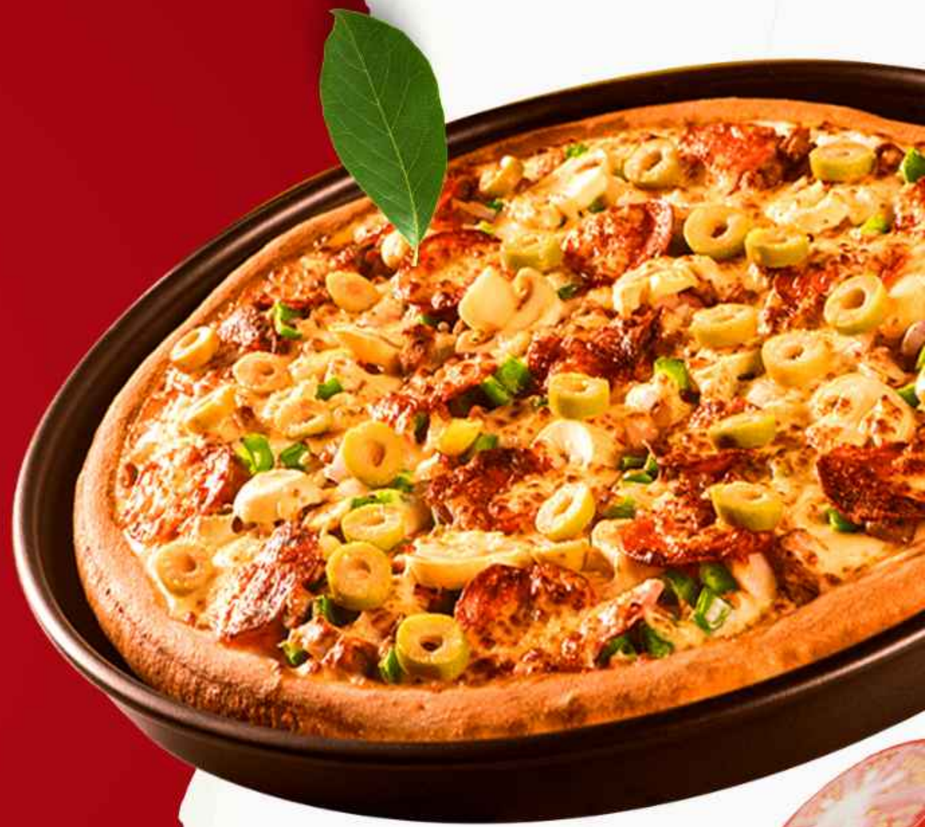
```sql
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS Total_sales
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```



	Total_orders
▶	4042

IDENTIFY THE HIGHEST PRICED PIZZA

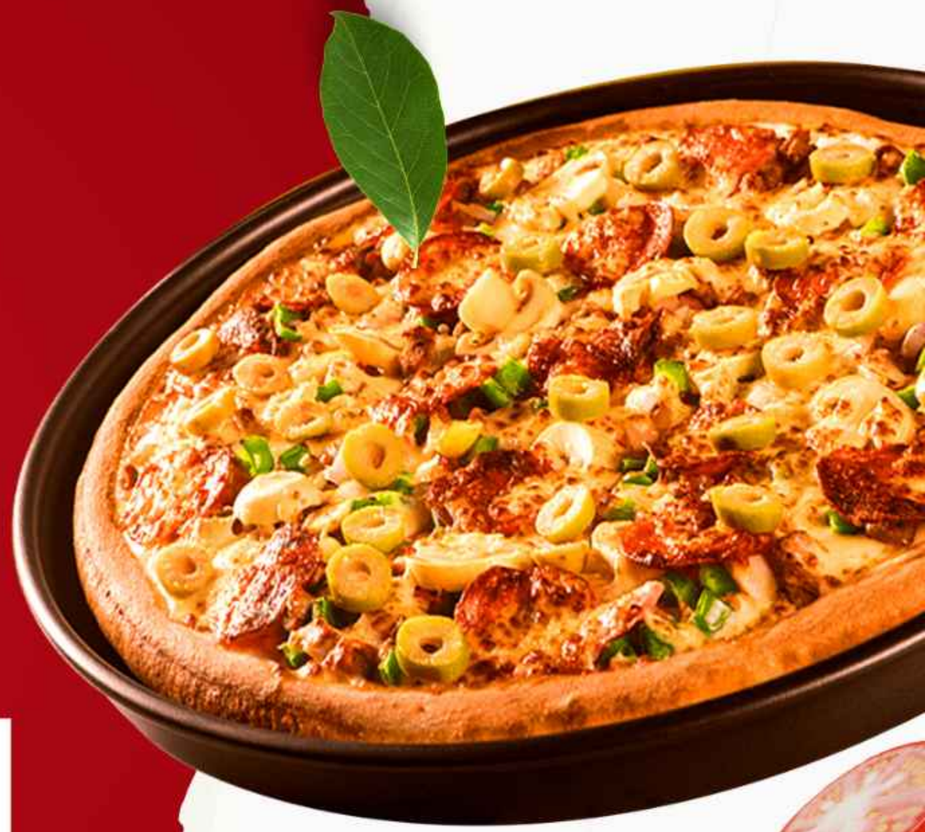
```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```



	name	price
▶	The Greek Pizza	35.95

IDENTIFY MOST COMMONLY ORDERED PIZZA SIZE

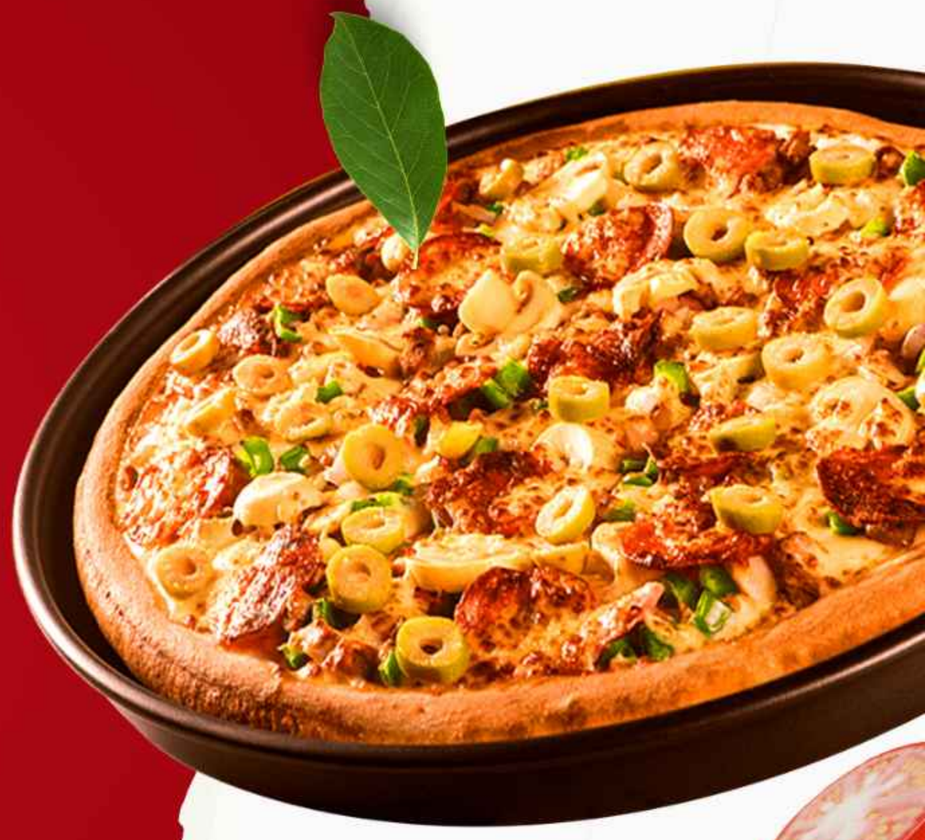
```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS Total_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY Total_count DESC
LIMIT 1;
```



	size	Total_count
▶	L	18526

LIST THE TOP 5 MOST ORDERED PIZZAS ALONG WITH THEIR QUANTITIES

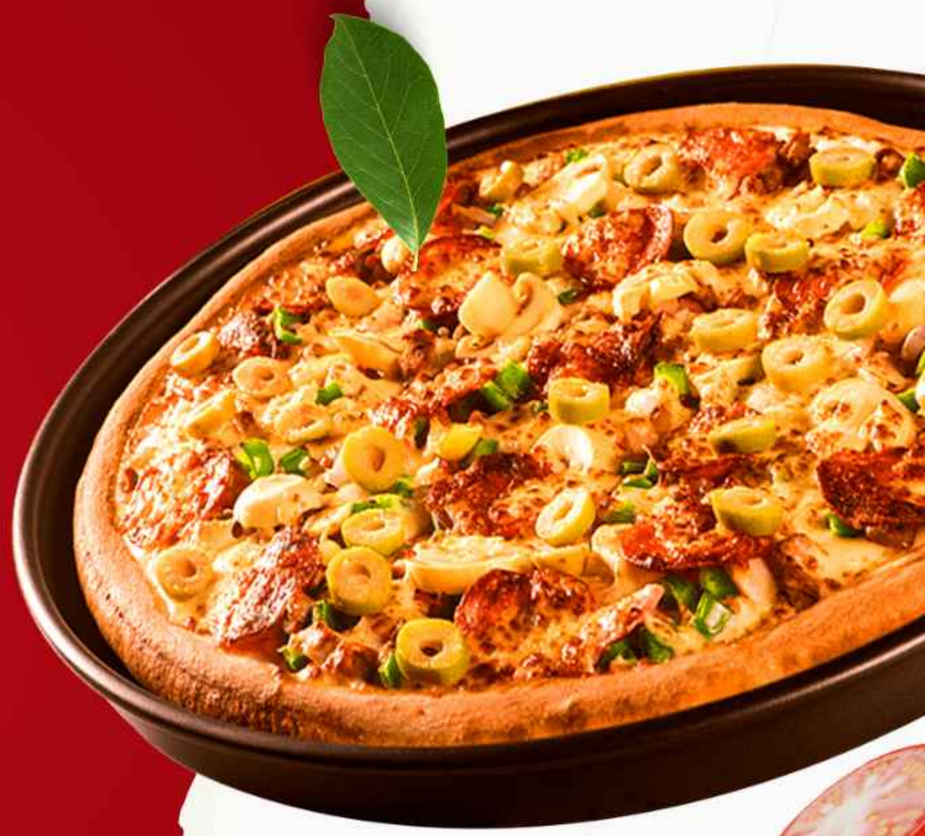
```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS Total_Quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY Total_Quantity DESC
LIMIT 5;
```



	name	Total_Quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

JOIN THE NECCESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA ORDERED CATEGORY

```sql
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;
```

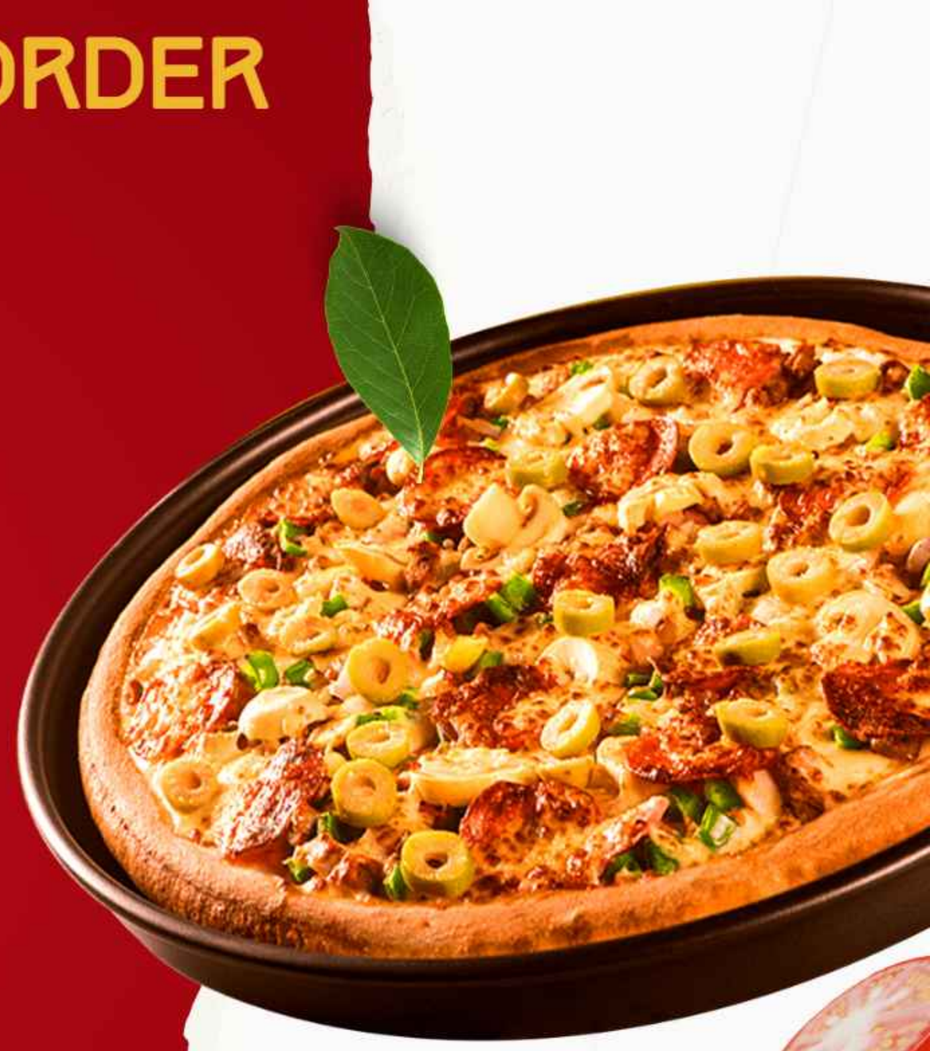


	category	Quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDER BY HOUR OF THE DAY

```sql
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

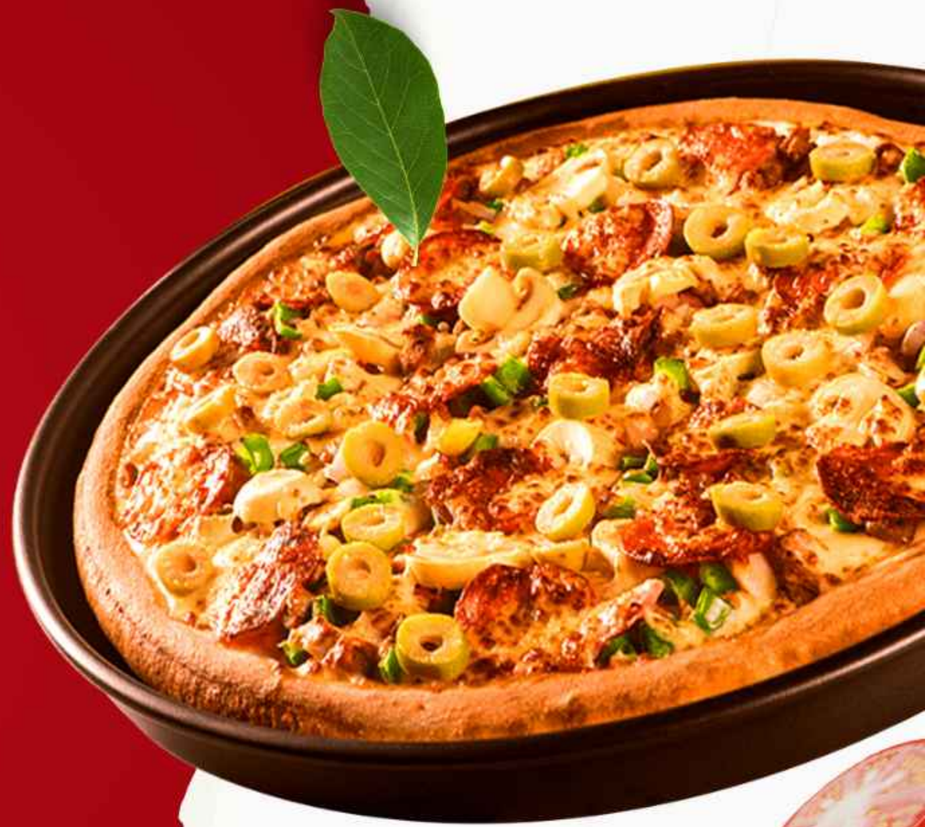
	hour	order_count
▶	11	224
	12	478
	13	448
	14	327
	15	284
	16	344
	17	472
	18	454
	19	358
	20	310
	21	224



JOIN RELEVANT TABLES TO FIND THE CATEGORY WISE DISTRIBUTION OF PIZZA

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

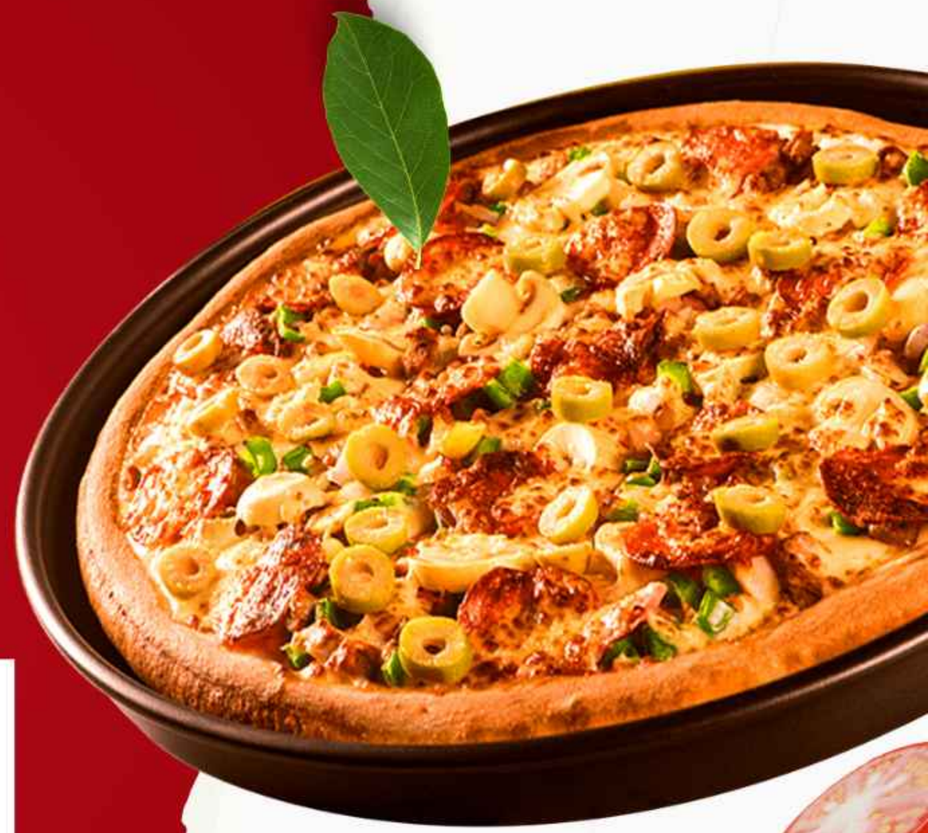
	category	COUNT(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBERS OF PIZZAS ORDERED BY DAY

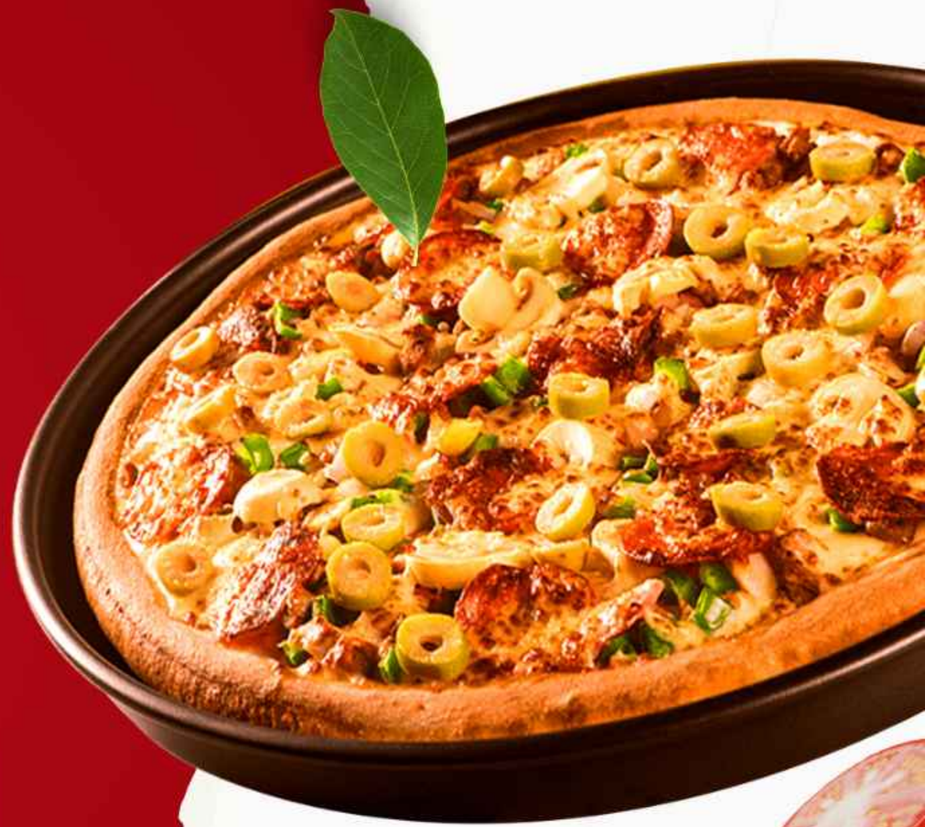
```sql
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

	avg_pizza_ordered_per_day
▶	138



DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```sql
SELECT
    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.name
ORDER BY revenue DESC
LIMIT 3;
```

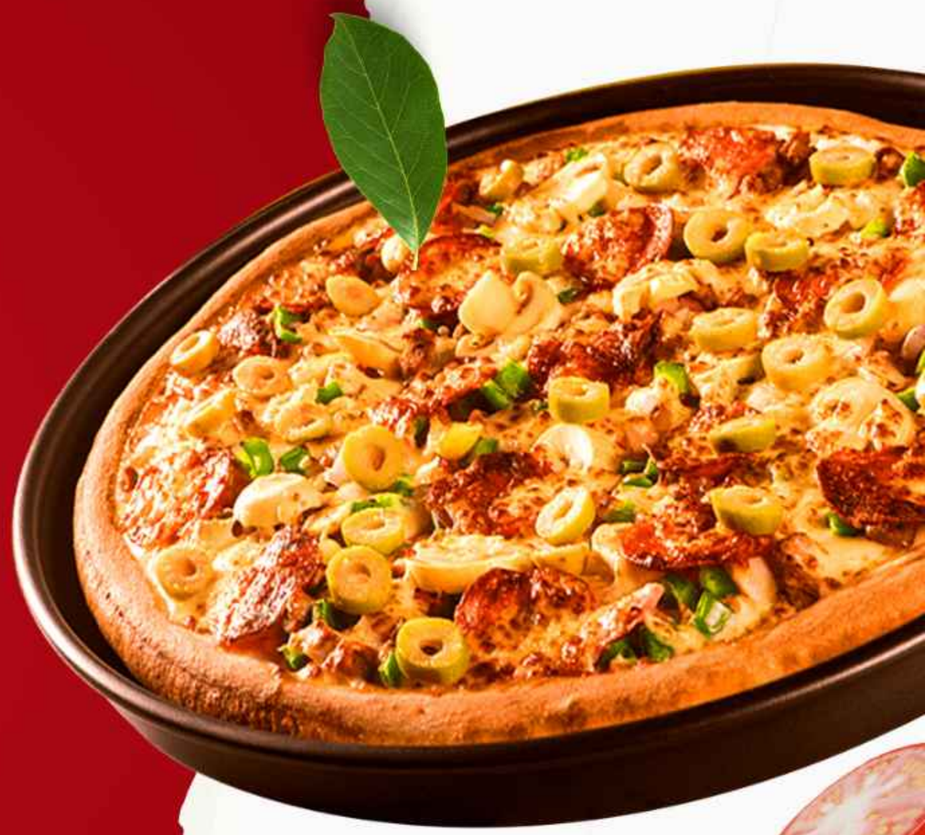


	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

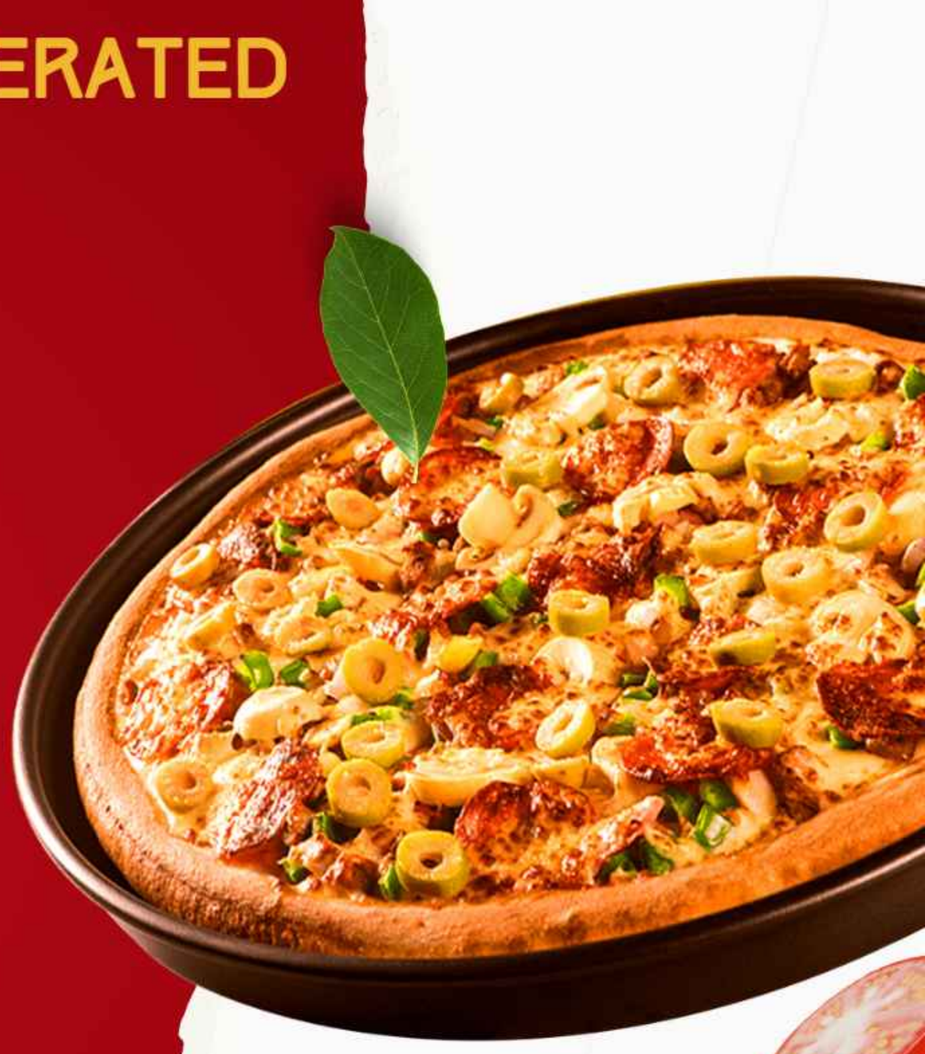
```sql
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

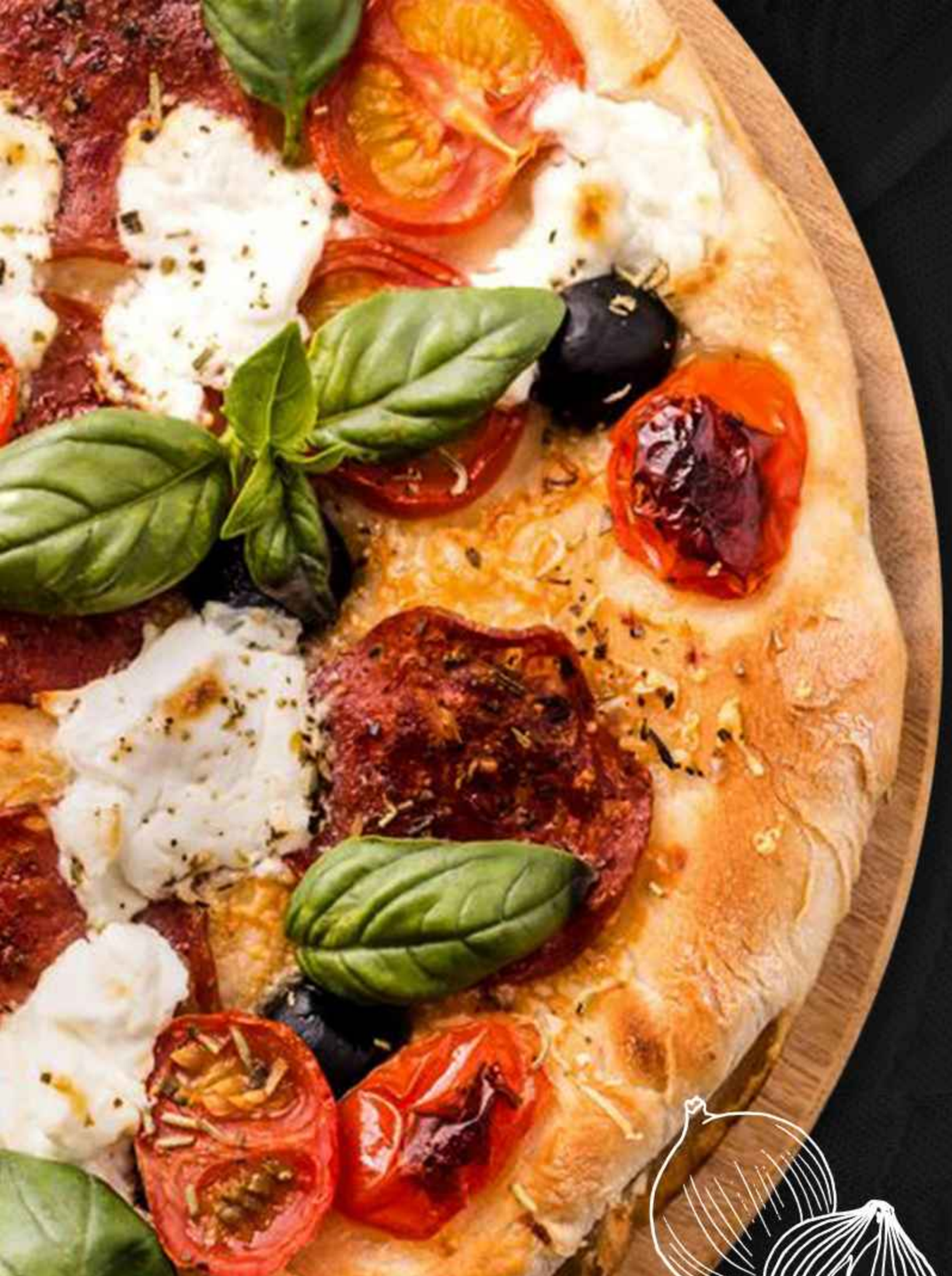


ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```sql
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 daily_revenue
```



order_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



I HOPE YOU FIND THIS
VALUABLE
FOLLOW ME FOR MORE

@Bala Murugan

Like and
comment

SAVE for
later

THANK YOU!