

DATA ANALYSIS OF Dominos

Using SQL for Insights

- Anshu Rashi

The background of the slide is a dark, textured surface. In the top-left corner, there are fresh ingredients: a whole red tomato, a yellow bell pepper, and a head of garlic. In the bottom-left corner, a pizza is shown, partially cut into slices, with toppings like olives, red peppers, and onions. In the bottom-right corner, there are more vegetables, including a yellow bell pepper and some mushrooms. The title 'INTRODUCTION' is centered in a white banner with a double-headed arrow shape.

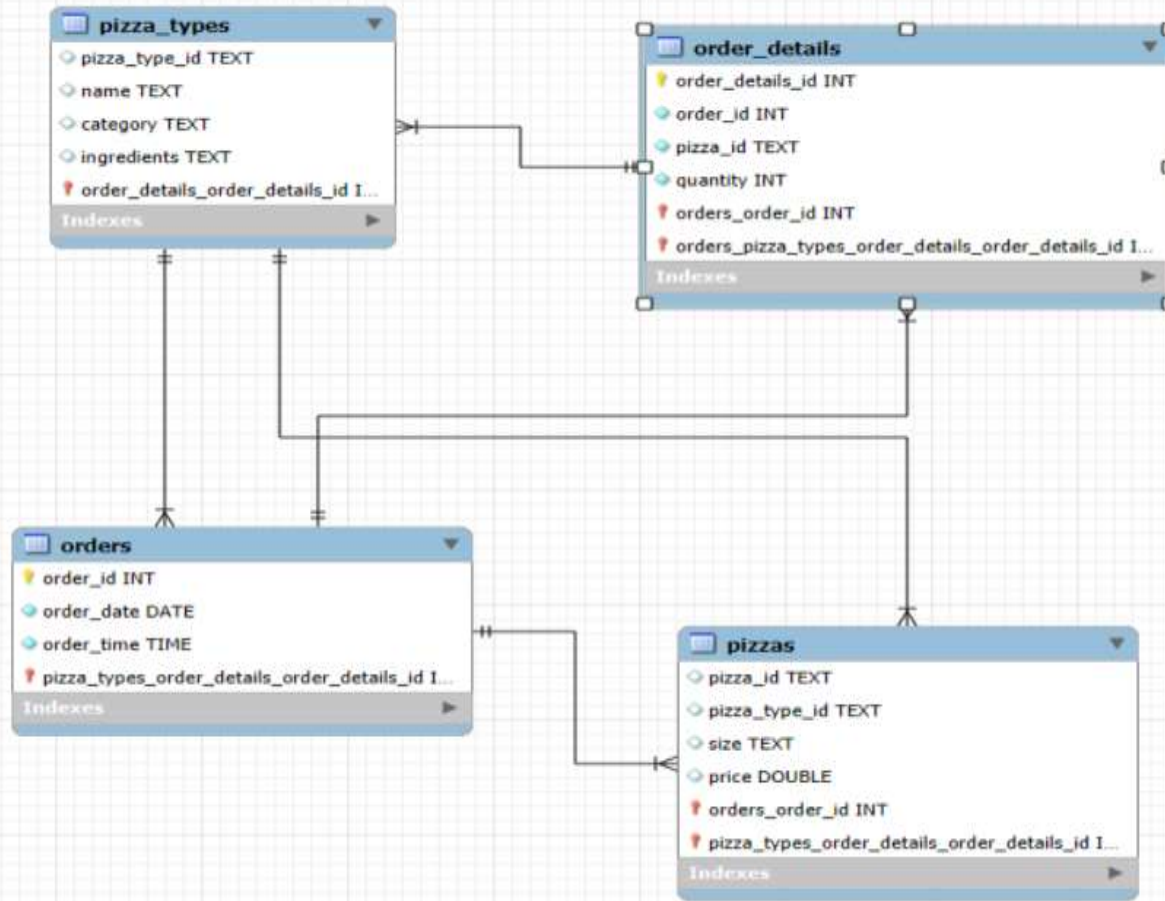
INTRODUCTION

In this project, SQL is utilized to perform a comprehensive data analysis for a pizza company. The primary objective of this analysis is to derive actionable insights from the company's data to improve decision-making processes, enhance customer satisfaction, and optimize business operations.

By leveraging SQL, I have been able to query and manipulate large datasets efficiently, allowing for a detailed examination of various aspects of the business, such as sales performance, customer behavior, and product inventory.

This presentation will walk you through the data preparation process, key analysis metrics, SQL queries used, and the insights gained. Additionally, it will provide recommendations for the pizza company based on the findings, highlighting how data-driven decisions can contribute to the company's growth and success.

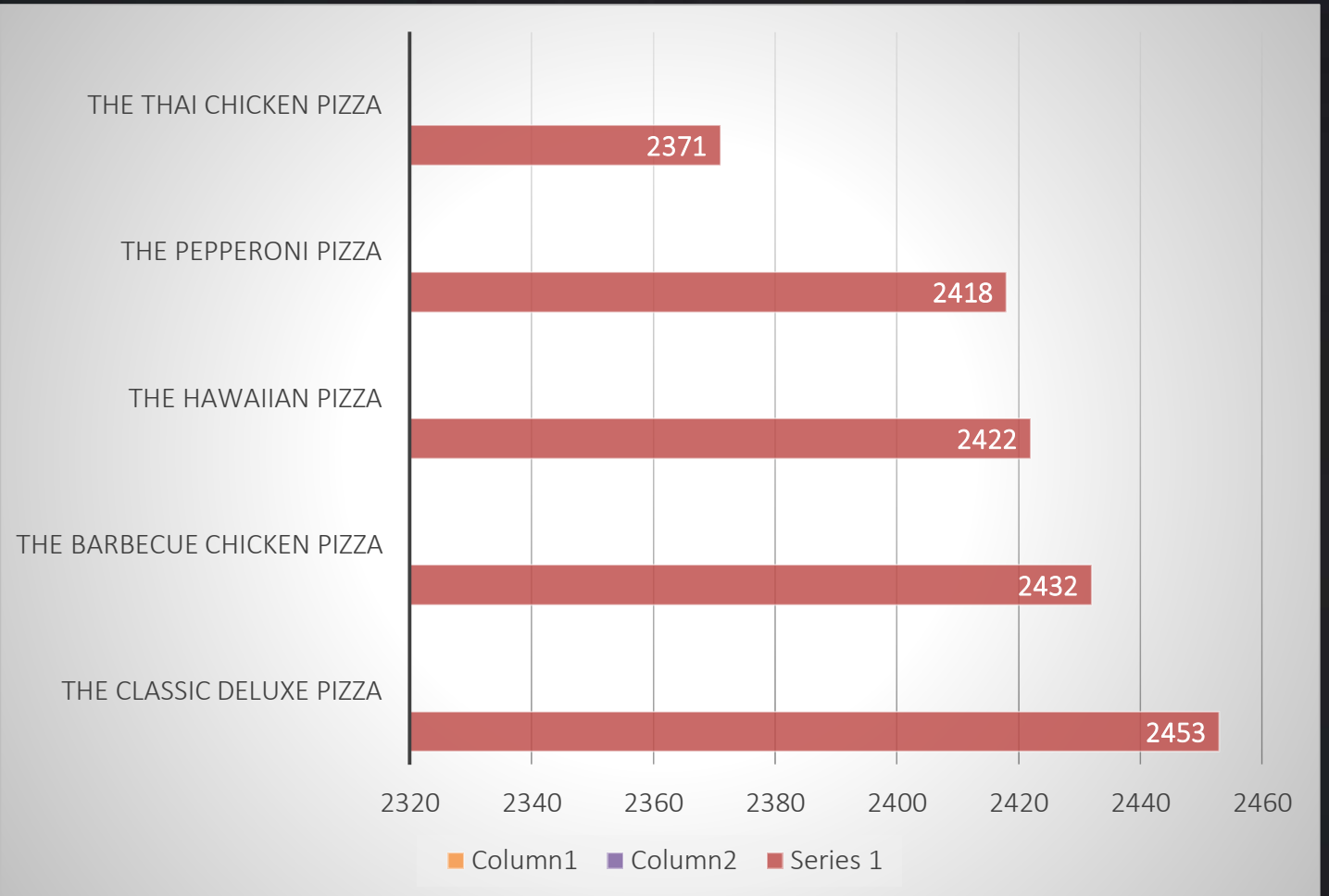
TABLES USED



List the top 5 most ordered pizza types 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;
```

Result Grid			Filter Rows:
	name	quantity	
▶	The Classic Deluxe Pizza	2453	
	The Barbecue Chicken Pizza	2432	
	The Hawaiian Pizza	2422	
	The Pepperoni Pizza	2418	
	The Thai Chicken Pizza	2371	

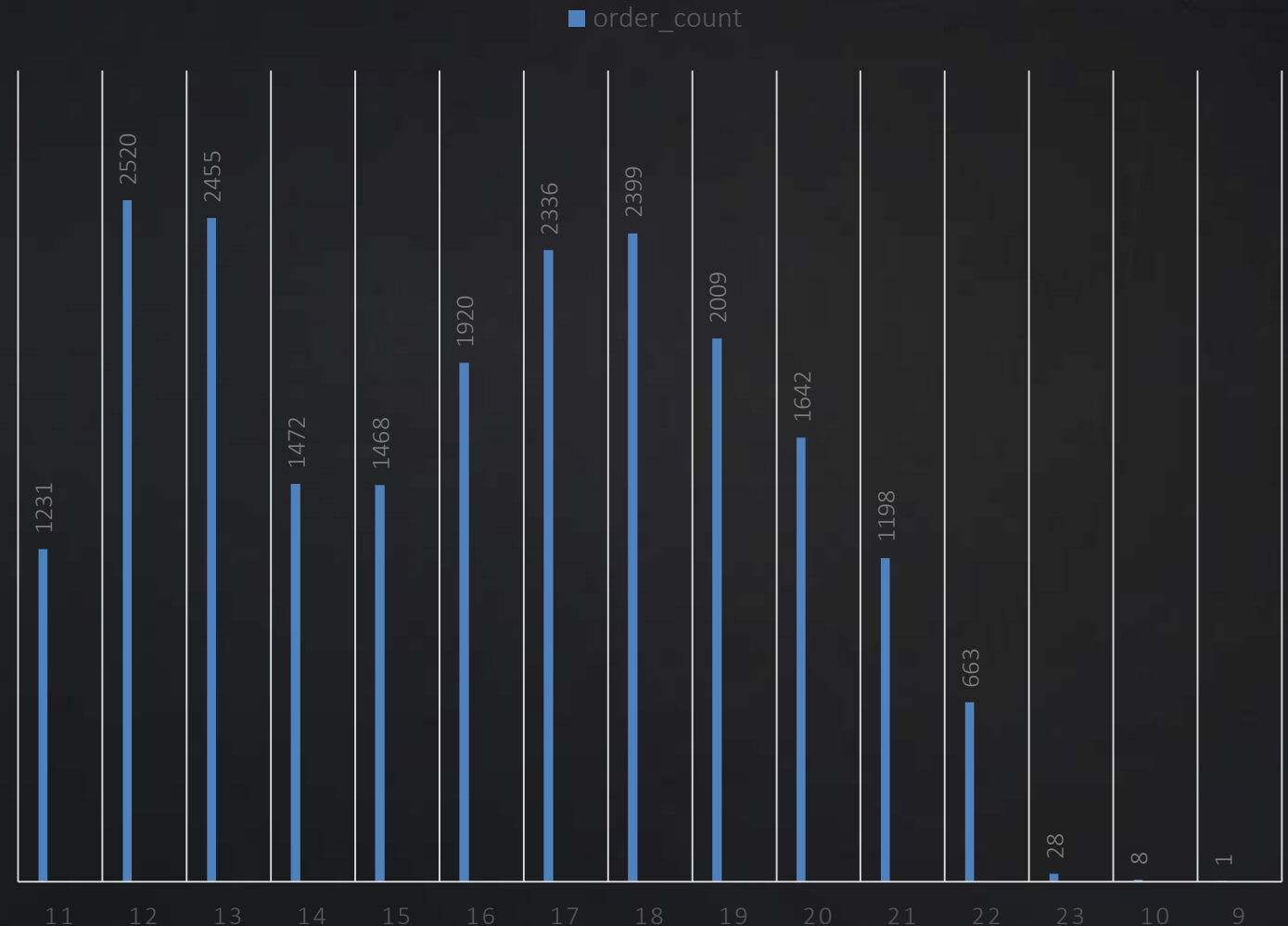


Determine the distribution of orders by hour of the day.

ORDERS VS HOUR

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

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8
	9	1



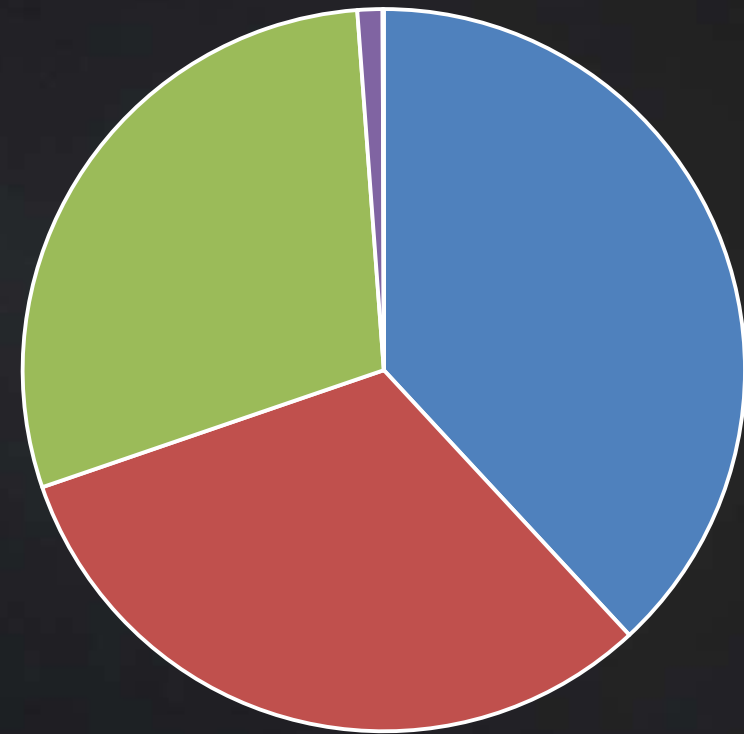
Identify the most common pizza size ordered.



Sales

```
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	18526	
	M	15385	
	S	14137	
	XL	544	
	XXL	28	



■ L ■ M ■ S ■ XL ■ XXL

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



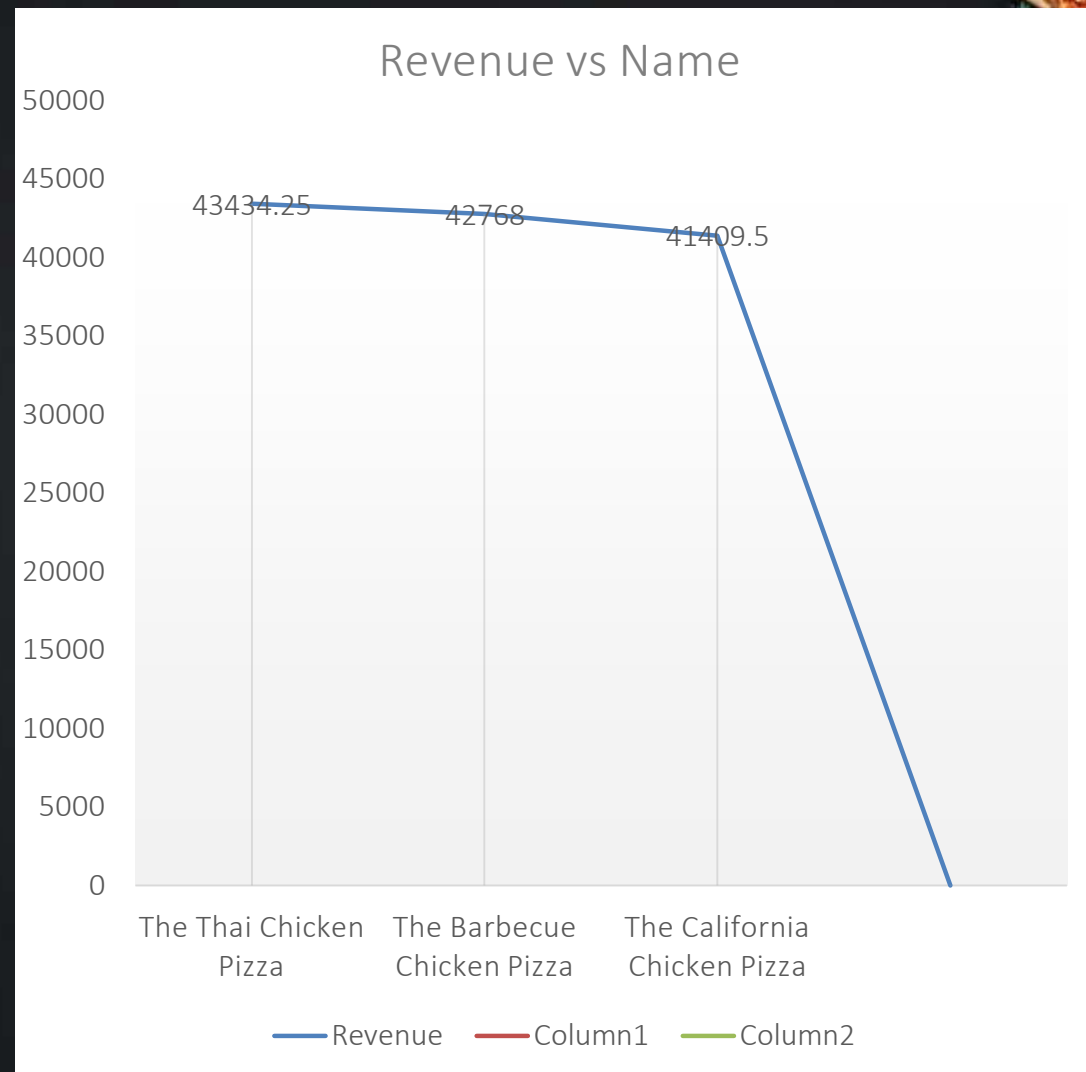
```
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid		Filter Rows:
	avg_pizza_ordered_per_day	
▶	138	

Determine the top 3 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;
```

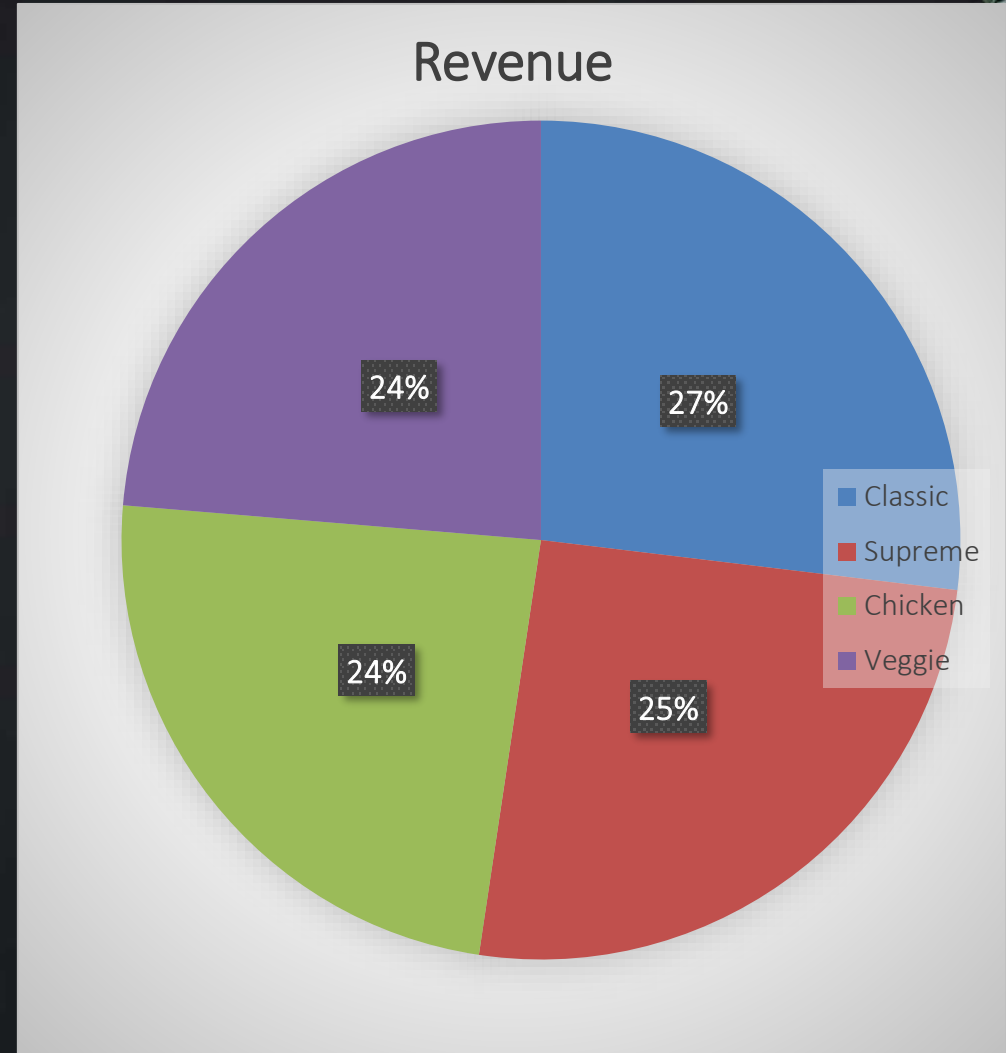
Result Grid			Filter Rows:
	name	revenue	
▶	The Thai Chicken Pizza	43434.25	
	The Barbecue Chicken Pizza	42768	
	The California Chicken Pizza	41409.5	



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

```
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (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)) * 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;
```

Result Grid	Filter
category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



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 rn
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 a) as b
where rn<=3;
```

Result Grid			Filter Rows:	Exp
	name	revenue		
▶	The Thai Chicken Pizza	43434.25		
	The Barbecue Chicken Pizza	42768		
	The California Chicken Pizza	41409.5		
	The Classic Deluxe Pizza	38180.5		
	The Hawaiian Pizza	32273.25		
	The Pepperoni Pizza	30161.75		
	The Spicy Italian Pizza	34831.25		
	The Italian Supreme Pizza	33476.75		
	The Sicilian Pizza	30940.5		
	The Four Cheese Pizza	32265.70000000065		
	The Mexicana Pizza	26780.75		
	The Five Cheese Pizza	26066.5		



THANK
YOU