

# Pizza Sales SQL Report





# Project aim

This project uses SQL queries to analyze pizza sales data. The analysis covers various aspects such as pizza types, order values ,order quantities and order dates and times.





# Questions:

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.
- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.
- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.





Retrieve the total number of orders placed.

```
-- Retrieve the total number of orders placed.  
● SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    orders;
```

	total_orders
▶	21350





# Calculate the total revenue generated from pizza sales.

```
-- Calculate the total revenue generated from pizza sales.  
  
SELECT  
    ROUND(SUM(quantity * price), 2) AS total_sales  
FROM  
    pizzas p  
    JOIN  
    order_details o ON p.pizza_id = o.pizza_id;
```

	total_sales
▶	817860.05





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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

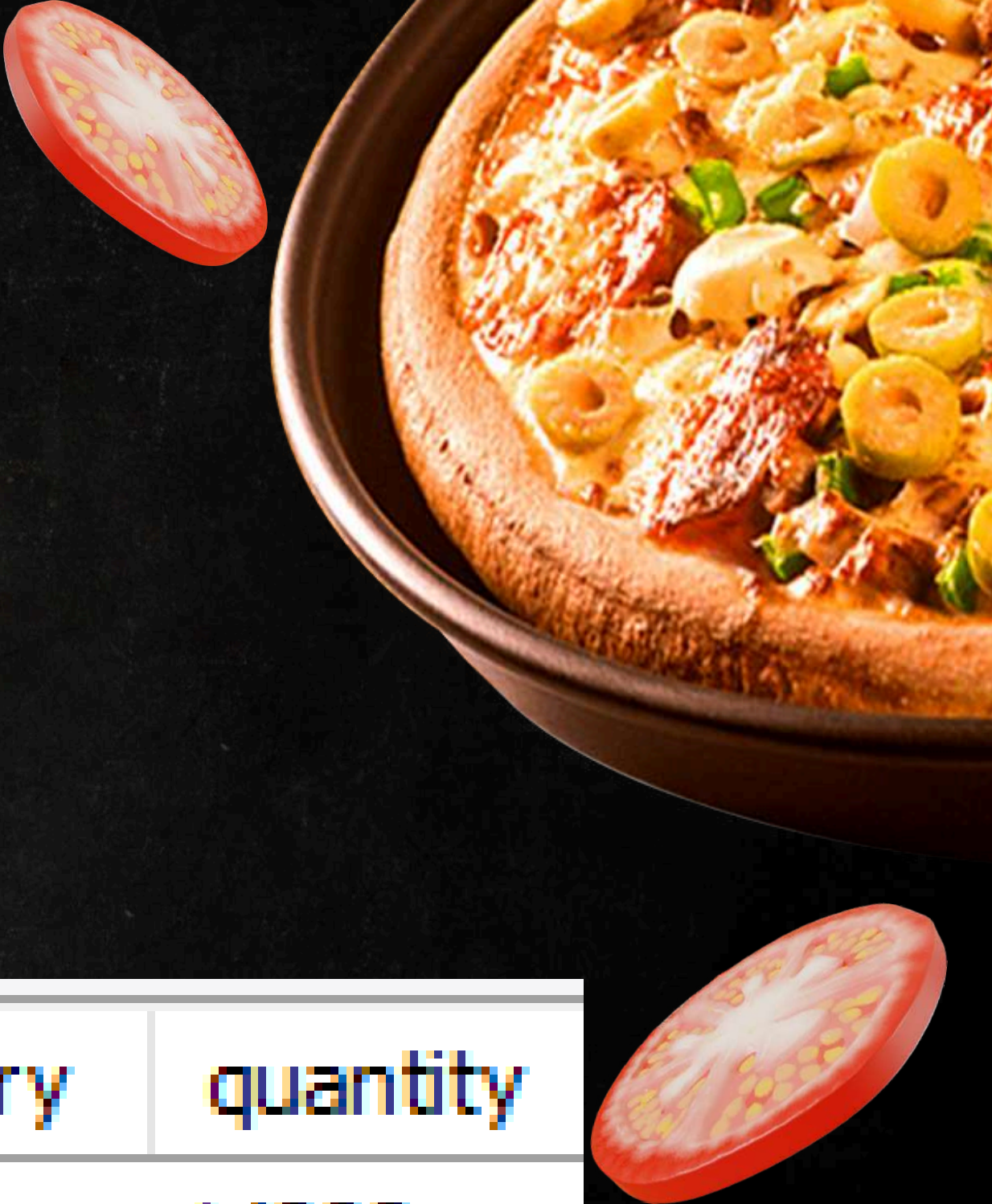




Join the necessary tables to find the total quantity of each pizza category ordered.

```
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;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050





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;
```



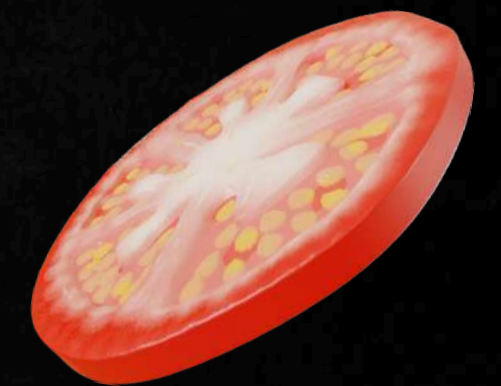
	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



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

```
SELECT
    HOUR(orders.time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(orders.time);
```

HOUR(orders.time)	COUNT(order_id)
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





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

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

	ROUND(AVG(quantity), 0)
▶	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;
```

	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;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

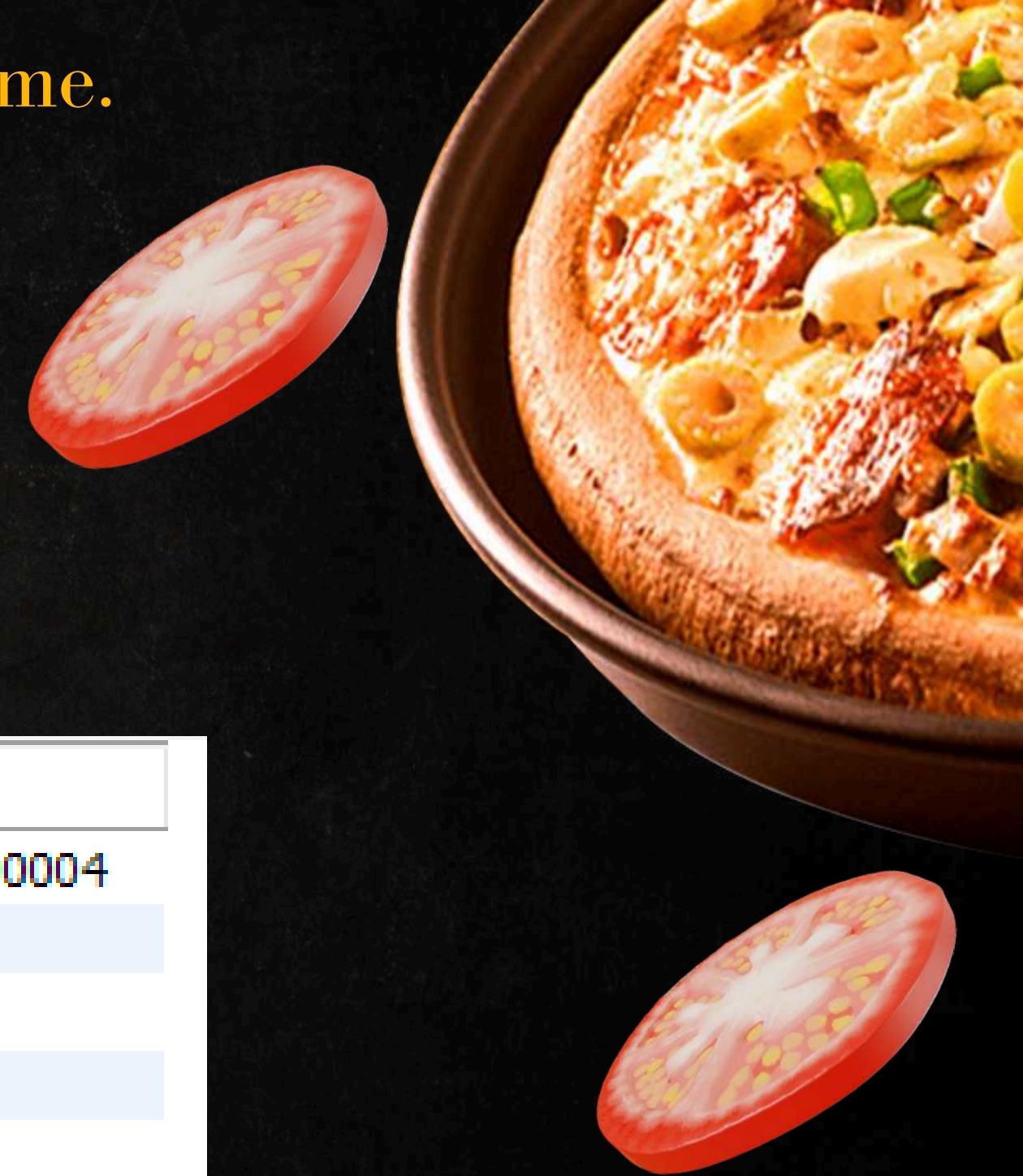




# Analyze the cumulative revenue generated over time.

```
select date,sum(revenue) over (order by date) as cum_revenue
from
(select 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
```

	date	cum_revenue
►	2015-01-01	2713.85000000000004
	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
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4
	2015-01-10	23990.3500000000002

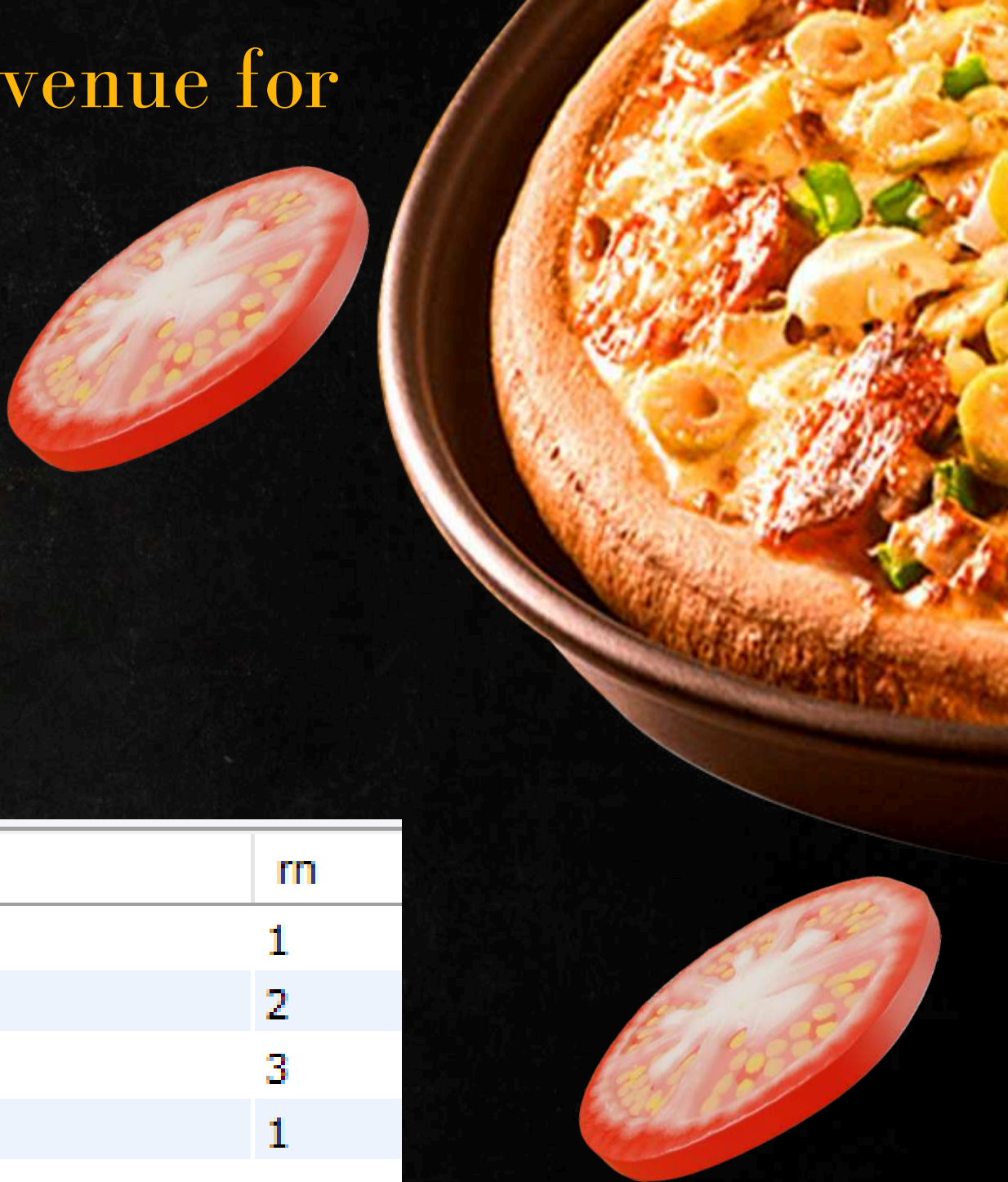




# Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
SELECT category, name, revenue, rn
FROM (
  SELECT pizza_types.category, pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue,
    RANK() OVER (PARTITION BY pizza_types.category ORDER BY SUM(order_details.quantity * pizzas.price) DESC) AS rn
  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
WHERE rn <= 3;
```

	category	name	revenue	rn
►	Chicken	The Thai Chicken Pizza	43434.25	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3
	Classic	The Classic Deluxe Pizza	38180.5	1
	Classic	The Hawaiian Pizza	32273.25	2
	Classic	The Pepperoni Pizza	30161.75	3
	Supreme	The Spicy Italian Pizza	34831.25	1
	Supreme	The Italian Supreme Pizza	33476.75	2
	Supreme	The Sicilian Pizza	30940.5	3
	Veggie	The Four Cheese Pizza	32265.700000000065	1
	Veggie	The Mexicana Pizza	26780.75	2
	Veggie	The Five Cheese Pizza	26066.5	3





# INSIGHTS

- The Classic Deluxe Pizza takes the crown as the most ordered item, with an impressive 2453 orders .
- The Greek pizza is the highest - priced pizza.
- The Classic Pizza reigns supreme in terms of revenue ,contributing a significant 26.91% of our total earnings.
- Large pizzas dominate our orders ,with a staggering 18,526 orders . Clearly our customers love their pizza with a generous serving.
- On average ,we sell around 138 pizzas every day ,showcasing our consistent popularity and demand

