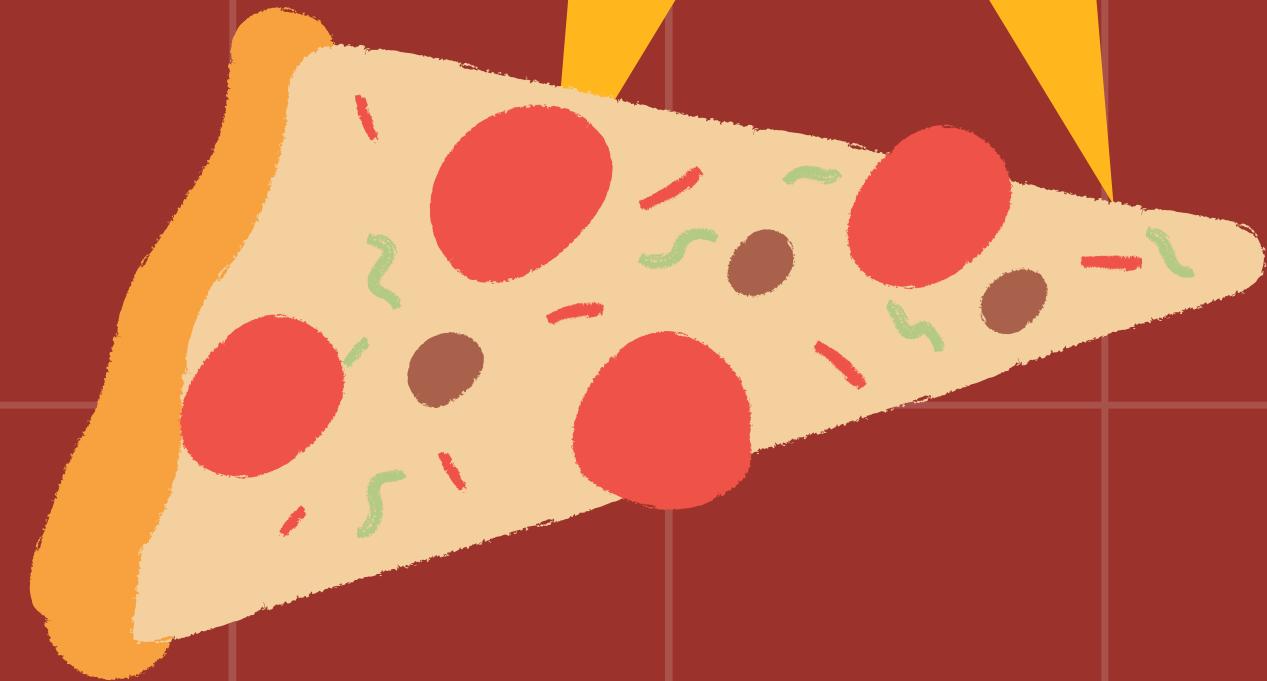


PINOCCHIO'S PIZZA & SUBS SALES ANALYSIS



INTRODUCTION

The "Pizza Sales Analysis" project aims to analyze pizza sales data by leveraging SQL queries to gain actionable insights into sales performance. This analysis helps in understanding customer preferences, sales trends, and revenue distribution, enabling better business decisions and strategies.

Let's start our adventure in the world of pizza!

QUERIES

Retrieve the total number of orders placed.

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

```
use pizza_sales;  
select count(order_id) as total_orders from orders;
```



Result Grid	
	total_orders
▶	21350



-- Calculate the total revenue generated from pizza sales.

```
SELECT  
    SUM(order_details.quantity * pizzas.price) AS total_revenue  
FROM  
    order_details  
    JOIN  
    pizzas ON order_details.pizza_id = pizzas.pizza_id
```

Result Grid | Filter Rows:

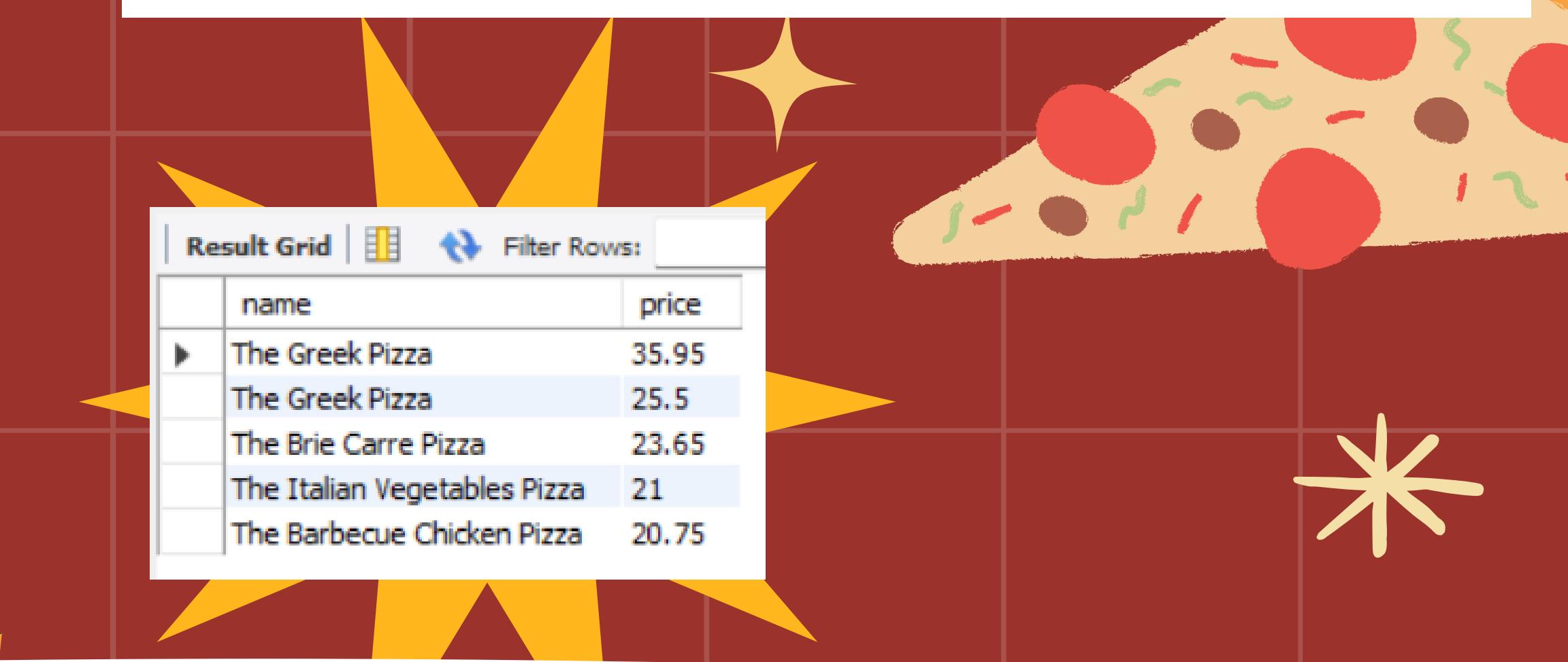
	total_revenue
▶	817860.049999993

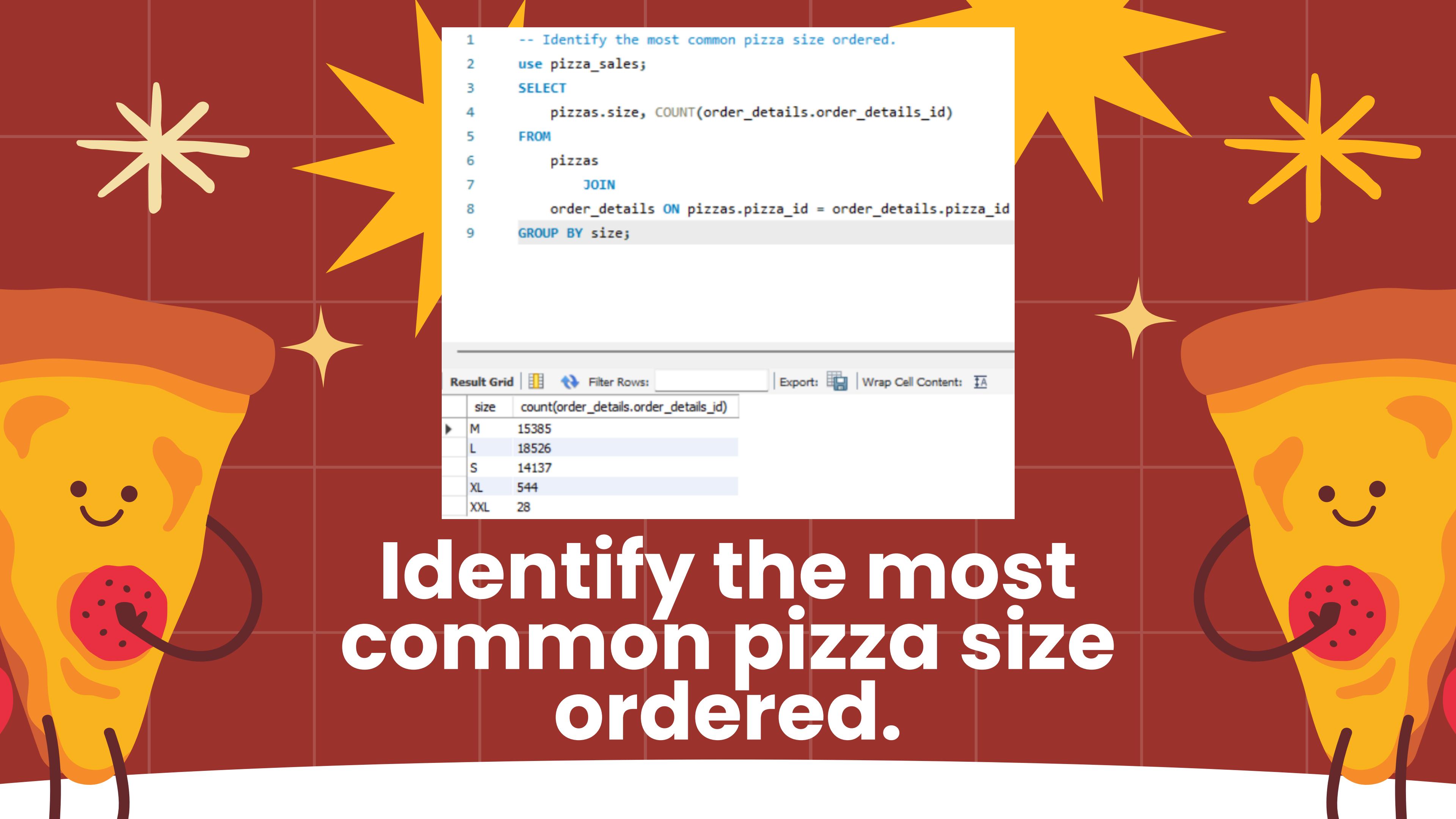
Calculate the total revenue generated from pizza sales.

Identify the highest-priced pizza.

```
-- Identify the highest-priced pizza.  
use pizza_sales;  
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 5;
```

	name	price
▶	The Greek Pizza	35.95
	The Greek Pizza	25.5
	The Brie Carre Pizza	23.65
	The Italian Vegetables Pizza	21
	The Barbecue Chicken Pizza	20.75





```
1  -- Identify the most common pizza size ordered.
2  use pizza_sales;
3  SELECT
4      pizzas.size, COUNT(order_details.order_details_id)
5  FROM
6      pizzas
7      JOIN
8          order_details ON pizzas.pizza_id = order_details.pizza_id
9  GROUP BY size;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

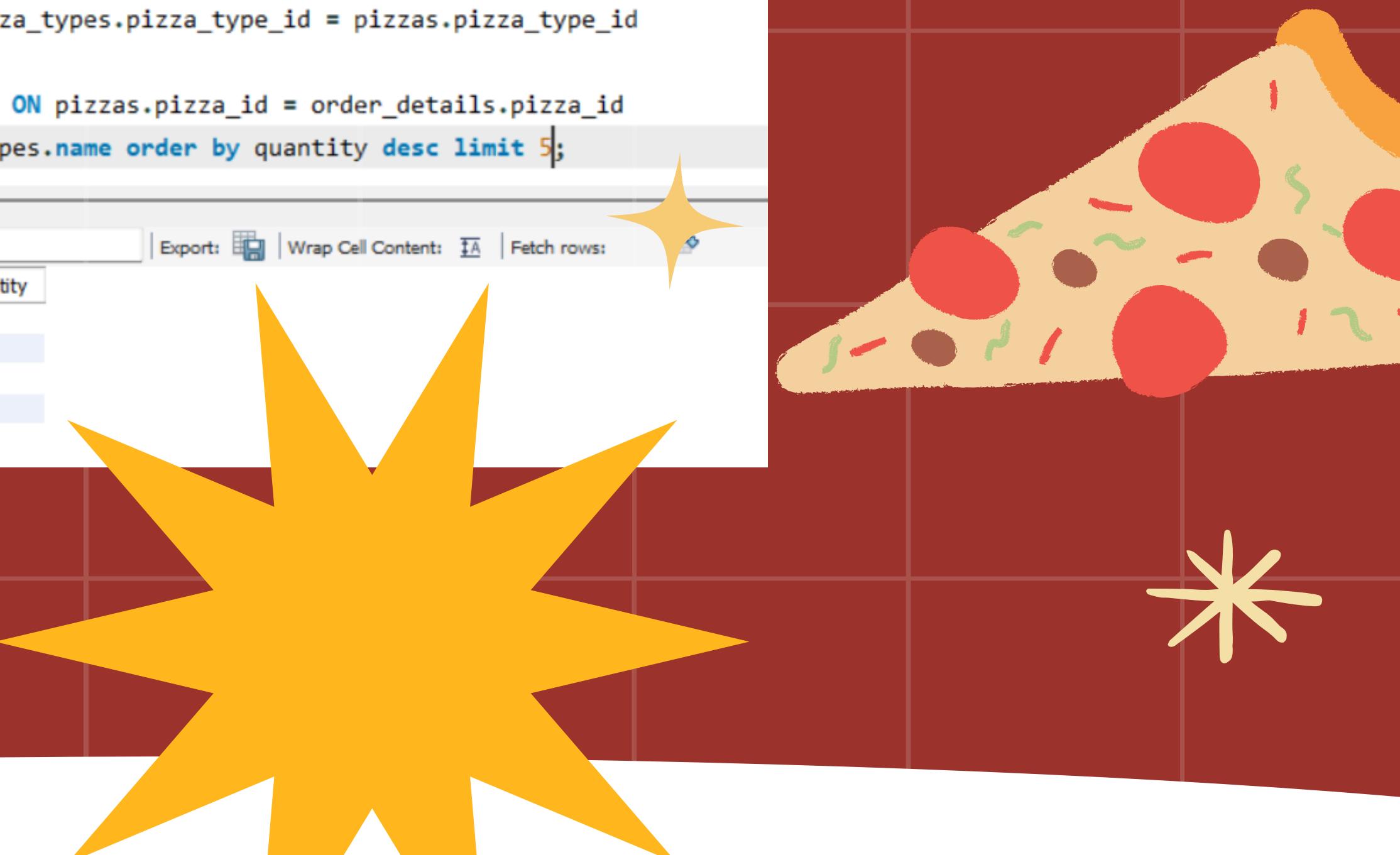
size	count(order_details.order_details_id)
M	15385
L	18526
S	14137
XL	544
XXL	28

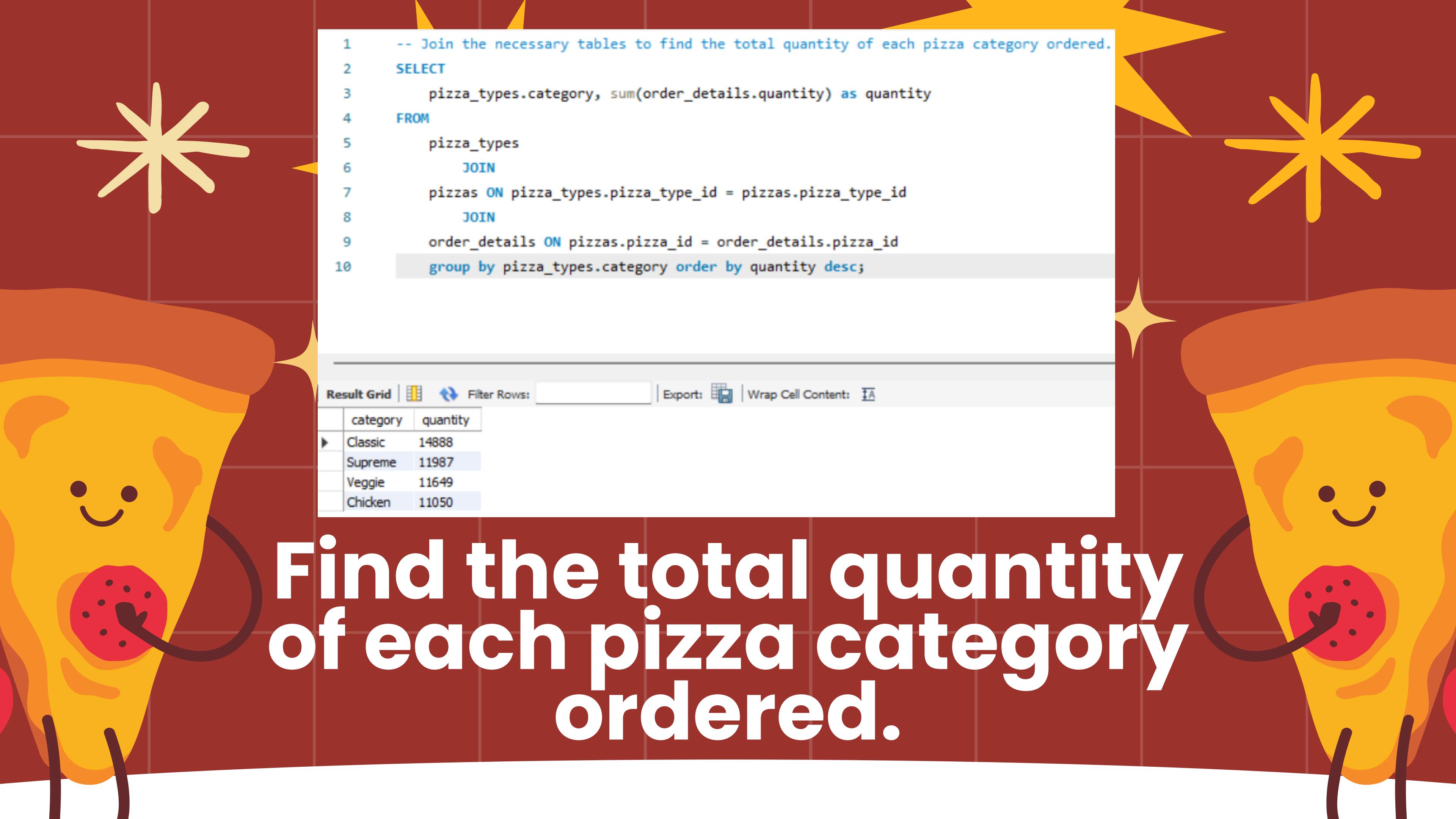
Identify the most common pizza size ordered.

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

```
1  -- List the top 5 most ordered pizza types along with their quantities.  
2  use pizza_sales;  
3  SELECT  
4      pizza_types.name, sum(order_details.quantity) as quantity  
5  FROM  
6      pizza_types  
7      JOIN  
8      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
9      JOIN  
10     order_details ON pizzas.pizza_id = order_details.pizza_id  
11    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





```
1  -- Join the necessary tables to find the total quantity of each pizza category ordered.
2  SELECT
3      pizza_types.category, sum(order_details.quantity) as quantity
4  FROM
5      pizza_types
6      JOIN
7          pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8      JOIN
9          order_details ON pizzas.pizza_id = order_details.pizza_id
10     group by pizza_types.category order by quantity desc;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

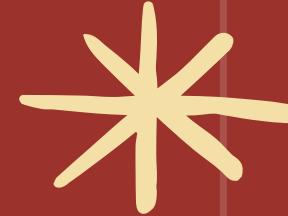
Find the total quantity of each pizza category ordered.

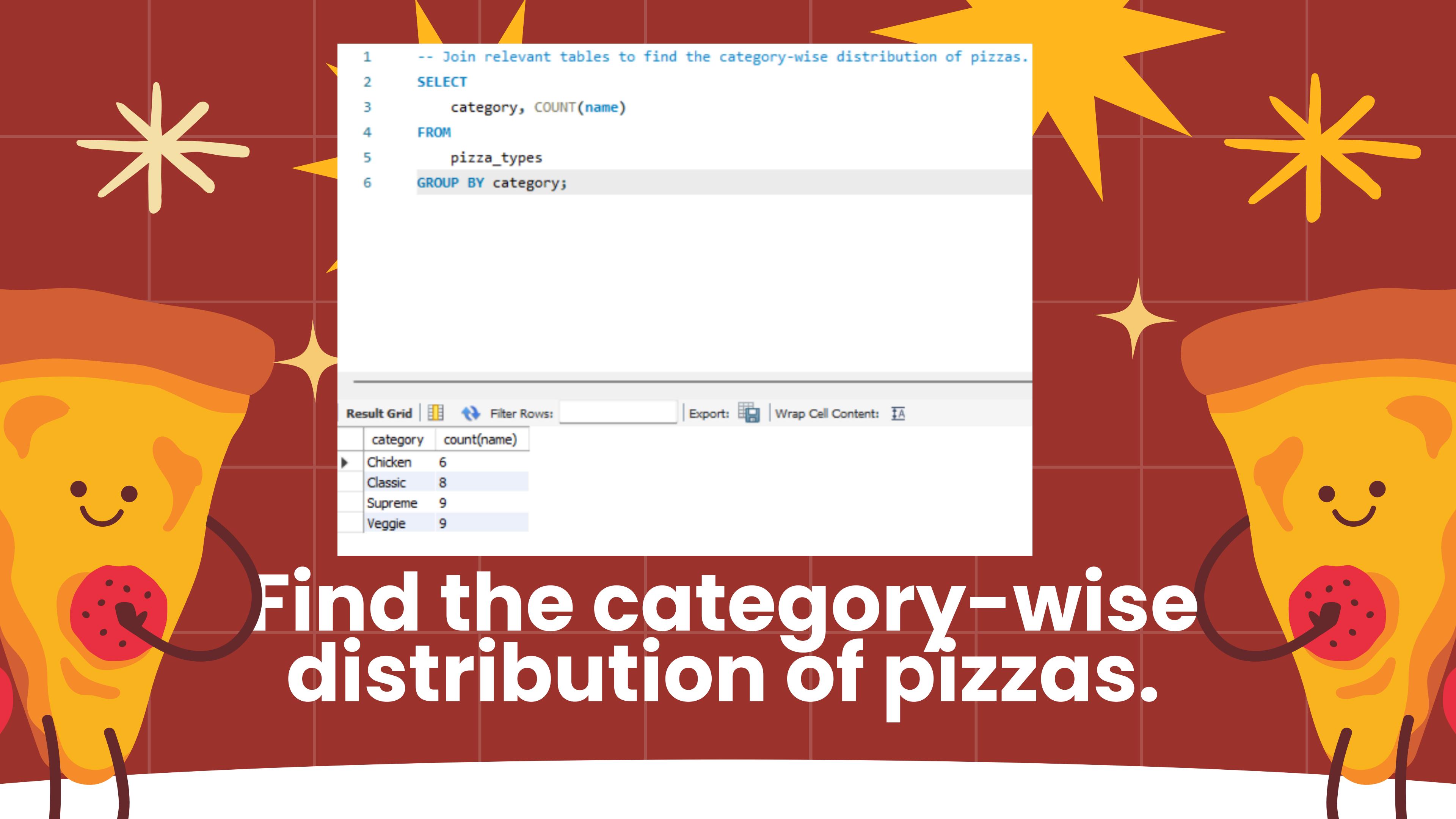
Determine the distribution of orders by hour of the day.

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	orders	hours
1	1231	11
2	2520	12
3	2455	13
4	1472	14
5	1468	15
6	1920	16
7	2336	17
8	2399	18
9	2009	19
10	1642	20
11	1198	21
12	663	22
13	28	23
14	8	10
15	1	9





```
1  -- Join relevant tables to find the category-wise distribution of pizzas.  
2  SELECT  
3      category, COUNT(name)  
4  FROM  
5      pizza_types  
6  GROUP BY category;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

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

Find the category-wise distribution of pizzas.

Analyze the cumulative revenue generated over time.

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

order_date	cum_revenue
2015-01-01 00:00:00	2713.8500000000004
2015-01-02 00:00:00	5445.75
2015-01-03 00:00:00	8108.15
2015-01-04 00:00:00	9863.6
2015-01-05 00:00:00	11929.55
2015-01-06 00:00:00	14358.5
2015-01-07 00:00:00	16560.7
2015-01-08 00:00:00	19399.05
2015-01-09 00:00:00	21526.4
2015-01-10 00:00:00	23990.35000000002
2015-01-11 00:00:00	25862.65
2015-01-12 00:00:00	27781.7
2015-01-13 00:00:00	29831.30000000003
2015-01-14 00:00:00	32358.70000000004
2015-01-15 00:00:00	34343.50000000001
2015-01-16 00:00:00	36937.65000000001



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

Result Grid	
name	profit
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

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

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

```
1 -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2 • select name, category, revenue  from
3   (select category, name ,revenue, rank() over (partition by category order by revenue desc) as rnk
4   from
5     (select pizza_types.category,pizza_types.name, (sum(order_details.quantity * pizzas.price)) as revenue
6     from pizza_types
7     join
8       pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
9     join
10    order_details on pizzas.pizza_id = order_details.pizza_id
11   group by pizza_types.name,pizza_types.category  ) as sales) as a
12  where rnk <= 3;
```

name	category	revenue
The Thai Chicken Pizza	Chicken	43434.25
The Barbecue Chicken Pizza	Chicken	42768
The California Chicken Pizza	Chicken	41409.5
The Classic Deluxe Pizza	Classic	38180.5
The Hawaiian Pizza	Classic	32273.25
The Pepperoni Pizza	Classic	30161.75
The Spicy Italian Pizza	Supreme	34831.25
The Italian Supreme Pizza	Supreme	33476.75
The Sicilian Pizza	Supreme	30940.5
The Four Cheese Pizza	Veggie	32265.7000000065
The Mexicana Pizza	Veggie	26780.75
The Five Cheese Pizza	Veggie	26066.5

CONCLUSION

The Pizza Sales Analysis project provides valuable insights into sales performance, customer preferences, and revenue distribution.

By utilizing SQL queries, the analysis helps in making informed business decisions, optimizing product offerings, and improving overall sales strategies.