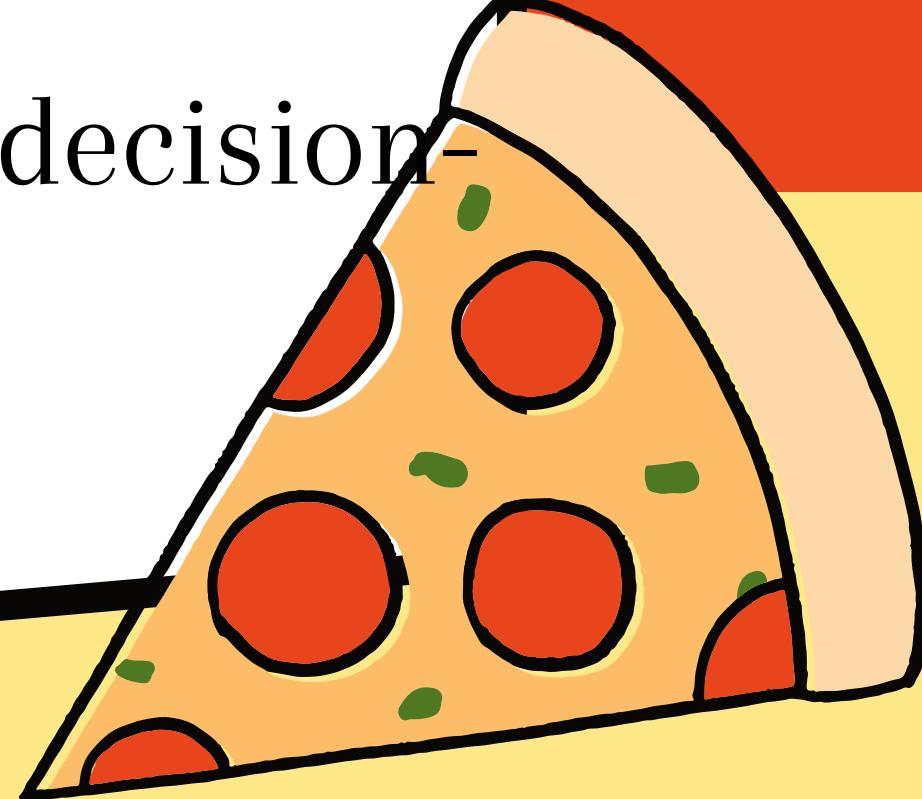




SQL Project on Pizza Sales

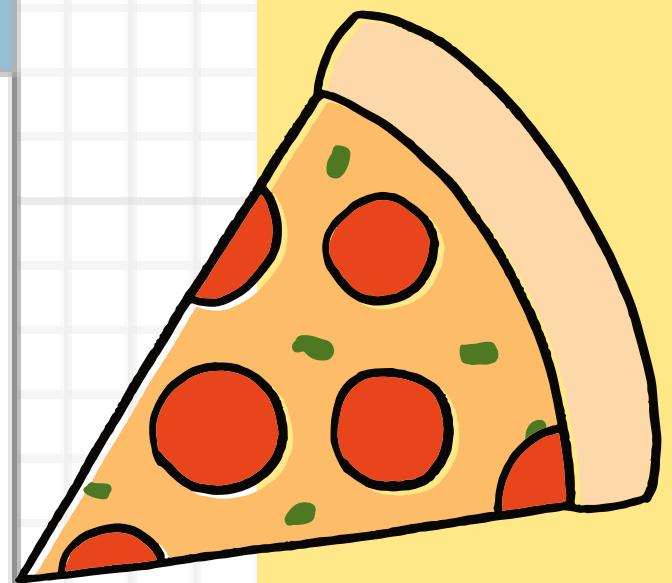
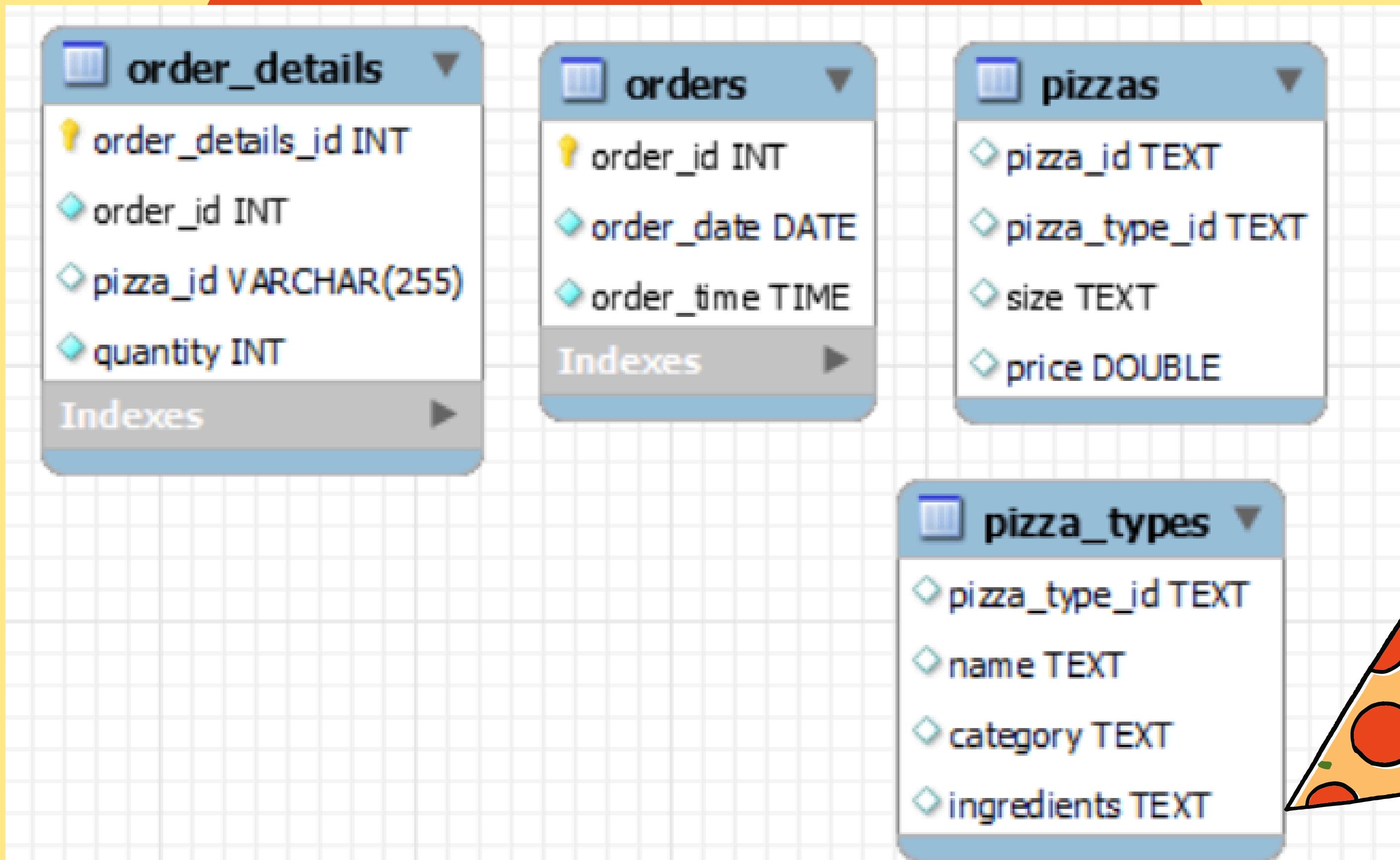


Introduction



Hello, I'm *Somnath Pandit*. In this project, I've leveraged SQL queries to analyze pizza sales data, uncovering insights for strategic decision-making.

Schema





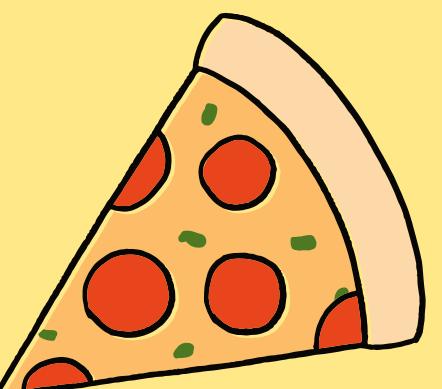
RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED



```
USE pizza_hut;  
  
SELECT COUNT(*) Total_Orders  
FROM orders;
```

Result Grid

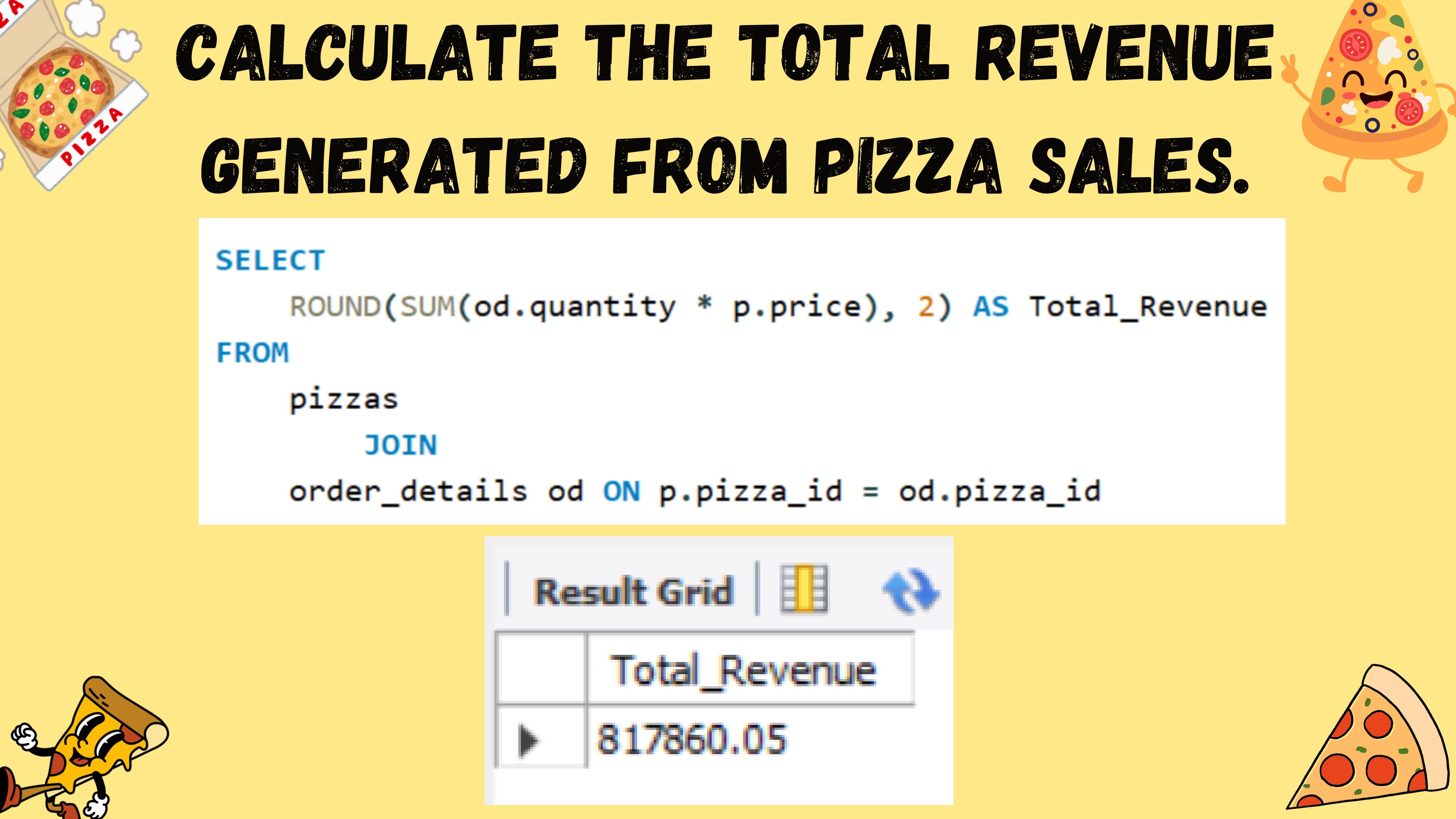
Total_Orders
21350



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT  
    ROUND(SUM(od.quantity * p.price), 2) AS Total_Revenue  
FROM  
    pizzas  
    JOIN  
    order_details od ON p.pizza_id = od.pizza_id
```

Result Grid	
	Total_Revenue
▶	817860.05



IDENTIFY THE HIGHEST-PRICED PIZZA.

```
SELECT  
    *  
FROM  
    pizzas  
ORDER BY price DESC  
LIMIT 1;
```

Result Grid | Filter Rows:

	pizza_id	pizza_type_id	size	price
▶	the_greek_xxL	the_greek	XXL	35.95

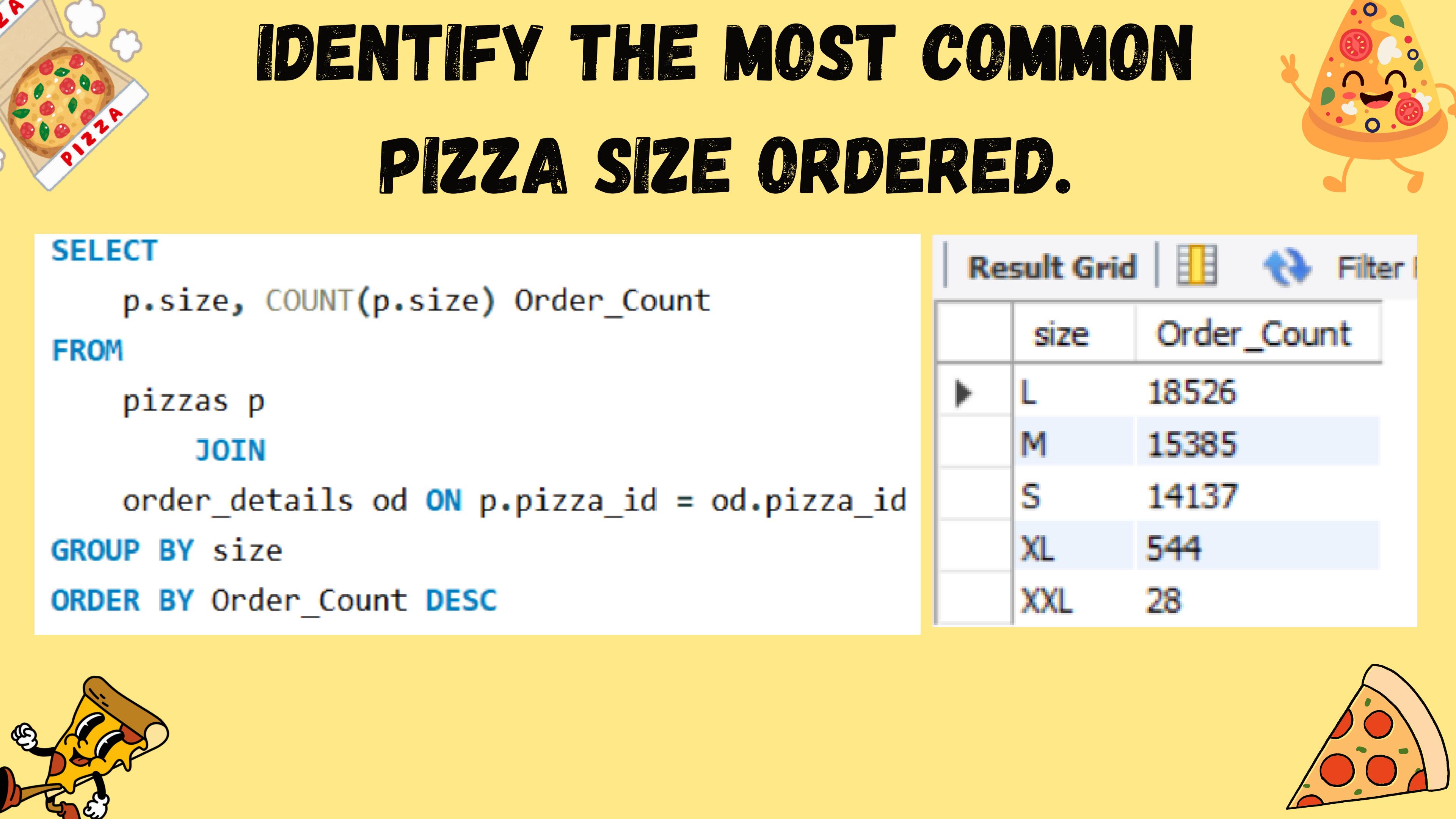


IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT  
    p.size, COUNT(p.size) Order_Count  
FROM  
    pizzas p  
    JOIN  
    order_details od ON p.pizza_id = od.pizza_id  
GROUP BY size  
ORDER BY Order_Count DESC
```

Result Grid | Filter

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

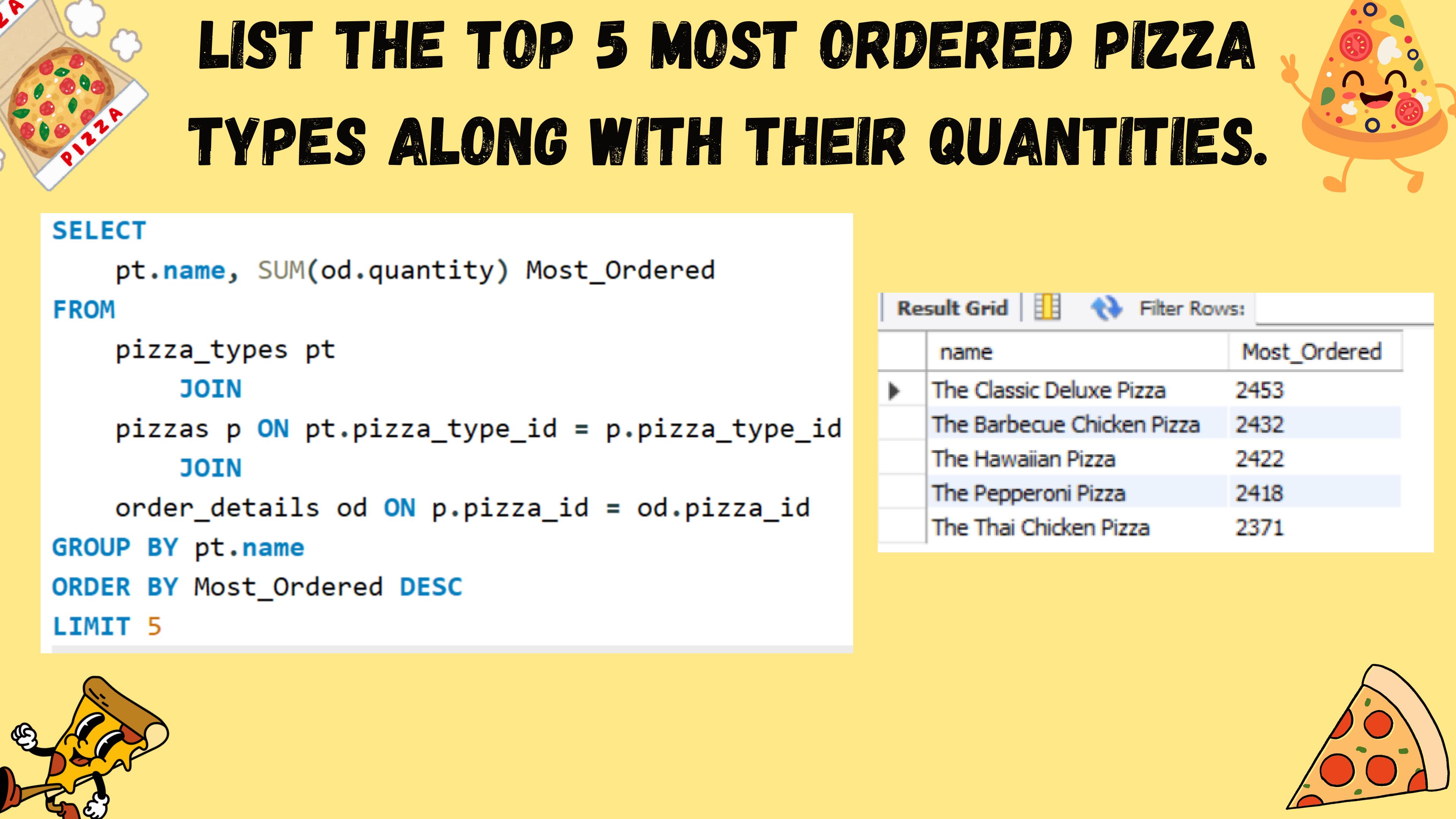


LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

SELECT

```
pt.name, SUM(od.quantity) Most_Ordered  
FROM  
    pizza_types pt  
        JOIN  
    pizzas p ON pt.pizza_type_id = p.pizza_type_id  
        JOIN  
    order_details od ON p.pizza_id = od.pizza_id  
GROUP BY pt.name  
ORDER BY Most_Ordered DESC  
LIMIT 5
```

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

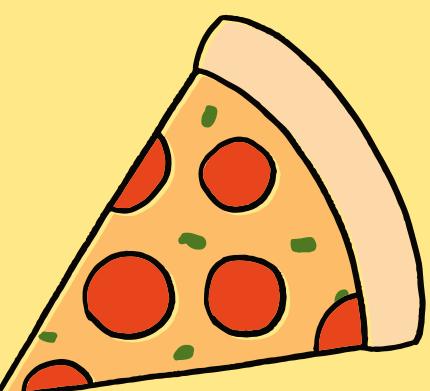


JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
SELECT
    category, SUM(od.quantity) Category_Count
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY category
ORDER BY Category_Count DESC;
```

Result Grid | Filter Rows:

	category	
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

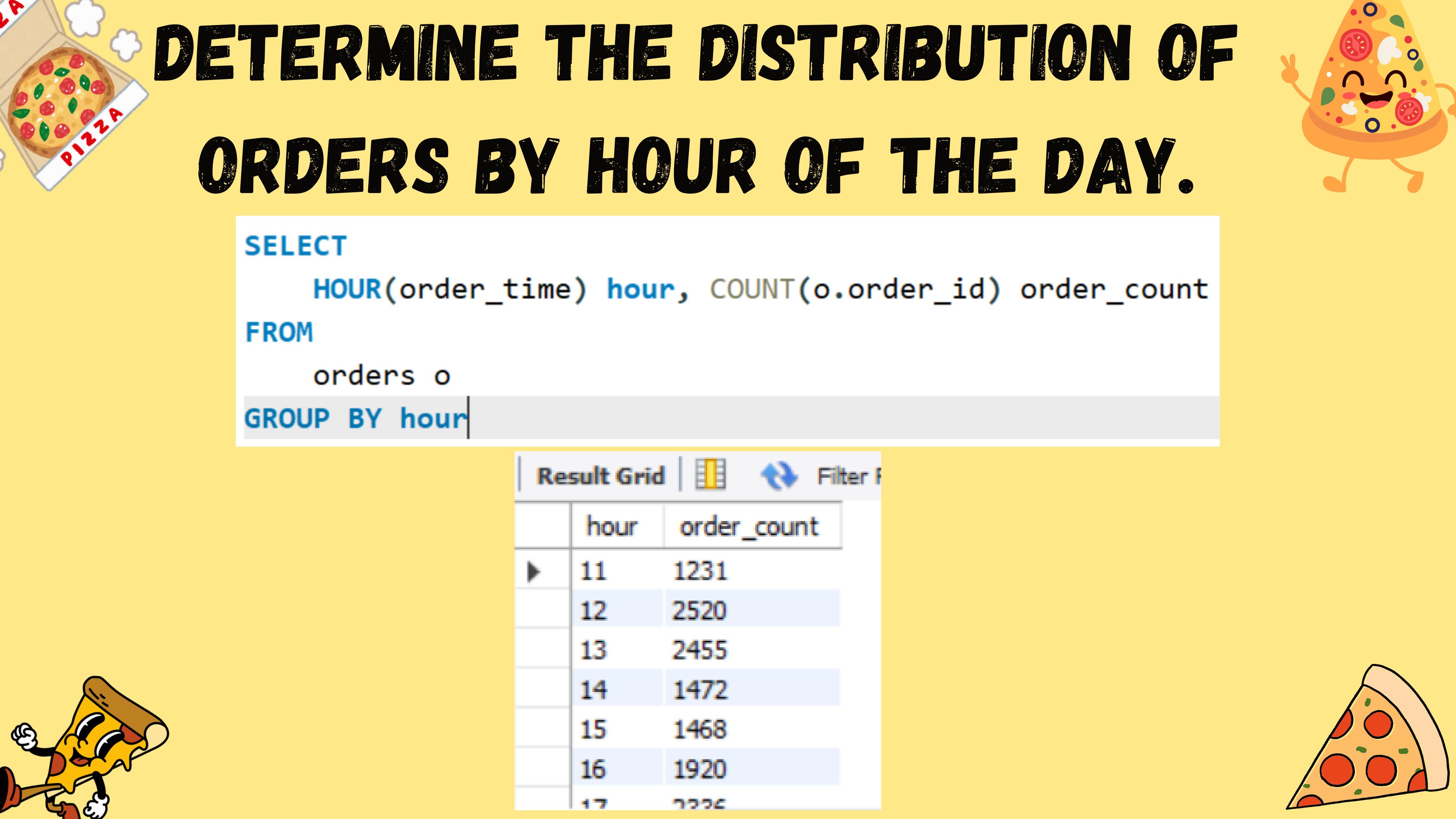


DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT  
    HOUR(order_time) hour, COUNT(o.order_id) order_count  
FROM  
    orders o  
GROUP BY hour
```

Result Grid | Filter F

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2226

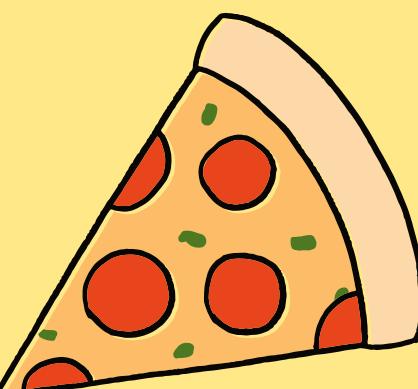


JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

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

Result Grid | Filter Rows

	category	category_count
▶	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(ordered_quantity)) Avg_pizza_order_per_day

FROM

(**SELECT**

order_date, SUM(od.quantity) ordered_quantity

FROM

orders o

JOIN order_details od **ON** o.order_id = od.order_id

GROUP BY order_date) **AS** Date_wise_order_quantity

Result Grid



Filter Rows:

Avg_pizza_order_per_day
138

138

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

SELECT

pt.name, SUM(od.quantity * p.price) Total_revenue

FROM

pizzas p

JOIN

pizza_types pt ON p.pizza_type_id = pt.pizza_type_id

JOIN

order_details od ON p.pizza_id = od.pizza_id

GROUP BY pt.name

ORDER BY Total_revenue DESC

LIMIT 3

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



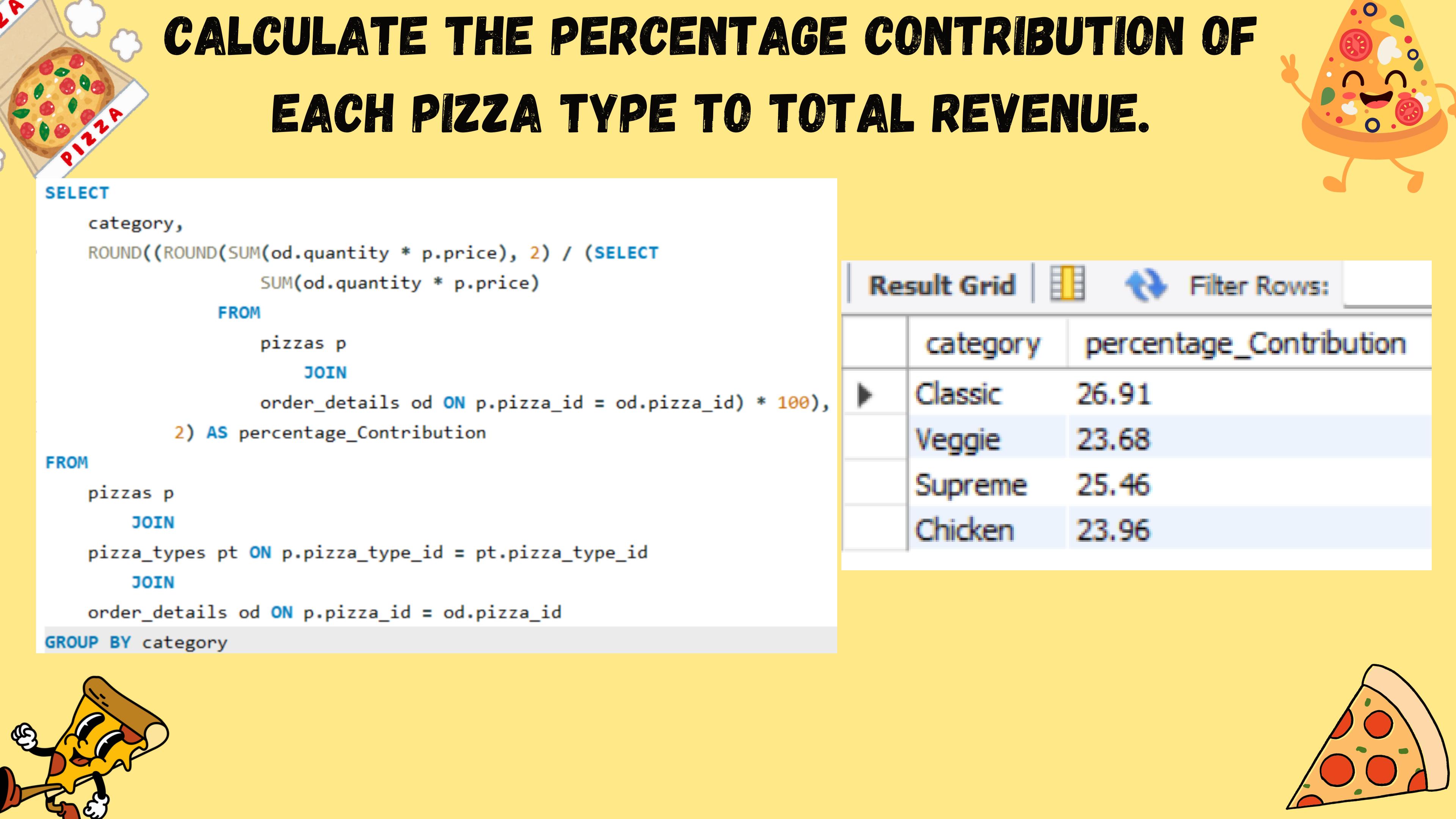
CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

SELECT

```
category,  
ROUND((ROUND(SUM(od.quantity * p.price), 2) / (SELECT  
SUM(od.quantity * p.price)  
FROM  
pizzas p  
JOIN  
order_details od ON p.pizza_id = od.pizza_id) * 100),  
2) AS percentage_Contribution  
FROM  
pizzas p  
JOIN  
pizza_types pt ON p.pizza_type_id = pt.pizza_type_id  
JOIN  
order_details od ON p.pizza_id = od.pizza_id  
GROUP BY category
```

Result Grid | Filter Rows:

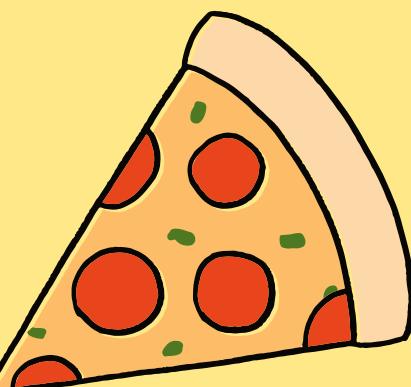
	category	percentage_Contribution
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96



ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
SELECT order_date,  
       SUM(Revenue) OVER (ORDER BY order_date) Cumulative_Revenue  
  FROM  
    (SELECT o.order_date, ROUND(SUM(od.quantity * p.price),2) Revenue  
     FROM orders o  
    JOIN order_details od  
    ON o.order_id = od.order_id  
    JOIN pizzas p  
    ON od.pizza_id = p.pizza_id  
   GROUP BY o.order_date) sales
```

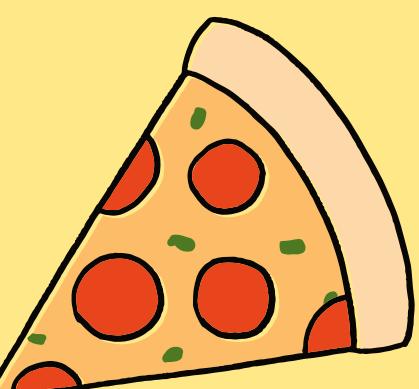
	order_date	Cumulative_Revenue
▶	2015-01-01	2713.85
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55



DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
SELECT category, name, revenue
FROM
(SELECT category, name, revenue,
RANK() OVER (Partition by category ORDER BY revenue desc) rn_category
FROM
(SELECT pt.name, category, SUM(od.quantity * p.price) revenue
FROM pizza_types pt
JOIN pizzas p
ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od
ON p.pizza_id = od.pizza_id
GROUP BY category, pt.name) a) b
WHERE rn_category <= 3
```

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





THANK
YOU