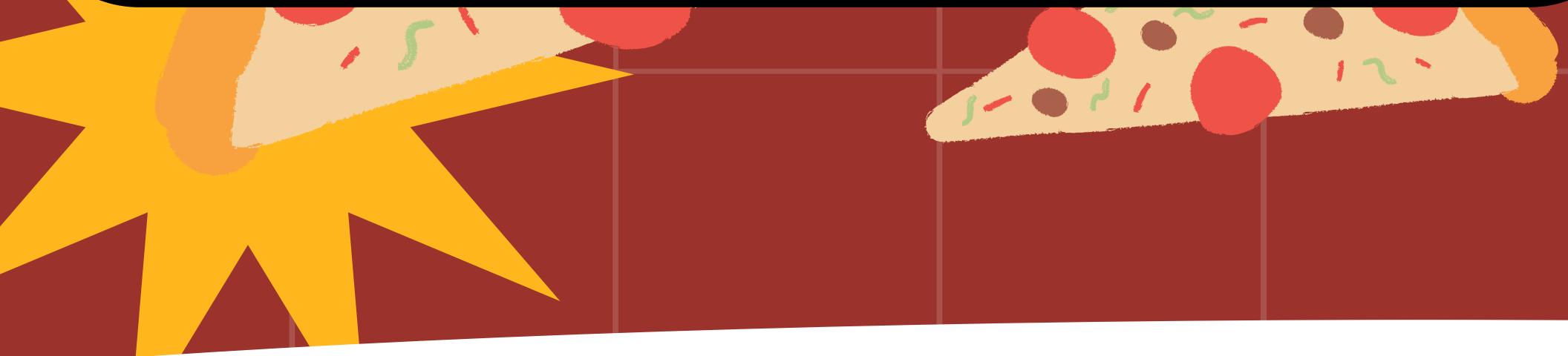


PIZZA-SALE SQL PROJECT



HELLO!



Hello my name is Arpit Rathore
and in this Project i have utilize
SQL query to solve question that
where related to pizzas sales
generaly



INTRODUCTION



This SQL project focuses on analyzing and managing data related to pizza sales. The project aims to create a database schema that efficiently tracks customer orders, inventory, and sales transactions. By utilizing SQL queries, the project will provide insights into sales patterns, popular products, and inventory management, ultimately helping to optimize business operations and enhance decision-making for a pizza restaurant.

[Pizza_Sales_SQL Project](#)

Retrieve the total number of orders placed.
aragraph text

```
use pizza_hut;  
  
SELECT  
  
    COUNT(order_id)  
  
FROM  
  
    orders;
```

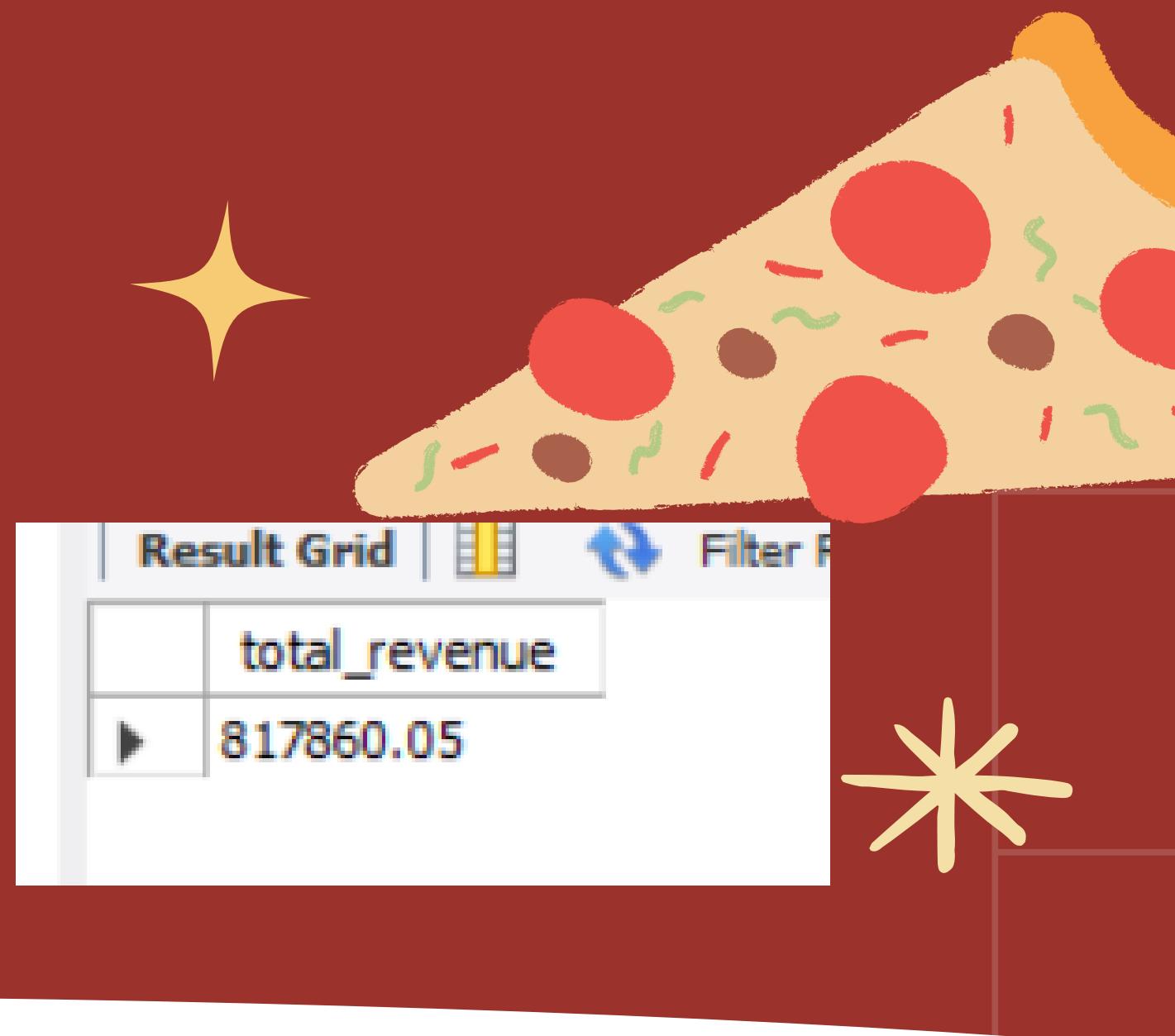


A slice of pepperoni pizza with green toppings, including green bell peppers and onions, resting on a dark surface.

Result Grid	
	COUNT(order_id)
▶	21350

Calculate the total revenue generated from pizza sales.

```
3 •   SELECT
4     ROUND(SUM(quantity * price), 2) AS total_revenue
5
6   FROM
7     orders_details
8   JOIN
9     pizzas ON orders_details.pizza_id = pizzas.pizza_id;
```



dentify the highest-priced pizza

```
3 •   SELECT
4       pizza_types.name, pizzas.price
5   FROM
6       pizza_types
7       JOIN
8           pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9   ORDER BY price DESC
10  LIMIT 1;
```



Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

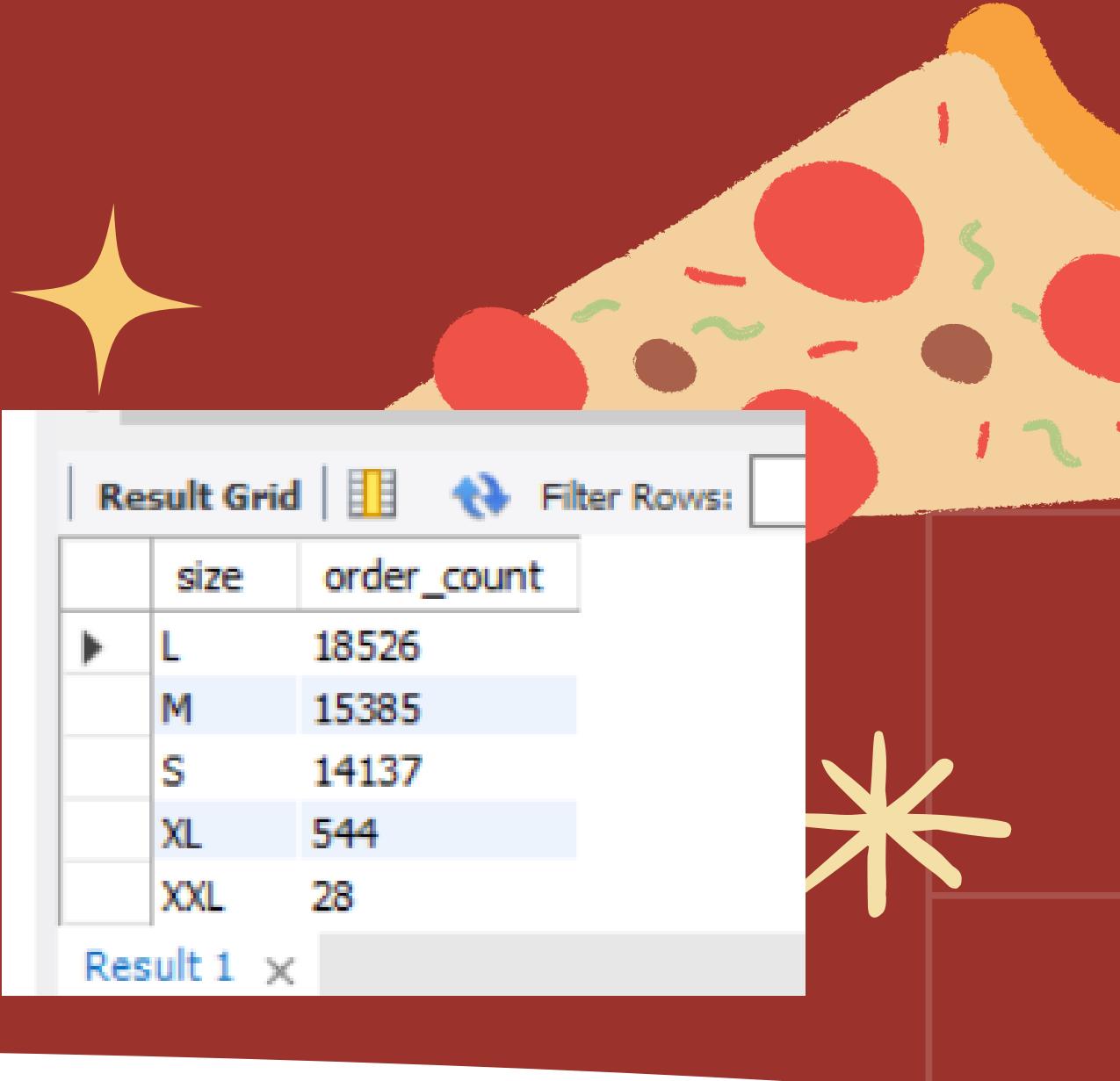
Identify the most common pizza size ordered.

```
3 • SELECT
4     pizzas.size,
5     COUNT(orders_details.order_details_id) as order_count
6 FROM
7     pizzas
8     JOIN
9         orders_details ON pizzas.pizza_id = orders_details.pizza_id
0 GROUP BY pizzas.size
1 ORDER BY (COUNT(orders_details.order_details_id)) DESC;
```

Result Grid | Filter Rows:

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

Result 1 ×



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

```
2 •   SELECT
3       pizza_types.name, SUM(orders_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       orders_details ON orders_details.pizza_id = pizzas.pizza_id
10  GROUP BY name
11  ORDER BY quantity desc
12  LIMIT 5;
```



A decorative illustration of a slice of pepperoni pizza with a starburst effect in the background.

	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

Result 1 ×

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

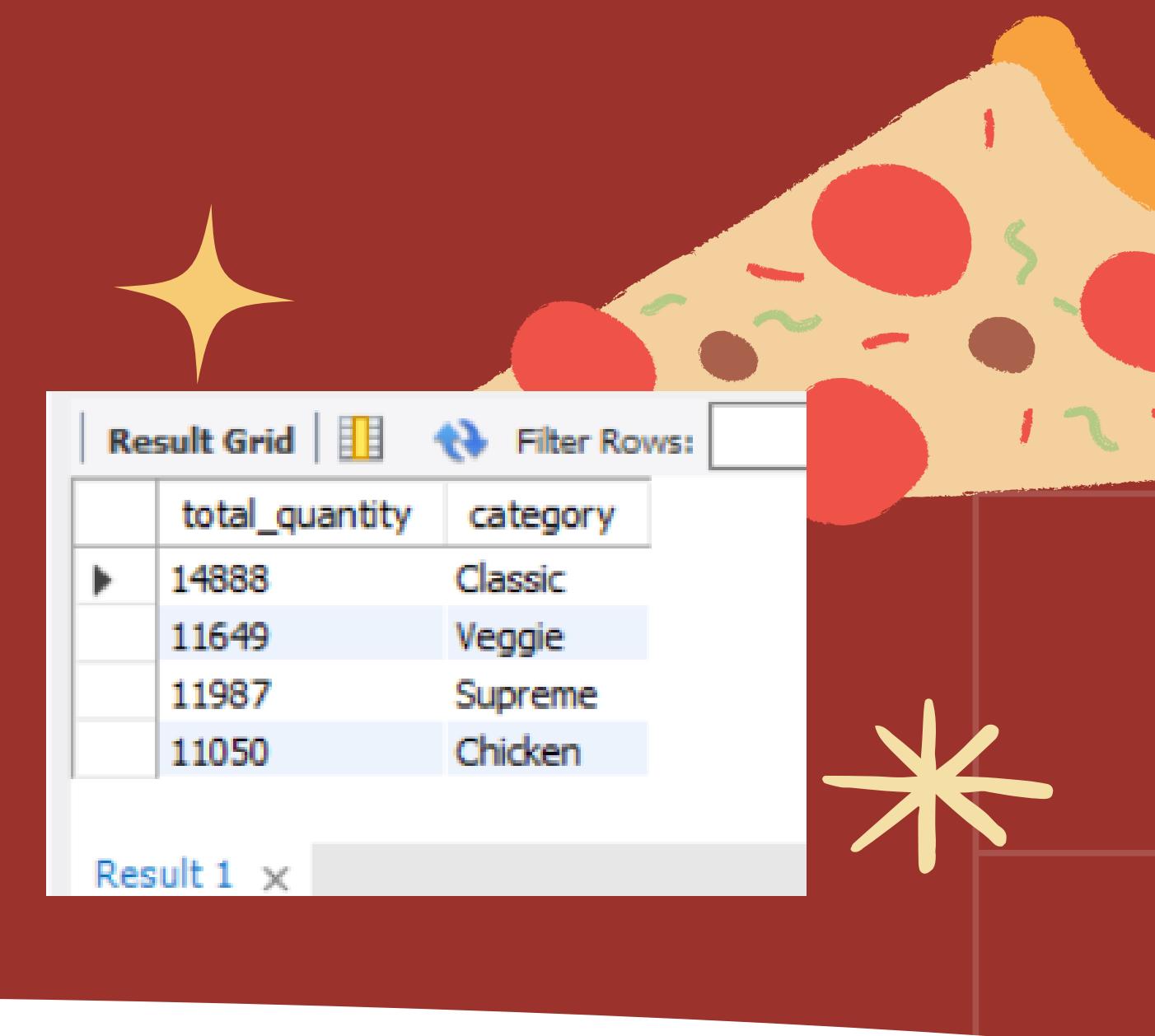
- **SELECT**

```
    SUM(orders_details.quantity) AS total_quantity,  
    pizza_types.category  
FROM  
    orders_details  
        JOIN  
    pizzas ON orders_details.pizza_id = pizzas.pizza_id  
        JOIN  
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
GROUP BY category;
```

Result Grid | Filter Rows:

	total_quantity	category
▶	14888	Classic
	11649	Veggie
	11987	Supreme
	11050	Chicken

Result 1 ×



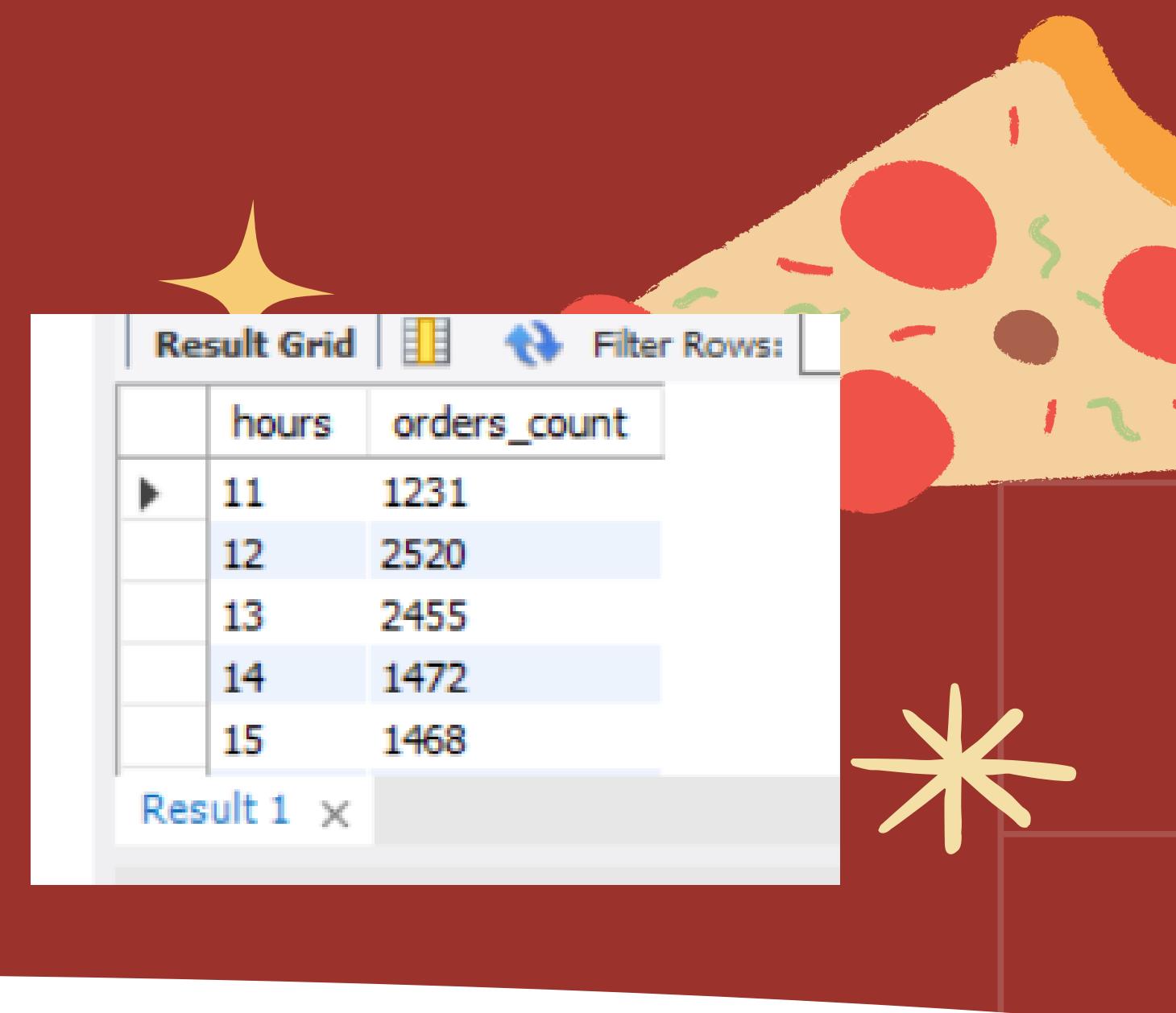
Determine the distribution of orders by hour of the day.

```
• SELECT  
    HOUR(order_time) AS hours, COUNT(order_id) AS orders_count  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

Result Grid | Filter Rows:

	hours	orders_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468

Result 1 ×



join relevant tables to find the category-wise distribution of pizzas.

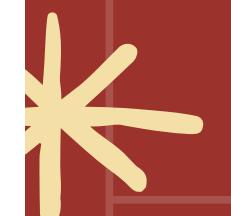
```
SELECT  
    category, COUNT(name)  
FROM  
    pizza_types  
GROUP BY category
```

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

Download

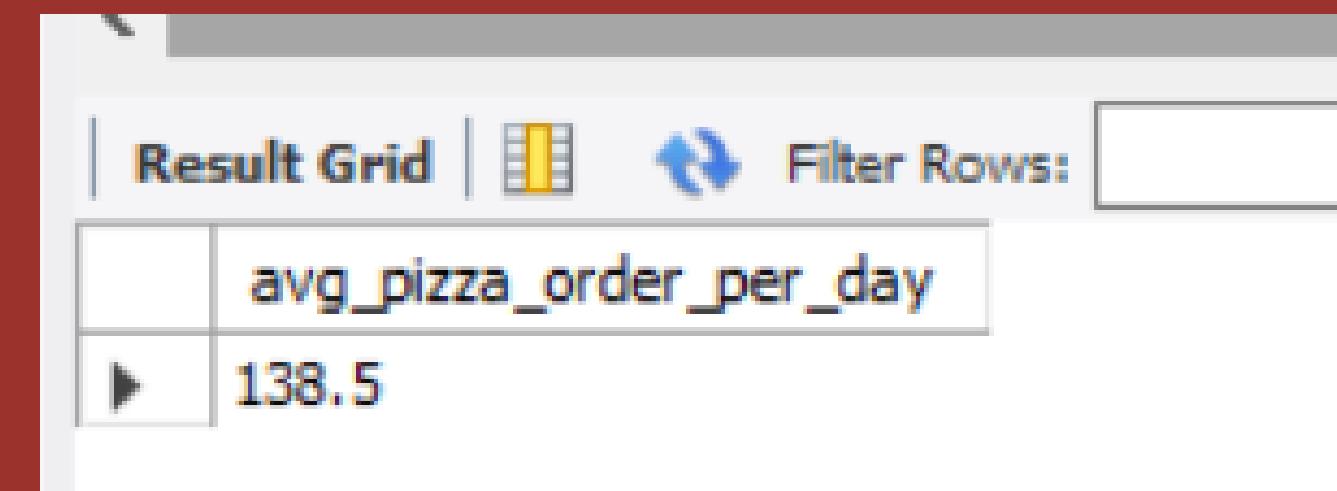
Print

Close



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

```
3 • SELECT
4     ROUND(AVG(order_quantity), 1) as avg_pizza_order_per_day
5 FROM
6     (SELECT
7         orders.order_date,
8             SUM(orders_details.quantity) AS order_quantity
9     FROM
10        orders
11    JOIN orders_details ON orders.order_id = orders_details.order_id
12    GROUP BY orders.order_date) AS total_order;
```



The screenshot shows the MySQL Workbench interface with the results of the executed SQL query. The results are displayed in a grid with one row and two columns. The first column contains the column name 'avg_pizza_order_per_day' and the second column contains the value '138.5'.

avg_pizza_order_per_day	138.5
-------------------------	-------

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

```
3 • SELECT
4     pizza_types.name,
5     SUM(orders_details.quantity * pizzas.price) AS revenue
6 FROM
7     pizza_types
8     JOIN
9     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10    JOIN
11    orders_details ON orders_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY revenue DESC
14 LIMIT 3;
```

Result Grid | Filter Rows: | Export

	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(orders_details.quantity * pizzas.price) / (SELECT  
        ROUND(SUM(quantity * price), 2) AS total_sales  
    FROM  
        orders_details  
        JOIN  
        pizzas ON orders_details.pizza_id = pizzas.pizza_id) * 100,  
    2) AS revenue  
FROM  
    pizza_types  
    JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
    JOIN  
    orders_details ON orders_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 order_date, sum(revenue) over(order by order_date) as com_revenue from
  (SELECT
    orders.order_date,
    round(sum(orders_details.quantity * pizzas.price),2) as revenue
  FROM
    orders_details
    JOIN
    pizzas ON orders_details.pizza_id = pizzas.pizza_id
    JOIN
    orders ON orders.order_id = orders_details.order_id
  GROUP BY orders.order_date) as sales;
```

	order_date	com_revenue
▶	2015-01-01	2713.85
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	2015-01-03
	2015-01-05	11929.55
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05

Result 4 ×

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(orders_details.quantity * pizzas.price) as revenue
FROM
pizza_types
JOIN
pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN
orders_details ON orders_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:

	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

A festive illustration set against a red background with a white grid. In the center, the words "THANK YOU" are written in large, bold, white capital letters. To the left, a person with dark curly hair and a green sweater holds a slice of pizza. To the right, another person with glasses and a green sweater holds a small wrapped gift. Above them, a reindeer with a yellow and orange patterned collar and a bell hangs from its neck. The reindeer has large, expressive eyes and a small smile. The background is decorated with yellow stars and a large yellow starburst at the bottom.

THANK YOU