

PIZZA SALES

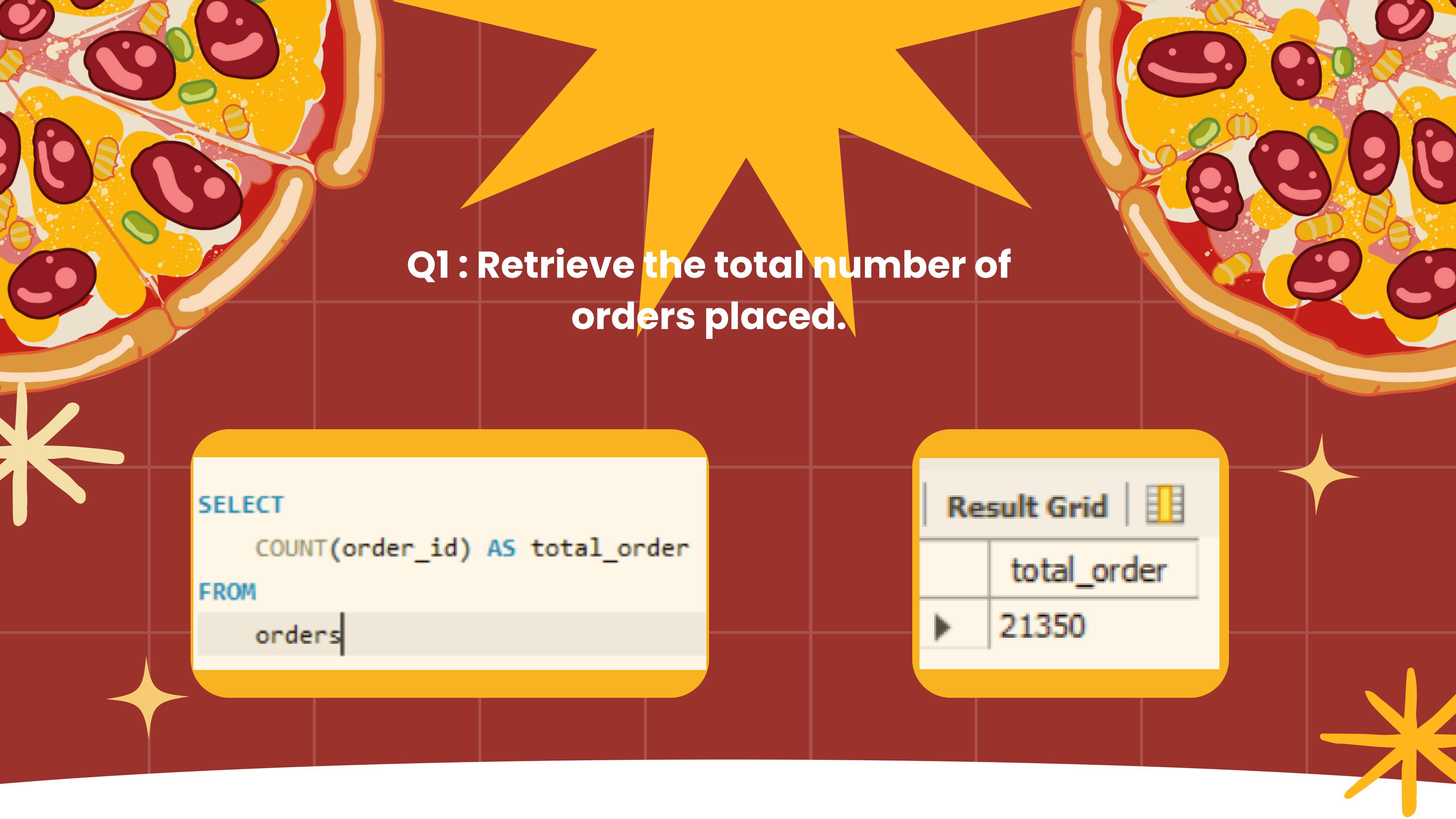
SQL PROJECT USING MYSQL

SQL Database Projects





**MAKE SURE YOU HAVE
ALL THESE INGREDIENTS
BEFORE STARTING THE
PIZZA MAKING PROCESS.**



Q1: Retrieve the total number of orders placed.

```
SELECT  
    COUNT(order_id) AS total_order  
FROM  
    orders
```

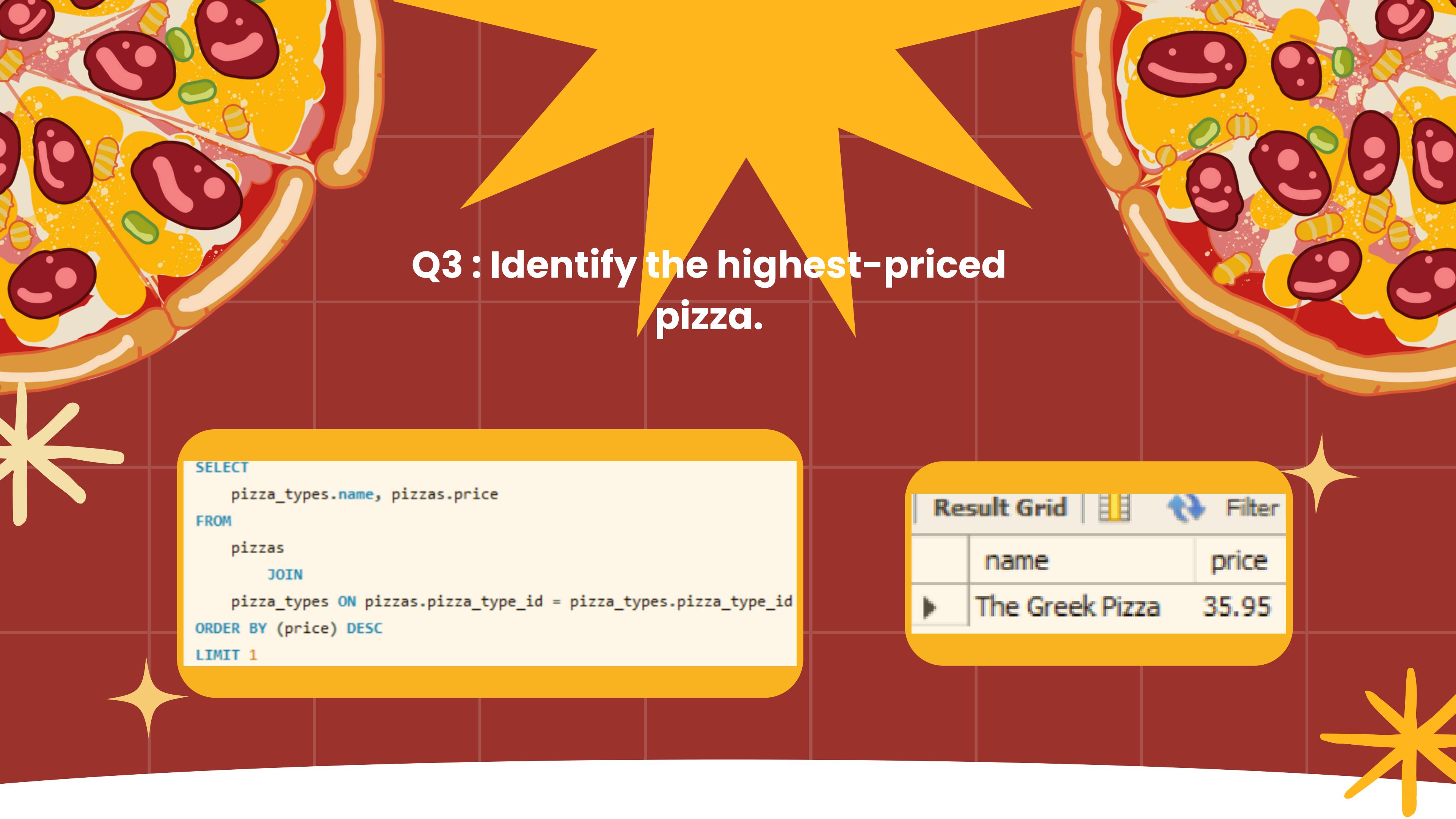
Result Grid

total_order
21350

Q2 : Calculate the total revenue generated from pizza sales.

```
SELECT  
    ROUND(SUM(orderid_detail.quantity * pizzas.price),  
        3) AS total_revenue  
FROM  
    orderid_detail  
    JOIN  
    pizzas ON pizzas.pizza_id = orderid_detail.pizza_id
```

Result Grid	
	total_revenue
▶	334475.25



Q3 : Identify the highest-priced pizza.

```
SELECT  
    pizza_types.name, pizzas.price  
FROM  
    pizzas  
    JOIN  
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
ORDER BY (price) DESC  
LIMIT 1
```

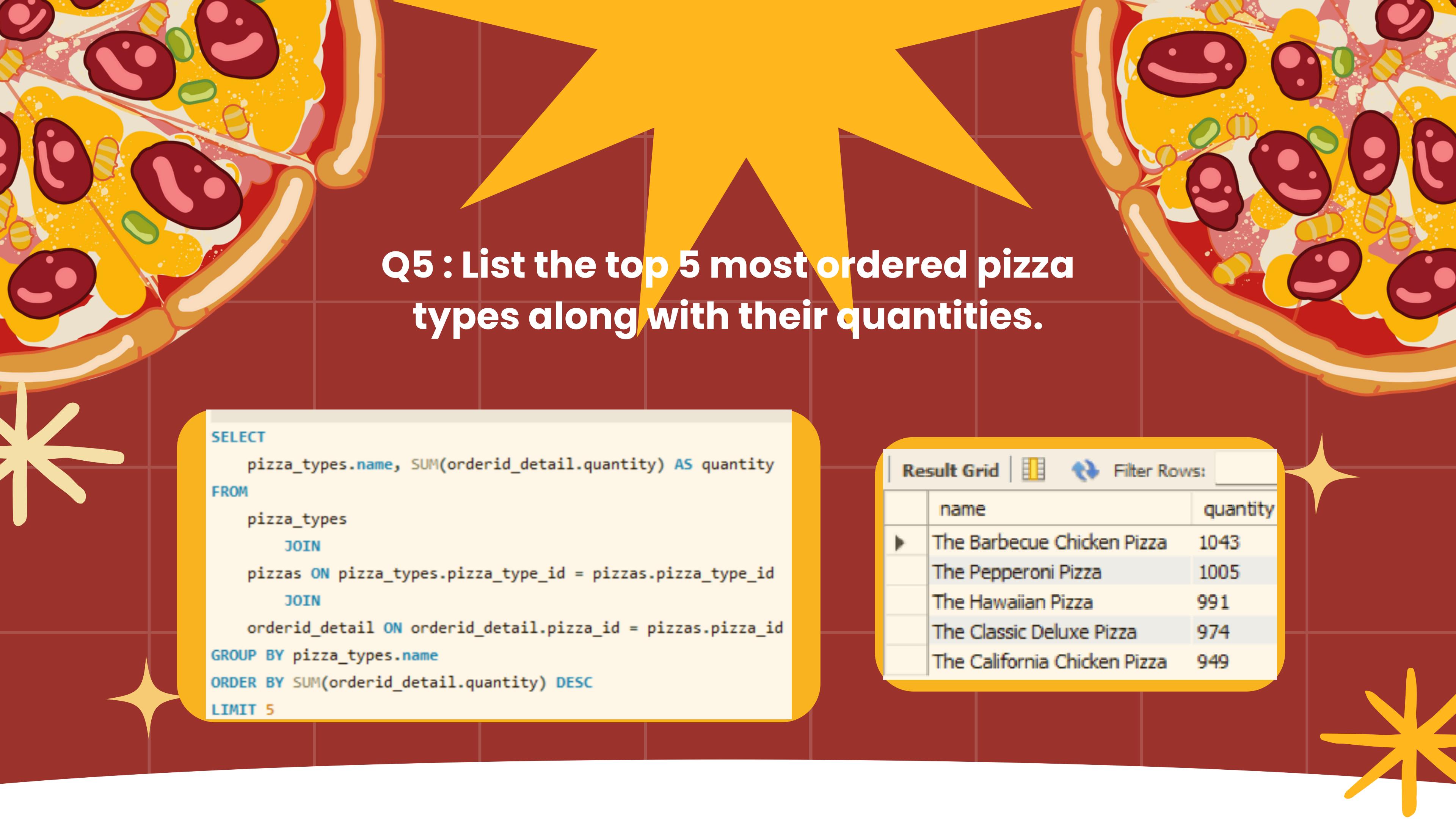
Result Grid | Filter

	name	price
▶	The Greek Pizza	35.95

Q4 : Identify the most common pizza size ordered.

```
SELECT
    pizzas.size, COUNT(orderid_detail.order_detail) as count
FROM
    pizzas
        JOIN
    orderid_detail ON orderid_detail.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY COUNT(orderid_detail.order_detail) DESC
limit 1
```

Result Grid		
	size	count
▶	L	7589



Q5 : List the top 5 most ordered pizza types along with their quantities.

```
SELECT
    pizza_types.name, SUM(orderid_detail.quantity) AS quantity
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    orderid_detail ON orderid_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY SUM(orderid_detail.quantity) DESC
LIMIT 5
```

	name	quantity
▶	The Barbecue Chicken Pizza	1043
	The Pepperoni Pizza	1005
	The Hawaiian Pizza	991
	The Classic Deluxe Pizza	974
	The California Chicken Pizza	949

Q6 : Join the necessary tables to find the total quantity of each pizza category ordered.

```
SELECT
    pizza_types.category,
    SUM(orderid_detail.quantity) AS quantity
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    orderid_detail ON orderid_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC
```

	category	quantity
▶	Classic	6046
▶	Supreme	4883
▶	Veggie	4839
▶	Chicken	4498

Q7 : Determine the distribution of orders by hour of the day.

```
SELECT  
    HOUR(time) AS Hour, COUNT(order_id) AS count_order  
FROM  
    orders  
GROUP BY HOUR(time)
```

Hour	count_order
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

Q8 : Join relevant tables to find the category-wise distribution of pizzas.

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

Result Grid		Filter Rows:
	category	COUNT(category)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Q9 : Group the orders by date and calculate the average number of pizzas ordered per day.

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

Result Grid

average

138

Q10 : Determine the top 3 most ordered pizza types based on revenue.

```
SELECT  
    pizza_types.name,  
    SUM(orderid_detail.quantity * pizzas.price) AS revenue  
FROM  
    pizza_types  
    JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
    JOIN  
    orderid_detail ON orderid_detail.pizza_id = pizzas.pizza_id  
GROUP BY (pizza_types.name)  
ORDER BY revenue DESC  
LIMIT 3
```

	name	revenue
▶	The Barbecue Chicken Pizza	18410.25
	The Thai Chicken Pizza	17236.75
	The California Chicken Pizza	16655.75

Q11: Calculate the percentage contribution of each pizza type to total revenue.

```
select pizza_types.category ,  
round((sum(orderid_detail.quantity * pizzas.price) / (SELECT  
    ROUND(SUM(orderid_detail.quantity * pizzas.price),  
    3) AS total_revenue  
FROM  
    orderid_detail  
    JOIN  
    pizzas ON pizzas.pizza_id = orderid_detail.pizza_id)) * 100,2) as revenue  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join orderid_detail  
on orderid_detail.pizza_id = pizzas.pizza_id  
group by pizza_types.category order by revenue desc
```

	category	per_revenue
▶	Classic	26.71
	Supreme	25.39
	Veggie	24.02
	Chicken	23.87

Q12 : Analyze the cumulative revenue generated over time.

```
select date,  
round(sum(revenue) over(order by date),2) as cum_revenue  
from  
(select (orders.date) ,  
sum(orderid_detail.quantity * pizzas.price) as revenue  
from orderid_detail join pizzas  
on orderid_detail.pizza_id = pizzas.pizza_id  
join orders  
on orders.order_id = orderid_detail.order_id  
group by orders.date)as sales
```

Result Grid		Filter Rows
	date	cum_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
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4

Q13 : 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 rnk
from
(select pizza_types.category,
pizza_types.name,
sum((orderid_detail.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orderid_detail
on orderid_detail.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rnk<=3
```

	name	revenue
▶	The Barbecue Chicken Pizza	18410.25
	The Thai Chicken Pizza	17236.75
	The California Chicken Pizza	16655.75
	The Classic Deluxe Pizza	15165.5
	The Hawaiian Pizza	13151.25
	The Pepperoni Pizza	12524
	The Spicy Italian Pizza	14367
	The Italian Supreme Pizza	13829.25
	The Sicilian Pizza	12383.5
	The Four Cheese Pizza	13717.1000000001
	The Five Cheese Pizza	10878
	The Mexicana Pizza	10623.75



THANK YOU

"Have fun and enjoy every bite"