

SQL PROJECT ON PIZZA SALES

LARANA PIZZA





-- • ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select date,  
       sum(revenue) over (order by date) as cum_revenue  
  from  
(select orders.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  
 group by orders.date) as sales;
```

Result Grid | Filter Rows:

	date	cum_revenue
▶	2015-01-01	2713.850000000004
	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

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;
```

Result Grid | Filter

	category	revenue
▶	Classic	26.96
	Supreme	25.45
	Chicken	23.91
	Veggie	23.69

-- 10 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;
```

Result Grid | Filter Rows:

	name	revenue
▶	The Thai Chicken Pizza	36845.75
	The Barbecue Chicken Pizza	36817.75
	The California Chicken Pizza	35419.5

-- S 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)
→	139



-- 1 RETRIEVE
THE TOTAL
NUMBER OF
ORDERS PLACED.

```
SELECT  
    COUNT(*) AS total_orders  
FROM  
    orders;
```

Result Grid	
	total_orders
▶	21350

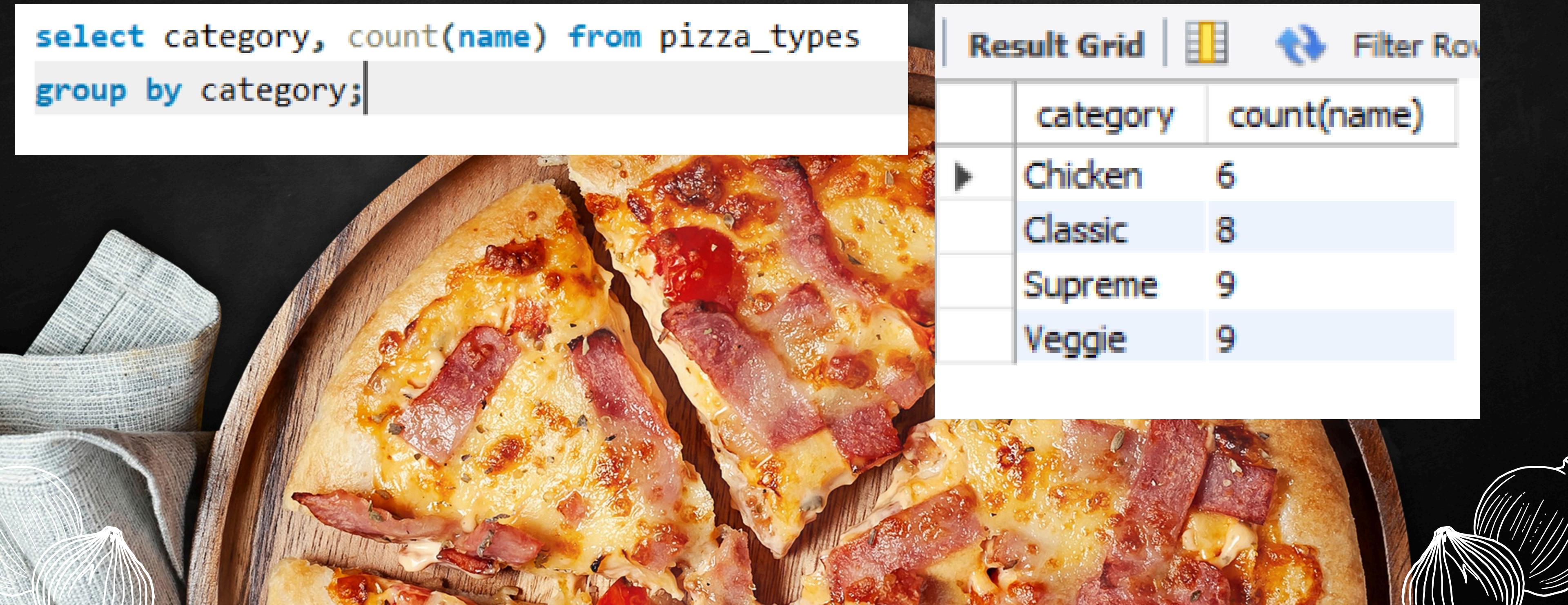


-- 8 JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```
select category, count(name) from pizza_types  
group by category;
```

Result Grid | Filter Row

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

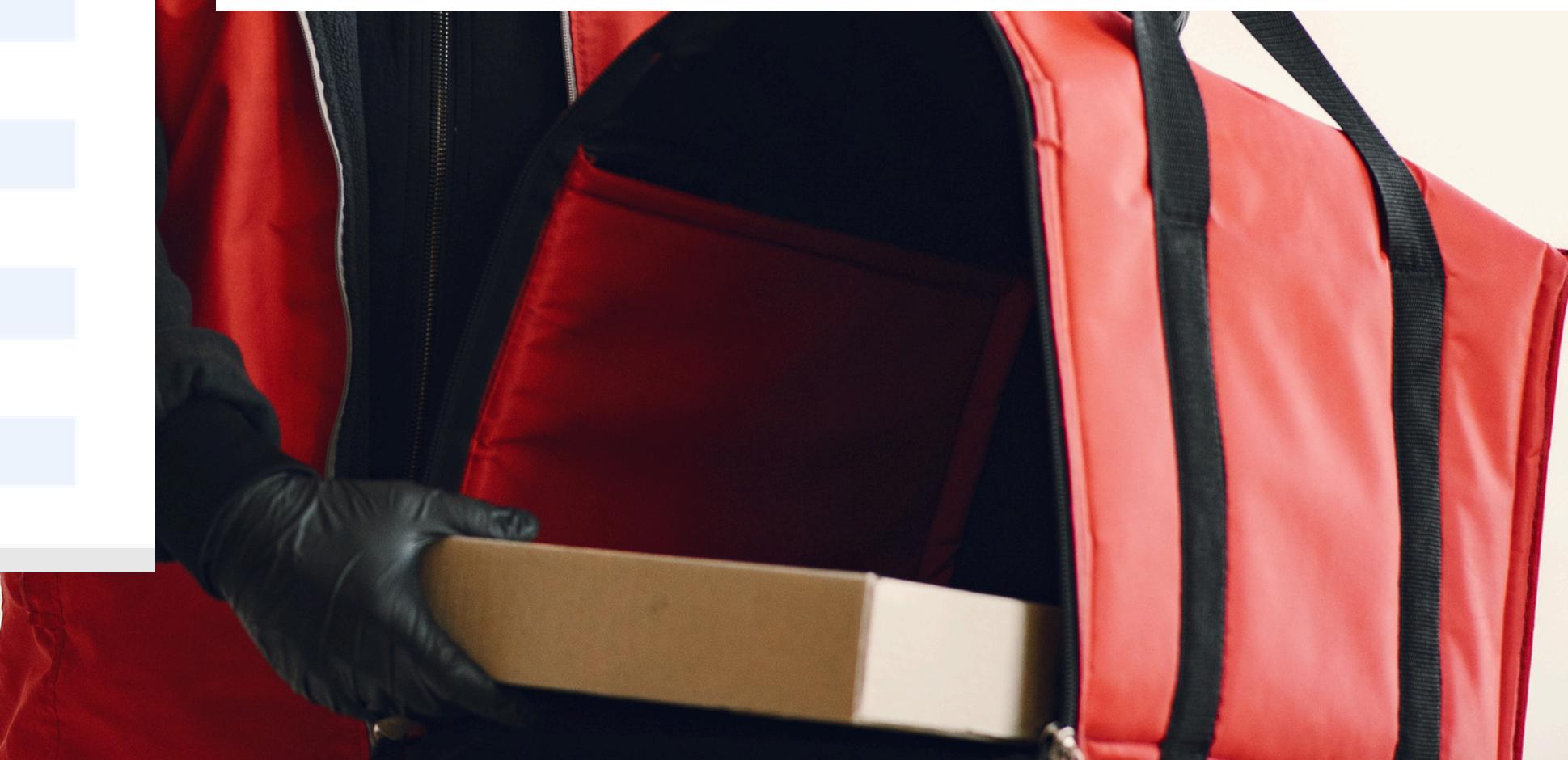


-- 7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

Result Grid | Filter Rows:

	HOUR(time)	COUNT(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009

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





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

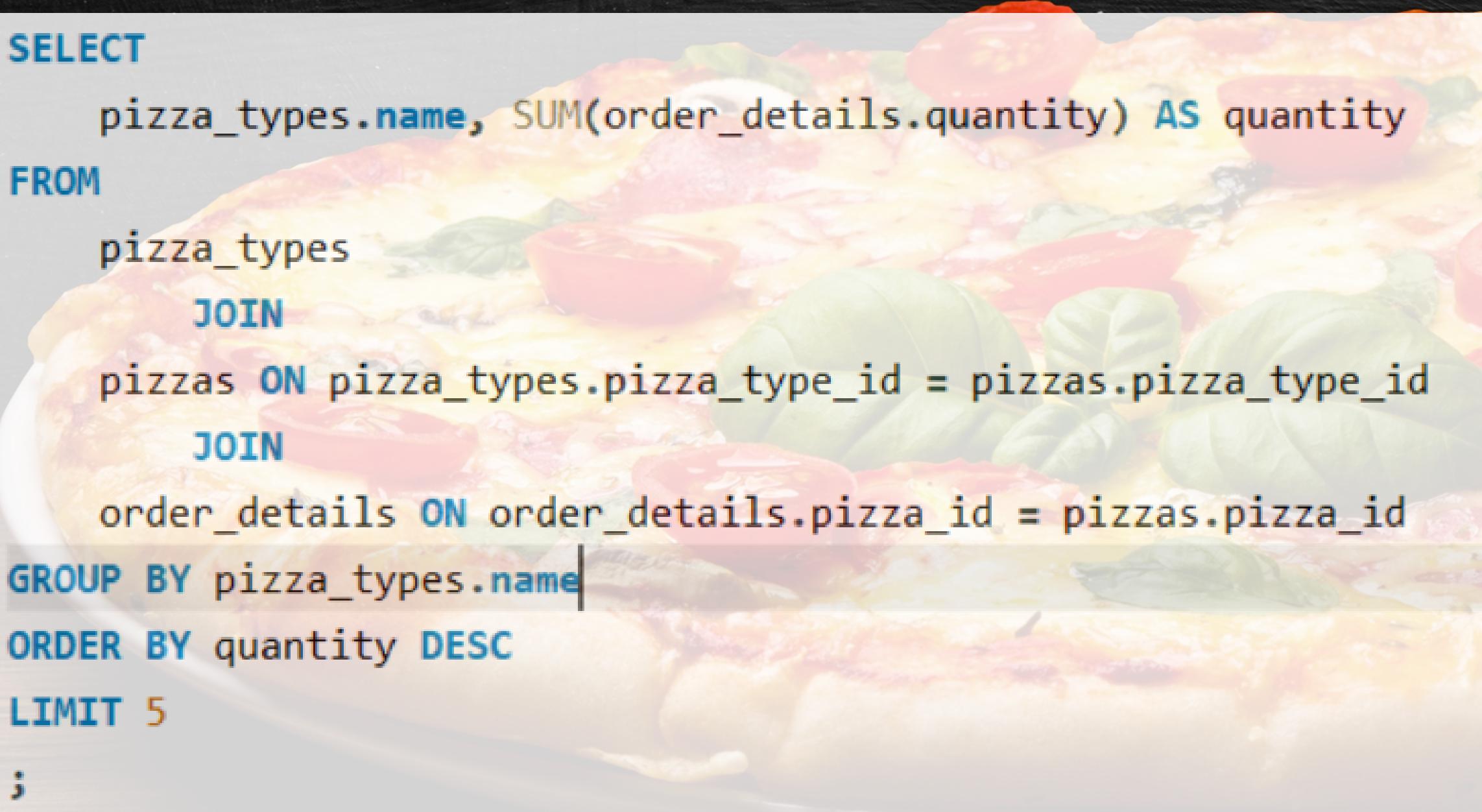
```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

Result Grid | Filter Row

	category	quantity
▶	Classic	12817
	Supreme	10301
	Veggie	10007
	Chicken	9479

-- 5. 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;
```



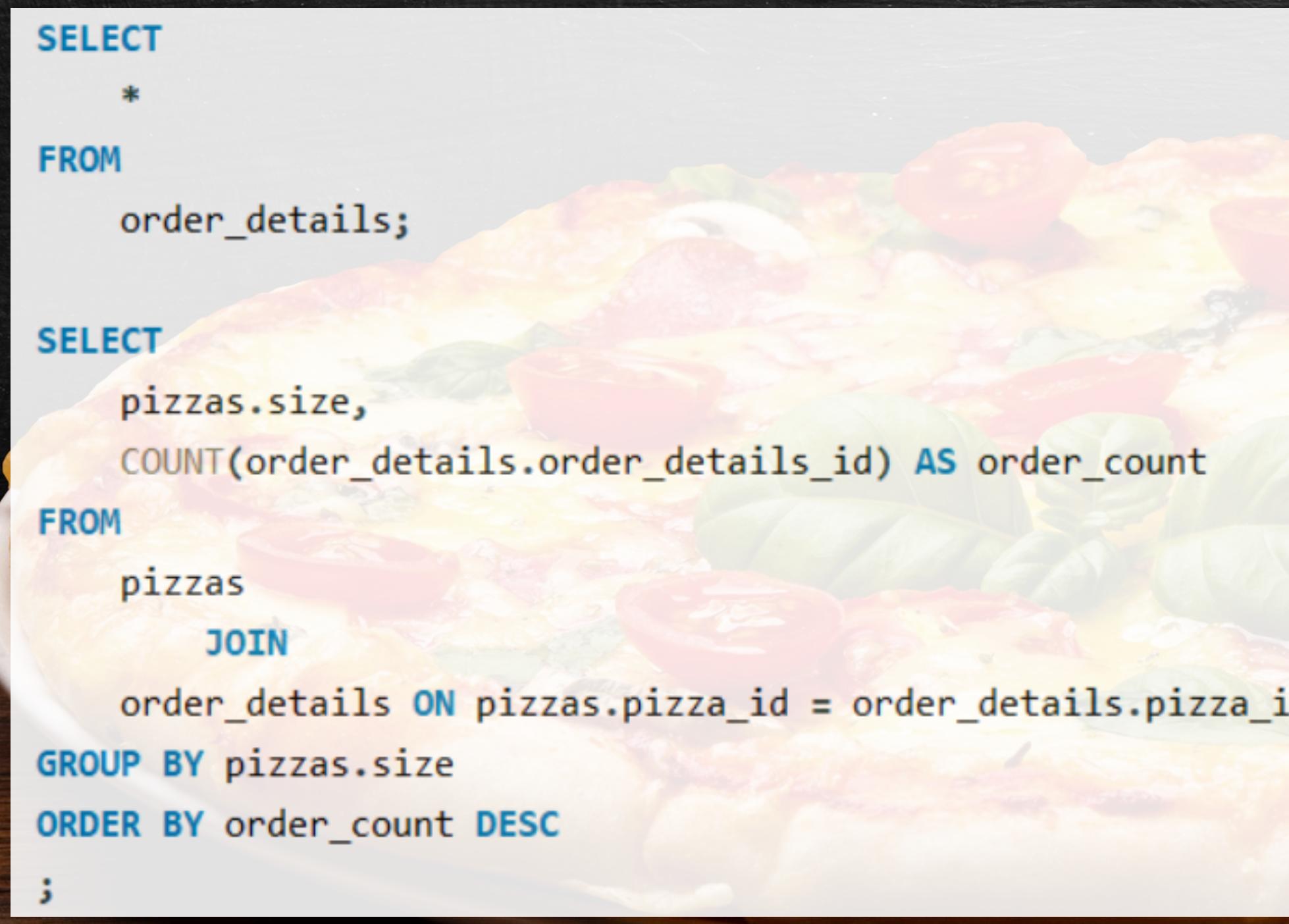
A screenshot of a MySQL Workbench interface showing the results of the SQL query. The results are displayed in a grid with two columns: 'name' and 'quantity'. The data shows the top 5 most ordered pizza types:

	name	quantity
▶	The Pepperoni Pizza	2108
	The Barbecue Chicken Pizza	2093
	The Classic Deluxe Pizza	2092
	The Hawaiian Pizza	2044
	The California Chicken Pizza	2026

-- 4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
*
FROM
    order_details;

SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
;
```



A large pizza with various toppings like pepperoni, cheese, and vegetables.

	size	order_count
	L	15936
	M	13242
	S	12109
	XL	479
	XXL	25

3-- IDENTIFY THE HIGHEST-PRICED PIZZA.

SELECT

 pizza_types.name, pizzas.price

FROM

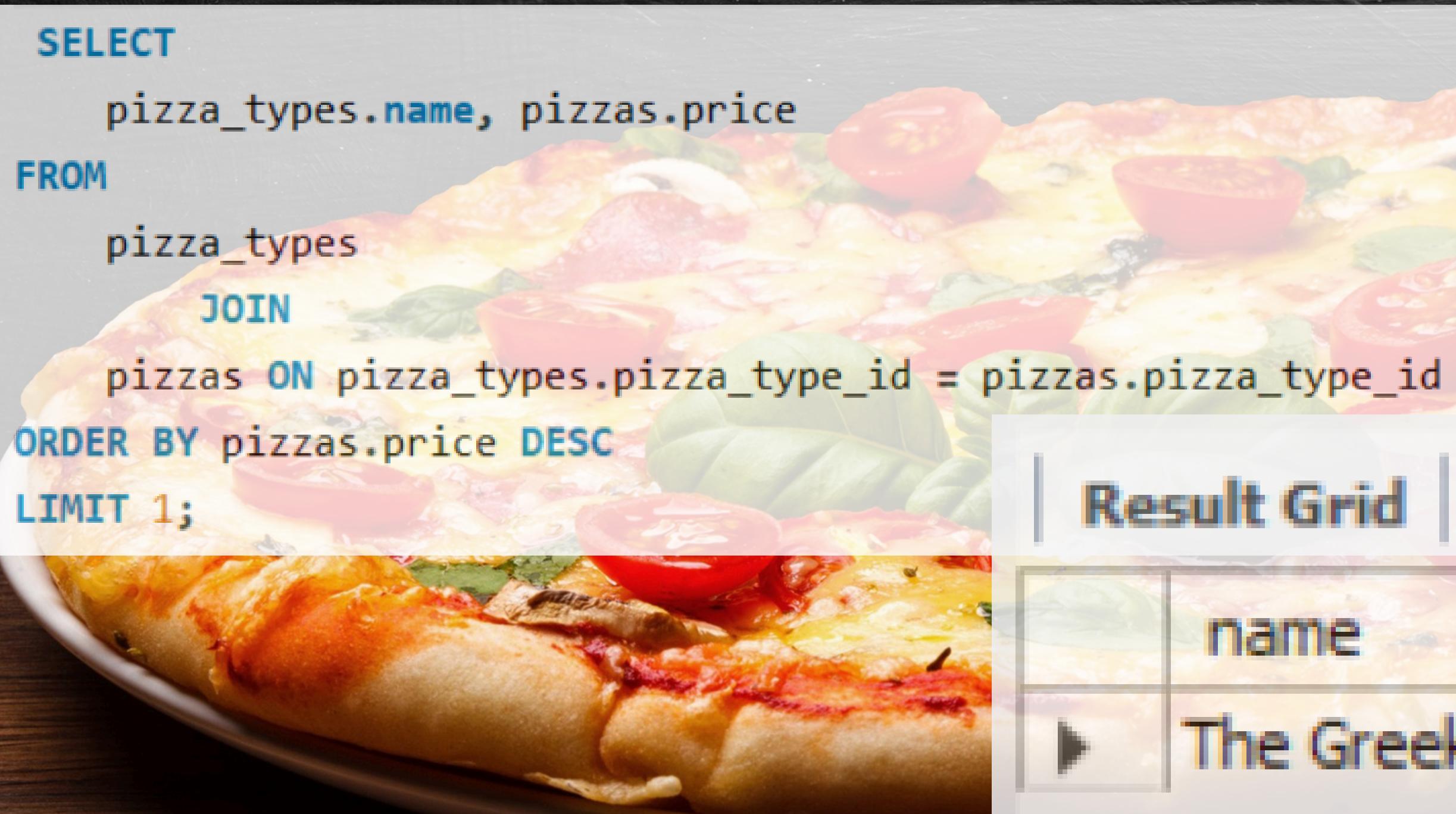
 pizza_types

JOIN

 pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id

ORDER BY pizzas.price DESC

LIMIT 1;



The image shows a data visualization interface for the query results. At the top right, there are two buttons: "Result Grid" with a grid icon and "Filter Rows:" with a filter icon. Below these are two rows of data in a table:

	name	price
▶	The Greek Pizza	35.95



LARANA PIZZA

THANK YOU!



