



"ANALYZED PIZZA SALES DATA USING SQL,  
FOCUSING ON ORDERS, REVENUE, TRENDS, AND  
PERFORMANCE TO PROVIDE ACTIONABLE  
BUSINESS INSIGHTS."



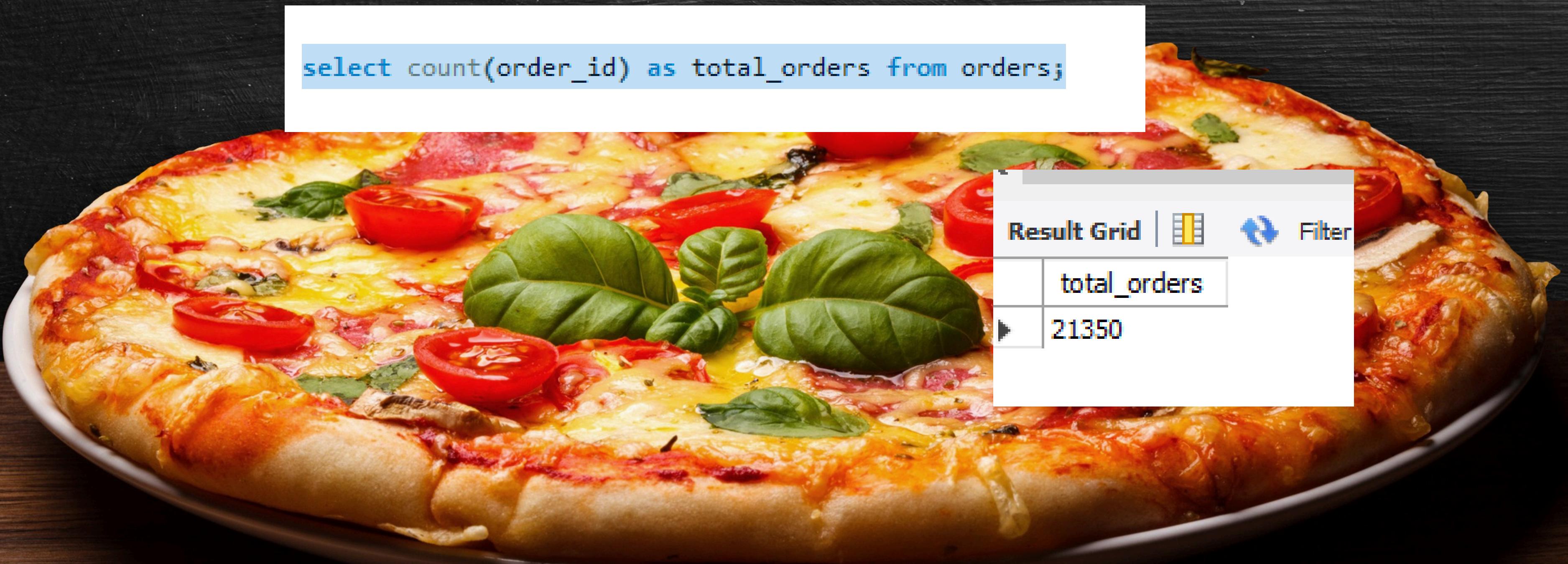
# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

Result Grid | 

 Filter

total_orders
21350





LARANA PIZZA

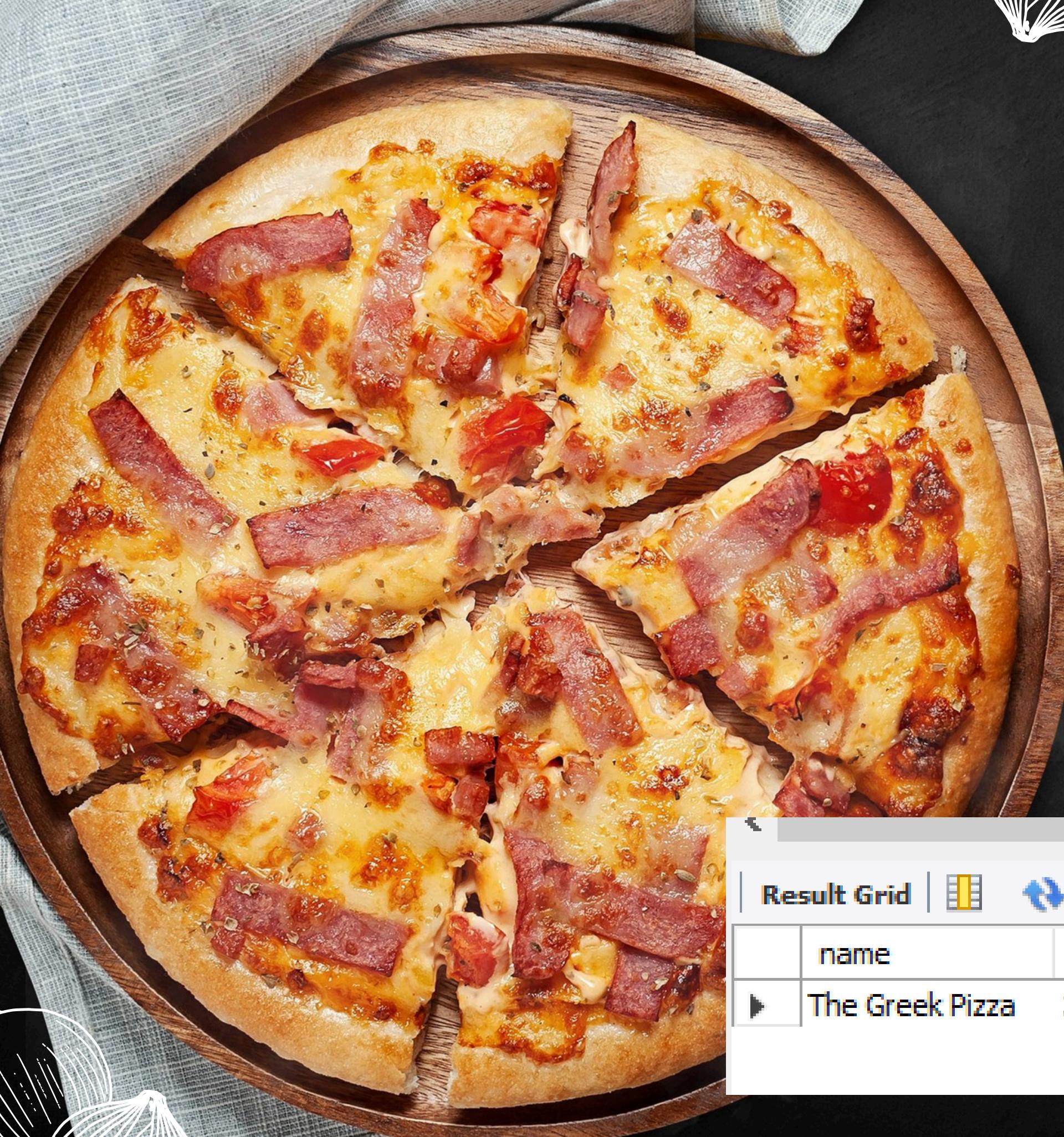
## CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT  
    ROUND(SUM(orders_details.quantity * pizzas.price),  
        2) AS total_sales  
  
FROM  
    orders_details  
    JOIN  
    pizzas ON pizzas.pizza_id = orders_Details.pizza_id;
```

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Result Grid	
	total_sales
▶	817860.05





# IDENTIFY THE HIGHEST- PRICED PIZZA.



LARANA PIZZA

```
select max(price) as expensive_one from pizzas;
```

```
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;
```

	Result Grid			Filter
	name	price		
▶	The Greek Pizza	35.95		

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

- `select max(price) as expensive_one from pizzas;`
- `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;`

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

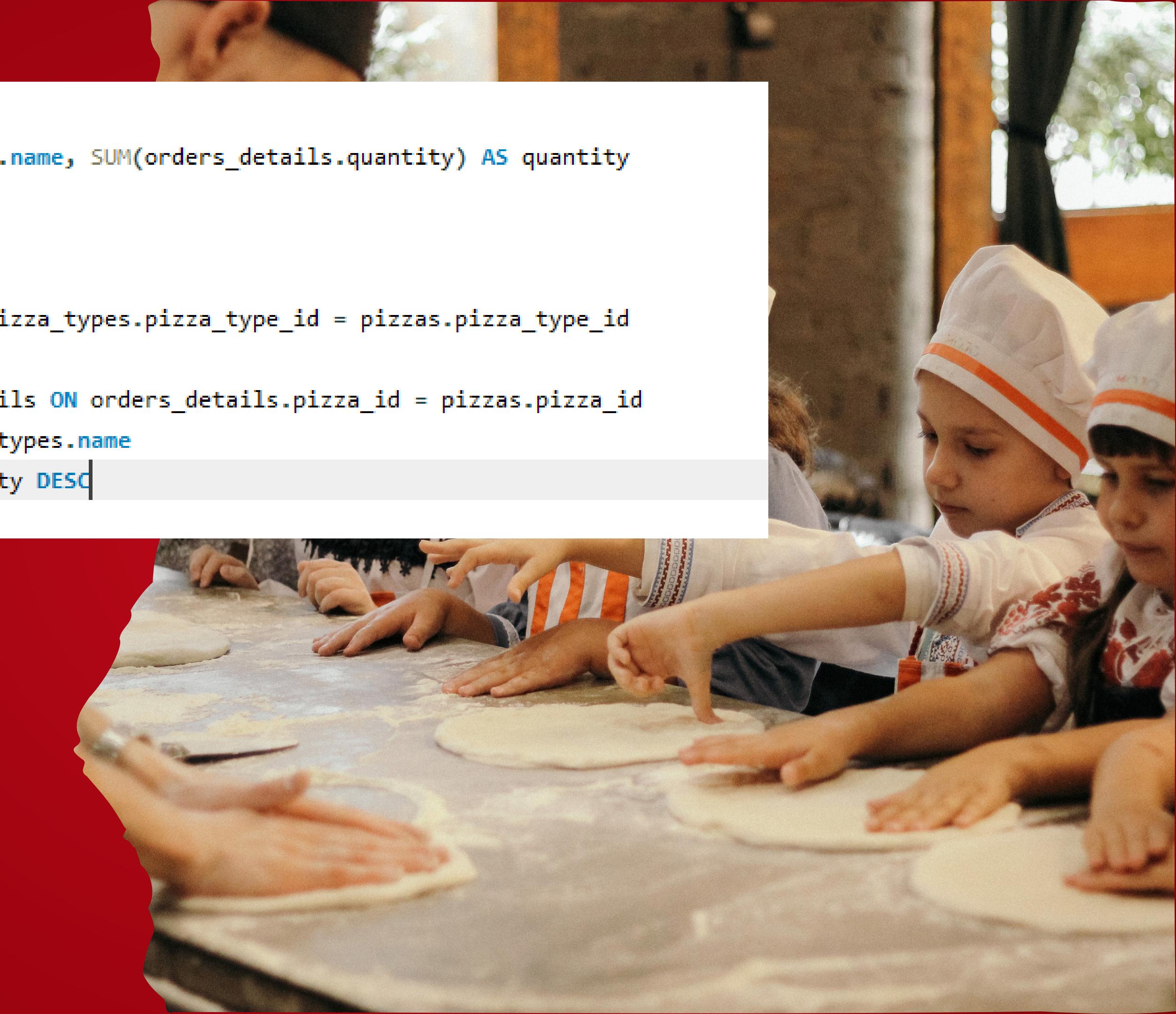
# LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
SELECT pizza_types.name, SUM(orders_details.quantity) AS quantity
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.name
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

	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  
    pizza_types.category,  
    SUM(orders_details.quantity) AS quantity  
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 quantity DESC;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

# DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
select hour(order_time) as hour, count(order_id) as total  
from orders  
group by hour
```

Result Grid | Filter Row

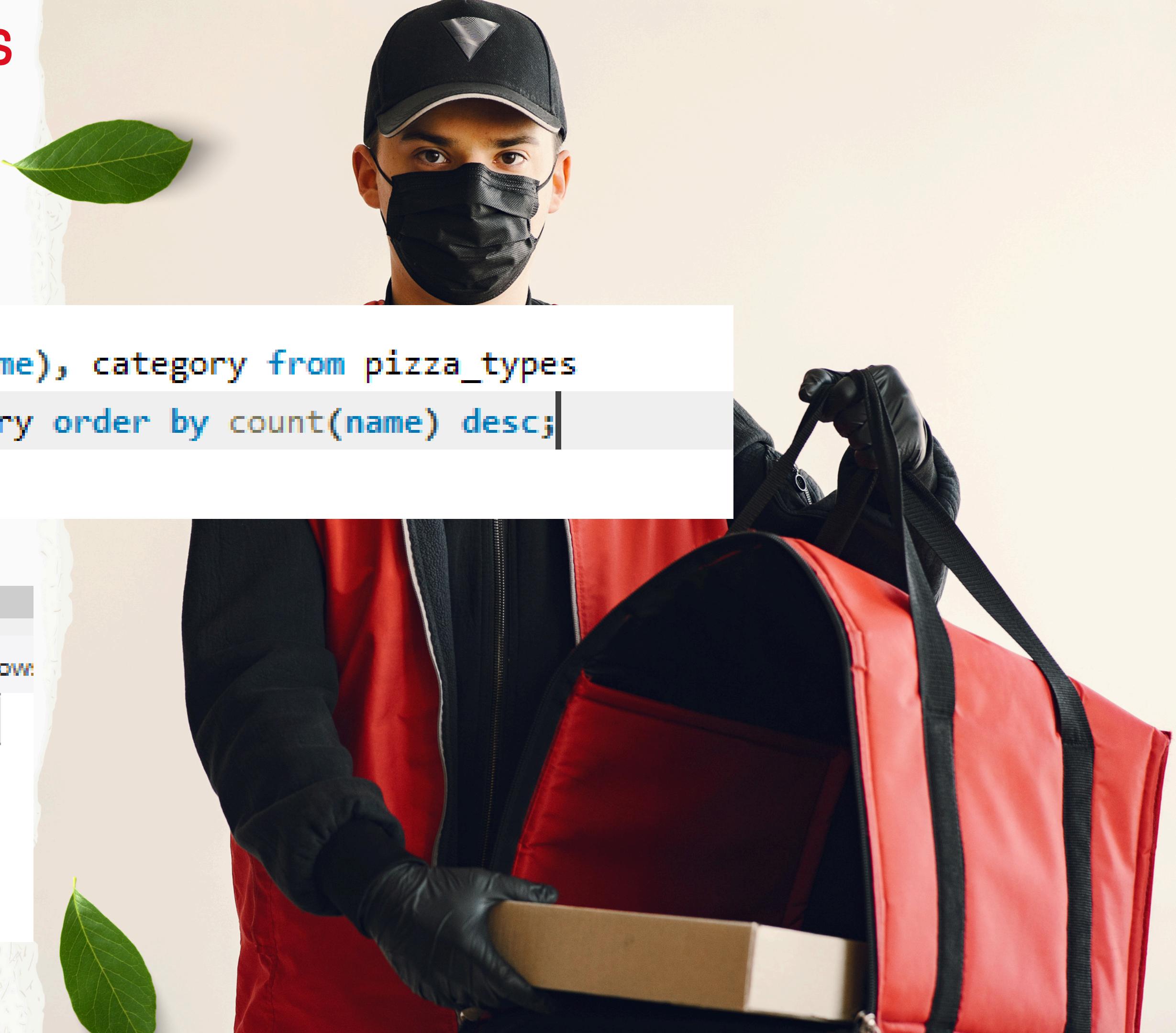
	hour	total
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468



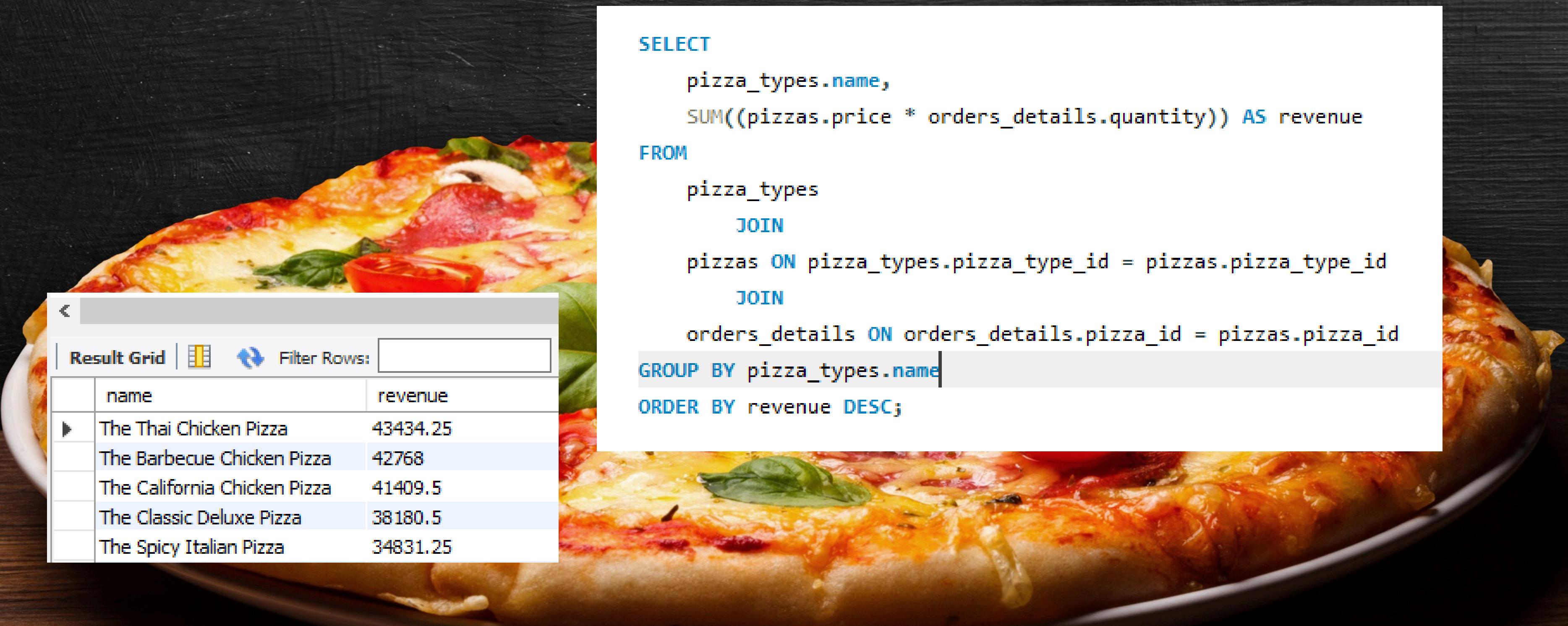
# JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

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

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



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



A large pizza is shown on a white plate, occupying the bottom half of the frame. The pizza has a golden-brown crust and is topped with melted cheese, sliced pepperoni, and several fresh basil leaves. The background is a dark, textured surface.

**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 Spicy Italian Pizza	34831.25

```
SELECT
    pizza_types.name,
    SUM((pizzas.price * orders_details.quantity)) 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.name
ORDER BY revenue DESC;
```

GROUP  
THE  
ORDERS  
BY DATE  
AND  
CALCULA  
TE THE  
AVERAGE  
NUMBER  
OF  
PIZZAS  
ORDERED  
PER DAY.

SELECT

```
ROUND(AVG(quantity), 0) average_pizza_order_per_day  
FROM  
(SELECT  
    orders.order_date, SUM(orders_details.quantity) AS quantity  
FROM  
    orders  
JOIN orders_details ON orders.order_id = orders_details.order_id  
GROUP BY orders.order_date) AS order_quantity;
```

Result Grid	
	average_pizza_order_per_day
▶	138

# 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(orders_details.quantity * pizzas.price),  
    2) AS total_sales  
FROM  
    orders_details  
    JOIN  
    pizzas ON pizzas.pizza_id = orders_Details.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.

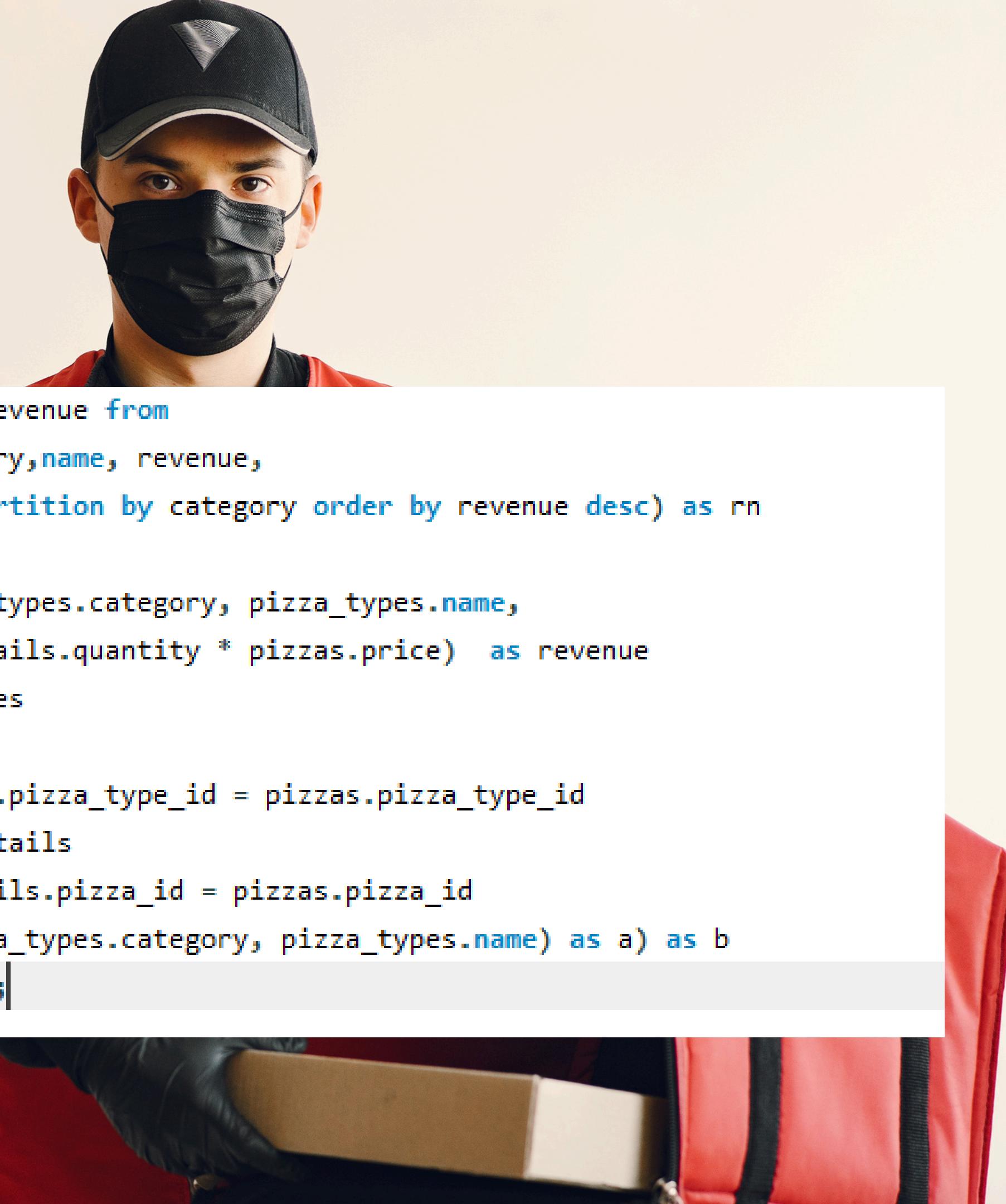
```
select order_date, sum(revenue) over (order by order_date) as cum_revenue
from
  (select orders.order_date, sum(orders_details.quantity * pizzas.price) as revenue
  from orders_details
  join pizzas
  on orders_details.pizza_id = pizzas.pizza_id
  join orders
  on orders.order_id = orders_details.order_id
  group by orders.order_date) as sales ;
```

Result Grid | Filter Rows:

	order_date	cum_revenue
	2015-05-03	280891.20000000007
	2015-05-04	283180.1000000001
	2015-05-05	284893.70000000007
	2015-05-06	287203.50000000006
	2015-05-07	289432.35000000003

Result 2 ×

# 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:  | Export

	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

Result 3 ×