

Pizza Sales Data Analysis PROJECT ON MySQL

order now



Welcome to the Pizza Sales Data Analysis project led by Ashish. This project aims to delve into the extensive dataset of a pizza restaurant chain to extract valuable insights using SQL. By leveraging SQL queries and analytical techniques, Ashish and the team will uncover patterns, trends, and metrics that can enhance business decision-making and operational efficiency.

Project Overview:

This SQL project revolves around a database schema designed to manage and analyze data for a pizza store. The database consists of four primary tables: `order_details`, `pizzas`, `orders`, and `pizza_types`. Each table plays a crucial role in storing different facets of the business operations, from individual orders to the types of pizzas offered. Below is a detailed description of each table and its columns:

1. `order_details`:

- `order_details_id`: A unique identifier for each entry in the `order_details`.
- `order_id`: References the ID from the `orders` table, linking the order detail to a specific order.
- `pizza_id`: References the ID from the `pizzas` table, identifying which pizza was ordered.
- `quantity`: The number of pizzas ordered of the specified type.

2. `pizzas`:

- `pizza_id`: A unique identifier for each type of pizza available.
- `pizza_type_id`: Links to the `pizza_types` table, specifying the type of pizza.
- `size`: The size of the pizza (e.g., small, medium, large).
- `price`: The cost of the pizza.

3. `orders`:

- `order_id`: A unique identifier for each order placed.
- `date`: The date on which the order was placed.
- `time`: The time at which the order was placed.

4. `pizza_types`:

- `pizza_type_id`: A unique identifier for each type of pizza.
- `name`: The name of the pizza type (e.g., Margherita, Pepperoni).
- `category`: Categorizes the pizza (e.g., Vegetarian, Non-Vegetarian).
- `ingredients`: Lists the ingredients used in the pizza.

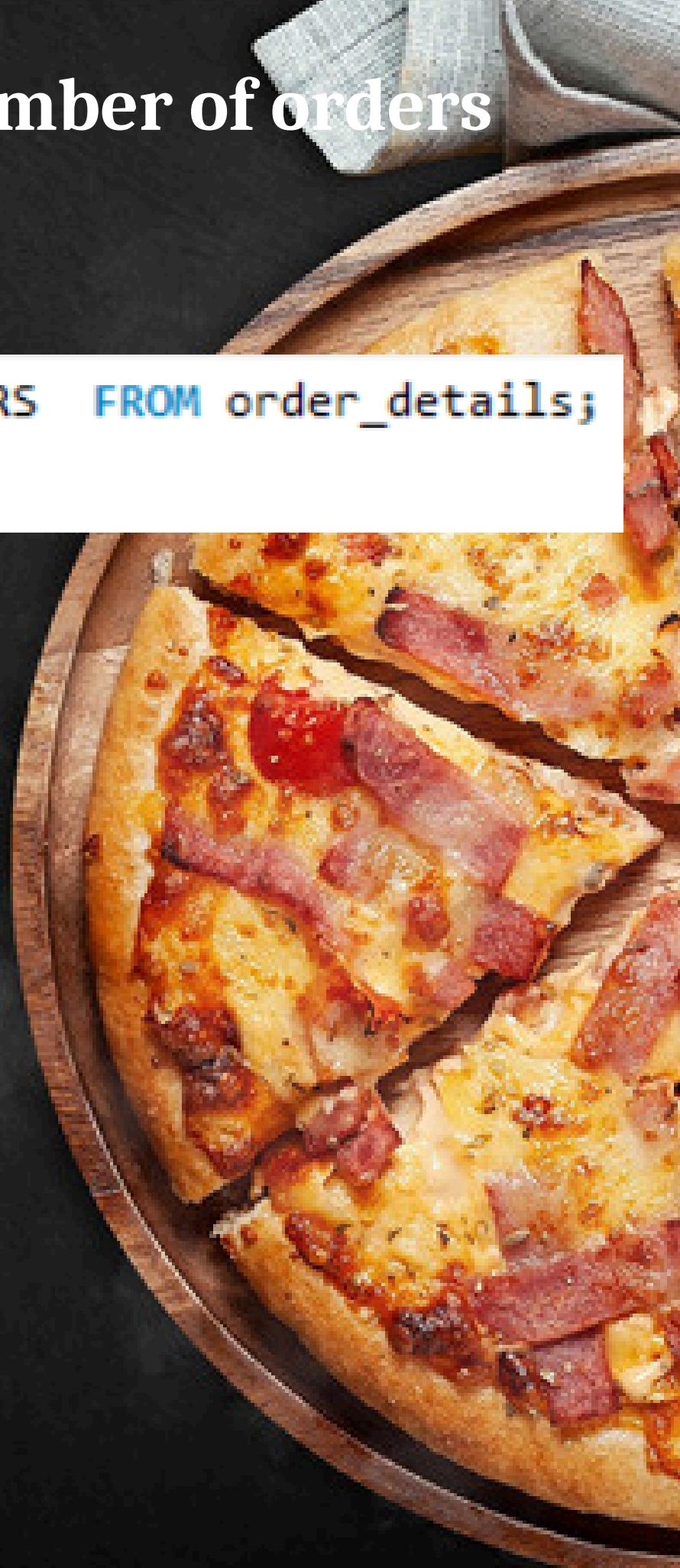


Q1 Retrieve the total number of orders placed.

```
SELECT COUNT(*) AS TOTAL_NO_ORDERS FROM order_details;
```

	TOTAL_NO_ORDERS
▶	48620

48620

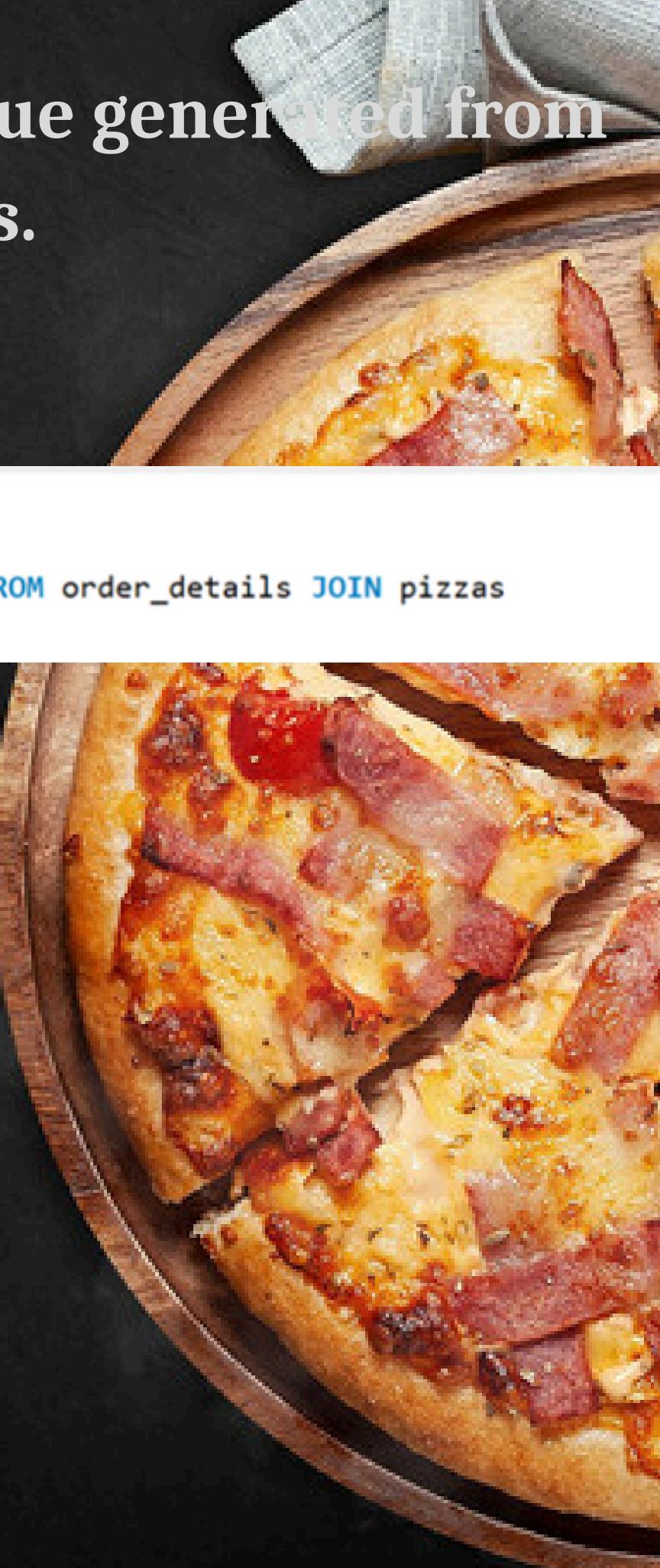


Q2 Calculate the total revenue generated from pizza sales.

```
SELECT * FROM ORDER_DETAILS;  
SELECT * FROM pizzas;  
SELECT round(sum(PRICE*QUANTITY),2) as total_revenue FROM order_details JOIN pizzas  
on order_details.pizza_id=pizzas.pizza_id;
```

total_revenue

817860.05



Q3 Identify the highest-priced pizza.

```
select * from pizza_types join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id  
order by price desc limit 1;
```

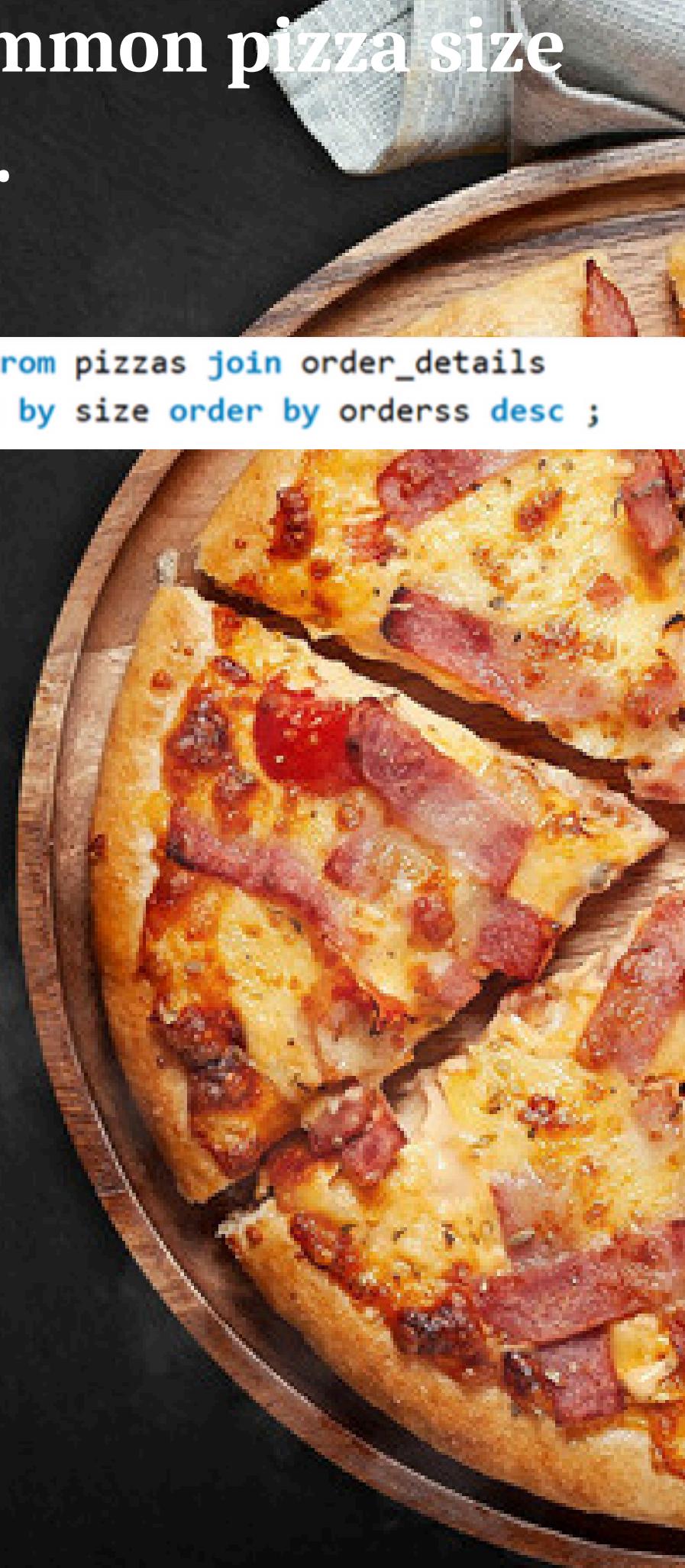
pizza_type_id	name	category	ingredients	pizza_id	pizza_type_id	size	price
the_greek	The Greek Pizza	Classic	Kalamata Olives, Feta Cheese, Tomatoes, Garli...	the_greek_xxL	the_greek	XXL	35.95



Q4 Identify the most common pizza size ordered.

```
select size ,count(order_details_id) as orderss from pizzas join order_details  
on pizzas.pizza_id =order_details.pizza_id group by size order by orderss desc ;
```

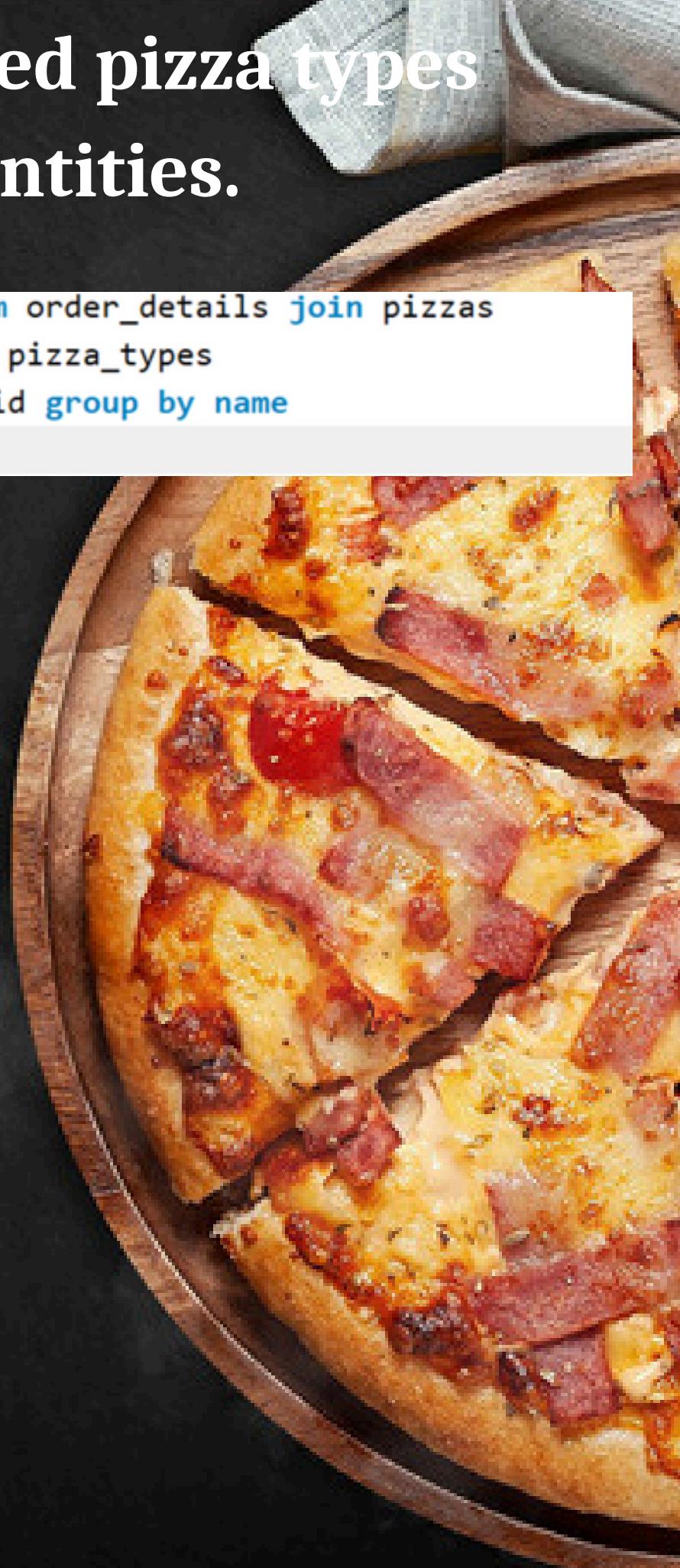
size	orderss
L	18526
M	15385
S	14137
XL	544
XXL	28



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

```
select name ,sum(quantity) as total_quantity from order_details join pizzas  
on order_details.pizza_id =pizzas.pizza_id join pizza_types  
on pizza_types.pizza_type_id=pizzas.pizza_type_id group by name  
order by total_quantity desc limit 5;
```

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



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

```
select category ,sum(quantity) as total_quantity from order_details join pizzas  
on order_details.pizza_id=pizzas.pizza_id join pizza_types  
on pizzas.pizza_type_id=pizza_types.pizza_type_id group by category ;
```

category	total_quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050



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

```
select * from orders;  
select hour(order_time) ,count(order_id ) from orders group by hour(order_time);
```

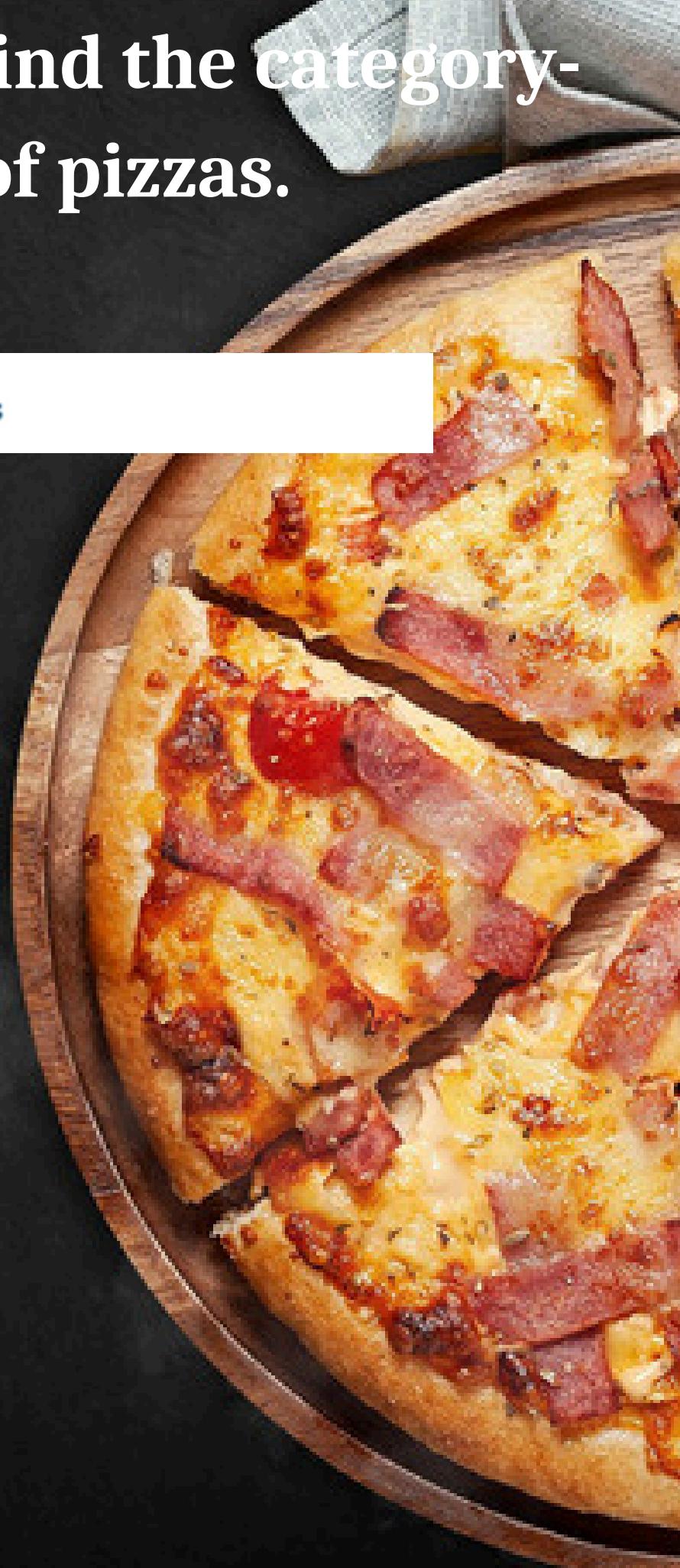
hour(order_time)	count(order_id)
11	1231
12	2520
13	2455
14	1472



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

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

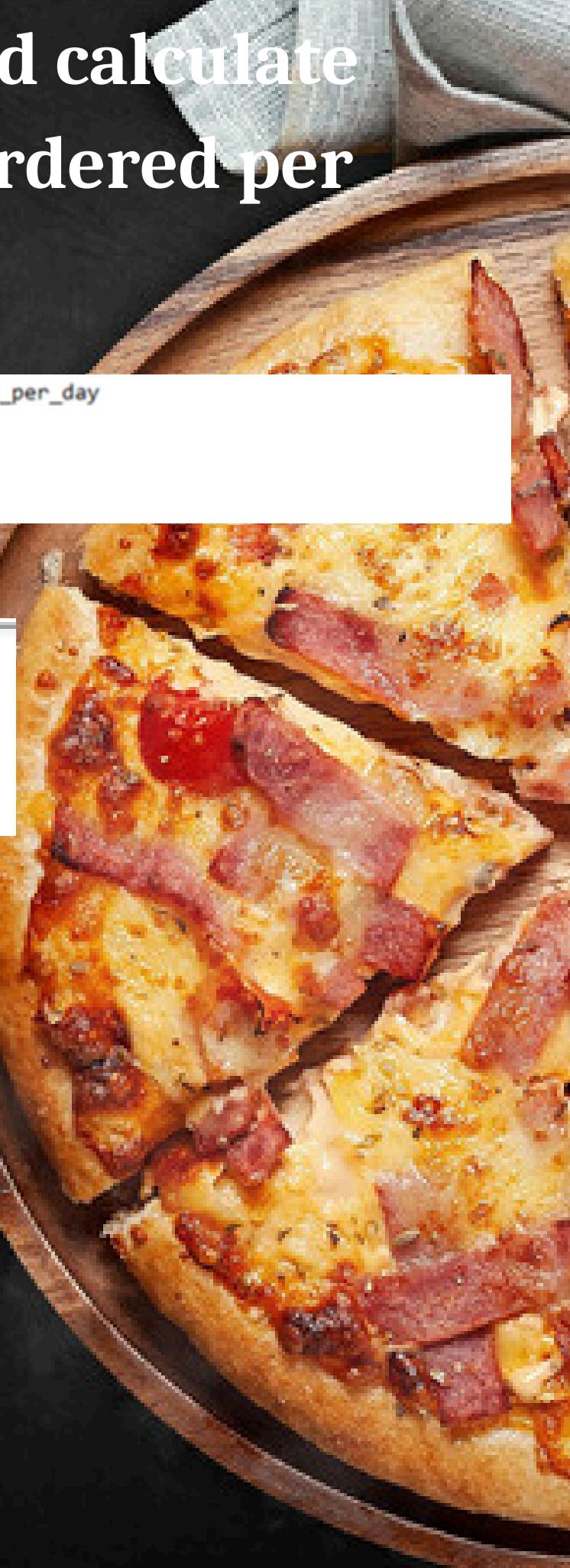
	category	count(name)
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9



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

```
select round(avg(order_per_day),0)from (select order_date ,sum(quantity) as order_per_day  
from orders join order_details on order_details.order_id=orders.order_id  
group by order_date) as ordersss;
```

round(avg(order_per_day),0)
138



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

```
select name ,sum(quantity*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 name order by revenue desc limit 5;
```

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



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

```
select category,round((sum(quantity*price)/817860.05*100),2) as revenue  
from order_details join pizzas on order_details.pizza_id =pizzas.pizza_id join  
pizza_types on pizza_types.pizza_type_id = pizzas.pizza_type_id group by category ;
```

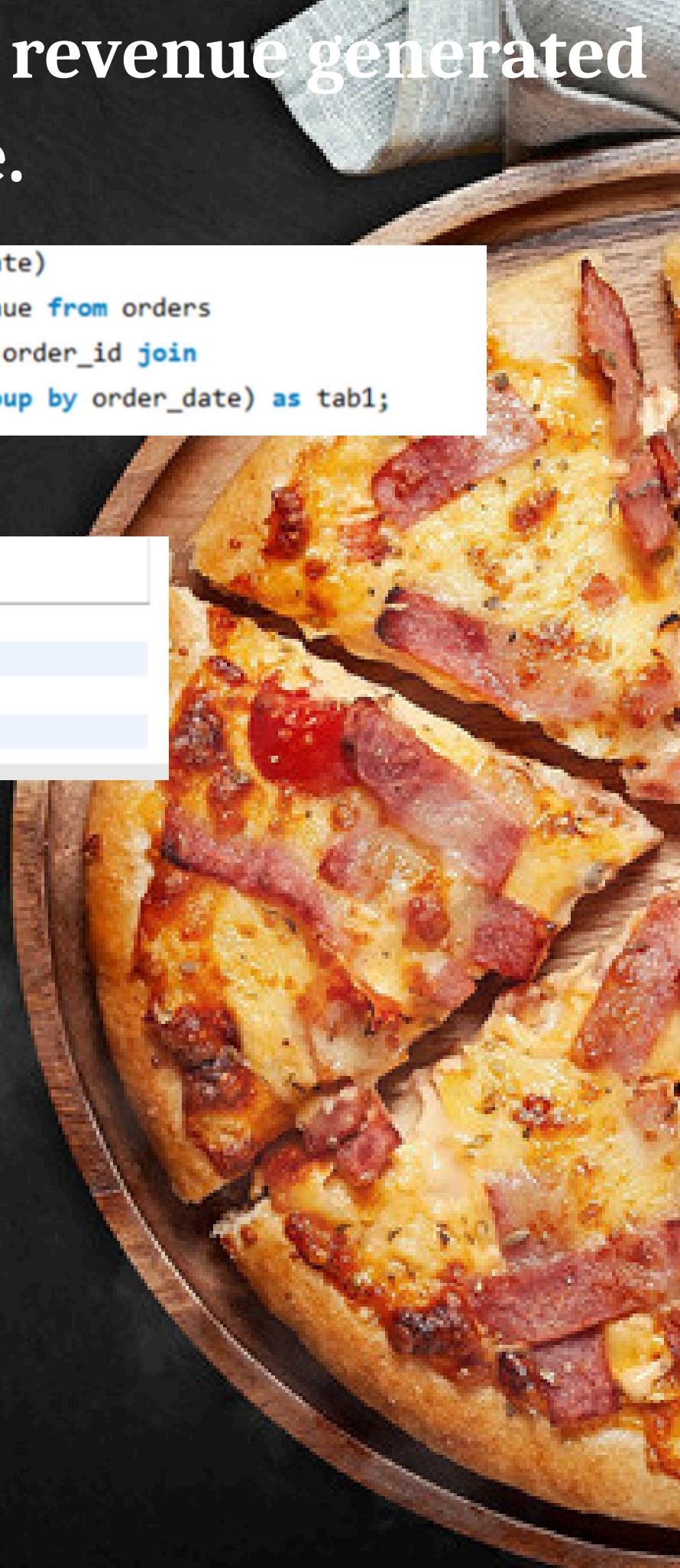
category	revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96



Q12 Analyze the cumulative revenue generated over time.

```
select order_date ,sum(revenue) over (order by order_date)
from (select order_date , sum(quantity*price)  as revenue from orders
      join order_details on orders.order_id = order_details.order_id join
      pizzas on pizzas.pizza_id = order_details.pizza_id group by order_date) as tab1;
```

order_date	sum(revenue) over (order by order_date)
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	-----



Q13 Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
select name,category,revenue ,rank( )over(partition by category order by revenue desc)
from (select name,category,sum(quantity*price)as revenue   from order_details join pizzas
on order_details.pizza_id =pizzas.pizza_id join
pizza_types on pizza_types.pizza_type_id = pizzas.pizza_type_id group by category,name) as a ;
```

name	category	revenue	rank()over(partition by category order by revenue desc)
The Thai Chicken Pizza	Chicken	43434.25	1
The Barbecue Chicken Pizza	Chicken	42768	2
The California Chicken Pizza	Chicken	41409.5	3
The Southwest Chicken Pizza	Chicken	34705.75	4
...	-

