



Pizza\_sales Dashboard with the help of Excel  
and SQL

**\$35**

Easy to change colors, photos.

**\$50**

Get a modern PowerPoint  
Presentation that is beautifully  
designed.

**\$65**

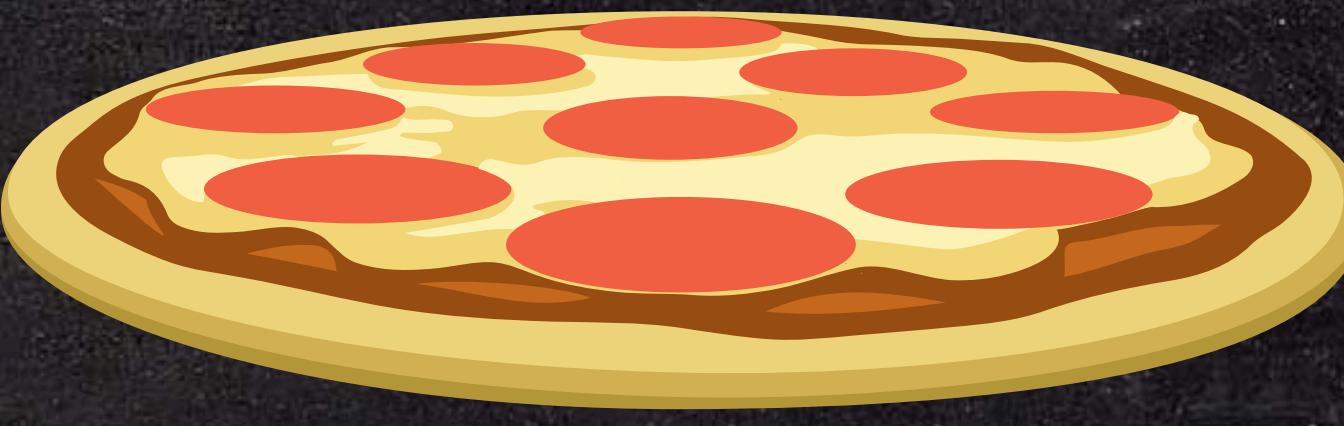
You can simply impress your  
audience and add a unique zing  
and appeal to your Presentations.



1 .Retrieve the total number of orders placed.

```
2
3 • select count(order_id) as total_orders from orders;
```

	total_orders
▶	21350



2. Calculate the total revenue generated from pizza sales.

```
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
```

Result Grid	
	total_sales
▶	817860.05

### 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;
```

Result Grid		Filter Rows:
	name	price
▶	The Greek Pizza	35.95



# 4. Identify the most common pizza size ordered.

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

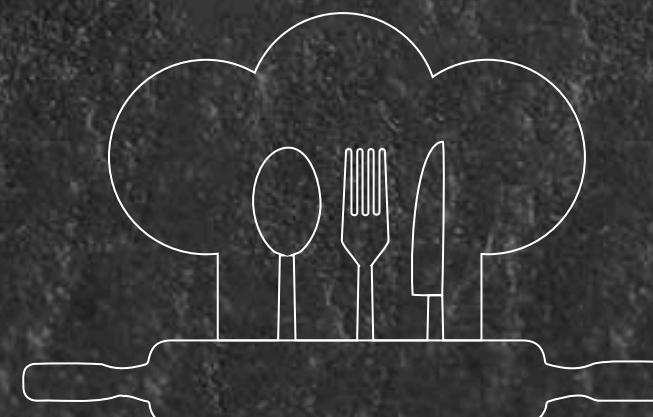
Result Grid | Filter Rows:

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

## 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
;
```

	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



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

```
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 Rows

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

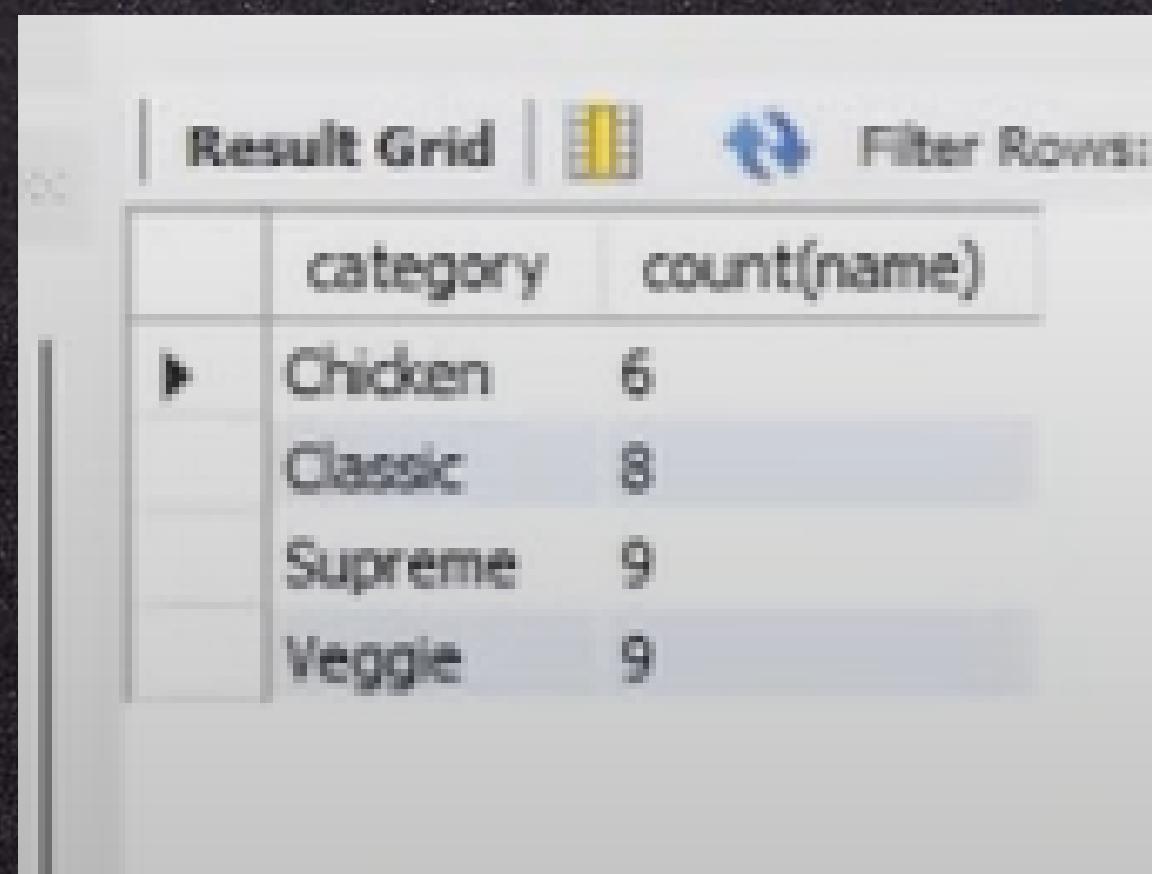
# 7. Determine the distribution of orders by hour of the day.

```
3 • SELECT  
4     HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
5 FROM  
6     orders  
7 GROUP BY HOUR(order_time);
```

	hour	order_count
11	1231	
12	2520	
13	2455	
14	1472	
15	1469	
16	1920	
17	2336	
18	2399	
19	2009	
20	1642	

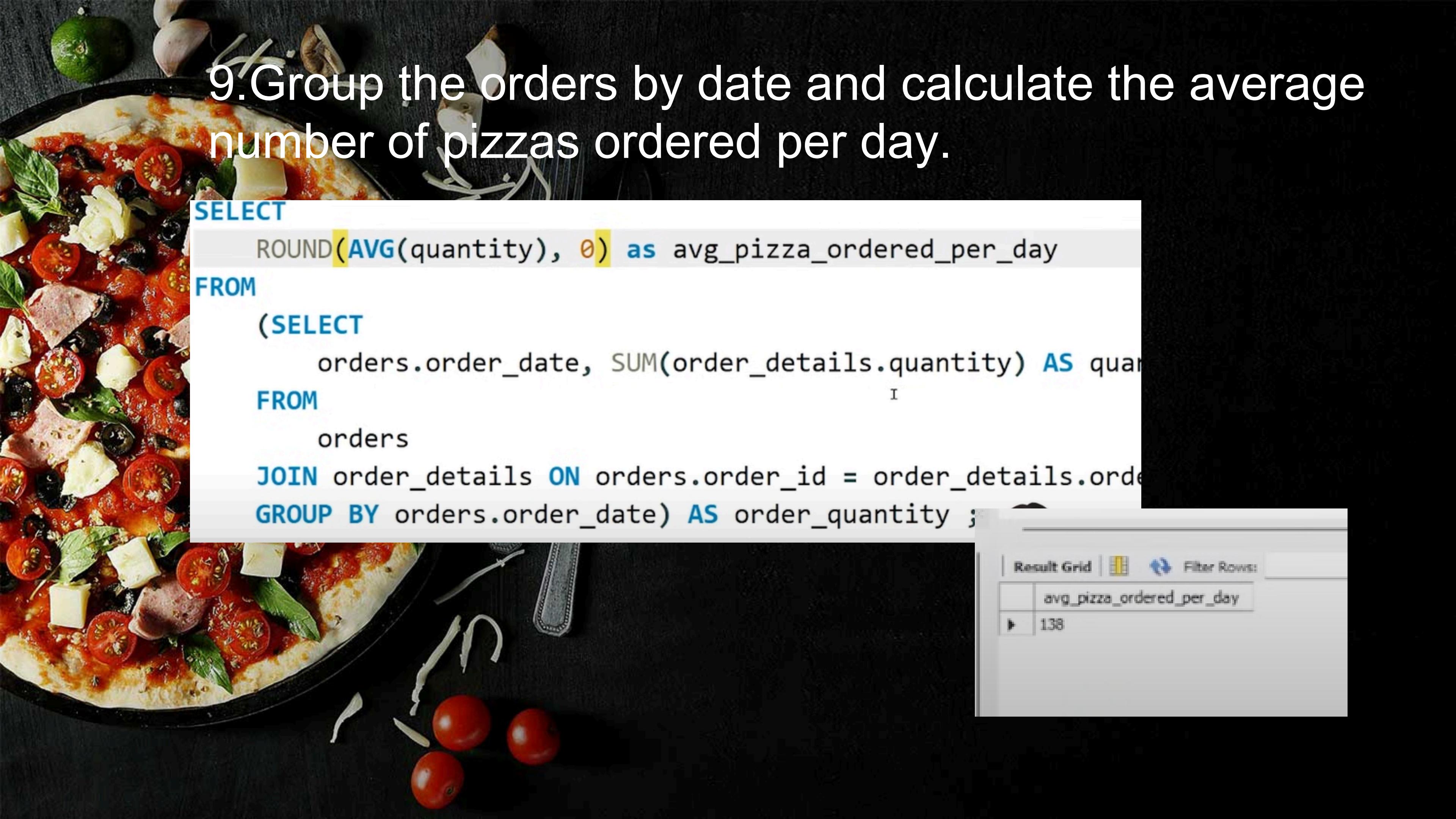
## 8. Join relevant tables to find the category-wise distribution of pizzas.

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



The screenshot shows a MySQL Workbench result grid. The title bar includes 'Result Grid' and 'Filter Rows'. The data is presented in a table with two columns: 'category' and 'count(name)'. The table contains four rows with the following data:

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



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

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

Result Grid	
	avg_pizza_ordered_per_day
▶	138

# 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:  Export: 

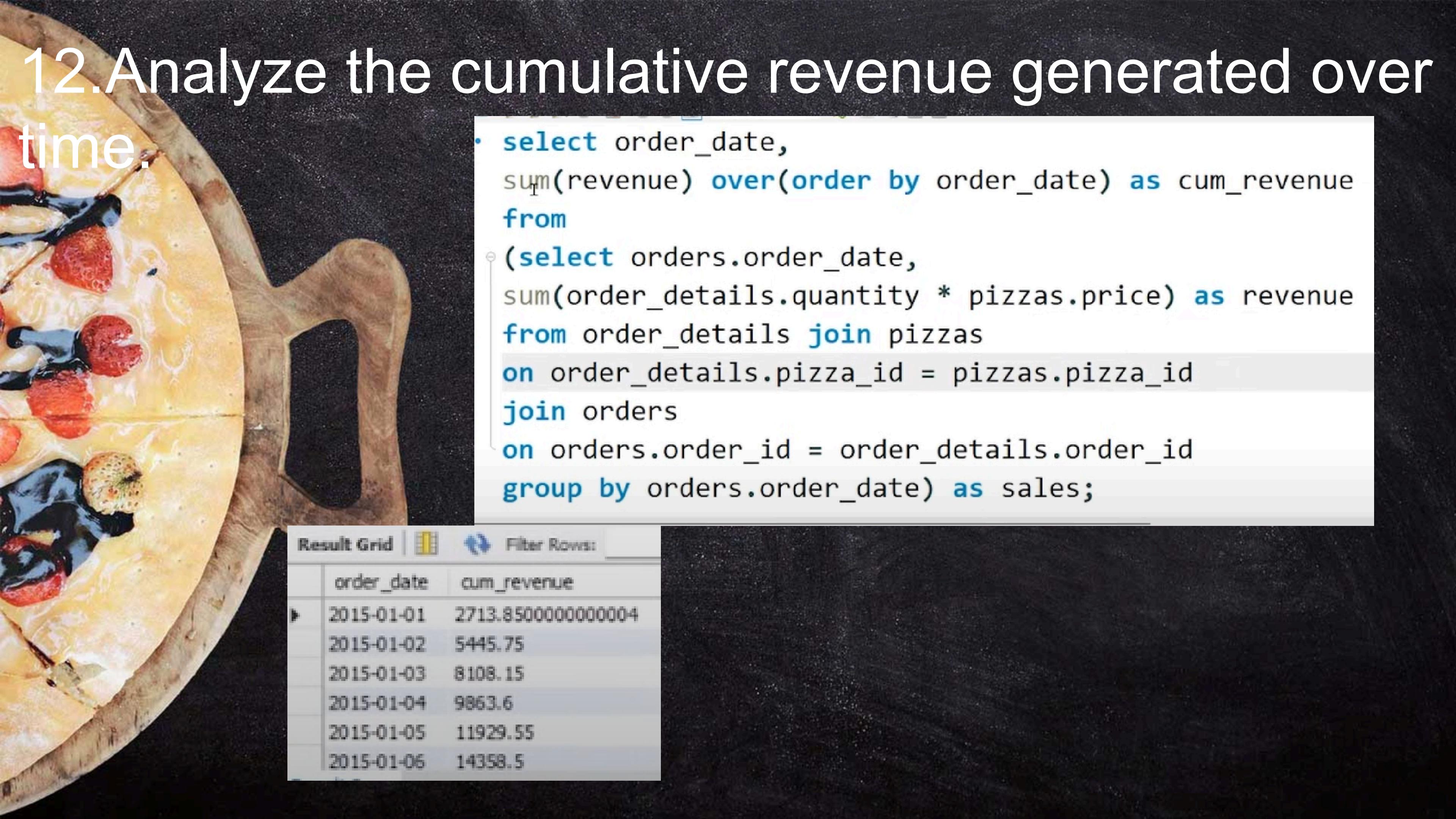
	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

# 11. Calculate the percentage contribution of each pizzatype 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 Row

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



## 12. 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(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.order_date) as sales;
```

Result Grid | Filter Rows:

	order_date	cum_revenue
▶	2015-01-01	2713.8500000000004
	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

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

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
The Pepperoni Pizza	30161.75

```
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((order_details.quantity) * pizzas.price) 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, pizza_types.name) as a) as b
where rn <= 3;
```



THANK YOU