

# SQL PROJECT

**Pizza Sales Analysis: Uncovering Insights**

Presented by Sivabalan V

# WELCOME

This SQL project delves into pizza sales data analysis, aiming to extract actionable insights from a comprehensive dataset. By leveraging SQL queries, we explore sales trends, popular pizza varieties. Through this analysis, we aim to empower pizza businesses with valuable intelligence to drive growth and enhance the customer experience in a competitive market.

# Retrieve the total number of orders placed.

QUERY:

```
1 -- Retrieve the total number of orders placed.  
2 • select count(order_id) as total_orders from orders;
```

OUTPU:

Result Grid	
#	total_order:
1	2982

# Calculate the total revenue generated from pizza sales

## QUERY:

```
1 -- Calculate the total revenue generated from pizza sales.  
2 • select  
3     round(sum(order_details.quantity * pizzas.price), 2) as total_sales  
4   from order_details join pizzas on  
5     pizzas.pizza_id = order_details.pizza_id;
```

## OUTPUT:

Result Grid	
#	total_sales
1	43998.65

# Identify the highest-priced pizza

## QUERY:

```
1  -- Identify the highest-priced pizza.  
2 • select pizza_types1.name, pizzas.price  
3   from pizza_types1 join pizzas  
4     on pizza_types1.pizza_type_id = pizzas.pizza_type_id  
5   order by pizzas.price desc limit 1;
```

## OUTPUT:

Result Grid Filter Rows:

#	name	price
1	The Greek Pizza	25.5

# Identify the most common pizza size ordered

QUERY:

```
1 -- Identify the most common pizza size ordered.  
2 • select pizzas.size, count(order_details.order_details_id) as order_count  
3   from pizzas join order_details  
4     on pizzas.pizza_id = order_details.pizza_id  
5   group by pizzas.size order by order_count desc;
```

OUTPUT:

#	size	order_coun
1	M	2299
2	S	2045
3	L	1809
4	big_meat_s	307
5	napolitana_s	241
6	bbq_ckn_l	241
7	green_garden_m	223

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

## QUERY:

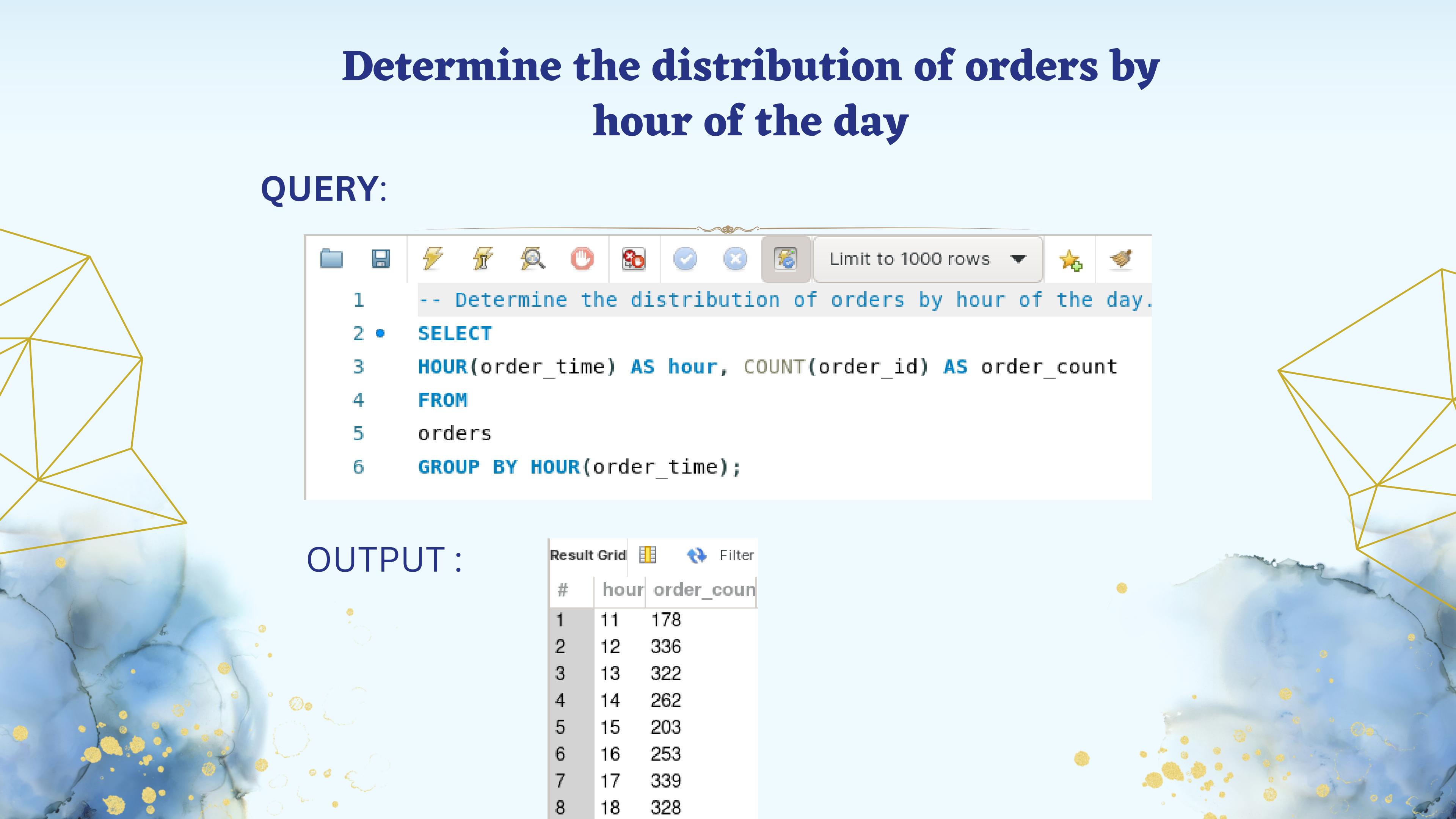
```
1 -- Join the necessary tables to find the
2 -- total quantity of each pizza category ordered.
3
4 • select pizza_types1.category,
5     sum(order_details.quantity) as quantity
6     from pizza_types1 join pizzas
7     on pizza_types1.pizza_type_id = pizzas.pizza_type_id
8     join order_details
9     on order_details.pizza_id = pizzas.pizza_id
10    group by pizza_types1.category order by quantity desc;
```

## OUTPUT :

#	category	quantity
1	Classic	1959
2	Supreme	1948
3	Veggie	1426
4	Chicken	1421

# Determine the distribution of orders by hour of the day

QUERY:



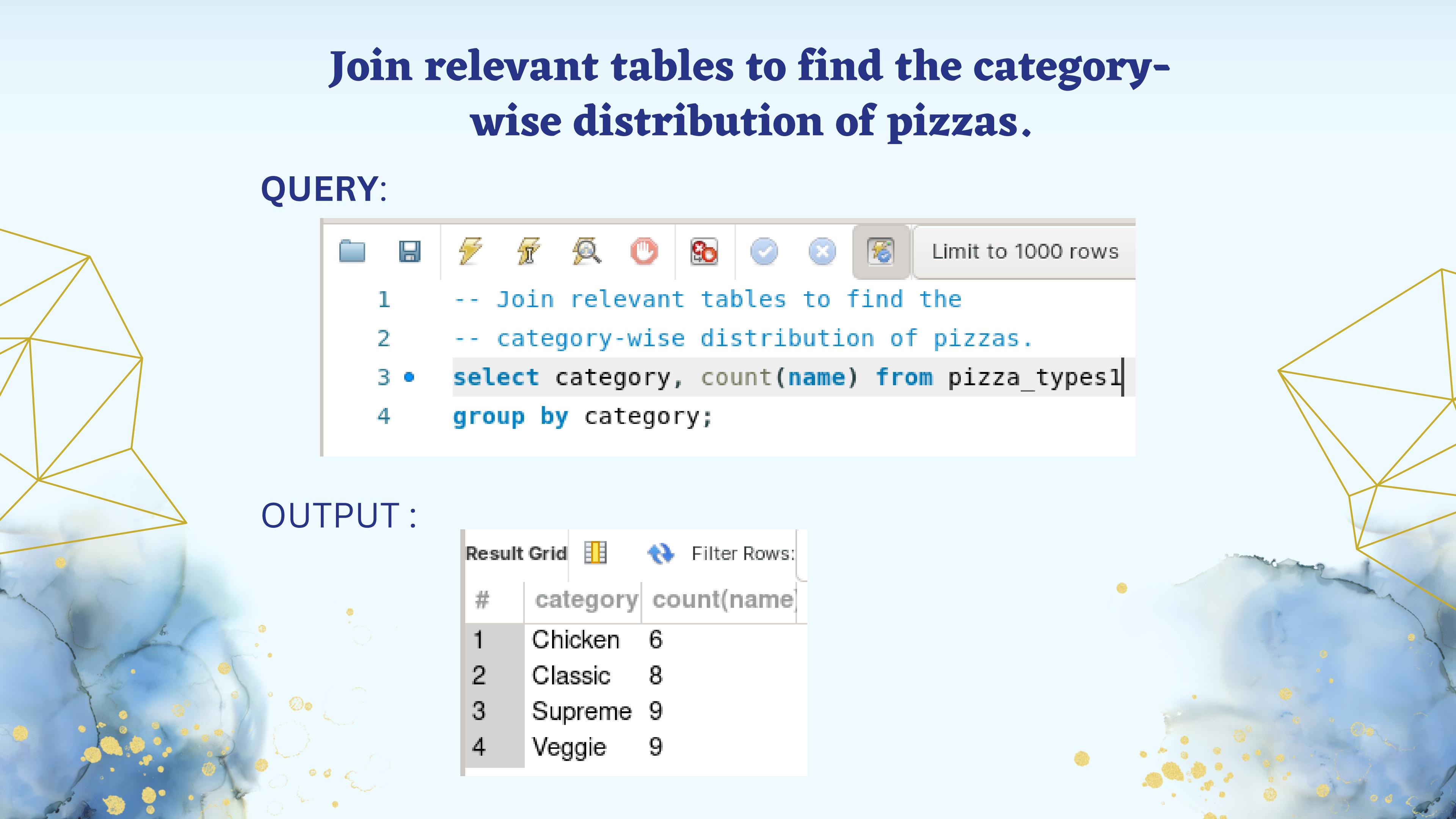
```
1 -- Determine the distribution of orders by hour of the day.  
2 • SELECT  
3     HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
4 FROM  
5 orders  
6 GROUP BY HOUR(order_time);
```

OUTPUT :

#	hour	order_count
1	11	178
2	12	336
3	13	322
4	14	262
5	15	203
6	16	253
7	17	339
8	18	328

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

## QUERY:



```
1 -- Join relevant tables to find the
2 -- category-wise distribution of pizzas.
3 • select category, count(name) from pizza_types1
4 group by category;
```

## OUTPUT:

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

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

QUERY:

```
1 -- Group the orders by date and calculate the average
2 -- number of pizzas ordered per day.
3 • SELECT
4 ROUND (AVG(quantity), 0) as avg_pizza_ordered_per_day
5 FROM
6 (SELECT
7 orders.order_date, SUM(order_details.quantity) AS quantity
8 FROM
9 orders
10 JOIN order_details ON orders.order_id = order_details.order_id
11 GROUP BY orders.order_date) AS order_quantity ;
```

OUTPUT :

Result Grid		Filter Rows:
#	avg_pizza_ordered_per_da	
1	68	

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

QUERY:

```
1 -- Determine the top 3 most ordered pizza types based on revenue.  
2 • select pizza_types1.name,  
3     sum(order_details.quantity * pizzas.price) as revenue  
4     from pizza_types1 join pizzas  
5     on pizzas.pizza_type_id = pizza_types1.pizza_type_id  
6     join order_details  
7     on order_details.pizza_id  
8     =  
9     pizzas.pizza_id  
10    group by pizza_types1.name order by revenue desc limit 3;
```

OUTPUT:

#	name	revenue
1	The California Chicken Pizza	6007
2	The Spicy Italian Pizza	5590
3	The Italian Supreme Pizza	5516.5

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

QUERY:

```
1 -- pizza type to total revenue.
2 • select pizza_types1.category,
3   round(sum(order_details.quantity * pizzas.price) / (SELECT
4     ROUND(SUM(order_details.quantity * pizzas.price),2)as total_sales
5   FROM
6   order_details
7   JOIN
8   pizzas ON pizzas.pizza_id = order_details.pizza_id) *100,2) as revenue
9   from pizza_types1 join pizzas
10  on pizza_types1.pizza_type_id = pizzas.pizza_type_id
11  join order_details
12  on order_details.pizza_id = pizzas.pizza_id
13  group by pizza_types1.category order by revenue desc;
```

OUTPUT:

#	category	revenue
1	Supreme	30.22
2	Classic	28.08
3	Chicken	21.19
4	Veggie	16.66

# Analyze the cumulative revenue generated over time.

QUERY:

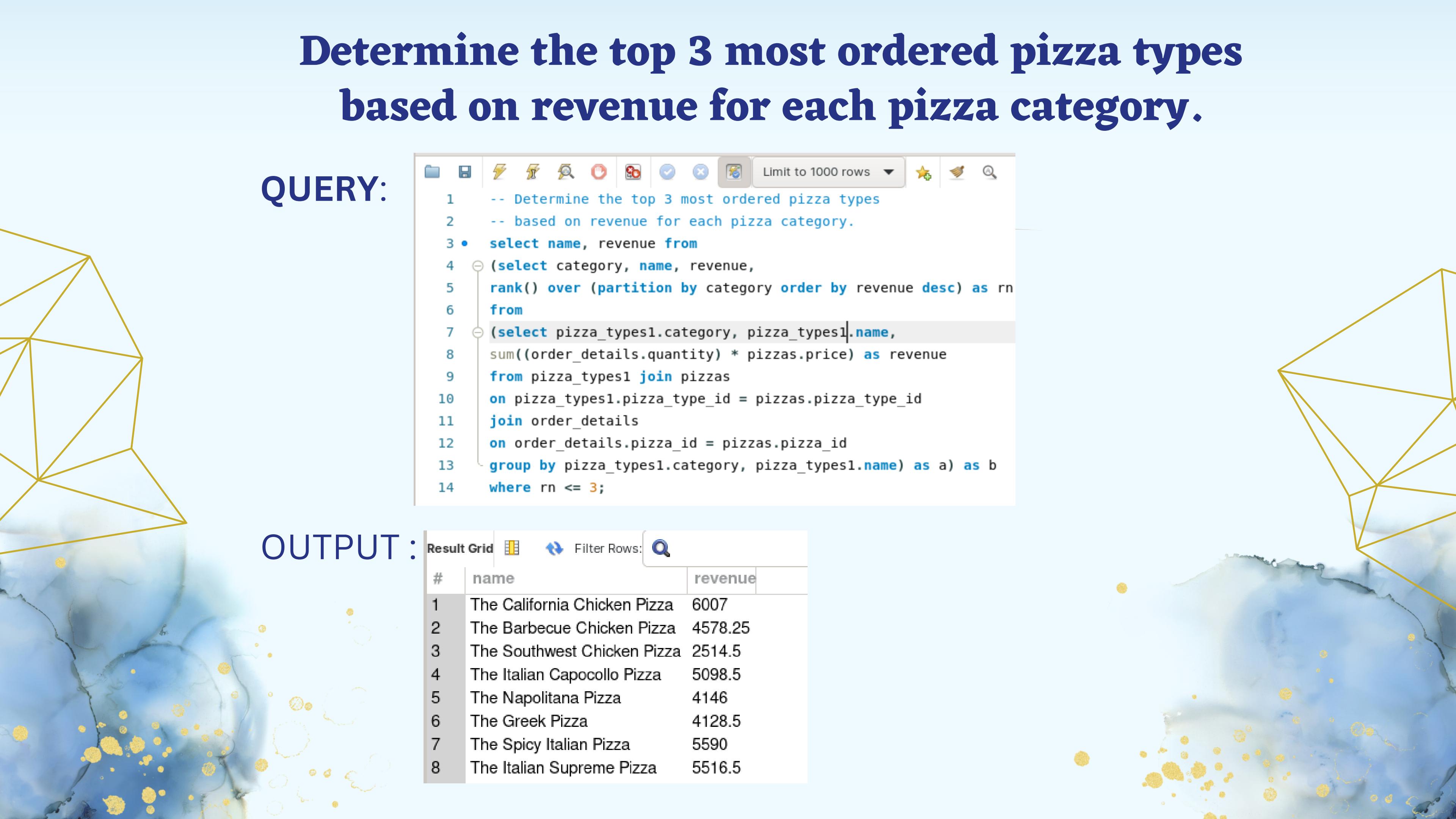
```
1 -- analyze the cumulative revenue generated over time
2 • select order_date,
3   sum(revenue) over(order by order_date) as cum_revenue
4   from
5   (select orders.order_date,
6    sum(order_details.quantity * pizzas.price) as revenue
7    from order_details join pizzas
8    on order_details.pizza_id = pizzas.pizza_id
9    join orders
10   on orders.order_id = order_details.order_id
11   group by orders.order_date) as sales;
```

OUTPUT:

Result Grid		
#	order_date	cum_revenue
1	2015-01-01	1210.5
2	2015-01-02	2281.25
3	2015-01-03	3062.4
4	2015-01-04	3591.35
5	2015-01-05	4169.35
6	2015-01-06	4808.85

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

QUERY:



```
1 -- Determine the top 3 most ordered pizza types
2 -- based on revenue for each pizza category.
3 • select name, revenue from
4 (select category, name, revenue,
5 rank() over (partition by category order by revenue desc) as rn
6 from
7 (select pizza_types1.category, pizza_types1.name,
8 sum((order_details.quantity) * pizzas.price) as revenue
9 from pizza_types1 join pizzas
10 on pizza_types1.pizza_type_id = pizzas.pizza_type_id
11 join order_details
12 on order_details.pizza_id = pizzas.pizza_id
13 group by pizza_types1.category, pizza_types1.name) as a) as b
14 where rn <= 3;
```

OUTPUT:

#	name	revenue
1	The California Chicken Pizza	6007
2	The Barbecue Chicken Pizza	4578.25
3	The Southwest Chicken Pizza	2514.5
4	The Italian Capocollo Pizza	5098.5
5	The Napolitana Pizza	4146
6	The Greek Pizza	4128.5
7	The Spicy Italian Pizza	5590
8	The Italian Supreme Pizza	5516.5