

Pizza_Hut Sales Analysis Project Using SQL



• Saurabh Mishra

Project Overview : Pizza_hut Sales Analysis



This project involves analyzing pizza sales data using SQL to extract valuable insights. By querying the database, we aim to uncover trends in sales, popular pizza types, customer preferences, and peak ordering times. The analysis will provide actionable information for optimizing inventory management, marketing strategies, and overall business operations. Key metrics such as revenue by region, customer demographics, and seasonal variations will be examined to guide decision-making and enhance profitability.

• Saurabh Mishra



The main queries we are looking at

Basic :-

01

Retrieve the total number of orders placed.

02

Calculate the total revenue generated from pizza sales

03

Identify the highest-priced pizza.

04

Identify the most common pizza size ordered.

05

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

Intermediate :-

06

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

07

Determine the distribution of orders by hour of the day.

08

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

09

Determine the top 3 most ordered pizza types based on revenue





Advanced



10

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

11

Analyze the cumulative revenue generated over time.

12

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



Data Tables

```
1• use pizza_hut;
```

```
2• select * from order_details;
```

	order_details_id	order_id	pizza_id	quantity
▶	1	1	hawaiian_m	1
2	2	2	classic_dlx_m	1
3	2	2	five_cheese_l	1
4	2	2	ital_supr_l	1
5	2	2	mexicana_m	1
6	2	2	thai_dkn_l	1

```
1• use pizza_hut;
```

```
2• select * from pizzas;
```

	pizza_type_id	name	category	ingredients
▶	bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Pepp...
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...	
ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...	
ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Gar...	
southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, ...	
thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, T...	

```
1• use pizza_hut;
```

```
2• select * from orders;
```

	order_id	date	time
▶	1	2015-01-01	11:38:36
2	2015-01-01	11:57:40	
3	2015-01-01	12:12:28	
4	2015-01-01	12:16:31	
5	2015-01-01	12:21:30	
6	2015-01-01	12:29:36	

```
1• use pizza_hut;
```

```
2• select * from pizzas;
```

	pizza_id	pizza_type_id	size	price
▶	bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75	
bbq_ckn_l	bbq_ckn	L	20.75	
cali_ckn_s	cali_ckn	S	12.75	
cali_ckn_m	cali_ckn	M	16.75	
cali_ckn_l	cali_ckn	L	20.75	

01. Retrieve the total number of orders placed.



```
1 -- Retrieve the total number of orders placed.  
2 • SELECT  
3     COUNT(order_id) AS total_orders  
4 FROM  
5     order_details;
```

Result Grid	
	total_orders
▶	48620

2. Calculate the total revenue generated from pizza sales.

```
1  -- Calculate the total revenue generated from pizza sales.
2 • SELECT
3   ROUND(SUM((order_details.quantity * pizzas.price)),
4         2) AS Total_revenue
5 FROM
6   order_details
7   JOIN
8     pizzas ON pizzas.pizza_id = order_details.pizza_id;
```



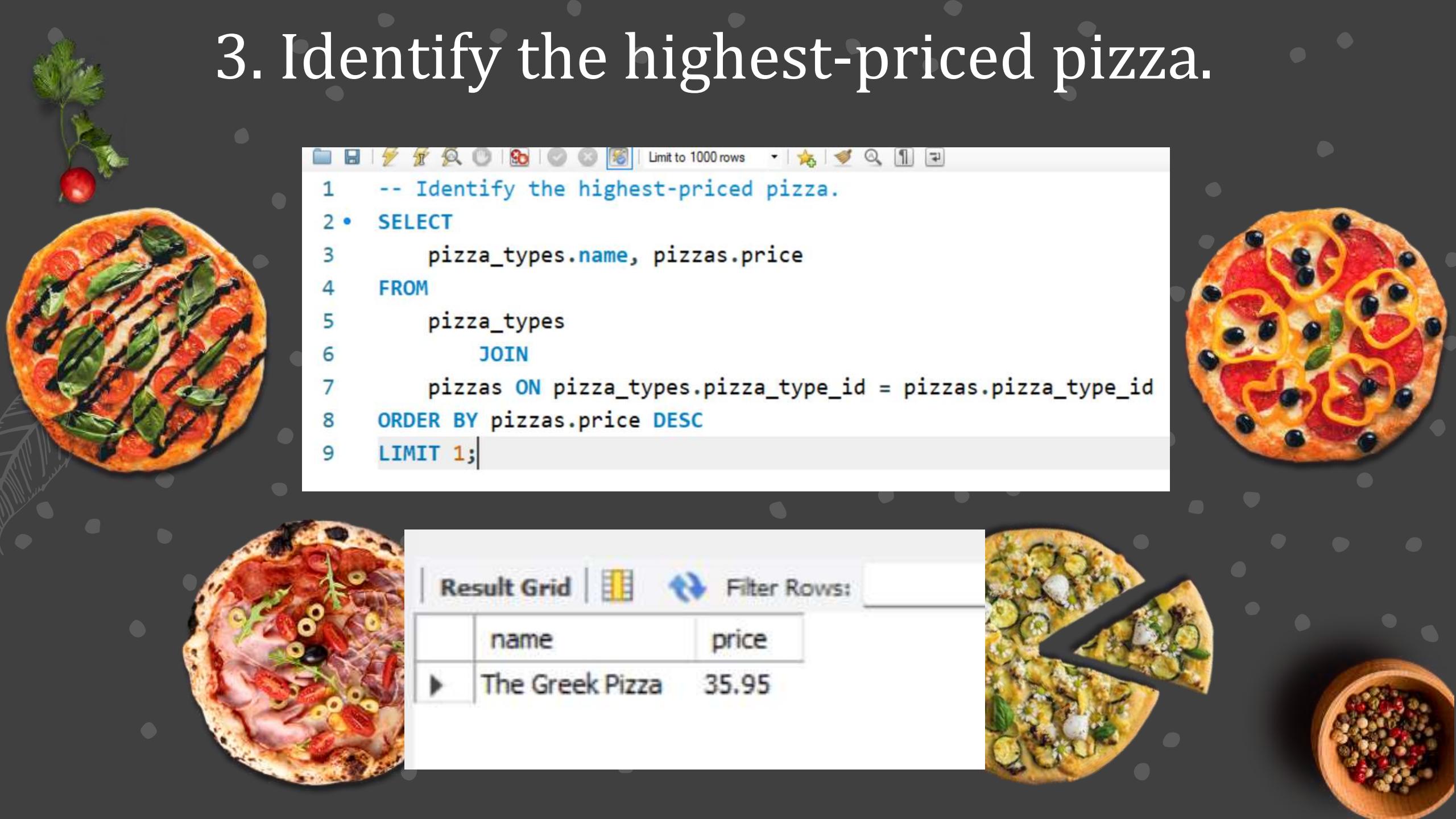
Result Grid	
	Total_revenue
▶	817860.05



3. Identify the highest-priced pizza.

```
1  -- Identify the highest-priced pizza.  
2 • SELECT  
3      pizza_types.name, pizzas.price  
4  FROM  
5      pizza_types  
6      JOIN  
7      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
8  ORDER BY pizzas.price DESC  
9  LIMIT 1;
```

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



4. Identify the most common pizza size ordered.

```
1 -- Identify the most common pizza size ordered.  
2 • SELECT  
3     pizzas.size,  
4     COUNT(order_details.order_details_id) AS order_count  
5   FROM  
6     pizzas  
7       JOIN  
8       order_details ON pizzas.pizza_id = order_details.pizza_id  
9   GROUP BY pizzas.size  
10  ORDER BY order_count DESC;
```

Result Grid | Filter Row

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

Pizza of love

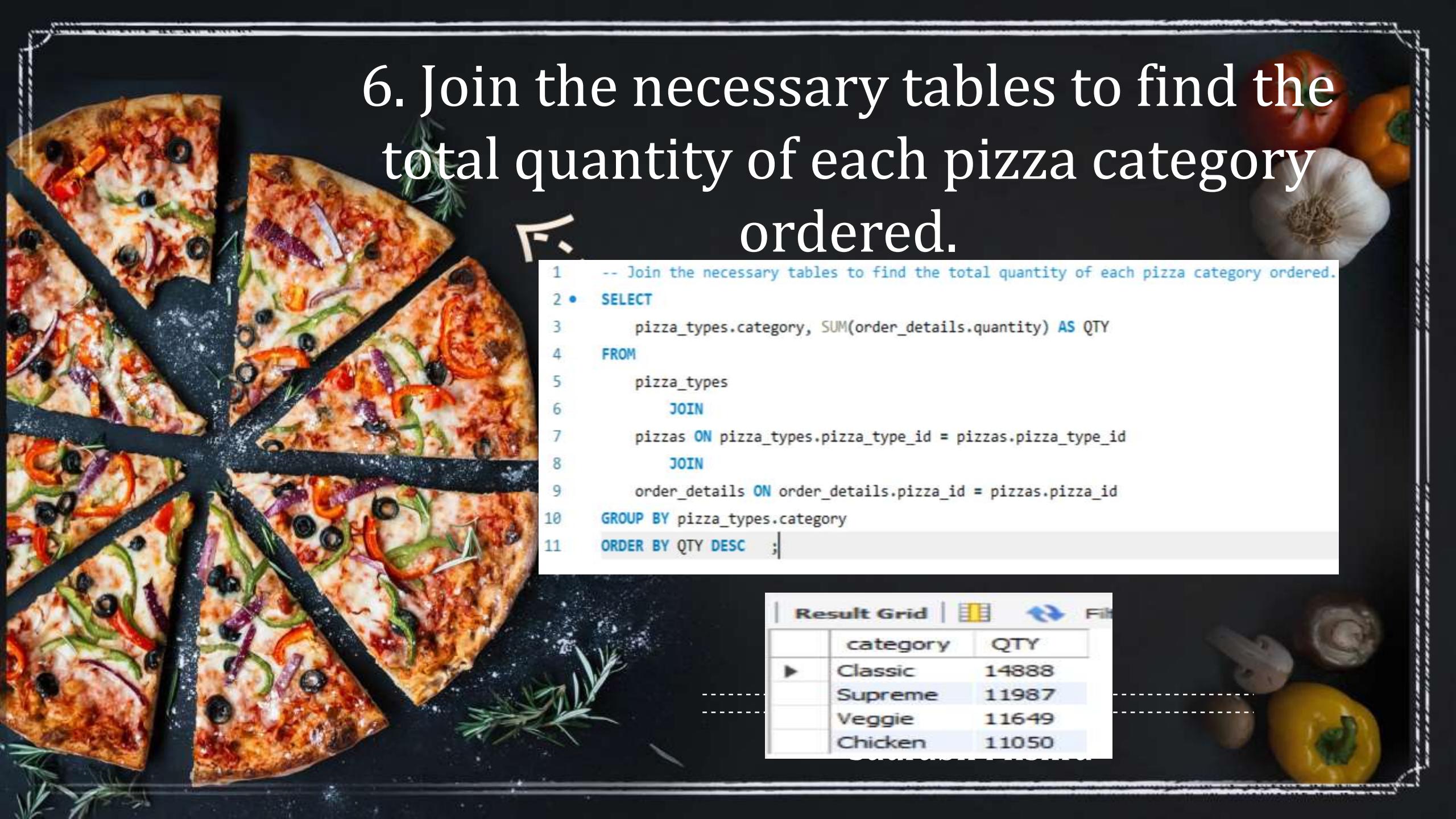




5. List the top 5 most ordered pizza types along with their quantities.

```
1  -- List the top 5 most ordered pizza types along with their quantities.
2 • SELECT
3   pizza_types.name, SUM(order_details.quantity) AS Qty
4   FROM
5     pizza_types
6   JOIN
7     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8   JOIN
9     order_details ON order_details.pizza_id = pizzas.pizza_id
10  GROUP BY pizza_types.name
11  ORDER BY Qty DESC LIMIT 5;
```

Result Grid		Filter Rows:
	name	Qty
▶	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.

F.

```
1 -- Join the necessary tables to find the total quantity of each pizza category ordered.  
2 • SELECT  
3     pizza_types.category, SUM(order_details.quantity) AS QTY  
4 FROM  
5     pizza_types  
6     JOIN  
7     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
8     JOIN  
9     order_details ON order_details.pizza_id = pizzas.pizza_id  
10 GROUP BY pizza_types.category  
11 ORDER BY QTY DESC ;
```

Result Grid		
	category	QTY
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

	Hourly	Order_Count
11	1231	
12	2520	
13	2455	
14	1472	1472
15	1468	
16	1920	
17	2336	
18	2399	
19	2009	
20	1642	
21	1198	
22	663	
23	28	

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



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

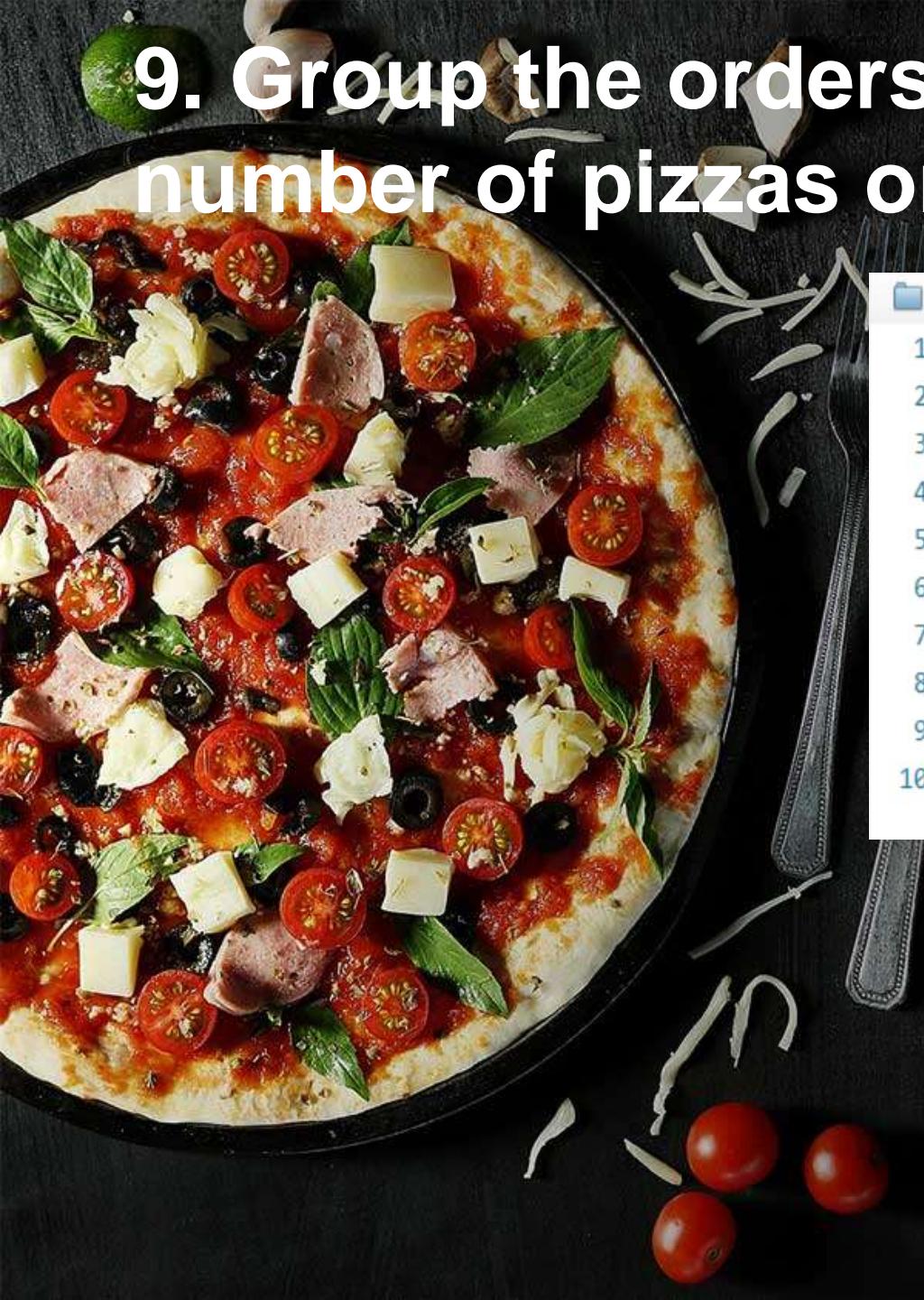


```
1 -- Join relevant tables to find the category-wise distribution of pizzas.  
2 • SELECT  
3     category, COUNT(name)  
4 FROM  
5     pizza_types  
6 GROUP BY category;
```

Result Grid | Filter Rows:

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

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



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

Result Grid | Filter Rows:

	AVG_QTY_Per_Day
▶	138



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

```
1  -- Determine the top 3 most ordered pizza types based on revenue.
2 • SELECT
3     pizza_types.name,
4     SUM(order_details.quantity * pizzas.price) AS revenue
5 FROM
6     pizza_types
7     JOIN
8     pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
9     JOIN
10    order_details ON order_details.pizza_id = pizzas.pizza_id
11 GROUP BY pizza_types.name
12 ORDER BY revenue DESC
13 LIMIT 3;
```

The screenshot shows a SQL query editor interface with a code editor at the top and a result grid at the bottom. The code editor contains the SQL query provided above. The result grid displays the top three pizza types ordered by revenue, showing three rows: "The Thai Chicken Pizza" with revenue 43434.25, "The Barbecue Chicken Pizza" with revenue 42768, and "The California Chicken Pizza" with revenue 41409.5. The result grid has columns for "name" and "revenue".

	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 pizza type to total revenue

```
1  -- Calculate the percentage contribution of each pizza type to total revenue
2 • SELECT
3     pizza_types.category,
4     ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
5         ROUND(SUM((order_details.quantity * pizzas.price)),
6             2) AS Total_revenue
7     FROM
8         order_details
9         JOIN
10            pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
11        2) AS revenue
12   FROM
13     pizza_types
14     JOIN
15       pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
16     JOIN
17       order_details ON order_details.pizza_id = pizzas.pizza_id
18   GROUP BY pizza_types.category
19   ORDER BY revenue DESC;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



12. Analyze the cumulative revenue generated over time.

```
1 -- Analyze the cumulative revenue generated over time.  
2 • select date, sum(Revenue) over(order by date) as Cum_Revenue from  
3 (select orders.date, sum(order_details.quantity * pizzas.price) as Revenue  
4 from order_details join pizzas  
5 on order_details.pizza_id = pizzas.pizza_id  
6 join orders  
7 on orders.order_id = order_details.order_id  
8 group by orders.date) as Sales_revenue;
```

Result Grid		
	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
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4
	2015-01-10	23990.35000000002
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.30000000003

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

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

	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
	The Spicy Italian Pizza	34831.25
	The Italian Supreme Pizza	33476.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.70000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5

THANK YOU

Saurabh Mishra

<https://www.linkedin.com/in/saurabhmishra9/>

<https://github.com/SaurabhMsr>