

Pizza Sales Analysis - SQL Project

PIZZA STORE

Hello!

In this project, the data from a pizza store is analyzed to gain insights into customer behavior, sales patterns, and popular items. By leveraging SQL queries and techniques, this project demonstrates how data can be used to improve business decisions and drive success in the competitive pizza market.

Business Queries

Basic:

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.

Intermediate:

1. Join the necessary tables to find the total quantity of each pizza category ordered.
2. Determine the distribution of orders by hour of the day.
3. Join relevant tables to find the category-wise distribution of pizzas.
4. Group the orders by date and calculate the average number of pizzas ordered per day.
5. Determine the top 3 most ordered pizza types based on revenue.

Advanced:

1. Calculate the percentage contribution of each pizza type to total revenue.
2. Analyze the cumulative revenue generated over time.
3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Retrieve the total number of orders placed.

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

Result Grid	
	total_orders
▶	21350

Calculate the total revenue generated from pizza sales.

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

Result Grid	
	total_sales
▶	817860.05

Identify the highest-priced pizza.

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

Result Grid | Filter Row

	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

```
1 -- Identify the most common pizza size ordered.  
2  
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  
10 GROUP BY pizzas.size  
11 ORDER BY order_count DESC;
```

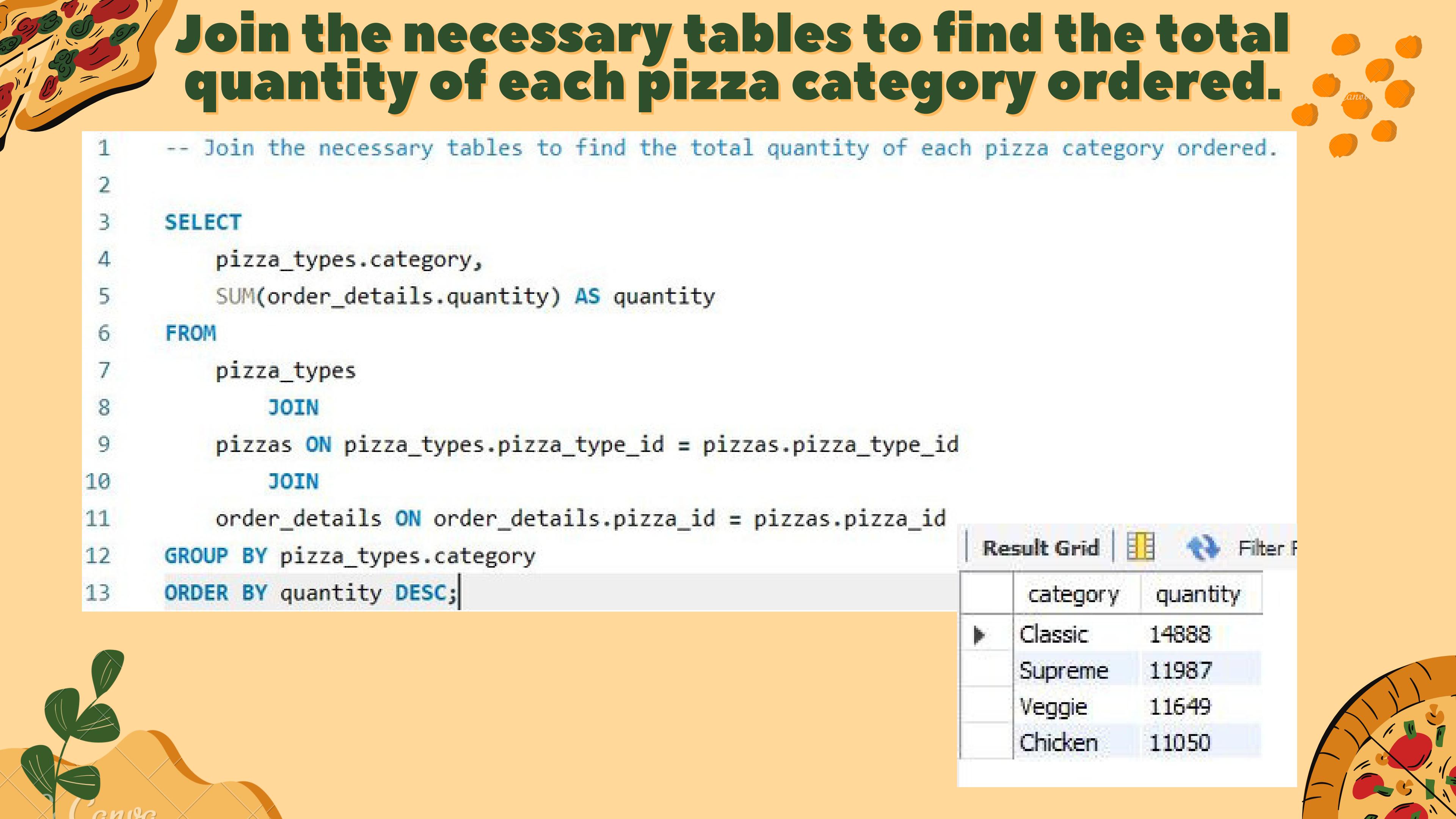
	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

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.pizza_types
2
3 SELECT
4     pizza_types.name, SUM(order_details.quantity) AS quantity
5 FROM
6     pizza_types
7     JOIN
8         pizzas ON pizza_types.pizza_type_id = pizzas.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 quantity DESC
13 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



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

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

Result Grid | Filter F

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day.

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

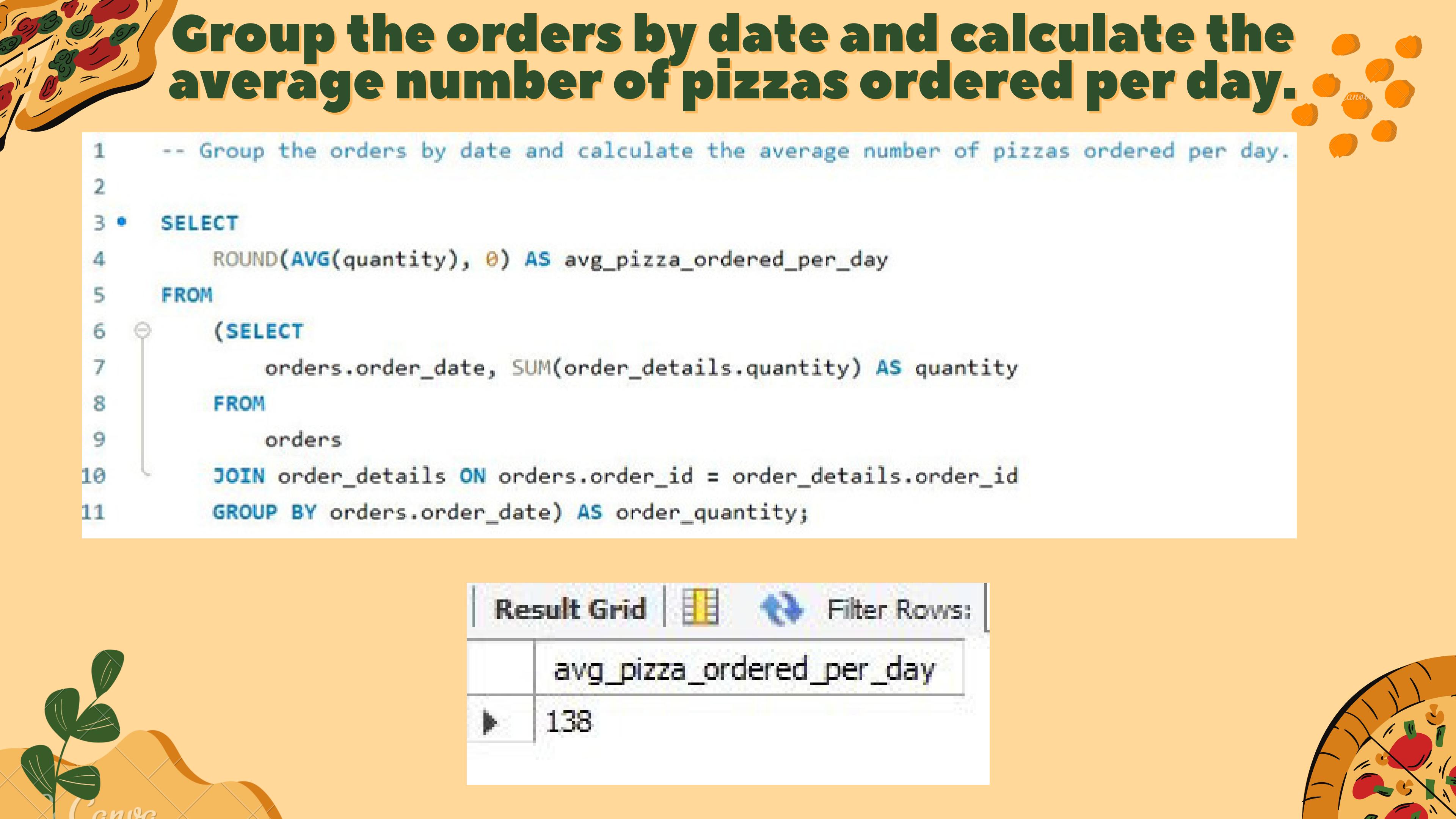
	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8
	9	1

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

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

Result Grid | Filter Rows

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

Result Grid | Filter Rows: []

	avg_pizza_ordered_per_day
▶	138

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

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

Result Grid | Filter Rows:

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

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

```
1  -- Calculate the percentage contribution of each pizza type to total revenue.  
2  
3 • SELECT  
4      pizza_types.category,  
5      ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT  
6          ROUND(SUM(order_details.quantity * pizzas.price),  
7          2) AS total_sales  
8      FROM  
9          order_details  
10     JOIN  
11         pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100,  
12     2) AS revenue  
13    FROM  
14      pizza_types  
15     JOIN  
16         pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
17     JOIN  
18         order_details ON order_details.pizza_id = pizzas.pizza_id  
19     GROUP BY pizza_types.category  
20     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.

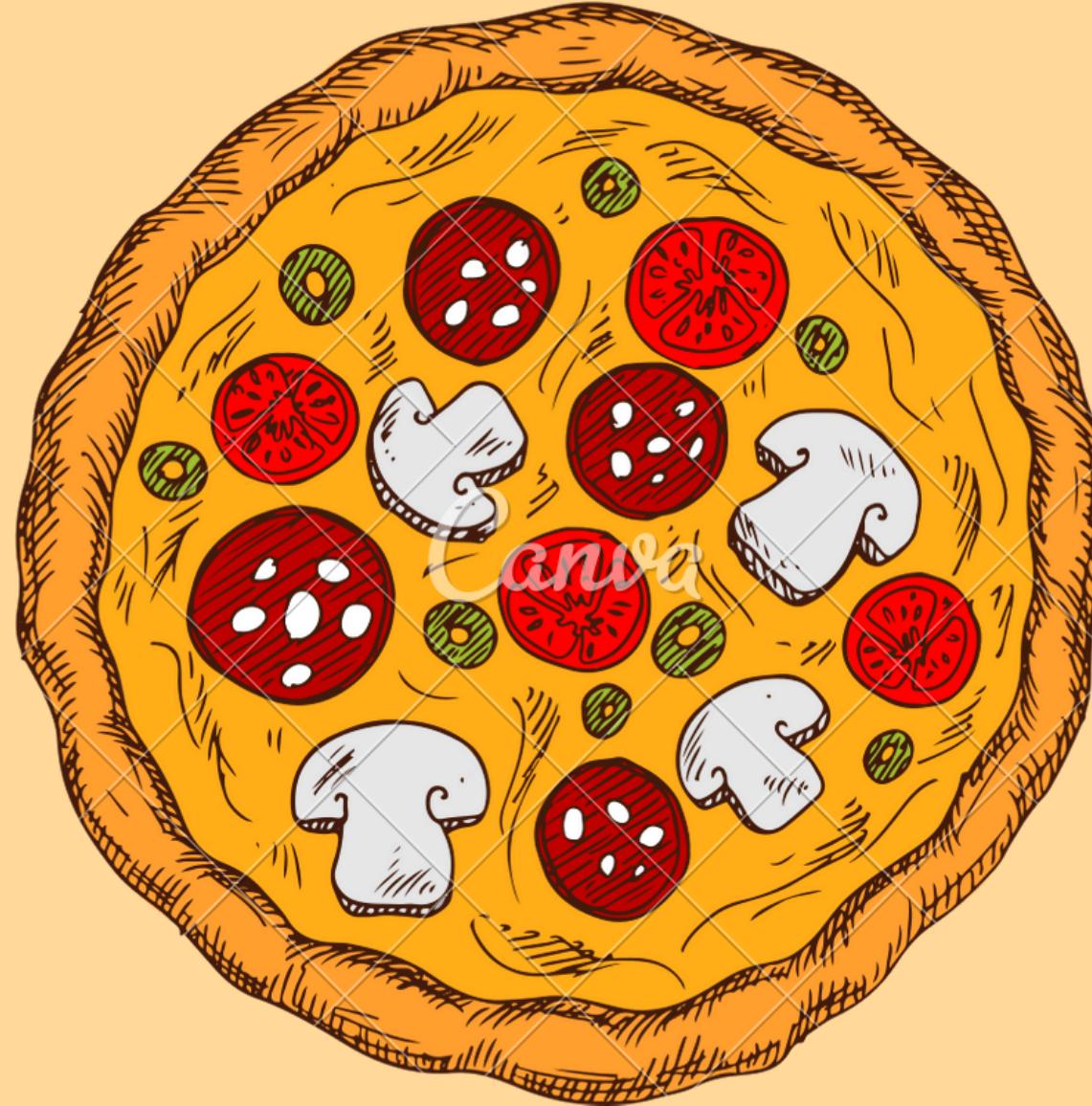
```
1 -- Analyze the cumulative revenue generated over time.  
2  
3 • select order_date,  
4     sum(revenue) over (order by order_date) as cum_revenue  
5   from  
6   (select orders.order_date, sum(order_details.quantity * pizzas.  
7       price) as revenue  
8     from order_details join pizzas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

	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
	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

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
2   each pizza category.
3
4 • select name,revenue from
5   (select category, name, revenue,
6     rank() over (partition by category order by revenue desc)as ranking
7   from
8     (select pizza_types.name, pizza_types.category, sum(order_details.
9       quantity * pizzas.price) as revenue
10      from pizza_types join pizzas
11        on pizza_types.pizza_type_id = pizzas.pizza_type_id
12      join order_details
13        on order_details.pizza_id = pizzas.pizza_id
14      group by pizza_types.name, pizza_types.category) as a) as b
15   where ranking <=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



By: Shubham Khandelwal

**THANK
YOU**