

# *Analyzing Pizza Sales Using SQL*


Leveraging Data Insights for Improved  
Decision Making

Presented By: Sana Arshad





# *Introduction to Pizza Sales Analysis*

- Objective: Understand how SQL can be used to analyze pizza sales data.
  - Importance: Analyzing sales data can help identify trends, optimize inventory, and enhance marketing strategies.
  - Agenda:
    - Overview of Dataset
    - SQL Queries for Analysis
    - Insights and Recommendations
- 

# Overview of Dataset

- Source: GitHub
- Contents:
- Tables:
  - Pizzas, Pizza\_types, Order\_details, orders
  -
- Fields:
  - order\_details\_id, order\_id, pizza\_id, quantity
  - order\_id, order\_date, order\_time
  - pizza\_type\_id, name, category, ingredients
  - pizza\_id, pizza\_type\_id, size, price

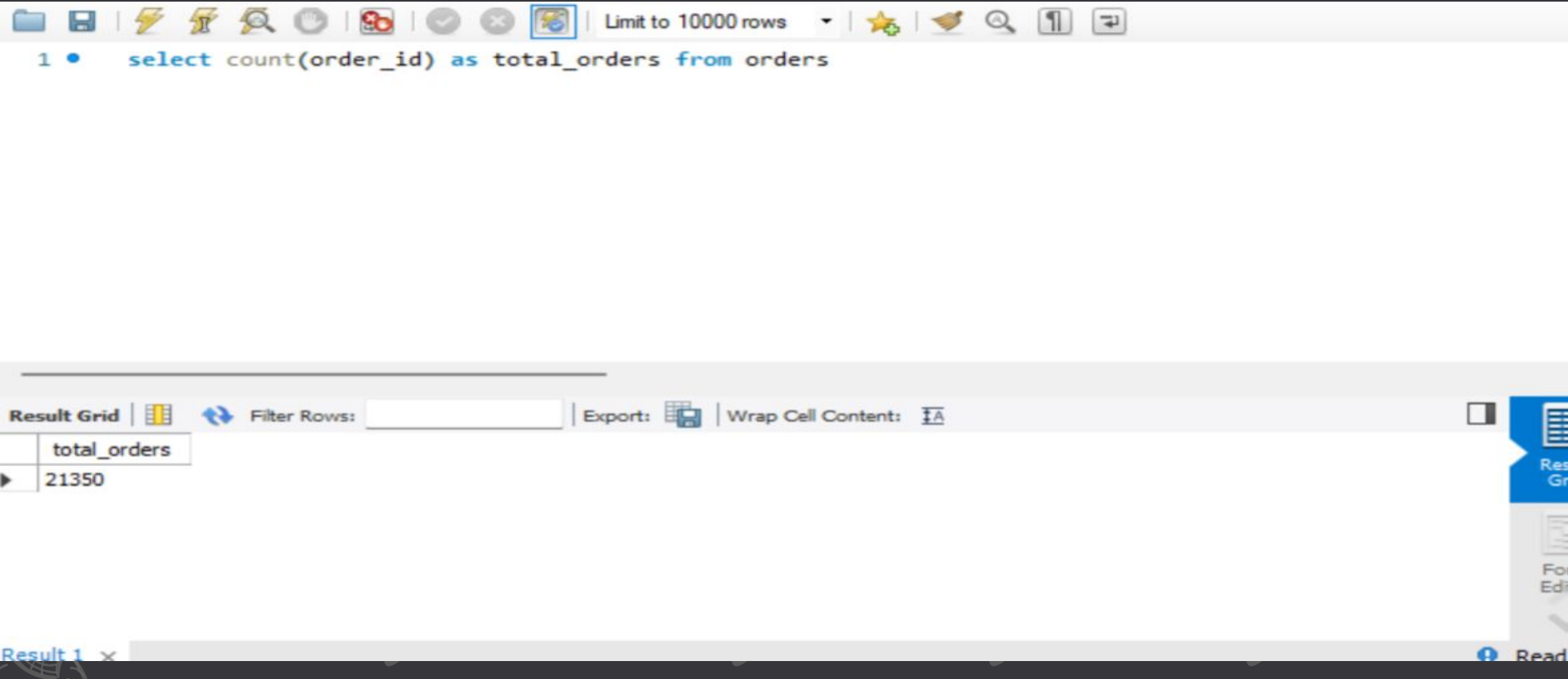


## *SQL Queries for Analysis*

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.
6. Join the necessary tables to find the total quantity of each pizza category ordered.
7. Determine the distribution of orders by hour of the day.
8. Join relevant tables to find the category-wise distribution of pizzas.
9. Group the orders by date and calculate the average number of pizzas ordered per day.
10. Determine the top 3 most ordered pizza types based on revenue.
11. Calculate the percentage contribution of each pizza type to total revenue.



Retrieve the total number of orders placed.



The screenshot shows a SQL query editor with a toolbar at the top containing icons for file operations, execution, and search. A dropdown menu is set to "Limit to 10000 rows". The query editor contains a single SQL statement. Below the editor, the "Result Grid" tab is active, showing a table with one column and one row of data.

```
1 • select count(order_id) as total_orders from orders
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	total_orders
▶	21350

Calculate the total revenue generated from pizza sales.

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

Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content: 

total_revenue
817860.05

# Identify the highest-priced pizza

```
1  -- Identify the highest-priced pizza
2  •  select pizza_types.name, pizzas.price
3     from pizza_types join pizzas on
4     pizza_types.pizza_type_id=pizzas.pizza_type_id
5     order by pizzas.price desc limit 1
6
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



Fetch rows:



	name	price
▶	The Greek Pizza	35.95



Result  
Grid

Form  
Editor

Identify the most common pizza size ordered.



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

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

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



Result  
Grid



Form  
Editor



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 pizza_types.name, sum(order_details.quantity) as total
3     from pizza_types join pizzas on
4     pizza_types.pizza_type_id= pizzas.pizza_type_id
5     join order_details on
6     order_details.pizza_id=pizzas.pizza_id
7     group by pizza_types.name order by total desc limit 5
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



Fetch rows:



Result  
Grid



Form  
Editor

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

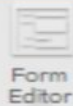
Join the necessary tables to find the total quantity of each pizza category ordered.



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

Result Grid | Filter Rows: | Export: | Wrap Cell Content: ☐

	category	total
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



Determine the distribution of orders by hour of the day.



Query 1   q1   q2   q3   q4   q5   q6   **q7** ×

Limit to 10000 rows

```
1  -- Determine the distribution of orders by hour of the day.
2  • select hour(order_time) as hours ,
3     count(order_id) as order_count from orders group by hour(order_time)
```

**Result Grid** | Filter Rows: | Export: | Wrap Cell Content: |

	hours	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336

Result 1 ×

Result Grid  
Form Editor  
Read Only

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



Query 1   q1   q2   q3   q4   q5   q6   q7   **q8** ×

Limit to 10000 rows

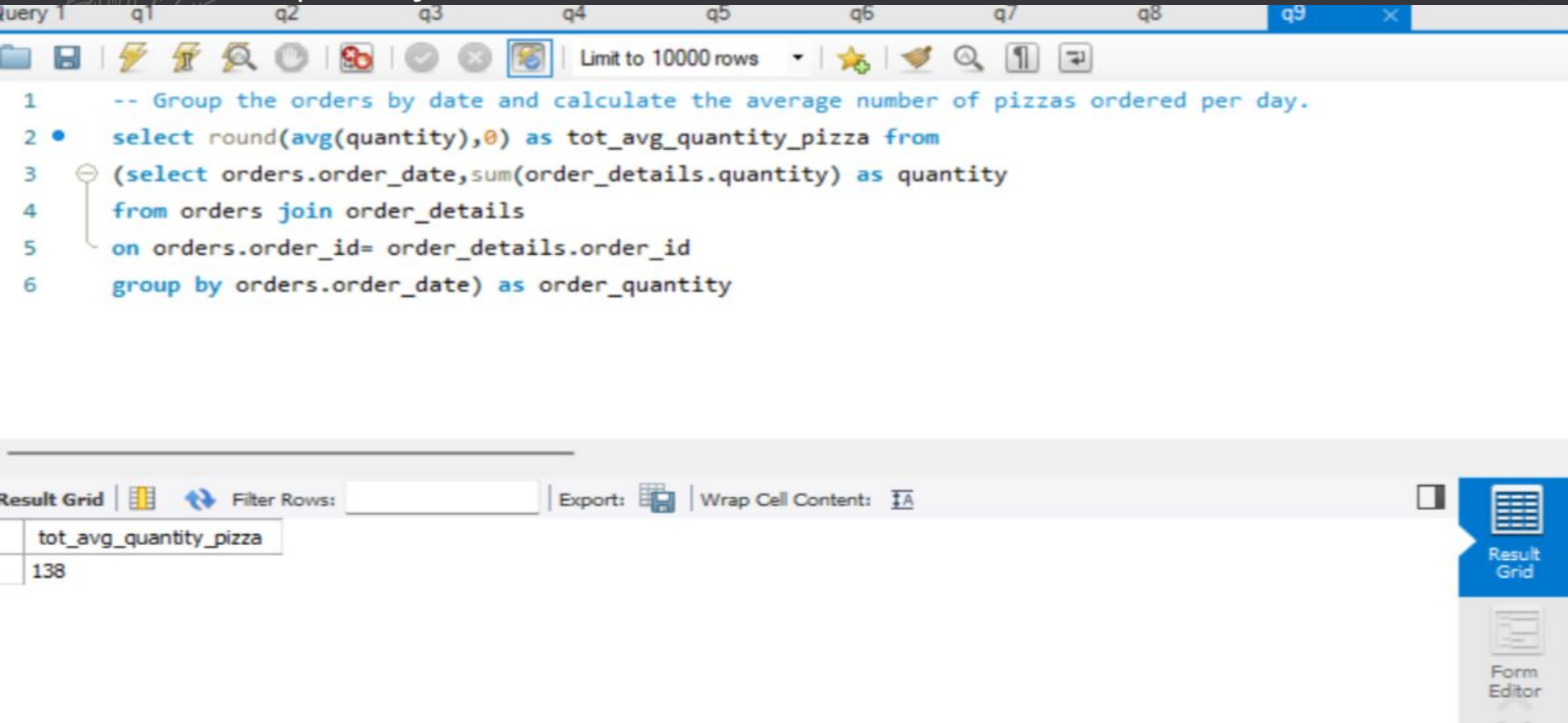
```
1 -- Join relevant tables to find the category-wise distribution of pizzas.
2 • select category, count(name) from pizza_types group by category
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

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

Result Grid  
Form

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



The screenshot shows a database query editor interface. At the top, there are tabs for queries q1 through q9, with q9 selected. Below the tabs is a toolbar with various icons for saving, undo, redo, and other functions. The main area displays a SQL query:

```
1  -- Group the orders by date and calculate the average number of pizzas ordered per day.
2  • select round(avg(quantity),0) as tot_avg_quantity_pizza from
3  (select orders.order_date,sum(order_details.quantity) as quantity
4   from orders join order_details
5   on orders.order_id= order_details.order_id
6   group by orders.order_date) as order_quantity
```

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid shows the following data:

tot_avg_quantity_pizza
138

On the right side of the interface, there are buttons for "Result Grid" and "Form Editor".

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



Query 1   q1   q2   q3   q4   q5   q6   q7   q8   q9   **q10** x

Limit to 10000 rows

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows: |

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

Result 1 x

Result Grid  
Form Editor  
Read Only



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



q1 q2 q3 q4 q5 q6 q7 q8 q9 q10 q11

Limit to 10000 rows

```
1  -- Calculate the percentage contribution of each pizza type to total revenue.
2  • select pizza_types.category, round(sum(order_details.quantity*pizzas.price ) / (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 )*100,2 ) as revenue
6      from pizza_types join pizzas on
7      pizza_types.pizza_type_id=pizzas.pizza_type_id
8      join order_details on order_details.pizza_id=pizzas.pizza_id
9      group by pizza_types.category order by revenue desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Result 1 x

Read Only



# *Insights and Recommendations*

## Insight 1: Monthly Sales Trend




Analysis: Identify seasonal variations and peak months.

Recommendation: Adjust marketing strategies and inventory levels accordingly.

## Insight 2: Best Selling Pizza

Analysis: Determine which pizza is most popular among customers. Recommendation: Promote the best-selling pizza through targeted campaigns.







## *Conclusion*

Summary: SQL analysis provides valuable insights into pizza sales trends.



Next Steps: Implement recommendations to optimize sales and enhance customer satisfaction.

## *Thank You!*

