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

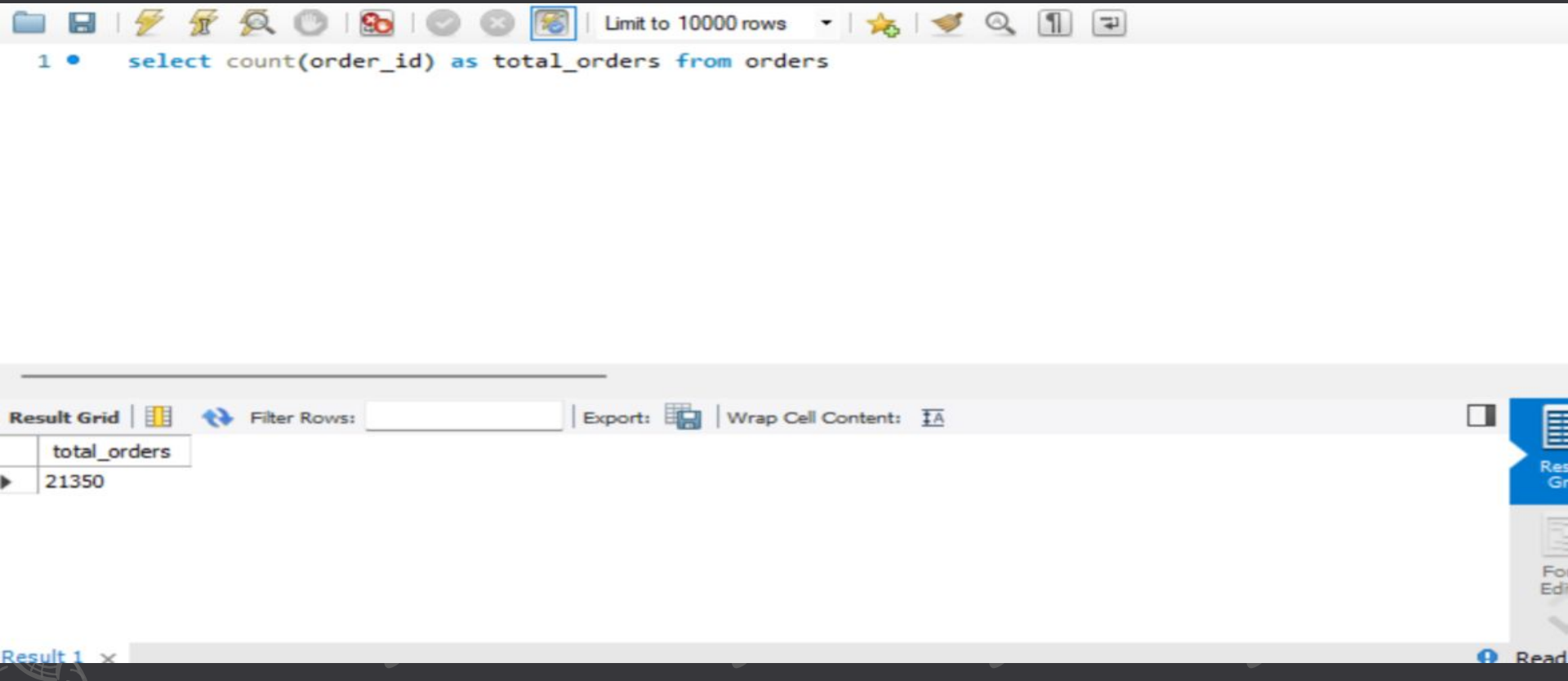
Overview of Dataset

- 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.



Calculate the total revenue generated from pizza sales.

Limit to 10000 rows

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

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

total_revenue
817860.05

Identify the highest-priced pizza

```sql
-- 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: | Export: | Wrap Cell Content: | Fetch rows:

name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

```sql
-- Identify the most common pizza size ordered.
select pizzas.size,count(order_details.order_details_id) as total_orders
from order_details join pizzas on
order_details.pizza_id=pizzas.pizza_id
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

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

Query 1 | q1 | q2 | q3 | q4 | q5

Limit to 10000 rows

```sql
-- List the top 5 most ordered pizza types along with along with their quantities.
select pizza_types.name, sum(order_details.quantity) as total
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 total desc limit 5
```

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

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

Result Grid

Form Editor

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

Result Grid

Form Editor

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
```

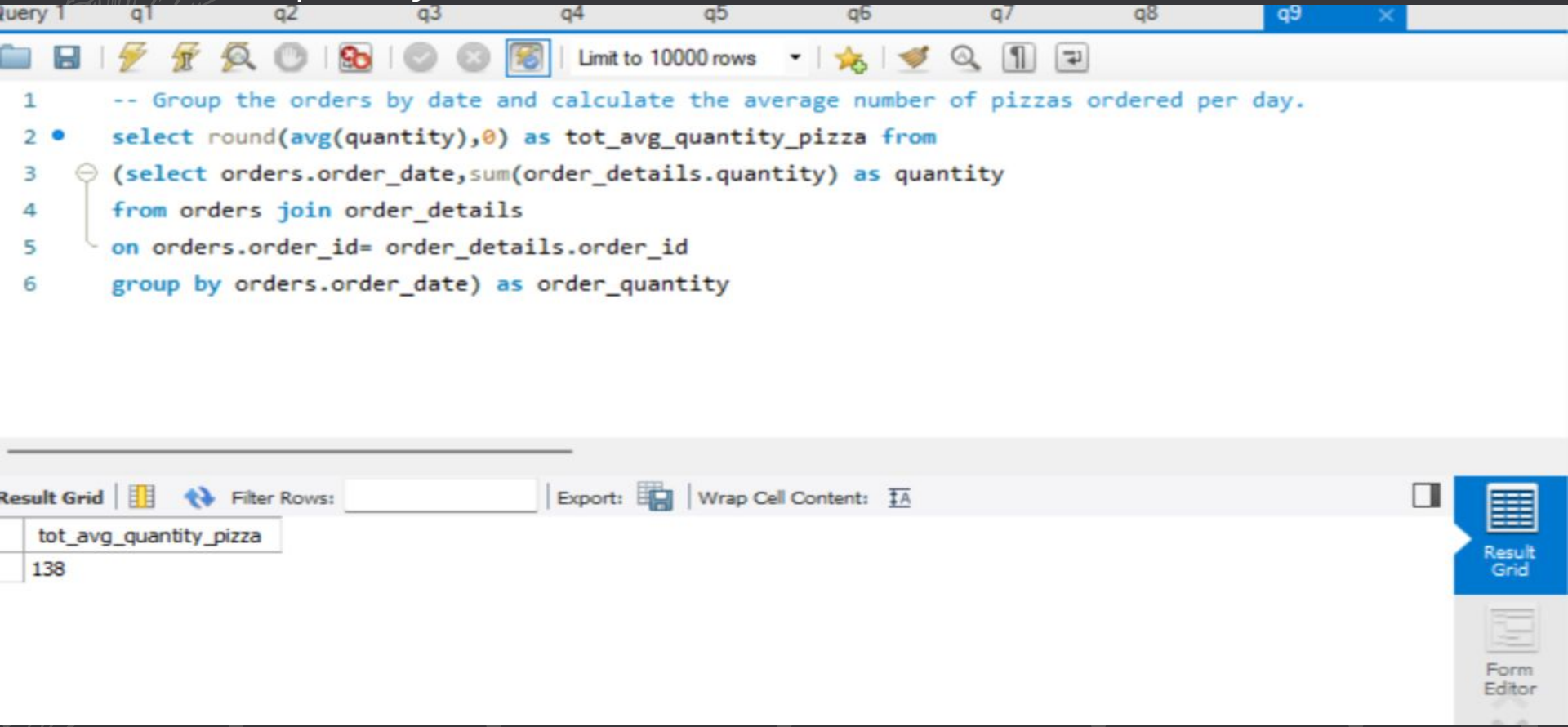
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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.



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

uery 1 | q1 | q2 | q3 | q4 | q5 | q6 | q7 | q8 | q9 | q10

Limit to 10000 rows

```sql
-- 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
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 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 Grid | Form Editor | Read Only

esult 1

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

```sql
-- Calculate the percentage contribution of each pizza type 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 Rows: | Export: | Wrap Cell Content:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Result Grid

Form Editor

Result 1

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!