

Special Pizza Sales Analysis

Presented by
Preet Borad for Mentorness

Pizza
Food



Batch :- MIP-DA-12

introduction Intro

This is a pizza restaurant. Daily how many orders come in one restaurant we are not know . After we are pizza sales analysis based on past data and data analysis through take out insights based on past data to pizza restaurant.





content
Content

1. Data Exploration
2. Data Analysis
3. SQL Proficiency
4. Insights Generation
5. Conclusion

Mentorness

Importing Data in MYSQL

01 Create Database

- create database pizzasales;

02 Load Data from CSV

- Open the database schema.
- Right click and Select “Table data Import wizard”.
- Choose the location of the CSV file.

03 Check the data types

- Check and adjust data types as need.



Dataset

O1 order_details

- order_details_id: Unique identifier for the order detail.
- order_id: Identifier linking to the orders table.
- pizza_id: Identifier linking to the pizza table.
- quantity: Number of pizzas ordered.

O2 orders

- order_id: Unique identifier for the order.
- date: Date the order was placed.
- time: Time the order was placed.



Mentorness

Dataset

03 pizza_types

- pizza_type_id: Unique identifier for the pizza type.
- name: Name of the pizza.
- category: Category of the pizza (e.g., vegetarian, meat).
- ingredients: List of ingredients used in the pizza.

04 pizzas

- pizza_id: Unique identifier for the pizza.
- pizza_type_id: Identifier linking to the pizza_type table.
- size: Size of the pizza (e.g., small, medium, large).
- price: Price of the pizza.

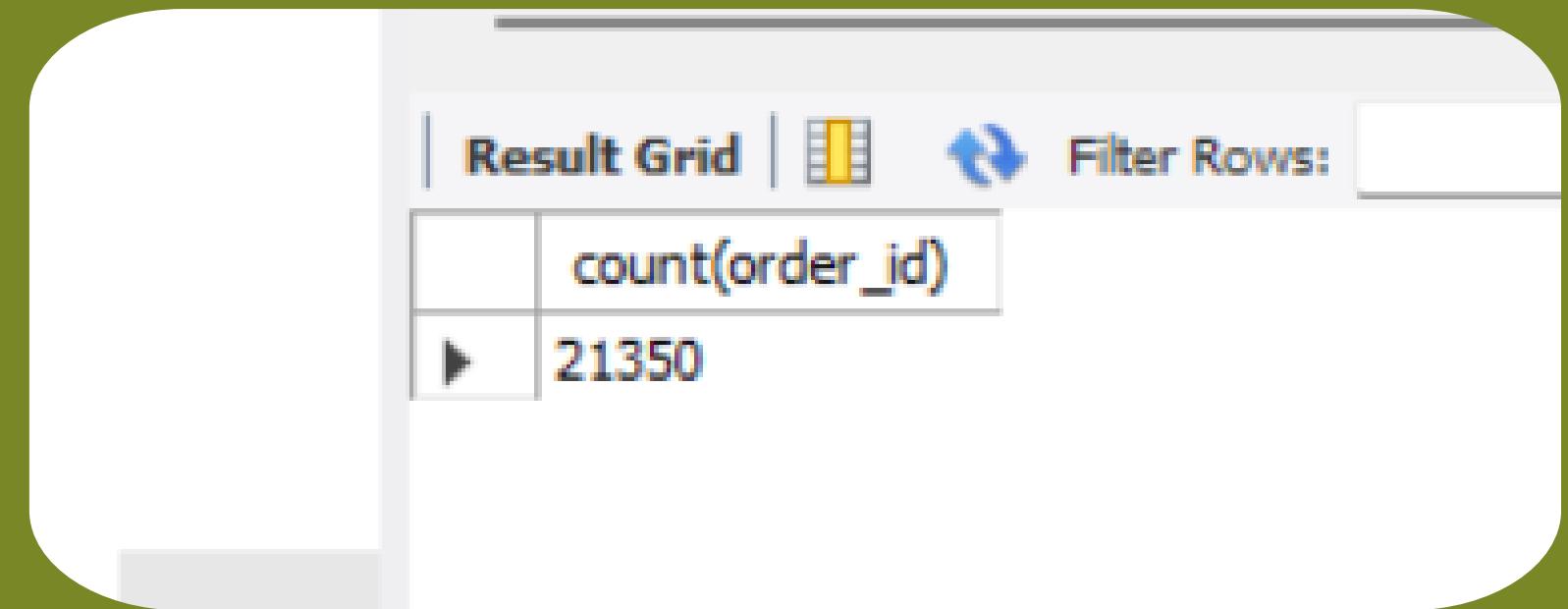


Mentorness

Q1: The total number of order place

Code :-

```
create database pizzasales;  
use pizzasales;  
select count(order_id) from orders;
```



	count(order_id)
▶	21350

Output

Insights :-

- It is show the total orders of pizza in past data to pizza restaurant .

Q2: The total revenue generated from pizza sales

⋮
⋮
• ⋮

Code :- SELECT

```
    ROUND(SUM(order_details.quantity *  
pizzas.price),
```

```
        2) AS revenue
```

```
FROM
```

```
    order_details
```

```
    JOIN
```

```
    pizzas ON order_details.pizza_id =
```

```
    pizzas.pizza_id;
```

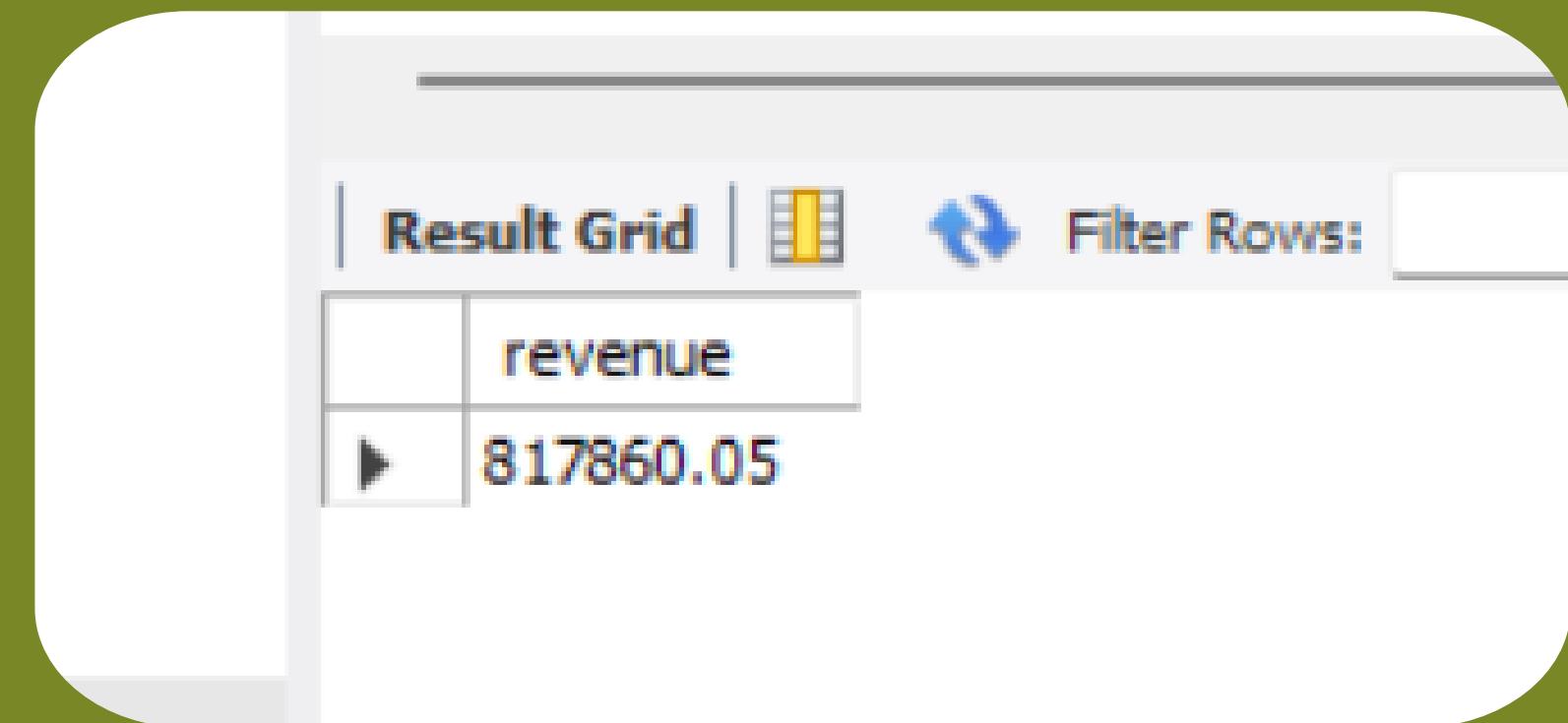
• ⋮

• ⋮

• ⋮

Q2: The total revenue generated from pizza sales

⋮
⋮
⋮
⋮



A screenshot of a database query result grid. The grid has a header row with a single column labeled "revenue". Below it is a data row with a single cell containing the value "817860.05". The grid is contained within a white rounded rectangle with a thin gray border.

revenue
817860.05

Output

Insights :-

- It is show the total revenue based on sales of pizza of quantity.

•
•
•
•

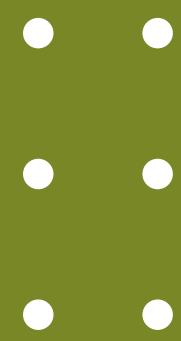
Mentorness

Q3: The highest priced pizza

Code :-

```
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;
```

Q3: The highest priced pizza



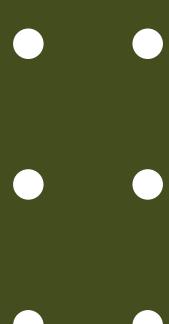
The screenshot shows a software interface with a title bar and a toolbar. The toolbar includes a 'Result Grid' button, a refresh icon, and a 'Filter Rows:' input field. Below the toolbar is a table with two columns: 'name' and 'price'. A single row is displayed, showing 'The Greek Pizza' in the 'name' column and '35.95' in the 'price' column.

	name	price
▶	The Greek Pizza	35.95

Output

Insights :-

- It is show the highest price of pizza to past data of pizza restaurent.



Mentorness

Q4: The most common pizza size ordered

Code :-

```
SELECT  
    pizzas.size,  
    count(order_details.order_details_id) as  
    count_pizza  
FROM  
    order_details  
    JOIN  
    pizzas ON order_details.pizza_id =  
    pizzas.pizza_id  
GROUP BY pizzas.size  
order by count_pizza desc;
```

Q4: The most common pizza size ordered

⋮
⋮
⋮
⋮

The screenshot shows a database query results grid titled "Result Grid". The grid has three columns: "size", "count_pizza", and a header row. The data rows are: L (18526), M (15385), S (14137), XL (544), and XXL (28). The "count_pizza" column values are highlighted in orange.

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

Output

Insights :-

- Here shows total order based on size of different types of pizza
- size of total orders.

•
•
•
•

Q5: The top 5 most ordered pizza types along their quantities :

Code :-

```
SELECT  
    pizza_types.name,  
    count(order_details.quantity) AS tot_quantity  
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 tot_quantity DESC  
LIMIT 5;
```

Q5: The top 5 most ordered pizza types along their quantities :

The screenshot shows a database query results window titled "Result Grid". The table has two columns: "name" and "tot_quantity". The data is as follows:

	name	tot_quantity
▶	The Classic Deluxe Pizza	2416
	The Barbecue Chicken Pizza	2372
	The Hawaiian Pizza	2370
	The Pepperoni Pizza	2369
	The Thai Chicken Pizza	2315

Output

Insights :-

- Here are the top 5 most ordered pizza types with their quantity
- insights are the pizza restaurants that generate the most revenue from these top 5 pizza types.

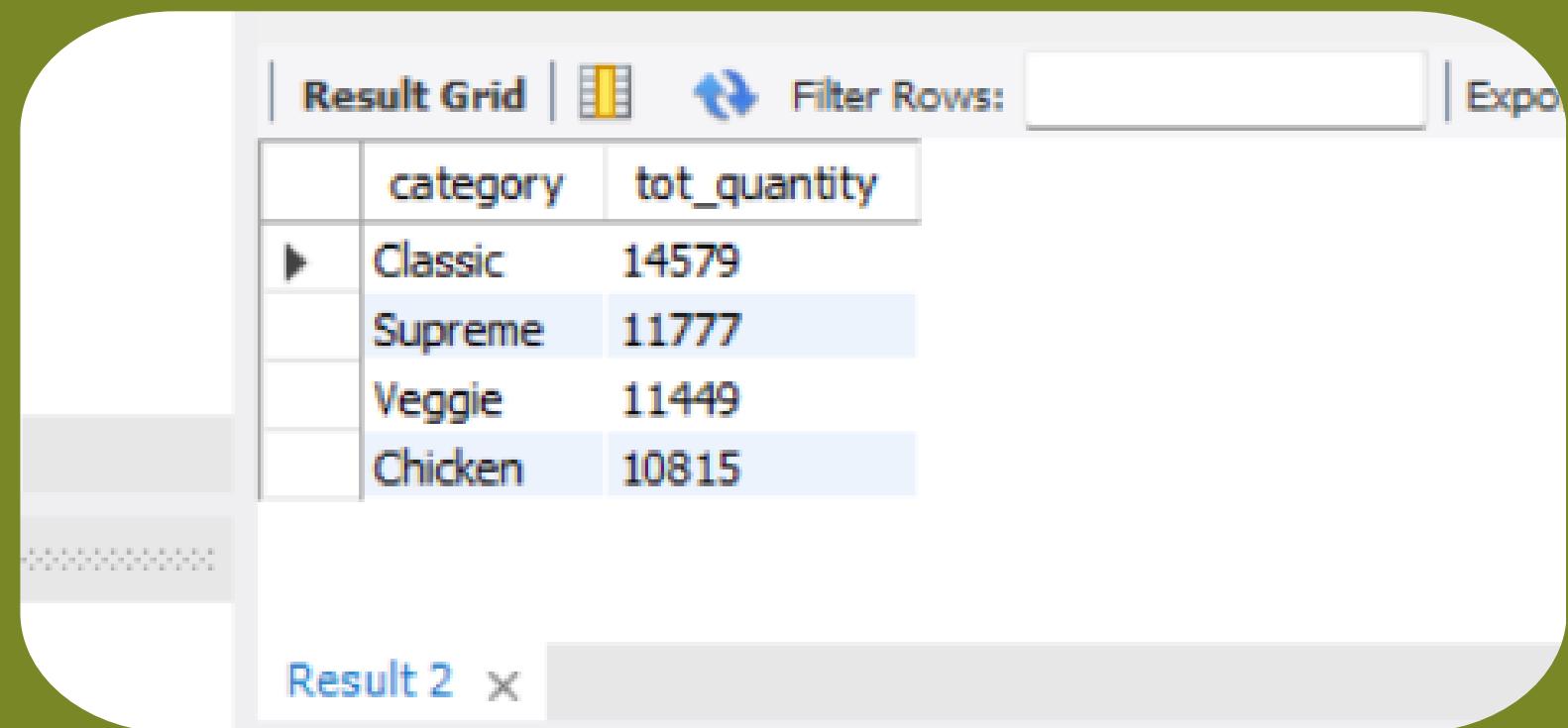
Q6: The quantity of each pizza categories ordered

Code :-

```
SELECT  
    pizza_types.category,  
    COUNT(order_details.order_details_id) AS tot_quantity  
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 tot_quantity DESC;
```

Q6: The quantity of each pizza categories ordered

⋮ ⋮
• ⋮



The screenshot shows a database query results grid titled "Result Grid". The grid has two columns: "category" and "tot_quantity". There are four rows of data: Classic (14579), Supreme (11777), Veggie (11449), and Chicken (10815). The "Supreme" row is highlighted with a light blue background.

	category	tot_quantity
▶	Classic	14579
	Supreme	11777
	Veggie	11449
	Chicken	10815

Output

Insights :-

- Here most of order comes from classic pizza of category.

• ⋮
• ⋮
• ⋮

Mentorness

Q7: The distribution of orders by hours of the day

⋮
⋮
• ⋮

Code :-

```
SELECT  
    HOUR(orders.order_time) AS  
    hour,  
    COUNT(orders.order_id) AS  
    count_order  
FROM  
    orders  
GROUP BY  
    HOUR(orders.order_time);
```

• ⋮
• ⋮
• ⋮

Q7: The distribution of orders by hours of the day



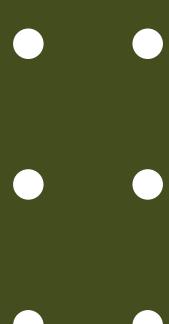
The screenshot shows a database query result titled "Result 3" containing a single table with two columns: "hour" and "count_order". The table data is as follows:

	hour	count_order
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642

Output

Insights :-

- Here show hour of days to show how many orders comes in one hours.



Q8: The category-wise distribution of pizzas

Code :-

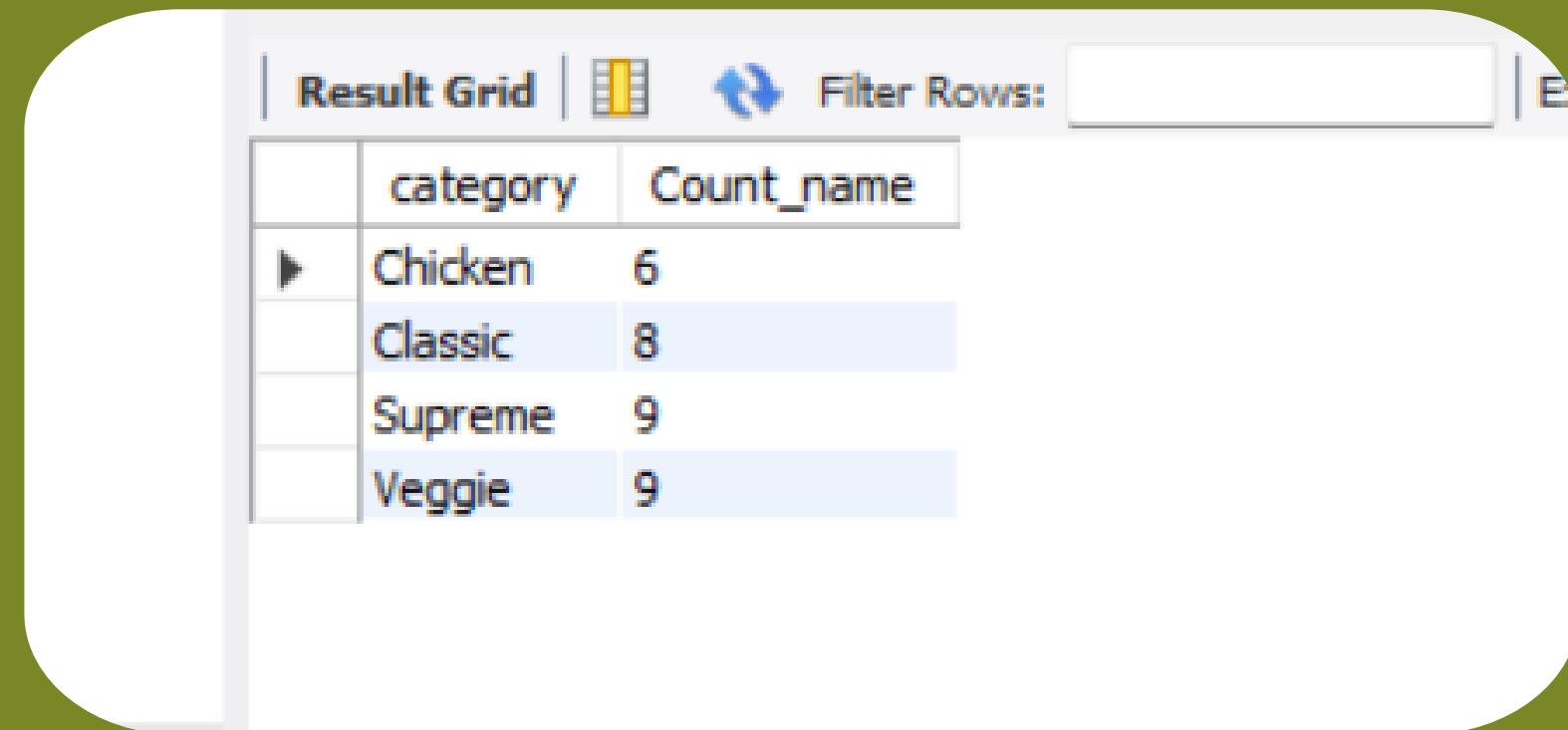
```
SELECT  
    pizza_types.name, (pizza_types.category)  
FROM  
    pizza_types;
```

-- count name wise distribution

```
SELECT  
    pizza_types.category, COUNT(pizza_types.name) AS Tot_name  
FROM  
    pizza_types  
GROUP BY pizza_types.category;
```

Q8: The category-wise distribution of pizzas

⋮
⋮
⋮
⋮



The screenshot shows a software interface for viewing database results. At the top, there are buttons for 'Result Grid' (highlighted in blue), 'Filter Rows:' with a search bar, and other icons. Below is a table with the following data:

	category	Count_name
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Output

Insights :-

- Here category wise most pizza name types is in supreme and veggies to pizza category.

•
•
•
•

Mentorness

Q9: The average number of pizzas ordered per day

Code :-

```
SELECT  
    ROUND(AVG(quantity), 2) as Average_order_day  
FROM  
(SELECT  
    orders.order_date, COUNT(order_details.quantity) AS  
    quantity  
FROM  
    orders  
JOIN order_details ON orders.order_id =  
    order_details.order_id  
GROUP BY orders.order_date) AS a;
```

Q9: The average number of pizzas ordered per day

	Average_order_day
▶	135.81

Output

Insights :-

- Here show the how many orders comes in per days in pizza restaurant.

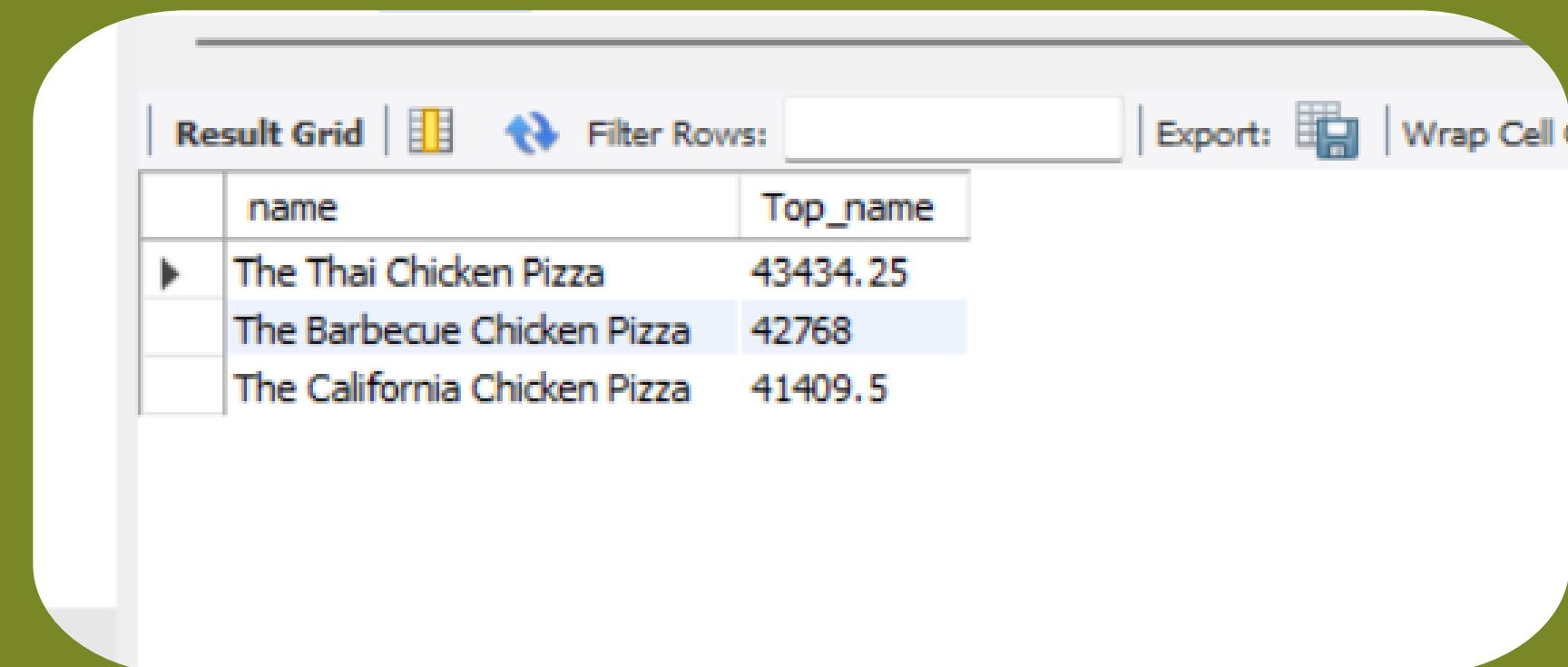
Q10: Top 3 most ordered pizza type base on revenue

• •
• •
• •

Code :-

```
SELECT  
    pizza_types.name,  
    ROUND(SUM(order_details.quantity * pizzas.price),  
        2) AS Top_name  
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 Top_name DESC  
LIMIT 3;
```

Q10: Top 3 most ordered pizza type base on revenue



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

Output

Insights :-

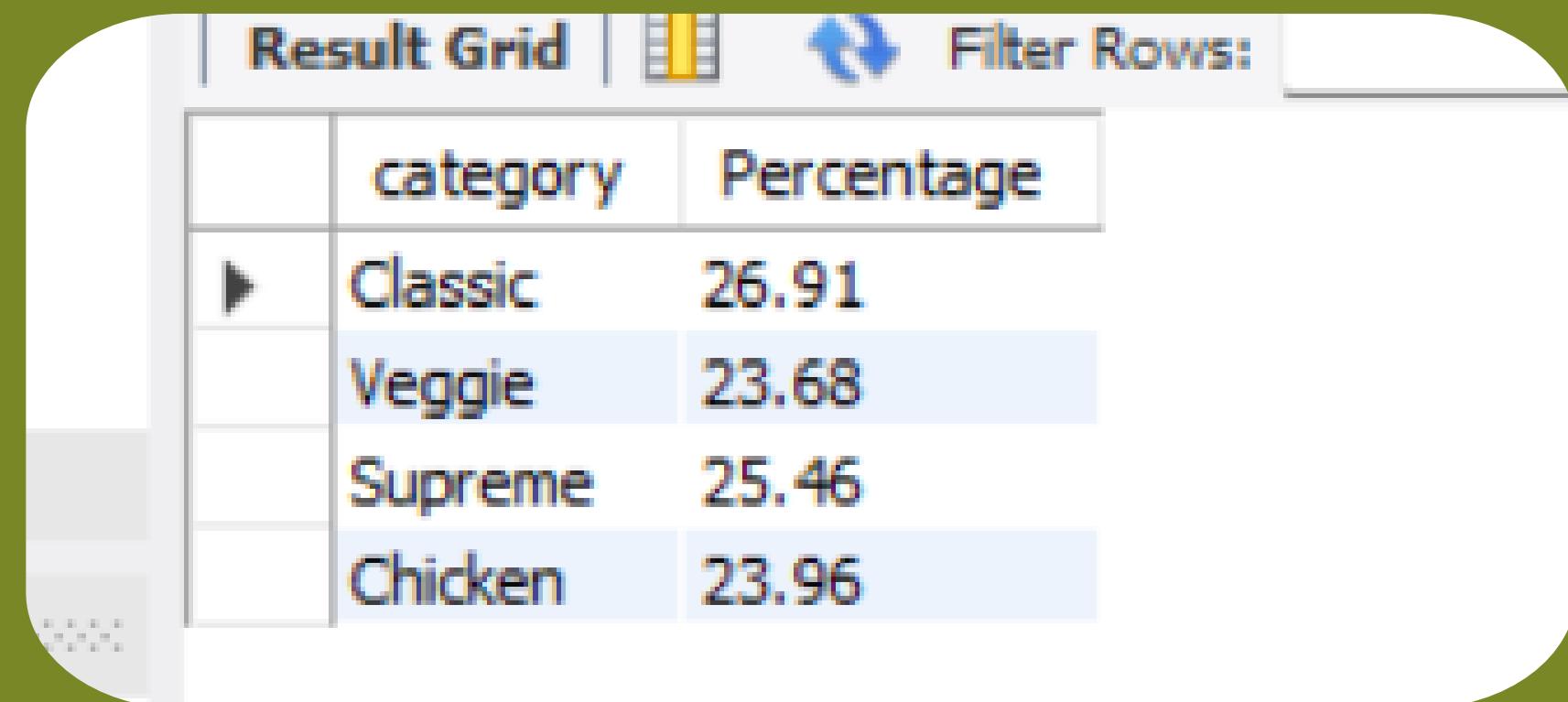
- Here most of the restaurant revenue is generated based on these top 3 pizza names.

Q11: The percentage contribution of each pizza type to revenue :

Code :-

```
SELECT  
    pizza_types.category,  
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT  
        ROUND(SUM(order_details.quantity * pizzas.price), 2) AS  
        tot_sales FROM order_details JOIN  
            pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100, 2) AS  
    Percentage  
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;
```

Q11: The percentage contribution of each pizza type to revenue :



The screenshot shows a software interface with a toolbar at the top containing 'Result Grid' and 'Filter Rows' buttons. Below the toolbar is a table with the following data:

	category	Percentage
>	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

Output

Insights :-

- Here different types of percentage revenue generated based on different pizza category.

Q12: The cumulative revenue generated over time

Code :-

```
select order_date,
       sum(Tot_revenue) over(order by order_date) Comulative
    from
        (select orders.order_date,
               sum(order_details.quantity * pizzas.price) as
                  Tot_revenue
            from order_details join pizzas
              on order_details.pizza_id = pizzas.pizza_id
            join orders
              on orders.order_id = order_details.order_id
           group by orders.order_date) as total;
```

Q12: The cumulative revenue generated over time



The screenshot shows a database query results grid titled "Result Grid". The columns are "order_date" and "Comulative". The data shows the cumulative revenue for each day from January 1st to January 9th, 2015. The cumulative value increases daily, starting at 2713.85 and reaching 21526.4 by January 9th.

	order_date	Comulative
▶	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

Output

Insights :-

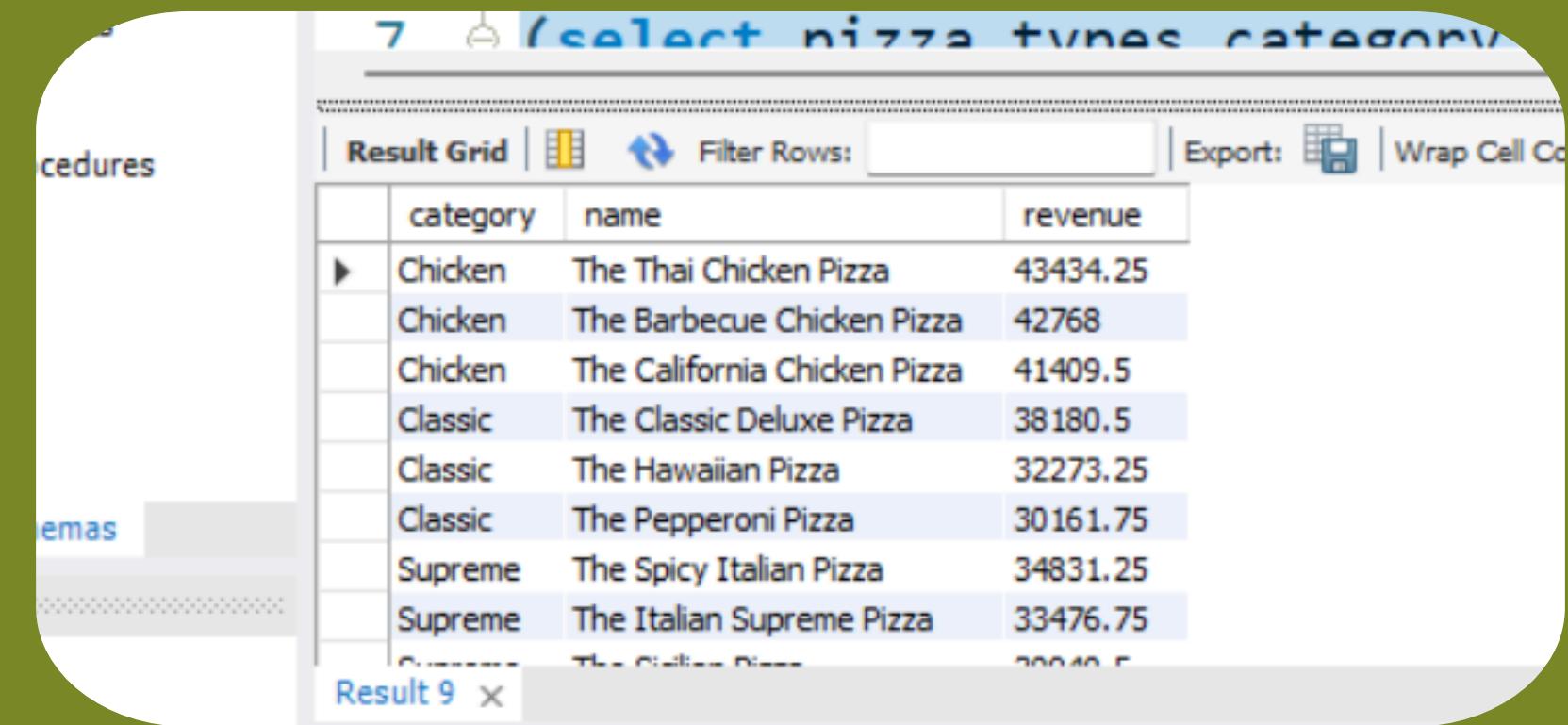
- Here revenue is seen separately on daily basis to show the difference in revenue from previous day and following days.

Q13: The top 3 most ordered pizza type based on revenue for each pizza category.

Code :-

```
select category,name,revenue from
(select category,name,revenue,
rank() over(partition by category order by revenue desc) as corr_order
from
(select pizza_types.category,pizza_types.name,
round(sum(order_details.quantity * pizzas.price),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, pizza_types.name) as a) as b
where corr_order <= 3;
```

Q13: The top 3 most ordered pizza type based on revenue for each pizza category.



The screenshot shows a database query result grid titled '(select pizza types category)'. The grid has three columns: 'category', 'name', and 'revenue'. The data is sorted by revenue in descending order. The top three rows represent the highest-grossing pizzas:

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30040.5

Output

Insights :-

- Here it is seen that the top 3 names of the highest grossing pizzas are found among the types of pizza.
- Here's where most of the revenue comes from.

Conclusion

Finally Done pizza sales analysis.
here lot of insights find on past data
to pizza restaurant. for example high
price for pizza and total revenue
based on pizza category.

Pizza
Food



Mentorness

Thank You
Pizza
Sales
Analysis

Pizza
Food

Mentorness

