

PIZZA SALES

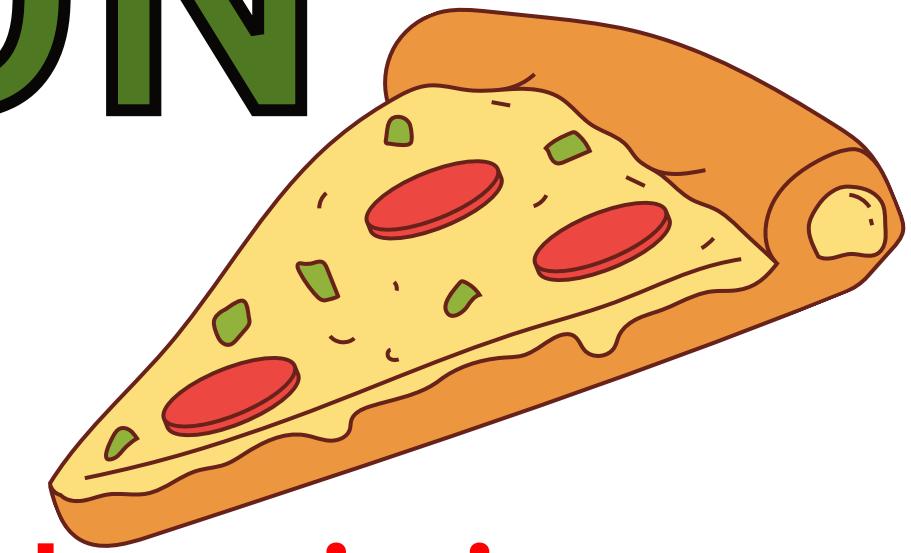
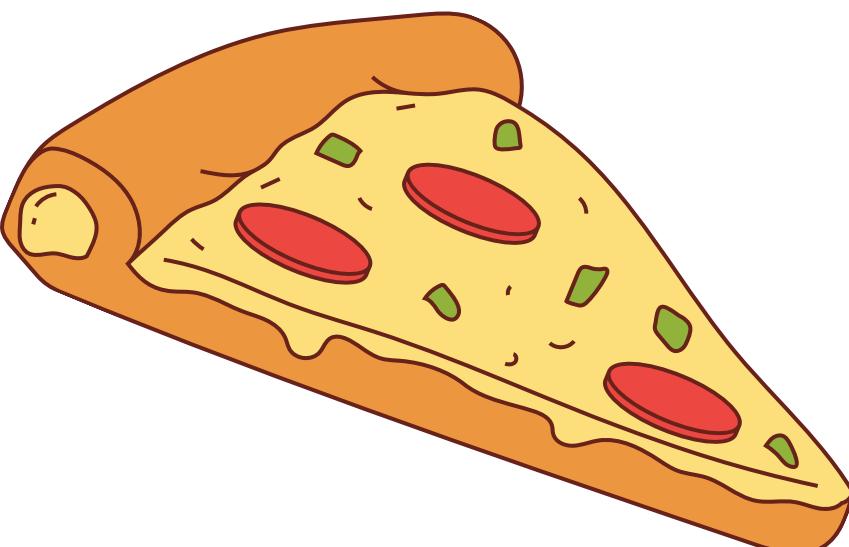
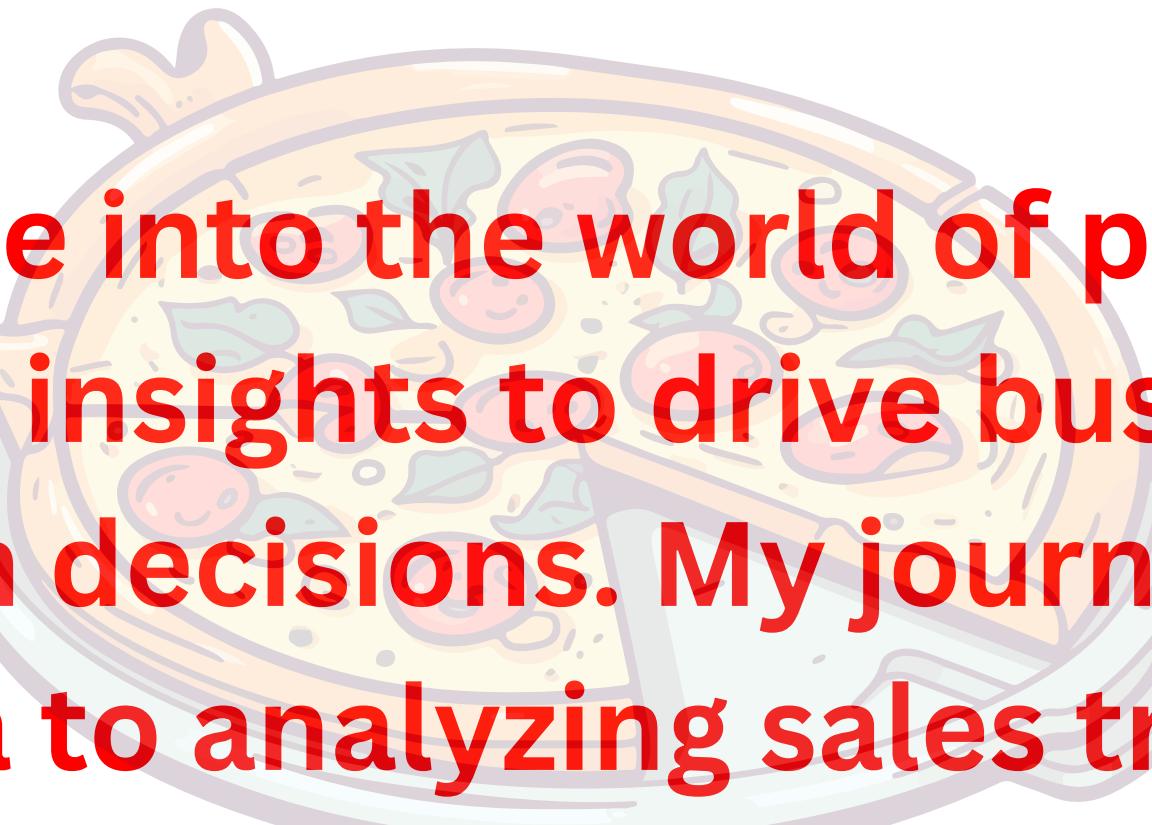
PERFORMANCE REPORT

Using SQL



INTRODUCTION

In this report, I delve into the world of pizza sales, aiming to uncover valuable insights to drive business growth and make data-driven decisions. My journey takes me from exploring the data to analyzing sales trends and product performance.



Problem Statement

- Retrieve the total number of orders placed
- Calculate the total revenue generated from Pizza Sales.
- Identify the highest priced Pizza
- Identify the Most Common Pizza Size ordered.
- List the 5 most ordered pizza type along with their quantity
- Join the necessary tables to find the total quantity of each pizza category ordered
- Determine the distribution of orders by hours of the day
- join the relevant tablet find the category wise disributionof pizzas
- Group The order by data and calculate the average number of pizzas ordered per day
- Determine the top 3 most ordered pizza type based on revenue
- calculate the %age contribution of each pizza type to total revenue
- Analyse the cummulative revenue generated over time
- Determine the top 3 most ordered pizza type based on revenue for each pizza category

Data Exploration

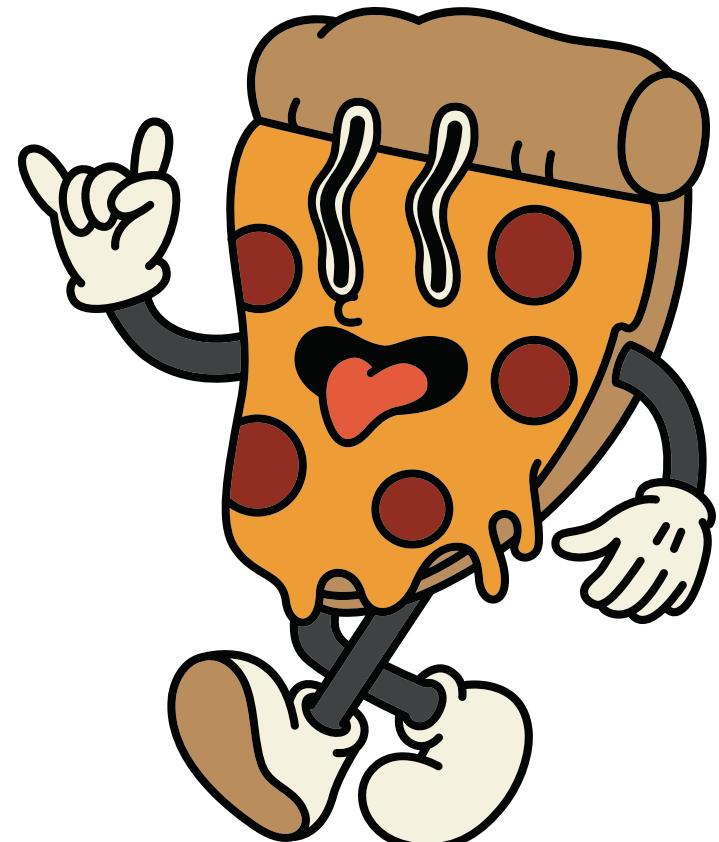
To kickstart my analysis, I first take a look at the dataset. It provides information about pizza orders, including the total price, quantity sold, order date, pizza name, size, and category.

- Retrieve the total number of orders placed

```
Select count (order_id) as total_orders from Pizza_project.dbo.orders;
```

Results	
	total_orders
1	21350

A total number of 21350 pizzas were ordered



- Calculate the total revenue generated from Pizza Sales

```
select sum(dbo.order_details.quantity*dbo.pizzas.price) as total_revenue  
from dbo.order_details  
join dbo.pizzas on  
dbo.order_details.pizza_id=dbo.pizzas.Pizza_id;
```

The screenshot shows a SQL query results window. At the top, there are two tabs: "Results" (selected) and "Messages". The results table has one row with the following data:

	total_revenue
1	817860.05083847



The total revenue generated is 817860.05



- Identify the highest priced Pizza



```
SELECT TOP 1 dbo.pizza_types.name, dbo.pizzas.price, dbo.pizzas.size  
FROM dbo.pizzas  
JOIN dbo.pizza_types ON dbo.pizzas.pizza_type_id = dbo.pizza_types.pizza_type_id  
ORDER BY dbo.pizzas.price DESC;
```

The screenshot shows a SQL query results window with two tabs: "Results" and "Messages". The "Results" tab is selected, displaying a table with three columns: "name", "price", and "size". A single row is shown, representing the most expensive pizza found by the query.

	name	price	size
1	The Greek Pizza	35.9500007629395	XXL

The most expensive pizza selling is “The Greek Pizza”, Size - XXL, priced at \$35.95

- Identify the Most Common Pizza Size ordered.

```
select top 1 dbo.pizzas.size, count(dbo.order_details.order_details_id) as total_count  
from dbo.order_details  
join dbo.pizzas on  
dbo.pizzas.pizza_id=dbo.order_details.pizza_id  
group by dbo.pizzas.size  
order by total_count desc;
```

Screenshot of SQL Server Management Studio (SSMS) showing the results of the query. The 'Results' tab is selected, displaying a table with two columns: 'size' and 'total_count'. The data shows a single row where size 'L' has a total count of 18526.

	size	total_count
1	L	18526

Large Sized Pizzas are the most preferred pizza that is 18526 numbers, equal to 87% of the total orders.



- List the 5 most ordered pizza type along with their quantity

```
select top 5 dbo.pizza_types.name,sum(dbo.order_details.quantity) as QUANTITY  
from dbo.pizza_types join  
dbo.pizzas on dbo.pizza_types.pizza_type_id=dbo.pizzas.pizza_type_id  
join dbo.order_details  
on dbo.order_details.pizza_id=dbo.pizzas.pizza_id  
group by dbo.pizza_types.name  
order by QUANTITY desc;
```

Results Messages

	name	QUANTITY
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

Top 5 Best selling pizzas



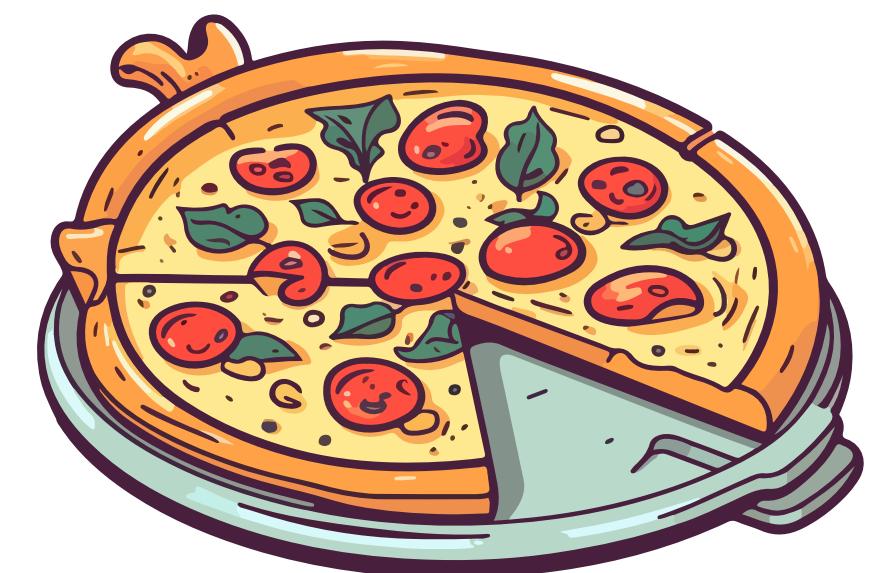
- Join the necessary tables to find the total quantity of each pizza category ordered

```
select dbo.pizza_types.category, sum(dbo.order_details.quantity) As QUANTITY  
from dbo.pizza_types  
join dbo.pizzas on dbo.pizza_types.pizza_type_id=dbo.pizzas.pizza_type_id  
Join dbo.order_details on  
dbo.order_details.pizza_id=dbo.pizzas.pizza_Id  
group by dbo.pizza_types.category  
order by QUANTITY desc;
```



Results

	category	QUANTITY
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050



Classic being the highest selling Pizza category

- Determine the distribution of orders by hours of the day

```
SELECT DATEPART(HOUR, dbo.orders.time) AS hours,  
COUNT(dbo.orders.order_id) AS order_count  
FROM dbo.orders  
GROUP BY DATEPART(HOUR, dbo.orders.time)  
ORDER BY hours;
```

There is high in flow of orders in lunch
(12:00pm-2:00pm) and Snacks(4:00pm-
7:00pm)

	hours	order_count
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009
12	20	1642
13	21	1198
14	22	663
15	23	28

- join the relevant tables find the category wise distribution of pizzas

```
select category, count(name)as pizza_names from dbo.pizza_types  
group by category;
```

Screenshot of a SQL query results window showing the distribution of pizzas by category.

	category	pizza_names
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9

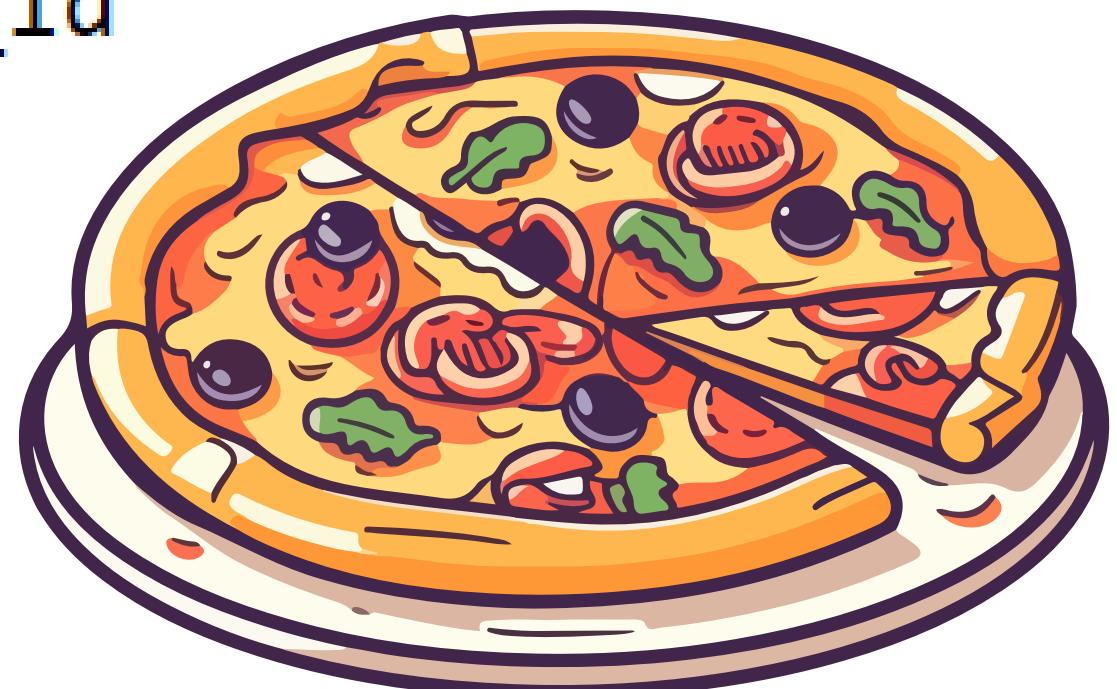


- Group The order by date and calculate the average number of pizzas ordered per day

```
select avg(Quantity) as avg_orders from (select dbo.orders.date,  
sum(dbo.order_details.quantity) as Quantity from  
dbo.orders join  
dbo.order_details on  
dbo.orders.order_id=dbo.order_details.order_id  
group by dbo.orders.date)as order_quantity;
```

Results Messages

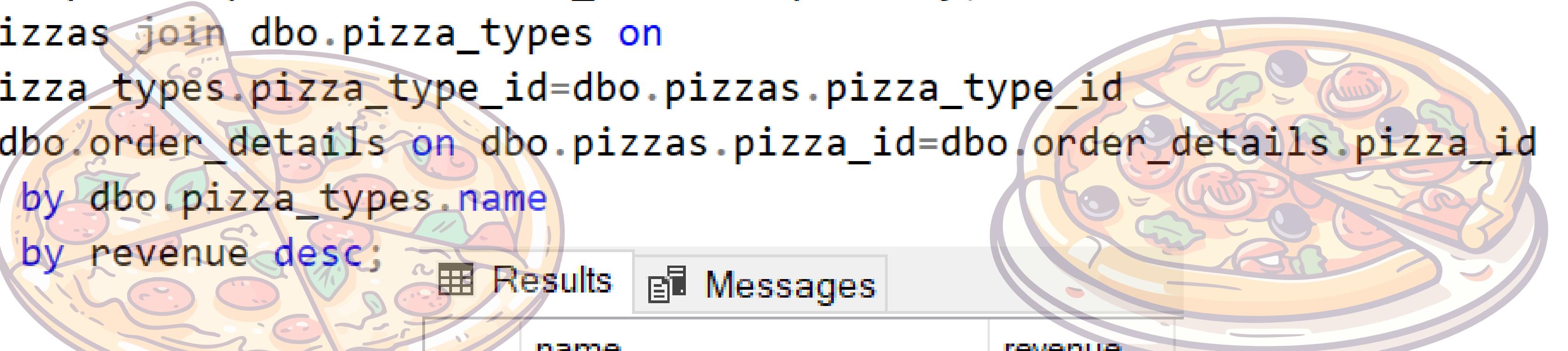
	avg_orders
1	138



On an Average 138 orders are received per day.

- Determine the top 3 most ordered pizza type based on revenue

```
select top 3 dbo.pizza_types.name,  
sum(dbo.pizzas.price*dbo.order_details.quantity) as revenue from  
dbo.pizzas join dbo.pizza_types on  
dbo.pizza_types.pizza_type_id=dbo.pizzas.pizza_type_id  
join dbo.order_details on dbo.pizzas.pizza_id=dbo.order_details.pizza_id  
group by dbo.pizza_types.name  
order by revenue desc;
```

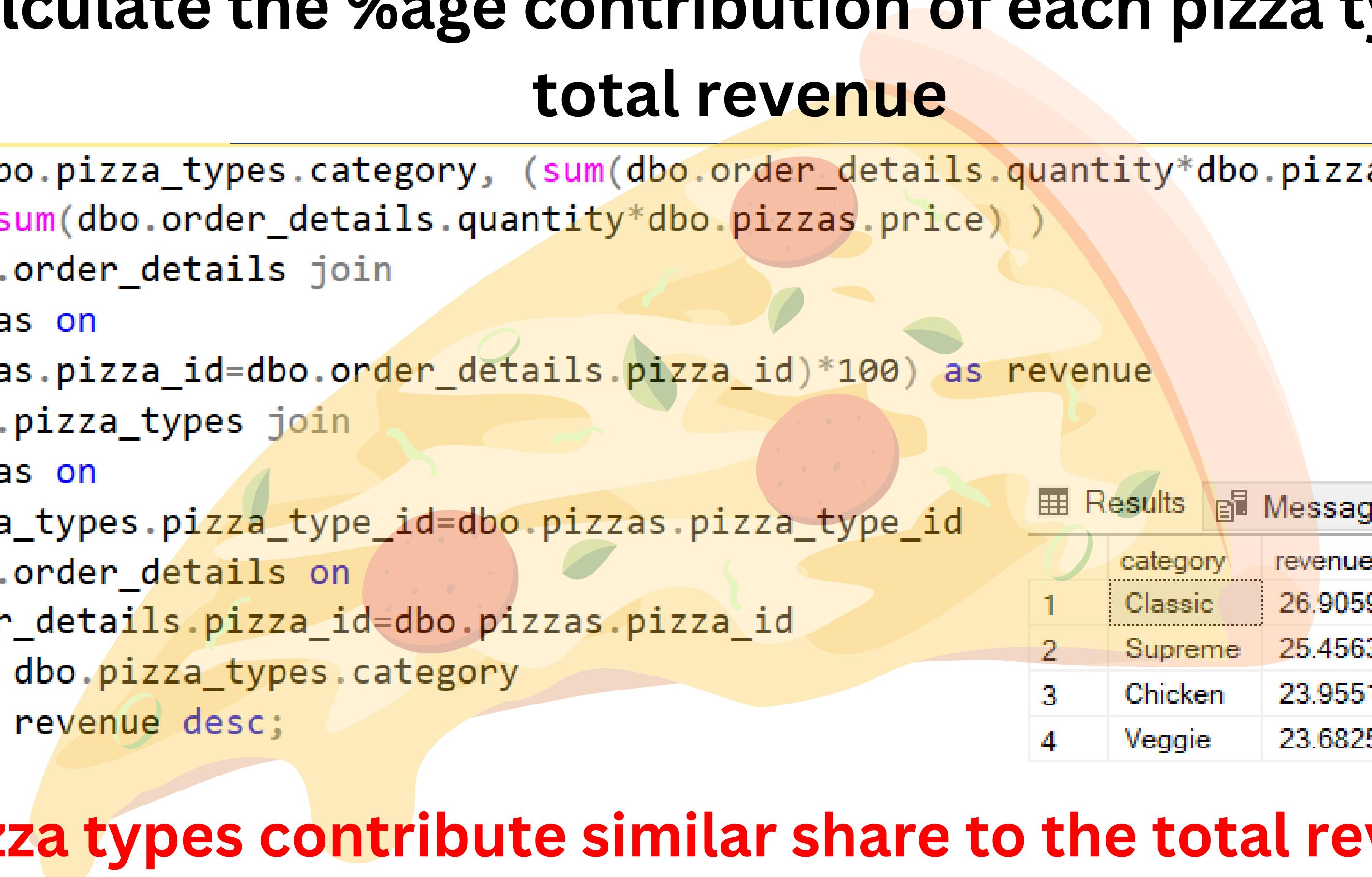


	name	revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

Non Veg(Chicken contributes the maximum share in terms of Revenue

- Calculate the %age contribution of each pizza type to total revenue

```
select dbo.pizza_types.category, (sum(dbo.order_details.quantity*dbo.pizzas.price)/  
(select(sum(dbo.order_details.quantity*dbo.pizzas.price) )  
from dbo.order_details join  
dbo.pizzas on  
dbo.pizzas.pizza_id=dbo.order_details.pizza_id)*100) as revenue  
from dbo.pizza_types join  
dbo.pizzas on  
dbo.pizza_types.pizza_type_id=dbo.pizzas.pizza_type_id  
join dbo.order_details on  
dbo.order_details.pizza_id=dbo.pizzas.pizza_id  
group by dbo.pizza_types.category  
order by revenue desc;
```



	category	revenue
1	Classic	26.9059602306976
2	Supreme	25.4563112111462
3	Chicken	23.9551375322885
4	Veggie	23.6825910258677

All Pizza types contribute similar share to the total revenue

- Analyze the cumulative revenue generated over time

```
SELECT
```

```
    sales.date,  
    SUM(sales.revenue) OVER (ORDER BY sales.date) AS cumm_revenue
```

```
FROM (
```

```
    SELECT
```

```
        dbo.orders.date,  
        SUM(dbo.order_details.quantity * dbo.pizzas.price) AS revenue
```

```
    FROM dbo.order_details
```

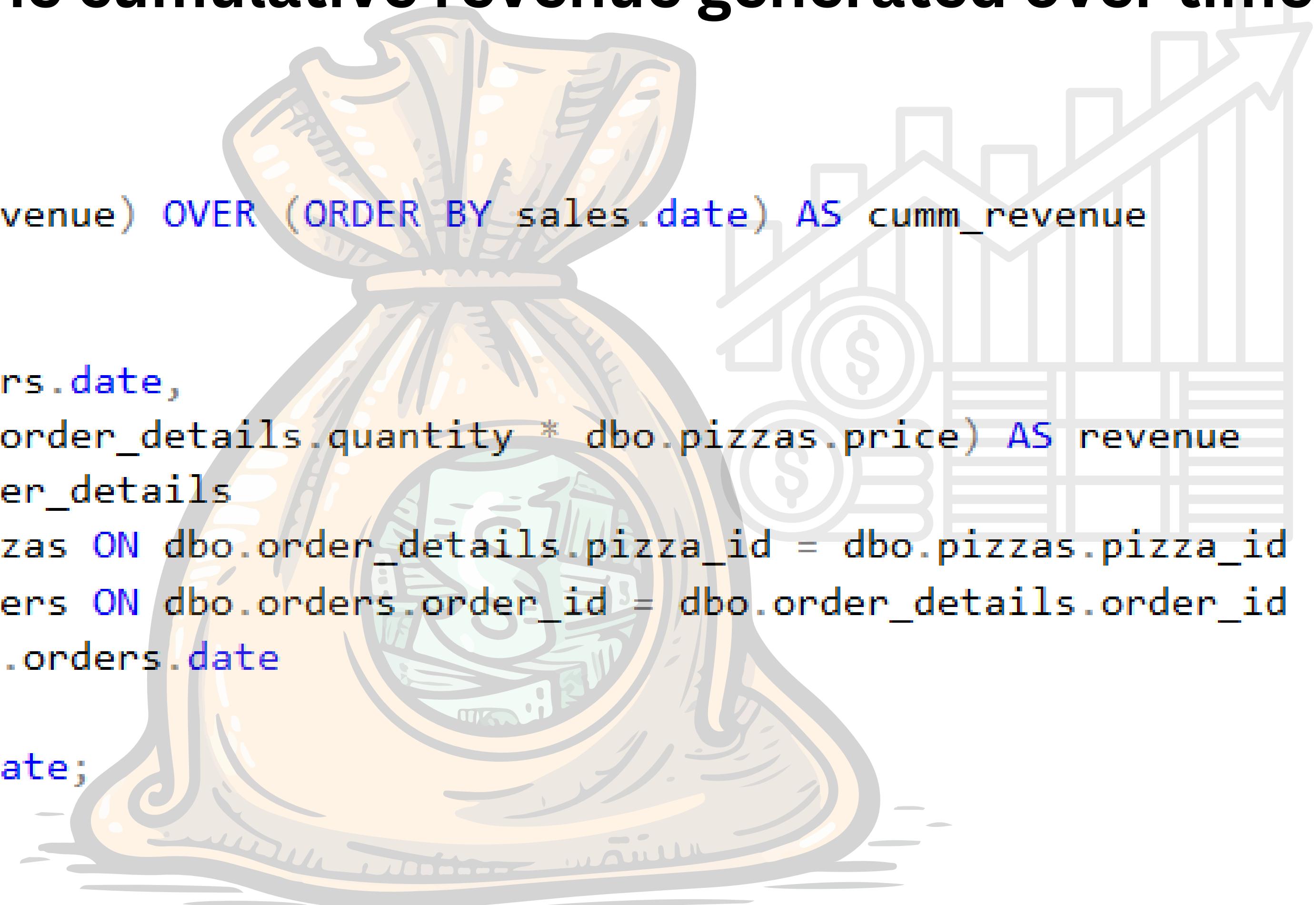
```
    JOIN dbo.pizzas ON dbo.order_details.pizza_id = dbo.pizzas.pizza_id
```

```
    JOIN dbo.orders ON dbo.orders.order_id = dbo.order_details.order_id
```

```
    GROUP BY dbo.orders.date
```

```
) AS sales
```

```
ORDER BY sales.date;
```



- Determine the top 3 most ordered pizza type based on revenue for each pizza category

SELECT

```

category, name, revenue
FROM (SELECT category, name, revenue,
            RANK() OVER (PARTITION BY category ORDER BY revenue DESC) AS rn
      FROM (
        SELECT
            dbo.pizza_types.category,
            dbo.pizza_types.name,
            SUM(dbo.order_details.quantity * dbo.pizzas.price) AS revenue
          FROM dbo.pizza_types
          JOIN dbo.pizzas ON dbo.pizza_types.pizza_type_id = dbo.pizzas.pizza_type_id
          JOIN dbo.order_details ON dbo.order_details.pizza_id = dbo.pizzas.pizza_id
          GROUP BY dbo.pizza_types.category, dbo.pizza_types.name
        ) AS a
    ) AS b
WHERE rn <= 3;

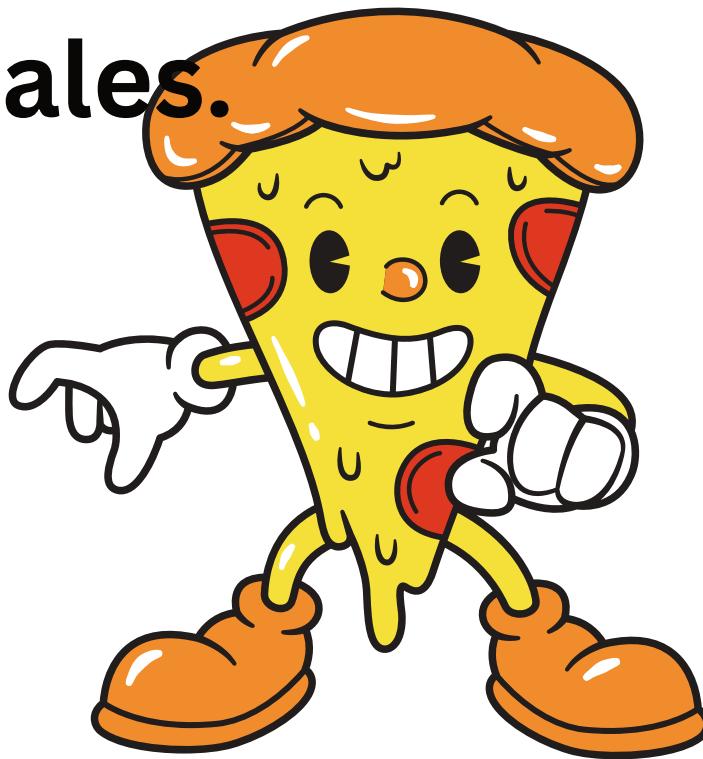
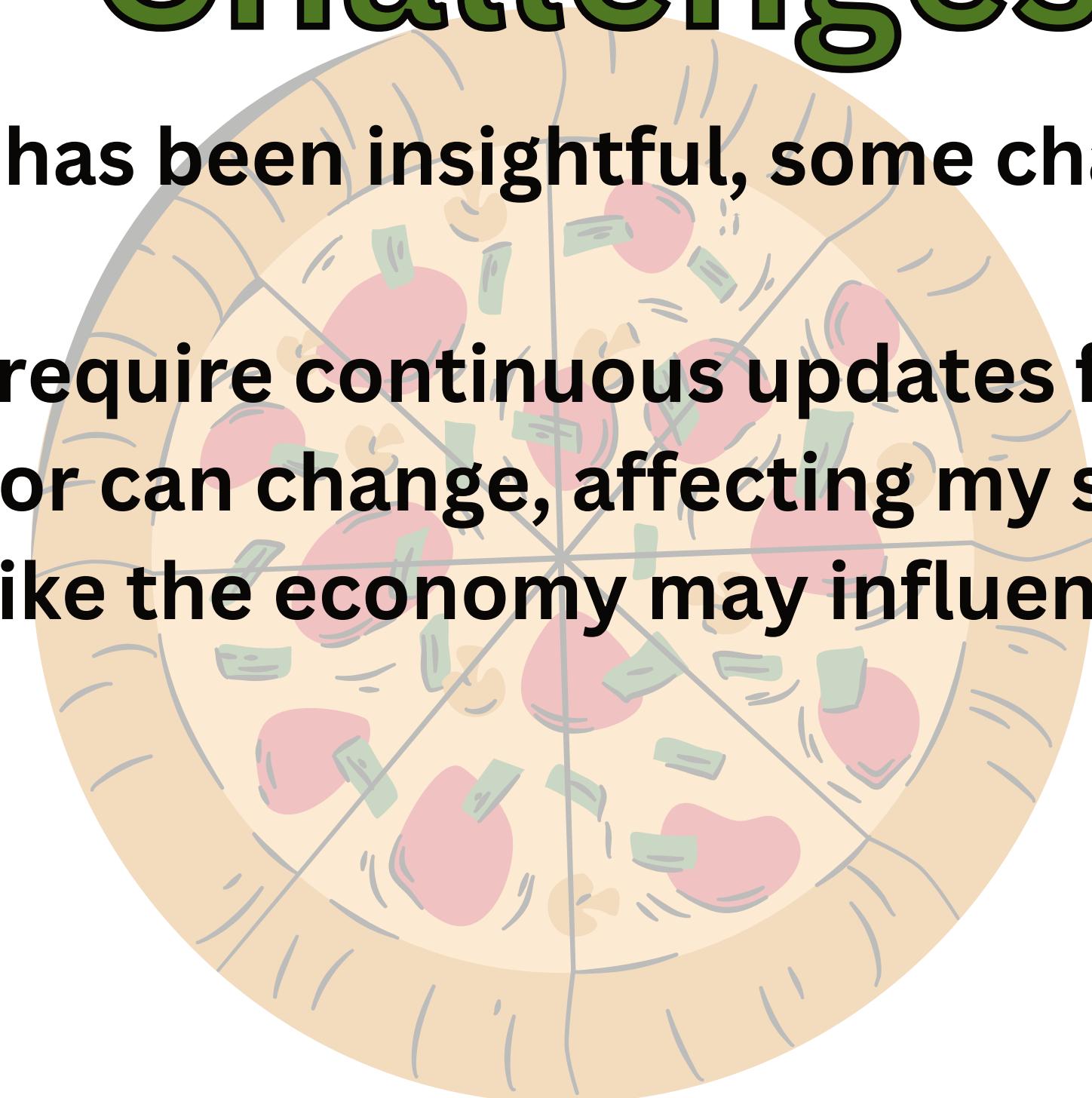
```

	category	name	revenue
1	Chicken	The Thai Chicken Pizza	43434.25
2	Chicken	The Barbecue Chicken Pizza	42768
3	Chicken	The California Chicken Pizza	41409.5
4	Classic	The Classic Deluxe Pizza	38180.5
5	Classic	The Hawaiian Pizza	32273.25
6	Classic	The Pepperoni Pizza	30161.75
7	Supreme	The Spicy Italian Pizza	34831.25
8	Supreme	The Italian Supreme Pizza	33476.75
9	Supreme	The Sicilian Pizza	30940.5
10	Veggie	The Four Cheese Pizza	32265.7010040283
11	Veggie	The Mexicana Pizza	26780.75
12	Veggie	The Five Cheese Pizza	26066.5

Challenges

While the analysis has been insightful, some challenges may arise:

- The dataset may require continuous updates for real-time insights.
- Customer behavior can change, affecting my sales trends.
- External factors like the economy may influence pizza sales.



CONCLUSION

This analysis has unveiled critical insights that can guide my business decisions. By understanding my sales data and customer preferences, I can optimize my offerings and enhance my growth.

All data and files are available on this link

#PizzaSales #DataAnalysis #BusinessGrowth #PizzaBusiness 🍕📈

