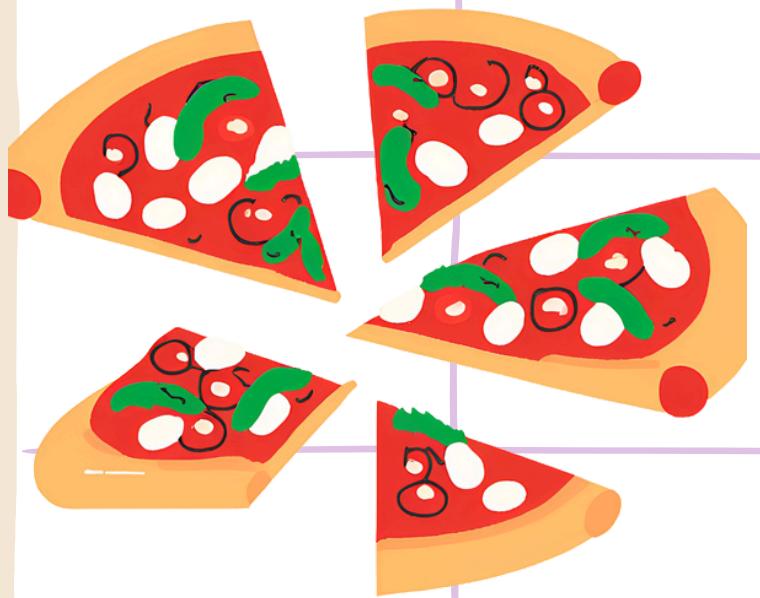


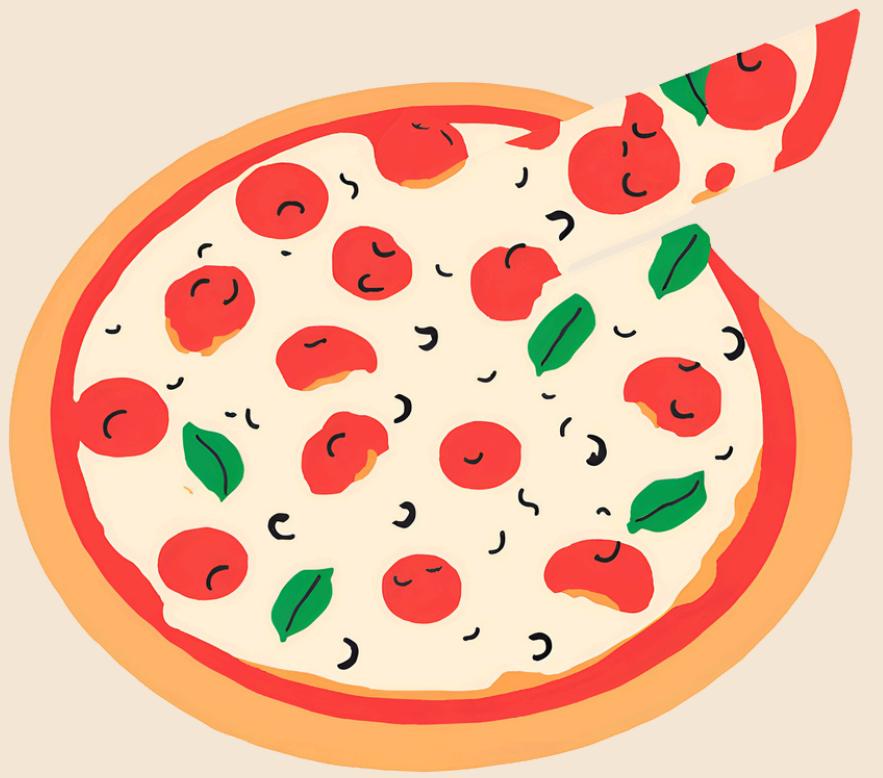
SQL Project on Pizza Sales



Presented by Suraj Kumar

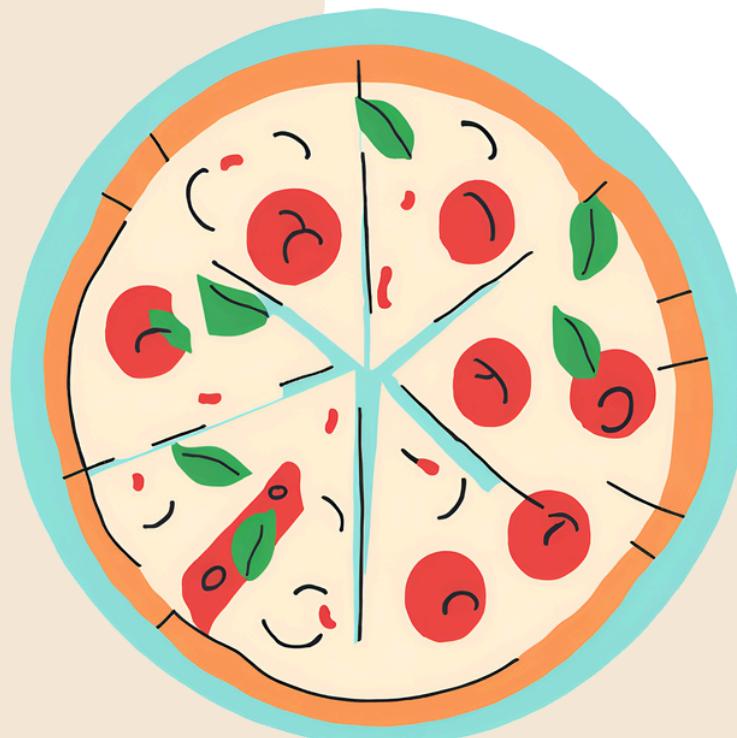


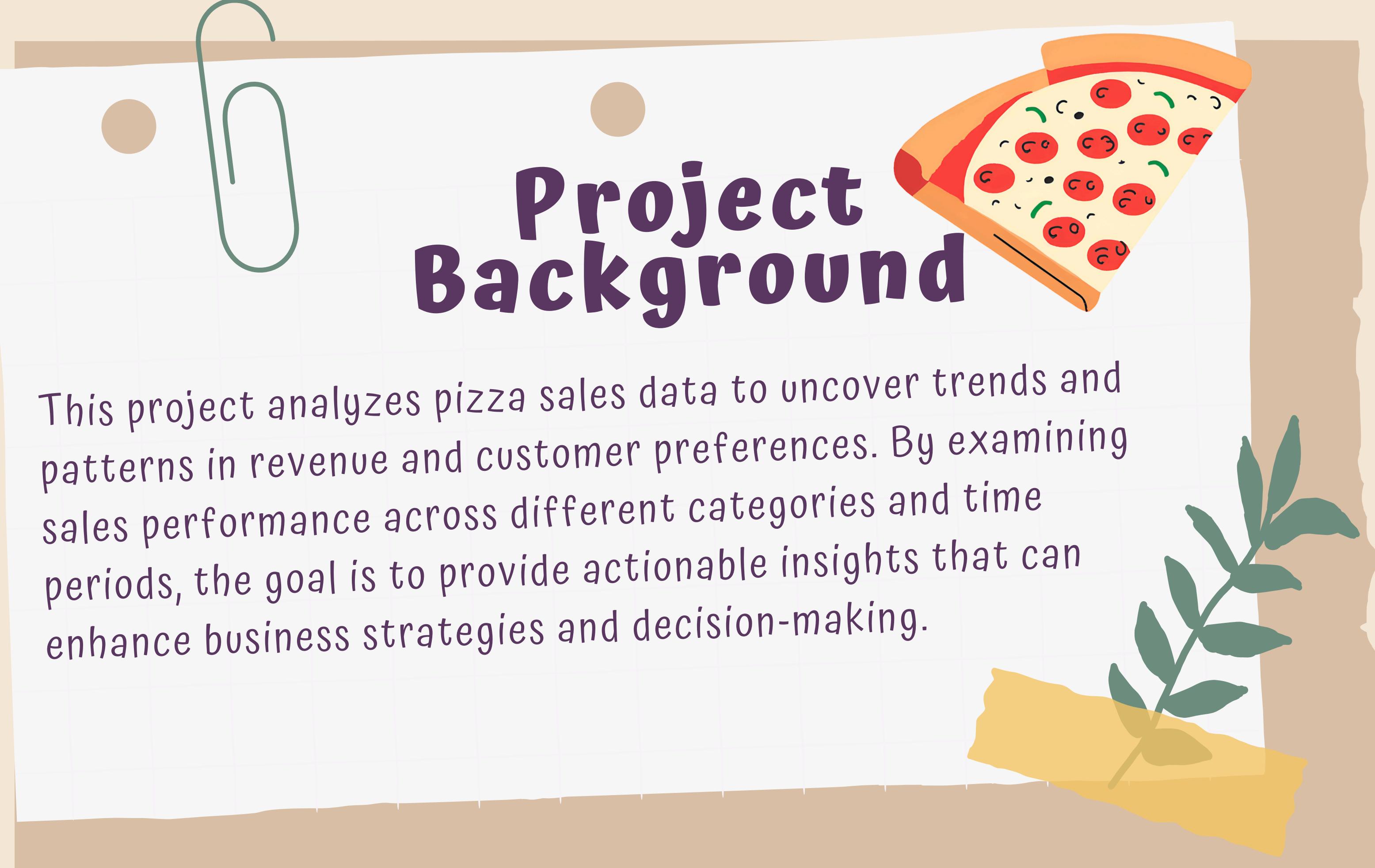
Hello Everyone!



Welcome!

I'm Suraj Kumar, and I'm diving deep into the world of data analytics. Today, I'm excited to present my inaugural SQL project, where I analyzed pizza sales data to uncover fascinating trends and patterns. Through this exploration, I've discovered intriguing insights about customer preferences, sales performance, and market dynamics. Get ready to see how data can slice through complexity to reveal actionable business intelligence!



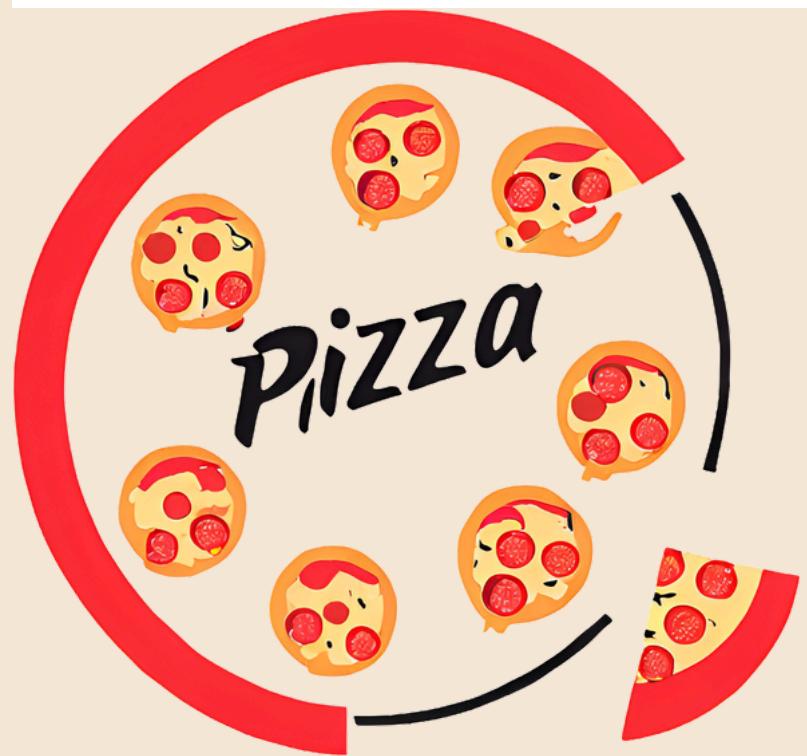


Project Background

This project analyzes pizza sales data to uncover trends and patterns in revenue and customer preferences. By examining sales performance across different categories and time periods, the goal is to provide actionable insights that can enhance business strategies and decision-making.

Retrieve the total number of orders placed.

```
SELECT  
    COUNT(order_id) AS Total_Orders  
FROM  
    Orders;
```



Result Grid	
	Total_Orders
▶	21350



Calculate the total revenue generated from pizza sales.

- **SELECT**

```
ROUND(SUM(orders_details.quantity * pizzas.price),  
      2) AS Total_Revenue  
  
FROM  
    orders_details  
  JOIN  
    pizzas ON orders_details.pizza_id = pizzas.pizza_id;
```



Result Grid	
	Total_Revenue
▶	817860.05



Identify the highest-priced pizza.

SELECT

```
    pizza_types.name as Highest_Price_Pizza, pizzas.price
```

FROM

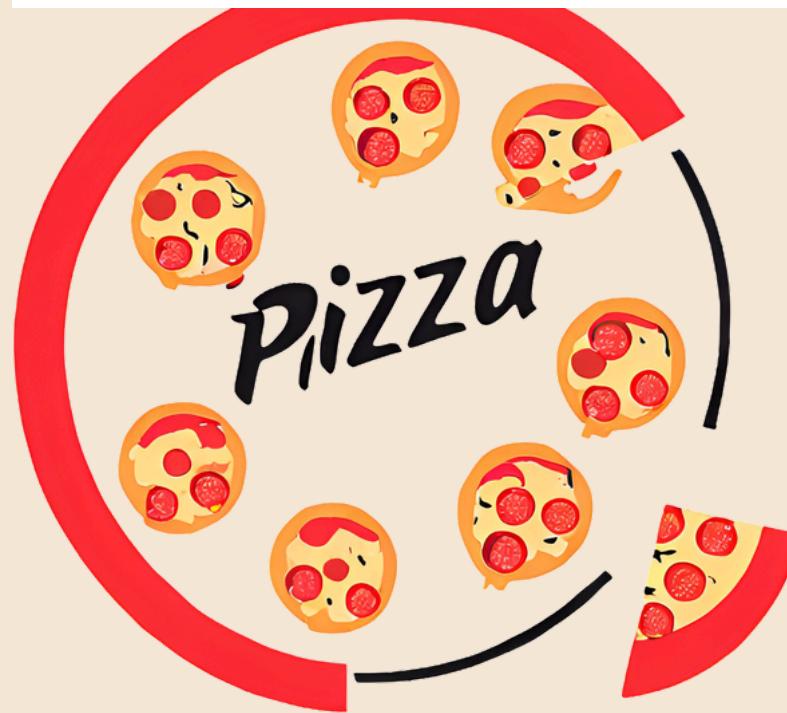
```
    pizza_types
```

JOIN

```
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
```

ORDER BY pizzas.price DESC

LIMIT 1;



Result Grid | Filter Rows:

	Highest_Price_Pizza	price
▶	The Greek Pizza	35.95



Identify the most common pizza size ordered.

```
SELECT  
    pizzas.size AS Most_Common_Pizza_Size,  
    SUM(orders_details.quantity) AS Total_Orders  
FROM  
    pizzas  
        JOIN  
    orders_details ON orders_details.pizza_id = pizzas.pizza_id  
GROUP BY pizzas.size  
ORDER BY SUM(orders_details.quantity) DESC  
LIMIT 1;
```



Result Grid | Filter Rows:

	Most_Common_Pizza_Size	Total_Orders
▶	L	18956

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

SELECT

```
    pizza_types.name AS Most_Ordered_Pizzas,  
    SUM(orders_details.quantity) AS Overall_Quantity  
  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
        JOIN  
    orders_details ON pizzas.pizza_id = orders_details.pizza_id  
  
GROUP BY pizza_types.name  
  
ORDER BY SUM(orders_details.quantity) DESC  
  
LIMIT 5;
```



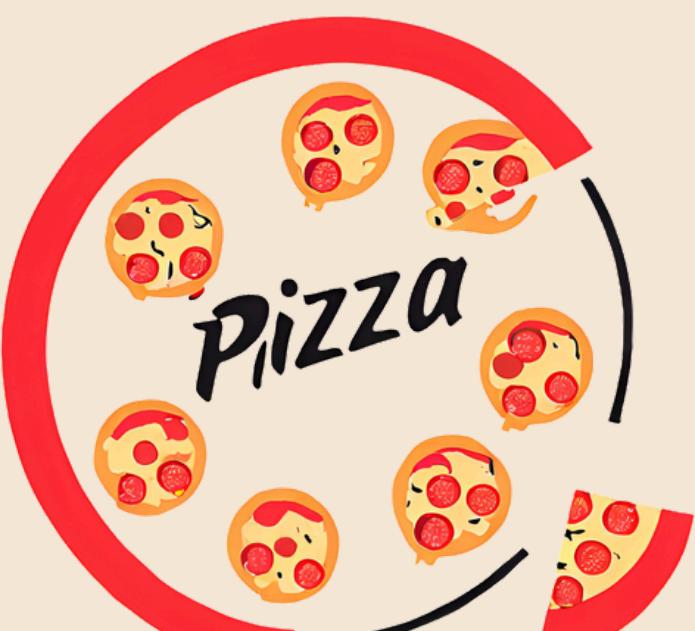
	Most_Ordered_Pizzas	Overall_Quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

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

```
SELECT  
    pizza_types.category,  
    SUM(orders_details.quantity) AS Total_QTY  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
        JOIN  
    orders_details ON pizzas.pizza_id = orders_details.pizza_id  
GROUP BY pizza_types.category  
ORDER BY Total_QTY DESC
```

Result Grid | Filter

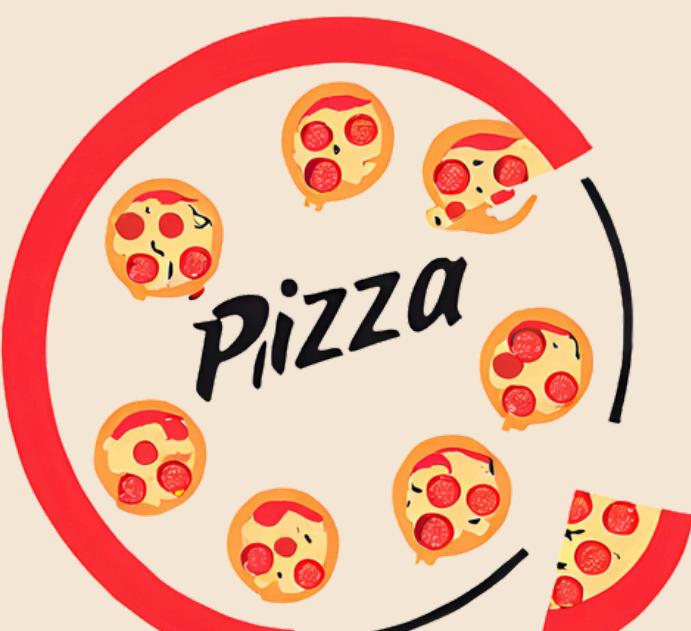
	category	Total_QTY
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



Determine the distribution of orders by hour of the day.

```
SELECT  
    HOUR(orders.order_time) as Hour, COUNT(orders.order_id) as Distribution  
FROM  
    orders  
GROUP BY HOUR(orders.order_time)
```

	Hour	Distribution
▶	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.

SELECT

Category, COUNT(name) Pizza

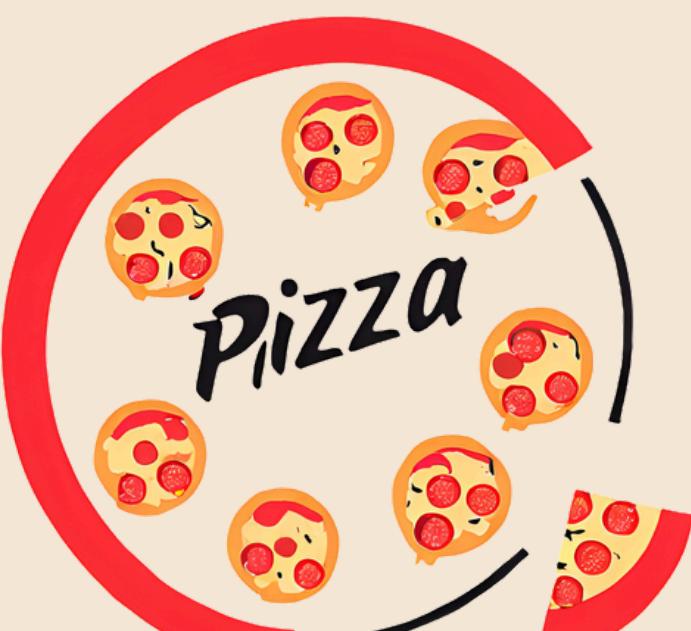
FROM

pizza_types

GROUP BY category;

Result Grid

	Category	Pizza
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

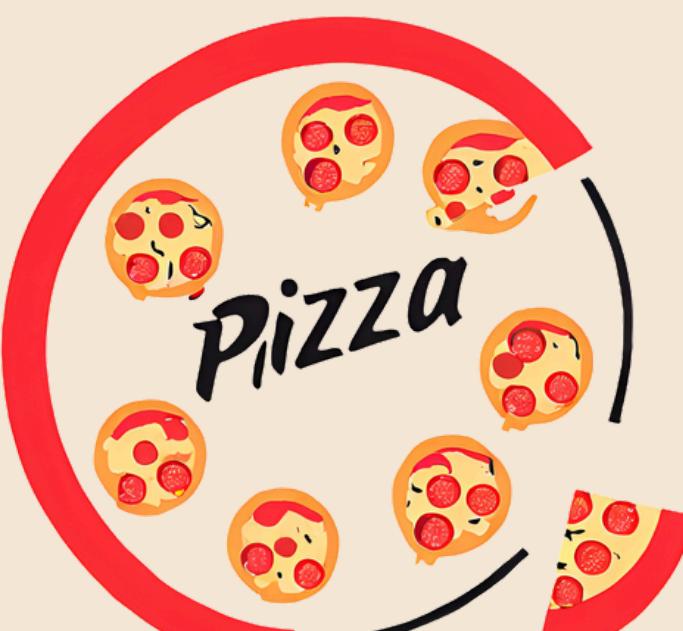


Group the orders by date and calculate the average number of pizzas ordered per day.

```
WITH Order_QTY as (SELECT  
  
    SELECT  
        ROUND(AVG(QTY), 0) AS AVG_Orders_Per_Day  
  
    FROM  
        Order_QTY
```

Result Grid | Filter Rows

	AVG_Orders_Per_Day
▶	138

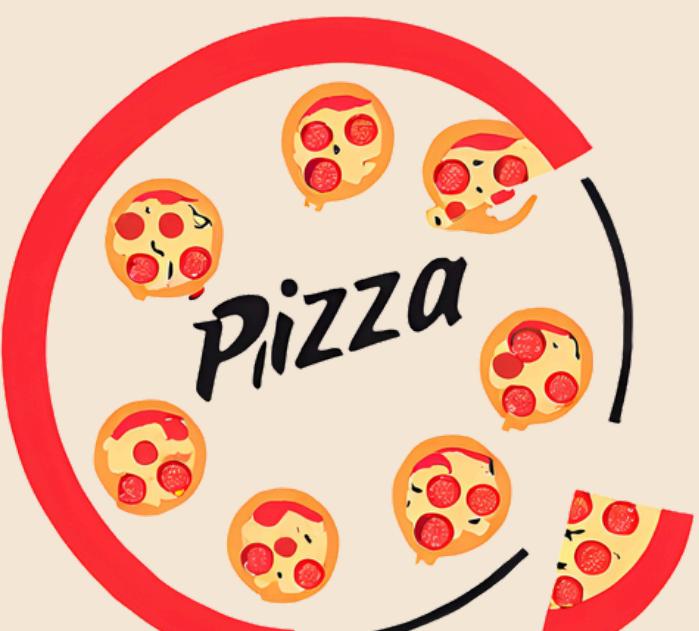


Determine the top 3 most ordered pizza types based on revenue.

```
SELECT  
    pizza_types.Name,  
    ROUND(SUM(orders_details.quantity * pizzas.price), 2) AS Total_Revenue  
FROM  
    orders_details  
    JOIN pizzas ON orders_details.pizza_id = pizzas.pizza_id  
    JOIN pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
GROUP BY  
    pizza_types.name  
ORDER BY  
    Total_Revenue DESC  
LIMIT 3;
```

Result Grid | Filter Rows:

	Name	Total_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 category to total revenue.

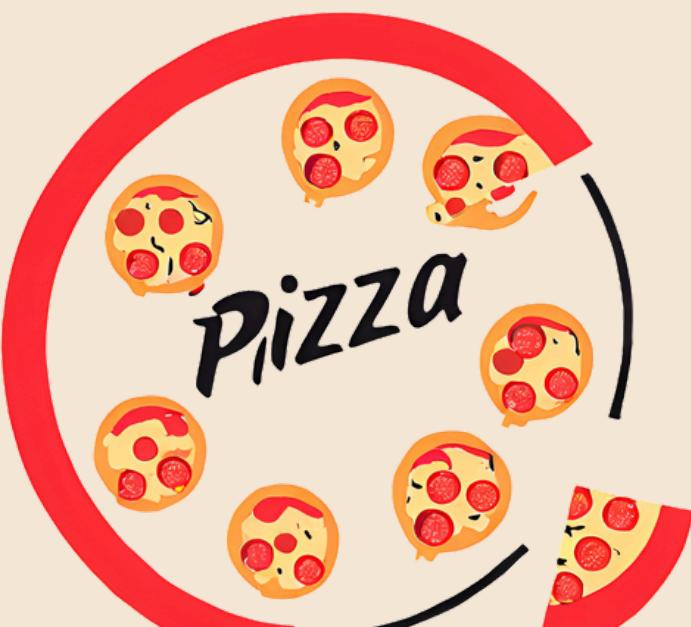
```
with A as (SELECT
```

```
    SUM(orders_details.quantity * pizzas.price) as Total_Revenue  
FROM  
    orders_details  
    JOIN  
    pizzas ON orders_details.pizza_id = pizzas.pizza_id)
```

	category	Revenue in %
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

```
with A as (SELECT
```

```
SELECT  
    pizza_types.category,  
    ROUND(SUM(orders_details.quantity * pizzas.price) * 100 / (select Total_Revenue from A),2) AS 'Revenue in %'  
FROM  
    orders_details  
    JOIN  
    pizzas ON orders_details.pizza_id = pizzas.pizza_id  
    JOIN  
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
GROUP BY pizza_types.category  
ORDER BY 'Revenue in %' DESC
```



Analyze the cumulative revenue generated over time.

```
SELECT
```

```
    order_date,  
    SUM(Revenue) OVER (ORDER BY order_date) AS cumulative_revenue
```

```
) FROM (
```

```
    SELECT  
        orders.order_date,  
        SUM(orders_details.quantity * pizzas.price) AS Revenue
```

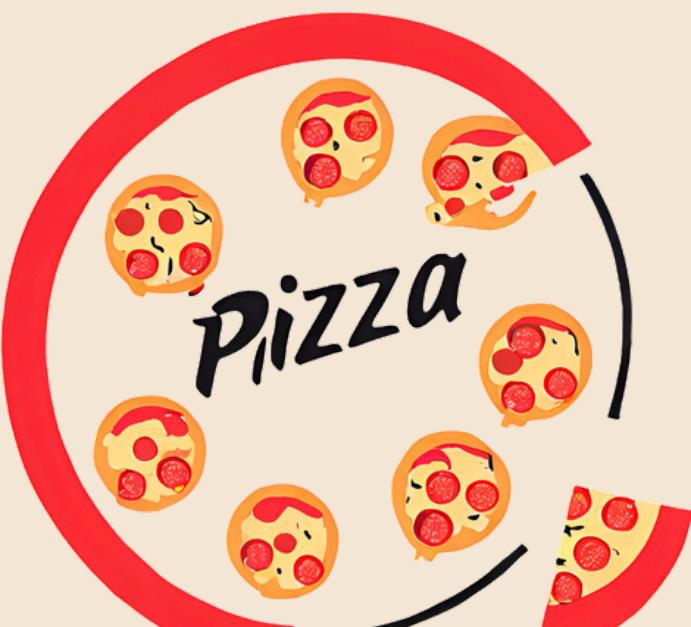
```
    FROM
```

```
        orders_details  
        JOIN pizzas ON orders_details.pizza_id = pizzas.pizza_id  
        JOIN orders ON orders.order_id = orders_details.order_id
```

```
    GROUP BY
```

```
        orders.order_date
```

```
) AS sales;
```



Result Grid | Filter Rows:

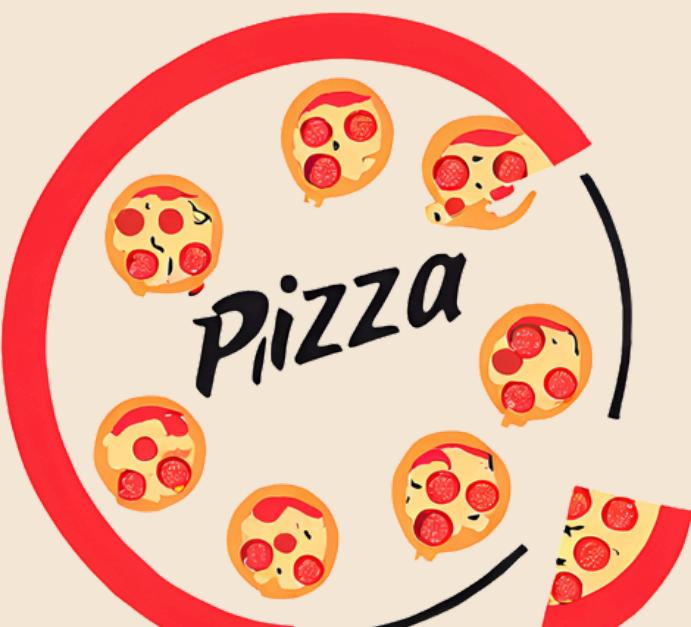
	order_date	cumulative_revenue
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015-01-06	14358.5



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

```
• SELECT
  Ranking,
  Category,
  Name,
  Revenue
  FROM (
    SELECT
      category,
      name,
      Revenue,
      RANK() OVER (PARTITION BY category
      ORDER BY Revenue DESC) AS Ranking
    FROM (
      SELECT
        pizza_types.category,
        pizza_types.name,
```

```
  pizza_types.name,
  SUM(orders_details.quantity
  * pizzas.price) AS Revenue
  FROM
  pizzas
  JOIN orders_details ON pizzas.pizza_id = orders_details.pizza_id
  JOIN pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
  GROUP BY
  pizza_types.category,
  pizza_types.name
  ORDER BY
  pizza_types.category,
  Revenue DESC
  ) AS A
  ) AS B WHERE Ranking <= 3;
```



	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	30940.5
	Veggie	The Four Cheese Pizza	32265.700000000004
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5



Analysis Highlights

Pizzz hut

1.

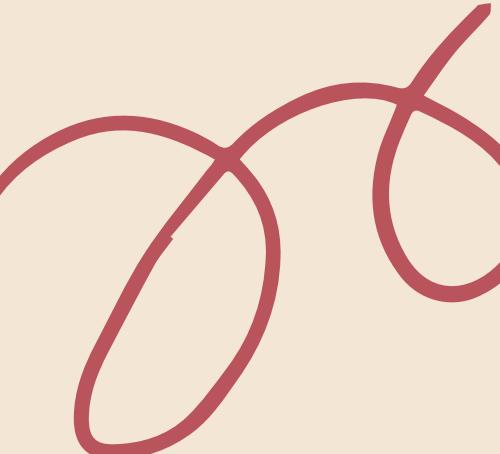
Top Revenue-
Generating Pizza
Category:
CLASSIC

2.

Top Revenue
Generating Hour :
12 PM

3.

Top Revenue-
Generating Pizzas:
'The Thai Chicken Pizza'



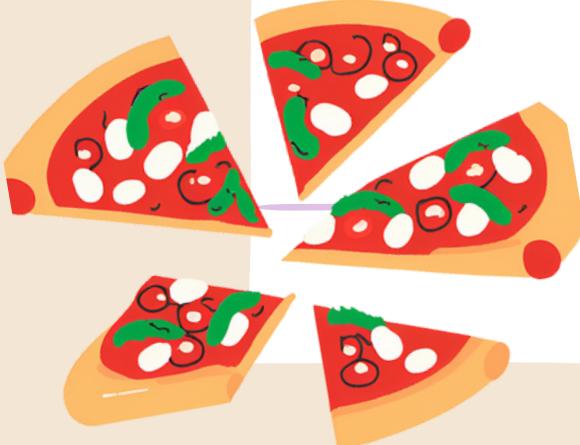
Final Thoughts

This analysis revealed critical insights into our pizza sales, highlighting top-performing pizzas, peak sales hours, and profitable categories. These findings provide a foundation for data-driven decisions to optimize inventory, enhance marketing strategies, and ultimately boost sales performance. Moving forward, regular data analysis will be key to maintaining and improving our competitive edge.





Open
for Questions



Appreciate
Your Attention

