



Project Announcement: **Pizza Sales Analysis**

I'm excited to share my latest project, where I deep-dived into pizza sales data to extract valuable insights.



Nahidul Islam

Junior Data Analyst | Data Science Enthusiast

Pizza Sales Analysis Project Overview:

In this project, I explored pizza sales data to uncover valuable insights:

Total Orders & Revenue:

Determined the total orders placed and revenue generated.

Top Pizzas:

Identified the highest-priced pizza, most popular size, and top 5 ordered pizzas

Advanced Analytics:

Performed category-wise distribution, cumulative revenue analysis, and identified the top 3 pizzas by revenue for each category.

Retrieve the total number of orders placed.

```
-- Retrieve the total number of orders placed.  
  
SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    orders;
```

	total_orders
▶	21350

Calculate the total revenue generated from pizza sales.

```
-- 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 pizzas.pizza_id = order_details.pizza_id
```

	total_revenue
▶	817860.05

Identify the highest-priced pizza.

```
-- 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 price DESC  
LIMIT 1;
```

Result Grid			Filter Row
	name	price	
▶	The Greek Pizza	35.95	

Identify the most common pizza size ordered

```
-- Identify the most common pizza size ordered.--  
SELECT  
    pizzas.size,  
    COUNT(order_details.order_details_id) AS order_count  
FROM  
    pizzas  
        JOIN  
        order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY size  
ORDER BY order_count DESC;
```

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

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

```
-- List the top 5 most ordered pizza types along with their quantities.  
SELECT  
    pizza_types.name,  
    SUM(order_details.quantity) AS Top_5_most_ordered_pizza  
FROM  
    pizza_types  
    JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
    JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY name  
ORDER BY Top_5_most_ordered_pizza DESC  
LIMIT 5;
```

	name	Top_5_most_ordered_pizza
►	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

```
-- Join the necessary tables to find the total quantity
-- of each pizza category ordered.
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
    JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY category
ORDER BY quantity DESC;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day

```
-- Determine the distribution of orders by hour of the day.  
  
SELECT  
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

	hour(order_time)	count(order_id)
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8

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

```
-- Join relevant tables to find the
-- category-wise distribution of pizzas.
SELECT
    category, COUNT(name) AS Pizzas_type
FROM
    pizza_types
GROUP BY category
ORDER BY pizzas_type DESC;
```

	category	Pizzas_type
▶	Supreme	9
	Veggie	9
	Classic	8
	Chicken	6

Determine the top 3 most ordered pizza types based on revenue



```
-- 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 pizzas.pizza_id = order_details.pizza_id
GROUP BY name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid			Filter Rows:	
	name	revenue		
	The Thai Chicken Pizza	43434.25		
	The Barbecue Chicken Pizza	42768		
▶	The California Chicken Pizza	41409.5		

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

```
-- Group the orders by date and calculate the average number of pizzas ordered per day.  
SELECT  
    round(AVG(total_quantity),0 )AS avg_pizza_ordered_per_day  
FROM  
    (SELECT  
        orders.order_date,  
        SUM(order_details.quantity) AS total_quantity  
    FROM  
        orders  
    JOIN order_details ON orders.order_id = order_details.order_id  
    GROUP BY order_date) AS Per_day;
```

Result Grid			 Filter Rows:
	avg_pizza_ordered_per_day		
▶	138		



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

```
-- 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_revenue
    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 category;
```

Result Grid			Filter Rows
	category	revenue	
▶	Classic	26.91	
	Veggie	23.68	
	Supreme	25.46	
	Chicken	23.96	

Analyze the cumulative revenue generated over time

```
-- 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(order_details.quantity * pizzas.price) AS 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 sales
ORDER BY
  order_date;
```

Result Grid   Filter Rows: <input type="text"/>		
	order_date	cumulative_revenue
	2015-01-13	29831.3000000000003
	2015-01-14	32358.7000000000004
	2015-01-15	34343.5000000000001
	2015-01-16	36937.6500000000001
	2015-01-17	39001.7500000000001

Retrieve top 3 pizzas by revenue for each category

```
-- Retrieve top 3 pizzas by revenue for each category
SELECT
    name,
    revenue
FROM
    (SELECT
        pizza_types.category,
        pizza_types.name,
        SUM(order_details.quantity * pizzas.price) AS revenue,
        RANK() OVER(PARTITION BY pizza_types.category ORDER BY SUM(order_details.quantity * pizzas.price) DESC) AS rn
    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 ranked_pizzas
WHERE
    rn <= 3;
```

Result Grid			Filter Rows:	Exp
	name	revenue		
	The California Chicken Pizza	41409.5		
	The Classic Deluxe Pizza	38180.5		
	The Hawaiian Pizza	32273.25		
	The Pepperoni Pizza	30161.75		
	The Spicy Italian Pizza	34831.25		

Result 5 ×