

PIELOGY SALES ANALYSIS

Objective :- This project will show you how to analyze sales of Pizza company. I have examined the dataset with SQL and helped the company understand its business growth by answering simple and complex questions based on , total revenue, Most sold pizza types ,sales distribution across the hours,average pizza sales for each day. This analysis will provide great insights to the Company to make the right and effective decisions

Database and Tools Used :- MySQL

Concepts used for this Project:

- 1.Aggregation functions like sum, average, count
- 2.Window functions
- 3.Sub-queries
4. Joins: join, Left join, Inner join

1.To get the total number of orders

```
-- To get the total number of orders--  
SELECT  
    COUNT(order_id)  
FROM  
    orders;
```

Result:-

Result Grid		Filter Rows:
	COUNT(order_id)	
▶	21350	

2.To get total number of pizza sales

```
-- To get toatl number of pizza sales --  
  
SELECT  
    SUM(quantity) AS Total_pizza_sales  
FROM  
    order_details;
```

Result:-

Result Grid		Filter Rows:
	Total_pizza_sales	
▶	49574	

3.To calculate total revenue

```
-- To caluculate total revenue--  
  
SELECT  
    ROUND(SUM(quantity * price), 2) AS revenue  
FROM  
    order_details  
    LEFT JOIN  
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

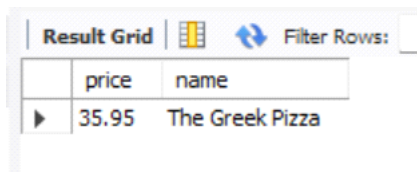
Result:-

Result Grid		Filter Rows:
	revenue	
▶	817860.05	

4.To get the most expensive pizza

```
-- To get the most expensive pizza--  
  
SELECT  
    price, pizza_types.name  
FROM  
    pizzas  
    JOIN  
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
ORDER BY price DESC  
LIMIT 1;
```

Result:-



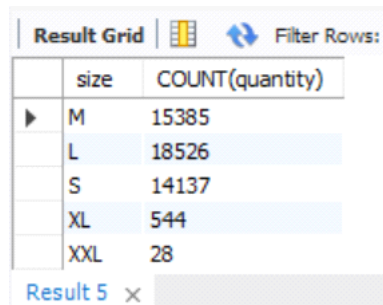
The screenshot shows a 'Result Grid' window with a table containing two columns: 'price' and 'name'. The first row of data shows a price of 35.95 and the name 'The Greek Pizza'.

	price	name
▶	35.95	The Greek Pizza

5.To get top 5 commonly ordered pizza size

```
-- To get top 5 commonly ordered pizza size--  
  
SELECT  
    size, COUNT(quantity)  
FROM  
    pizzas  
    JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY size;
```

Result:-



A screenshot of a database application's 'Result Grid'. The grid has two columns: 'size' and 'COUNT(quantity)'. It contains five rows of data. The first row is highlighted with a mouse cursor. Above the grid, there are icons for 'Result Grid', a grid icon, and a 'Filter Rows' button. Below the grid, it says 'Result 5' with a close button.

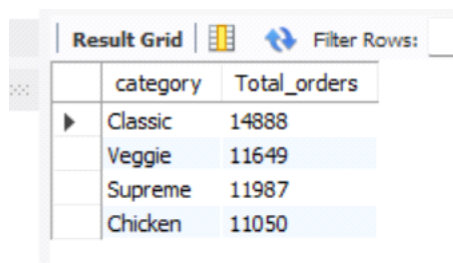
	size	COUNT(quantity)
▶	M	15385
	L	18526
	S	14137
	XL	544
	XXL	28

Result 5 ×

6.To get most number of orders category in pizza

```
-- To get most number of orders category in pizza --  
  
SELECT  
    category, SUM(order_details.quantity) AS Total_orders  
FROM  
    pizzas  
    JOIN  
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
    JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizza_types.category;
```

Result:-



A screenshot of a database application's 'Result Grid'. The grid has two columns: 'category' and 'Total_orders'. It contains four rows of data. The first row is highlighted with a mouse cursor. Above the grid, there are icons for 'Result Grid', a grid icon, and a 'Filter Rows' button. Below the grid, there is a close button.

	category	Total_orders
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

7.To know Hourly distribution of sales

-- To know Which Hour has most number of sales --

```
SELECT
    HOUR(order_time) AS x, COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time);
```

Result:-

Result Grid			Filter Rows:
	x	COUNT(order_id)	
▶	11	1231	
	12	2520	
	13	2455	
	14	1472	
	15	1468	
	16	1920	
	17	2336	
	18	2399	
	19	2009	
	20	1642	

Result Grid			Filter Rows:
	x	COUNT(order_id)	
	21	1198	
	22	663	
	23	28	
	10	8	
	9	1	

Result 7 x

8.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(sold), 0) AS daily_avg
FROM
    (SELECT
        order_date AS z, SUM(quantity) AS sold
    FROM
        order_details
    JOIN orders ON orders.order_id = order_details.order_id
    GROUP BY z
    ORDER BY sold DESC) AS orders_quantity;
```

Result:-





daily_avg
138

9.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)
    FROM
        order_details
    JOIN
        pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
    2) AS revenue
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 pizza_types.category
ORDER BY revenue DESC;
```

Result:-

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