This project focuses on analyzing a comprehensive pizza sales dataset to uncover key business insights and support data-driven decision-making. Using advanced SQL queries, I explored and answered critical business questions such as identifying top-selling pizzas, analyzing revenue contribution by pizza category and size, tracking sales trends over time, and evaluating customer order patterns.

PIZZA SALES ANALYSIS

CHEKE NOW

Mrunmay Kanade

Q.1 RETRIVE THE TOTAL NUMBER OF ORDERS ?

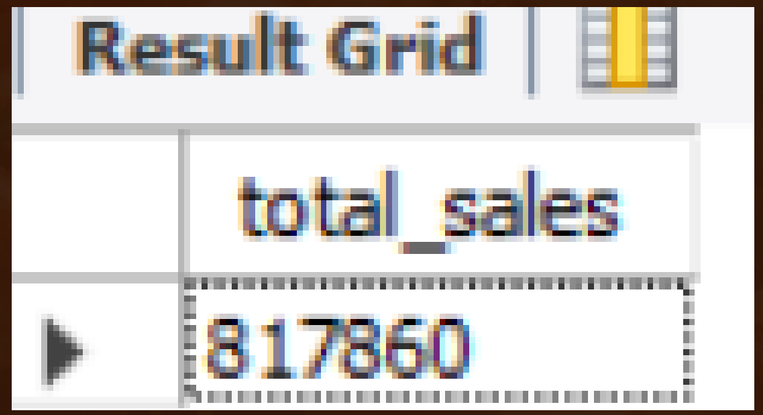
```
-- Retrieve the total number of orders placed.

SELECT
    COUNT(Order_Id) AS total_Orders
FROM
    orders;
```

Result Grid

total_Orders
21350

Q.2 CALCULATE THE TOTAL REVENUE FROM SALES ?

```sql
-- Calculate the total revenue generated from pizza sales.
SELECT
    ROUND(SUM(orders_details.qunatity * pizzas.price)) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.Pizza_Id;
```



IDENTIFY THE HIGHEST PRICED PIZZA?

```sql
-- 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 Rows:

name	price
The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED ?

```sql
-- Identify the most common pizza size ordered.

SELECT
    pizzas.size,
    COUNT(orders_details.Order_details_Id) AS Order_Count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.Pizza_Id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

Result Grid | Filter Rows:

size	Order_Count
L	18526
M	15385
S	14137
XL	544
XXL	28

WHAT IS COUNT OF PIZZA BY NAME?

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

Result Grid | Filter Rows:

name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

WHAT IS COUNT OF PIZZAS BY CATEGORY?

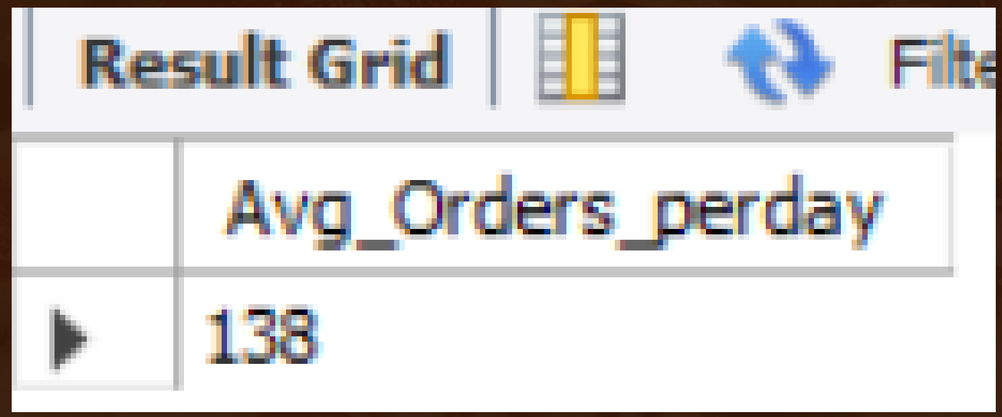
```
    pizza_types.category,
    SUM(orders_details.qunatity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.Pizza_Id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;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

Result Grid | Filter R

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

WHAT IS AVERAGE NUMBER OF PIZZAS ORDER PER DAY?

```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.
SELECT
    ROUND(AVG(quantity), 0) as Avg_Orders_perday
FROM
    (SELECT
        orders.order_date, SUM(orders_details.qunatity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.Order_Id = orders_details.order_Id
    GROUP BY orders.order_date) AS order_Quantity;
```



DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES ?

```sql
-- Determine the top 3 most ordered pizza types based on revenue.
SELECT
    pizza_types.name,
    SUM(orders_details.qunatity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.Pizza_Id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Save the script to a file.

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Pizza Sales Analysis Presentation

THANK YOU FOR ATTENTION

The analysis highlights:

- Sales performance metrics: Total revenue, average order value, and peak order times
- Top & least-selling pizzas: Based on both revenue and quantity sold.
- Category & size contribution: Share of revenue and sales volume across pizza categories and sizes.
- Time-based trends: Daily, weekly, and monthly sales patterns to identify demand fluctuations.

https://Mrunmay08.github.io