Where Every Slice is a Taste of Perfection



**Start Your Slide** 



# HELLO!

My Name is Aniket Sanjay Mistry





In this project i have used SQL queries to solve questions related to pizza sales

# QUERIES TO BE SOLVED IN THIS ::: PROJECT ...

- 1. Retrieve the total number of orders placed.
- 2. Calculate the total revenue generated from pizza sales.
- 3. Identify the highest-priced pizza.
- 4. Identify the most common pizza size ordered.
- 5. List the top 5 most ordered pizza types along with their quantities.

### Intermediate:

- 6. Join the necessary tables to find the total quantity of each pizza category ordered.
- 7. Determine the distribution of orders by hour of the day.
- 8. Join relevant tables to find the category-wise distribution of pizzas.
- 9. Group the orders by date and calculate the average number of pizzas ordered per day.
- 10. Determine the top 3 most ordered pizza types based on revenue.

### Advanced:

- 11. Calculate the percentage contribution of each pizza type to total revenue.
- 12. Analyze the cumulative revenue generated over time.
- 13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES



```
SELECT

ROUND(SUM(order_details.quantity * pizzas.price),

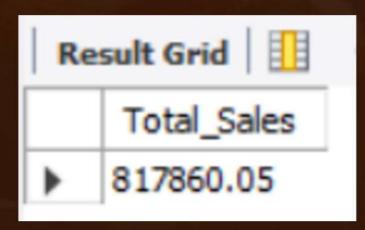
2) AS Total_Sales

FROM

order_details

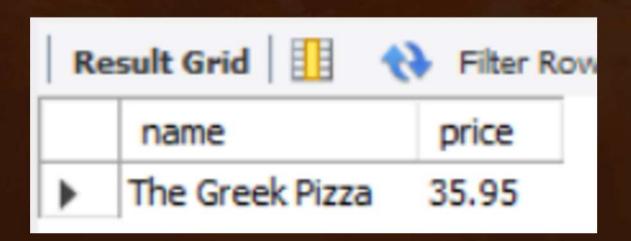
JOIN

pizzas ON pizzas.pizza_id = order_details.pizza_id
```



### IDENTIFY THE HIGHEST-PRICED PIZZA.





# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



Result Grid		
	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.

```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
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.name
ORDER BY quantity DESC
LIMIT 5;
```



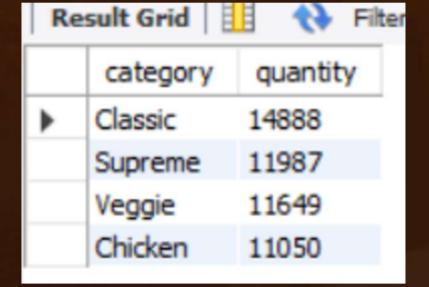
# OUTPUT:

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

Result Grid Filter Rows:

# 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 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
ORDER BY quantity DESC;
```



# 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);
```

1	7	T	8	

,	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

order\_count

1231

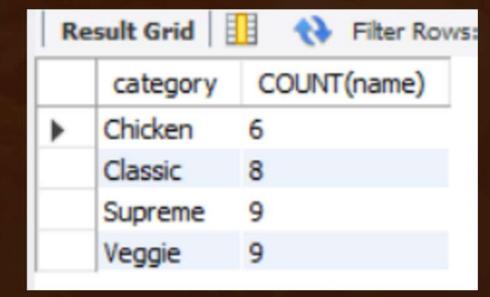
Result Grid



# JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.



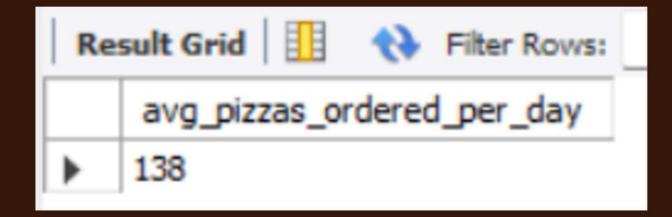
```
category, COUNT(name)
FROM
pizza_types
GROUP BY category;
```







# GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.



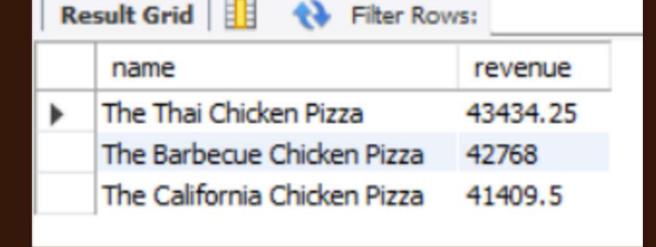




# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.



```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```







# CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
                    SUM(order_details.quantity * pizzas.price)
                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 pizza_types.category
ORDER BY revenue DESC;
```



Re	sult Grid	☐ <b>()</b> Filt
	category	revenue
•	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



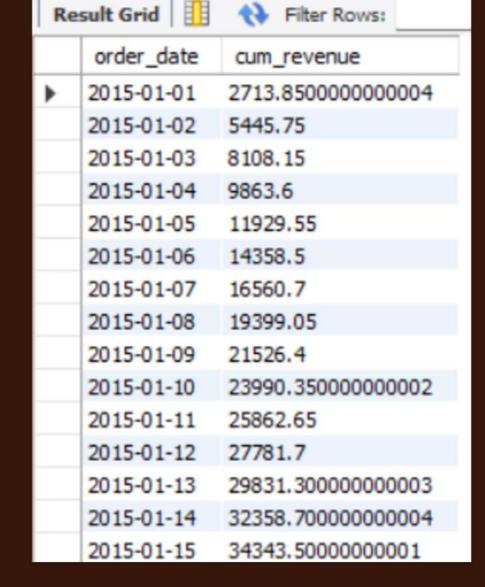


### ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



```
select order_date,
sum(revenue) over(order by order_date) as cum_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;
```











Pizza Resto Presentation

# THANKYOU E FORATTENTION

**See You Next** 

www.reallygreatsite.com