

# Hello!

**My name is MD SHOAIB. In  
this project, I have utilised  
MySQL to solve queries  
related to Pizza sales.**



# Questions

- 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.**
- 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.**
- 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

```
select count(orders.order_id) as total_orders  
from orders;
```

Result Grid	
	total_orders
	21350





# Calculate the total revenue generated from pizza sales

```
SELECT
  round(SUM(order_details.quantity * pizzas.price),0) AS total_revenue
FROM
  order_details
  JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```



Result Grid	
	total_revenue
▶	817860





# Identify the highest-priced pizza

```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizzas
    JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid     Filter Rows		
	name	price
▶	The Greek Pizza	35.95





# Identify the most common pizza size ordered



```
SELECT
  pizzas.size,
  COUNT(order_details.quantity) AS quantity_ordered
FROM
  pizzas
  JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY quantity_ordered DESC
LIMIT 1;
```

Result Grid		Filter R
size	quantity_ordered	
L	18526	





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



```
SELECT  
    pizza_types.name,  
    COUNT(order_details.quantity) AS quantity_ordered  
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_ordered DESC  
LIMIT 5;
```

Result Grid			Filter Rows:
	name	quantity_ordered	
	The Classic Deluxe Pizza	2416	
	The Barbecue Chicken Pizza	2372	
	The Hawaiian Pizza	2370	
	The Pepperoni Pizza	2369	
	The Thai Chicken Pizza	2315	





# Find the total quantity of each type of category of pizza ordered

```
SELECT
    pizza_types.category,
    COUNT(order_details.quantity) AS quantity_ordered
FROM
    order_details
    JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
    JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category
ORDER BY quantity_ordered DESC;
```

	category	quantity_ordered
	Classic	14579
	Supreme	11777
	Veggie	11449
	Chicken	10815





# Distribution of orders by hour of the day.

Result Grid		
	hour	orders
▶	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

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





# Calculate the average number of pizzas ordered per day.



SELECT

ROUND(SUM(quantity\_ordered) / COUNT(order\_date),  
0) AS avg\_orders\_perday

FROM

(SELECT

orders.order\_date,

COUNT(order\_details.order\_details\_id) AS quantity\_ordered

FROM

orders

JOIN order\_details ON orders.order\_id = order\_details.order\_id

GROUP BY orders.order\_date) AS sales;

Result Grid



Fi

avg\_orders\_perday

136





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

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





# Calculate the percentage distribution of each pizza category to total revenue



```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
            0) AS revenue
    FROM
        order_details
        JOIN
            pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
    2) AS percentage_revenue
FROM
    order_details
    JOIN
        pizzas ON pizzas.pizza_id = order_details.pizza_id
    JOIN
        pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category
ORDER BY percentage_revenue DESC;
```

category	percentage_revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68





# Analyze the cumulative revenue generated over time

```
select order_date,  
round(sum(revenue) over(order by order_date),2) as cumulative_revenue  
from  
(select orders.order_date,  
sum(order_details.quantity*pizzas.price) as revenue  
from pizzas join order_details  
on pizzas.pizza_id = order_details.pizza_id  
join orders  
on orders.order_id = order_details.order_id  
group by orders.order_date) as a;
```





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



```
select 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,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    order_details
    JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
    JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category,pizza_types.name) as a) as b
where ranking <= 3;
```

result Grid				Filter Rows:	Export:
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	32265.70000000065			
Veggie	The Mexicana Pizza	26780.75			
Veggie	The Five Cheese Pizza	26066.5			





Total orders

21.35K

Total revenue

817.86K

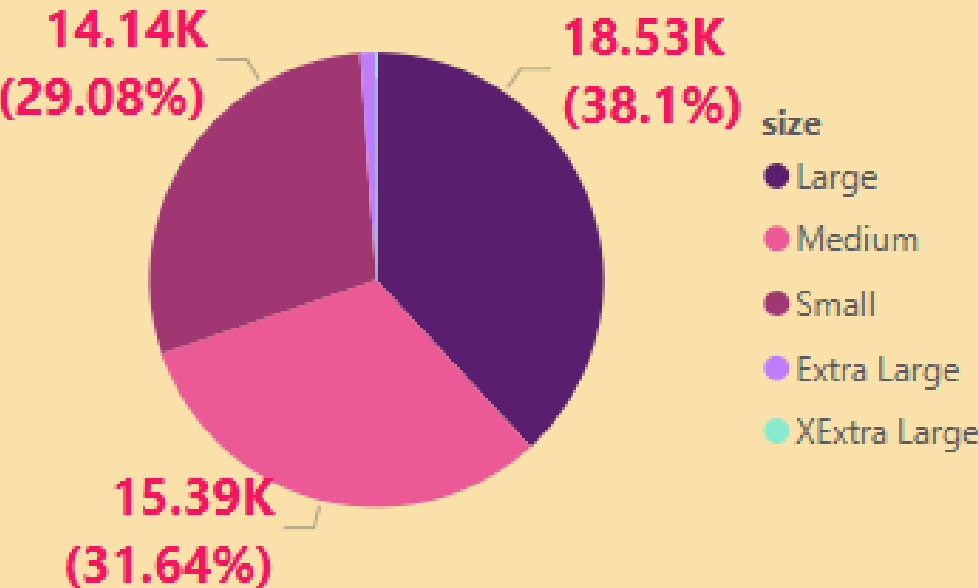
Most Ordered Pizza

The Classic  
Deluxe Pizza  
(2416)

Most Common  
Pizza Size Ordered

Large (18526)

orders by size of pizzas



Month Name

January

February

March

April

May

June

July

August

Septem...

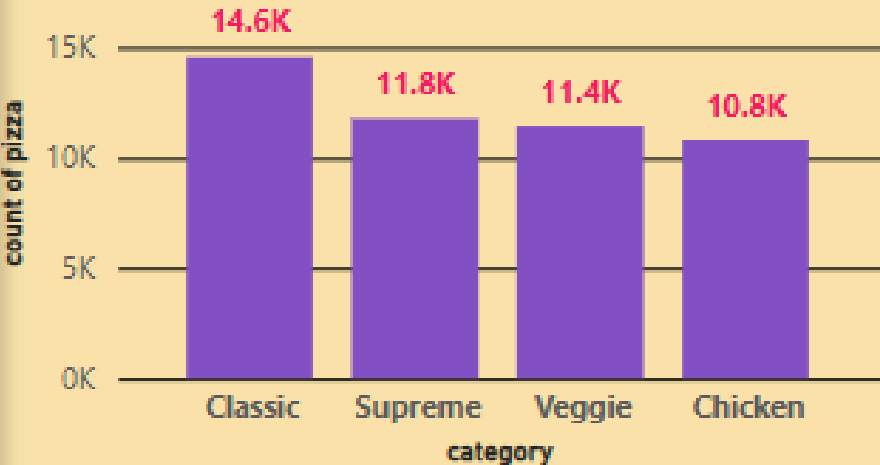
October

November

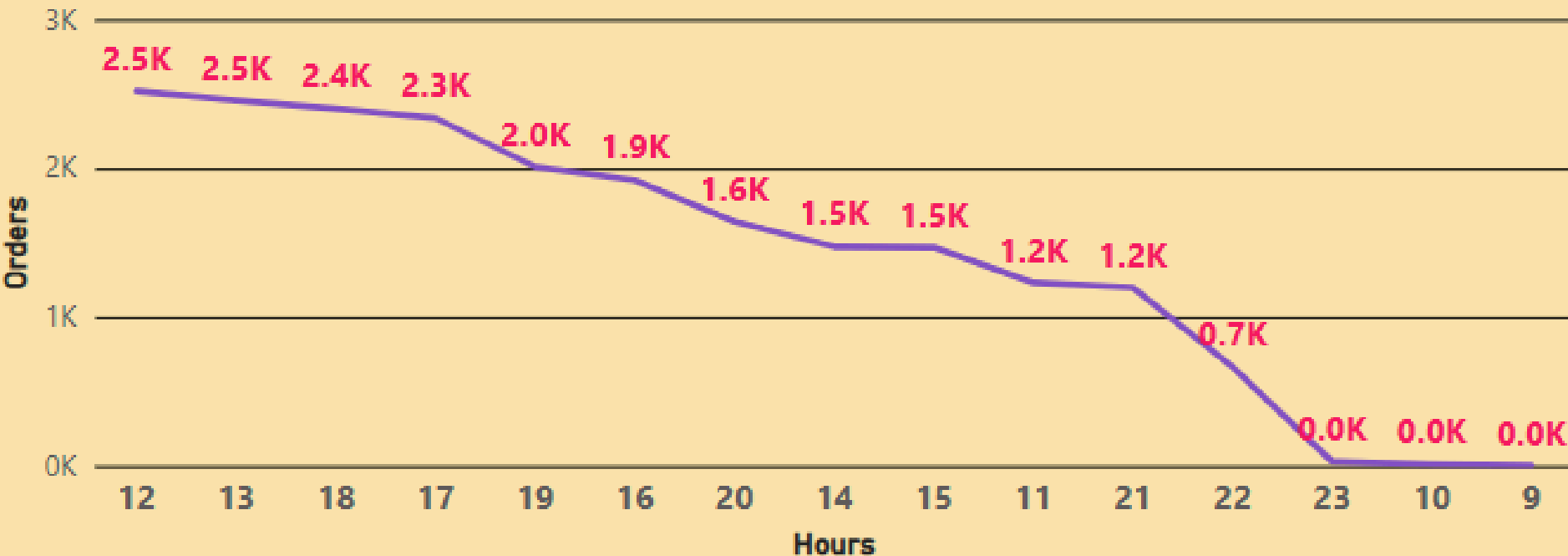
Lists of Pizzas with price

category	pizza name	price
Chicken	The Barbecue Chicken...	50.25
Chicken	The California Chicken...	50.25
Chicken	The Chicken Alfredo P...	50.25

Count of pizza\_id by category



Orders by hour





THANK  
YOU





