

# *Pizza Sales Analysis*

## *Using MySQL*



# HELLO

My name is Pooja Gupta and in this project i  
have utilize sql queries to solve the question  
which are related to pizza sales

# *Project Purpose:*

- The primary objective of this project is to analyze pizza sales data to answer a series of business-related questions. By doing so, we can uncover valuable trends and patterns that can help improve sales strategies and customer satisfaction.



# *Total Number of Orders*

## Query.

```
SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    orders;
```

## Output

total_orders
21350

# Total Revenue Generated

## Query

```
SELECT  
    ROUND(SUM(p.price * od.quantity), 2) AS total_revenue  
FROM  
    order_details od  
    JOIN  
    pizzas p ON od.pizza_id = p.pizza_id;
```

## Output

total_revenue
817860.05



# *Highest-Priced Pizza*

## Query

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



## Output

name	price
The Greek Pizza	35.95

# Most Common Pizza Size

## Query

```
SELECT  
    p.size, COUNT(od.order_details_id) AS order_count  
FROM  
    pizzas p  
    JOIN  
    order_details od ON p.pizza_id = od.pizza_id  
GROUP BY p.size  
ORDER BY COUNT(od.order_details_id) DESC;
```

## Output

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

# Top 5 Most Ordered Pizza Types

## Query

SELECT

```
pt.name, SUM(od.quantity) AS total_quantity
```

FROM

```
pizza_types pt
```

JOIN

```
pizzas ON pt.pizza_type_id = pizzas.pizza_type_id
```

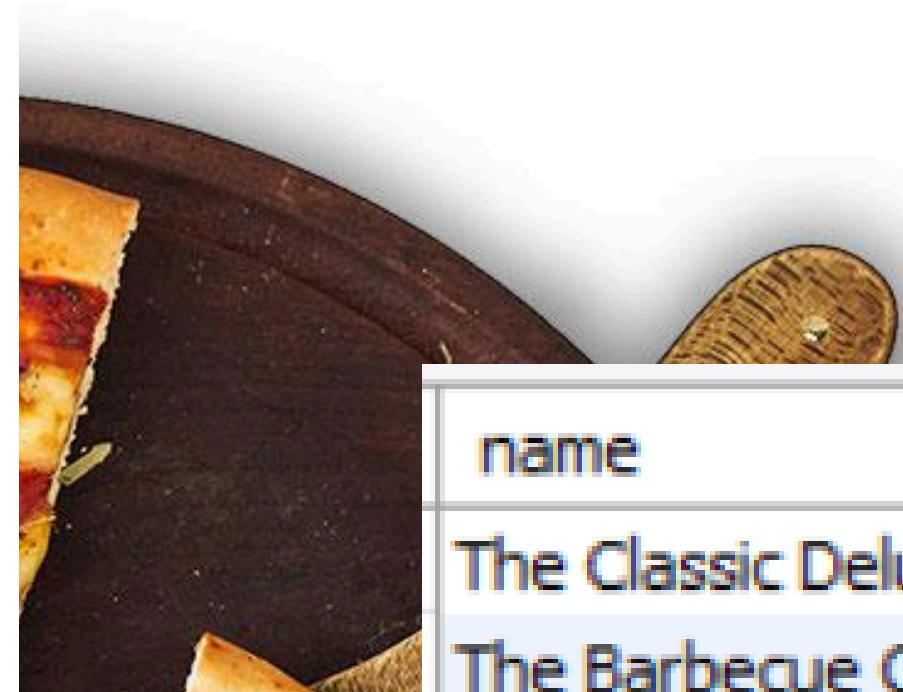
JOIN

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

GROUP BY pt.name

ORDER BY SUM(od.quantity) DESC

LIMIT 5;



## Output

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

# Total Quantity of Each Pizza Category Ordered

## Query

```
SELECT  
    pizza_types.category,  
    SUM(order_details.quantity) AS total_quantity  
  
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 pizza_types.category  
ORDER BY total_quantity DESC;
```



## Output

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

# *Distribution of Orders by Hour*

## Query

**SELECT**

```
HOUR(order_time) AS hours, COUNT(order_id) AS order_id
```

**FROM**

```
orders
```

```
GROUP BY HOUR(order_time);
```



## Output

hours	order_id
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

# *Category-Wise Distribution of Pizzas*

## Query

**SELECT**

category, COUNT(name)

**FROM**

pizza\_types

**GROUP BY category;**



## Output

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

# *Average Number of Pizzas Ordered per Day*

## Query

```
SELECT  
    ROUND(AVG(quantity), 0) AS average_no_of_pizza_ord  
FROM  
(SELECT  
    o.order_date, SUM(od.quantity) AS quantity  
FROM  
    orders o  
JOIN order_details od ON o.order_id = od.order_id  
GROUP BY o.order_date) AS quantity_ordered  
;
```

## Output

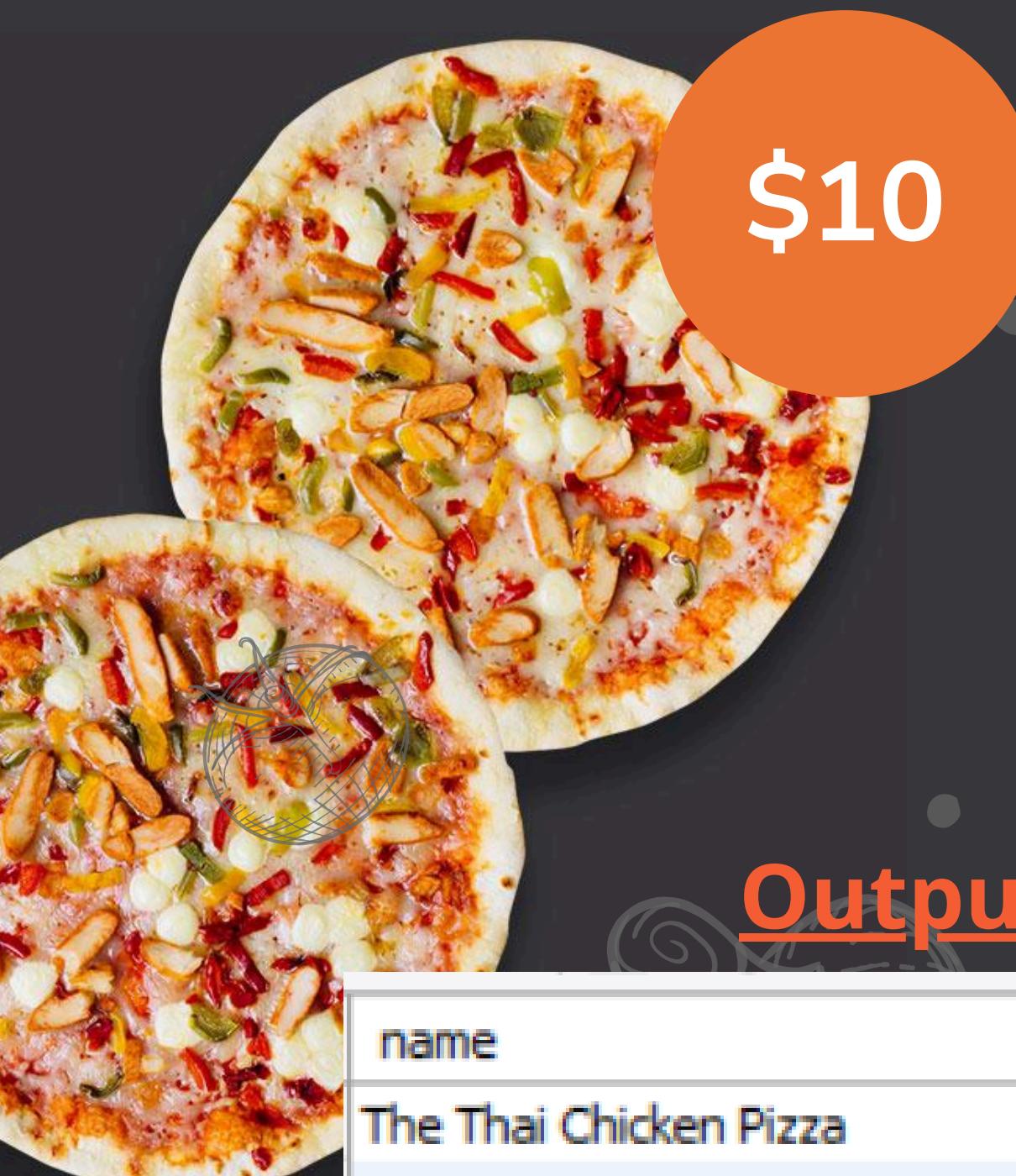
average_no_of_pizza_ord
138



# Top 3 Most Ordered Pizza Types by Revenue

## Query.

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



## Output

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

# Percentage Contribution to Total Revenue

## Query

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



## Output

category	percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



# Cumulative Revenue Over Time

## Query.

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



## Output

order_date	revenue	cummulative_revenue
2015-01-01	2713.85	2713.85
2015-01-02	2731.9	5445.75
2015-01-03	2662.4	8108.15
2015-01-04	1755.45	9863.6
2015-01-05	2065.95	11929.55
2015-01-06	2428.95	14358.5
2015-01-07	2202.2	16560.7
2015-01-08	2838.35	19399.05
2015-01-09	2127.35	21526.4
2015-01-10	2463.95	23990.35
2015-01-11	1872.3	25862.65
2015-01-12	1919.05	27781.7
2015-01-13	2049.6	29831.3
2015-01-14	2527.4	32358.7
2015-01-15	1984.8	34343.5
2015-01-16	2594.15	36937.65
2015-01-17	2064.1	39001.75
2015-01-18	1976.85	40978.6
2015-01-19	2387.15	43365.75
2015-01-20	2397.9	45763.65
2015-01-21	2040.55	47804.2
2015-01-22	2496.7	50300.9

# *Top 3 Most Ordered Pizza Types by Category*

## Query.

```
select name , revenue from
(select pizza_types.category , pizza_types.name ,
sum(od.quantity * p.price) as revenue ,
rank() over(partition by pizza_types.category
order by sum(od.quantity * p.price) desc) as rn
from pizza_types join pizzas p
on pizza_types.pizza_type_id = p.pizza_type_id
join order_details od
on p.pizza_id = od.pizza_id
group by pizza_types.category , pizza_types.name) as a
where rn <= 3
```

## Output

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
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	4831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

# Conclusion

- The analysis provides a comprehensive understanding of sales patterns, customer preferences, and revenue drivers.
- Insights from the data can be used to optimize inventory, improve marketing strategies, and enhance customer satisfaction.
- Understanding peak order times and popular products can help in better resource allocation and promotional planning.

# *Future Recommendations:*

- Regularly update the analysis to keep track of changing trends and customer preferences.
- Use the insights gained to introduce new products, improve existing offerings, and plan targeted marketing campaigns.
- Continuously monitor revenue contributions and adjust the menu to maximize profitability.

# *Thank You*

I hope this presentation offered meaningful insights into our pizza sales data and demonstrated how data analysis can drive better business decisions.

