



PIZZAHUT

PIZZA HUT





WELCOME TO PIZZAHUT

HELLO

"Hello everyone, my name is Nikhil Lonkar, and welcome to my presentation on analyzing pizza sales using MYSQL. In this presentation, we'll explore how MYSQL can help us answer key questions about pizza sales, such as which toppings are most popular, which days of the week see the highest sales, and how sales vary by region. Let's get started!"





"PIZZA SALES ANALYSIS"

Title: "Pizza Sales Analysis Questions"

- Briefly introduce the purpose of the analysis.
- Mention that the questions are categorized into Basic, Intermediate, and Advanced levels.

"Basic Analysis Goals"

Use bullet points to present the questions:

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

"Intermediate Analysis Goals"

Split the questions into subcategories with visuals:

1. Joining Tables:

- Join necessary tables to find the total quantity of each pizza category ordered.
- Join relevant tables to find the category-wise distribution of pizzas.

2. Order Patterns:

- Determine the distribution of orders by the hour of the day.
- Group orders by date and calculate the average number of pizzas ordered per day.

3. Revenue Insights:

- Determine the top 3 most ordered pizza types based on revenue.

" Advanced Analysis Goals"

Organize questions in a step-by-step progression:

1. Revenue Analysis:

- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.

2. Category-Specific Insights:

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

🍕 RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.🍕

SELECT

COUNT(order_id) **AS** total_orders

FROM

orders;

total_orders

21350

PIZZA CHALLENGE

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
```

	total_sales
▶	817860.05



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;
```

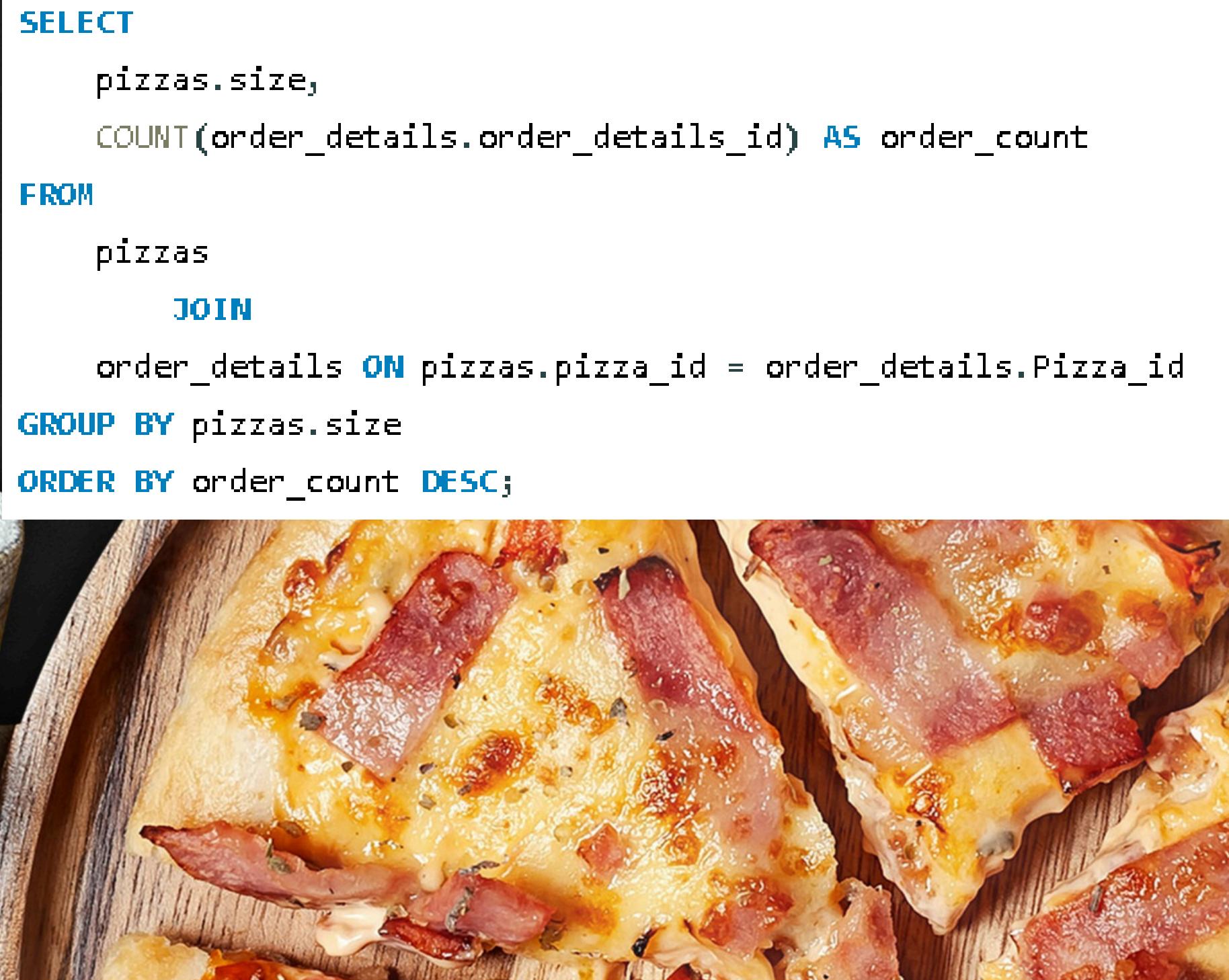


name	price
The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



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



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;

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



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;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



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);

hour	order_count
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



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

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



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

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

```
SELECT  
    ROUND(AVG(quantity), 0) AS per_day_order  
FROM  
(SELECT  
    orders.order_date, SUM(order_details.quantity) AS quantity  
FROM  
    orders  
JOIN order_details ON orders.order_id = order_details.order_id  
GROUP BY orders.order_date) AS order_quantity;
```



per_day_order
138



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 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 revenue **DESC**

LIMIT 3;

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

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) AS total_sales  
  
  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;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



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

```
select order_date,  
sum(revenue) over( order by order_date) as cummulative_revenue  
from  
(select orders.order_date,  
round(sum(order_details.quantity*pizzas.price),2) 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;
```

order_date	cummulative_revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
select name, revenue
from
    (select category, name , revenue,
    rank() over(partition by category order by revenue desc) as rn
    from
        (select pizza_types.category, pizza_types.name,
        sum(order_details.quantity * pizzas.price) 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, pizza_types.name) as a) as b
where rn <=3;
```

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	34831.25
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

"THANK YOU & KEY TAKEAWAYS"

1. SUMMARY OF INSIGHTS:

"THIS ANALYSIS ENABLES DATA-DRIVEN DECISIONS TO OPTIMIZE SALES, ENHANCE CUSTOMER SATISFACTION, AND MAXIMIZE REVENUE IN THE PIZZA BUSINESS."

2. ACTIONABLE NEXT STEPS:

"FOCUS ON HIGH-REVENUE PIZZA TYPES, STREAMLINE OPERATIONS DURING PEAK HOURS, AND REFINE MENU PRICING BASED ON SALES PERFORMANCE."

3. ACKNOWLEDGMENT:

"THANK YOU FOR JOINING US ON THIS DATA-DRIVEN JOURNEY TO IMPROVE PIZZA SALES!"

"DATA IS THE SECRET INGREDIENT FOR EVERY SUCCESSFUL RECIPE!"

