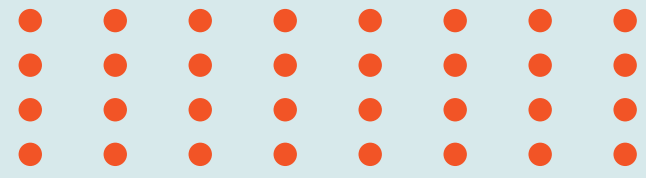


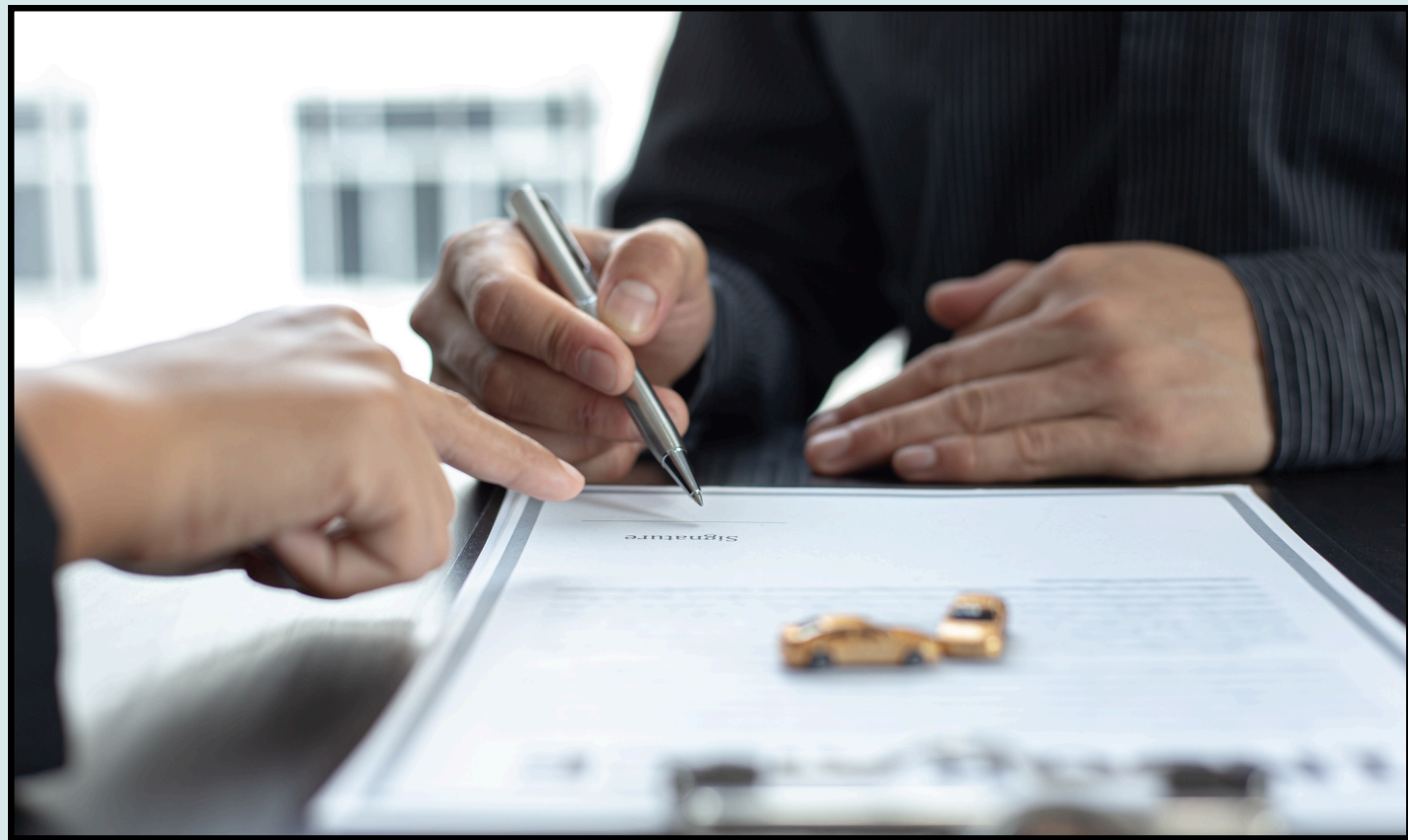
PIZZA 🍕 | SALES REPORT

26, DEC, 2024





Introduction



SQL PIZZAHUT
SALES REPORT

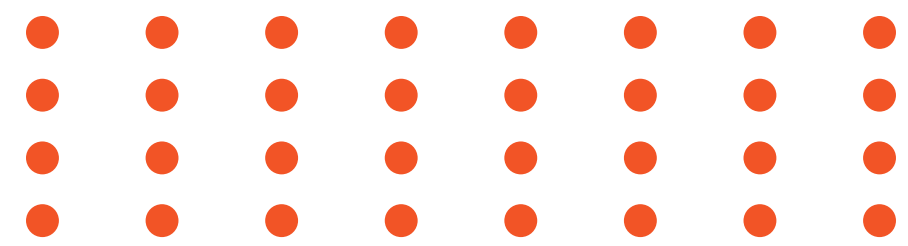
Welcome to this Sales Report Presentation. my name is Vikas dhakad and I delve into a comprehensive overview of our pizza sales performance, exploring the highs, challenges, and strategic insights that have shaped my journey. This presentation is more than just numbers, it's a narrative of my collective efforts, showcasing the impact of our sales strategies and the pathways to future success.

PIZZAHUT

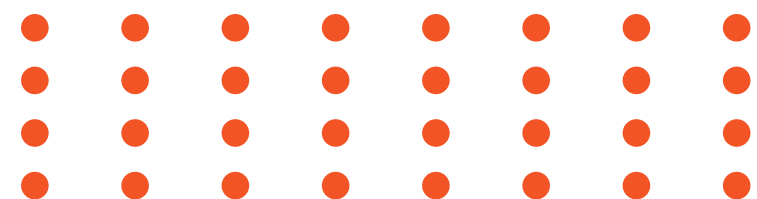
- 01 **PIZZAS**
- 02 **PIZZA_TYPES**
- 03 **ORDERS**
- 04 **ORDER_DETAILS**



Queries-->



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. Calculate the percentage contribution of each pizza type to total 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.



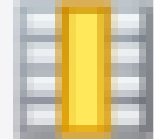

1.Retrieve the total number of orders placed

```
SELECT  
    COUNT(*)  
FROM  
    orders;
```

Result Grid	
	COUNT(*)
▶	21350



2. Calculate the total revenue generated from pizza sales.

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

Result Grid			
	total_revenue		
▶	817860.05		

3. Identify the highest-priced pizza.

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

Result Grid   Filter Rows		
	name	price
▶	The Greek Pizza	35.95

4. Identify the most common pizza size ordered.

```
SELECT
    p.size, COUNT(od.quantity) AS counts
FROM
    pizzas p,
    order_details od
WHERE
    p.pizza_id = od.pizza_id
GROUP BY p.size
ORDER BY counts DESC
LIMIT 1;
```

Result Grid		
	size	counts
▶	L	18526



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

```
SELECT
    pt.name, SUM(od.quantity) AS quantity
FROM
    pizza_types pt,
    order_details od,
    pizzas p
WHERE
    pt.pizza_type_id = p.pizza_type_id
    AND od.pizza_id = p.pizza_id
GROUP BY pt.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

6. Join the necessary tables to find the total quantity of each pizza category ordered

```
SELECT
    pt.category, SUM(od.quantity) AS quantity
FROM
    pizza_types pt,
    order_details od,
    pizzas p
WHERE
    pt.pizza_type_id = p.pizza_type_id
    AND od.pizza_id = p.pizza_id
GROUP BY pt.category;
```

Result Grid					Filter
	category	quantity			
▶	Classic	14888			
	Supreme	11987			
	Veggie	11649			
	Chicken	11050			

7. Determine the distribution of orders by hour of the day.

```
SELECT  
    HOUR(time) AS hours, COUNT(order_id) AS count_of_orders  
FROM  
    orders  
GROUP BY hours
```

	hours	count_of_orders
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2226


8. Join relevant tables to find the category-wise distribution of pizzas

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

	category	count_pizzas
	Chicken	6
	Classic	8
	Supreme	9
▶	Veggie	9

9. Group the orders by date and calculate the average number of pizzas ordered per day.

```
select
round(avg(quantity),0) as average_pizza_ordered_per_day
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;
```

	average_pizza_ordered_per_day 
1	138

10. Calculate the percentage contribution of each pizza type to total revenue.

```
SELECT
    pizza_types.category,
    ROUND((SUM(pizzas.price * order_details.quantity)) / (SELECT
        SUM(pizzas.price * order_details.quantity)
    FROM
        pizzas,
        order_details
    WHERE
        pizzas.pizza_id = order_details.pizza_id) * 100,
    2) AS percentage
FROM
    pizzas,
    order_details,
    pizza_types
WHERE
    pizzas.pizza_id = order_details.pizza_id
    AND pizza_types.Pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.Category
```

category	percentage
Supreme	25.46
Classic	26.91
Veggie	23.68
Chicken	23.96



11. Calculate the percentage contribution of each pizza type to total revenue.

```
    pizza_types.category,  
    ROUND((SUM(pizzas.price * order_details.quantity)) / (SELECT  
        SUM(pizzas.price * order_details.quantity)  
        FROM  
            pizzas,  
            order_details  
        WHERE  
            pizzas.pizza_id = order_details.pizza_id) * 100,  
    2) AS percentage  
FROM  
    pizzas,  
    order_details,  
    pizza_types  
WHERE  
    pizzas.pizza_id = order_details.pizza_id  
    AND pizza_types.Pizza_type_id = pizzas.pizza_type_id  
GROUP BY pizza_types.Category
```

category	percentage
Supreme	25.46
Classic	26.91
Veggie	23.68
Chicken	23.96

12. 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 orders,order_details,pizzas  
where orders.order_id=order_details.order_id  
and pizzas.pizza_id=order_details.pizza_id  
group by orders.order_date ) as sales;
```

	order_date  date	cum_revenue  numeric
253	2015-09-10	577546.10
254	2015-09-11	580308.00
255	2015-09-12	582896.15
256	2015-09-13	584734.30
257	2015-09-14	586899.55
258	2015-09-15	589449.75

Total rows: 358 of 358

Query com

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

```
SELECT
    pizza_types.Category,
    ROUND(SUM(pizzas.price * order_details.quantity),
          2) AS revenue
FROM
    pizza_types,
    order_details,
    pizzas
WHERE
    pizza_types.Pizza_type_id = pizzas.pizza_type_id
    AND order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.Category
ORDER BY revenue DESC
LIMIT 3
```

	Category	revenue
▶	Classic	220053.10
	Supreme	208197.00
	Chicken	195919.50



Thank You

26, DEC, 2024

