PIZZA SALES REPORT

2024

presented by
Rakesh Yelve

INTRODUCTION

2024

Hello, my name is Rakesh Yelve. In this SQL database project for Pizza Hut, I have executed various queries to answer specific questions, demonstrating my skills in data retrieval, manipulation, and analysis to gain insights relevant to the business.

presented by
Rakesh Yelve

02

1.Retrieve the total number of orders placed.

```sql
SELECT
  count(order_id) as total_orders
from orders;
```

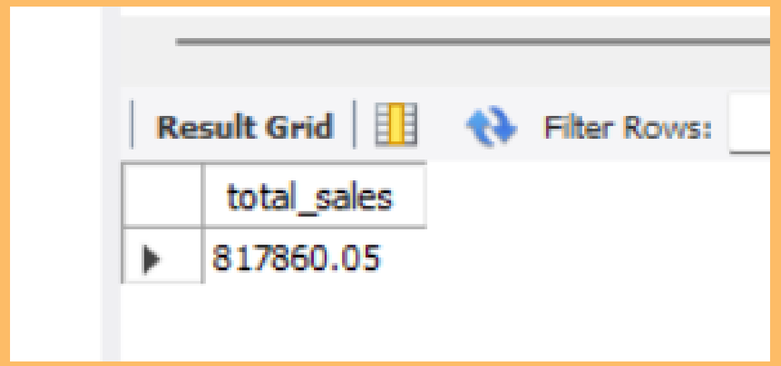
Result Grid | Filter

total_orders
21350

2.Calculate the total revenue generated from pizza sales

2024

```sql
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;
```



3.Identify the highest-priced pizza

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

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

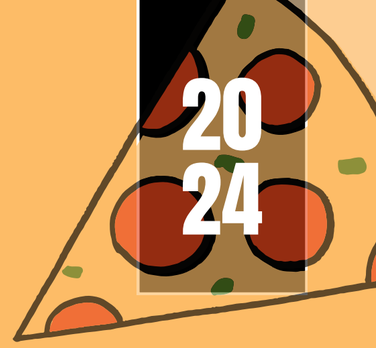
4.Identify the most common pizza size ordered

```
SELECT
    pizzas.size,
    COUNT(order_details.order_detail_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;
```

Result Grid | Filte

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

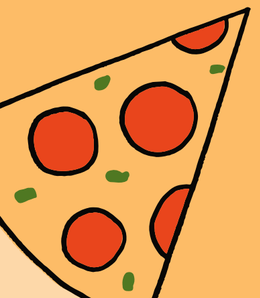
5.Determine the top 3 most ordered pizza types based on revenue for each pizza category



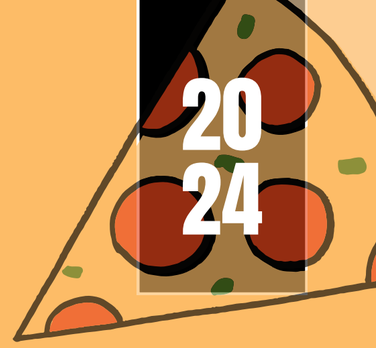

```sql
SELECT name, revenue
FROM (
    SELECT category,
           name,
           revenue,
           RANK() OVER (PARTITION BY category ORDER BY revenue DESC) AS rn
    FROM (
        SELECT pizza_types.category,
               pizza_types.name,
               SUM(order_details.quantity * pizzas.price) 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;
```

Result Grid | Filter Rows:

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



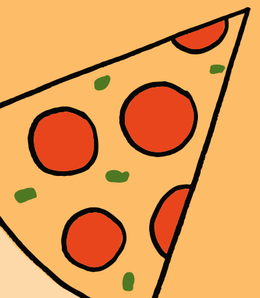
6.Join the necessary tables to find the total quantity of each pizza category ordered




```sql
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;
```

Result Grid | Filter Rows:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



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

2024

```sql
select hour(order_time) as hour , count(order_id) as order_count from orders
group by hour(order_time);
```

Result Grid | Filter Rows:

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399

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

2024

```
select category, count(name) from pizza_types
group by category;
```

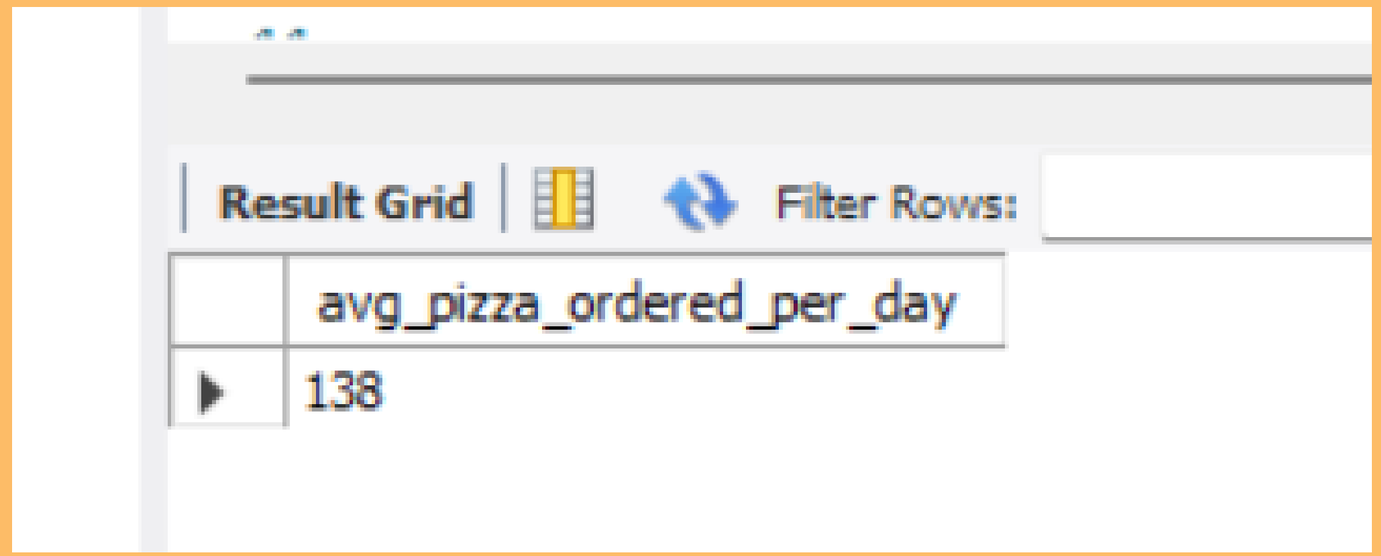
Result Grid | Filter Rows:

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

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

2024

```sql
SELECT ROUND(AVG(quantity), 0) as avg_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;
```



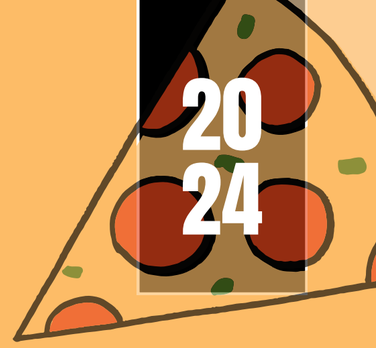
10.Determine the top 3 most ordered pizza types based on revenue

```
SELECT pizza_types.name,
sum(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;
```

Result Grid | Filter Rows:

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

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

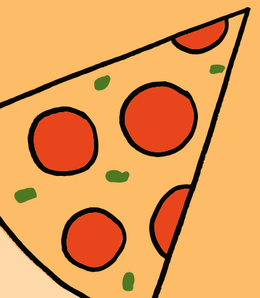


2024

```sql
SELECT pizza_types.category,
       round((SUM(order_details.quantity * pizzas.price) /
        (SELECT ROUND(SUM(order_details.quantity * pizzas.price), 2)
         FROM order_details
         JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id)
       ) * 100 ,2) AS percentage_of_total_sales
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 percentage_of_total_sales DESC;
```

Result Grid | Filter Rows:

category	percentage_of_total_sales
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

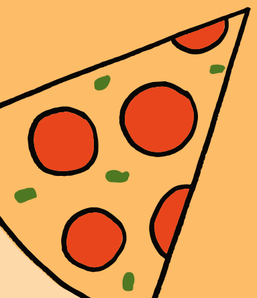


12.Analyze the cumulative revenue generated over time

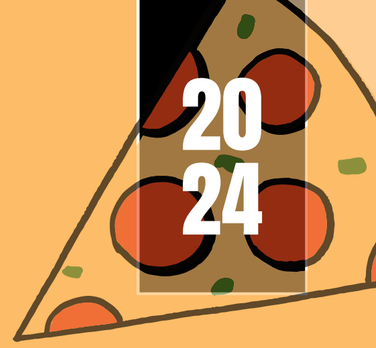

```sql
select order_date,
sum(revenue) over (order by order_date) as cum_revenue
from
(SELECT orders.order_date,
        SUM(order_details.quantity * pizzas.price) AS revenue
FROM order_details
JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
JOIN orders ON orders.order_id = order_details.order_id
GROUP BY orders.order_date) as sales;
```

Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001



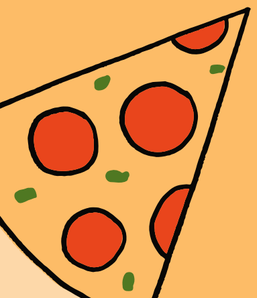
13.Determine the top 3 most ordered pizza types based on revenue for each pizza category




```sql
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;
```

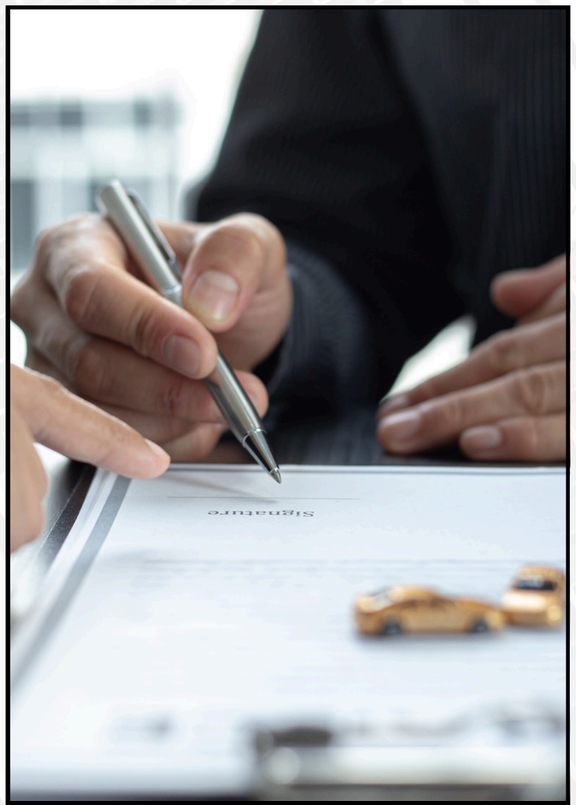
Result Grid | Filter Rows:

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



2024

SALES PERFORMANCE



, This analysis provides a comprehensive view of sales performance, customer preferences, and demand patterns. These insights can be applied to optimize product offerings, improve operational efficiency, and enhance marketing strategies for better customer alignment and revenue growth. By examining these metrics, we can identify areas of strength and opportunities for improvement.

2024

ACTION PLAN

Based on our analysis and insights, we will outline an action plan to address key findings and capitalize on opportunities. This may include adjustments to sales strategies, investment in new technologies or resources, or targeted initiatives to address specific market segments or customer needs.

THANK YOU!

Thank you for your attention to our sales report presentation. If you have any questions or would like to discuss the findings in more detail, please don't hesitate to reach out to our sales team. We appreciate your continued support and partnership.