



About

Contact



WHERE EVERY SLICE TELLS A STORY



About

Contact





HELLO

I am Anshu Prasad, and I worked on a SQL-based project focused on analyzing sales data from a pizza restaurant. Through the use of complex queries, I examined sales trends and revenue generation. This project highlights my proficiency in SQL, data analysis, and my ability to address real-world business challenges.

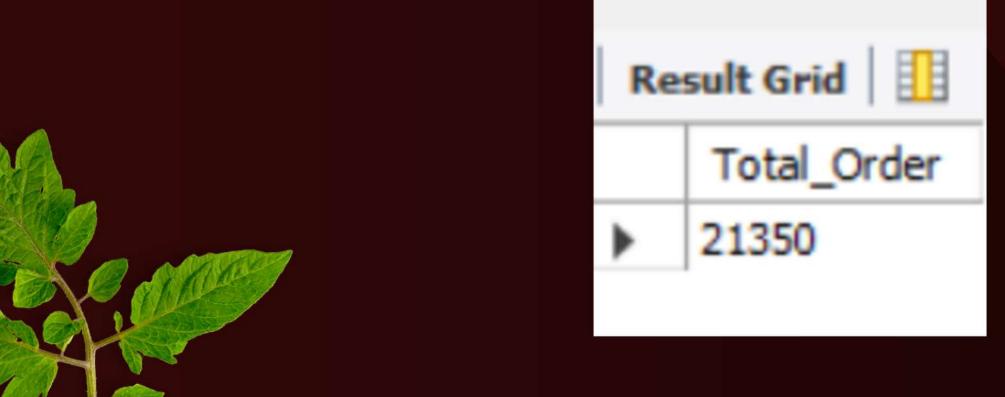




1) Retrieve the total number of orders placed

SELECT COUNT(ordeR_id) AS Total_Order FROM orders;







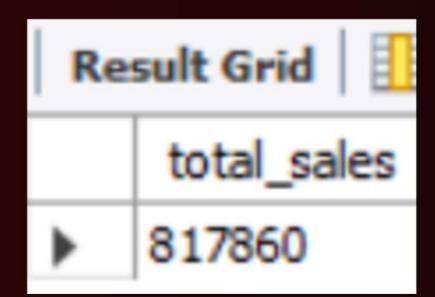


2) Calculate the total revenue generated from pizza sales

```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price)) AS total_sales
FROM
    order_details
    JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```









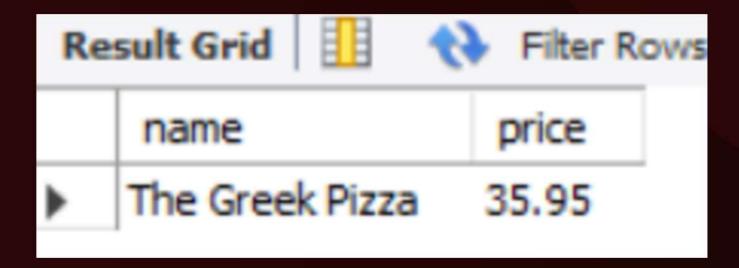


3) Identify the highest-priced pizza

```
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;
```



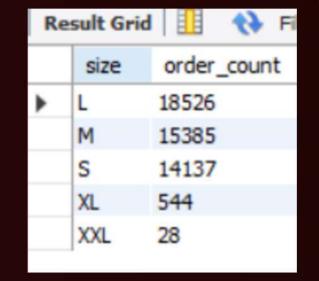








4) Identify the most common pizza size ordered











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

```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantit
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_i
GROUP BY pizza_types.name
```

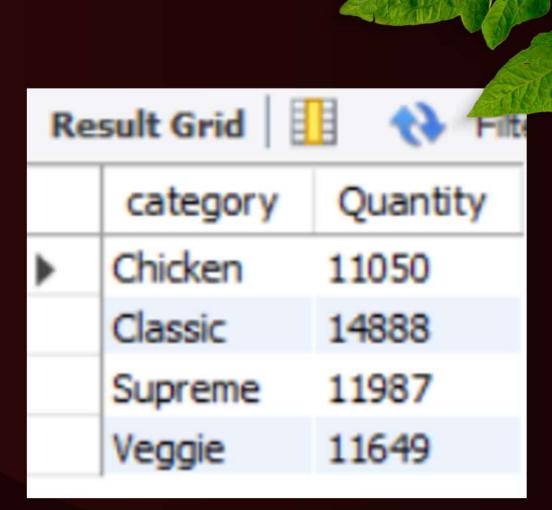






6) Join the necessary tables to find the total quantity of each pizza category ordered.

```
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 category
ORDER BY quantity DESC;
```

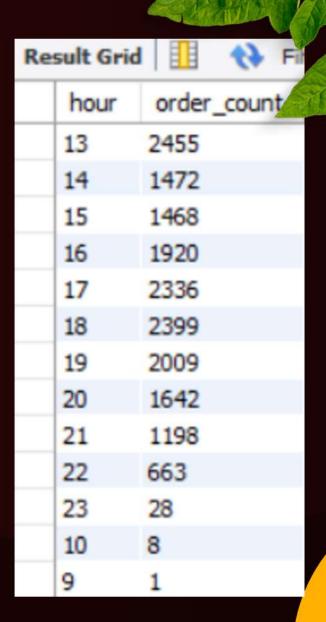






7) Determine the distribution of orders by hour of the day

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









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

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

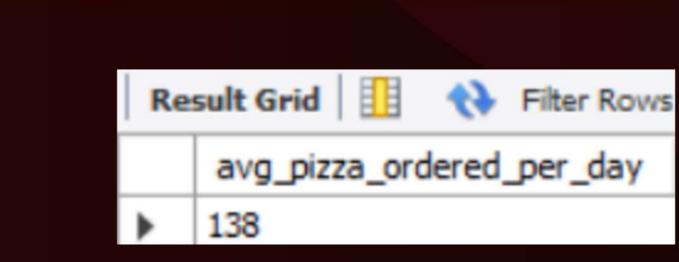




9) Group the orders by date and calculate the average number of pizzas ordered per day

```
SELECT
    ROUND(AVG(quantity)) 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 order_date) AS Order_quantity;
```



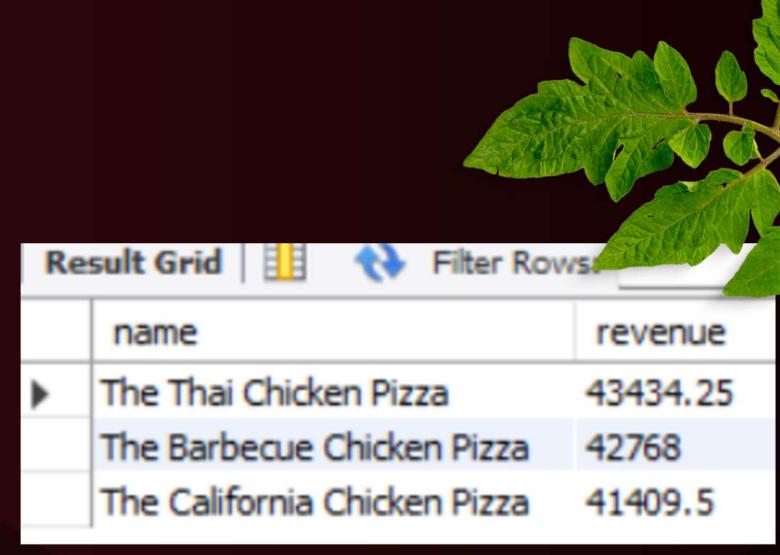






10) Determine the top 3 most ordered pizza types based on revenue

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON 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;
```

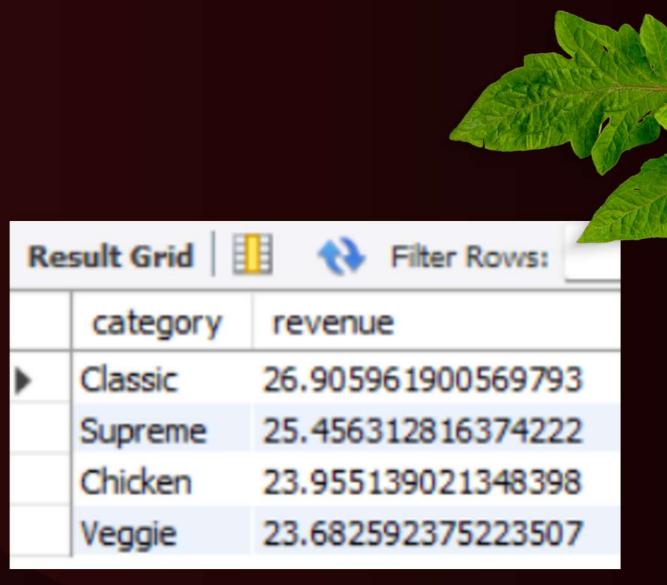






11) Calculate the percentage contribution of each pizza type to total revenue

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

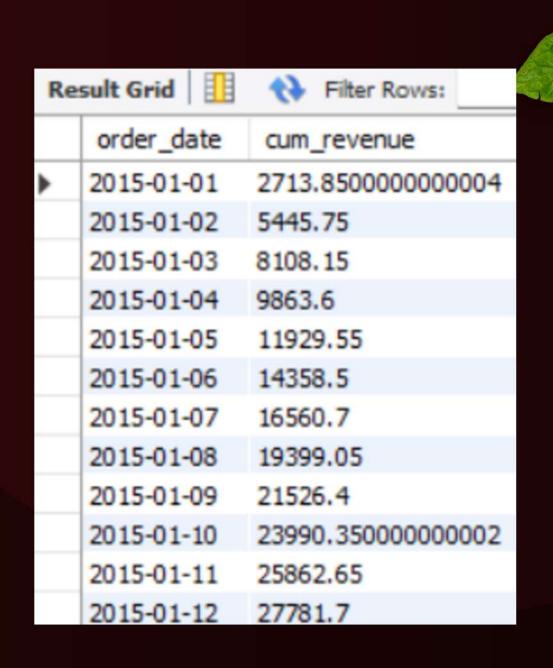






12)Analyze the cumulative revenue generated over time.

```
select order_date,
sum(revenue) over (order by order_date) as cum_reve
FROM
(select orders.order_date,
  sum(order_details.quantity * pizzas.price) as reve
from order_details  Join pizzas
ON order_details.pizza_id = pizzas.pizza_id
Join orders
ON orders.order_id = order_details.order_id
```

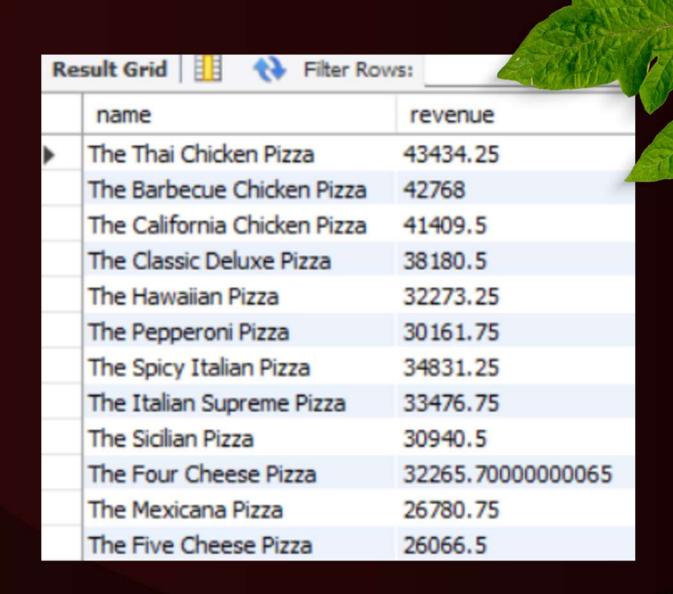






13) Determine the top 3 most ordered pizza types based on revenue for each pizza category

```
select name ,revenue
from
(select category ,name ,revenue ,
    rank() over (partition by category order by revenue desc) as ran
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 ran <=3;</pre>
```







FOR ATTENTION

2025 PIZZA RESTO PRESENTATION