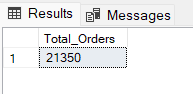
1. Count the number of orders.

SELECT COUNT(DISTINCT orders.order\_id) AS Total\_Orders FROM orders;

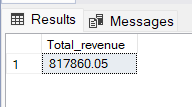


1. Calculate the total revenue?

SELECT ROUND(SUM(pizzas.price\*order\_details.quantity),2) as Total\_revenue FROM pizzas

JOIN order\_details

ON pizzas.pizza\_id = order\_details.pizza\_id;



1. Identify the highyest priced Pizza?

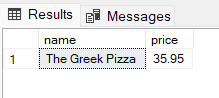
SELECT TOP 1 pizza\_types. name, pizzas.price FROM pizza\_types

JOIN pizzas

ON pizza\_types.pizza\_type\_id = pizzas.pizza\_type\_id

ORDER BY price DESC

;



1. Identify the most common pizza ordered?

SELECT TOP 1 pizza\_types.name, COUNT (quantity) AS Quantities FROM pizza\_types

JOIN pizzas

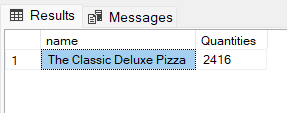
ON pizza\_types.pizza\_type\_id = pizzas.pizza\_type\_id

JOIN order\_details

ON pizzas.pizza\_id = order\_details.pizza\_id

GROUP BY pizza\_types.name

ORDER BY Quantities DESC;



1. Identify the most common pizza size ordered?

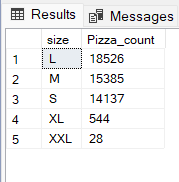
SELECT pizzas.size, COUNT(order\_details.quantity) AS Pizza\_count FROM pizzas

JOIN order\_details

ON pizzas.pizza\_id = order\_details.pizza\_id

GROUP BY pizzas.size

ORDER BY Pizza\_count DESC;



1. Identify the most common pizza types along with quantity?

SELECT TOP 5 pizza\_types.name, SUM(order\_details.quantity) AS Most\_common FROM pizza\_types

JOIN pizzas

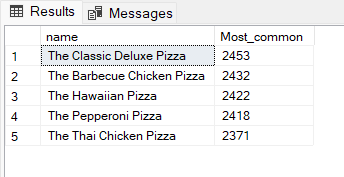
ON pizza\_types.pizza\_type\_id = pizzas.pizza\_type\_id

JOIN order\_details

ON pizzas.pizza\_id = order\_details.pizza\_id

GROUP BY pizza\_types.name

ORDER BY Most\_common DESC;



1. Find the total quantity of each pizza?

SELECT pizza\_types.category, SUM(order\_details.quantity) AS Total\_Quantity FROM pizza\_types

JOIN pizzas

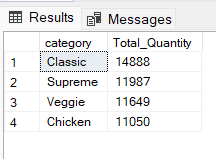
On pizza\_types.pizza\_type\_id = pizzas.pizza\_type\_id

JOIN order\_details

ON pizzas.pizza\_id = order\_details.pizza\_id

GROUP BY category

ORDER BY Total\_Quantity DESC;

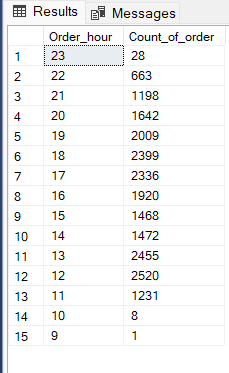


1. Determine the distribution of orders by hour of the day?

SELECT DATEPART(HOUR, orders.time) AS Order\_hour, COUNT (orders.order\_id) AS Count\_of\_order FROM orders

GROUP BY DATEPART(HOUR, orders.time)

ORDER BY DATEPART(HOUR, orders.time) DESC



1. Find the category wise distribution of pizzas?

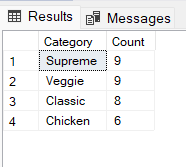
SELECT pizza\_types.category AS Category, COUNT(pizza\_types.category) AS Count

FROM pizza\_types

GROUP BY Category

ORDER BY COUNT(pizza\_types.category) DESC

;



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

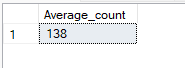
SELECT AVG(Quantity) AS Average\_count FROM

(SELECT orders.date, SUM(order\_details.quantity) AS Quantity

FROM orders JOIN order\_details

ON orders.order\_id = order\_details.order\_id

GROUP BY orders.date) AS Order\_quantity;



1. Determine the top 3 most ordered piza based on Revenue?

SELECT TOP 3 pizza\_types.name, SUM(pizzas.price\*order\_details.quantity) AS Summation

FROM pizza\_types

JOIN pizzas

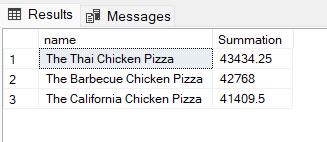
ON pizza\_types.pizza\_type\_id = pizzas.pizza\_type\_id

JOIN order\_details

ON pizzas.pizza\_id = order\_details.pizza\_id

GROUP BY pizza\_types.name

ORDER BY Summation DESC;



1. Calculate the percentage distribution 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

JOIN order\_details

ON pizzas.pizza\_id = order\_details.pizza\_id))\*100,2) AS Revenue

FROM pizza\_types

JOIN pizzas

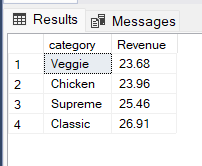
ON pizza\_types.pizza\_type\_id = pizzas.pizza\_type\_id

JOIN order\_details

ON order\_details.pizza\_id = pizzas.pizza\_id

GROUP by category

ORDER by Revenue;



1. Analyze the cumulative revenue generated over time?

SELECT date, SUM(Revenue) OVER (ORDER BY date) AS Cum\_Revenue

FROM

(SELECT orders.date, SUM(pizzas.price\*order\_details.quantity) AS Revenue FROM pizzas

JOIN order\_details

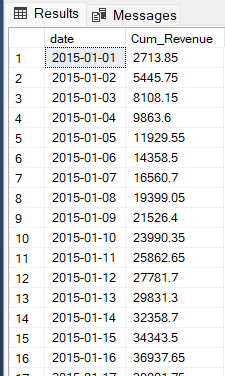
ON pizzas.pizza\_id = order\_details.pizza\_id

JOIN orders

ON order\_details.order\_id = orders.order\_id

GROUP BY orders.date) AS totals

;



1. Determine the top 3 most ordered pizza types based on revenue for each pizza?

SELECT category, name, Revenue

FROM

(SELECT category, name, REVENUE, RANK () over (PARTITION BY category ORDER BY REVENUE DESC) AS Ranks 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 category, name) AS Table\_1) AS Table\_2

WHERE Ranks<=3;

