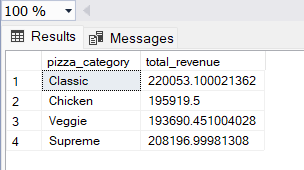
--Intermediate SQL Queries

--1.Total Revenue per Pizza Category

SELECT pizza\_category, SUM(total\_price) AS total\_revenue

FROM pizza\_sales

GROUP BY pizza\_category;



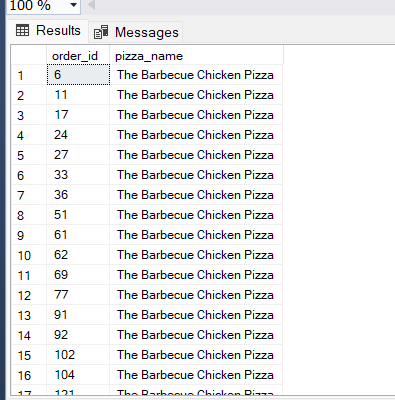
--2.Orders with Unique Pizza Names

SELECT order\_id ,pizza\_name

FROM pizza\_sales

GROUP BY order\_id,pizza\_name

HAVING COUNT(DISTINCT pizza\_name) = COUNT(pizza\_name);



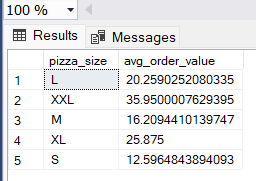
--3.Average Order Value by Pizza Size

SELECT pizza\_size, AVG(total\_price) AS avg\_order\_value

FROM pizza\_sales

WHERE YEAR(order\_date) = 2015

GROUP BY pizza\_size;



--4.Ingredient Co-Occurrence

WITH IngredientPairs AS (

SELECT i1.pizza\_ingredients AS ingredient1, i2.pizza\_ingredients AS ingredient2

FROM pizza\_sales i1

JOIN pizza\_sales i2 ON i1.order\_id = i2.order\_id AND i1.pizza\_ingredients < i2.pizza\_ingredients

)

, RankedPairs AS (

SELECT ingredient1, ingredient2, COUNT(\*) AS co\_occurrences,

ROW\_NUMBER() OVER (ORDER BY COUNT(\*) DESC) AS rownum

FROM IngredientPairs

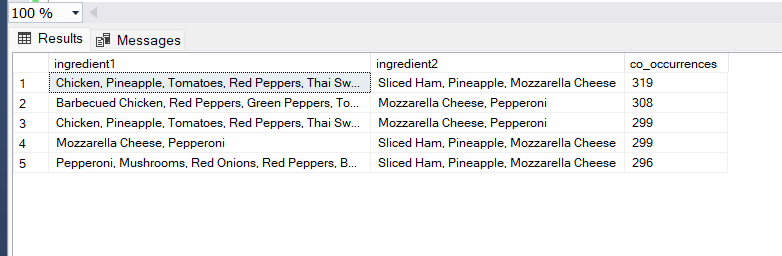
GROUP BY ingredient1, ingredient2

)

SELECT ingredient1, ingredient2, co\_occurrences

FROM RankedPairs

WHERE rownum <= 5;



--5.Revenue Growth by Month

SELECT YEAR(order\_date) AS year, MONTH(order\_date) AS month,

SUM(total\_price) AS monthly\_revenue,

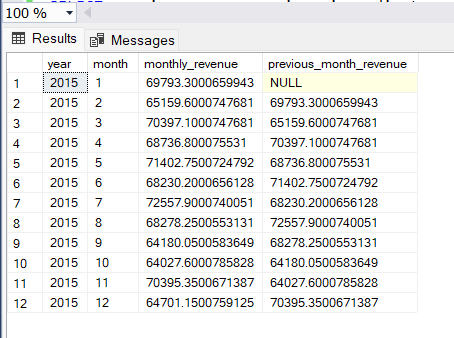
LAG(SUM(total\_price)) OVER (ORDER BY YEAR(order\_date), MONTH(order\_date)) AS previous\_month\_revenue

FROM pizza\_sales

WHERE YEAR(order\_date) = 2015

GROUP BY YEAR(order\_date), MONTH(order\_date)

ORDER BY YEAR(order\_date), MONTH(order\_date);



--6.pizza names and their ingredients for pizzas with a unit price above the average unit price for their respective categories

SELECT ps.pizza\_name, ps.pizza\_ingredients

FROM pizza\_sales AS ps

INNER JOIN (

SELECT pizza\_category, AVG(unit\_price) AS avg\_unit\_price

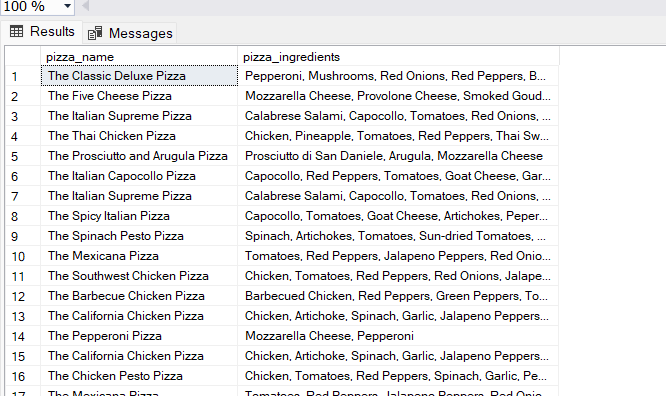
FROM pizza\_sales

GROUP BY pizza\_category

) AS avg\_prices

ON ps.pizza\_category = avg\_prices.pizza\_category

WHERE ps.unit\_price > avg\_prices.avg\_unit\_price;



--7.Determine the percentage of pizzas category wise out of all pizzas sold

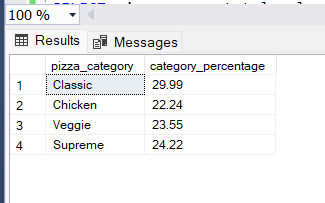
SELECT

pizza\_category,

FORMAT(COUNT(\*) \* 100.0 / (SELECT COUNT(\*) FROM pizza\_sales), '0.00') AS category\_percentage

FROM pizza\_sales

GROUP BY pizza\_category;



--8.List the top 10 pizza names with the highest total sales (quantity \* unit price)

SELECT pizza\_name, total\_sales

FROM (

SELECT

pizza\_name,

SUM(quantity \* unit\_price) AS total\_sales,

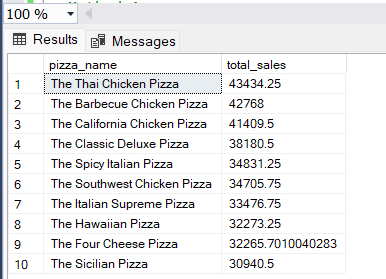
ROW\_NUMBER() OVER (ORDER BY SUM(quantity \* unit\_price) DESC) AS row\_num

FROM pizza\_sales

GROUP BY pizza\_name

) AS ranked

WHERE row\_num <= 10;



--9.Find the month with the highest total revenue

--Method 1

SELECT TOP 1

DATEPART(MONTH, order\_date) AS month,

format(SUM(total\_price),'0.00') AS total\_revenue

FROM pizza\_sales

GROUP BY DATEPART(MONTH, order\_date)

ORDER BY total\_revenue DESC;

--Method 2

SELECT TOP 1

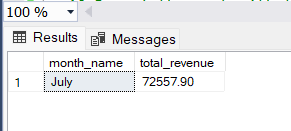
DATENAME(MONTH, order\_date) AS month\_name,

Format(SUM(total\_price),'0.00') AS total\_revenue

FROM pizza\_sales

GROUP BY DATENAME(MONTH, order\_date)

ORDER BY total\_revenue DESC;



--10.Day of the week with the highest total sales

SELECT TOP 1

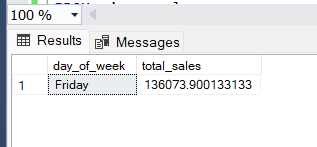
DATENAME(WEEKDAY, order\_date) AS day\_of\_week,

SUM(total\_price) AS total\_sales

FROM pizza\_sales

GROUP BY DATENAME(WEEKDAY, order\_date)

ORDER BY total\_sales DESC;



--11. Week with the highest total sales

SELECT TOP 1

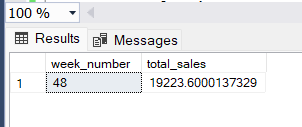
DATEPART(WK, order\_date) AS week\_number,

SUM(total\_price) AS total\_sales

FROM pizza\_sales

GROUP BY DATEPART(WK, order\_date)

ORDER BY total\_sales DESC;



--12.pizza with the highest total price in each pizza category

WITH RankedPizzas AS (

SELECT

pizza\_id,

pizza\_name,

total\_price,

pizza\_category,

ROW\_NUMBER() OVER (PARTITION BY pizza\_category ORDER BY total\_price DESC) AS category\_rank

FROM pizza\_sales

)

SELECT pizza\_id, pizza\_name, total\_price, pizza\_category

FROM RankedPizzas

WHERE category\_rank = 1;

