Pizza Sales Queries with Output 

-- 1. Retrieve the total number of orders placed.

SELECT

COUNT(order\_id) AS Total\_Orders

FROM

Orders;

Result:



-- 2. Calculate the total revenue generated from pizza sales.

SELECT

ROUND(SUM(Quantity \* pizzas.price), 2) AS Total\_Sales

FROM

Order\_details

JOIN

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

Result:



-- 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;

Result:



-- 4. Identify the most common pizza size ordered.

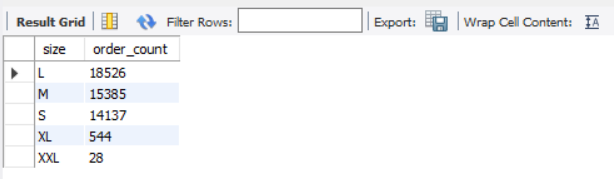
Select Pizzas.size, count(order\_details.Order\_Details\_id) as order\_count

from Pizzas join order\_details

on Pizzas.pizza\_id=order\_details.Pizza\_id

Group by Pizzas.size order by order\_count Desc;

Result:



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

SELECT

Pizza\_types.name, SUM(order\_details.Quantity) AS Quantity

FROM

Pizza\_types

JOIN

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

JOIN

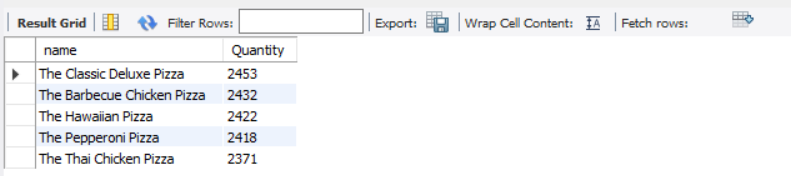
order\_details ON order\_details.pizza\_id = pizzas.pizza\_id

GROUP BY Pizza\_types.name

ORDER BY Quantity DESC

LIMIT 5;

Result:



-- 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

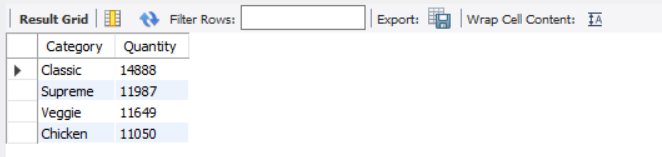
JOIN

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

GROUP BY Pizza\_types.Category

ORDER BY Quantity DESC;

Result:



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

SELECT

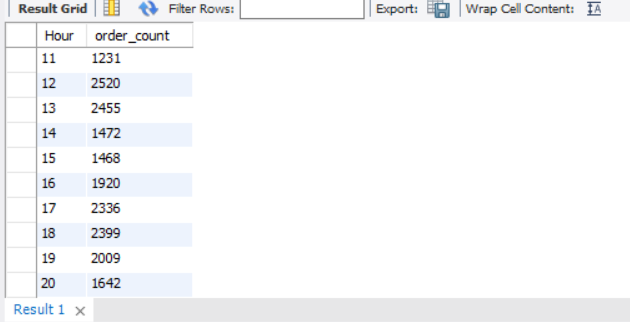
HOUR(order\_time) AS Hour, COUNT(order\_id) AS order\_count

FROM

orders

GROUP BY HOUR(order\_time);

Result:



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

SELECT

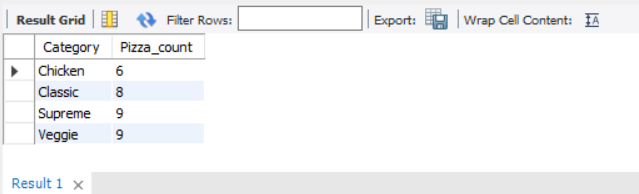
Category, COUNT(name) AS Pizza\_count

FROM

pizza\_types

GROUP BY category;

Result:



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

SELECT

ROUND(AVG(sum\_Quantity), 0) as pizzas\_ordered\_per\_day

FROM

(SELECT

Order\_date, SUM(quantity) AS sum\_Quantity

FROM

orders

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

GROUP BY Order\_date) AS Order\_Quantity;

Result:



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

JOIN

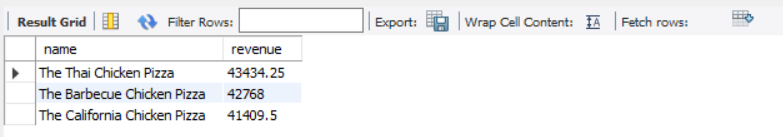
order\_details ON order\_details.Pizza\_id = pizzas.pizza\_id

GROUP BY pizza\_types.name

ORDER BY revenue DESC

LIMIT 3;

Result:



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

SELECT

pizza\_types.category,

ROUND(SUM(order\_details.Quantity \* pizzas.price) / (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) \* 100,

2) AS revenue

FROM

pizza\_types

JOIN

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

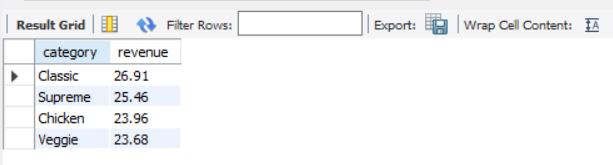
JOIN

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

GROUP BY pizza\_types.category

ORDER BY revenue DESC;

Result:



-- 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,

Round(sum(order\_details.quantity \* pizzas.price)) as Revenue

from order\_details join pizzas

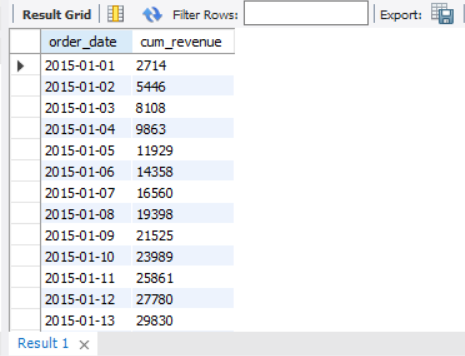
on order\_details.Pizza\_id = pizzas.pizza\_id

join orders

on orders.Order\_id = order\_details.Order\_id

group by orders.order\_date) as Sales;

Result:



-- 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 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

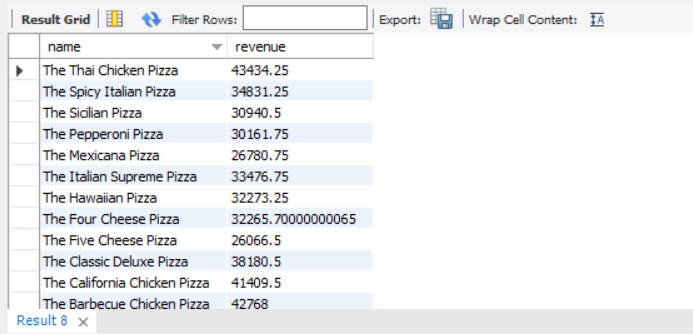
join order\_details

on order\_details.Pizza\_id = pizzas.pizza\_id

Group by pizza\_types.category, pizza\_types.name) as a) as b

where rn <= 3;

Result:

-