**USE pizzahut;**

**-- Basic:**

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

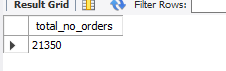
SELECT

COUNT(\*) AS total\_no\_orders

FROM

orders;

**RESULT:**

****

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

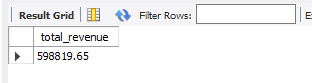
SELECT

ROUND(SUM(od.quantity\*p.price),2) AS total\_revenue

FROM order\_details od JOIN pizzas p

ON p.pizza\_id=od.pizza\_id;

**RESULT:**

****

**-- 3.Identify the highest-priced pizza.**

SELECT

pt.name, p.price

FROM

pizza\_types pt

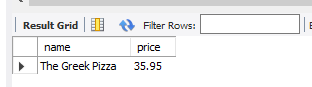
JOIN

pizzas p ON pt.pizza\_type\_id = p.pizza\_type\_id

ORDER BY p.price DESC

LIMIT 1;

**RESULT:**

****

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

SELECT

p.size, COUNT(od.order\_details\_id) order\_count

FROM

pizzas p

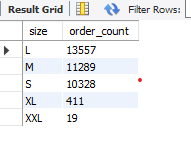
JOIN

order\_details od ON p.pizza\_id = od.pizza\_id

GROUP BY p.size

ORDER BY order\_count DESC;

**RESULT:**

****

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

SELECT

pt.name, SUM(od.quantity) Quantity

FROM

pizza\_types pt

JOIN

pizzas p ON pt.pizza\_type\_id = p.pizza\_type\_id

JOIN

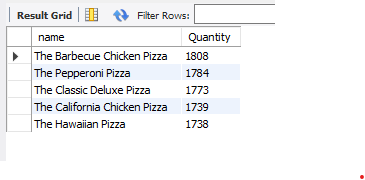
order\_details od ON od.pizza\_id = p.pizza\_id

GROUP BY pt.name

ORDER BY Quantity DESC

LIMIT 5;

**RESULT:**

****

**-- Intermediate:**

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

SELECT

pt.category, SUM(od.quantity) AS Quantity

FROM

pizza\_types pt

JOIN

pizzas p ON pt.pizza\_type\_id = p.pizza\_type\_id

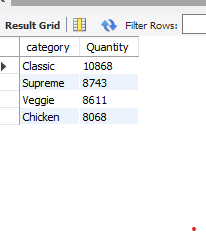
JOIN

order\_details od ON od.pizza\_id = p.pizza\_id

GROUP BY pt.category

ORDER BY Quantity DESC;

**RESULT:**

****

**-- 7.Determine the distribution of MAX 10 orders by hour of the day.**

SELECT

HOUR(order\_time) , COUNT(order\_id)

FROM

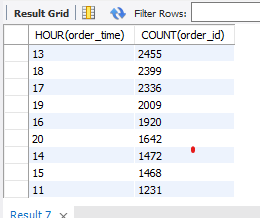
orders

GROUP BY HOUR(order\_time)

ORDER BY COUNT(order\_id) DESC

LIMIT 10 ;

**RESULT:**

****

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

SELECT

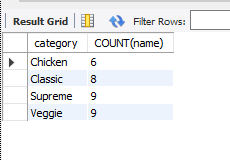
category, COUNT(name)

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(quantity), 0) avg\_pizza\_ordered\_per\_day

FROM

(SELECT

o.order\_date, SUM(od.quantity) AS quantity

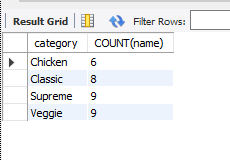
FROM

orders o

JOIN order\_details od ON o.order\_id = od.order\_id

GROUP BY o.order\_date) AS Order\_quantity;\

**RESULT:**

****

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

SELECT

pt.name, SUM(od.quantity \* p.price) AS revenue

FROM

pizza\_types pt

JOIN

pizzas p ON pt.pizza\_type\_id = p.pizza\_type\_id

JOIN

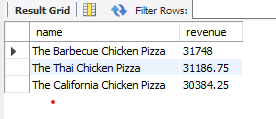
order\_details od ON od.pizza\_id = p.pizza\_id

GROUP BY pt.name

ORDER BY revenue DESC

LIMIT 3;

**RESULT:**

****

**-- Advanced:**

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

SELECT

pt.category,

CONCAT( ROUND(SUM(od.quantity \* p.price) / (SELECT

ROUND(SUM(od.quantity \* p.price), 2) AS total\_sales

FROM

order\_details od

JOIN

pizzas p ON p.pizza\_id = od.pizza\_id) \* 100, 2),"%") AS revenue

FROM

pizza\_types pt

JOIN

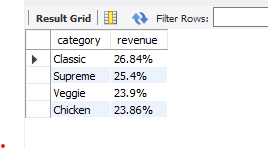
pizzas p ON pt.pizza\_type\_id = p.pizza\_type\_id

JOIN order\_details od ON od.pizza\_id=p.pizza\_id

GROUP BY pt.category

ORDER BY revenue DESC;

**RESULT:**

****

**-- 12.Analyze the cumulative revenue generated over time.**

WITH salesCTE AS

(

SELECT o.order\_date AS order\_date,

SUM(od.quantity \* p.price) AS revenue

FROM order\_details od JOIN pizzas p

ON od.pizza\_id=p.pizza\_id

JOIN orders o

on o.order\_id=od.order\_id

GROUP BY o.order\_date

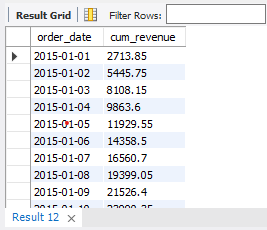
)

SELECT order\_date,

ROUND(SUM(revenue)over(ORDER BY order\_date) ,2)AS cum\_revenue

FROM salesCTE;

**RESULT:**

****

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

WITH CTE AS

(

SELECT

pt.category,pt.name,

ROUND(SUM((od.quantity\*p.price)),2)AS revenue

FROM pizza\_types pt JOIN pizzas p

ON pt.pizza\_type\_id=p.pizza\_type\_id

JOIN

order\_details od

ON od.pizza\_id=p.pizza\_id

GROUP BY pt.category,pt.name

)

SELECT category,name,revenue

FROM

(

SELECT category,name,revenue,

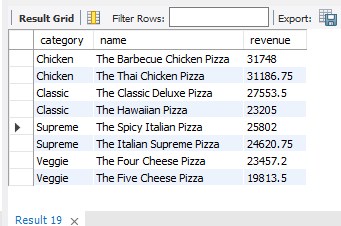
RANK() OVER (PARTITION BY category ORDER BY revenue DESC) AS rn

FROM CTE) AS a

WHERE rn <3

order by category;

**RESULT:**

****