🍕 Pizza Sales(SQL) Performance Report

**A. KPI’s**

**1. Total Revenue:**

select

round(sum(orders\_detail.quantity\*pizzas.price)::NUMERIC,2 )as revenue

from

orders\_detail

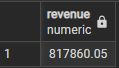
join

pizzas

on

orders\_detail.pizza\_id = pizzas.pizza\_id

OUTPUT:



**2. Average Order Per Day**

select

round(avg(quantity),0) as average\_pizzas\_orderd\_per\_day

from

(select orders.date,sum(orders\_detail.quantity) as quantity

from orders join orders\_detail

on orders.order\_id = orders\_detail.order\_id

group by orders.date) as order\_quantity;

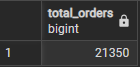
OUTPUT:



**3. Total Pizzas Orderd**

select count(order\_id) as total\_orders from orders

OUTPUT:



**4. Top 3 Most Ordered Pizzas**

SELECT

pizza\_name,

quantity

FROM (

SELECT

pizza\_type.pizza\_name,

SUM(orders\_detail.quantity) AS quantity,

RANK() OVER (ORDER BY SUM(orders\_detail.quantity) DESC) AS rnk

FROM

pizza\_type

JOIN

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

JOIN

orders\_detail ON orders\_detail.pizza\_id = pizzas.pizza\_id

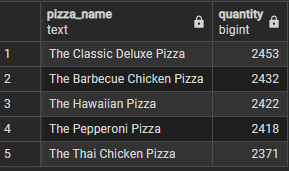
GROUP BY

pizza\_type.pizza\_name

) AS ranked\_pizzas

WHERE

rnk <= 3;



**5. Highest-Priced Pizza.**

select

pizza\_type.pizza\_name, pizzas.price

from

pizza\_type

join

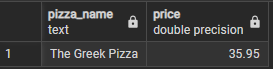
pizzas

on

pizza\_type.pizza\_type\_id = pizzas.pizza\_type\_id

where

pizzas.price= (select max(pizzas.price) from pizzas)



**6. Most common pizza size orderd.**

select

pizzas.size,count(orders\_detail.order\_id) as order\_count

from

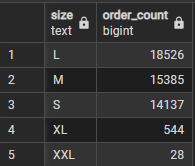
pizzas join orders\_detail

on

pizzas.pizza\_id=orders\_detail.pizza\_id

group by

pizzas.size order by order\_count desc



**7. Total quantity of each pizza category ordered.**

SELECT

pizza\_type.category,

SUM(orders\_detail.quantity) AS total\_quantity

FROM

orders\_detail

JOIN

pizzas ON orders\_detail.pizza\_id = pizzas.pizza\_id

JOIN

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

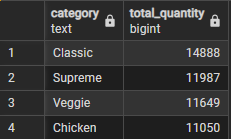
GROUP BY

pizza\_type.category

ORDER BY

total\_quantity DESC

OUTPUT:



**B. Daily Trend for Orders (distribution of orders by hour of the day).**

SELECT

EXTRACT(HOUR FROM time) AS hour,

COUNT(order\_id) AS order\_count

FROM

orders

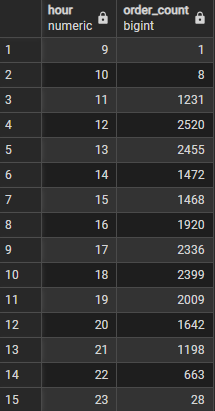
GROUP BY

EXTRACT(HOUR FROM time)

ORDER BY

hour;

OUTPUT:



**C. % of Sales by Pizza Type**

select pizza\_type.category,

round(sum(orders\_detail.quantity\*pizzas.price):: NUMERIC /(select

round(sum(orders\_detail.quantity\*pizzas.price)::NUMERIC, 2) as total\_sales

from orders\_detail

join pizzas on pizzas.pizza\_id = orders\_detail.pizza\_id)\*100,2) as revenue

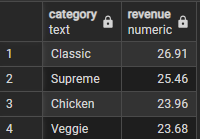
from pizza\_type join pizzas on

pizza\_type.pizza\_type\_id = pizzas.pizza\_type\_id

join orders\_detail on

orders\_detail.pizza\_id = pizzas.pizza\_id

group by pizza\_type.category order by revenue desc



**D. The Cumulative Revenue Generated Over Time.**

SELECT

date,

revenue,

SUM(revenue) OVER (ORDER BY date) AS cum\_revenue

FROM (

SELECT

orders.date,

SUM(orders\_detail.quantity \* pizzas.price) AS revenue

FROM

orders\_detail

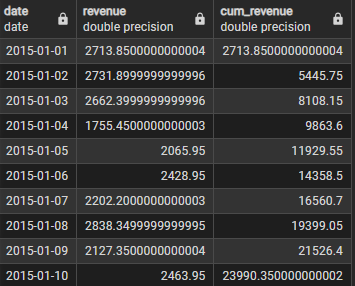
JOIN pizzas ON orders\_detail.pizza\_id = pizzas.pizza\_id

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

GROUP BY orders.date

) AS sales;

OUTPUT:



**E. Top 3 most ordered pizza types based on revenue for each pizza category.**

SELECT pizza\_name, revenue

FROM (

SELECT category, pizza\_name, revenue,

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

FROM (

SELECT pizza\_type.category, pizza\_type.pizza\_name,

SUM(orders\_detail.quantity \* pizzas.price) AS revenue

FROM pizza\_type

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

JOIN orders\_detail ON orders\_detail.pizza\_id = pizzas.pizza\_id

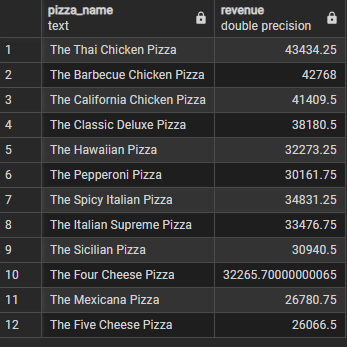
GROUP BY pizza\_type.category, pizza\_type.pizza\_name

) AS a

) AS b

WHERE rn <= 3;

OUTPUT:

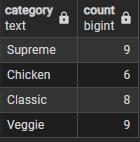


**F. The Category-wise Distribution of Pizzas.**

select category, count(pizza\_name) from pizza\_type

group by category

OUTPUT:



**B. Analyze Monthly Sales Performance and Track Growth Trends.**

SELECT

DATE\_TRUNC('month', orders.date) AS month,

ROUND(SUM(orders\_detail.quantity \* pizzas.price)::NUMERIC, 2) AS monthly\_sales,

LAG(ROUND(SUM(orders\_detail.quantity \* pizzas.price)::NUMERIC,2)) OVER (ORDER BY DATE\_TRUNC('month', orders.date)) AS prev\_month\_sales,

ROUND(

(ROUND(SUM(orders\_detail.quantity \* pizzas.price)::NUMERIC,2) -

LAG(ROUND(SUM(orders\_detail.quantity \* pizzas.price)::NUMERIC, 2)) OVER (ORDER BY DATE\_TRUNC('month', orders.date)))

, 2

) AS month\_on\_month\_change

FROM

orders

JOIN

orders\_detail ON orders.order\_id = orders\_detail.order\_id

JOIN

pizzas ON orders\_detail.pizza\_id = pizzas.pizza\_id

GROUP BY

DATE\_TRUNC('month', orders.date)

ORDER BY

month;

OUTPUT:

