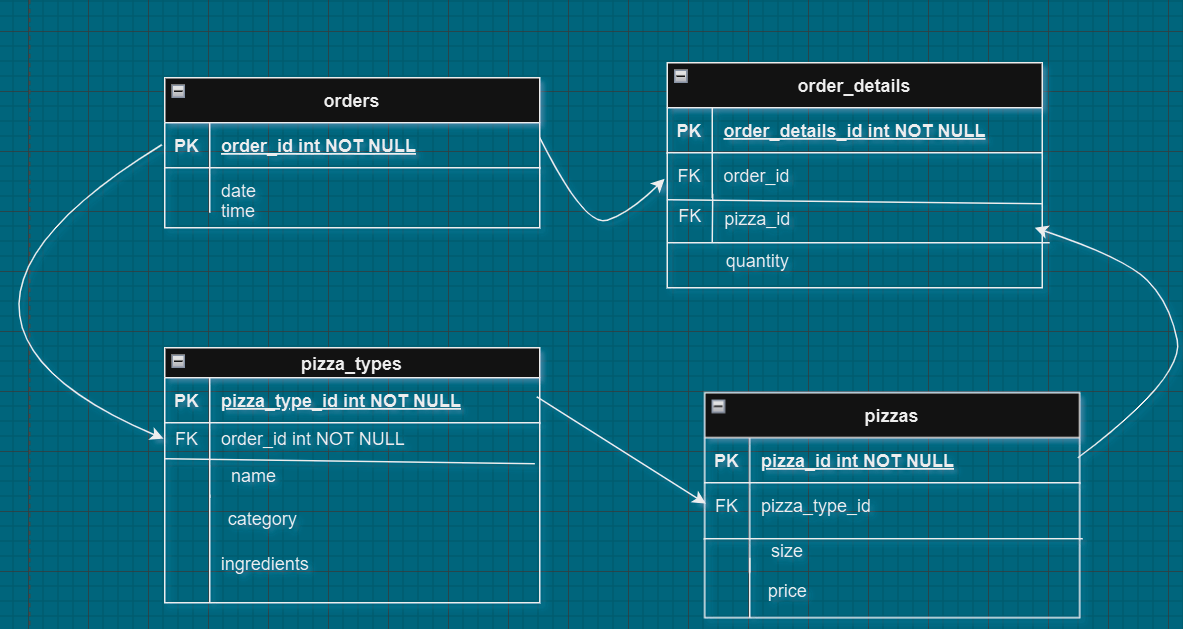
**ER Diagram :**

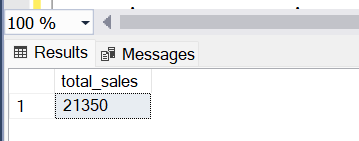
****

**SQL QUERIES**

**Basic:**

1. Retrieve the total number of orders placed.

select count(order\_id) as total\_sales from orders;

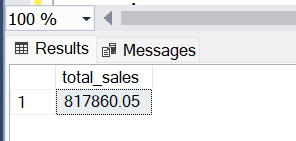


1. Calculate the total revenue generated from pizza sales.

select round(sum(o.quantity\*p.price),2) as total\_sales

from

order\_details o join pizzas p on o.pizza\_id = p.pizza\_id;



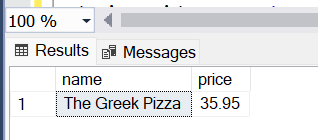
1. Identify the highest-priced pizza.

SELECT TOP 1 pt.name, round(p.price,2) as price

FROM pizza\_types pt

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

ORDER BY p.price DESC;



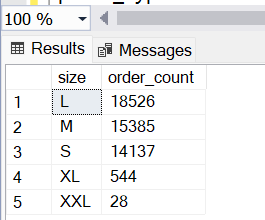
1. Identify the most common pizza size ordered.

select p.size, count(od.order\_details\_id) as 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;



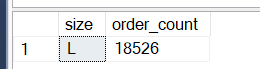
OR

select top 1 p.size, count(od.order\_details\_id) as

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;



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

select top 5 pizza\_types.name, sum(order\_details.quantity)

as quantity from pizza\_types

join

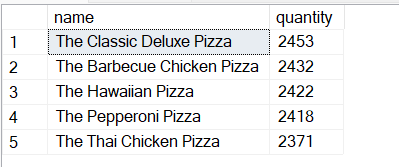
pizzas on pizza\_types.pizza\_type\_id = pizzas.pizza\_type\_id

join

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

group by pizza\_types.name

order by quantity desc;



**Intermediate :**

1. Join the necessary tables to find the total quantity of each pizza category ordered.

select pizza\_types.category, sum(order\_details.quantity)

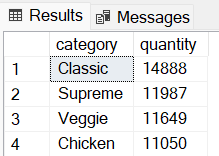
as quantity from pizza\_types

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;

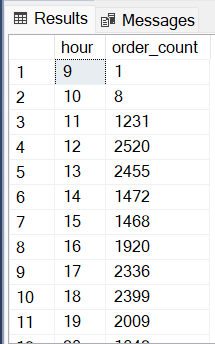


1. Determine the distribution of orders by hour of the day.

SELECT DATEPART(HOUR, time) as hour, count(order\_id) as

order\_count FROM orders group by DATEPART(HOUR, time)

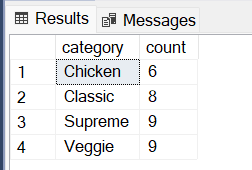
order by DATEPART(HOUR, time);



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

select category, count(name) as count from pizza\_types

group by category;



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

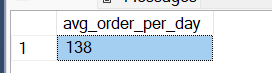
select avg(quantity) as avg\_order\_per\_day from

(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 pizza types based on revenue.

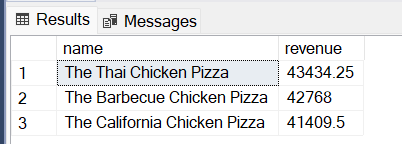
select top 3 pizza\_types.name, sum(order\_details.quantity\*pizzas.price) as revenue from

pizza\_types join pizzas on

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

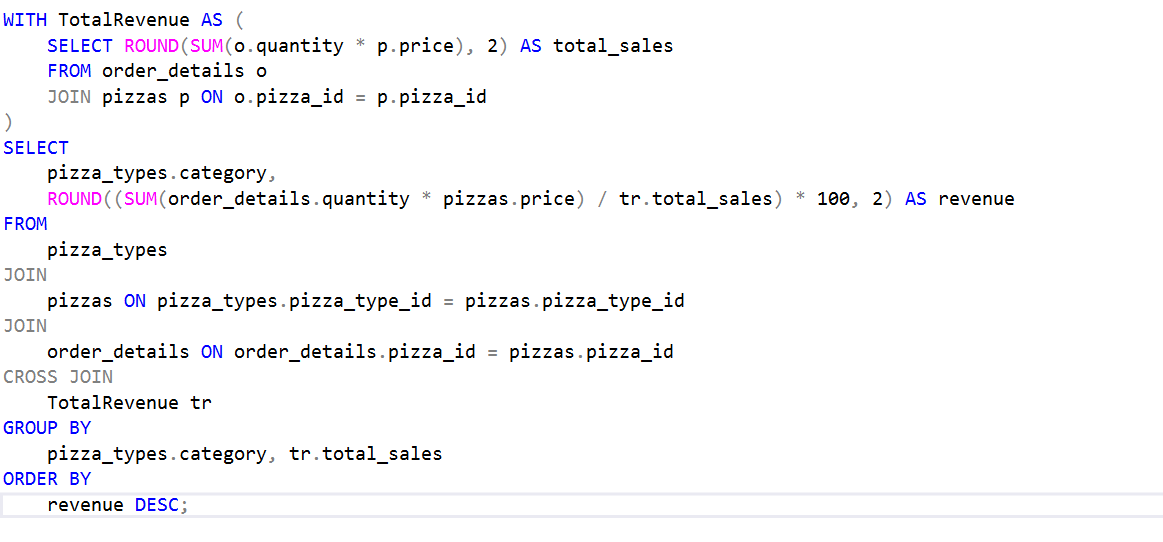
join order\_details on order\_details.pizza\_id = pizzas.pizza\_id

group by pizza\_types.name order by revenue desc;



**Advanced:**

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



OR

select pizza\_types.category,

round((sum(order\_details.quantity\*pizzas.price) /

(select round(sum(o.quantity\*p.price),2) as total\_sales

from order\_details o join pizzas p on

o.pizza\_id = p.pizza\_id))\*100,2)

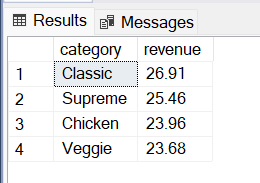
as revenue from

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;



1. Analyze the cumulative revenue generated over time.

select date, round(sum(revenue) over (order by date),2)

as cumulative\_revenue from

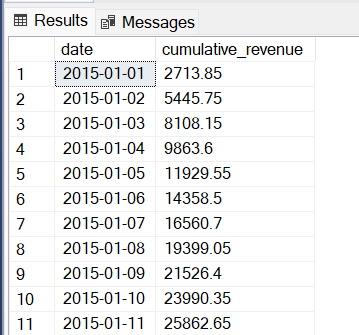
(select orders.date, sum(order\_details.quantity\*pizzas.price)

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.date) as sales;



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

select category, 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 ;

