Pizza Sale Report





My name is NISHA RANA, and i have been using SQL Queries to solve question relation pizza set for this project



Questions

- Q1. Retrieve the total number of order placed.
- Q2. Calculate the total revenue generated from pizza sales.
- Q3. Identify the highest-priced pizza.
- Q4. Identify the most common pizza size ordered.
- Q5. List the top 5 most ordered pizza type, along with their quantities.
- Q6. join the necessary tables to find the total quantity of each pizza category ordered.
- Q7. Determine the distribution of orders by hour of the day
- Q8. Join relevant tables to find the category-wise distruibution of pizzas
- Q9. Group the orders by date and calculate the average number of pizzas orderd per day
- Q10. Determine the top 3 most ordered pizza type based on revenue
- Q11. Calculate the percentage contribution of each pizza type to total revenue
- Q12. analyze the cumulative revenue generated over time
- Q13. Determine the top 3 most ordered pizza type based on revenue for each pizza category

Q1. retrieve the total number of order placed.

```
SELECT

COUNT(order_id) AS total_order

FROM

orders;
```







Q2. Calculate the total revenue generated from pizza sales.

```
SELECT

ROUND(SUM(order_details.quantity * pizzas.price),

2) AS total_revenue

FROM

order_details

JOIN

pizzas ON pizzas.pizza_id = order_details.pizza_id;
```



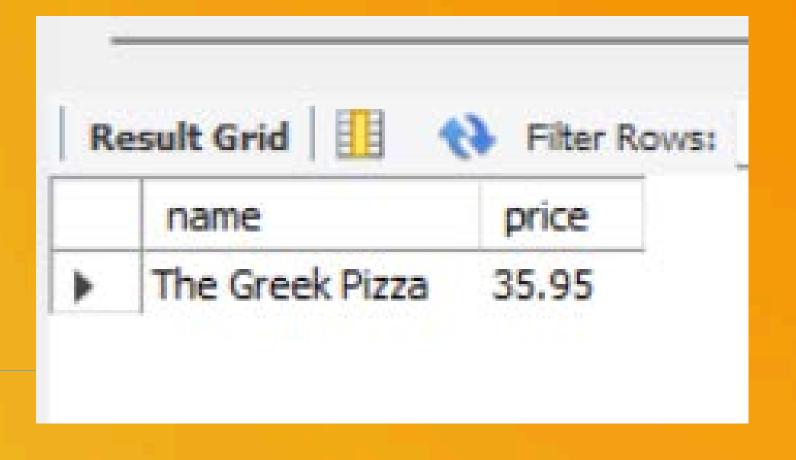


Q3. Identify the highest-priced pizza.

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;





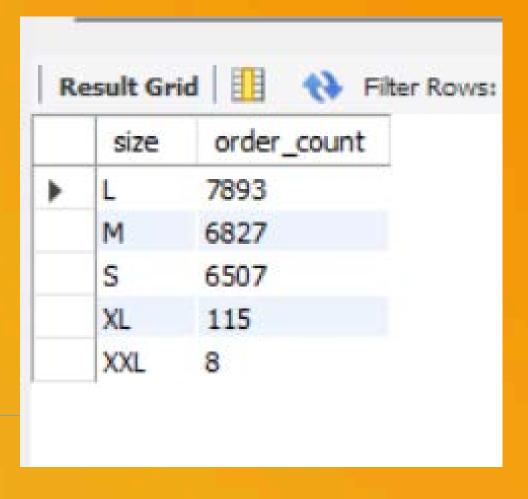


Q4. Identify the most common pizza size ordered.

```
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;
```







Q5. List the top 5 most ordered pizza type, 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

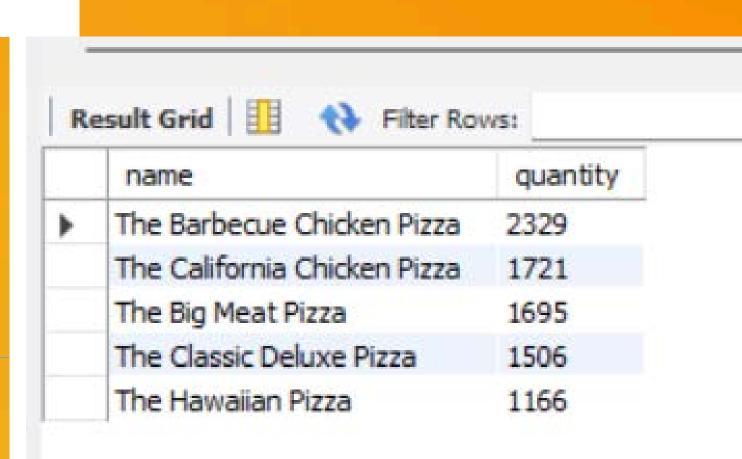
ON order_details.pizza_id = pizzas.pizza_id

GROUP BY pizza_types.name

ORDER BY quantity DESC

LIMIT 5;







Q6. 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;
```





	category	quantity
>	Classic	6975
	Chicken	6238
	Veggie	4357
	Supreme	4134

Q7. 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);



Re	esult Gri	d 🔠 🙌 Filter
	hour	order_count
>	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8 8
	9	1



Q8.Join relevant tables to find the category-wise distruibution of pizzas

select category, count(name)
 from pizza_types
 group by category;



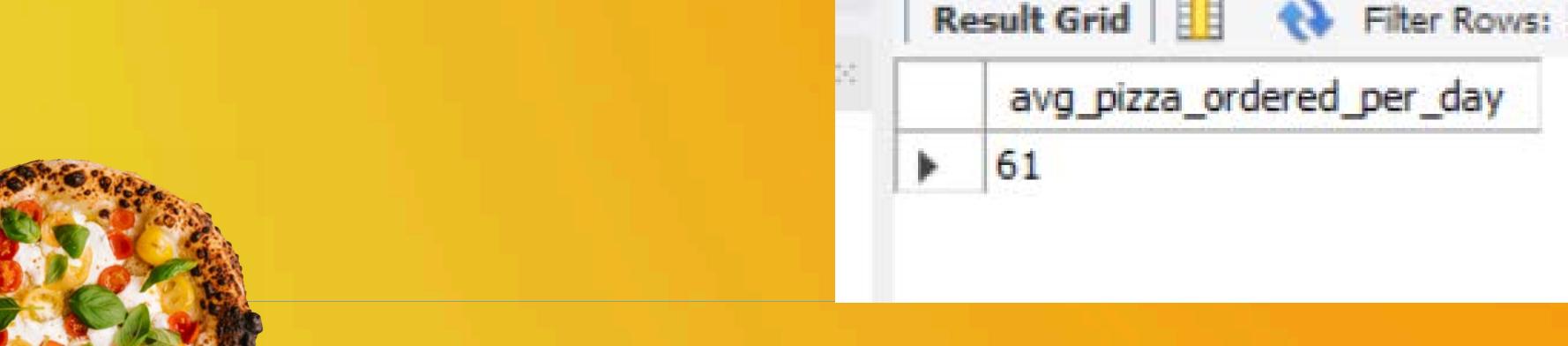


	category	count(name)
>	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Q9. Group the orders by date and calculate the average number of pizzas orderd per day

- SELECT ROUND(AVG(quantity), 0) avg_pizza_ordered_per_day FROM
- FROM orders JOIN order_details ON orders.order_id = order_details.order_id GROUP BY orders.order_date) AS order_quantity;







Q10. Determine the top 3 most ordered pizza type 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;
```





	name	revenue
>	The Barbecue Chicken Pizza	41230.75
	The California Chicken Pizza	30102.75
	The Classic Deluxe Pizza	23548

Q11. 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_sale
    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;
```



	category	revenue
Þ	Chicken	30.8
	Classic	28.06
	Supreme	20.63
	Veggie	20.51



Q12. 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, 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;





	order_date	cum_revenue	
•	2015-01-01	1136.3500000000001	
	2015-01-02	2245.8500000000004	
	2015-01-03	3374.8500000000004	
	2015-01-04	4260.150000000001	
	2015-01-05	5159.1	
	2015-01-06	6232.3	
	2015-01-07	7195.25	
	2015-01-08	8352.4	
	2015-01-09	9386.85	
	2015-01-10	10501.15	
	2015-01-11	11341.94999999999	
	2015-01-12	12223.74999999998	
	2015-01-13	13031.64999999998	
	2015-01-14	14148.64999999998	
	2015-01-15	15126.69999999997	
	2015-01-16	16257.89999999998	
	2015-01-17	17215.449999999997	
	2015-01-18	18064.399999999998	
	2015-01-19	18961.749999999996	
	2015-01-20	20071.749999999996	
	2015-01-21	20859.84999999995	
	2015-01-22	21960.59999999999	
	2015-01-23	23103.749999999996	
	2015-01-24	24114.549999999996	

Q13. Determine the top 3 most ordered pizza type 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;
```





	name	revenue	
•	The Barbecue Chicken Pizza	41230.75	-0.00 -0.00
	The California Chicken Pizza	30102.75 11606	
	The Chicken Alfredo Pizza		
	The Classic Deluxe Pizza	23548	
	The Big Meat Pizza	20340	
	The Hawaiian Pizza	15546.5	
	The Italian Supreme Pizza	14359.75	
	The Calabrese Pizza	12463.5	
	The Brie Carre Pizza	9057.949999999937	
	The Four Cheese Pizza	17526.550000000232	
	The Five Cheese Pizza	15299.5	
	The Mexicana Pizza	9428.5	

Thank You

