PIZZA SALES- SQL PROJECT

Welcome to pizza sales analysis presentation! Using SQL, I have delved the queries across three levels:



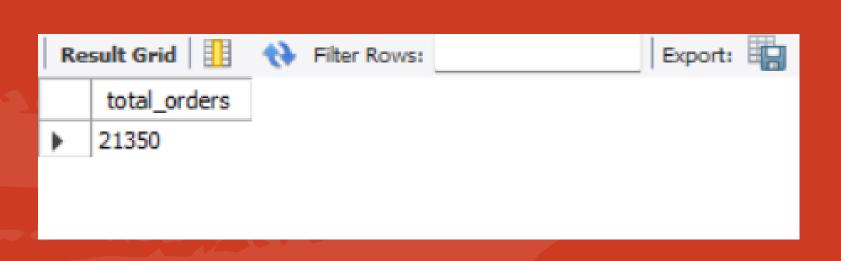
Basic: We'll cover total orders, revenue, the highest-priced pizza, the most common size, and the top 5 pizza types by quantity.

Intermediate: We've examined quantities by category, hourly order distribution, category-wise pizza distribution, average daily orders, and the top 3 pizza types by revenue.

Advanced: We'll explore each pizza type's revenue contribution, cumulative revenue over time, and the top 3 pizza types by revenue within each category.

Retrieve the total number of orders placed

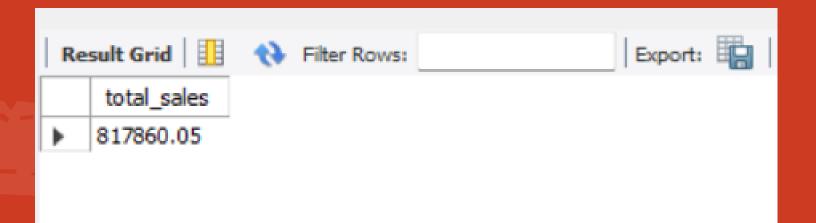
```
3
4 • SELECT
5          COUNT(order_id) AS total_orders
6          FROM
7          orders;
8
```



Calculate the total revenue generated from pizza sales.

```
select
       sum(order_details.quantity * pizzas.price) as total_sales
       from order_details join pizzas
       on pizzas.pizza_id = order_details.pizza_id;
       -- roundoff
       SELECT
           ROUND(SUM(order_details.quantity * pizzas.price),
12
                   2) AS total_sales
13
       FROM
           order_details
14
               JOIN
15
           pizzas ON pizzas.pizza_id = order_details.pizza_id
16
```

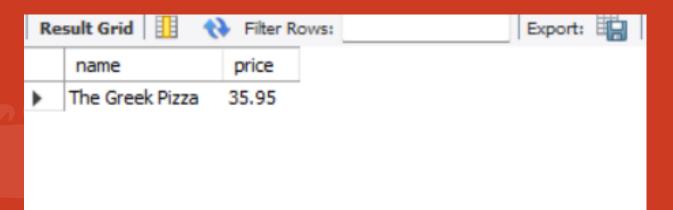




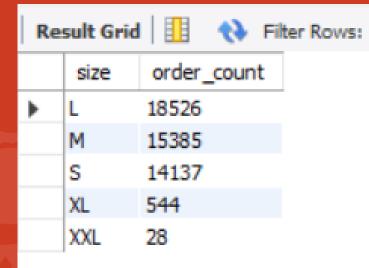
Identify the highest-priced pizza.

```
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:
```

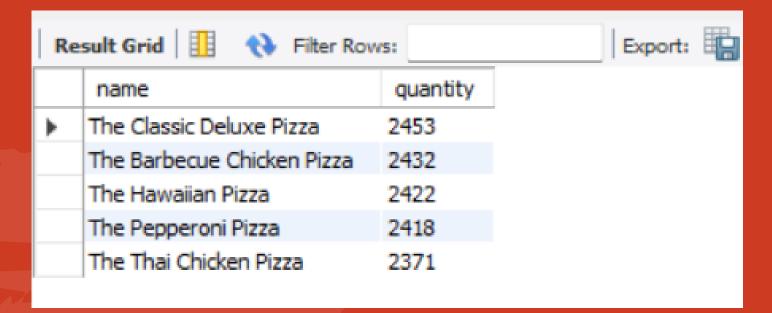


Identify the most common pizza size ordered.



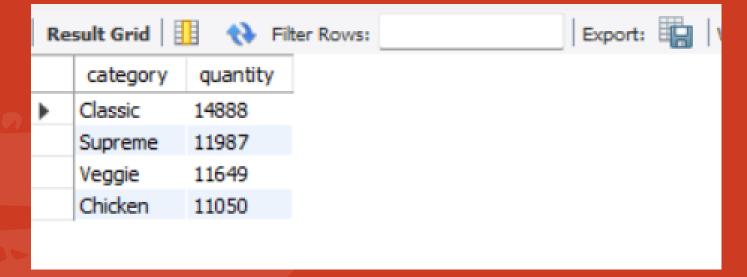
List the top 5 most ordered pizza types along with their quantities.

```
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;
```



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;
```



Determine the distribution of orders by hour of the day.

```
SELECT

HOUR(order_time) AS hour, COUNT(order_id) AS order_count

FROM

orders AS order_count

GROUP BY HOUR(order_time);
```



Res	sult Grid	Filter Rows:	Export:	
	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		

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

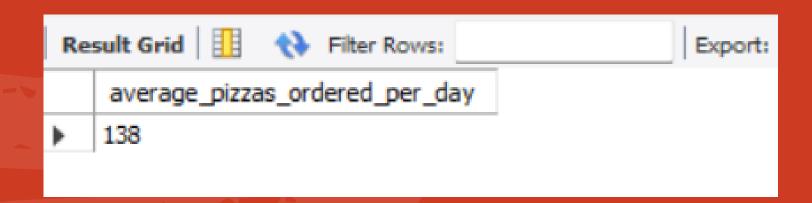
select category , count(name) from pizza_types
group by category



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

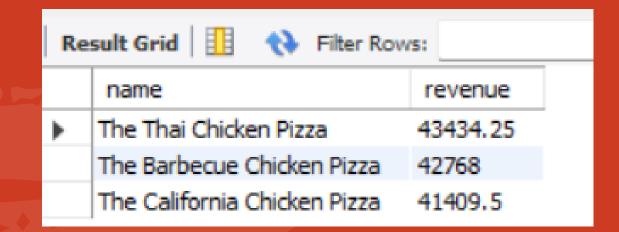
Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT
    round(AVG(quantity),0) as average_pizzas_ordered_per_day
FROM
    (SELECT
         orders.order_date, SUM(order_details.quantity) AS quantity
FROM
         orders
         JOIN order_details ON orders.order_id = order_details.order_id
         GROUP BY orders.order_date) AS order_quantity;
```



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 pizzas.pizza_type_id = pizza_types.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;
```



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

```
pizza_types.category,

(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) as revenue

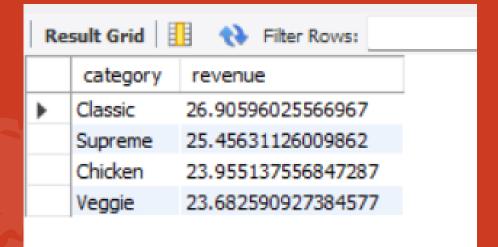
from pizza_types JOIN

pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id JOIN

order_details ON order_details.pizza_id = pizzas.pizza_id

GROUP BY pizza_types.category

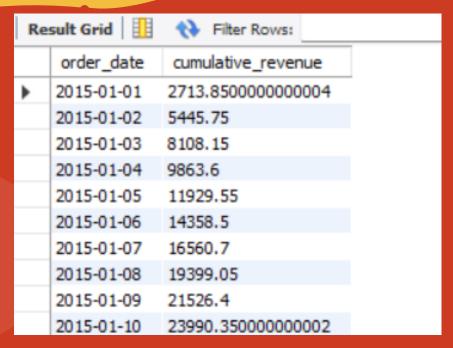
ORDER BY revenue DESC;
```



Analyze the cumulative revenue generated over time.

```
select order_date,
sum(revenue) over (order by order_date) as cumulative_revenue from

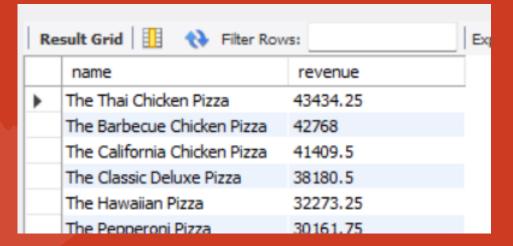
(SELECT orders.order_date,
SUM(order_details.quantity * pizzas.price) AS revenue FROM order_details JOIN
pizzas ON order_details.pizza_id = pizzas.pizza_id JOIN
orders ON orders.order_id = order_details.order_id
GROUP BY orders.order_date) as sales;
```



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;</pre>
```





Thank You!



Email akankshalanghani13@gmail.com



LinkedIn@AakankshaLanghani



GitHubhttps://github.com/AakankshaLanghani

