

Delicious **PIZZA**

This is a Pizza Sales project, where I have utilised SQL queries to address various questions related to pizza sales





Retrieve the total number of orders placed.

```
SELECT  
    COUNT(order_id) AS Total_Orders  
FROM  
    orders;
```

Result Grid	
	Total_Orders
▶	21350

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 order_details.pizza_id = pizzas.pizza_id;
```

Result Grid			
	Total_Revenue		
▶	817860.05		



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 10;
```

Result Grid				 Filter Rows:
	name	price		
▶	The Greek Pizza	35.95		
	The Greek Pizza	25.5		
	The Brie Carre Pizza	23.65		
	The Italian Vegetables Pizza	21		
	The Barbecue Chicken Pizza	20.75		
	The Spinach Supreme Pizza	20.75		
	The Italian Supreme Pizza	20.75		
	The California Chicken Pizza	20.75		
	The Thai Chicken Pizza	20.75		
	The Spinach Pesto Pizza	20.75		



Identify the most common pizza size ordered.

```
SELECT
    pizzas.size AS Pizza_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;
```

Result Grid   Filter Rows:		
	Pizza_Size	Order_Count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



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

```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS Total_Order
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 Total_Order DESC
LIMIT 5;
```

Result Grid				 Filter Rows:	
	name	Total_Order			
▶	The Classic Deluxe Pizza	2453			
	The Barbecue Chicken Pizza	2432			
	The Hawaiian Pizza	2422			
	The Pepperoni Pizza	2418			
	The Thai Chicken Pizza	2371			



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

```
select pizza_types.category, sum(order_details.quantity) as Total_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 Total_Quantity desc;
```

Result Grid				 Filter Rows:
	category	Total_Quantity		
▶	Classic	14888		
	Supreme	11987		
	Veggie	11649		
	Chicken	11050		



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) order by count(order_id) desc;
```

Result Grid					Filter
	Hour	Order_Count			
▶	12	2520			
	13	2455			
	18	2399			
	17	2336			
	19	2009			
	16	1920			
	20	1642			
	14	1472			
	15	1468			
	11	1231			
	21	1198			
	22	663			
	23	28			
	10	8			
	9	1			

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

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

Result Grid				 Filter Results
	category	count(name)		
▶	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(Total_Order), 0) AS Avg_Order
FROM
    (SELECT
        orders.order_date AS Order_Date,
        SUM(order_details.quantity) AS Total_Order
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY Order_Date) AS order_quantity;
```

Result Grid	
	Avg_Order
▶	138

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 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;
```

Result Grid				 Filter Rows:
	name	Revenue		
▶	The Thai Chicken Pizza	43434.25		
	The Barbecue Chicken Pizza	42768		
	The California Chicken Pizza	41409.5		



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_Revenue
FROM
    order_details
    JOIN
    pizzas ON order_details.pizza_id = pizzas.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;
```

Result Grid			Filter
	category	revenue	
▶	Classic	26.91	
	Supreme	25.46	
	Chicken	23.96	
	Veggie	23.68	

Analyze the cumulative revenue generated over time.

```
select order_date,  
round(sum(Revenue) over(order by order_date), 2) 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;
```

Result Grid   Filter Rows:		
	order_date	Cum_Revenue
▶	2015-01-01	2713.85
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4
	2015-01-10	23990.35
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.3
	2015-01-14	32358.7
	2015-01-15	34343.5
	2015-01-16	36937.65
	2015-01-17	39001.75
	2015-01-18	40978.6
	2015-01-19	43365.75

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

```
select category, name, revenue, rn from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select pizza_types.category, pizza_types.name,
round(sum(order_details.quantity*pizzas.price),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, pizza_types.name) as A) as B
where rn <=3;
```

Result Grid					Filter Rows:	Export:
	category	name	revenue	rn		
▶	Chicken	The Thai Chicken Pizza	43434.25	1		
	Chicken	The Barbecue Chicken Pizza	42768	2		
	Chicken	The California Chicken Pizza	41409.5	3		
	Classic	The Classic Deluxe Pizza	38180.5	1		
	Classic	The Hawaiian Pizza	32273.25	2		
	Classic	The Pepperoni Pizza	30161.75	3		
	Supreme	The Spicy Italian Pizza	34831.25	1		
	Supreme	The Italian Supreme Pizza	33476.75	2		
	Supreme	The Sicilian Pizza	30940.5	3		
	Veggie	The Four Cheese Pizza	32265.7	1		
	Veggie	The Mexicana Pizza	26780.75	2		
	Veggie	The Five Cheese Pizza	26066.5	3		