



# BORCELLE

## P I Z Z A

*Sales Report Project*







BORCELLE  
P I Z Z A

# ABOUT

Hello everyone, my name is Trisha, and I am a Computer Science graduate with a strong interest in business analytics and data-driven decision-making. Today, I am excited to present my project on analyzing pizza sales using SQL





# OBJECTIVE

In this project, I aimed to address several key questions related to our pizza sales data. Here's a brief overview of my approach and the solutions I developed







## Total number of orders placed

```
SELECT  
    COUNT(ord_id) AS total_orders  
FROM  
    orders
```

## Solution

Result Grid			
	total_orders		
▶	21350		





Total revenue generated from pizza sales

```
SELECT  
    ROUND(SUM(ord_dets.quantity * pizzas.price), 2) AS total_sales  
FROM  
    ord_dets  
    JOIN  
    pizzas ON pizzas.pizza_id = ord_dets.pizza_id;
```



Solution

Result Grid	
	total_sales
▶	817860.05





## 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;
```

## Solution

Result Grid			Filter Rows:
	name	price	
▶	The Greek Pizza	35.95	





## The most common pizza size ordered

```
SELECT
    pizzas.size, COUNT(ord_dets.ord_det_id) AS ord_count
FROM
    pizzas
    JOIN
        ord_dets ON pizzas.pizza_id = ord_dets.pizza_id
GROUP BY pizzas.size
ORDER BY ord_count DESC;
```



## Solution

Result Grid			Filter R
	size	ord_count	
▶	L	18526	
	M	15385	
	S	14137	
	XL	544	
	XXL	28	





## Top 5 most ordered pizza types along with their quantities

```
SELECT
    pizza_types.name, SUM(ord_dets.quantity) AS quantity
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    ord_dets ON ord_dets.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

## Solution

	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371





Find the total quantity of each pizza category ordered

```
SELECT
    pizza_types.category, SUM(ord_dets.quantity) AS quantity
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    ord_dets ON ord_dets.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

Solution

	category	quantity
►	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050





## Distribution of orders by hour of the day

```
SELECT  
    HOUR(ord_time) AS hour, COUNT(ord_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(ord_time);
```

## Solution

	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
	9	1





Find the category-wise distribution of pizzas

```
SELECT  
|   category, COUNT(name)  
FROM  
   pizza_types  
GROUP BY category;
```

Solution

	category	COUNT(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9





Calculate the average number of pizzas ordered per day

```
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_ord_per_day
FROM
    (SELECT
        orders.ord_date, SUM(ord_dets.quantity) AS quantity
    FROM
        orders
    JOIN ord_dets ON orders.ord_id = ord_dets.ord_id
    GROUP BY orders.ord_date) AS ord_quant;
```

Solution

	avg_pizza_ord_per_day
▶	138





## Top 3 most ordered pizza types based on revenue

```
SELECT
    pizza_types.name,
    SUM(ord_dets.quantity * pizzas.price) AS revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
    ord_dets ON ord_dets.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

## Solution

	name	revenue
►	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5





## Contribution of each pizza type to total revenue

```
SELECT
    pizza_types.category,
    ROUND(SUM(ord_dets.quantity * pizzas.price) / (SELECT
        ROUND(SUM(ord_dets.quantity * pizzas.price), 2) AS total_sales
    FROM
        ord_dets
    JOIN
        pizzas ON pizzas.pizza_id = ord_dets.pizza_id) * 100,
    2) AS revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    ord_dets ON ord_dets.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

## Solution

	category	revenue
►	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68





## The cumulative revenue generated over time

```
select ord_date,  
sum(revenue) over (order by ord_date) as cum_revenue  
from  
(select orders.ord_date,  
sum(ord_dets.quantity*pizzas.price) as revenue  
from ord_dets join pizzas  
on ord_dets.pizza_id = pizzas.pizza_id  
join orders  
on orders.ord_id=ord_dets.ord_id  
group by orders.ord_date) as sales;
```

## Solution

ord_date	cum_revenue
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.500000000001
2015-01-16	36937.650000000001
2015-01-17	39001.750000000001
2015-01-18	40978.600000000006
2015-01-19	43365.750000000001





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((ord_dets.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join ord_dets
on ord_dets.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn <= 3;
```

Solution

	name	revenue
►	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Hawaiian Pizza	32273.25
	The Pepperoni Pizza	30161.75





# CONCLUSION

Through this project, I demonstrated my ability to leverage SQL for in-depth data analysis, providing valuable insights that can drive strategic business decisions.

The results not only help us understand current sales trends but also highlight areas for growth and improvement.





BORCELLE  
P I Z Z A

THANK YOU

