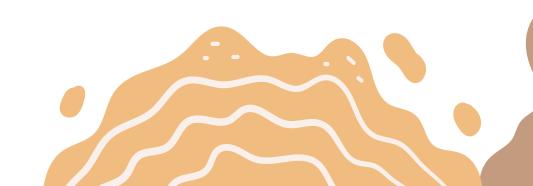


#### INSIGHTS

Over 95% of orders come from three popular pizza sizes, with XXL and XS sizes contributing to less than 2%. Focusing on high-demand sizes can reduce waste by 20% and free up storage space, optimizing operations.

• The top 5 pizzas, out of 32 options, account for 24% of total sales, indicating strong customer preference for a limited selection. Focusing marketing on these top pizzas could increase revenue by 10-15% and reduce costs tied to less popular options.

The four pizza categories–Classic, Veggie, Supreme, and Chicken–each contribute 23-27% to revenue, supporting a well-balanced menu that drives revenue stability.



#### SHOW NUMBER OF PIZZA ORDERED AS PER THEIR SIZE



A-Z Size of Pizza	123 Total Pizza Ordered
М	15,385
L	18,526
S	14,137
XL	544
XXL	28

### LIST THE TOP 5 MOST ORDERED PIZZA TYPES WITH QUANTITY

```
⊖ select
     p3.name,
     sum(p1.quantity) as 'Total quantity ordered'
 from
     order_details p1
 join pizzas p2
 on
     p1.pizza_id = p2.pizza_id
 join pizza_types p3
 on
     p3.pizza_type_id = p2.pizza_type_id
 group by
     p3.name
 order by
     sum(p1.quantity) desc
 limit 5;
```

A-ℤ name	123 Total quantity ordered
The Classic Deluxe Pizza	2,453
The Barbecue Chicken Pizza	2,432
The Hawaiian Pizza	2,422
The Pepperoni Pizza	2,418
The Thai Chicken Pizza	2,371

## CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```
    select

     p1.category ,
     round((sum(p3.quantity * p2.price)) /
     (select sum(order_details.quantity * pizzas.price)
     from order_details join pizzas
 on
     order_details.pizza_id = pizzas.pizza_id), 4)* 100
 as revenue
 from
     pizza_types p1
 join pizzas p2
 on
     p1.pizza_type_id = p2.pizza_type_id
 join order details p3
     p3.pizza_id = p2.pizza_id
 group by
     p1.category;
```

A-Z category	123 revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

#### DETERMINE THE TOP 3 MOST ORDERED CATEGORY WISE PIZZA TYPES (REVENUE)

```
select name, category, rev
 from
 (select
 name, category, rev,
 rank() over(partition by category order by rev desc) as alpha
 from
 (select
     p3.name,
     p3.category,
     sum(p1.quantity * p4.price) as rev
 from
     pizza_types p3
 join pizzas p4 on
     p3.pizza_type_id = p4.pizza_type_id
 join order_details p1 on
     p1.pizza id = p4.pizza id
 group by
     p3.name,
     p3.category) as abcd) as bcd
 where bcd.alpha in (1,2,3);
```

<sup>A-ℤ</sup> name	A-Z category ▼	<sup>123</sup> rev ▼
The Thai Chicken Pizza	Chicken	43,434.25
The Barbecue Chicken Pizza	Chicken	42,768
The California Chicken Pizza	Chicken	41,409.5
The Classic Deluxe Pizza	Classic	38,180.5
The Hawaiian Pizza	Classic	32,273.25
The Pepperoni Pizza	Classic	30,161.75
The Spicy Italian Pizza	Supreme	34,831.25
The Italian Supreme Pizza	Supreme	33,476.75
The Sicilian Pizza	Supreme	30,940.5

#### ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
Select date1 , round(sum(rev) over(order by date1),2) as 'cum rev'
from
  (select orders.date1 ,
        sum(order_details.quantity * pizzas.price) as 'rev'
from
        order_details
join pizzas
on
        order_details.pizza_id = pizzas.pizza_id
        join orders on orders.order_id = order_details.order_id

201
```

group by orders.date1) as sales ;

<sup>A-Z</sup> date1 ▼	123 cum rev
2015-01-01	2,713.85
2015-01-02	5,445.75
2015-01-03	8,108.15
2015-01-04	9,863.6
2015-01-05	11,929.55
2015-01-06	14,358.5
2015-01-07	16,560.7
2015-01-08	19,399.05
2015-01-09	21,526.4
2015-01-10	23,990.35

#### DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```
⊖ select
     o3.name ,
     round(sum(o1.quantity * o2.price), 2) as 'Total Sales'
 from
     order_details o1
 join pizzas o2
 on
     o1.pizza_id = o2.pizza_id
 join pizza_types o3 on
     o2.pizza_type_id = o3.pizza_type_id
 group by
     o3.name
 order by
     round(sum(o1.quantity * o2.price), 2) desc
 limit 3;
```

A-Z name ▼	123 Total Sales
The Thai Chicken Pizza	43,434.25
The Barbecue Chicken Pizza	42,768
The California Chicken Pizza	41,409.5



# GITHUB FULL PROJECT LINK





