

PIZZA SALES



ABOUT US



I hold a BBA degree and bring a strong foundation in business strategy and decision-making. Alongside my academic background, I have built technical expertise in Excel, SQL, and Power BI, with certifications in these tools. I applied my SQL skills to a pizza sales analysis project, where I explored sales data to identify trends and key performance drivers. My goal is to leverage both my business acumen and analytical skills to grow as a Data Analyst, transforming raw data into actionable insights that support smarter business decisions.

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

```
• with cte as (SELECT  
    (o.order_date) AS day, SUM(od.quantity) AS quantity  
  FROM orders o JOIN  
    order_details od USING (order_id)  
 GROUP BY day)  
  SELECT ROUND(AVG(quantity)) AS avg_pizza_ordered  
  FROM cte;
```

Result Grid | Filter Rows

	avg_pizza_ordered
▶	138

join the relevant table to find the category wise distribution of pizzas

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

Result Grid | Filter Rows:

	category	pizza_count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

analyze the cumulative revenue generated over time

- **with**

```
cte as (select  
o.order_date,  
round(sum(od.quantity*p.price))as revenue  
from  
order_details od , pizza_types pt, pizzas p ,orders o  
where  
od.pizza_id = p.pizza_id and  
od.order_id = o.order_id  
group by o.order_date  
)  
select order_date,  
sum(revenue) over (order by order_date) as cum_revenue  
from cte
```

	order_date	cum_revenue
▶	2015-01-01	86843
	2015-01-02	174264
	2015-01-03	259461
	2015-01-04	315635
	2015-01-05	381745
	2015-01-06	459471
	2015-01-07	529941
	2015-01-08	620768
	2015-01-09	688843

Identify the highest-priced pizza

- **SELECT**

```
pt.name, p.price  
FROM  
pizza_types pt  
JOIN  
pizzas p USING (pizza_type_id)  
ORDER BY price DESC  
LIMIT 1
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

Determine the distribution of order by hour of the day

- SELECT
 HOUR(order_time) AS Hour,
 COUNT(order_id) AS order_count
FROM
 orders
GROUP BY Hour

	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

calculate the percentage contribution of each pizza type to total revenue

```
• with cte as
  (select round(sum(od.quantity*p.price),2)as total_sales
   from order_details od
   join
   pizzas p using(pizza_id)
  )
  SELECT
    pt.category,
    ROUND(SUM((od.quantity*p.price) /cte.total_sales) *100,2) AS revenue
  FROM
    cte,
    pizza_types pt,
    order_details od,
    pizzas p
  WHERE
    pt.pizza_type_id = p.pizza_type_id
    AND
    od.pizza_id = p.pizza_id
  GROUP BY pt.category
  ORDER BY revenue DESC
```

Result Grid | Filter Rows:

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

join the necessary table to find the total quantity of each pizza category

- **SELECT**

```
    pt.category, SUM(od.quantity) AS quantity
FROM
    pizza_types pt
        JOIN
    pizzas p USING (pizza_type_id)
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY category
ORDER BY quantity DESC
```

Result Grid | Filter

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

indentify the most comman pizza type ordered

- **SELECT**

```
    quantity, count(order_details_id) as quantity_count  
FROM  
    order_details  
GROUP BY quantity
```

Result Grid | Filter Rows:

	quantity	quantity_count
▶	1	47693
	2	903
	3	21
	4	3

identify the most common pizza size ordered

- `SELECT p.size, COUNT(od.order_details_id) AS sizes_ordered
FROM pizzas p
JOIN order_details od USING (pizza_id)
GROUP BY size
ORDER BY sizes_ordered DESC`

Result Grid | Filter Row

	size	sizes_ordered
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

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

```
• with cte as
  (SELECT
    pizza_types.category,
    pizza_types.name as pizza_name ,
    round(SUM(order_details.quantity * pizzas.price)) AS revenue
  FROM
    pizza_types,
    order_details,
    pizzas
  WHERE
    pizza_types.pizza_type_id = pizzas.pizza_type_id
      AND order_details.pizza_id = pizzas.pizza_id
  GROUP BY pizza_types.category , pizza_name
),
x as(select* ,
       rank() over(partition by category order by revenue desc) as rn
     from cte)

select* from x
where rn <= 3
```

	category	pizza_name	revenue	rn
▶	Chicken	The Thai Chicken Pizza	43434	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41410	3
	Classic	The Classic Deluxe Pizza	38180	1
	Classic	The Hawaiian Pizza	32273	2
	Classic	The Pepperoni Pizza	30162	3
	Supreme	The Spicy Italian Pizza	34831	1
	Supreme	The Italian Supreme Pizza	33477	2
	Supreme	The Sicilian Pizza	30940	3
	Veggie	The Four Cheese Pizza	32266	1
	Veggie	The Mexicana Pizza	26781	2
	Veggie	The Five Cheese Pizza	26066	3

determine the top 3 most ordered pizza types based on revenue

- `SELECT pt.name, SUM(od.quantity * p.price) AS revenue
FROM pizzas p
JOIN order_details od USING (pizza_id)
JOIN pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY 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

list the top 5 most ordered pizza Types along with their quantities

• SELECT

```
pt.name, SUM(od.quantity) AS quantity
FROM
    pizzas p
        JOIN
    order_details od USING (pizza_id)
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.name
ORDER BY quantity DESC
LIMIT 5
```

Result Grid | Filter Rows:

	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

Retrive the total number of orders placed

- `SELECT`
- `COUNT(order_id) total_order`
- `FROM`
- `orders`

Result Grid	
	<code>total_order</code>
▶	21350

calculate total revenue generated from pizza sales

- **SELECT**

```
    ROUND(SUM((o.quantity * p.price)), 2) AS total_revenue  
FROM  
    order_details o  
    JOIN  
    pizzas p  
    USING (pizza_id)
```

Result Grid |

	total_revenue
▶	817860.05



Home

**THANK YOU
FOR ATTENTION.**