

# SQL PROJECT ON PIZZA SALES

Presented by – Smriti Gupta

 smritig212@gmail.com

Home

About

Contact



Tools used –





# Hello!



## PIZZA SALES ANALYSIS USING SQL

My name is Smriti Gupta , and in this project , I used SQL queries to analyze pizza sales data and answer various related questions . By applying techniques such as filtering , joining , and aggregation , I was able to gain insights into factors influencing sales , customer preferences , and order trends .








# DATA SCHEMA OVERVIEW

[About](#)[Contact](#)

01

PIZZAS TABLE

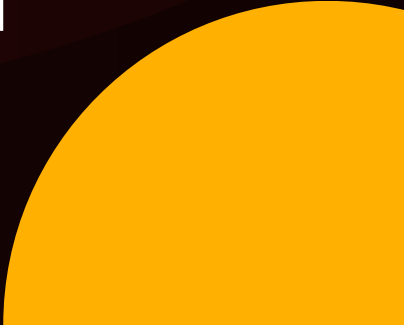


Column Name	Data type
pizza_id	NVARCHAR(50)
pizza_type_id	NVARCHAR(50)
size	NVARCHAR(50)
price	FLOAT

02

PIZZA\_TYPE TABLE

Column Name	Data type
pizza_type_id	NVARCHAR(50)
name	NVARCHAR(50)
category	NVARCHAR(50)
ingredients	NVARCHAR(100)



**03**

## ORDERS TABLE

Column Name	Data type
order_id	INT
order_date	DATE
order_time	TIME(7)

**04**

## ORDER\_DETAILS TABLE

Column Name	Data type
order_details_id	INT
order_id	INT
pizza_id	TEXT
quantity	INT



# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
select count(order_id) as Total_orders from orders;
```



Results		Messages
	Total_orders	
1	21350	



# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

Contact

```
select  
round(sum(order_details.quantity * pizzas.price),2) as Total_sales  
from order_details join pizzas  
on pizzas.pizza_id = CAST(order_details.pizza_id as nvarchar)
```



Results		Messages
	Total_sales	
1	817860.05	

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

Contact

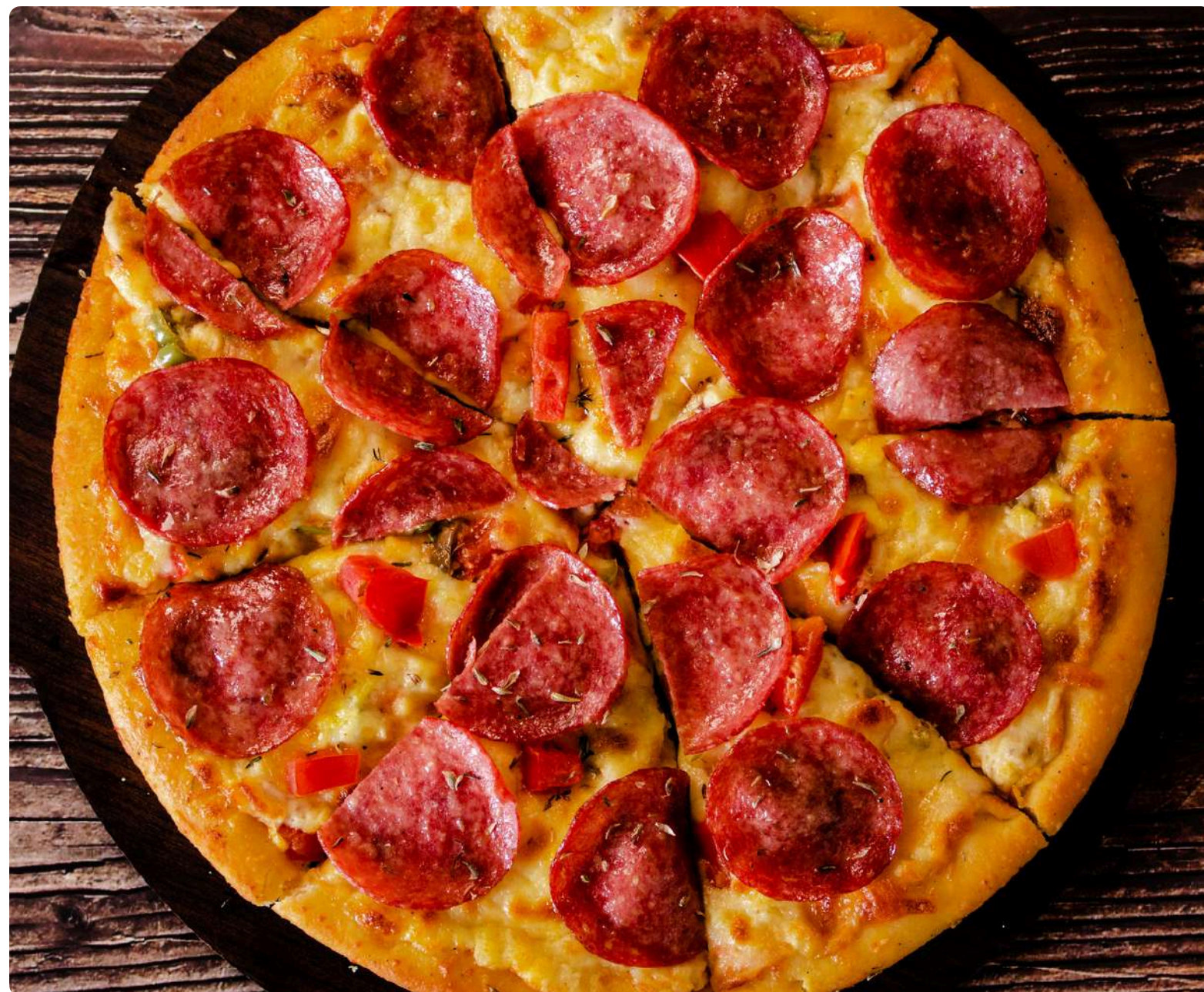
```
select pizzas.size, COUNT(order_details.order_details_id) as order_count
from pizzas join order_details
on pizzas.pizza_id = cast(order_details.pizza_id as nvarchar)
group by pizzas.size order by order_count desc ;
```

Results		Messages
	size	order_count
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28





# IDENTIFY THE HIGHEST-PRICED PIZZA.



```
select top 1 pizza_types.name,  
round(pizzas.price,2)  
from pizza_types  
join pizzas on  
pizza_types.pizza_type_id =  
pizzas.pizza_type_id  
order by pizzas.price DESC ;
```

Results			Messages		
	name	(No column name)			
1	The Greek Pizza	35.95			



# LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
select top 5 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 cast(order_details.pizza_id as nvarchar) = pizzas.pizza_id  
group by pizza_types.name order by quantity desc ;
```



Results			Messages		
	name	quantity			
1	The Classic Deluxe Pizza	2453			
2	The Barbecue Chicken Pizza	2432			
3	The Hawaiian Pizza	2422			
4	The Pepperoni Pizza	2418			
5	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 quantity  
from pizza_types join pizzas  
on pizza_types.pizza_type_id =pizzas.pizza_type_id  
join order_details  
on cast(order_details.pizza_id as nvarchar) =pizzas.pizza_id  
group by pizza_types.category order by quantity desc;
```



Results			Messages
	category	quantity	
1	Classic	14888	
2	Supreme	11987	
3	Veggie	11649	
4	Chicken	11050	



# DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

Contact

```
select DATEPART(hour,order_time)as hour ,count (order_id) as order_count from orders
group by datepart(hour,order_time);
```

Results		Messages
	hour	order_count
1	23	28
2	15	1468
3	9	1
4	12	2520
5	21	1198
6	18	2399
7	10	8
8	19	2009
9	13	2455
10	22	663
11	16	1920

11	16	1920
12	17	2336
13	11	1231
14	20	1642
15	14	1472



# JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

[Home](#)[About](#)[Contact](#)

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



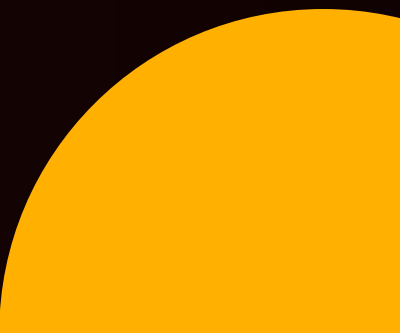
Results		Messages
	category	(No column name)
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9



# GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
select round(avg(quantity),0) as avg_pizza_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;
```

Results		Messages	
	avg_pizza_ordered_per_day		
1	138		

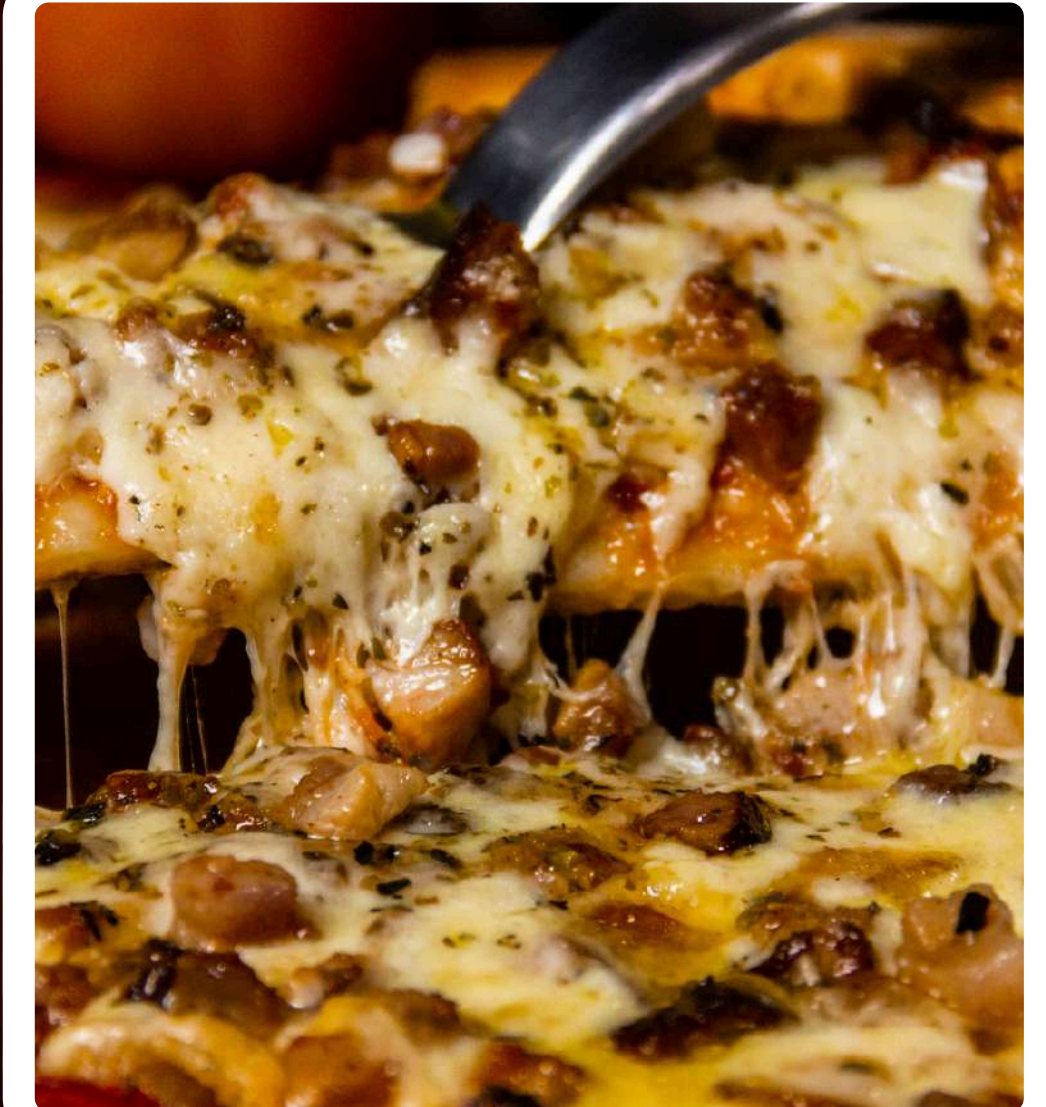




# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
select top 3 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 cast(order_details.pizza_id as nvarchar) = pizzas.pizza_id  
group by pizza_types.name order by revenue desc ;
```

Results			Messages		
	name	revenue			
1	The Thai Chicken Pizza	43434.25			
2	The Barbecue Chicken Pizza	42768			
3	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_sales  
from order_details  
join  
pizzas on pizzas.pizza_id = cast(order_details.pizza_id as nvarchar)) *100,2) as revenue  
  
from pizza_types join pizzas  
on pizzas.pizza_type_id=pizza_types.pizza_type_id  
join order_details  
on cast(order_details.pizza_id as nvarchar) = pizzas.pizza_id  
group by pizza_types.category order by revenue desc ;
```



Results		Messages
	category	revenue
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68





# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date ,  
       sum(revenue) over (order by order_date) as cum_revenue  
from  
  (select orders.order_date,  
   SUM(order_details.quantity * pizzas.price) as revenue  
  from order_details join pizzas  
  on cast(order_details.pizza_id as nvarchar) =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 cast(order_details.pizza_id as nvarchar) = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn <=3;
```

Results			Messages		
	name	revenue			
1	The Thai Chicken Pizza	43434.25			
2	The Barbecue Chicken Pizza	42768			
3	The California Chicken Pizza	41409.5			
4	The Classic Deluxe Pizza	38180.5			
5	The Hawaiian Pizza	32273.25			
6	The Pepperoni Pizza	30161.75			
7	The Spicy Italian Pizza	34831.25			
8	The Italian Supreme Pizza	33476.75			
9	The Sicilian Pizza	30940.5			
10	The Four Cheese Pizza	32265.7010040283			
11	The Mexicana Pizza	26780.75			
12	The Five Cheese Pizza	26066.5			





[Home](#)

[About](#)

[Contact](#)

# THANK YOU

## FOR ATTENTION