

Pizza Sales **Analysis Using SQL** Developed a comprehensive **SQL-based**



MY NAME IS SHUBHAM KUMAR,
AND IN THIS PROJECT, I USED
SQL TO ANALYZE PIZZA SALES
DATA. THE GOAL IS TO SOLVE
VARIOUS BUSINESS-RELATED
QUESTIONS USING SQL QUERIES,
FOCUSING ON ASPECTS LIKE
REVENUE, CUSTOMER

REVENUE, CUSTOMER
PREFERENCES, AND PIZZA
POPULARITY.T

PIZZA
CORNER

SINCE 1999

Retrieve the total number of orders placed



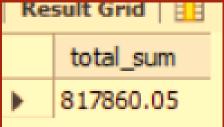
select count(order_id) as total_orders
from orders;



Calculate the total revenue generated from pizza sales.



SINCE 1999



dentify the lowest-priced pizza



```
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
```

Result Grid 11		
	name	price
•	The Greek Pizza	35.95

Identify the most common pizza size ordered



```
COUNT(orders_details.order_details_id), pizzas.size

FROM

orders_details

JOIN

pizzas ON pizzas.pizza_id = orders_details.pizza_id

GROUP BY pizzas.size;
```

Re	Result Grid			
	COUNT(orders_details.order_details_id)	size		
	15385	М		
	18526	L		
	14137	S		
	544	XL		
	28	XXL		

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



```
SELECT

pizza_types.category,

SUM(orders_details.quantity) AS quantity

FROM

pizza_types

JOIN

pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id

JOIN

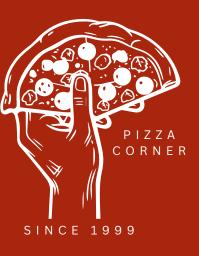
orders_details ON orders_details.pizza_id = pizzas.pizza_id

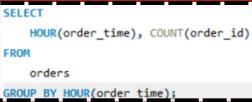
GROUP BY category
```

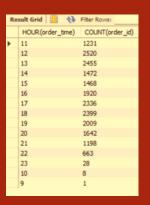
Result Grid		
	category	quantity
•	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the

day







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

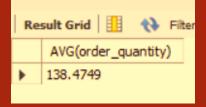


select category , count(name)
from pizza_types
group by category;

Result Grid			
	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.





Determine the top 3 most ordered pizza types based on revenue



```
pizza_types.name,
SUM(pizzas.price * orders_details.quantity) AS revenue
ROM

pizzas

JOIN

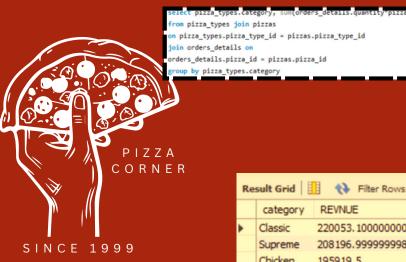
pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id

JOIN

orders_details ON orders_details.pizza_id = pizzas.pizza_id
ROUP BY pizza_types.name
RDER BY revenue DESC
```

Re	sult Grid 🔠 🙌 Filter Row	S:
	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.





Analyze the cumulative revenue generated over time

```
SELECT

orders.order_date,

SUM(pizzas.price * orders_details.quantity) AS daily_revenue,

SUM(SUM(pizzas.price * orders_details.quantity)) OVER (ORDER BY orders.order_date) AS cumulative_revenue

FROM

orders

JOIN

orders_details ON orders.order_id * orders_details.order_id

301N

pizzas ON orders_details.pizza_id * pizzas.pizza_id

GROUP BY orders.order_date

ORDER BY orders.order_date;

PIZZA

CORNER

Result Grid 

Filter Roves:

Export:

Export:

Wrap Ce
```

SINCE 1999

-				_
Res	sult Grid	Filter Rows:	Export: Wrap	c
	order_date	daily_revenue	cumulative_revenue	
•	2015-01-01	2713.8500000000004	2713.8500000000004	
	2015-01-02	2731,8999999999996	5445.75	
	2015-01-03	2662 2731.89999999999	996 B.15	
	2015-01-04	1755.4500000000003	9863.6	
	2015-01-05	2065.95	11929.55	
	2015-01-06	2428.95	14358.5	
	2015-01-07	2202.2000000000003	16560.7	
	2015-01-08	2838.3499999999995	19399.05	
	2015-01-09	2127.3500000000004	21526.4	
	2015-01-10	2463.95	23990.350000000002	
	2015-01-11	1872.3000000000000	25862.65	
	2015-01-12	1919.05000000000002	27781.7	
	2015-01-13	2049.60000000000004	29831.300000000003	
	2015-01-14	2527.399999999996	32358.700000000004	
2	2015 01 15	100/4 000000000000000	24242 50000000001	

THANK'S FOR WATCHING

This project by Shubham Kumar uses SQL queries to analyze pizza sales data. It focuses on revenue trends, top-selling pizzas, and customer behavior to provide insights for optimizing sales, marketing, and inventory strategies.

