



# Hello!



My name is Anshu Kumar and I am building a new SQL Pizza Sales Project is a comprehensive data analysis endeavor that aims to provide insightful metrics and trends within a pizza business using SQL. This project involves creating and managing a database that captures various aspects of pizza sales, including orders, customer details, delivery times, and inventory levels.



Retrieve the total number of  
orders placed.



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

	total_orders
▶	21350



# Calculate the total revenue generated from pizza sales.

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

	total_sales
▶	817860.05

# Identify the 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
```

Result Grid			Filter Rows
	name	price	
▶	The Greek Pizza	35.95	



# Identify the most common pizza size ordered.

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

Result Grid		
	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

# List the top 5 most ordered pizza types along with their quantities.

```
    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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
```

	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

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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC
```

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

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

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

Result Grid			Filter Rows
	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.

```
select round(avg(quantity),0) as avg_pizzas_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
```

Result Grid		Filter Rows:
	avg_pizzas_ordered_per_day	
▶	138	

# Analyze the cumulative revenue generated over time.

```
select order_date,  
sum(revenue) over(order by order_date) as cumulative_revenue  
from  
(select orders.order_date,  
sum(order_details.quantity * pizzas.price) as revenue  
from order_details  
join pizzas  
on order_details.pizza_id = pizzas.pizza_id  
join orders  
on orders.order_id = order_details.order_id  
group by orders.order_date) as sales
```

Result Grid			Filter Rows:
	order_date	cumulative_revenue	
▶	2015-01-01	2713.8500000000004	
	2015-01-02	5445.75	
	2015-01-03	8108.15	
	2015-01-04	9863.6	
	2015-01-05	11929.55	
	2015-01-06	14358.5	
	2015-01-07	16560.7	
	2015-01-08	19399.05	
	2015-01-09	21526.4	
	2015-01-10	23990.350000000002	
	2015-01-11	25862.65	
	2015-01-12	27781.7	
	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	
	2015-01-20	45763.650000000001	
	2015-01-21	47804.200000000001	
	2015-01-22	50300.900000000001	
	2015-01-23	52784.600000000006	

# 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 order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn <=3;
```

Result Grid			Filter Rows:
	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	
	The Spicy Italian Pizza	34831.25	
	The Italian Supreme Pizza	33476.75	
	The Sicilian Pizza	30940.5	
	The Four Cheese Pizza	32265.700000000065	
	The Mexicana Pizza	26780.75	
	The Five Cheese Pizza	26066.5	

**Thank You!**

