

# SQL PIZZA SALES PROJECT SHOWCASE



# WELCOME TO PIZZA SALES PROJECT SHOWCASE

This project explores a comprehensive SQL-based analysis of a fictional pizza sales dataset, answering questions ranging from basic queries to advanced analytics. The goal is to demonstrate practical SQL applications and problem-solving skills through real-world scenarios.

## Key Highlights:

- **Basic Queries:** Simple data retrieval, filtering, and sorting.
- **Intermediate Concepts:** Aggregations, joins, and subqueries for deeper insights.
- **Advanced Analysis:** CTEs, window functions, and performance optimization techniques.
- **Practical Solutions:** Clear problem statements with SQL scripts and output for each question.



# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
select count(order_id) as Total_order from pizzahut.orders;
```

Result Grid	
	Total_order
▶	21350

# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

**SELECT**

```
ROUND(SUM(order_details.quantity * pizzas.price),  
2) AS total_revenue
```

**FROM**

```
order_details
```

**JOIN**

```
pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

	total_revenue
▶	817860.05

# IDENTIFY THE HIGHEST-PRICED PIZZA

**SELECT**

**pizza\_types.name, pizzas.price AS max\_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:

	<b>name</b>	<b>max_price</b>
▶	The Greek Pizza	35.95

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT  
    pizzas.size,  
    COUNT(order_details.order_details_id) AS most_common_pizza_size  
FROM  
    pizzas  
    JOIN  
        order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizzas.size  
ORDER BY most_common_pizza_size DESC;
```

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

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

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS most_ordered_pizza
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY most_ordered_pizza DESC
LIMIT 5;
```

name	most_ordered_pizza
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 total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY total_quantity DESC;
```

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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

**SELECT**

**HOUR(time) AS Hour\_timne, COUNT(order\_id) AS order\_count**

**FROM**

**orders**

**GROUP BY HOUR(time);**

Hour_timne	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

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

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

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) from
(select orders.date, sum(order_details.quantity) as quantity
from orders join order_details
on orders.order_id = order_details.order_id
group by orders.date) as ordered_per_day;
```

round(avg(quantity), 0)

138

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

```
select pizza_types.name,  
round(sum(pizzas.price * order_details.quantity), 0) as revenue  
from pizza_types join pizzas  
ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
join order_details  
on pizzas.pizza_id = order_details.pizza_id  
group by pizza_types.name  
order by revenue desc  
limit 3;
```

name	revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

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

```
select pizza_types.category,  
round(sum(pizzas.price * order_details.quantity) / (SELECT  
    (ROUND(SUM(order_details.quantity * pizzas.price),  
        2)) AS total_revenue  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id)* 100,2) as revenue  
  
from pizza_types join pizzas  
ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
join order_details  
on pizzas.pizza_id = order_details.pizza_id  
group by pizza_types.category  
order by revenue desc  
limit 3;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96

# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select date,  
       round(sum(revenue) over(order by date),0) as cumulative  
  from  
(select orders.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 order_details.order_id = orders.order_id  
   group by orders.date) as sales;
```

date	cumulative
2015-01-01	2714
2015-01-02	5446
2015-01-03	8108
2015-01-04	9864
2015-01-05	11930
2015-01-06	14358
2015-01-07	16561
2015-01-08	19399
2015-01-09	21526
2015-01-10	23990
2015-01-11	25863
2015-01-12	27782
2015-01-13	29831
2015-01-14	32359
2015-01-15	34344
2015-01-16	36938
2015-01-17	39002
2015-01-18	40979
2015-01-19	43366

# 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) as rn
from
(select pizza_types.category, pizza_types.name,
round(sum(order_details.quantity * pizzas.price), 0) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category, pizza_types.name )as a)as b
where rn <= 3;
```

name	revenue
The Chicken Pesto Pizza	16702
The Chicken Alfredo Pizza	16900
The Southwest Chicken Pizza	34706
The Pepperoni, Mushroom, and Peppers Pizza	18834
The Big Meat Pizza	22968
The Napolitana Pizza	24087
The Brie Carre Pizza	11588
The Spinach Supreme Pizza	15278
The Calabrese Pizza	15934
The Green Garden Pizza	13956
The Mediterranean Pizza	15360
The Spinach Pesto Pizza	15596

# THANK YOU!

