

# PIZZA DATASET

DATA  
PIZZA



Hello my name is Ramya,  
In this project, I utilized SQL  
queries to analyze and solve  
questions related to pizza  
sales. The project involved  
leveraging structured query  
language to extract  
meaningful insights from  
sales data, enabling data-  
driven decision-making.



# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

```
SELECT  
    SUM(order_details.quantity * pizzas.price) AS total_revenue  
FROM  
    pizzas  
    JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid	
	total_revenue
▶	817860.0499999993

# IDENTIFY THE HIGHEST-PRICED PIZZA

```
1  SELECT
2      pizza_types.name, pizzas.price
3  FROM
4      pizzas
5      JOIN
6          pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
7  ORDER BY pizzas.price DESC
8  LIMIT 1;
```

	name	price
▶	The Greek Pizza	35.95

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
SELECT  
    pizzas.size, COUNT(order_details.order_details_id)  
FROM  
    pizzas  
    JOIN  
        order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizzas.size  
ORDER BY COUNT(order_details.order_details_id) DESC  
LIMIT 1;
```

	size	COUNT(order_details.order_details_id)
▶	L	18526

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

```
SELECT  
    pizza_types.name, SUM(order_details.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.name  
ORDER BY SUM(order_details.quantity) DESC  
LIMIT 5;
```

	name	SUM(order_details.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 total_quantity
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category
ORDER BY SUM(order_details.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(order_time) AS hour, COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

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

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

```
SELECT  
    AVG(quantity)  
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;
```

	AVG(quantity)
▶	138.4749

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

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity * pizzas.price)/(select round(SUM(order_details.quantity * pizzas.price),2) as total_sales
FROM
    order_details
    JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id)*100 as revenue
FROM
    order_details
    JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
    JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category
```

category	revenue
Classic	26.90596025566967
Veggie	23.682590927384577
Supreme	25.45631126009862
Chicken	23.955137556847287

# 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(pizzas.price*order_details.quantity) as revenue  
         from orders join order_details  
          on orders.order_id = order_details.order_id  
         join pizzas on pizzas.pizza_id = order_details.pizza_id  
        group by orders.order_date) as sales;
```

order_date	cum_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	23990.35000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.30000000003
2015-01-14	32358.70000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.60000000006
2015-01-19	43365.75000000001

# 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;
```

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	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

# THANK YOU!

