



SQL PROJECT ON PIZZA SALES

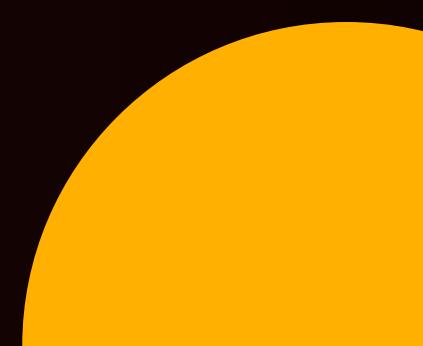


- WHERE EVERY SLICE TELLS A STORY



HELLO!

My name is Karumbaiah C B This project focuses on Pizza Sales Data Analysis using SQL. It explores key business insights such as daily revenue trends, top-selling pizzas, and category-wise sales contributions. The project highlights my ability to write optimized SQL queries for real-world data analysis."



SQL Business Questions Solved in Pizza Sales Data Analysis

1. Total number of orders placed
2. Total revenue generated from pizza sales
3. Highest-priced pizza
4. Most common pizza size ordered
5. Top 5 most ordered pizza types by quantity
6. Total quantity of pizzas ordered by category
7. Distribution of orders by hour of the day
8. Category-wise distribution of pizzas
9. Average pizzas ordered per day
10. Top 3 pizza types based on revenue
11. Percentage contribution of each pizza type to revenue
12. Cumulative revenue generated over time
13. Top 3 pizza types by revenue within each category

Retrieve the total number of orders placed

```
SELECT  
    COUNT(order_id) AS total_orders_Placed  
FROM  
    orders;
```

Result Grid	
	total_orders_Placed
▶	21350

Calculate the total revenue generated from pizza sales

```
SELECT  
    ROUND(SUM(orders_details.oquantity * pizzas.price),  
        2) AS total_sales  
FROM  
    orders_details  
    JOIN  
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

Result Grid	
	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(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

	size	order_count
▶	L	18526

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

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

	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.name, 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 pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

	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

Determine the distribution of orders by hour of the day

```
SELECT  
    HOUR(order_time), COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

	HOUR(order_time)	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) as avg_pizza_ordered_per_day  
FROM  
    (SELECT  
        orders.order_date, SUM(orders_details.quantity) AS quantity  
    FROM  
        orders  
    JOIN orders_details ON orders.order_id = orders_details.order_id  
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid	
	avg_pizza_ordered_per_day
▶	138

Determine the top 3 most ordered pizza types based on revenue

```
SELECT  
    pizza_types.name,  
    SUM(orders_details.oquantity * pizzas.price) AS revenue  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
        JOIN  
    orders_details ON orders_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.name  
ORDER BY revenue DESC  
LIMIT 3;
```

	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

```
SELECT
    pizza_types.category,
    ROUND((SUM(orders_details.oquantity * pizzas.price) / (SELECT
        SUM(orders_details.oquantity * pizzas.price)
    FROM
        orders_details
    JOIN
        pizzas ON pizzas.pizza_id = orders_details.pizza_id)) * 100,
    2) AS revenue
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 pizza_types.category
ORDER BY revenue DESC;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	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(orders_details.oquantity * pizzas.price) as revenue  
from orders_details join pizzas  
on orders_details.pizza_id = pizzas.pizza_id  
join orders  
on orders.order_id = orders_details.order_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
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.60000000006
2015-01-24	55013.85000000006
2015-01-25	56631.40000000001
2015-01-26	58515.80000000001
2015-01-27	61043.85000000001
2015-01-28	63059.85000000001
2015-01-29	65105.150000000016

Determine the top 3 most ordered pizza types based on revenue for each pizza category

```
(select category, name, revenue,  
rank() over(partition by category order by revenue desc) as rn  
from  
(select pizza_types.category, pizza_types.name,  
sum((orders_details.quantity) * pizzas.price) as revenue  
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 pizza_types.category, pizza_types.name) as a)as b  
where rn<=3;
```

	name	revenue	rn
▶	The Thai Chicken Pizza	43434.25	1
	The Barbecue Chicken Pizza	42768	2
	The California Chicken Pizza	41409.5	3
	The Classic Deluxe Pizza	38180.5	1
	The Hawaiian Pizza	32273.25	2
	The Pepperoni Pizza	30161.75	3
	The Spicy Italian Pizza	34831.25	1
	The Italian Supreme Pizza	33476.75	2
	The Sicilian Pizza	30940.5	3
	The Four Cheese Pizza	32265.70000000065	1
	The Mexicana Pizza	26780.75	2
	The Five Cheese Pizza	26066.5	3

THANK YOU

Home