

Uncovering insights from pizza sales data using SQL

# PIZZA SALES ANALYSIS

Start Your Slide





# ABOUT

This project is about studying pizza sales data using SQL.

It helps to find out how many pizzas were sold, how much money was made, and which pizzas people liked the most.

We also check which size of pizza is ordered the most and when people order more.

The goal is to understand sales trends and improve business decisions using data.

# QUESTIONS:



## ? Project Questions

- ◆ How many total orders were placed?
- ◆ What is the total revenue from pizza sales?
  - ◆ Which is the highest-priced pizza?
  - ◆ What is the most common pizza size ordered?
  - ◆ Which are the top 5 most ordered pizza types?
  - ◆ What time of the day do most people order pizzas?
  - ◆ Which pizza category sells the most?
  - ◆ What are the top 3 pizzas by revenue?
- ◆ How much does each pizza type contribute to total revenue
  - ◆ How does revenue grow over time?

# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

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

Result Grid	
	total_orders
▶	21350

# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT  
    ROUND(SUM(orders_details.quantity * 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;
```

Result Grid | Filter

	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.



```
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

# 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 pizza_types.category  
ORDER BY quantity DESC;
```

Result Grid | Filter

	category	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);
```

Result Grid | Filter Rows:

	hour(order_time)	count(order_id)
12	2520	
13	2455	
14	1472	
15	1468	
16	1920	

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



```
SELECT  
    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_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 | Filter Rows

	avg_pizza_ordered_per_day
▶	138

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



```
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * 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;
```

Result Grid | Filter Rows:

	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.quantity * pizzas.price) / (SELECT
        ROUND(SUM(orders_details.quantity * pizzas.price),
        2) AS total_sales
    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;
```

Result Grid | Filter

	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.quantity * pizzas.price) AS revenue  
     FROM  
        orders_details  
        JOIN  
        pizzas ON orders_details.pizza_id = pizzas.pizza_id  
        JOIN  
        orders ON orders_details.order_id = orders.order_id  
   GROUP BY orders.order_date) as sales;
```

Result Grid | Filter Rows:

	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

# 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(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;
```

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 project helped analyze pizza sales trends using SQL. It showed which pizzas sell best, when people order most, and how much revenue each type brings in — giving useful insights for better business planning.

pizza sales analysis

# THANK YOU FOR ATTENTION

See You Next

rathodarunnayak1432@gmail.com