



# PIZZAHUT INSIGHTS

## USING SQL & POWER BI



# ABOUT

I analyzed Pizza Hut's dataset using SQL and Power BI, uncovering key insights like the most sold pizzas and popular sizes. This project showcased my data analysis skills and ability to extract actionable business intelligence for strategic decision-makings.



# SCHEMA

SCHEMAS

Filter objects

- pizzahut
  - Tables
    - order\_details
    - orders
    - pizza\_types
    - pizzas



# 33

Types of Pizzas



# 5

Unique size

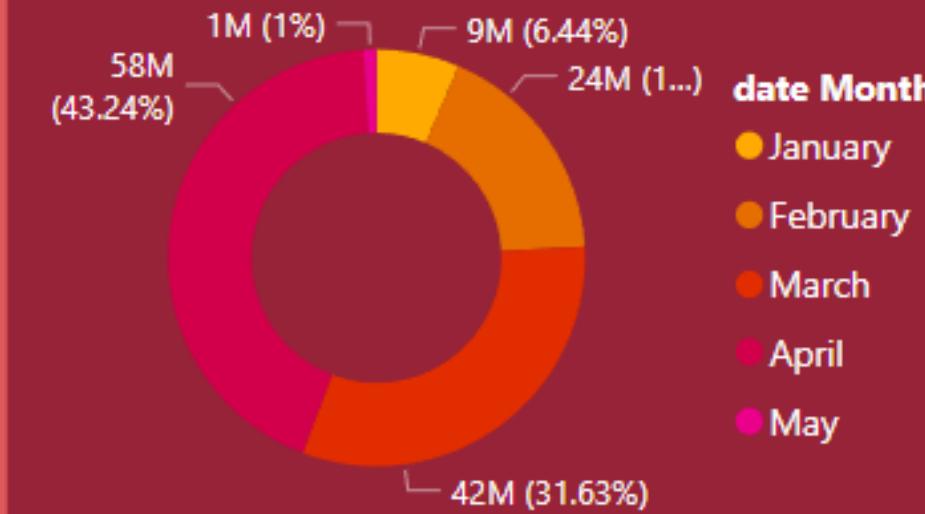
pizza by size

S

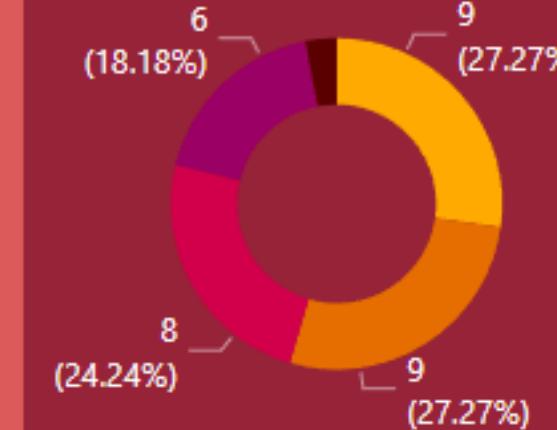
L

M

Sum of order\_details\_id by Month



Category by Pizza Category



# 96

Unique Pizza Id

# 1.02

Avg. quantity ordered

# 7211

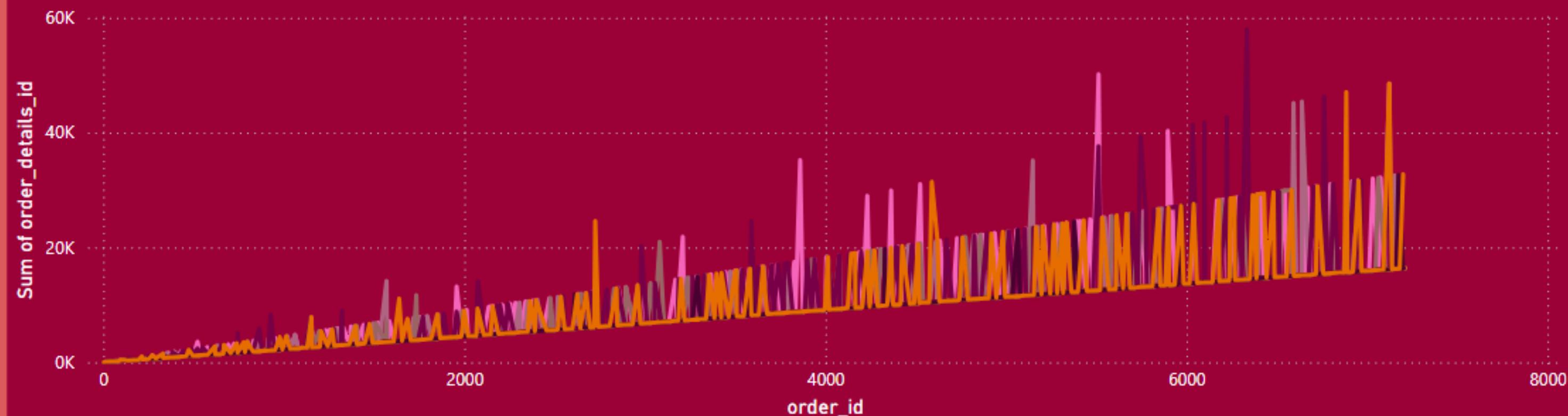
Unique Customers

Pizza Category

category
Chicken
Classic

Price Vs Order id

price 9.75 10.5 11 12 12.25 12.5 12.75 13.25 14.5 14.75 15.25 16 16.25 16.5



A close-up photograph of a person's hand holding a wooden spoon and stirring a pan of pasta. The pasta appears to be fettuccine with cherry tomatoes and possibly some meat or vegetables. The pan is dark-colored and sits on a stove. The background is blurred.

# 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 order_details.pizza_id = pizzas.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;
```

	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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

	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(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
LIMIT 5;
```

	<b>name</b>	<b>quantity</b>
▶	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;
```

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)  
FROM  
orders  
GROUP BY HOUR(order_time)  
ORDER BY COUNT(order_id) DESC  
LIMIT 10;
```

hour	COUNT(order_id)
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231



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_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;
```

Avg_pizza_per_day
138



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

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
    order_details ON order_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(order_details.quantity * pizzas.price) / (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) * 100,2) AS revenue
FROM pizza_types JOIN pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN order_details ON order_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(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;
```

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	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7

# THANK YOU

