

Pizza Sales Analyzing Using SQL

By Pramod Kumar

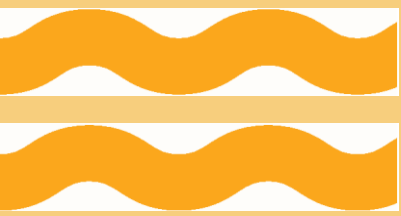




Introduction

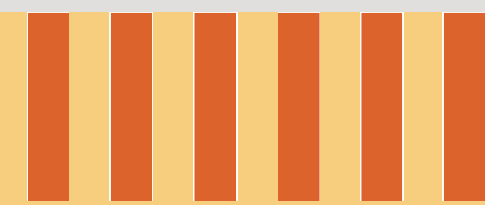
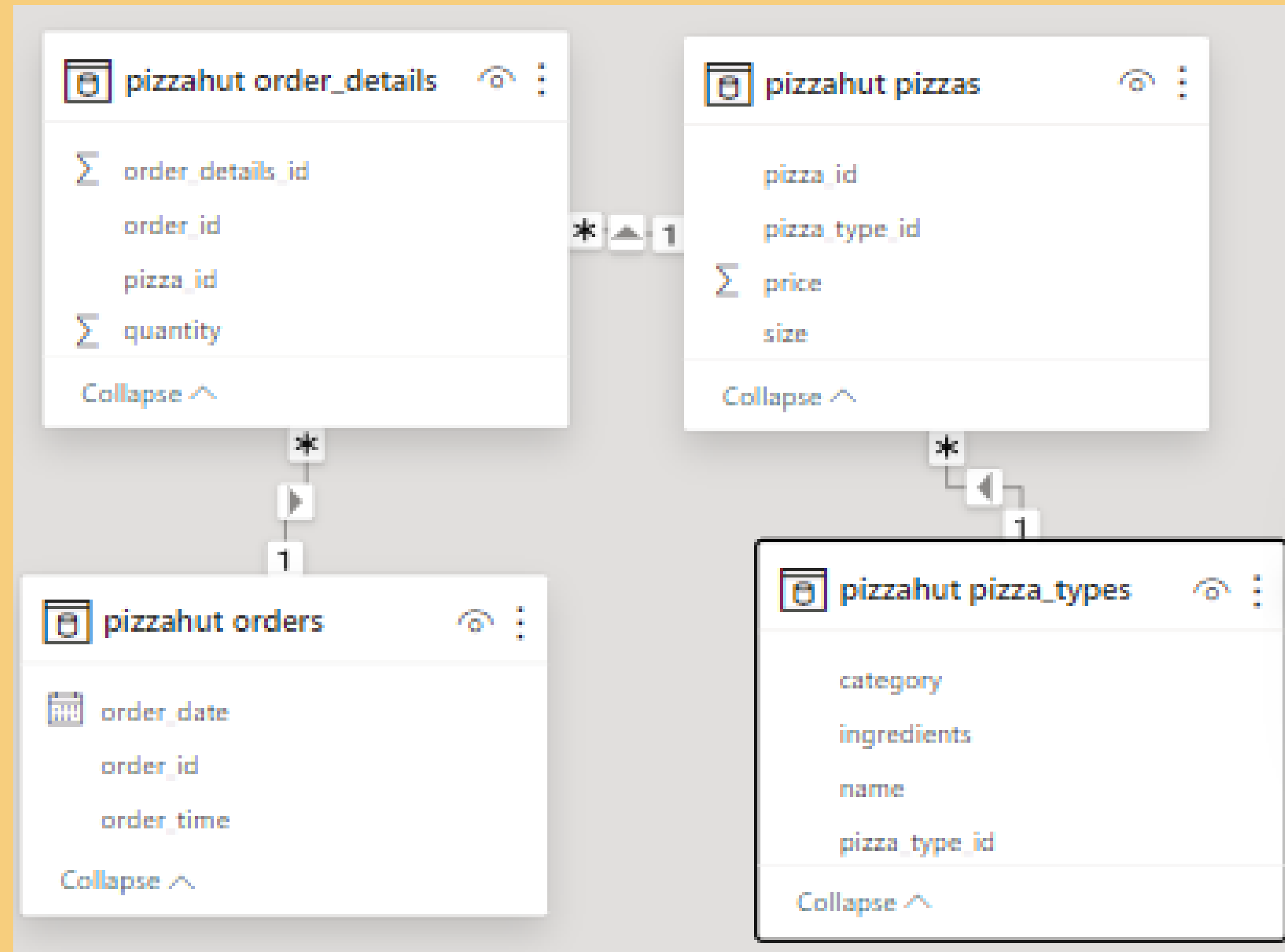
Welcome to the presentation on *Analyzing Pizza Sales Data Using SQL Queries*. This presentation will explore the **analysis** of pizza sales data through SQL queries, providing valuable insights into customer preferences and sales trends.





Understanding the Data

Before diving into the analysis, it's crucial to **understand** the structure and content of the pizza sales data. We will examine the relevant **tables** and fields to gain insights into the dataset.



* Retrieve total no of orders place



```
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(order_details.quantity * pizzas.price),  
      2) AS Total_revenue
```

FROM

```
order_details
```

JOIN

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



Result Grid



Filter

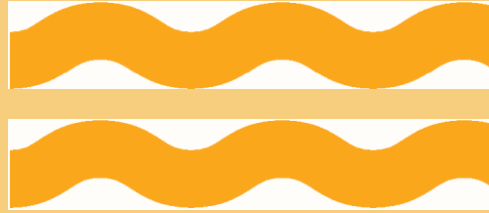
Total_revenue



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 price DESC
LIMIT 1;
```



Result Grid			Filter Rows:
	name	price	
▶	The Greek Pizza	35.95	



* Identify the most common pizza size ordered.

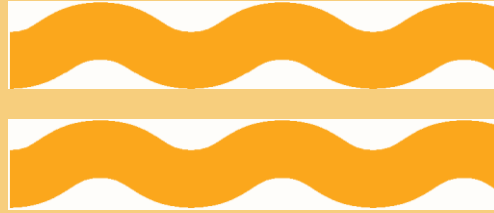


```
SELECT
    pizzas.size AS size,
    SUM(order_details.quantity) AS Total_orders
FROM
    pizzas
    JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY size
;
```

Result Grid			Filter Rows
	size	Total_orders	
▶	M	15635	
	L	18956	
	S	14403	
	XL	552	
	XXL	28	




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



```
SELECT
    pizzas.size AS size,
    SUM(order_details.quantity) AS Total_orders
FROM
    pizzas
    JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY size
;
```



Result Grid  Filter Rows: <input type="text"/>		
	Pizza_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 AS Pizza_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_category
ORDER BY quantity DESC;
```

Result Grid			Filter Rows:
	Pizza_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)
ORDER BY hour;
```

Result Grid				
	hour	order_count		
▶	9	1		
	10	8		
	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		



- * 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.



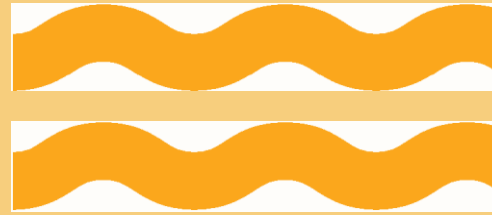
```
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizza_orderd_per_day
FROM
    (SELECT
        orders.order_date AS date,
        SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY date) AS order_quantity;
```



Result Grid		Filter Rows:
	avg_pizza_orderd_per_day	
▶	138	



- * Determine the top 3 most ordered pizza types based on revenue.



```
SELECT
    pizza_types.name AS name,
    SUM(order_details.quantity * pizzas.price) AS rev
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 name
ORDER BY rev DESC
LIMIT 3;
```

Result Grid			Filter Rows:
	name	rev	
▶	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
  Category,
  ROUND(revenue / total_revenue * 100, 2) AS Revenue_percent
FROM
  (SELECT
    pt.category AS Category,
    SUM(od.quantity * pz.price) AS revenue
  FROM
    pizza_types AS pt
  JOIN pizzas AS pz USING (pizza_type_id)
  JOIN order_details od USING (pizza_id)
  GROUP BY category
  ORDER BY revenue DESC) AS one,
  (SELECT
    SUM(od.quantity * pz.price) AS total_revenue
  FROM
    pizzas AS pz
  JOIN order_details AS od USING (pizza_id)) AS total;
```

Result Grid			Filter Rows:
	Category	Revenue_percent	
▶	Classic	26.91	
	Supreme	25.46	
	Chicken	23.96	
	Veggie	23.68	



* Analyze the cumulative revenue generated over time.



```
select date, round(sum(revenue) over(order by date),2) as cumulative_revenue from
  (select os.order_date as date, sum(od.quantity*pz.price) as revenue
    from order_details as od join orders os
      using(order_id)
    join pizzas as pz
      using(pizza_id)
   group by date) as revenue_by_day;
```

Result Grid			Filter Rows:
	date	cumulative_revenue	
▶	2015-01-01	2713.85	
	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	



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



```
select category, name, revenue, rn as Top_3 from
(select category,name, revenue, rank() over(partition by category order by revenue desc) as rn from
(select pt.category , pt.name as name, sum(od.quantity*pz.price) as revenue
from pizza_types as pt join pizzas as pz
    using(pizza_type_id)
join order_details as od
    using(pizza_id)
group by pt.category, pt.name) as a) as b
where rn <=3;
```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Center				
	category	name	round(revenue,1)	Top_3
▶	Chicken	The Thai Chicken Pizza	43434.2	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3
	Classic	The Classic Deluxe Pizza	38180.5	1
	Classic	The Hawaiian Pizza	32273.2	2
	Classic	The Pepperoni Pizza	30161.8	3
	Supreme	The Spicy Italian Pizza	34831.2	1
	Supreme	The Italian Supreme Pizza	33476.8	2
	Supreme	The Sicilian Pizza	30940.5	3
	Veggie	The Four Cheese Pizza	32265.7	1
	Veggie	The Mexicana Pizza	26780.8	2
	Veggie	The Five Cheese Pizza	26066.5	3



Sales Trends Over Time

Examining sales data over time using SQL queries allows us to identify **trends** and patterns in pizza orders. By understanding these **fluctuations**, businesses can make informed decisions to enhance sales strategies.





Conclusion

In conclusion, the analysis of pizza sales data through SQL queries offers valuable insights into customer preferences, sales trends, and business opportunities. By leveraging these insights, businesses can make informed decisions to enhance marketing strategies, optimize menu offerings, and drive profitability.



Thanks!

Do you have any questions?



Ask me in comment the
or
you can also connect with me
on
LINKEDIN

