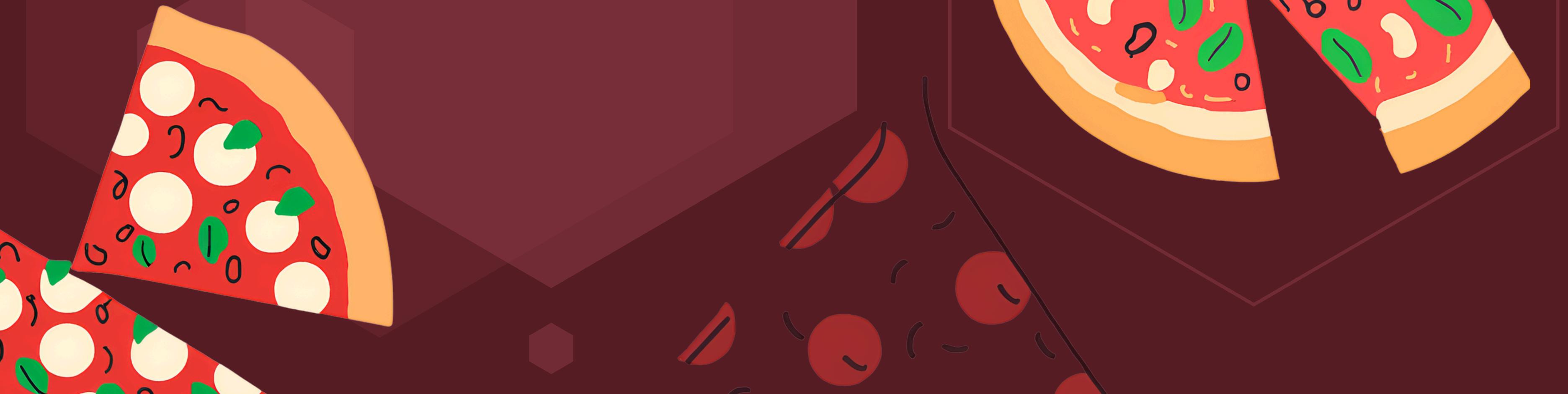


SQL PROJECT ON PIZZA HUT SALES

19 JULY, 2024



#PIZZAHUT_SALES_PROJECT USING SQL

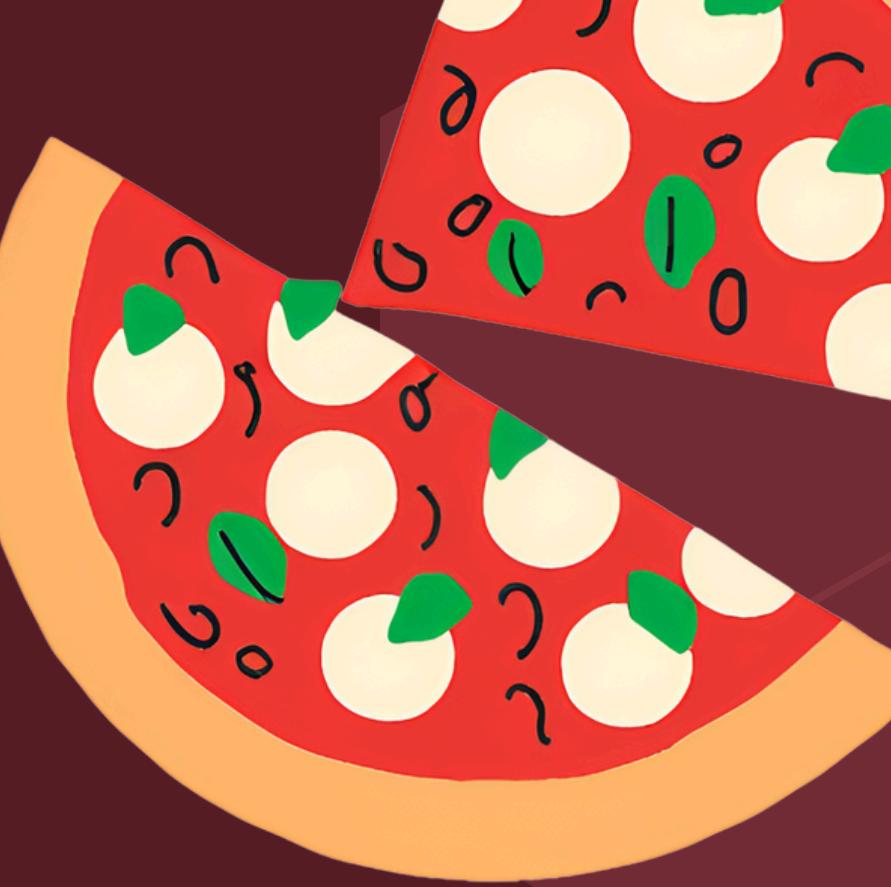


HELLO!!

My name is Nikhil Pratap Singh, Welcome to my pizza hut Sales project using sql.Todays,I have utilized sql queries to solve questions that were related to pizza sales.The objective of this project is to develop a comprehensive SQL-based system for analyzing and reporting pizza sales data. This system will provide insights into sales trends, customer preferences, and operational efficiency to support strategic decision-making for a pizza chain. It's a narrative of our collective efforts, showcasing the impact of our sales strategies.

QUESTIONS-

- 1-Retrieve the total number of orders placed.
- 2-Calculate the total revenue generated from pizza sales.
- 3-Identify the highest-priced pizza.
- 4-Identify the most common pizza size ordered.
- 5-List the top 5 most ordered pizza types along with their quantities.
- 6-Join the necessary tables to find the total quantity of each pizza category ordered.
- 7-Determine the distribution of orders by hour of the day.
- 8-Join relevant tables to find the category-wise distribution of pizzas.
- 9-Group the orders by date and calculate the average number of pizzas ordered per day.
- 10-Determine the top 3 most ordered pizza types based on revenue.
- 11-Calculate the percentage contribution of each pizza type to total revenue.
- 12-Analyze the cumulative revenue generated over time.
- 13-Determine the top 3 most ordered pizza types based on revenue for each pizza category.



Q1-Retrieve the total number of orders placed.

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

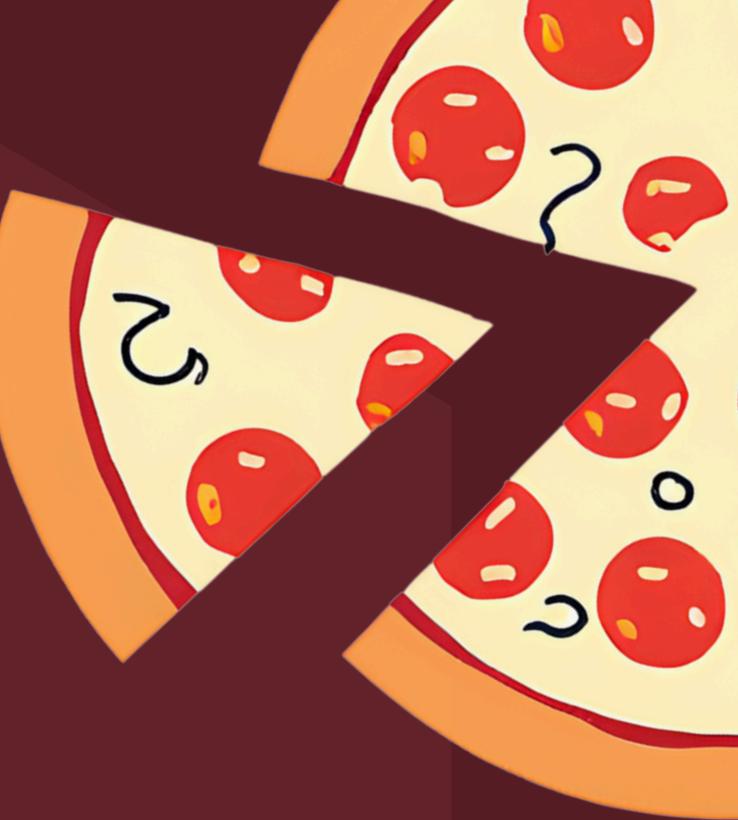
Result Grid

	total_orders
21350	

Q2-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 pizzas.pizza_id = order_details.pizza_id
```

Result Grid	
	total_sales
▶	817860.05



Q3-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



Q4-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

Q5-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;
```

	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

Q6-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

Q7-Determine the distribution of orders by hour of the day.

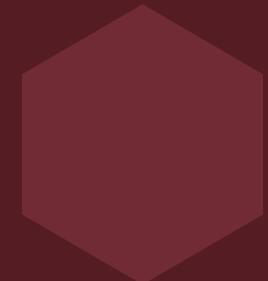


```
SELECT  
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

Result Grid

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

Q8-Join relevant tables to find the category-wise distribution of pizzas.



```
select category, count(name) from pizza_types  
group by category;
```

Result Grid | Filter R

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Q9-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(order_details.quantity) AS quantity
```

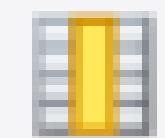
```
FROM
```

```
    orders
```

```
JOIN order_details ON orders.order_id = order_details.order_id
```

```
GROUP BY orders.order_date) AS order_quantity;
```

Result Grid



Filter Rows

avg_pizza_ordered_per_day
138

Q10-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;
```

Result Grid | Filter Rows:

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

Q11-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 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 revenue desc;
```

Result Grid |

	category	revenue
	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Q12-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	2713.850000000004
	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	34343.5000000001
	2015-01-16	36937.65000000001
	2015-01-17	39001.7500000001
	2015-01-18	40978.60000000006
	2015-01-19	43365.75000000001
	2015-01-20	45763.65000000001
	2015-01-21	47804.2000000001
	2015-01-22	50300.9000000001
	2015-01-23	52724.60000000006

Q13-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 rnk  
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 rnk<=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	The Classic Deluxe Pizza
	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!!

19 JULY, 2024