

PIZZA SALES

ANALYSIS USING SQL





HELLO !

My name is Hariom Sinhmar. In this project, I have utilized SQL queries to solve the questions to uncover the important trends related to pizza sales

OBJECTIVE

This project analyzes the pizza sales data to find out how well the restaurant is doing and what trends there are. This information is used to make smarter decisions and better plans for the future to grow sales.



QUESTIONS

1. Retrieve the total no. 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.

Q.1: RETRIEVE THE TOTAL NO. OF ORDERS PLACED.

```
SELECT
    COUNT(order_id) AS total_orders
FROM
    pizzahut.orders
-- Total no. of orders are 21350
```

Q.2: CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

SELECT

```
ROUND(SUM(order_details.quantity * pizzas.price),  
2) AS total_sales
```

FROM

```
pizzahut.order_details
```

JOIN

```
pizzahut.pizzas ON pizzas.pizza_id = order_details.pizza_id
```

-- Hence the total revenue generated from Pizza Sales is 817860.05

Q.3: IDENTIFY THE HIGHEST PRICED PIZZA

```
SELECT pt.name, pz.price
FROM pizzahut.pizza_types AS pt
JOIN pizzahut.pizzas AS pz ON pt.pizza_type_id = pz.pizza_type_id
ORDER BY price DESC
LIMIT 1
-- Hence the highest priced pizza is "The Greek Pizza" with the price 35.95
```

Q.4 : IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
SELECT
    pizzas.size, COUNT(ord.quantity) AS order_count
FROM
    pizzahut.pizzas
        JOIN
    pizzahut.order_details AS ord ON pizzas.pizza_id = ord.pizza_id
GROUP BY size
ORDER BY order_count DESC
-- Hence the most common pizza size ordered is "L" .
```

Q.5: LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
SELECT pt.name, SUM(quantity) AS Total_Quantities
FROM pizzahut.pizza_types AS pt
JOIN (
    SELECT pizzas.pizza_type_id, order_details.quantity
    FROM pizzahut.pizzas
    JOIN pizzahut.order_details ON pizzas.pizza_id = order_details.pizza_id
    ON pt.pizza_type_id = table1.pizza_type_id
GROUP BY name ORDER BY Total_Quantities DESC Limit 5
```

name	Total_Quantities
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Q.6: JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
SELECT pizza_types.category,  
       SUM(order_details.quantity) AS Total_Quantity  
FROM   pizzahut.pizza_types  
       JOIN pizzahut.pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
       JOIN pizzahut.order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizza_types.category  
ORDER BY Total_Quantity DESC
```

category	Total_Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q.7: DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT  
    HOUR(order_time) AS Hours, COUNT(order_id) AS Total_Orders  
FROM  
    orders  
GROUP BY hours  
ORDER BY Total_Orders DESC
```

TOP 5 HOURS

Hours	Total_Orders
12	2520
13	2455
18	2399
17	2336
19	2009

Q.8: 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

Q.9: GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
Select round(avg(quantity),0)
as avg_pizza_order_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
-- Hence the Average pizza Orders per day are 138.
```

Q.10: DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
Select pizza_types.name,  
sum(pizzas.price*order_details.quantity) as Revenue  
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 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

Q.11: 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(quantity*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
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q.12: ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date,  
round(sum(revenue) over(order by order_date), 2) 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 Order_date) as sales
```

Q.13: DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
select name, category, 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(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 table1) as table2
where rn<= 3
```

THANK YOU

