

PIZZA SALES DATA ANALYSIS PROJECT USING MY SQL

Ppt by - Mehak Sharma



ABOUT THE PROJECT

This project simulates a real-world business scenario using SQL to analyze sales data. It involves importing and cleaning data, then performing analysis to find sales trends, top products, and high-value customers. Key SQL concepts like joins, aggregate functions, and window functions were used. While not connected to a live system, the project closely reflects real-time business data analysis practices.





SQL QUERIES FOR PIZZA SALES ANALYSIS AND BUSINESS INSIGHTS

QUERY QUESTION 1

Retrieve the total number of order placed.

```
select count(order_id) as total_orders from orders;
```

QUERY QUESTION 2

Calculate the total revenue generated from pizza Sales.

```
SELECT  
    ROUND(SUM(o.quantity * p.price), 2) AS total_revenue  
FROM  
    orders_details AS o  
    JOIN  
    pizzas AS p ON o.pizza_id = p.pizza_id;
```

QUERY QUESTION 3

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;
```

QUERY QUESTION 4

Identify the most common pizza size ordered.

```
SELECT  
    p.size, COUNT(o.order_details_id) AS num_pizza  
FROM  
    pizzas AS p  
    JOIN  
    orders_details AS o ON p.pizza_id = o.pizza_id  
GROUP BY p.size  
ORDER BY num_pizza DESC  
LIMIT 1;
```

QUERY QUESTION 5

List the 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;
```

QUERY QUESTION 6

Join the necessary tables to find the total quantity of each pizza 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;
```

QUERY QUESTION 7

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);
```

QUERY QUESTION 8

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

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

QUERY QUESTION 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;
```

QUERY QUESTION 10

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;
```

QUERY QUESTION 11

Calculate the %age 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;
```

QUERY QUESTION 12

Analyse 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;
```

QUERY QUESTION 13

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((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 rn <= 3;
```



MY CONTACT

📞 8595596902

🌐 mehaksharma272003@gmail.com

THANK YOU!

Project by - Mehak Sharma

Pizza Sales Data Analysis project

