

PIZZA SALES ANALYSIS



INTRODUCTION

Hi good people, let me introduce my self. I am Snehal Devkar, BSc graduate. In this project I utilise sql queries to solve questions related to pizza sales.

PROJECT INTRODUCTION

The Pizza Sales Data Analysis Project is designed to analyze and visualize sales trends, customer preferences, and revenue performance in a pizza restaurant. By exploring datasets that include order details, pizza types, prices, and quantities,

BASIC

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

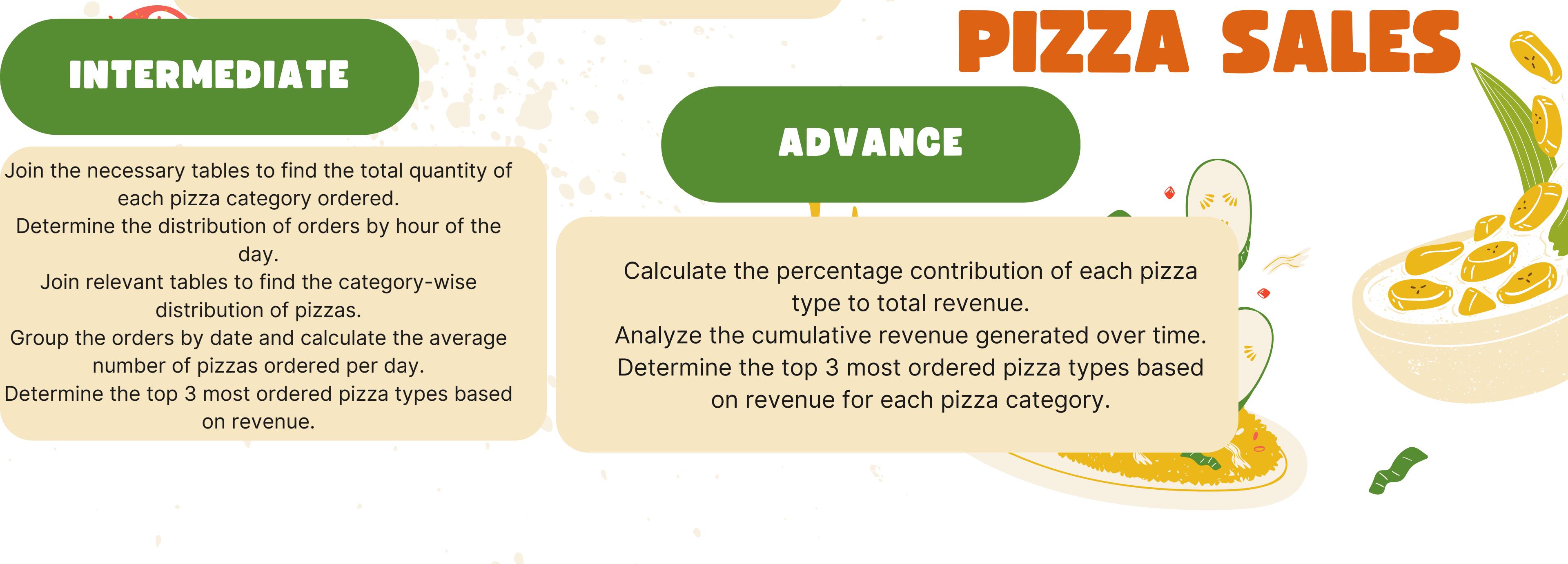
INTERMEDIATE

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

ADVANCE

- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

HERE IS QUESTIONS ON PIZZA SALES



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

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

	total_num_of_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 pizzas.pizza_id = order_details.pizza_id;
```

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

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

SELECT

```
pizzas.size, COUNT(order_details.order_details_id) as total_ordered_quantity
```

FROM

```
order_details
```

JOIN

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

```
GROUP BY (pizzas.size);
```

	size	total_ordered_quantity
▶	M	15385
	L	18526
	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 total_ordered_quantity

FROM

 pizzas

 JOIN

 pizza_types 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 (SUM(order_details.quantity)) DESC

LIMIT 5;

	name	total_ordered_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, SUM(order_details.quantity) as total_ordered_quantity
```

FROM

```
order_details
```

JOIN

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

JOIN

```
pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
```

GROUP BY (pizza_types.category);

	category	total_ordered_quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

SELECT

HOUR(order_time), COUNT(order_id)

FROM

orders

GROUP BY HOUR(order_time);

	HOUR(order_time)	COUNT(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399

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 average_order_quantity  
FROM  
(SELECT  
    orders.order_date, SUM(order_details.quantity) AS quantity  
FROM  
    orders  
JOIN order_details ON order_details.order_id = orders.order_id  
GROUP BY orders.order_date) as order_quantity;
```

	average_order_quantity
▶	138

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

SELECT

```
order_details.pizza_id,  
SUM(ROUND(pizzas.price * order_details.quantity, 2)) as revenue
```

FROM

```
order_details
```

JOIN

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

GROUP BY order_details.pizza_id

order by SUM(ROUND(pizzas.price * order_details.quantity, 2)) desc

```
limit 3;
```

	pizza_id	revenue
▶	thai_ckn_1	29257.5
	five_cheese_1	26066.5
	four_cheese_1	23622.200000000554

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

SELECT

```
    pizza_types.category,  
    ROUND(SUM(pizzas.price * order_details.quantity) / (SELECT  
        ROUND(SUM(pizzas.price * order_details.quantity),  
        2))  
FROM  
    pizzas  
    JOIN  
        order_details 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 pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizza_types.category;
```

	category	revenue
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

ANALYZE YOUR BUSINESS REVENUE WITH ME.

select order_date,

order_date	cummulative_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

GROUP BY orders.order_date) as sales;

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

```
SELECT
    category,
    name,
    revenue,
    rn
FROM (
    SELECT
        pizza_types.category,
        pizza_types.name,
        SUM(order_details.quantity * pizzas.price) AS revenue,
        RANK() OVER (PARTITION BY pizza_types.category ORDER BY SUM(order_details.quantity * pizzas.price) DESC) AS rn
    FROM
        order_details
    JOIN
        pizzas ON order_details.pizza_id = pizzas.pizza_id
    JOIN
        pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    GROUP BY
        pizza_types.category,
        pizza_types.name
) AS a
WHERE
    rn <= 3;
```

	category	name	revenue	rn
▶	Chicken	The Thai Chicken Pizza	43434.25	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	32273.25	2
	Classic	The Pepperoni Pizza	30161.75	3
	Suopreme	The Soicy Italian Pizza	34831.25	1



**THANK
YOU!**

