Pizza Sales Analysis Report

Problem Statement: Pizza Sales Analysis

Business Context:

A pizza restaurant wants to improve its **sales performance, customer satisfaction, and operational efficiency**. By analyzing historical **sales data**, the business aims to gain insights into **customer preferences, order patterns, and revenue trends**. This will help optimize **menu offerings, pricing strategies, and promotional campaigns**.

Problem Statement:

The objective of this project is to use **SQL queries** to analyze **pizza sales data** and extract meaningful insights that can drive business decisions. Specifically, the project focuses on:

1. Calculating **total orders and overall revenue** to assess business performance.
2. Identifying **best-selling pizzas** based on order count and revenue.
3. Determining **most preferred pizza sizes** to optimize inventory and pricing.
4. Analyzing **peak ordering times** to enhance staffing and promotional strategies.
5. Evaluating **category-wise sales distribution** to understand customer choices.
6. Ranking pizzas based on **revenue contribution** to focus on high-profit items.
7. Tracking **cumulative revenue trends** to monitor business growth over time.

Business Impact & Expected Outcomes:

- **Menu Optimization:** Prioritize best-selling pizzas and reconsider underperforming items.
- **Revenue Growth:** Focus on high-revenue items and develop effective pricing strategies.
- **Marketing Strategy:** Run targeted promotions during peak hours and for high-demand pizzas.

Retrieve the total number of orders placed

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

total_orders
21350

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
```

total_sales
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 pizzas.price DESC
LIMIT 1;
```

name	price
The Greek Pizza	35.95

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

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

```
SELECT pizza_types.name, order_details.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;
```

name	quantity
The Hawaiian Pizza	1
The Classic Deluxe Pizza	1
The Five Cheese Pizza	1
The Italian Supreme Pizza	1
The Mexicana Pizza	1
The Thai Chicken Pizza	1
The Italian Supreme Pizza	1
The Prosciutto and Arugula Pizza	1
The Italian Supreme Pizza	1
The Italian Supreme Pizza	1
The Barbecue Chicken Pizza	1
The Greek Pizza	1
The Spinach Supreme Pizza	1

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

Determine the distribution of orders by hour of the day.

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

hour	COUNT(order_id)
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
```

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_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;
```

avg_pizza_ordered_per_day
138

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;
```

	name	revenue
▶	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
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                SUM(order_details.quantity * pizzas.price)
            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

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_details_id
group by orders.order_date) as sales;
```

	order_date	cum_revenue
▶	2015-01-01	1171.45
	2015-01-02	2316.1000000000004
	2015-01-03	3433.8
	2015-01-04	4341.8
	2015-01-05	5247.25

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;
```

	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 Pizza	32273.25

Result 4