PIZZA

SQL For Data Analysis
Full Portfolio Project

INTRODUCTION

Pizza Sales Data Analysis Overview

This analysis delves into various aspects of pizza sales data, from basic metrics to advanced insights. The objectives are divided into three tiers: Basic, Intermediate, and Advanced, each focusing on different levels of data aggregation and interpretation.

Basic Analysis

1. **Total Number of Orders**: Calculate the overall count of orders placed.
2. **Total Revenue**: Sum up the revenue generated from all pizza sales.
3. **Highest-Priced Pizza**: Identify which pizza has the highest price.
4. **Most Common Pizza Size**: Determine the pizza size that is ordered most frequently.
5. **Top 5 Most Ordered Pizza Types**: List the five pizza types with the highest order quantities along with their respective quantities.

Intermediate Analysis

1. **Total Quantity per Pizza Category**: Join necessary tables to ascertain the total number of pizzas ordered within each category.
2. **Order Distribution by Hour**: Analyze how orders are distributed across different hours of the day.
3. **Category-Wise Pizza Distribution**: Join relevant tables to find out how pizzas are distributed across different categories.
4. **Average Number of Pizzas per Day**: Group orders by date to compute the average number of pizzas ordered each day.
5. **Top 3 Pizza Types by Revenue**: Determine the top three pizza types generating the highest revenue.

Advanced Analysis

1. **Percentage Contribution to Total Revenue**: Calculate the revenues contribution percentage of each pizza type to the total revenue.
2. **Cumulative Revenue Over Time**: Analyze how the total revenue accumulates over a specified period.
3. **Top 3 Pizza Types by Revenue in Each Category**: Determine the top three pizza types based on revenue within each pizza category.

By conducting these analyses, we aim to provide comprehensive insights into pizza sales patterns, revenue drivers, and customer preferences. This will help in making informed decisions for inventory management, marketing strategies, and sales optimization.

Pizza Sales Data Analyst

01

Basic Analysis

Total orders, revenue calculation, highest-priced pizza, most common pizza size, and top 5 most ordered pizza types.

02

Intermediate Analysis

Quantity per pizza category, hourly order distribution, category-wise distribution, daily average orders, and top 3 pizza types by revenue.

03

Advanced Analysis

Percentage revenue contribution by pizza type, cumulative revenue trends, and top 3 pizza types by revenue for each category.

01

Basic Analysis

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.

1. Retrieve the total number of orders placed.

```
Don't Limit
1   -- Q.1 Retrieve the total number of orders placed.
2
3   select count(order_id) as total_orders from orders;
```

Result Grid

	total_orders
▶	21350

2. Calculate the total revenue generated from pizza sales.

```sql
-- Q. 2 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

3. Identify the highest-priced pizza.

```sql
-- Q.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;
```

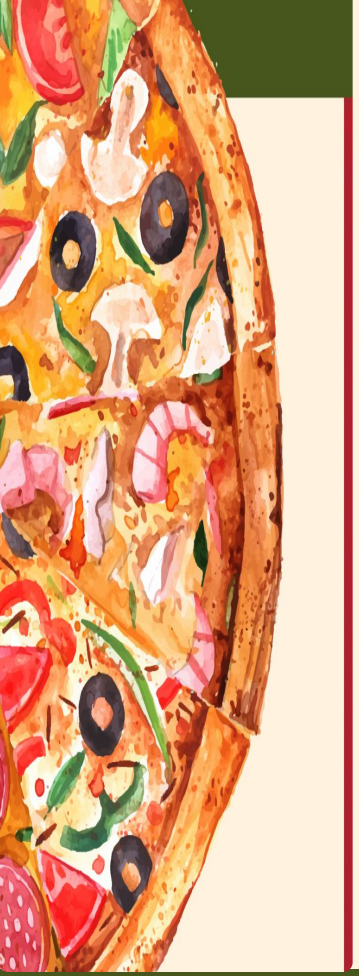
4. Identify the most common pizza size ordered.

```sql
-- Q.4 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;
```

Result Grid

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28



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

```sql
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;
```

Result Grid | Filter Rows:

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

02

Intermediate Analysis

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

1. Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
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;
```

Result Grid | Filter

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

2. Determine the distribution of orders by hour of the day.

```sql
-- Q.7 Determine the distribution of orders by hour of the day

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

Result Grid | Filter

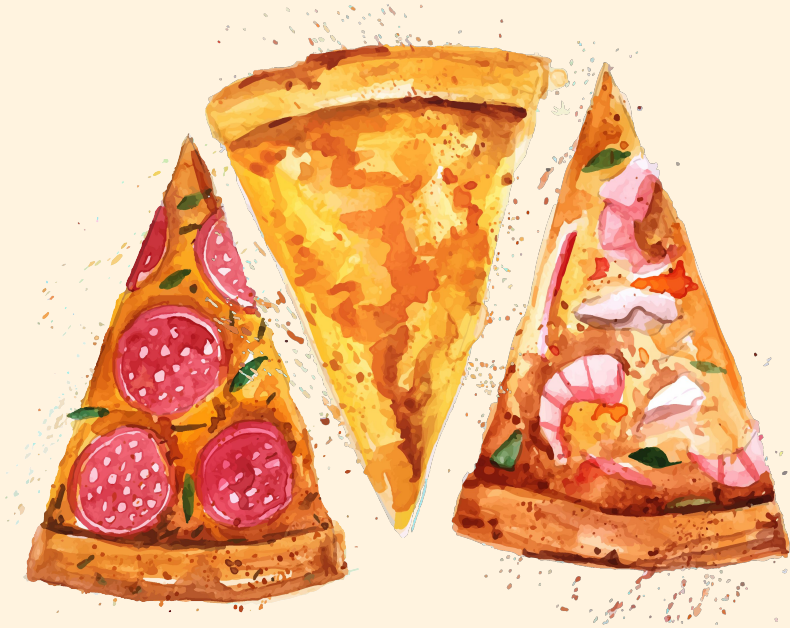
hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663

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

```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category
```

Result Grid | Filter Rows:

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



4. Group the orders by date and calculate the average number of pizzas ordered per day.

```sql
SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        orders.date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS order_quantity;
```

Result Grid | Filter Rows

	ROUND(AVG(quantity), 0)
▶	138

5. Determine the top 3 most ordered pizza types based on revenue.

```sql
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:

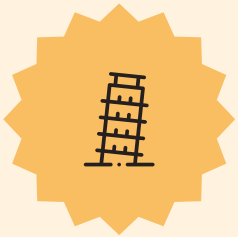
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

3. Advanced Analysis

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

1. Calculate the percentage contribution of each pizza type to total revenue.

```sql
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 pizzas.pizza_type_id = pizza_types.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 | Filter

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

2. Analyze the cumulative revenue generated over time.

```sql
select date,
sum(revenue) over(order by date) as cum_revenue
from
(select orders.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.date) as sales;
```

Result Grid | Filter Rows:

date	cum_revenue
1/1/2015	2713.8500000000004
1/10/2015	5177.8
1/11/2015	7050.1
1/12/2015	8969.150000000001
1/13/2015	11018.750000000002
1/14/2015	13546.150000000001
1/15/2015	15530.95
1/16/2015	18125.100000000002
1/17/2015	20189.200000000004
1/18/2015	22166.050000000003
1/19/2015	24553.200000000004
1/2/2015	27285.100000000006
1/20/2015	29683.000000000007

3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
(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;
```

Result Grid | Filter Rows:

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
The Pepperoni Pizza	30161.75

THANKS!

Do you have any questions?

payalumate@gmail.com
https://github.com/payalumate

Payal Umate