MySQL project

Pizza sales analysis

Divyansh Shah

ABOUT THE PROJECT

This project involves analyzing a dataset of pizza sales to extract actionable insights using MySQL. The objective is to uncover patterns in customer preferences, revenue generation, and ordering trends, which can be used to improve business strategies for a pizza chain. The analysis is divided into three levels of complexity: Basic, Intermediate, and Advanced.

Tools and Techniques used:

- Database: MySQL
- Query Techniques: Aggregations (SUM, COUNT, AVG), Joins, Subqueries, Grouping, Ranking, and Date-Time functions.
- Tables Analyzed: Orders, Order details, Pizza types, and Pizzas.

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

INTERMEDIATE ANALYSIS

1. Find the Total Quantity of Each Pizza Category Ordered
2. Determine the Distribution of Orders by Hour of the Day
3. Group Orders by Date and Calculate the Average Number of Pizzas Ordered Per Day
4. Top 3 Most Ordered Pizza Types Based on Revenue

ADVANCED ANALYSIS

1. Percentage Contribution of Each Pizza Type to Total Revenue
2. Analyze the Cumulative Revenue Generated Over Time
3. Top 3 Most Ordered Pizza Types Based on Revenue for Each Pizza Category

Retrieve the Total Number of Orders Placed

QUERY:

```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

OUTPUT:

	total_orders
▶	21350

Basic

Calculate the Total Revenue Generated from Pizza Sales

QUERY:

```
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;
```

OUTPUT:

total_revenue
655250.45

Basic

Identify the Highest-Priced Pizza

QUERY:

```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

OUTPUT:

	name	price
▶	The Greek Pizza	35.95

Basic

Identify the Most Common Pizza Size Ordered

QUERY:

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS number_of_orders
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY number_of_orders DESC
LIMIT 1;
```

OUTPUT:

size	number_of_orders
L	14857

Basic

List the Top 5 Most Ordered Pizza Types Along with Their Quantities

QUERY:

```sql
SELECT
    pizza_types.name, COUNT(order_details.quantity) AS qauntity
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 qauntity DESC
LIMIT 5;
```

OUTPUT:

name	qauntity
The Barbecue Chicken Pizza	1929
The Pepperoni Pizza	1914
The Classic Deluxe Pizza	1910
The Hawaiian Pizza	1871
The Thai Chicken Pizza	1830

Basic

Find the Total Quantity of Each Pizza Category Ordered

QUERY:

```sql
SELECT
    pizza_types.category,
    COUNT(order_details.quantity) AS qauntity
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
ORDER BY qauntity DESC;
```

OUTPUT:

	category	qauntity
▶	Classic	11685
	Supreme	9404
	Veggie	9212
	Chicken	8641

Intermediate

Determine the Distribution of Orders by Hour of the Day

QUERY:

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

OUTPUT:

HOUR(order_time)	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
10	8
9	1

Intermediate

Group Orders by Date and Calculate the Average Number of Pizzas Ordered Per Day

QUERY:

```sql
SELECT
    ROUND(AVG(quantity), 0) as avg_pizzas
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;
```

OUTPUT:

avg_pizzas
139

Intermediate

Top 3 Most Ordered Pizza Types Based on Revenue

QUERY:

```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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

OUTPUT:

name	revenue
The Barbecue Chicken Pizza	34675
The Thai Chicken Pizza	34286.25
The California Chicken Pizza	32725.5

Intermediate

Percentage Contribution of Each Pizza Type to Total Revenue

QUERY:

```sql
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) * 100 / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2)
                FROM
                    order_details
                        JOIN
                    pizzas ON pizzas.pizza_id = order_details.pizza_id)),
            2) AS percentage_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
ORDER BY percentage_revenue DESC;
```

OUTPUT:

category	percentage_revenue
Classic	26.92
Supreme	25.42
Chicken	23.87
Veggie	23.8

Advanced

Analyze the Cumulative Revenue Generated Over Time

QUERY:

```sql
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
    orders ON order_details.order_id = orders.order_id
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY orders.order_date) as sales;
```

OUTPUT:

order_date	cum_revenue
2015-01-01	2713.8500000000004
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
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003

and so on...

Advanced

Top 3 Most Ordered Pizza Types Based on Revenue for Each Pizza Category

QUERY:

```sql
SELECT category,name, revenue, rn 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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category , pizza_types.name) as a) as b
where rn<=3;
```

OUTPUT:

category	name	revenue	rn
Chicken	The Barbecue Chicken Pizza	34675	1
Chicken	The Thai Chicken Pizza	34286.25	2
Chicken	The California Chicken Pizza	32725.5	3
Classic	The Classic Deluxe Pizza	30180	1
Classic	The Hawaiian Pizza	25513.5	2
Classic	The Pepperoni Pizza	24340.5	3
Supreme	The Spicy Italian Pizza	27888	1
Supreme	The Italian Supreme Pizza	26781.75	2
Supreme	The Sicilian Pizza	24411.25	3
Veggie	The Four Cheese Pizza	25701.80000000045	1
Veggie	The Five Cheese Pizza	21589.5	2
Veggie	The Mexicana Pizza	20934.25	3

Advanced

CONCLUSION

Through this analysis, valuable insights were uncovered, such as the most popular pizzas, high-revenue earners, and ordering patterns by time and category. These findings can help optimize the menu, adjust pricing strategies, and improve inventory management for enhanced profitability and customer satisfaction.

THANK YOU