Pizza sales Analysis Using Sql

BY NISHANT GORE

Description Of Projet

Pizza Sales Analysis

This project analyzes pizza sales data to uncover trends, revenue insights, and customer preferences. Using four datasets, we explored order patterns, revenue generation, and pizza popularity. The findings include:

1. Total orders, revenue, and popular pizza types.
2. Distribution of orders by time and category.
3. Top-performing pizzas by revenue.

Key SQL concepts applied:

1. Data extraction with SELECT queries.
2. Joins to combine multiple datasets.
3. Aggregations (SUM, COUNT, AVG) for insights.
4. Filtering data using WHERE and HAVING clauses.
5. Grouping data with GROUP BY.
6. Calculating percentages and rankings for advanced analysis.

Key insights support strategic decision-making for pricing, promotions, and operational efficiency.

TABLES FOR DATASETS

pizza.csv dataset

```
mysql> create table pizza(
 -> pizza_id varchar(30) primary key
-> pizza_type_id varchar(20),
-> size varchar(10),
 -> price float);

  Query OK, 0 rows affected (0.11 sec)
```

order.csv dataset

```
mysql> create table orders(
    -> order_id int primary key,
    -> date date,
    -> time time
    );

  Query OK, 0 rows affected (0.04 sec)
```

order_details .csv dataset

```
mysql> create table order_details(
-> orders_details_id int primary key,
  -> order_id int,
  -> pizza_id varchar(30),
  -> quantity int
);

  Query OK, 0 rows affected (0.04 sec)
```

pizza_types .csv dataset

```
mysql> create table pizza_types(
 -> pizza_type_id varchar(30) primary key,
 -> name varchar(50),
 -> category varchar(30),
 -> ingredient varchar(200));

Query OK, 0 rows affected (0.03 sec)
```

Retrieve the total number of orders placed.

QUERY:=>

select count(order_id) as total_orders from orders;

output:->

```
+--------------+
| total_orders |
+--------------+
|        21350 |
+--------------+
```

Calculate the total revenue generated from pizza sales

QUERY:=>

select round(sum(order_details.quantity* pizza.price),2)as
total_revenue from order_details join pizza on
order_details.pizza_id=pizza.pizza_id;

output:->

```
+----------------+
| total_revenue |
+----------------+
|    817860.05 |
+----------------+
```

Identify the highest-priced pizza.

QUERY:=>

select   pizza_types.name, pizza.price from pizza_types join pizza on
pizza_types.pizza_type_id =pizza.pizza_type_id order by pizza.price
desc limit 1;

output:->

```
+----------------+-------+
| name           | price |
+----------------+-------+
| The Greek Pizza | 35.95 |
+----------------+-------+
```

Identify the most common pizza size ordered.

Queriies:=>

select pizza.size,count(order_id) from pizza join order_details on pizza.pizza_id=order_details.pizza_id group by (pizza.size) order by count(order_id) desc limit 1;

Output:=>

```
+------+----------------+
| size | count(order_id) |
+------+----------------+
| L    |          18526 |
+------+----------------+
1 row in set (0.10 sec)
```

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

QUERY:=>

select pizza_types .name,sum(order_details.quantity) from
pizza_types join pizza on
pizza_types.pizza_type_id=pizza.pizza_type_id
-> join order_details on order_details.pizza_id=pizza.pizza_id group
by pizza_types.name order by sum(order_details.quantity) desc limit 1;

output:->

```
+---------------------+---------------------------+
| name                | sum(order_details.quantity) |
+---------------------+---------------------------+
| The Classic Deluxe Pizza |                    2453 |
+---------------------+---------------------------+
```

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

Queriies:=>

```
select pizza_types .category,sum(order_details.quantity) as quantity
                from pizza_types join pizza on
            pizza_types.pizza_type_id=pizza.pizza_type_id
        -> join order_details on order_details.pizza_id=pizza.pizza_id
group by pizza_types.category order by sum(order_details.quantity)
                            desc ;
```

Output:=>

```
+----------+----------+
| category | quantity |
+----------+----------+
| Classic  |   13529  |
| Supreme  |   11987  |
| Chicken  |   11050  |
| Veggie   |   10652  |
| Mushroom |   1359   |
+----------+----------+
```

Calculate the total revenue generated from pizza sales

QUERY:=>

select round(sum(order_details.quantity* pizza.price),2)as
total_revenue from order_details join pizza on
order_details.pizza_id=pizza.pizza_id;

output:->

```
+----------------+
| total_revenue |
+----------------+
|    817860.05 |
+----------------+
```

Determine the distribution of orders by hour of the day

Queriies:=>

select hour(time),count(order_id) as orders from orders group
by(hour(time)) order by hour(time) asc;

Output:=>

```
+----------+--------+
|hour(time)| orders |
+----------+--------+
|        9 |      1 |
|       10 |      8 |
|       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 |
+----------+--------+
```

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

Queriies:=>

select date,sum(quantity) from orders join order_details on
orders.order_id=order_details.order_id group by(date) limit 5;

Output:=>

```
+------------+---------------+
|date     | sum(quantity)|
+------------+---------------+
| 2015-01-01|        162|
| 2015-01-02|        165|
| 2015-01-03|        158|
| 2015-01-04|        106|
| 2015-01-05|        125|
+------------+---------------+
```

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

Queriies:=>

select pizza_types.name
,round(sum(pizza.price*order_details.quantity),0) as revenue from
pizza_types join pizza on pizza_types.pizza_type_id=pizza.pizza_type_id
join order_details on pizza.pizza_id=order_details.pizza_id group
by(pizza_types.name) order by revenue desc limit 3;

Output:=>

```
+-------------------------------+----------+
| name                          | revenue  |
+-------------------------------+----------+
| The Thai Chicken Pizza        |   43434  |
| The Barbecue Chicken Pizza    |   42768  |
| The California Chicken Pizza  |   41410  |
+-------------------------------+----------+
```

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

Queriies:=>

1) select sum(pizza.price*order_details.quantity) into @total from pizza join order_details on pizza.pizza_id=order_details.pizza_id ;

2) select pizza_types.category ,round((sum(pizza.price*order_details.quantity)/@total)*100,0) as percentage_from_revenue from pizza_types join pizza on pizza_types.pizza_type_id=pizza.pizza_type_id join order_details on pizza.pizza_id=order_details.pizza_id group by(pizza_types.category) order by percentage_from_revenue desc;

Output:=>

```
+-----------+-------------------------+
| category  | percentage_from_revenue |
+-----------+-------------------------+
| Classic   |                      25 |
| Supreme   |                      25 |
| Chicken   |                      24 |
| Veggie    |                      22 |
| Mushroom  |                       2 |
+-----------+-------------------------+
```

Analyze the cumulative revenue generated over time

Queriies:=>
select sales.date,sum(revenue) over (order by sales.date) as
commulative_revenue from(
-> select orders.date,sum(pizza.price*order_details.quantity) as revenue
from orders join order_details on orders.order_id =order_details.order_id join
pizza on order_details.pizza_id=pizza.pizza_id group by(date) order by
sum(pizza.price*order_details.quantity) desc)as sales limit 5;

Output:=>

```
+-----------+--------------------+
| date      | commulative_revenue |
+-----------+--------------------+
| 2015-01-01 | 2713.8500022888184 |
| 2015-01-02 | 5445.750003814697 |
| 2015-01-03 | 8108.150007247925 |
| 2015-01-04 | 9863.600008010864 |
| 2015-01-05 | 11929.550008773804 |
+-----------+--------------------+
```

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

Queriies:=>

```sql
SELECT
    -> pizza_types.category,
    -> pizza_types.name,
    -> SUM(pizza.price * order_details.quantity) AS revenue
    -> FROM
    -> pizza_types
    -> JOIN
    -> pizza ON pizza_types.pizza_type_id = pizza.pizza_type_id
    -> JOIN
    -> order_details ON pizza.pizza_id = order_details.pizza_id
    -> GROUP BY
    -> pizza_types.category, pizza_types.name
    -> ORDER BY
    -> revenue DESC;
```

Output:=>

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Supreme	The Spicy Italian Pizza	34831.25
Chicken	The Southwest Chicken Pizza	34705.75
Supreme	The Italian Supreme Pizza	33476.75
Classic	The Hawaiian Pizza	32273.25
Veggie	The Four Cheese Pizza	32265.70200402832
Supreme	The Sicilian Pizza	30940.5
Classic	The Pepperoni Pizza	30161.75
Classic	The Greek Pizza	28454.100011362305
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5
Supreme	The Pepper Salami Pizza	25529
Classic	The Italian Capocollo Pizza	25094
Veggie	The Vegetables + Vegetables Pizza	24374.75
Supreme	The Prosciutto and Arugula Pizza	24293.25
Classic	The Napolitana Pizza	24087
Veggie	The Spinach and Feta Pizza	23271.25
Classic	The Big Meat Pizza	22968
Mushroom	"The Pepperoni	18834.5
Chicken	The Chicken Alfredo Pizza	16900.25
Chicken	The Chicken Pesto Pizza	16701.75
Supreme	The Soppressata Pizza	16425.75
Veggie	The Italian Vegetables Pizza	16019.25
Supreme	The Calabrese Pizza	15934.25
Veggie	The Spinach Pesto Pizza	15596
Veggie	The Mediterranean Pizza	15360.5
Supreme	The Spinach Supreme Pizza	15277.75
Supreme	The Brie Carre Pizza	11588.499813079834

Thanks a lot