Exploratory Analysis of the PIZZA PALACE

Report by: Farrukh Nizam Arain
Designation: Data Analyst
Date: 28th February 2023

Introduction:

Pizza Palace is a fictious regional pizza maker located in the suburbs of a metropolitan area. Pizza Palace strives to deliver healthy, speciality pizzas and side dishes. They offer a wide spectrum of competitive pricing. Pizza Palace targets customers like commuters and workers who lack the time or ability to prepare a family dinner.

Problem Statement:

The purpose of my assessment in this project is to answer the questions below by analyzing the different aspects of sales in the fictious Pizza Palace. The important questions were:

1. How many customers do we have each day? Are there any peak hours?
2. How many pizzas are typically in an order? Do we have any bestsellers?
3. How much money did we make this year? Can we identify any seasonality in the sales?
4. Are there any pizzas we should take of the menu, or any promotions we could leverage?

Preparation of the Datasets:

Public datasets were downloaded at the following link provided by Maven Analytics under this license. No issues with bias and credibility were found with the data through the methodology of ROCCC.

Following are the list of files (in CSV format) and the descriptions:

Filename	Description
Order_details.csv	Details of the Pizza order of a year
orders.csv	Dates and times of the orders received
pizza_types.csv	Types of the Pizzas baked in the restaurant
pizzas.csv	Pricing & size details of all the pizzas

After inspecting the datasets provided by the DB administrators hired by the company, it was seen that they are normalized, and I proceeded to processing the data.

Processing of the Datasets:

The tools that I used for data processing are:

- Microsoft Excel 365
- SQL Server 2022
- SSMS 19

To remove any ambiguity of the fields used in the database, I created a data dictionary to make it easier for the reader to better understand it .

Table	Field	Description
orders	order_id	Unique identifier for each order placed by a table
orders	date	Date the order was placed (entered the system prior to cooking & serving)
orders	time	Time the order was placed (entered the system prior to cooking & serving)
order_details	order_details_id	Unique identifier for each pizza placed within each order (pizzas of the same type and size are kept in the same row, and the quantity increases)
order_details	order_id	Foreign key that ties the details in each order to the order itself
order_details	pizza_id	Foreign key that ties the pizza ordered to its details, like size and price
order_details	quantity	Quantity ordered for each pizza of the same type and size
pizzas	pizza_id	Unique identifier for each pizza (constituted by its type and size)
pizzas	pizza_type_id	Foreign key that ties each pizza to its broader pizza type
pizzas	size	Size of the pizza (Small, Medium, Large, X Large, or XX Large)
pizzas	price	Price of the pizza in USD
pizza_types	pizza_type_id	Unique identifier for each pizza type
pizza_types	name	Name of the pizza as shown in the menu
pizza_types	category	Category that the pizza falls under in the menu (Classic, Chicken, Supreme, or Veggie)
pizza_types	ingredients	Comma-delimited ingredients used in the pizza as shown in the menu (they all include Mozzarella Cheese, even if not specified; and they all include Tomato Sauce, unless another sauce is specified)

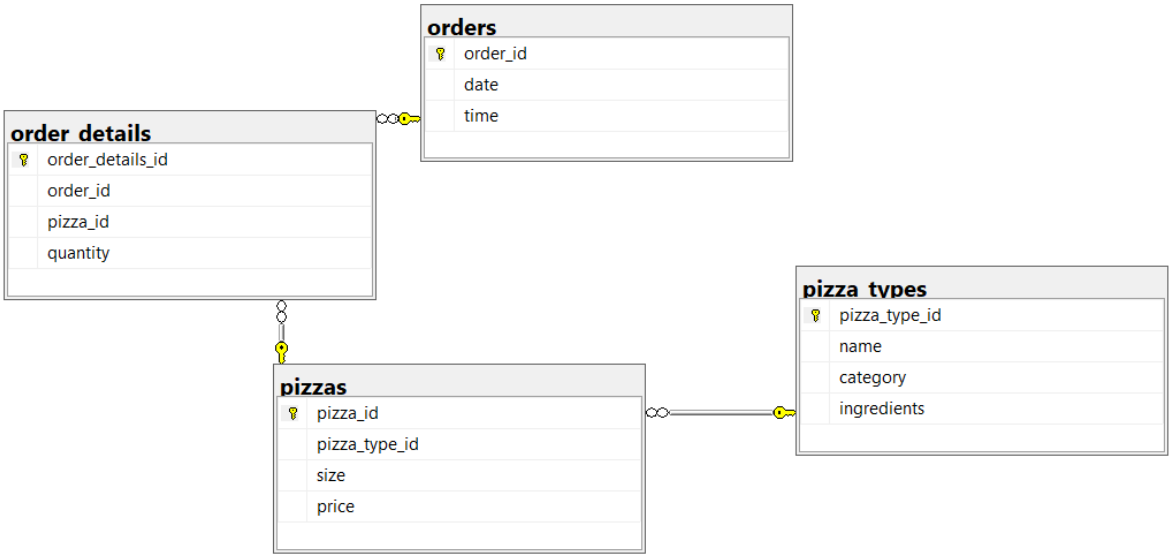
I created database named ‘pizzapalaceanalysis’, created relevant tables, formed relationships, and created database diagram in between them.

```
CREATE TABLE pizza_types (
            pizza_type_id INT NOT NULL PRIMARY KEY,
            name VARCHAR NOT NULL,
            category VARCHAR NOT NULL,
            ingredients VARCHAR NOT NULL
)

CREATE TABLE pizzas (
            pizza_id INT NOT NULL PRIMARY KEY,
            pizza_type_id INT NOT NULL FOREIGN KEy REFERENCES pizza_types(pizza_type_id),
            size VARCHAR NOT NULL,
            price FLOAT NOT NULL
)

CREATE TABLE orders (
            order_id INT NOT NULL PRIMARY KEY,
            date date NOT NULL,
            time time NOT NULL
)

CREATE TABLE order_details (
            order_details_id INT NOT NULL PRIMARY KEY,
            order_id INT NOT NULL FOREIGN KEY REFERENCES orders(order_id),
            pizza_id INT NOT NULL FOREIGN KEY REFERENCES pizzas(pizza_id),
            quantity INT NOT NULL
)
```



After laying out the structure of the table, I appended data via SQL Server's Import and Export Wizard from CSV format.

Analysis of the Datasets:

I discussed what the clients wanted from their data and what questions they want to be answered to help expand / model their existing business model for the better.

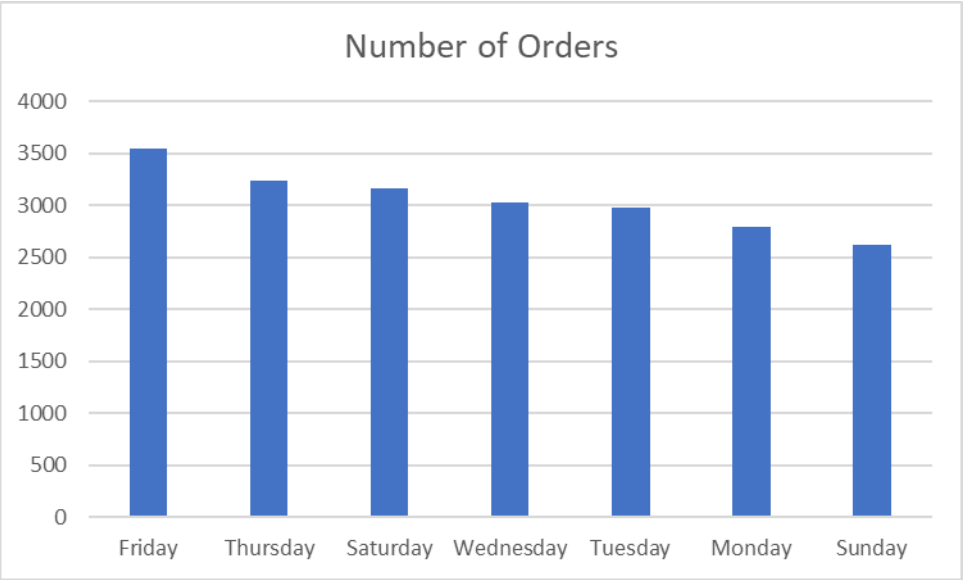
1. **How many customers do we have each day? Are there any peak hours?**
   If we count the number of orders each day throughout the years, we can conclude that Thursdays, Fridays, and Saturdays were the days when most orders were recorded. Surprisingly, Sunday was the day when least number of customers approached our restaurants.

   I queried the orders table and included the relevant columns for analysis.

```
SELECT FORMAT(CAST(date as DATE), 'dddd') AS Weekday, COUNT(FORMAT(CAST(date as DATE), 'dddd')) AS Number_of_Orders
FROM orders
GROUP BY FORMAT(CAST(date as DATE), 'dddd')
ORDER BY COUNT(FORMAT(CAST(date as DATE), 'dddd')) DESC
```

   The Tabular analysis and Visualization further confirmed my analysis. The marketing department needs to launch promotional activities to lure the customers to visit the food chain on Sundays to contribute to the overall profitability of the company.

Weekdays	Number of Orders
Friday	3538
Thursday	3239
Saturday	3158
Wednesday	3024
Tuesday	2973
Monday	2794
Sunday	2624

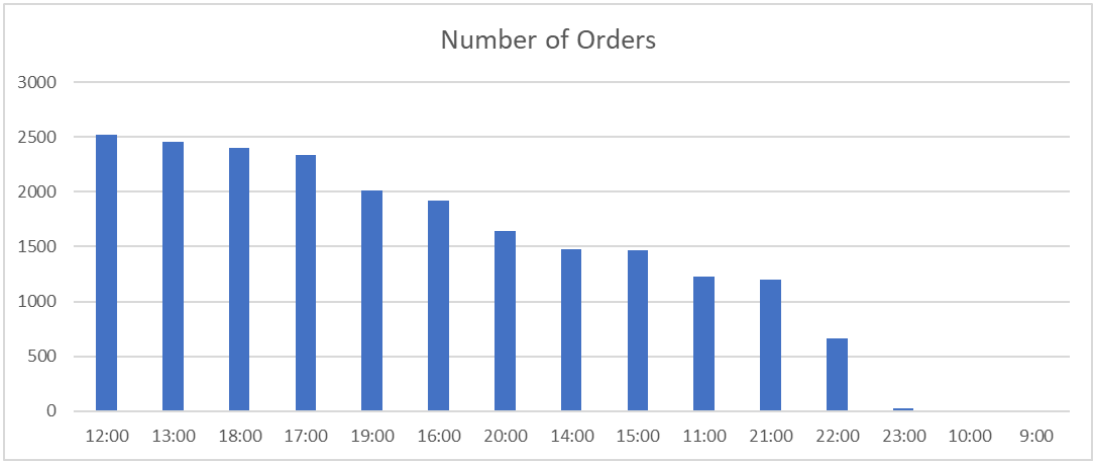


Regarding peak hours, I made use of the same orders table is extracting the time of the orders, compiled them by hours and then grouped / ordered them accordingly.

```
SELECT FORMAT(CAST(time as TIME), 'hh')+':00' AS Hours, COUNT(FORMAT(CAST(time as TIME), 'hh')) As Number_of_Orders
FROM orders
GROUP BY FORMAT(CAST(time as TIME), 'hh')
ORDER BY COUNT(FORMAT(CAST(time as TIME), 'hh')) DESC
```

Although it was encouraging to observe that pizzas were popular by the customers during their lunch time (12:00 – 13:00), the demand dipped to the lowest levels during the dinner time. It is an alarming situation because people visit these places more often than the lunch hours. Additionally, location of some of the restaurants somewhat contributed to the sales during suppertime, else the sales wouldn't have been touching near those during lunch times.

Hours	Number of Orders
12:00	2520
13:00	2455
18:00	2399
17:00	2336
19:00	2009
16:00	1920
20:00	1642
14:00	1472
15:00	1468
11:00	1231
21:00	1198
22:00	663
23:00	28
10:00	8
09:00	1



2. **How many pizzas are typically in an order? Do we have any bestsellers?**
To calculate average number of pizzas per order, I created a temporary table and named it number_of_order. The reason for the creation was to extract the quantities of pizzas and calculate the average of the quantities.

```
CREATE TABLE #number_of_order (
order_id INT,
quantity INT
)

INSERT INTO #number_of_order
SELECT order_id AS Order_ID, COUNT(quantity) AS Quantity
FROM order_details
GROUP BY order_id
ORDER BY order_id ASC

SELECT COUNT(*) AS Number_of_Orders, (SUM(quantity) / (SELECT COUNT(*) FROM #number_of_order)) AS Average_Pizzas_per_Order
FROM #number_of_order
```

According to the extracted data, it was found out that 2 pizzas were on an average ordered in each order.

Number of Orders	Average Number of Pizzas / Order
21350	2

To calculate the top sellers, I added up the different pizza types ordered and grouped them accordingly.

```
SELECT TOP 10 pizza_id, SUM(quantity) AS Quantities_Ordered
FROM order_details
GROUP BY pizza_id
ORDER BY Quantities_Ordered DESC
```

Due to the popularity of BBQ Chicken flavored pizza, both large and medium sized pizzas have collectively sold **1948** items last year, with Big Meat (Small) and Thai Chicken (Large) Pizzas coming close at second and third places, respectively.

Pizza Flavors	Quantities Ordered
big_meat_s	1914
thai_ckn_l	1410
five_cheese_l	1409
four_cheese_l	1316
classic_dlx_m	1181
spicy_ital_l	1109
7awaiian_s	1020
southw_ckn_l	1016
bbq_ckn_l	992
bbq_ckn_m	956

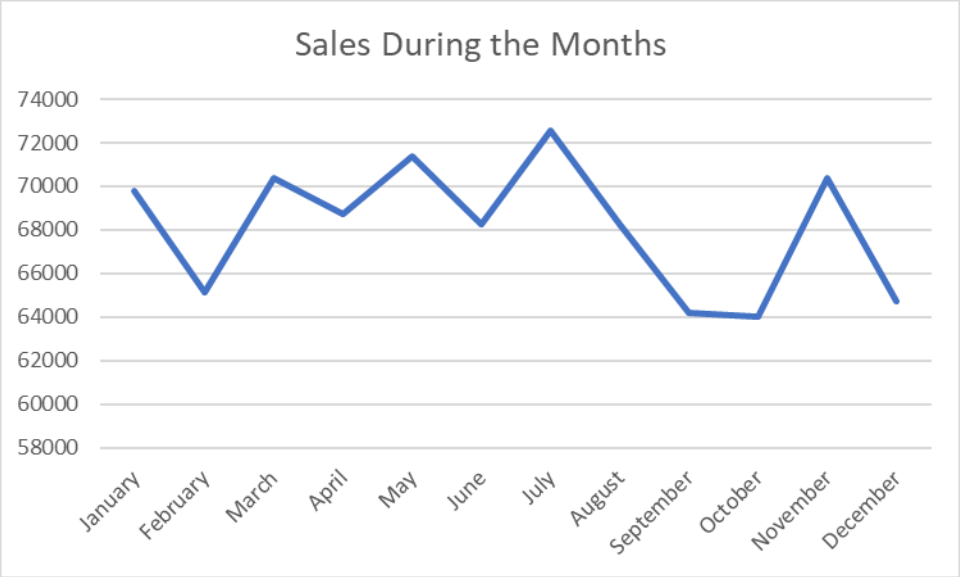
3. **How much money did we make this year? Can we identify any seasonality in the sales?**
For calculating up the sales for the year, I added up the quantities and the sales amount of all the items within order_details table.

```
SELECT SUM(quantity) As Items_Sold, SUM(order_details.quantity * pizzas.price) AS Sales
FROM order_details
INNER JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
```

Items Sold	Sales for the Year
49574	$817860.0508

Regarding finding out any patterns in seasonality in sales per month, I bifurcated the sales in all the months of the year. After analyzing and modeling data, I was observed that during the first three quarters of the year, sales hovered around $77,000 to $66,000. However, during September, October, the sales dipped to the lowest sales at just above $64,000. Interestingly, November saw a significant increase to just above $70,000, but it dipped again to around $65,000 during December.

Months	Sales Amount
January	69793.30007
February	65159.60007
March	70397.10007
April	68736.80008
May	71402.75007
June	68230.20007
July	72557.90007
August	68278.25006
September	64180.05006
October	64027.60008
November	70395.35007
December	64701.15008



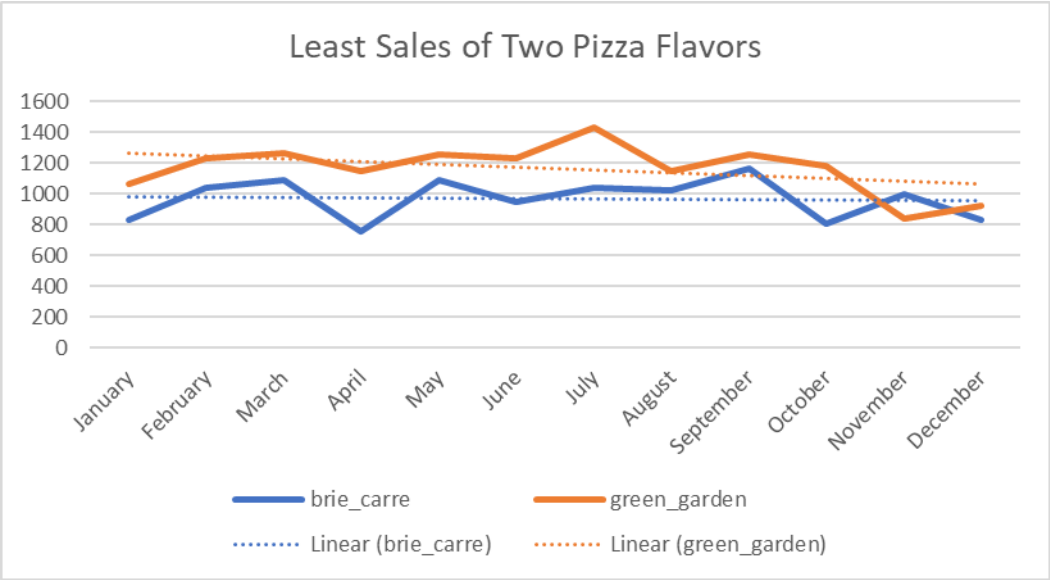
4. **Are there any pizzas we should take off the menu, or any promotions we could use?**
The last query was a bit tricky because I had to go through the sales of each pizza type throughout the last year via executing this query.

```
SELECT DATENAME(month, orders.date) AS Sales_Month, pizzas.pizza_type_id, SUM(order_details.quantity * pizzas.price) AS Sales
FROM order_details
INNER JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
INNER JOIN orders ON order_details.order_id = orders.order_id
GROUP BY DATENAME(month, orders.date), pizzas.pizza_type_id
HAVING SUM(order_details.quantity * pizzas.price) < 1500
ORDER BY
        CASE
        WHEN DATENAME(month, orders.date) = 'January' THEN 1
        WHEN DATENAME(month, orders.date) = 'February' THEN 2
        WHEN DATENAME(month, orders.date) = 'March' THEN 3
        WHEN DATENAME(month, orders.date) = 'April' THEN 4
        WHEN DATENAME(month, orders.date) = 'May' THEN 5
        WHEN DATENAME(month, orders.date) = 'June' THEN 6
        WHEN DATENAME(month, orders.date) = 'July' THEN 7
        WHEN DATENAME(month, orders.date) = 'August' THEN 8
        WHEN DATENAME(month, orders.date) = 'September' THEN 9
        WHEN DATENAME(month, orders.date) = 'October' THEN 10
        WHEN DATENAME(month, orders.date) = 'November' THEN 11
        WHEN DATENAME(month, orders.date) = 'December' THEN 12
        END, Sales ASC
```

After consulting with the respective stakeholders, they wanted to know the flavors which had sales around $1000 and asked which flavors should be taken off the menu and which can have promotions to increase the sales.

So, I found out that 2 flavors of pizzas had the least number of sales throughout the year. Brie Carre flavor had the average of just $965 of sales each month, so it should be taken off the menu because it is heavily contributing to the operational costs. Moreover, Green Garden flavor had the average of $1162 per month, so the marketing team can launch a promotional campaign if they want to continue with the sale of his product.

Pizza Flavor	January	February	March	April	May	June	July	August	September	October	November	December
brie_carre	827.75	1040.6	1087.9	756.8	1087.9	946	1040.6	1016.95	1158.84998	804.1	993.299984	827.749987
green_garden	1065.5	1226.5	1262.75	1145.75	1258.25	1230.5	1426.5	1150	1254	1181.5	836.5	918



Conclusion of the Analysis:

During this project, I analyzed the sales results of a fictious Pizza Palace company to not only extract and analyze the trends and patterns, but also suggested actions based on data for better profitability for the company.