

PIZZA SALES ANALYSIS 2022

Using SQL Data Retrieval and Power BI Visualization

Assumption

The client has provided two parts of the problem statement. In the first part, they have given the requirement to analyze key performance indicators (KPIs) related to pizza sales data in order to gain insights into their business performance. The specific KPIs to be calculated are:

1. Total Revenue: The sum of the total price of all pizza orders.
2. Average Order Value: The average amount spent per order, calculated by dividing the total revenue by the total number of orders.
3. Total Pizzas Sold: The sum of the quantities of all pizzas sold.
4. Total Orders: The total number of orders placed.
5. Average Pizzas Per Order: The average number of pizzas sold per order, calculated by dividing the total number of pizzas sold by the total number of orders.

Moving on to the second part of the requirement, the client wants visualization in the form of charts to explore different aspects of the data. The chart requirements are as follows:

1. Hourly Trend for Total Orders: Visualize the frequency of orders throughout the day using a stacked bar chart.
2. Weekly Trend for Orders: Analyze the fluctuation of orders on a weekly basis using a line chart.
3. Percentage of Sales by Pizza Category: Represent the distribution of sales across different pizza categories using a pie chart.
4. Percentage of Sales by Pizza Size: Visualize the proportion of sales for different pizza sizes, such as small, medium, and large.
5. Total Pizzas Sold by Pizza Category: Show the total number of pizzas sold for each pizza category using a funnel chart or an alternative chart like a butterfly chart.
6. Top Five Best-Selling Pizzas: Identify the top five performers based on revenue, total quantity, and total orders.
7. Bottom Five Worst-Selling Pizzas: Identify the bottom five performers based on revenue, total quantity, and total orders.

Data Sources - [Please download Here](#)

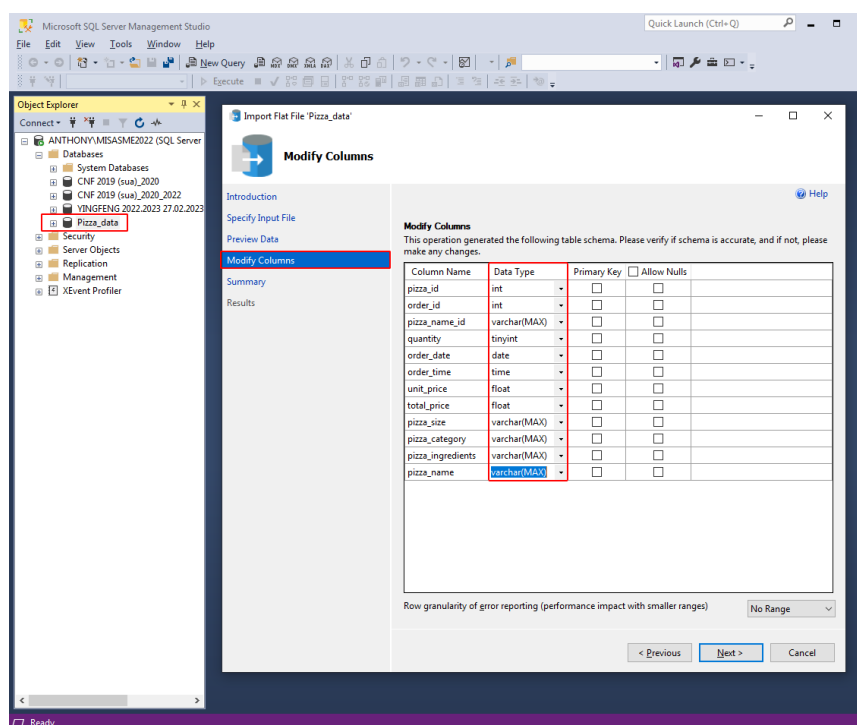
Processing

➤ Quá trình thực hiện công việc của tôi sẽ trải qua 5 bước như sau:

- Build a Database
- Analyze and Retrieve Data with SQL
- Visualize Data in Power BI

I. Build a Database

I use SQL Server 19 to access my server and create a database named "Pizza_data". Then, I import the processed data into my database and set up the data type as follows:



Here is my current data after import successfully:

SQLQuery1.sql - A...THONY\antho (53)* -> X

```
1 SELECT *
2 FROM pizza_sales;
```

100 %

	pizza_id	order_id	pizza_name_id	quantity	order_date	order_time	unit_price	total_price	pizza_size	pizza_category	pizza_ingredients
1	1	1	hawaiian_m	1	2022-01-01	11:38:36.0000000	13.25	13.25	M	Classic	Sliced Ham, Pineapple
2	2	2	classic_dlx_m	1	2022-01-01	11:57:40.0000000	16	16	M	Classic	Pepperoni, Mushrooms
3	3	2	five_cheese_l	1	2022-01-01	11:57:40.0000000	18.5	18.5	L	Veggie	Mozzarella Cheese, Pr
4	4	2	ital_supr_l	1	2022-01-01	11:57:40.0000000	20.75	20.75	L	Supreme	Calabrese Salami, Cap
5	5	2	mexicana_m	1	2022-01-01	11:57:40.0000000	16	16	M	Veggie	Tomatoes, Red Peppe
6	6	2	thai_ckn_l	1	2022-01-01	11:57:40.0000000	20.75	20.75	L	Chicken	Chicken, Pineapple, T
7	7	3	ital_supr_m	1	2022-01-01	12:12:28.0000000	16.5	16.5	M	Supreme	Calabrese Salami, Cap
8	8	3	prsc_argla_l	1	2022-01-01	12:12:28.0000000	20.75	20.75	L	Supreme	Prosciutto di San Dani
9	9	4	ital_supr_m	1	2022-01-01	12:16:31.0000000	16.5	16.5	M	Supreme	Calabrese Salami, Cap
10	10	5	ital_supr_m	1	2022-01-01	12:21:30.0000000	16.5	16.5	M	Supreme	Calabrese Salami, Cap
11	11	6	bbq_ckn_s	1	2022-01-01	12:29:36.0000000	12.75	12.75	S	Chicken	Barbecued Chicken, F
12	12	6	the_greek_s	1	2022-01-01	12:29:36.0000000	12	12	S	Classic	Kalamata Olives, Feta
13	13	7	spinach_sup...	1	2022-01-01	12:50:37.0000000	12.5	12.5	S	Supreme	Spinach, Red Onions,
14	14	8	spinach_sup...	1	2022-01-01	12:51:37.0000000	12.5	12.5	S	Supreme	Spinach, Red Onions,
15	15	9	classic_dlx_s	1	2022-01-01	12:52:01.0000000	12	12	S	Classic	Pepperoni, Mushrooms
16	16	9	green_garde...	1	2022-01-01	12:52:01.0000000	12	12	S	Veggie	Spinach, Mushrooms,
17	17	9	ital_cpcllo_l	1	2022-01-01	12:52:01.0000000	20.5	20.5	L	Classic	Capocollo, Red Peppe
18	18	9	ital_supr_l	1	2022-01-01	12:52:01.0000000	20.75	20.75	L	Supreme	Calabrese Salami, Cap

Query executed successfully.

Pizza_data 00:00:00 48,620 rows

II. Analyze and Retrieve Data with SQL

1. Total Revenue

To determine the total revenue, we execute a query that calculates the sum of the total prices of all pizza orders. The query utilizes the 'total_price' column from the 'pizza_sales' table. The output of the query is the total revenue, which represents the overall sales generated from pizza orders.

```
SELECT SUM(total_price) AS Total_Revenue  
FROM pizza_sales;
```

Results Messages	
Total_Revenue	
1	817860.05083847

2. Average Order Value

I need to calculate the average order value by dividing the total revenue by the total number of orders. To find the total number of orders, we need to count the distinct order IDs. Again, I will save the query and the output in the document.

```
SELECT SUM(total_price) / COUNT(DISTINCT order_id) AS Avg_order_Value,  
FROM pizza_sales;
```

Results Messages	
Avg_order_Value	
1	38.3072623343546

3. Total Pizzas Sold

To find the total number of pizzas sold, we execute a query that sums the quantities of all pizza orders. This query provides the total count of pizzas sold across all orders.

```
SELECT SUM(quantity) AS Total_pizza_sold  
FROM pizza_sales;
```

Results Messages	
Total_pizza_sold	
1	49574

4. Total Orders

To determine the total number of orders placed, we execute a query that counts the distinct order IDs. This query reveals the overall count of unique orders made.

```
SELECT COUNT(DISTINCT order_id) AS Total_Orders
FROM pizza_sales;
```

Results	
Messages	
Total_Orders	
1	21350

5. Average Pizzas Per Order

```
SELECT CAST(CAST(SUM(quantity) AS DECIMAL(10,2)) /
COUNT(DISTINCT order_id) AS DECIMAL(10,2))
AS Avg_Pizzas_per_order
FROM pizza_sales
```

Results	
Messages	
Avg_Pizzas_per_order	
1	2.32

It is important to save all queries and their respective outputs for future reference and to share with clients and stakeholders.

SQLQuery1.sql - A...THONY(antho (53)) - X

```
1 SELECT *
2 FROM pizza_sales
3
4 SELECT
5 ---1. Total Revenue---
6 SUM(total_price) AS Total_Revenue,
7 ---2. Average Order Value---
8 CAST(SUM(total_price) / COUNT(DISTINCT order_id) AS DECIMAL(10,2)) AS Avg_Order_Value,
9 ---3. Total Pizza Sold---
10 SUM(quantity) AS Total_Pizza_Sold,
11 ---4. Total Order---
12 COUNT(DISTINCT order_id) AS Total_orders,
13 ---5. Average Pizzas Per Order---
14 CAST(CAST(SUM(quantity) AS DECIMAL(10,2)) / COUNT(DISTINCT order_id) AS DECIMAL(10,2)) AS Avg_Pizzas_Order
15 FROM pizza_sales;
```

100 %

Results Messages

	pizza_id	order_id	pizza_name_id	quantity	order_date	order_time	unit_price	total_price	pizza_size	pizza_category	pizza_ingredients
1	1	1	hawaiian_m	1	2022-01-01	11:38:36.00000000	13.25	13.25	M	Classic	Sliced Ham, Pineapple, Mozzarella Cheese
2	2	2	classic_dlx_m	1	2022-01-01	11:57:40.00000000	16	16	M	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppers, B...
3	3	2	five_cheese_l	1	2022-01-01	11:57:40.00000000	18.5	18.5	L	Veggie	Mozzarella Cheese, Provolone Cheese, Smoked Gouc
4	4	2	ital_supr_l	1	2022-01-01	11:57:40.00000000	20.75	20.75	L	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions,
5	5	2	mexicana_m	1	2022-01-01	11:57:40.00000000	16	16	M	Veggie	Tomatoes, Red Peppers, Jalapeno Peppers, Red Oni...
6	6	2	thai_chn_l	1	2022-01-01	11:57:40.00000000	20.75	20.75	L	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, Thai Sv...
7	7	3	ital_supr_m	1	2022-01-01	12:12:28.00000000	16.5	16.5	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions,
8	8	3	prsc_argla_l	1	2022-01-01	12:12:28.00000000	20.75	20.75	L	Supreme	Prosciutto di San Daniele, Arugula, Mozzarella Cheese
9	9	4	ital_supr_m	1	2022-01-01	12:16:31.00000000	16.5	16.5	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions,
10	10	5	ital_supr_m	1	2022-01-01	12:21:30.00000000	16.5	16.5	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions,
11	11	6	bbq_chn_s	1	2022-01-01	12:29:36.00000000	12.75	12.75	S	Chicken	Barbecued Chicken, Red Peppers, Green Peppers, Tr...
12	12	6	the creek s	1	2022-01-01	12:29:36.00000000	12	12	S	Classic	Kalamata Olives, Feta Cheese, Tomatoes, Garlic, Red...

	Total_Revenue	Avg_Order_Value	Total_Pizza_Sold	Total_orders	Avg_Pizzas_Order
1	817860.05083847	38.33	49574	21338	2.32

Query executed successfully.

Pizza_data 00:00:00 48,621 rows

Moving on to the second part of the request, before creating a presentation chart on Power BI, I need to query the data for comparison and verify the results with the generated chart. To query the time-based fluctuations, I utilized the DATEPART function, and for querying the top 5 best-selling/worst-selling products, I used the TOP clause. Then, I used the ORDER BY clause to sort the results. The query and its results are as follows:

1. Hourly Trend for Total Orders

SQLQuery2.sql - A...THONY\antho (53))

```

1  ---1. Hourly Trend for Total pizzas Sold---
2  SELECT DATEPART(HOUR, order_time) AS order_hour,
3         SUM(quantity) as Total_pizzas_sold
4  FROM pizza_sales
5  GROUP BY DATEPART(HOUR, order_time)
6  ORDER BY DATEPART(HOUR, order_time) ASC;

```

	order_hour	Total_pizzas_sold
1	9	4
2	10	18
3	11	2728
4	12	6776
5	13	6413
6	14	3613
7	15	3216
8	16	4239
9	17	5211
10	18	5417
11	19	4406
12	20	3534
13	21	2545
14	22	1386
15	23	68

2. Weekly Trend for Orders

SQLQuery2.sql - A...THONY\antho (53))

```

1  ---2. Weekly Trend for Total Orders---
2  SELECT DATEPART(ISO_WEEK, order_date) AS week_number,
3         YEAR(order_date) AS 'Year',
4         COUNT(DISTINCT order_id) AS Total_orders
5  FROM pizza_sales
6  GROUP BY DATEPART(ISO_WEEK, order_date), YEAR(order_date)
7  ORDER BY DATEPART(ISO_WEEK, order_date), YEAR(order_date) ASC;

```

	week_number	Year	Total_orders
1	1	2022	428
2	2	2022	411
3	3	2022	414
4	4	2022	402
5	5	2022	448
6	6	2022	425
7	7	2022	412
8	8	2022	394
9	9	2022	408
10	10	2022	419
11	11	2022	428
12	12	2022	406
13	13	2022	431
14	14	2022	408
15	15	2022	418
16	16	2022	431
17	17	2022	424
18	18	2022	405

3. Percentage sales by Pizza category

SQLQuery2.sql - A...THONY\antho (53))* -p X

```

1  ---2. Weekly Trend for Total Orders---
2  SELECT DATEPART(ISO_WEEK, order_date) AS week_number,
3         YEAR(order_date) AS 'Year',
4         COUNT(DISTINCT order_id) AS Total_orders
5  FROM pizza_sales
6  GROUP BY DATEPART(ISO_WEEK, order_date), YEAR(order_date)
7  ORDER BY DATEPART(ISO_WEEK, order_date), YEAR(order_date) ASC;

```

100 %

Results Messages

	week_number	Year	Total_orders
1	1	2022	428
2	2	2022	411
3	3	2022	414
4	4	2022	402
5	5	2022	448
6	6	2022	425
7	7	2022	412
8	8	2022	394
9	9	2022	408
10	10	2022	419
11	11	2022	428
12	12	2022	406
13	13	2022	431
14	14	2022	408
15	15	2022	418
16	16	2022	431
17	17	2022	424
18	18	2022	405

4. Percentage of Sales by Pizza Size

SQLQuery2.sql - A...THONY\antho (53))* -p X

```

1  ---4. Percentage Sales by Pizza Size---
2  SELECT pizza_size,
3         CAST(SUM(total_price) AS DECIMAL(10,2)),
4         CAST(SUM(total_price) * 100 /
5         (SELECT SUM(total_price)
6         FROM pizza_sales AS Total_Sales) AS DECIMAL(10,2)) AS PCT
7  FROM pizza_sales
8  GROUP BY pizza_size
9  ORDER BY PCT DESC;
10

```

100 %

Results Messages

	pizza_size	(No column name)	PCT
1	L	375318.70	45.89
2	M	249382.25	30.49
3	S	178076.50	21.77
4	XL	14076.00	1.72
5	XXL	1006.60	0.12

Query executed successfully. ANTHONY\MISASME2022 (12.0 RTM) ANTHONY\antho (53) Pizza_data 00:00:00 5 rows

5. Total Pizzas Sold by Pizza Category

The screenshot shows a SQL query window with the following text:

```
SQLQuery1.sql - A...THONY\antho (55))* X
1  ---5. Total Pizzas Sold by Pizza Category---
2  SELECT  pizza_category,
3          SUM(quantity) AS Total_Quantity_Sold
4  FROM    pizza_sales
5  GROUP BY pizza_category
6  ORDER BY Total_Quantity_Sold DESC;
7
```

Below the query, the 'Results' tab is active, displaying a table with the following data:

	pizza_category	Total_Quantity_Sold
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

6. Top 5 Best-Selling Pizzas

- *Top 5 Pizzas by Revenue*

The screenshot shows a SQL query window with the following text:

```
SQLQuery1.sql - A...THONY\antho (55))* X
1  --Top 5 Pizzas by Revenue
2  SELECT  TOP 5 pizza_name,
3          SUM(total_price) AS Total_Revenue
4  FROM    pizza_sales
5  GROUP BY pizza_name
6  ORDER BY Total_Revenue DESC;
```

Below the query, the 'Results' tab is active, displaying a table with the following data:

	pizza_name	Total_Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5
4	The Classic Deluxe Pizza	38180.5
5	The Spicy Italian Pizza	34831.25

- *Top 5 Pizzas by Quantity*

The screenshot shows a SQL query window with the following text:

```
SQLQuery1.sql - A...THONY\antho (55))* X
1  --Top 5 Pizzas by Quantity
2  SELECT  TOP 5 pizza_name,
3          SUM(quantity) AS Total_Quantity
4  FROM    pizza_sales
5  GROUP BY pizza_name
6  ORDER BY Total_Quantity DESC;
```

Below the query, the 'Results' tab is active, displaying a table with the following data:

	pizza_name	Total_Quantity
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

- *Top 5 Pizzas by total orders*

SQLQuery1.sql - A...THONY\antho (55))*

```

1  ---Top 5 Pizzas by total orders
2  SELECT TOP 5 pizza_name,
3      COUNT(DISTINCT order_id) AS Total_Orders
4  FROM pizza_sales
5  GROUP BY pizza_name
6  ORDER BY Total_Orders DESC;
7

```

100 %

Results Messages

	pizza_name	Total_Orders
1	The Classic Deluxe Pizza	2329
2	The Hawaiian Pizza	2280
3	The Pepperoni Pizza	2278
4	The Barbecue Chicken Pizza	2273
5	The Thai Chicken Pizza	2225

7. Bottom 5 Best-Selling Pizzas

- *Bottom 5 Pizzas by Revenue*

SQLQuery1.sql - A...THONY\antho (55))*

```

1  ---Bottom 5 Pizzas by Revenue
2  SELECT TOP 5 pizza_name,
3      SUM(total_price) AS Total_Revenue
4  FROM pizza_sales
5  GROUP BY pizza_name
6  ORDER BY Total_Revenue ASC;
7

```

100 %

Results Messages

	pizza_name	Total_Revenue
1	The Brie Carré Pizza	11588.4998130798
2	The Green Garden Pizza	13955.75
3	The Spinach Supreme Pizza	15277.75
4	The Mediterranean Pizza	15360.5
5	The Spinach Pesto Pizza	15596

- *Bottom 5 Pizzas by Quantity*

SQLQuery1.sql - A...THONY\antho (55))*

```

1  ---Bottom 5 Pizzas by Quantity
2  SELECT TOP 5 pizza_name,
3      SUM(quantity) AS Total_Quantity
4  FROM pizza_sales
5  GROUP BY pizza_name
6  ORDER BY Total_Quantity ASC;
7

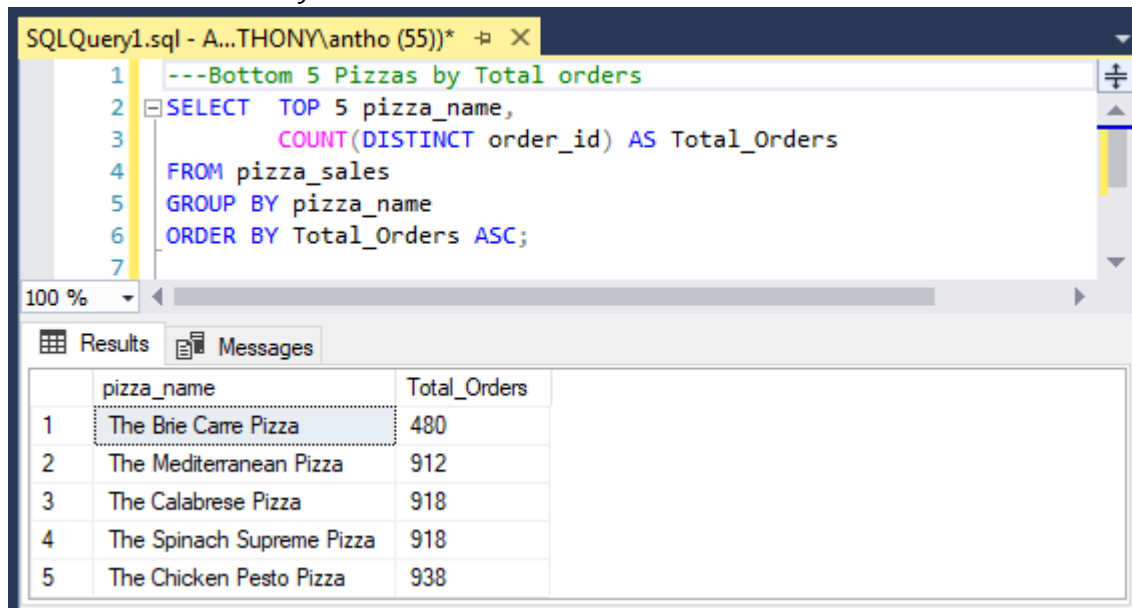
```

100 %

Results Messages

	pizza_name	Total_Quantity
1	The Brie Carré Pizza	490
2	The Mediterranean Pizza	934
3	The Calabrese Pizza	937
4	The Spinach Supreme Pizza	950
5	The Soppressata Pizza	961

- *Bottom 5 Pizzas by total orders*



The screenshot shows a SQL query window with the following text:

```

1  ---Bottom 5 Pizzas by Total orders
2  SELECT TOP 5 pizza_name,
3         COUNT(DISTINCT order_id) AS Total_Orders
4  FROM pizza_sales
5  GROUP BY pizza_name
6  ORDER BY Total_Orders ASC;
7

```

Below the query, the 'Results' tab is active, displaying a table with the following data:

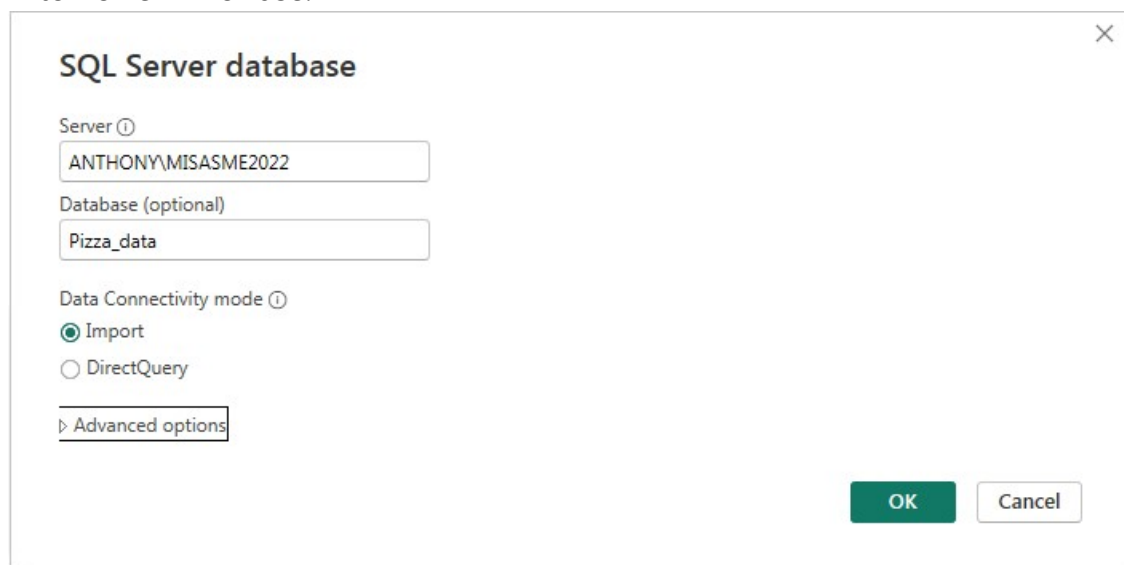
	pizza_name	Total_Orders
1	The Brie Carré Pizza	480
2	The Mediterranean Pizza	912
3	The Calabrese Pizza	918
4	The Spinach Supreme Pizza	918
5	The Chicken Pesto Pizza	938

After querying the necessary data, I proceed to visualize data

III. Visualize Data in Power BI

The main steps I took to visualize the data:

- I connected to the data source in SQL Server 19. I loaded the data from these sources into Power BI for use.



The screenshot shows the 'SQL Server database' connection dialog in Power BI. The fields are filled as follows:

- Server: ANTHONY\MISASME2022
- Database (optional): Pizza_data
- Data Connectivity mode: ☒ Import
- ☐ DirectQuery
- Advanced options: expanded (indicated by a right-pointing arrow)

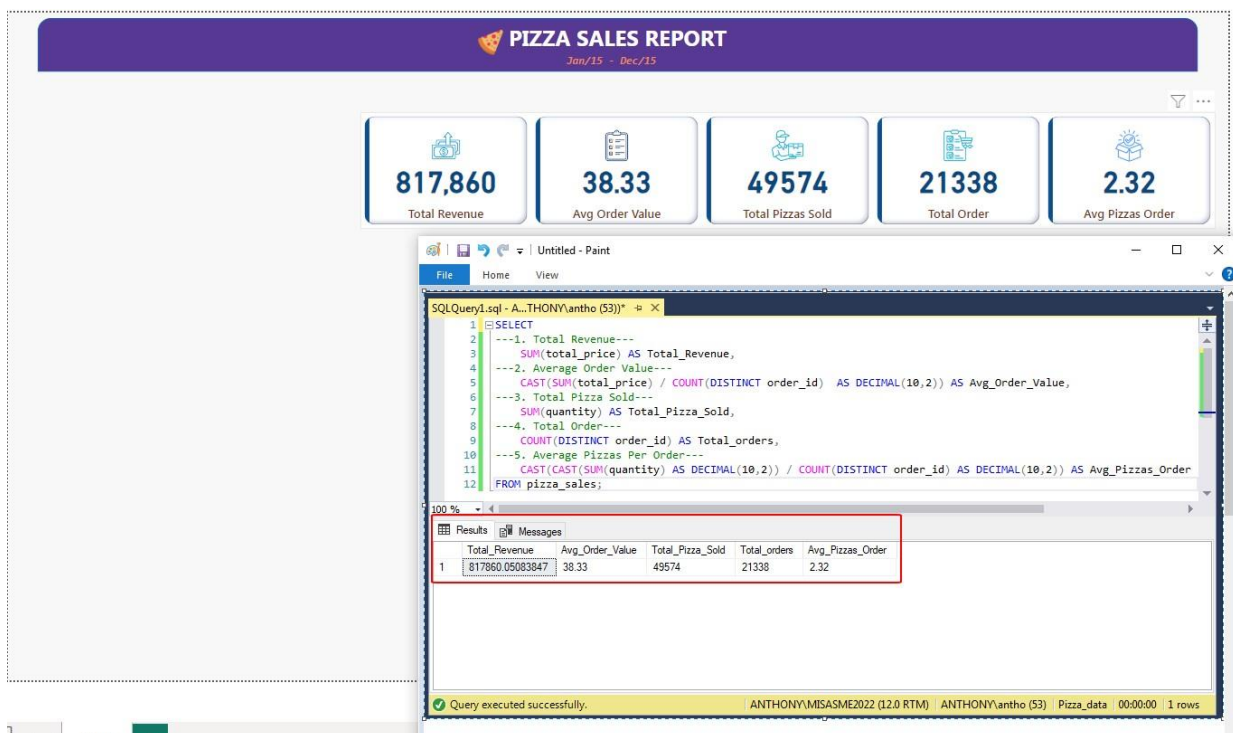
At the bottom right, there are 'OK' and 'Cancel' buttons.

- I previewed the data tables, checked the relationships between them, and identified important data fields for visualization.

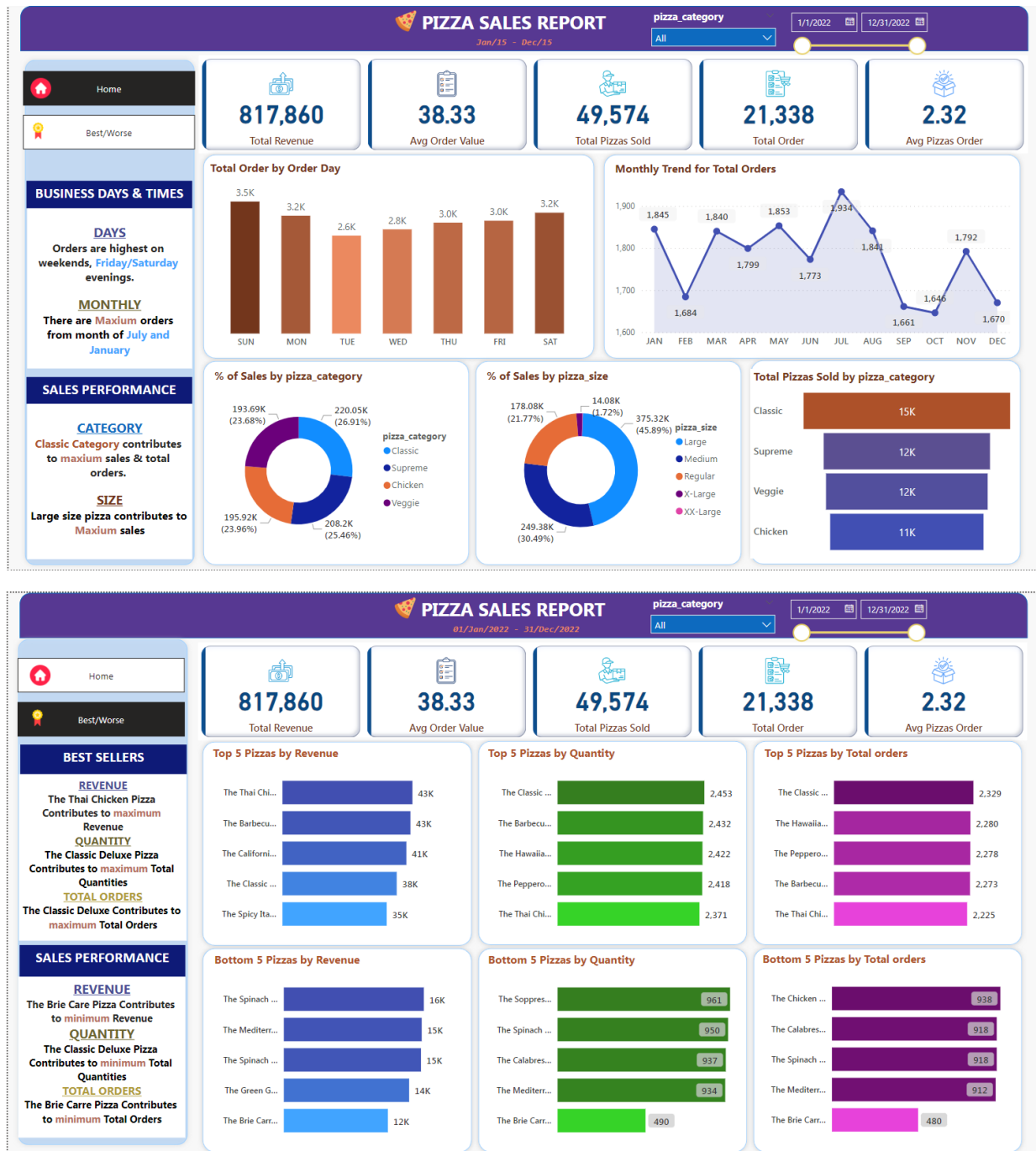
Table: pizza_sales (48,620 rows)

pizza_id	order_id	pizza_name_id	quantity	order_date	order_time	unit_price
6	2	thai_chn_l	1	Saturday, January 1, 2022	11:57:40 AM	20
42	16	thai_chn_l	1	Saturday, January 1, 2022	1:34:07 PM	20
68	25	thai_chn_l	1	Saturday, January 1, 2022	2:44:44 PM	20
76	30	thai_chn_l	1	Saturday, January 1, 2022	3:41:25 PM	20
89	35	thai_chn_l	1	Saturday, January 1, 2022	4:32:04 PM	20
131	54	thai_chn_l	1	Saturday, January 1, 2022	7:01:45 PM	20
164	71	thai_chn_l	1	Sunday, January 2, 2022	11:40:50 AM	20
188	79	thai_chn_l	1	Sunday, January 2, 2022	12:29:11 PM	20
191	81	thai_chn_l	1	Sunday, January 2, 2022	12:40:01 PM	20
209	89	thai_chn_l	1	Sunday, January 2, 2022	2:30:19 PM	20
340	143	thai_chn_l	1	Monday, January 3, 2022	1:41:36 PM	20
375	156	thai_chn_l	1	Monday, January 3, 2022	3:42:19 PM	20
408	172	thai_chn_l	1	Monday, January 3, 2022	5:29:24 PM	20
420	177	thai_chn_l	1	Monday, January 3, 2022	5:54:36 PM	20
439	186	thai_chn_l	1	Monday, January 3, 2022	7:36:00 PM	20
526	224	thai_chn_l	1	Tuesday, January 4, 2022	4:39:58 PM	20
558	242	thai_chn_l	1	Tuesday, January 4, 2022	8:15:29 PM	20

- In this case, I used card charts, bar charts, column charts, donut charts, and funnel charts to visualize the data. I customized the properties and formatting of the charts to display the data clearly and understandably. Afterward, I cross-validated the data with previous query results to ensure accuracy before sending it back to the customer.



- I arranged the charts and tables on report pages, providing descriptions and other visual elements such as images or supplementary charts.
- Additionally, I create interactions between the charts and tables to allow users to interact with and explore the data. Users can filter the data by selecting values on one chart and see its impact on other charts and tables. And there are my results:



Final visualize: -

Visualize data - Pizza sales.pbix

Finals

In the Pizza Sales Analysis 2022 project, we successfully utilized SQL data retrieval and Power BI visualization techniques to analyze key performance indicators (KPIs) and provide insightful visualizations for our client.

For the first part of the requirement, we calculated important KPIs related to pizza sales, including Total Revenue, Average Order Value, Total Pizzas Sold, Total Orders, and Average Pizzas Per Order. These metrics provide valuable insights into the client's business performance, allowing them to understand their revenue, customer spending patterns, and overall sales volume.

In the second part of the requirement, we created various charts to visually explore different aspects of the data. We designed a stacked bar chart to showcase the hourly trend of total orders, enabling the client to identify peak and off-peak periods. Additionally, we utilized a line chart to analyze the weekly fluctuation of orders, providing valuable insights into sales patterns over time.

To represent the distribution of sales, we utilized a pie chart to display the percentage of sales by pizza category. This visualization allows the client to understand the popularity and demand for each pizza category. We also visualized the proportion of sales for different pizza sizes using an appropriate chart, enabling the client to identify size preferences among customers.

To analyze the performance of individual pizza categories, we employed a funnel chart or an alternative chart like a butterfly chart to showcase the total number of pizzas sold for each category. This visualization helps the client identify the most successful categories and make data-driven decisions regarding their product offerings.

Furthermore, we identified the top five best-selling pizzas based on revenue, total quantity, and total orders. This information assists the client in understanding their top performers and potentially capitalizing on their success. Similarly, we identified the bottom five worst-selling pizzas, providing insights into underperforming products that may require adjustments or further analysis.

Overall, our utilization of SQL data retrieval and Power BI visualization techniques in the Pizza Sales Analysis 2022 project has enabled our client to gain valuable insights into their business performance and make informed decisions based on the analyzed KPIs and visualizations.