

# SQL ANALYSIS OF PIZZA SALES

---



Victoria Chiamaka Nwogo

# ANALYSIS OVERVIEW

---

- This report involves a comprehensive analysis of pizza sales data using SQL. The goal is to extract valuable and actionable insights to inform business decision also answer various business questions based on the dataset. This report provides a detailed account of the objectives, business problems, solutions, findings, and conclusions

# OBJECTIVES

---

- To conduct a comprehensive analysis of pizza sales data using SQL and present the findings in a clear, visually appealing, and user-friendly format using PowerPoint. The analysis will enable stakeholders to:
  1. Track sales performance: Understand sales trends and metrics.
  2. Identify popular pizza flavors and toppings: Discover top-selling pizzas and ingredients.
  3. Analyze customer preferences: Gain insights into customer behavior and preferences.
  4. Identify opportunities for growth and improvement: Pinpoint areas for business expansion and optimization.

# PIZZA SALES DATASET OVERVIEW

The screenshot displays the Microsoft SQL Server Enterprise Edition interface. The top menu bar includes File, Edit, View, Git, Project, Table Designer, Tools, Extensions, Window, and Help. A search bar and a 'Solution1' tab are also visible. The main toolbar contains icons for various database operations, including 'New Query', 'Execute', and 'Table View'. The 'Object Explorer' on the left shows the database hierarchy for 'VICENCY\SQLEXPRESS (SQL Se)'. The 'Databases' folder is expanded, showing 'System Databases', 'Database Snapshots', 'CENCY', 'music\_project', 'netflix', and 'PIZZA SALES'. The 'PIZZA SALES' database is selected, and its 'Tables' folder is expanded, showing 'System Tables', 'FileTables', 'External Tables', 'Graph Tables', and 'dbo.pizza\_sales'. The 'Columns' folder under 'dbo.pizza\_sales' is expanded, showing the following columns:

Column Name
pizza_id
order_id
pizza_name_id
quantity
order_date
order_time
unit_price
total_price
pizza_size
pizza_category
pizza_ingredients
pizza_name



# PIZZA TABLE OVERVIEW

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the database structure for 'VICENCY\SQLEXPRESS (SQL Server 16.0)'. The 'PIZZA SALES' table is highlighted under the 'Tables' folder. The main window shows a SQL query executed successfully, displaying the results of the 'select \* from pizza\_sales;' command. The query results are shown in a table with 12 columns: pizza\_id, order\_id, pizza\_name\_id, quantity, order\_date, order\_time, unit\_price, total\_price, pizza\_size, pizza\_category, pizza\_ingredients, and pizza\_name. The status bar at the bottom indicates 'Query executed successfully.' and '48,620 rows'.

	pizza_id	order_id	pizza_name_id	quantity	order_date	order_time	unit_price	total_price	pizza_size	pizza_category	pizza_ingredients	pizza_name
1	1	1	hawaiian_m	1	2015-01-01	11:38:36.0000000	13.25	13.25	M	Classic	Sliced Ham, Pineapple, Mozzarella Cheese	The F
2	2	2	classic_dlx_m	1	2015-01-01	11:57:40.0000000	16	16	M	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppers, Ba...	The C
3	3	2	five_cheese_l	1	2015-01-01	11:57:40.0000000	18.5	18.5	L	Veggie	Mozzarella Cheese, Provolone Cheese, Smoked Goud...	The F
4	4	2	ital_supr_l	1	2015-01-01	11:57:40.0000000	20.75	20.75	L	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions, ...	The I
5	5	2	mexicana_m	1	2015-01-01	11:57:40.0000000	16	16	M	Veggie	Tomatoes, Red Peppers, Jalapeno Peppers, Red Onio...	The M
6	6	2	thai_ckn_l	1	2015-01-01	11:57:40.0000000	20.75	20.75	L	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, Thai Sw...	The T
7	7	3	ital_supr_m	1	2015-01-01	12:12:28.0000000	16.5	16.5	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions, ...	The I
8	8	3	prsc_argla_l	1	2015-01-01	12:12:28.0000000	20.75	20.75	L	Supreme	Prosciutto di San Daniele, Arugula, Mozzarella Cheese	The F
9	9	4	ital_supr_m	1	2015-01-01	12:16:31.0000000	16.5	16.5	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions, ...	The I
10	10	5	ital_supr_m	1	2015-01-01	12:21:30.0000000	16.5	16.5	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions, ...	The I
11	11	6	bbq_ckn_s	1	2015-01-01	12:29:36.0000000	12.75	12.75	S	Chicken	Barbecued Chicken, Red Peppers, Green Peppers, To...	The F

# TOTAL REVENUE & TOTAL ORDERS

- TOTAL REVENUE: THE BUSINESS GENERATED N 817,860.05 IN TOTAL SALES DURING THE PERIOD.
- SHOWING STRONG REVENUE PERFORMANCE
- TOTAL ORDERS : A TOTAL OF 49574 ORDERS WERE PLACED, HIGHLIGHTING CUSTOMER DEMAND AND TRANSACTION VOLUME.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the 'PIZZA SALES' database selected. The right pane shows a SQL query window with the following query:

```
--- What is the Total Revenue generated? ---  
select sum(total_price)  
AS Total_Revenue  
from pizza_sales;
```

The query results are displayed in a table with one row:

Total_Revenue
817860.05083847

The status bar at the bottom indicates 'Query executed successfully.' and 'VICENCY\SQLEXPRESS (16.0 RTM) VICENCY\user (54) PIZZA SALES 00:00:00 1 rows'.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the 'PIZZA SALES' database selected. The right pane shows a SQL query window with the following query:

```
--- What are the Total number Orders placed? ---  
select count(order_id)  
AS Total_orders  
from pizza_sales;
```

The query results are displayed in a table with one row:

Total_orders
48620

The status bar at the bottom indicates 'Query executed successfully.' and 'VICENCY\SQLEXPRESS (16.0 RTM) VICENCY\user (54) PIZZA SALES 00:00:00 1 rows'.

# 5 TOP SELLING PIZZAS

- TOP SELLING PIZZA: CLASSIC DELUXE PIZZA IS THE MOST POPULAR PIZZA WITH THE TOTAL NUMBER OF 2453 SOLD PIZZA

The screenshot displays the Microsoft SQL Server Enterprise Edition interface. The 'Object Explorer' on the left shows the database structure for 'VICENCY\SQLEXPRESS (SQL Serve)', with the 'PIZZA SALES' database selected. The main window shows a SQL query titled 'SQLQuery1 pi...Y\user (54))\*' with the following code:

```
--- . Top 5 Best Sellers by Total Pizzas Sold ---  
SELECT Top 5 pizza_name,  
SUM(quantity) AS Total_Pizza_Sold  
FROM pizza_sales  
GROUP BY pizza_name  
ORDER BY Total_Pizza_Sold DESC;
```

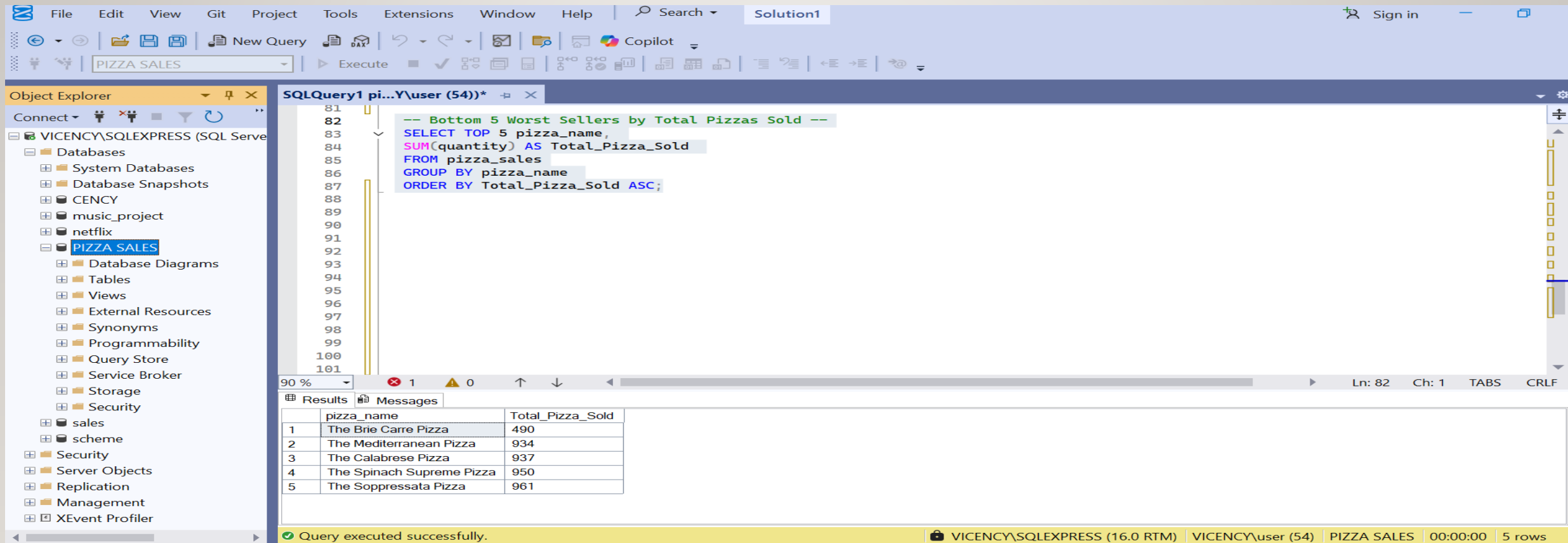
The 'Results' pane at the bottom shows the output of the query, which is a table with 5 rows and 2 columns: 'pizza\_name' and 'Total\_Pizza\_Sold'.

	pizza_name	Total_Pizza_Sold
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

The status bar at the bottom indicates 'Query executed successfully.' and provides details about the connection: 'VICENCY\SQLEXPRESS (16.0 RTM) | VICENCY\user (54) | PIZZA SALES | 00:00:00 | 5 rows'.

# 5 WORST SELLING PIZZAS

- WORST SELLING PIZZA: THE BRIE CARRE PIZZA IS LEAST POPULAR PIZZA WITH THE TOTAL NUMBER OF 490 SOLD PIZZA



The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'VICENCY\SQLEXPRESS (SQL Server)'. The 'PIZZA SALES' database is selected, showing its tables, views, and external resources. The main window displays a SQL query titled 'SQLQuery1 pi...Y\user (54))\*'. The query is as follows:

```
-- Bottom 5 Worst Sellers by Total Pizzas Sold --
SELECT TOP 5 pizza_name,
SUM(quantity) AS Total_Pizza_Sold
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Pizza_Sold ASC;
```

The query results are displayed in a table with the following data:

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

The status bar at the bottom indicates that the query was executed successfully, returning 5 rows in 00:00:00 seconds.



# SALES BY CATEGORY & SALES BY SIZE

- SALES BY CATEGORY: CLASSIC PIZZA GENERATED THE HIGHEST REVENUE CONTRIBUTING 26.91% OF THE TOTAL REVENUE, WHILE VEGGIE PIZZA GENERATED THE LOWEST REVENUE WITH 23.68% CONTRIBUTION TO THE TOTAL REVENUE.

- SALES BY SIZE: L SIZE PIZZA GENERATED THE HIGHEST REVENUE WITH 45.89% CONTRIBUTION TO THE TOTAL REVENUE WHILE XXL SIZE PIZZA GENERATED THE LOWEST REVENUE CONTRIBUTING 0.12% OF THE TOTAL REVENUE

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the 'PIZZA SALES' database selected. The central pane shows a SQL query titled 'SQLQuery1 pi...Y(user (54))\*' with the following text:

```
--- What is the total percentage of Sales by Pizza Category? ---  
SELECT pizza_category,  
CAST(SUM(total_price) AS DECIMAL(10, 2)) as total_revenue,  
CAST(SUM(total_price) * 100 / (SELECT SUM(total_price) from pizza_sales) AS DECIMAL(10, 2)) AS PCT  
FROM pizza_sales  
GROUP BY pizza_category;
```

The bottom pane shows the 'Results' tab with a table containing 4 rows:

	pizza_category	total_revenue	PCT
1	Chicken	195919.50	23.96
2	Supreme	208197.00	25.46
3	Classic	220053.10	26.91
4	Veggie	193690.45	23.68

The status bar at the bottom indicates 'Query executed successfully.' and 'VICENCY\SQLEXPRESS (16.0 RTM) VICENCY\user (54) PIZZA SALES 00:00:00 4 rows'.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the 'PIZZA SALES' database selected. The central pane shows a SQL query titled 'SQLQuery1 pi...Y(user (54))\*' with the following text:

```
--- . What is the total percentage of Sales by Pizza Size? ---  
SELECT pizza_size,  
CAST(SUM(total_price) AS DECIMAL(10, 2)) as total_revenue,  
CAST(SUM(total_price) * 100 / (SELECT SUM(total_price) from pizza_sales) AS DECIMAL(10, 2)) AS PCT FROM pizza_sales  
GROUP BY pizza_size  
ORDER BY pizza_size;
```

The bottom pane shows the 'Results' tab with a table containing 5 rows:

	pizza_size	total_revenue	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

The status bar at the bottom indicates 'Query executed successfully.' and 'VICENCY\SQLEXPRESS (16.0 RTM) VICENCY\user (54) PIZZA SALES 00:00:00 5 rows'.

# TIME OF THE DAY SALES & WEEK DAYS SALES

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the 'PIZZA SALES' database selected. The central pane shows a SQL query titled 'SQLQuery1 pi...Y(user (54))\*' with the following code:

```
-- Hourly Trend for Orders --  
SELECT DATEPART(HOUR, order_time) as order_hours,  
COUNT(DISTINCT order_id) as total_orders  
FROM pizza_sales  
GROUP BY DATEPART(HOUR, order_time)  
ORDER BY DATEPART(HOUR, order_time);
```

The bottom pane displays the results of the query, showing a table with two columns: 'order\_hours' and 'total\_orders'. The data is as follows:

order_hours	total_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

The status bar at the bottom indicates 'Query executed successfully. VICENCY\SQLEXPRESS (16.0 RTM) VICENCY\user (54) PIZZA SALES 00:00:00 15 rows'.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the 'PIZZA SALES' database selected. The central pane shows a SQL query titled 'SQLQuery1 pi...Y(user (54))\*' with the following code:

```
-- Daily Trend for Total Orders --  
SELECT DATENAME(DW, order_date) AS order_day,  
COUNT(DISTINCT order_id) AS total_orders  
FROM pizza_sales  
GROUP BY DATENAME(DW, order_date);
```

The bottom pane displays the results of the query, showing a table with two columns: 'order\_day' and 'total\_orders'. The data is as follows:

order_day	total_orders
Saturday	3158
Wednesday	3024
Monday	2794
Sunday	2624
Friday	3538
Thursday	3239
Tuesday	2973

The status bar at the bottom indicates 'Query executed successfully. VICENCY\SQLEXPRESS (16.0 RTM) VICENCY\user (54) PIZZA SALES 00:00:00 7 rows'.

# AVERAGE ORDER VALUE & AVERAGE NUMBER OF PIZZAS PER ORDERS

This screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the 'PIZZA SALES' database selected. The central pane shows a SQL query window with the following text:

```
10  
11 --- What is the Average Order Value? ---  
12 select sum(total_price) / count(distinct order_id)  
13 AS Average_order_value  
14 from pizza_sales;  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25
```

The bottom pane shows the query results in a table with one row:

Average_order_value
38.3072623343546

The status bar at the bottom indicates: "Query executed successfully. VICENCY\SQLEXPRESS (16.0 RTM) VICENCY\user (54) PIZZA SALES 00:00:00 1 rows".

This screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the 'PIZZA SALES' database selected. The central pane shows a SQL query window with the following text:

```
26  
27 --- What is the Average number of Pizzas Per Order? --  
28 SELECT CAST(CAST(SUM(quantity)  
29 AS DECIMAL(10,2)) / CAST(COUNT(DISTINCT order_id)  
30 AS DECIMAL(10,2)))  
31 AS DECIMAL(10,2))  
32 AS Avg_Pizzas_per_order  
33 FROM pizza_sales;  
34  
35  
36  
37  
38  
39  
40  
41
```

The bottom pane shows the query results in a table with one row:

Avg_Pizzas_per_order
2.32

The status bar at the bottom indicates: "Query executed successfully. VICENCY\SQLEXPRESS (16.0 RTM) VICENCY\user (54) PIZZA SALES 00:00:00 1 rows".

# RECOMMENDATION

---

- Promote Classic Deluxe Pizza through bundles and special offers to boost revenue.
- Reevaluate or remove Brie Carre Pizza due to consistently low sales.
- Focus on L-size pizzas for upselling; consider phasing out XXL size due to low demand.
- Target Thursday, Friday, and Saturday with exclusive deals to maximize peak-day performance.
- Align inventory and staffing with top-selling products and high-order days.
- Continuously monitor product performance to refine the menu and reduce waste.



# CONCLUSION

---

The analysis provides valuable insights into pizza sales trends, customer preferences, and opportunities for growth. Key takeaways include:

- Menu optimization: Consider emphasizing top-selling pizzas and categories.
- Targeted marketing: Focus on peak days and hours to maximize sales.
- Upselling and cross-selling: Leverage average order value and pizzas per order to increase revenue.
- Inventory management: Ensure adequate stock of popular pizza ingredients and sizes.
- Menu engineering: Consider adjusting menu offerings based on sales data and customer preferences.

By implementing these strategies, the pizza business can potentially increase revenue, improve customer satisfaction, and gain a competitive edge in the market.

