

# **Enhancing Power BI Performance: Optimizing Pizza Sales Dashboards for Improved User Experience**

**1. Introduction:** The optimization of Power BI dashboards presents a critical challenge for organizations looking to extract relevant insights quickly and easily in the fast-changing field of data analytics and visualization. The need for high-performance, user-friendly dashboards has never been greater due to the exponential data volume and complexity rise.

With an emphasis on improving efficiency and user experience, this project sets out to solve the urgent need for Power BI dashboard optimization. The objective is to fully realize the potential of data visualization within the framework of a Pizza sales dataset by utilizing the strong analytical skills of Python and SQL for database query optimization.

The goal of taking Power BI dashboards to new levels of effectiveness, responsiveness, and interactivity is at the core of this project. Through the process of locating and eliminating performance bottlenecks, this project aims to expedite data exploration while simultaneously providing users with simplified insights into key performance indicators related to pizza sales, such as revenue, top-selling pizzas, orders, quantity, and categories.

This project aims to establish a new benchmark for Power BI dashboard optimization, promoting informed decision-making and fostering business success in the fast-paced pizza sales industry. It does this by firmly adhering to excellence in data visualization and analytics.

## **2. Data Acquisition:**

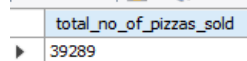
The project refers to the process of gathering and collecting the Pizza dataset from various sources, ensuring its cleanliness, completeness, and relevance. This involves identifying the required data elements such as sold pizza, revenue, top-selling pizzas, pizza orders, quantity, and category, extracting the data from appropriate sources, cleaning and transforming it as necessary, and integrating it into Power BI for analysis and visualization. Additionally, data acquisition may involve leveraging Python for in-depth analysis and SQL for optimizing database queries to enhance the performance of the dashboards. The goal of data acquisition is to provide a solid foundation for effective data exploration and analysis, thereby facilitating the optimization of Power BI dashboards for improved speed, responsiveness, and user experience.

## **3. SQL Query Optimization:**

- **Total no of pizzas sold:**

The SQL query calculates the sum of the quantity column from the pizza table, representing the total number of pizzas sold. The result is labeled as "total\_no\_of\_pizzas\_sold".

```
select sum(quantity) as total_no_of_pizzas_sold from pizza;
```

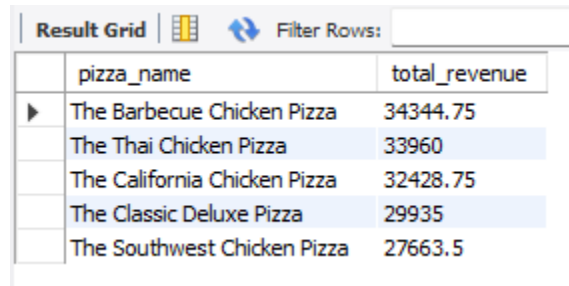


total_no_of_pizzas_sold
39289

## **2. Top 5 pizzas to be sold based on the total revenue:**

The SQL query retrieves the top 5 pizzas based on total revenue generated. It selects the pizza\_name and calculates the sum of total\_price for each pizza, grouping them by pizza\_name, ordering by total\_revenue in descending order, and limiting the result to 5 pizzas.

```
select pizza_name,sum(total_price) as total_revenue from pizza
group by pizza_name
order by total_revenue desc limit 5;
```

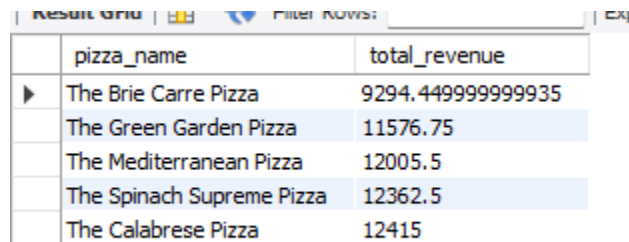


	pizza_name	total_revenue
▶	The Barbecue Chicken Pizza	34344.75
	The Thai Chicken Pizza	33960
	The California Chicken Pizza	32428.75
	The Classic Deluxe Pizza	29935
	The Southwest Chicken Pizza	27663.5

## **3. Least 5 pizzas sold based on the total revenue:**

The SQL query retrieves the least 5 pizzas sold based on total revenue. It selects pizza\_name, calculates the sum of total\_price for each pizza, groups by pizza\_name, orders by total\_revenue, and limits the result to 5 pizzas.

```
select pizza_name,sum(total_price) as total_revenue from pizza
group by pizza_name
order by total_revenue limit 5;
```



	pizza_name	total_revenue
▶	The Brie Carre Pizza	9294.449999999935
	The Green Garden Pizza	11576.75
	The Mediterranean Pizza	12005.5
	The Spinach Supreme Pizza	12362.5
	The Calabrese Pizza	12415

## **4. Sum of Total orders:**

The SQL query calculates the sum of distinct order IDs, representing the total number of orders. The result is labeled as "total\_orders".

```
select sum(distinct(order_id)) as total_orders from pizza;
```

	total_orders
▶	144304566

### **5. Averages per order can be calculated:**

The SQL query calculates the average number of pizzas per order by dividing the sum of quantities by the sum of distinct order IDs from the pizza table. The result is labeled as "average\_pizzas\_per\_order".

```
select sum(quantity) / sum(distinct(order_id)) as average_pizzas_per_order from pizza;
```

	average_pizzas_per_order
▶	0.0003

### **6. least 5 pizzas sold based on the quantity:**

The SQL query retrieves the least 5 pizzas sold based on total quantity. It selects pizza\_name, calculates the sum of quantity for each pizza, groups by pizza\_name, orders by total\_quantity, and limits the result to 5 pizzas.

```
select pizza_name,sum(quantity) as total_quantity from pizza
group by pizza_name
order by total_quantity limit 5;
```

	pizza_name	total_quantity
▶	The Brie Carre Pizza	393
	The Mediterra The Brie Carre Pizza	
	The Calabrese Pizza	732
	The Spinach Supreme Pizza	766
	The Chicken Pesto Pizza	767

### **7. top pizzas sold based on the quantity:**

The SQL query identifies the top 5 pizzas sold based on total quantity. It selects pizza\_name, sums the quantity for each pizza, groups by pizza\_name, orders by total\_quantity in descending order, and limits to 5 pizzas.

```
select pizza_name,sum(quantity) as total_quantity from pizza
group by pizza_name
order by total_quantity desc limit 5;
```

pizza_name	total_quantity
The Barbecue Chicken Pizza	1953
The Pepperoni Pizza	1933
The Classic Deluxe Pizza	1926
The Hawaiian Pizza	1887
The California Chicken Pizza	1857

### **8. Average order value in the pizzas sale:**

The SQL query calculates the average order value in the pizza sales by dividing the sum of total prices by the sum of quantities. The result is labeled as "average\_order\_value".

SELECT SUM(total\_price) / sum(quantity) as average\_order\_value FROM pizza ;

average_order_value
16.50161622846096

### **9. Percent based on the pizza category:**

The SQL query calculates the percentage of total revenue based on pizza category. It selects pizza\_category, computes the sum of total\_price for each category, and divides it by the total revenue, multiplied by 100.

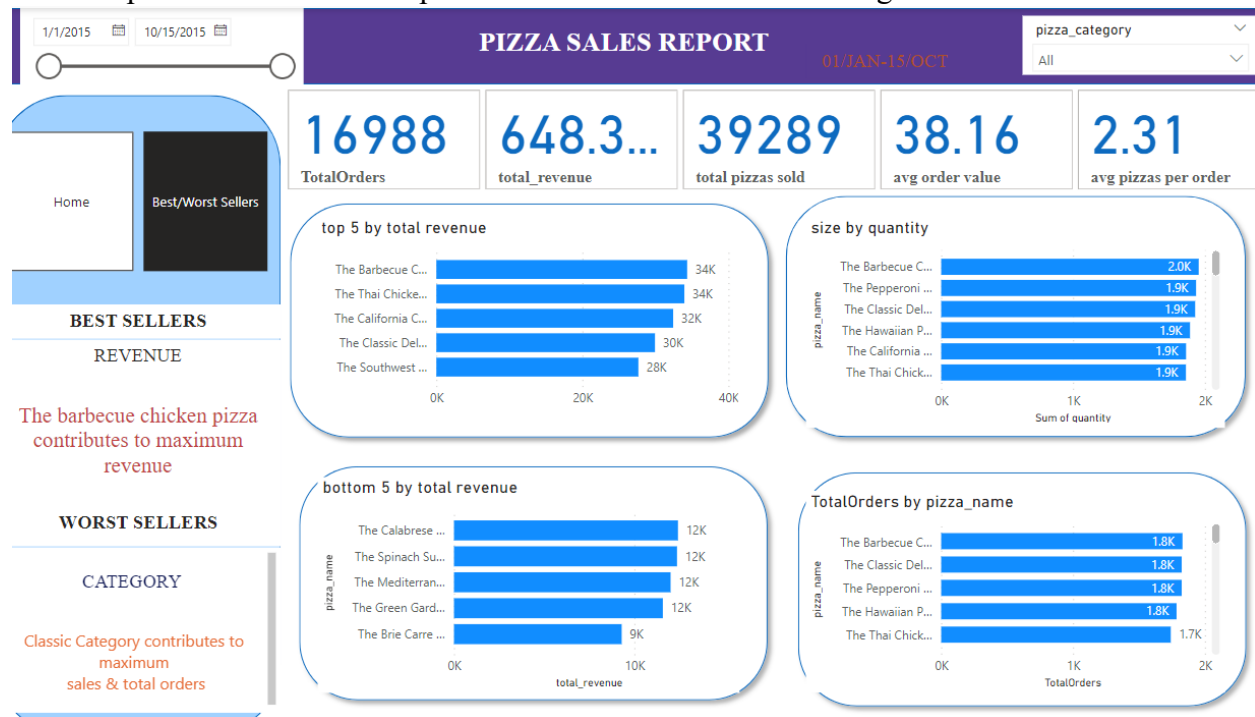
select pizza\_category ,sum(total\_price) \*100 /(select sum(total\_price) from pizza) as pct from pizza group by pizza\_category ;

pizza_category	pct
Classic	26.92144611094305
Veggie	23.804887002338216
Supreme	25.402293886465284
Chicken	23.87137300025286

### **4. User Experience Enhancement:**

Power BI holds immense importance in optimizing performance and enhancing user experience with the Pizza dataset. Its intuitive interface and robust visualization capabilities allow stakeholders to delve into key metrics like sold pizza, revenue, and top-selling items. Leveraging Power BI's integration with Python and SQL enables deeper analysis and query optimization, ensuring efficient data processing. By identifying and addressing performance bottlenecks, stakeholders can achieve faster dashboard load times and smoother interactions, ultimately empowering users with streamlined data exploration and analysis. This not only enhances decision-making but also drives operational efficiency and business success in the dynamic

realm of pizza sales. Below the picture shows the visualization using Power BI.



The report titled "PIZZA SALES REPORT" encompasses data from November 2015 to October 15, 2015. It provides insights into pizza sales metrics such as revenue, quantity sold, and top-selling items within this specified date range for analysis and decision-making purposes.

### 1. Key Metrics:

The report titled "PIZZA SALES REPORT" encompasses data from November 2015 to October 15, 2015. It provides insights into pizza sales metrics such as revenue, quantity sold, and top-selling items within this specified date range for analysis and decision-making purposes.

### 2. Top and Bottom Performers:

In the top performers category by total revenue, "The Barbecue C..." pizza stands out as the highest revenue generator. However, the image does not display the other top performers. This highlights the need for further exploration or clarification to identify additional top performers in the dataset.

### 3. Bottom 5 by Total Revenue:

The Calabrese..." pizza emerges as the lowest revenue generator among the bottom performers. However, the image does not reveal the other bottom performers, necessitating further investigation for comprehensive analysis.

### 4. Best and Worst Sellers:

In terms of revenue, "The barbecue chicken pizza" emerges as the best seller, making the highest contribution. Conversely, within the category perspective, the "Classic Category" stands out as the worst seller, despite contributing the most to sales and total orders.

### 5. **Visuals and Filters:**

The image showcases bar graphs for revenue comparison and size distribution. Additionally, it features an interactive filter enabling users to select specific pizza categories or date ranges, enhancing data exploration and analysis capabilities.

### 5. **Conclusion:**

In conclusion, the optimization of Power BI dashboards represents a critical endeavor in maximizing the efficiency and effectiveness of data exploration and analysis processes. Through the identification and mitigation of performance bottlenecks within Power BI, coupled with leveraging Python for comprehensive analysis and SQL for query optimization, significant improvements in dashboard speed, responsiveness, and user experience can be achieved. By focusing on enhancing the user experience for efficient data exploration and analysis, stakeholders can make informed decisions swiftly and effectively, thereby driving business success. The utilization of the Pizza dataset serves as a practical example, demonstrating the potential impact of optimized Power BI dashboards in facilitating data-driven decision-making processes across various domains.