

## Sales Performance Dashboard using MySQL & Power BI

**Prepared by:** Sanskruti Chandu Bhandarkar

**Domain:** Sales Analytics /Data Analytics

**Tools Used:** MySQL, SQL, Power BI, DAX

---

### 1. Introduction

Sales data plays a crucial role in understanding business performance, customer demand, and revenue trends. This project focuses on building a **Sales Performance Dashboard** by integrating a MySQL database with Power BI. The purpose of this project is to convert raw transactional sales data into meaningful insights that can support data-driven business decisions.

The project demonstrates an end-to-end analytics workflow, starting from database creation and SQL querying to data modeling and dashboard visualization in Power BI.

---

### 2. Objectives

The key objectives of this project are:

- To create and manage a structured sales database using MySQL
  - To perform sales analysis using SQL queries
  - To calculate key performance indicators (KPIs) such as total sales
  - To visualize sales trends and patterns using Power BI
  - To gain hands-on experience with SQL and Power BI integration
- 

### 3. Tools & Technologies Used

- **MySQL:** Used for creating the sales database and storing transactional data
  - **SQL:** Used for querying, aggregating, and analyzing sales data
  - **Power BI:** Used for data modeling, visualization, and dashboard creation
  - **DAX:** Used to create measures such as Total Sales
- 

### 4. Database Design

The sales database was designed using MySQL and consists of multiple related tables:

- **Customers:** Stores customer information such as customer ID, name, and region
- **Products:** Contains details of products sold
- **Orders:** Stores order-level information
- **Sales:** Contains transaction-level data including order date, quantity, and total sales amount

Primary and foreign key relationships were used to maintain data consistency and integrity across tables.

---

## 5. SQL Analysis

SQL queries were written to analyze sales data and generate meaningful summaries. Aggregate functions such as SUM() and COUNT() were used along with GROUP BY clauses to calculate sales metrics.

Key analysis performed using SQL includes:

- Calculation of total sales value
- Analysis of sales by order date
- Identification of sales distribution across regions

These queries helped transform raw transactional data into structured information suitable for visualization.

---

## 6. Power BI Dashboard Design

The sales data extracted from MySQL was imported into Power BI. Data types were verified, and relationships were validated before creating visualizations. A DAX measure was created to calculate **Total Sales**, which serves as the primary KPI of the dashboard.

The dashboard includes the following visuals:

- **KPI Card:** Displays Total Sales
- **Line Chart:** Shows sales trend by order date
- **Bar Chart:** Displays regional sales performance
- **Slicers:** Enable interactive filtering of data

The dashboard layout was designed with a clean and minimal design to enhance readability and user interaction.

---

## 7. Key Insights

The dashboard provides several important business insights, including:

- Total sales provide a quick overview of overall business performance
- Sales trend analysis helps identify periods of high and low sales activity
- Regional sales analysis highlights top-performing regions

These insights can help businesses optimize sales strategies and focus on high-revenue areas.

---

## 8. Learning Outcomes

Through this project, the following skills were developed:

- Practical application of SQL queries for data analysis
- Understanding of relational databases and data modeling
- Creation of DAX measures in Power BI
- Designing interactive and business-oriented dashboards
- Improved confidence in end-to-end data analytics projects

---

## 9. Conclusion

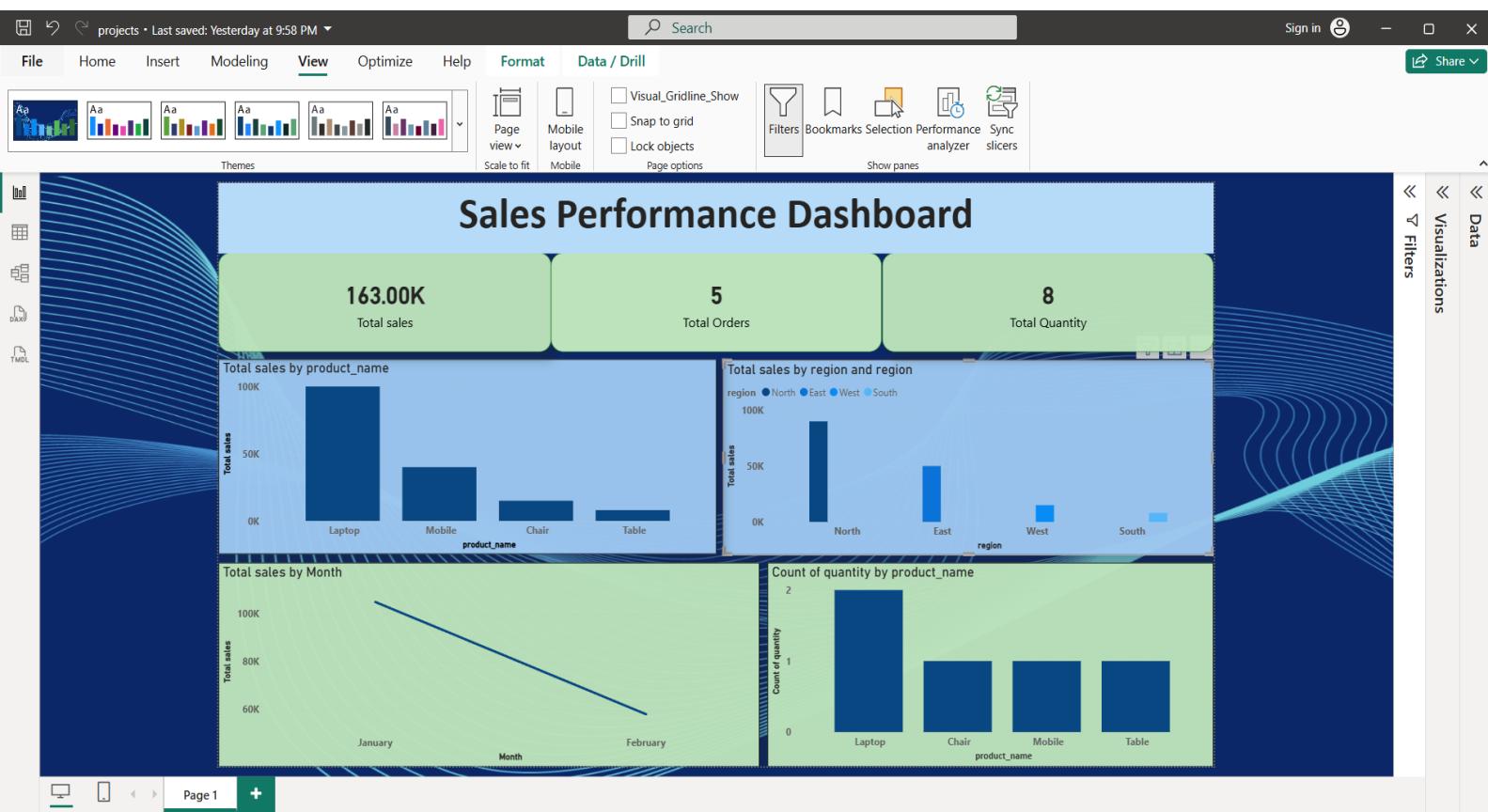
The Sales Performance Dashboard project successfully demonstrates how MySQL and Power BI can be combined to analyze and visualize sales data effectively. This project strengthened foundational data analytics skills and provided practical exposure to real-world sales analysis scenarios.

---

## 10. Future Enhancements

- Adding profit and cost analysis
- Including product-level performance metrics
- Automating data refresh
- Expanding the dashboard with advanced analytical insights

# Outputs:



projects • Last saved: Yesterday at 9:58 PM

Search

Sign in

Share

**File Home Insert Modeling View Optimize Help**

Cut Copy Format painter Clipboard

Get data Excel workbook catalog OneLake Server Data Recent sources Data Transform Refresh data New visual Text box More visual Calculations Sensitivity Publish Prep data for Copilot AI Copilot

**total sales**

Total sales by product\_name

product_name	Total sales
Laptop	100K
Mobile	50K
Chair	10K
Table	10K

**Total Orders**

Total sales by region and region

region	Total sales
North	100K
East	50K
West	10K
South	10K

**Total sales by Month**

Total sales

Month	Total sales
January	100K
February	60K

**Count of quantity by product\_name**

product_name	Count of quantity
Laptop	2
Chair	1
Mobile	1
Table	1

Page 1 of 1

30°C Sunny

107% 19-12-2023

## Command line:

```
MySQL 8.0 Command Line Client + v

+-----+-----+
| customer_id | customer_name | region |
+-----+-----+
| 1 | Apurva | North |
| 2 | Ganesh | West |
| 3 | Priya | South |
| 4 | Jitesh | East |
+-----+-----+
4 rows in set (0.07 sec)

mysql> SELECT * FROM products;
+-----+-----+-----+-----+
| product_id | product_name | category | price |
+-----+-----+-----+-----+
| 101 | Laptop | Electronics | 50000.00 |
| 102 | Mobile | Electronics | 20000.00 |
| 103 | Chair | Furniture | 5000.00 |
| 104 | Table | Furniture | 8000.00 |
+-----+-----+-----+-----+
4 rows in set (0.03 sec)

mysql> SELECT * FROM orders;
+-----+-----+-----+
| order_id | customer_id | order_date |
+-----+-----+-----+
| 1001 | 1 | 2024-01-10 |
| 1002 | 2 | 2024-01-15 |
| 1003 | 3 | 2024-02-05 |
| 1004 | 4 | 2024-02-20 |
+-----+-----+-----+
4 rows in set (0.02 sec)

mysql> SELECT * FROM sales;
+-----+-----+-----+-----+-----+
| sale_id | order_id | product_id | quantity | total_amount |
+-----+-----+-----+-----+-----+
| 1 | 1001 | 101 | 1 | 50000.00 |
| 2 | 1001 | 102 | 2 | 40000.00 |
| 3 | 1002 | 103 | 3 | 15000.00 |
| 4 | 1003 | 104 | 1 | 8000.00 |
| 5 | 1004 | 101 | 1 | 50000.00 |
+-----+-----+-----+-----+-----+
```