WideWorldImporters

Data Analysis Project

Prepared by:

- **AbdelRahman AbdelMoez Anwar**
- **Fatma Ali Khaled**
- **Youssef Mohamed Farag**
- **Noha Soliman Mohamed**

Supervised by: Eng. Ahmed Alaa
Group: ALX2_DAT1_G1 – Google Data Analyst Specialist

Table of Contents

---

1. Project Planning & Management

1.1 Project Overview

This project utilizes the **WideWorldImporters** database to generate actionable insights in the domains of **Sales, HR, Supply Chain, and Marketing**. The primary objective is to enhance **decision-making** through structured data analysis and visualization.

1.2 Objectives

- Provide **data-driven insights** for improved business decisions.
- Develop interactive **dashboards** and **reports** using SQL Server, Power BI, Python, and Excel.
- Ensure **data integrity** and optimize analytical methodologies.

1.3 Project Scope

- Exclusive focus on **data analysis methodologies**.
- Use of **SQL Server** for data extraction.
- **Python and Excel** for data processing.
- **Power BI** for dashboard creation and visualization.

1.4 Project Timeline

Week	Task
1	Database familiarization & task assignments
2	Data extraction & initial cleaning
3	Exploratory Data Analysis (EDA)
4	Dashboard design in Power BI
5	Insights refinement & dashboard usability testing
6	Drafting documentation
7	Final reporting & recommendations
8	Presentation preparation & submission

1.5 Task Assignments & Roles

- **AbdelRahman AbdelMoez Anwar** – Data Analysis & Visualization (Power BI), Documentation, Business Insights & Recommendations.
- **Fatma Ali Khaled** – Data Analysis & Data Visualization for 3 domains (Python).
- **Youssef Mohamed Farag** – SQL-based Data Extraction and Analysis.
- **Noha Soliman Mohamed** – Data Analysis & Data Visualization for HR domain (Python) & Presentation Preparation.

1.6 Risk Assessment & Mitigation Plan

Risk	Mitigation Strategy
Data Integrity Issues	Use **robust cleaning techniques** in Python (Pandas) & SQL validations.
Time Constraints	Adhere to timeline & conduct **regular progress reviews**.
Visualization Complexity	Focus dashboards on **Key Performance Indicators (KPIs)** for clarity.

1.7 Key Performance Indicators (KPIs)

- **Sales:** Revenue growth, best-selling products, customer retention rate.
- **HR:** Employee turnover rate, performance analysis.
- **Supply Chain:** Inventory turnover, supplier efficiency.
- **Marketing:** Campaign effectiveness, customer acquisition cost.

---

2. Literature Review

2.1 Feedback & Evaluation

- **Strengths:** Effective SQL queries and interactive dashboards.
- **Areas for Improvement:** Enhanced data cleaning techniques, deeper narrative insights.

2.2 Suggested Improvements

- **Advanced Analytics:** Implement predictive modeling.
- **Enhanced Dashboards:** Add interactive filters & drill-down capabilities.
- **Improved Documentation:** Provide in-depth business impact explanations.

2.3 Grading Criteria

- **Documentation:** Clarity, structure, and reporting depth.
- **Implementation:** Effective SQL, Python, and Power BI utilization.
- **Testing:** Accuracy in extraction, cleaning, and visualization.
- **Presentation:** Delivery quality, storytelling, and stakeholder relevance.

---

3. Requirements Gathering

3.1 Stakeholder Analysis

- **Sales Managers** – Revenue trends & product distribution.
- **HR Specialists** – Employee turnover & retention strategies.
- **Supply Chain Teams** – Supplier reliability & inventory management.
- **Marketing Executives** – Campaign performance & customer acquisition costs.

3.2 Functional & Non-Functional Requirements

Type	Requirements
Functional	Data Extraction (SQL), Data Cleaning (Python, Excel), Visualization (Power BI), Reporting (Excel).
Non-Functional	Performance (fast report generation), Security (restricted access), Usability (user-friendly UI).

---

4. System Analysis & Design

4.1 System Architecture

- **Backend:** SQL Server for data storage & querying.
- **Processing:** Python (Pandas, Matplotlib, Seaborn) for data manipulation.
- **Visualization:** Power BI for interactive dashboards.
- **Reporting:** Excel for detailed insights.

4.2 Data Flow & System Behavior

1. **Data Extraction** – Retrieve raw data using SQL.
2. **Data Cleaning** – Apply preprocessing using Python & Excel.
3. **Data Analysis** – Perform trend analysis & derive insights.
4. **Visualization & Reporting** – Develop dashboards & generate reports.

4.3 UI/UX Design Principles

- **Consistency:** Standardized color schemes & typography.
- **Accessibility:** Intuitive navigation & tooltips.
- **Interactivity:** Drill-down capabilities & filters.

4.4 Deployment Strategy

- **Hosting:** Power BI Service for dashboards, scheduled SQL & Python scripts.
- **Security Measures:** User-based access restrictions.

---

5. Data Analysis Track

5.1 Data Cleaning & Preprocessing

- Handle missing values using **imputation techniques**.
- Remove duplicates using **SQL DISTINCT** and **Pandas drop_duplicates()**.
- Standardize data formats (dates, currencies).

5.2 Exploratory Data Analysis (EDA)

- **Sales:** Top products, revenue trends.
- **HR:** Employee retention, salary distributions.
- **Supply Chain:** Supplier efficiency, inventory turnover.
- **Marketing:** Campaign performance & customer insights.

5.3 Data Visualization & Reporting

- Develop **interactive Power BI dashboards**.
- Create **Excel reports** with pivot tables & charts.

5.4 System Deployment & Automation

- Schedule **SQL queries** and **Python scripts** for real-time data updates.
- Secure **dashboard access** for stakeholders.

5.5 Final Deliverables

- **Executive Summary Report** – Key insights & business recommendations.
- **Stakeholder Presentation** – Storytelling through data visualizations.

---

6. Implementation (Source Code & Execution)

6.1 SQL Server Implementation

Tables to Import

Application Schema:

- Cities
- Countries
- DeliveryMethods

- PaymentMethods
- People
- StateProvinces
- SystemParameters
- TransactionTypes

Purchasing Schema:

- PurchaseOrderLines
- PurchaseOrders
- SupplierCategories
- Suppliers
- SupplierTransactions

Sales Schema:

- BuyingGroups
- CustomerCategories
- Customers
- CustomerTransactions
- InvoiceLines
- Invoices
- OrderLines
- Orders
- SpecialDeals

Warehouse Schema:

- Colors
- PackageTypes
- StockGroups

- StockItemHoldings
- StockItems
- StockItemStockGroups
- StockItemTransactions
- VehicleTemperatures

SQL Tasks

- Use `SELECT`, `WHERE`, and `JOIN` statements to retrieve required data.
- Perform aggregations for analysis.

Sample Aggregations

Total Sales by Customer:

```
SELECT CustomerID, SUM(LineTotal) AS TotalSales
FROM Sales.InvoiceLines
GROUP BY CustomerID;
```

Customer Orders Count:

```
SELECT C.CustomerName, COUNT(O.OrderID) AS TotalOrders
FROM Sales.Customers C
JOIN Sales.Orders O ON C.CustomerID = O.CustomerID
GROUP BY C.CustomerName;
```

SQL Queries for Analysis by Domain

Sales Domain

- Total Sales per Month:

```
SELECT FORMAT(InvoiceDate, 'yyyy-MM') AS SalesMonth, SUM(LineTotal) AS MonthlySales
FROM Sales.InvoiceLines
JOIN Sales.Invoices ON Sales.InvoiceLines.InvoiceID = Sales.Invoices.InvoiceID
GROUP BY FORMAT(InvoiceDate, 'yyyy-MM')
ORDER BY SalesMonth;
```

- Top 5 Customers by Sales:

```
SELECT TOP 5 C.CustomerName, SUM(IL.LineTotal) AS TotalSales
FROM Sales.InvoiceLines IL
JOIN Sales.Invoices I ON IL.InvoiceID = I.InvoiceID
JOIN Sales.Customers C ON I.CustomerID = C.CustomerID
GROUP BY C.CustomerName
ORDER BY TotalSales DESC;
```

HR Domain

- Employees Count by Department:

```
SELECT Department, COUNT(PersonID) AS EmployeeCount
FROM Application.People
GROUP BY Department;
```

- Gender Distribution:

```
SELECT Gender, COUNT(PersonID) AS Count
FROM Application.People
GROUP BY Gender;
```

Supply Chain Domain

- Inventory Movement by Month:

```
SELECT FORMAT(TransactionOccurredWhen, 'yyyy-MM') AS MovementMonth, SUM(Quantity) AS
TotalMovement
FROM Warehouse.StockItemTransactions
GROUP BY FORMAT(TransactionOccurredWhen, 'yyyy-MM')
ORDER BY MovementMonth;
```

- Top 5 Suppliers by Purchase Value:

```
SELECT TOP 5 S.SupplierName, SUM(POL.Quantity * POL.UnitPrice) AS PurchaseValue
FROM Purchasing.PurchaseOrderLines POL
JOIN Purchasing.PurchaseOrders PO ON POL.PurchaseOrderID = PO.PurchaseOrderID
JOIN Purchasing.Suppliers S ON PO.SupplierID = S.SupplierID
GROUP BY S.SupplierName
ORDER BY PurchaseValue DESC;
```

Marketing Domain

- Active Customers (last 12 months):

```
SELECT COUNT(DISTINCT CustomerID) AS ActiveCustomers
FROM Sales.Invoices
WHERE InvoiceDate >= DATEADD(MONTH, -12, GETDATE());
```

- Revenue by Customer Category:

```
SELECT CC.CustomerCategoryName, SUM(IL.LineTotal) AS Revenue
FROM Sales.InvoiceLines IL
JOIN Sales.Invoices I ON IL.InvoiceID = I.InvoiceID
JOIN Sales.Customers C ON I.CustomerID = C.CustomerID
JOIN Sales.CustomerCategories CC ON C.CustomerCategoryID = CC.CustomerCategoryID
GROUP BY CC.CustomerCategoryName;
```

6.2 Python Implementation

Objectives

- Data cleaning
- Exploratory Data Analysis (EDA)
- Visualization

Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Tasks

Clean Data:

```
df.dropna(inplace=True)
df.drop_duplicates(inplace=True)
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])
```

Create New Columns:

```
df['OrderValue'] = df['Quantity'] * df['UnitPrice']
```

Visualizations

Sales Over Time:

```
plt.figure(figsize=(10,6))
df.groupby(df['InvoiceDate'].dt.to_period('M'))['OrderValue'].sum().plot(kind='line')
plt.title('Sales Over Time')
plt.xlabel('Month')
plt.ylabel('Total Sales')
plt.show()
```

Top 10 Customers by Sales:

```
plt.figure(figsize=(12,6))
df.groupby('CustomerID')['OrderValue'].sum().nlargest(10).plot(kind='bar')
plt.title('Top 10 Customers by Sales')
plt.ylabel('Total Sales')
plt.show()
```

6.3 Power BI Implementation

Data Cleaning (Power Query)

- Remove nulls and duplicates.
- Format data types appropriately.
- Rename columns for clarity.
- Merge and shape data as needed.
- Create calculated columns (e.g., `Order Value = Quantity * UnitPrice`).

Relationships

- Define one-to-many relationships (e.g., Customers → Orders).
- Use primary keys such as `CustomerID`, `StockItemID`, `InvoiceID`.

DAX Measures Examples

```
Total Sales = SUM('InvoiceLines'[LineTotal])

Average Order Value = AVERAGE('Orders'[OrderValue])

Total Orders = COUNT('Orders'[OrderID])

Total Units Sold = SUM(Sales.OrderLines[Quantity])

Quantity of Purchases = SUM(Purchases[Ordered Quantity])

# Employees = DISTINCTCOUNT(Employees[Employee Key])

# Customers = DISTINCTCOUNT(Customers[Customer Key] )

# Suppliers = DISTINCTCOUNT(Suppliers[WWI Supplier ID] )
```

Dashboard Design (4 Pages)

Page 1: Sales Dashboard

- **Cards (KPIs):** Total Sales, Total Profits, Total Orders, Average Order Value, Total Units Sold
- **Slicers:** Order Date, Customer Name, Product Name
- **Charts:**
    - Line Chart: Sales Over Time
    - Bar Chart: Sales by Customer
    - Pie Chart: Sales by Product
    - Column Chart: Orders per Year

Page 2: HR Dashboard

- **Cards (KPIs):** Total Employees, Average Age, Gender Count, New Hires This Year
- **Slicers:** Department, Date , Employment Status
- **Charts:**
    - Bar Chart: Employees by Department
    - Pie Chart: Gender Distribution
    - Line Chart: Hires Over Time
    - Column Chart: Employee Status

Page 3: Supply Chain Dashboard

- **Cards (KPIs):** Total Products, Total Stock Value, Transactions, Total Suppliers
- **Slicers:** Product Name, Supplier Name, Transaction Date
- **Charts:**
    - Line Chart: Inventory Movements
    - Bar Chart: Orders by Supplier
    - Column Chart: Yearly Purchases

- Pie Chart: Inventory by Stock Category

Page 4: Marketing Dashboard

- **Cards (KPIs):** Active Customers, Customer Retention Rate, Avg Spend per Customer, Top Product
- **Slicers:** Customer Category, Region, Marketing Campaign

- **Charts:**
  - Bar Chart: Customer Spend Segments
  - Line Chart: Customer Growth
  - Pie Chart: Regional Sales
  - Column Chart: Marketing Campaign Revenue

---

7. Conclusion

This project provides a structured approach for **analyzing WideWorldImporters database** using industry-standard tools. The combination of **SQL, Python, and Power BI** ensures data integrity and enables effective decision-making in **Sales, HR, Supply Chain, and Marketing**. The insights generated will help stakeholders optimize business strategies and drive growth.

GitHub Link: https://github.com/abdelrahmanabdelmoez/DEPI-Graduation-Project/tree/main