Business Intelligence Project Report

Design and Implementation of a Sales Data Warehouse and Dashboard Using the Northwind Dataset

Bibliography

1. Introduction

In today's data-driven organizations, Business Intelligence (BI) systems play a critical role in transforming raw operational data into meaningful insights that support decision-making. Transactional databases are optimized for daily operations but are not suitable for analytical workloads. For this reason, data warehouses and BI tools are used to aggregate, clean, and analyze data efficiently.

The objective of this project is to design and implement a complete Business Intelligence solution for the fictitious company **Northwind Traders**. The project covers the entire BI pipeline, starting from heterogeneous data sources, through an ETL (Extract, Transform, Load) process implemented in Python, the creation of a data warehouse using a star schema, and finally the development of an interactive analytical dashboard using Power BI.

2. Project Objectives

The main objectives of this project are:

- To integrate data from multiple heterogeneous sources (SQL Server, Microsoft Access, and Excel).
- To design and implement a relational data warehouse optimized for analytical queries.
- To build an automated ETL process using Python.
- To calculate and store key performance indicators (KPIs) related to sales and delivery.
- To develop an interactive Power BI dashboard for business analysis.
- To provide insights that support managerial decision-making.

3. Data Sources

The project uses the **Northwind** dataset, which represents a sales and distribution company. Data is extracted from multiple sources in order to simulate a real enterprise environment.

3.1 SQL Server (OLTP System)

- Database: **Northwind**
- Tables used:

    - Customers
    - Orders
    - Order Details
    - Employees

This database represents the **operational transactional system (OLTP)**.

3.2 Microsoft Access

- File: **Northwind 2012.accdb**
- Used to simulate an additional external data source.
- Demonstrates heterogeneous data integration.

3.3 Excel Files

- Orders.xlsx
- Customers.xlsx
- Employees.xlsx
- Order Details.xlsx

Excel files represent flat-file data sources commonly used in organizations.

4. Global Architecture

The project follows a classic Business Intelligence architecture:

1. **Source Systems**
   SQL Server, Microsoft Access, and Excel files.
2. **ETL Layer**
   Python scripts using pandas and pyodbc to extract, transform, and load data.
3. **Data Warehouse**
   SQL Server database named **NEWW**, designed using a star schema.

4. **Visualization Layer**
   Power BI dashboard connected directly to the data warehouse.

This separation ensures performance, scalability, and data consistency.

5. Data Warehouse Design

5.1 Data Warehouse Creation

A dedicated database named **NEWW** was created to isolate analytical workloads from transactional systems.

5.2 Star Schema Model

The data warehouse is modeled using a **star schema**, which simplifies queries and improves performance.

5.2.1 Dimension Tables

DimDate

- Contains one row per calendar date.
- Attributes include Year, Quarter, Month, Day, Month Name, Day of Week, and Weekend indicator.
- Enables time-based analysis such as trends, YTD, and comparisons.

DimCustomer

- Stores customer information.
- Uses a surrogate key (CustomerKey).
- Includes a SourceSystem attribute to support multiple data sources.

DimEmployee

- Stores employee information.
- Enables sales performance analysis by employee.

5.2.2 Fact Table

FactOrders

- Stores transactional and aggregated sales data.
- Measures include:
  - TotalAmount
  - Freight
  - DeliveryDelayDays
- Contains foreign keys linking to all dimension tables.
- Includes calculated indicators such as IsDelivered.

6. ETL Process

The ETL process is implemented in Python using pandas and pyodbc.

6.1 Extraction

Data is extracted from:

- SQL Server (Northwind database)
- Microsoft Access database
- Excel files

A generic extraction function is used to read data from different sources using a unified interface.

6.2 Transformation

During the transformation phase:

- Data types are standardized.
- Missing values are handled.
- Customer and employee identifiers are normalized.
- Sales amounts are calculated using the formula:

$$\text{TotalAmount} = \text{UnitPrice} \times \text{Quantity} \times (1 - \text{Discount})$$

- Delivery indicators such as delivery delay and delivery status are calculated.
- A complete Date dimension is generated programmatically.

6.3 Loading

Cleaned and transformed data is loaded into the data warehouse:

- Dimension tables are loaded first.
- FactOrders is loaded after resolving foreign keys.
- Referential integrity is enforced after loading.

This approach ensures data consistency and reliability.

7. Key Performance Indicators (KPIs)

The following KPIs are calculated and analyzed:

- Total Sales
- Total Number of Orders
- Average Order Value
- Total Freight Cost
- Number of Delivered Orders
- Number of Pending Orders
- Average Delivery Delay
- Year-to-Date Sales
- Sales Growth Percentage

These indicators provide both financial and operational insights.

8. Dashboard

A dashboard was developed, connected directly to the data warehouse.

8.1 Dashboard Features

- KPI cards displaying global metrics.
- Line chart showing sales evolution over time.
- Bar chart showing sales by country.
- Column chart showing sales by employee.
- Pie/Donut chart displaying delivered vs pending orders.
- Slicers for year, country, and employee role.

8.2 Benefits

The dashboard allows users to:

- Monitor overall business performance.
- Identify top-performing regions and employees.
- Analyze delivery efficiency.
- Explore data interactively using filters.

9. Data Validation and Exploratory Analysis

Python was also used for data validation and exploratory analysis:

- KPI values were verified by recalculating them using pandas.
- Exploratory charts were created using matplotlib and seaborn.
- This step ensured consistency between the data warehouse and Power BI dashboard.

10. Results and Insights

The analysis highlights:

- Sales trends over time and seasonal variations.
- Key customer markets contributing the highest revenue.
- Differences in employee sales performance.
- Delivery delays impacting order fulfillment.

These insights can help management improve logistics, optimize sales strategies, and enhance customer satisfaction.

11. Limitations and Future Improvements

Limitations of the project include:

- Limited historical data.
- No cost data available to calculate profit margins.
- Static data without real-time updates.

Future improvements could include:

- Integration of real-time data sources.
- Addition of cost and inventory data.
- Implementation of Slowly Changing Dimensions (Type 2).
- Deployment of the dashboard to Power BI Service.

12. Conclusion

This project demonstrates the complete implementation of a Business Intelligence system, from data extraction to visualization. By integrating heterogeneous data sources, designing a star-schema data warehouse, and developing an interactive Power BI dashboard, the project provides a realistic example of a modern BI solution. The system enables efficient analysis of sales and operational performance and supports data-driven decision-making.