Business Intelligence Project

End-to-End ETL, Data Warehouse and Analytics Project

Ahmed Ayoub Benchetioui

3rd year Cyber Security, 232331451018

December 20, 2025

Contents

Chapter 1

Introduction

In today's data-driven world, organizations continuously generate large volumes of data through daily business operations such as sales, customer management, logistics, and supply chain activities. However, raw operational data stored in transactional databases is not directly suitable for analytical purposes. Business Intelligence (BI) systems address this limitation by providing structured processes and architectures that transform raw data into meaningful insights.

This project focuses on the design and implementation of a complete Business Intelligence solution. It covers the full BI lifecycle, starting from data extraction, transformation, and loading (ETL), continuing with data warehouse modeling, and ending with analytical dashboards and Key Performance Indicators (KPIs). The project is implemented using Python, SQL, and Jupyter notebooks, following industry-standard BI practices.

The main objective of this work is to demonstrate how structured operational data can be transformed into a reliable analytical system that supports decision-making at both operational and strategic levels.

Chapter 2

Project Objectives

The primary objectives of this Business Intelligence project are as follows:

- Design and implement a robust ETL pipeline
- Clean, transform, and validate raw transactional data
- Build a dimensional data warehouse optimized for analytics
- Define relevant business KPIs
- Provide analytical insights through dashboards and reports

This project aims to replicate real-world BI systems used in enterprises, while remaining modular, scalable, and easy to maintain.

Chapter 3

Global Architecture of the BI System

The BI system implemented in this project follows a layered architecture, which is a widely adopted design in data engineering and analytics.

3.1 Architecture Layers

The system is structured into three main layers:

1. **Data Source Layer**: This layer contains the operational database with transactional business data.
2. **Data Integration Layer**: This layer is responsible for ETL operations using Python and SQL.
3. **Analytics Layer**: This layer provides dashboards, KPIs, and analytical insights.

This separation ensures better scalability, maintainability, and clarity in data processing responsibilities.

Chapter 4

Source Data Analysis

4.1 Operational Data Description

The project uses a structured relational database similar to the Northwind dataset. The data models typical business operations and includes entities such as customers, orders, products, employees, suppliers, and categories.

Each table serves a specific operational purpose. For example, the orders and order details tables capture transactional sales information, while product and customer tables provide descriptive attributes.

4.2 Limitations of Operational Databases

Although operational databases are efficient for transaction processing, they are not suitable for analytical workloads. The main limitations include:

- Highly normalized schema requiring complex joins
- Poor performance for aggregation queries
- Lack of historical and analytical perspectives
- Difficulty in computing business KPIs efficiently

These limitations justify the need for a dedicated data warehouse.

Chapter 5

ETL Process Implementation

5.1 Extraction Phase

The extraction phase consists of retrieving data from the relational source database using SQL queries and Python database connectors. This phase ensures that all relevant tables are copied into a staging environment.

The extraction process is designed to be repeatable and reliable, allowing the BI system to refresh data periodically without manual intervention.

5.2 Transformation Phase

The transformation phase is the most critical part of the ETL pipeline. It includes:

- Data cleaning (handling missing or inconsistent values)
- Data type normalization
- Creation of derived attributes (e.g., total sales, time dimensions)
- Integrity and consistency validation

Python scripts and Jupyter notebooks are used extensively to perform transformations and validate intermediate results.

5.3 Loading Phase

In the loading phase, the transformed data is inserted into the data warehouse tables. SQL scripts ensure referential integrity between fact and dimension tables.

This phase finalizes the ETL pipeline and prepares the data for analytical queries and reporting.

Chapter 6

Data Warehouse Design

6.1 Dimensional Modeling

The data warehouse is designed using a star schema, which is the most common modeling approach in BI systems. The star schema consists of:

- A central fact table containing measurable business events
- Multiple dimension tables providing descriptive context

The fact table stores quantitative metrics such as sales amounts and quantities, while dimension tables store attributes related to customers, products, time, employees, and suppliers.

6.2 Benefits of the Star Schema

The star schema offers several advantages:

- Simplified query structure
- Improved query performance
- Clear business semantics
- Compatibility with BI and visualization tools

This design significantly improves analytical efficiency compared to normalized schemas.

Chapter 7

Key Performance Indicators

7.1 Definition of KPIs

Key Performance Indicators (KPIs) are essential for measuring business performance. In this project, the following KPIs were defined:

- Total revenue
- Number of orders
- Average order value
- Revenue by product category
- Revenue by region
- Monthly sales growth

These KPIs provide a comprehensive view of business performance from multiple perspectives.

7.2 Analytical Computation

KPIs are computed using SQL aggregation queries and Python-based analytics. Time-based analysis is enabled through a dedicated time dimension, allowing trend analysis and period comparisons.

Chapter 8

Dashboards and Analytics

8.1 Dashboard Design Principles

Dashboards are designed according to BI best practices, emphasizing clarity, readability, and decision support. The focus is on presenting key metrics concisely while avoiding unnecessary visual complexity.

8.2 Analytical Capabilities

The dashboards enable users to:

- Monitor business performance
- Identify trends and anomalies
- Compare performance across categories or periods
- Support strategic and operational decisions

Chapter 9

Scalability and Big Data Perspective

Although the dataset used in this project is relatively moderate in size, the architecture and methodology are aligned with Big Data principles. The ETL pipeline is modular and can be extended to handle larger datasets.

Potential scalability improvements include:

- Migration to distributed processing frameworks
- Integration with cloud-based data warehouses
- Automation of ETL scheduling

These enhancements would allow the system to evolve into a full-scale enterprise BI solution.

Chapter 10

Results and Discussion

The implemented BI system successfully transforms raw transactional data into structured analytical insights. The data warehouse enables efficient querying, while KPIs provide clear performance indicators.

The project demonstrates how proper data modeling and ETL design significantly improve the usability and analytical value of business data.

Chapter 11

Conclusion and Future Work

This project presents a complete Business Intelligence solution covering ETL processes, data warehouse design, KPI computation, and analytical dashboards. It highlights the importance of BI systems in enabling data-driven decision-making.

Future work may include advanced analytics such as forecasting, customer segmentation, and real-time data processing. Overall, this project provides a solid foundation for further exploration in Business Intelligence and Big Data systems.