Data Engineer Track

Integrated Data Infrastructure

CAI1_AIS4_S2e

Cairo

رواد مصر الرقمية

وزارة الاتصـــــالات
وتكنولوجيا المعلومات

Outline

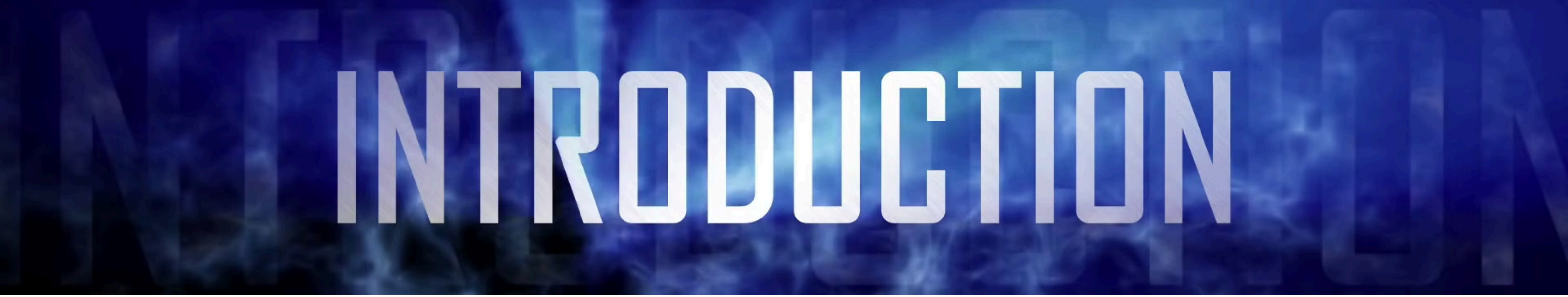
Introduction

TeamMembers

Project Breakdown and Timeline

Tools & Technologies Used

Conclusion



In our project

we focused on building a scalable data pipeline that extracts, transforms, and loads (ETL) data into a centralized data warehouse

We processed various data formats like CSV, XML, and JSON using powerful tools such as SQL, Python, Hadoop, and Apache Spark to handle big data.

We utilized Hadoop to manage and store large datasets efficiently, while Apache Spark enabled us to process and analyze this data quickly.

By leveraging these big data technologies, we transformed raw data into actionable insights.

Finally, we visualized these insights using Python and Power BI, making the data easy to understand and useful for decision-making.

Team Members

1

Ziad Abdelqader |Hamed Samir

- Develop a data pipeline using Python to load CSV files into PostgreSQL.
- Perform SQL-based analysis and data visualization using Python.

2

Belal Abdelraouf | Yousef Rady

- Create a data pipeline to load XML files into MySQL using Python.
- Conduct data analysis and visualization using Power BI.

3

Abdullah Ibrahim | Mohamed Ashraf

- Perform ETL on data from various sources (CSV, XML, JSON).
- Clean, transform, and load the data into a data warehouse using Python.
- Design and implement a star schema for the data warehouse using fact and dimension tables.
- Apply big data tools like Hadoop and Apache Spark for advanced data analysis and visualization.

Project Breakdown

1. Design and Implement the Data Pipeline

Objective:

To design and develop a robust data engineering pipeline capable of handling multiple data formats such as CSV, XML, and JSON, …… files

Data Gathering (All Team Members)

- Our team collected data from multiple formats (CSV, XML, JSON) to test the versatility of the pipeline.
- Team Members: All team members participated in gathering datasets aligned with the project’s goals.

Project Breakdown

1. Design and Implement the Data Pipeline

Data Cleaning (Abdullah | Mohamed)

- Using Python libraries such as Pandas andNumPy
- we cleaned and preprocessed the data.

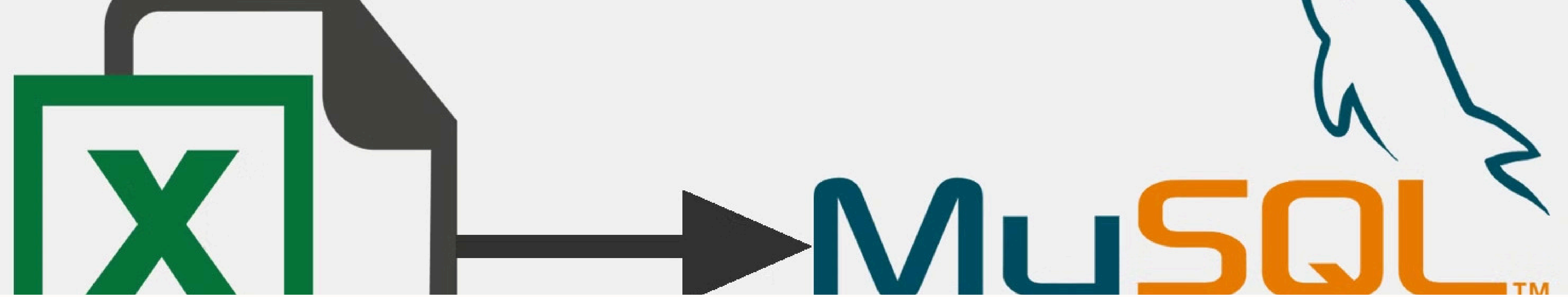
Removing duplicates.

Handling missing values via imputation.

Normalizing formats for consistency.

- Member Responsibility:

worked together to ensure the datasets were cleaned and ready for loading into databases.



Project Breakdown

1. Design and Implement the Data Pipeline

Data Loading and Analysis

- Loading XML to MySQL (Belal | Yousef)

MySQL was chosen:
to store and manage XML data due to its compatibility with hierarchical structures and its strong relational database capabilities. MySQL's performance with structured queries and relationships made it ideal for loading XML data, which often involves nested tags and requires a flexible schema.

- Member Responsibility

managed the MySQL database and load data into it and implemented python scripts to load data into MySQL and SQL query on the XML data

Project Breakdown

1. Design and Implement the Data Pipeline

Data Loading and Analysis

- Loading CSV to PostgreSQL (Ziad | Hamed):

PostgreSQL was selected

for its powerful support for large-scale data handling and its advanced features, such as indexing and full-text search, which made it ideal for managing and querying CSV data. PostgreSQL's ability to handle large datasets efficiently, along with its ACID compliance and extensibility, made it the perfect choice for CSV data analysis.

- Member Responsibility

configured the PostgreSQL environment for optimized performance

executed analytical queries on the data to extract meaningful insights.

Project Breakdown

2.Data Warehousing (DWH) Using Star Schema

Objective:

Create a star schema-based data warehouse for optimal data storage and retrieval, enabling OLAP for analytical processing.

1 **We use SSIS Tool**

for data integration (ETL), allowing easy migration of data into a centralized data warehouse.

SSIS is preferred for its flexibility in handling complex ETL tasks, scheduling automation, and compatibility with different data sources.

2 **Star Schema Design:**

- Create Fact and Dimension tables for optimized query performance.
- Why Star Schema? It simplifies queries and provides fast performance for OLAP processes, making it ideal for large-scale analytics.

3 **OLTP vs. OLAP**

- OLTP systems like MySQL and PostgreSQL handle real-time transactional data.
- OLAP is used for analytical processing in the DWH, where aggregated data is used for reporting and decision-making.

Project Breakdown

3. Data Analysis & Visualization

Objective:

Transform raw data into actionable insights using Python for visualization

Ziad | Hamed

Focused on visualization using Python, leveraging powerful libraries like Matplotlib and Seaborn to create detailed, custom visualizations and perform advanced data manipulation.

Why Python for Visualization?

- Powerful Libraries:

Utilizes libraries like Matplotlib and Seaborn for creating detailed and custom visualizations.

- Flexibility: Allows for advanced data manipulation and tailored visual output
- In-depth Analysis: Supports complex statistical visualizations and exploratory data analysis.

Project Breakdown

3. Data Analysis & Visualization

Objective:

Transform raw data into actionable insights using Power Bi for visualization

Team Responsibilities:

- Belal |Yousef

Specialized in using Power BI for creating interactive dashboards and reports, utilizing its user-friendly interface and seamless data integration capabilities.

Why Power BI for Visualization?

- User-Friendly Interface:

Provides an intuitive platform for building reports, making it accessible to non-technical users.

- Interactive Dashboards:

Enables users to explore data dynamically and in real time.

Project Breakdown

3. Big Data Processing with Hadoop and Spark.

Objective:

Handle large-scale data using Hadoop for storage and Apache Spark for faster, analysis.

3.1 Hadoop for Data Storage and Initial Processing (Mohamed Ashraf)

Why Hadoop?

Hadoop's distributed storage (HDFS) allowed us to manage large datasets that exceeded traditional database capacities

Steps Taken

Data was processed using Hadoop's MapReduce framework.

Hadoop Streaming allowed us to write jobs in Python, adding flexibility to the processing.

Challenges

Hadoop's batch-processing nature resulted in slower performance for analysis, which led us to explore Spark.

Member Responsibility:

handled all Hadoop-related tasks, including data ingestion into HDFS and processing with MapReduce

Project Breakdown

3. Big Data Processing with Hadoop and Spark.

Objective:

Handle large-scale data using Hadoop for storage and Apache Spark for faster, analysis.

3.2. Transition to Apache Spark for Enhanced Analysis (Abdullah Ibrahim)

Why Spark?

Spark's in-memory computation drastically improved the speed of data processing.

It also provided superior support for semi-structured data (JSON).

RDDs

for Semi-Structured Data (JSON) Initially,we used RDDs (Resilient Distributed Datasets) for fine-grained control over the semi-structured data.

Transition to Data Frames

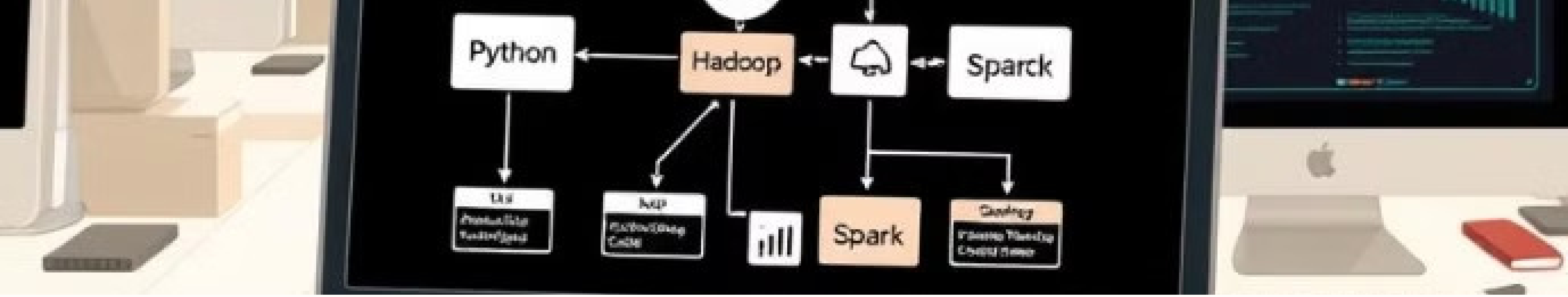
We transitioned to Data Frames, a higher-level abstraction optimized for SQL-like queries and complex analysis(window funcation).

Data Visualization and Dashboard

create dashboards that provided actionable insights.

Member Responsibility:

Abdullah Ibrahim was responsible for migrating to Py Spark and generating visualizations for insights on sales data.



Tools & Technologies Used

1

Python

Used for ETL tasks, data cleaning, and analysis.

2

SSIS

: For ETL to the data warehouse

3

PostgreSQL & MySQL

: Databases for SQL-based analysis and initial data storage.

4

Hadoop

Distributed storage and data processing.

5

Apache Spark

For fast, large-scale data processing and advanced analytics

6

Power BI & Python

For visualization and reporting

Conclusion

This project has demonstrated our team's ability to design and implement a scalable and efficient data pipeline and data warehouse solution.

By leveraging a variety of tools and technologies, we have created a comprehensive data infrastructure that enables effective data processing, analysis, and visualization.

We are grateful to DEPI and Engineer Hazem for their support and guidance throughout this journey,

GitHub repo: **https://github.com/ABDULLAH-ibrahimm/graduation_project-Depi.git**