Data Engineering Project

This Data Engineering project was carried out with the objective of efficiently handling, transforming, and visualizing data using various tools and techniques. Below are the main phases of the project..

About the Project:

Tools Used::

- Python
- Jupyter
- SQLAlchemy
- PostgreSQL
- PowerBI

About the Data:

- **FirstName:** The candidate's first name
- **LastName:** The candidate's last name
- **Email:** The candidate's email address
- **ApplicationDate:** The date the candidate applied for the job
- **YOE:** Years of experience
- **Seniority:** The position the candidate is applying for
- **Technology:** The area or field the candidate is applying for
- **Code Challenge Score:** A score from 0 to 10 for the practical test
- **Technical Interview Score:** A score from 0 to 10 for the theoretical or technical interview

Example of the CSV:

candidates.csv

First Name	Last Name	Email	Application Date	Country	YOE	Seniority	Technology	Code Challenge Score	Technical Interview Score
Bernadette	Langworth	leonard91@yahoo.com	2021-02-26	Norway	2	Intern	Data Engineer	3	3
Camryn	Reynolds	zelda56@hotmail.com	2021-09-09	Panama	10	Intern	Data Engineer	2	10
Larue	Spinka	okey_schultz41@gmail.com	2020-04-14	Belarus	4	Mid-Level	Client Success	10	9
Arch	Spinka	elvera_kulas@yahoo.com	2020-10-01	Eritrea	25	Trainee	QA Manual	7	1
Larue	Altenwerth	minnie.gislason@gmail.com	2020-05-20	Myanmar	13	Mid-Level	Social Media Community Management	9	7
Alec	Abbott	juanita_hansen@gmail.com	2019-08-17	Zimbabwe	8	Junior	Adobe Experience Manager	2	9
Allison	Jacobs	alba_rolfson27@yahoo.com	2018-05-18	Wallis and Futuna	19	Trainee	Sales	2	9
Nya	Skiles	madisen.zulauf@gmail.com	2021-12-09	Myanmar	1	Lead	Mulesoft	2	5
Mose	Lakin	dale_murazik@hotmail.com	2018-03-13	Italy	18	Lead	Social Media Community Management	7	10
Terrance	Zieme	dustin31@hotmail.com	2022-04-08	Timor-Leste	25	Lead	DevOps	2	0

1. Data Storage in PostgreSQL with SQLAlchemy

The process began with importing a CSV file that contained key data for analysis. This file was carefully selected for its relevance to the project, and its content included various columns with structured information that needed to be efficiently

stored in a database. To achieve this, SQLAlchemy, a powerful Python library that facilitates interaction with relational databases, was used.

First, a connection to a local PostgreSQL database was configured, leveraging SQLAlchemy's flexibility to handle communication between Python and the database. This connection was established using a connection URI, which includes specific details such as the database type, server location, and the credentials needed to access it.

With the connection established, the tables where the CSV data would be stored were defined. SQLAlchemy allows these tables to be defined using an object-oriented modeling approach, where each Python class represents a table in the database, and each class attribute corresponds to a column in the table. This method facilitates the creation of well-structured database schemas and ensures that the relationships between different data sets are clear and properly maintained.

Then, the data was inserted. The CSV file was read using pandas, a widely used library for data manipulation in Python, and the data was transformed into objects that SQLAlchemy could handle. These objects were then efficiently inserted into the database, taking advantage of SQLAlchemy's capabilities to manage large volumes of data. This process not only ensured that the data was organized and accessible but also laid the groundwork for the subsequent phases of the project, where the database structure would be crucial for data analysis and visualization.

2. Exploratory Data Analysis (EDA)

After storing the data in PostgreSQL, an Exploratory Data Analysis (EDA) was conducted, a crucial stage in the data science workflow. The main goal of this phase was to gain a deep understanding of the data and prepare for subsequent modeling and analysis stages.

The EDA began with a thorough inspection of the statistical characteristics of the data. This included calculations of metrics such as mean, median, standard deviation, and frequency distribution. Relationships between different variables were evaluated, looking for correlations that might be significant for future analysis.

Additionally, outliers, missing data, and potential errors in the dataset were identified and addressed during this phase. This was essential to ensure the quality and integrity of the data before proceeding with any transformation or modeling.

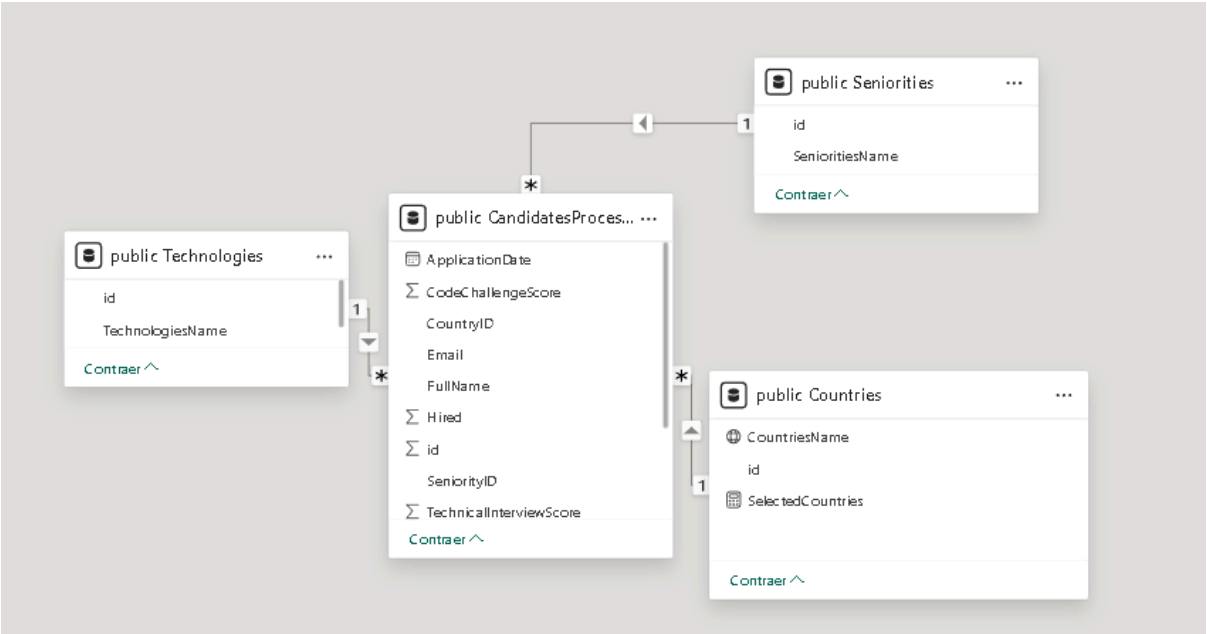
To complement the statistical analysis, tools like pandas and matplotlib were used to visualize the data. Visualizations, such as histograms, scatter plots, and box plots, provided a clear view of the data distribution and helped identify patterns, trends, and anomalies more intuitively. These visualizations not only facilitated the

understanding of the data but also effectively communicated the findings to other stakeholders.

Finally, the EDA served as a foundation for planning the necessary data transformations. With a clear understanding of the data's structure and characteristics, normalization, grouping, and aggregation strategies were defined for the next phase. This approach allowed for optimizing data usage in the star schema that would be constructed later, ensuring that the final data structure was the most suitable for analysis and visualization in the Power BI dashboard.

3. Data Transformation and Star Schema Creation

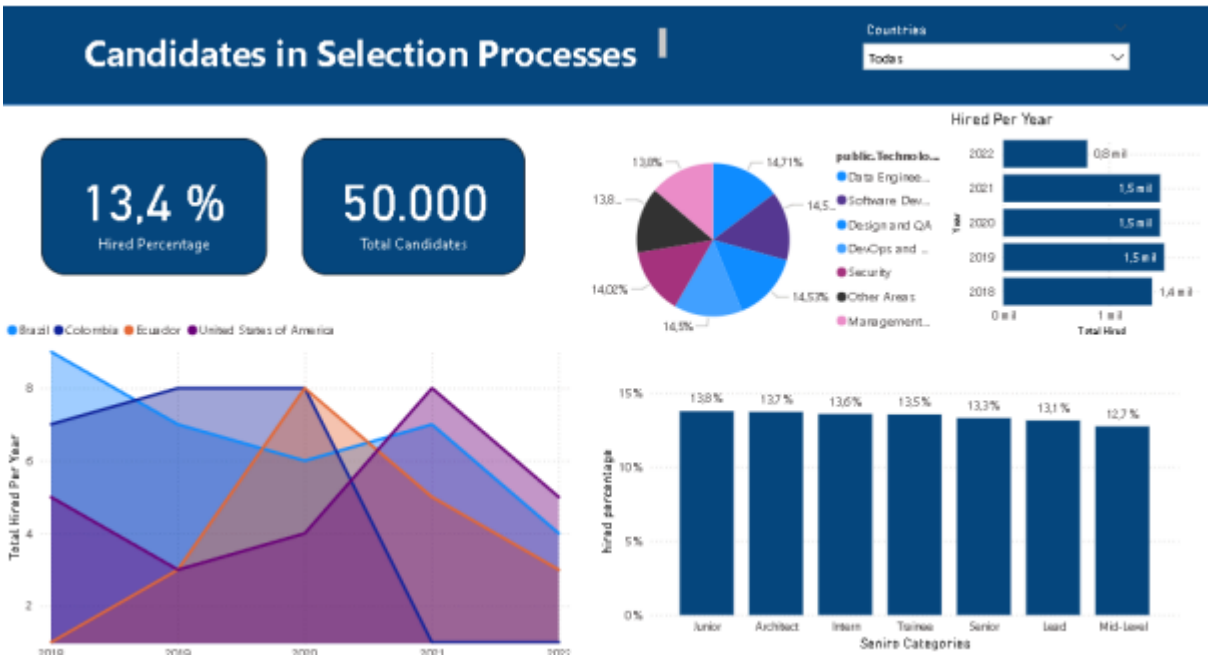
Based on the EDA results, it was decided to transform the data into a more efficient and query-friendly model, opting for a star schema. This type of organizational model is widely used in data warehouses to optimize analytical queries. The transformation involved creating fact and dimension tables, where fact tables store events or transactions, and dimension tables contain descriptive attributes. This restructuring significantly improved query performance and facilitated report generation in later stages.



For example, in the dashboard (as shown in the image), a line graph compares the total number of hired candidates by country over several years. This visualization provides a clear perspective on hiring trends in different countries, which could be useful for identifying emerging markets or areas that need attention. Additionally, the pie chart breaks down hiring by technological areas, allowing users to quickly see which areas are in the highest demand.

4.Data Visualization with Power BI

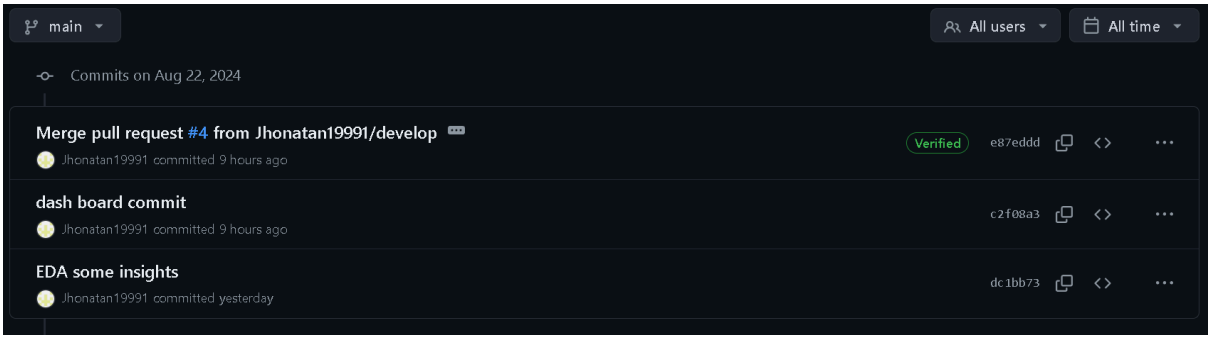
Finally, the transformed data was imported into Power BI, where an interactive dashboard was created. This dashboard provides an intuitive visualization of the data, allowing end-users to explore the information dynamically. Power BI offered various visualization options, such as bar charts, timelines, pivot tables, among others, which facilitated data-driven decision-making.



For example, in the dashboard (as shown in the image), a line graph compares the total number of hired candidates by country over several years. This visualization provides a clear perspective on hiring trends in different countries, which could be useful for identifying emerging markets or areas that need attention. Additionally, the pie chart breaks down hiring by technological areas, allowing users to quickly see which areas are in the highest demand.

5. Documentation and Git Repository

The entire process was carefully documented and uploaded to a Git repository. This documentation included technical details about the database configuration, the code used for data transformation, and the Power BI dashboard settings. By keeping a record in Git, it was ensured that the project would be saved on GitHub servers and not just locally, preventing the loss of locally made processes.



To run the project in your environment, you should follow the procedure outlined in the README. The repository was organized as follows:

- **data:** This folder contains the candidates.csv file used in this project.
- **python_code:** This folder contains the database configuration and the transform.py file, which is used to normalize the data.
- **notebooks:** This folder contains the notebooks used in this project, including the exploratory data analysis and processed data.
- **src:** This folder contains the Python code used to connect to the database and create the models on the tables.

6. Conclusión

This project demonstrated a complete and effective workflow in data engineering, from importing a CSV file to creating an interactive dashboard in Power BI. By using SQLAlchemy and PostgreSQL to store and manage the data, a solid structure was ensured for the Exploratory Data Analysis (EDA), which allowed for identifying patterns and planning transformations. Finally, the visualization in Power BI provided an intuitive way to explore the information, facilitating decision-making. The entire process was documented and uploaded to a Git repository, showcasing an efficient integration of tools and techniques to turn data into valuable information.