ETL Workshop 001 - Final Report: Candidates Analysis

David Alejandro Cajiao Lazt

Repository

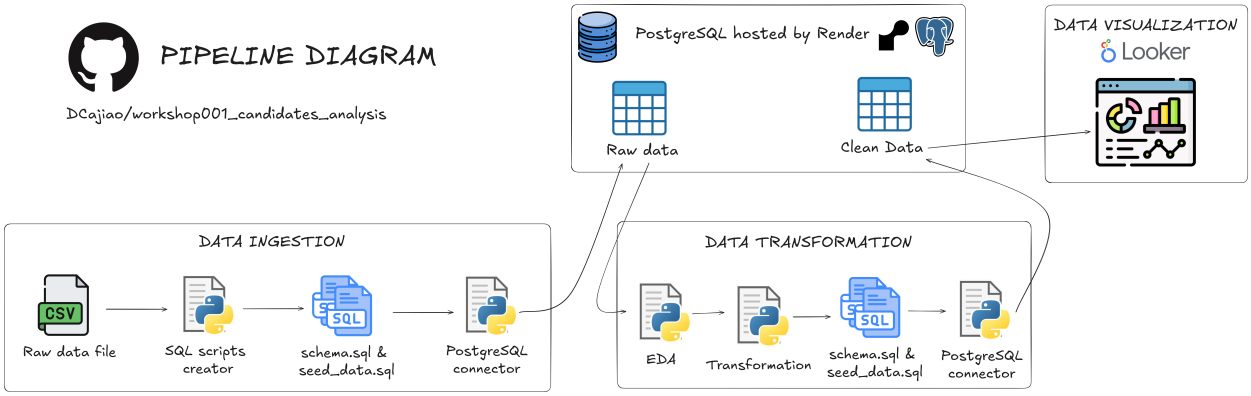
Looker Dashboard

Notion Page:

Introduction

This report provides a detailed overview of the ETL (Extract, Transform, Load) process performed in this workshop, focusing on the specific tasks performed at each step, the key findings and the relevance of certain tools and scripts, such as `pysqlschema.py`. The goal was to build a robust pipeline for analyzing candidate data, from data ingestion to final visualization.

The following diagram was designed for the pipeline to be created in the development of the project



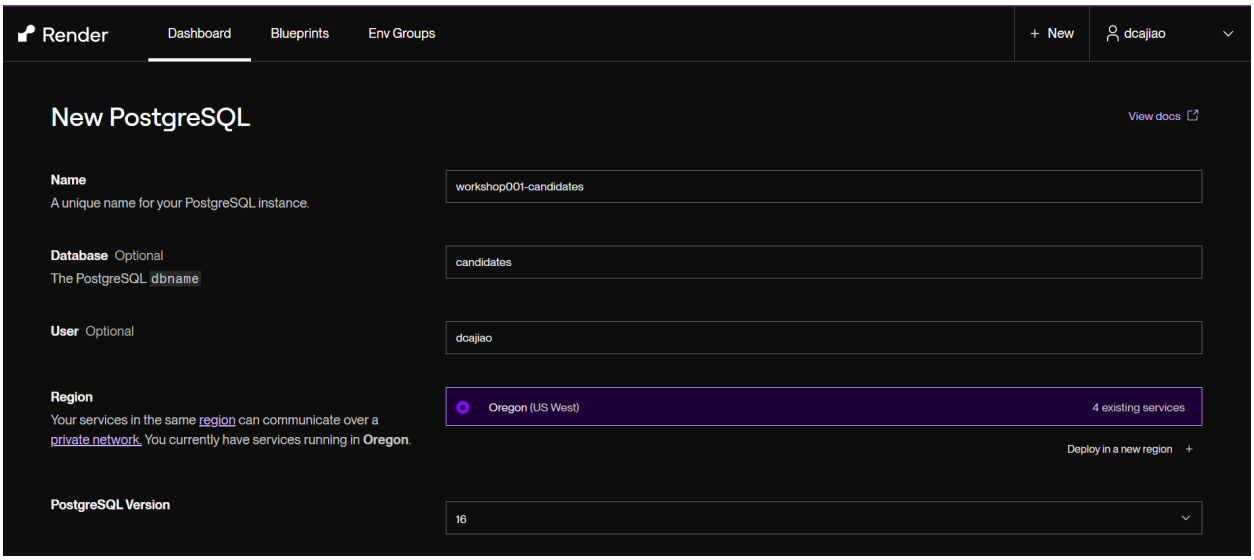
1. Data Loading (`00_data_load.ipynb`)

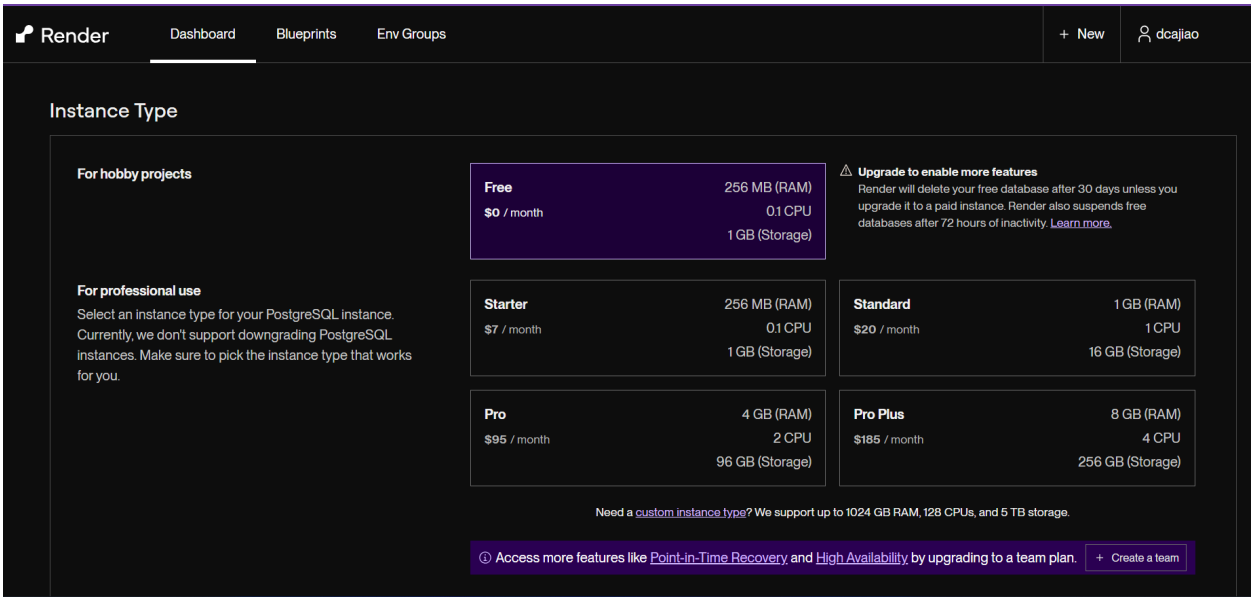
What was done?

The first step in the ETL pipeline involved loading the raw candidate data into a PostgreSQL database. The raw data was provided in CSV format and needed to be uploaded to a relational database to facilitate structured queries and further analysis.

- **Database Setup:** A connection was established to a free PostgreSQL instance deployed on **Render**. This cloud instance enabled efficient and scalable remote access.

- **Implementation Documentation:** Detailed public documentation was written to guide other users in creating and deploying databases on Render. This document is an essential resource for those looking to replicate the development environment.
- **Data Ingestion:** The raw data was read using Pandas and inserted into the corresponding tables in the database. The schema for these tables was previously defined in the `schema.sql` file, which was applied before the data load using `seed_data.sql`.





2. Data Exploration (`01_data_exploration.ipynb`)

What was done?

The second step involved exploring the data to understand its structure, content, and potential issues. This step was crucial to identifying the necessary transformations to clean and prepare the data for analysis.

- **Initial Query:** SQL queries were executed directly on the database to obtain the dataframe and process it further from the local environment.
- **Exploratory Data Analysis (EDA):** An EDA was performed using Pandas to generate descriptive statistics, identify outliers, and visualize distributions. This helped to understand the overall shape and characteristics of the dataset.

Key Findings

1. The dataset has 50k records and 10 columns.
2. We have only 3 numeric columns (`YOE`, `Code Challenge Score`, `Technical Interview Score`).
3. We have 2 categorical columns (`Seniority`, `Technology`).
4. The `Application Date` column is not in the correct format.
5. We have no null records.
6. We have no duplicate records.
7. The data have been randomly generated and therefore have nearly uniform distributions.
8. The distribution of Seniority is not very relevant since it is in almost equal proportions.
9. Regarding the distribution of Technology, we have an almost uniform distribution, except for the categories 'DevOps' and 'Game Development'.
10. The data are from a continuous period of 4 years.
11. The temporal range is from 2018 to 2022.
12. In 2022, we only have records up to month 7 (which gives us a continuous range of 3.5 years).

---

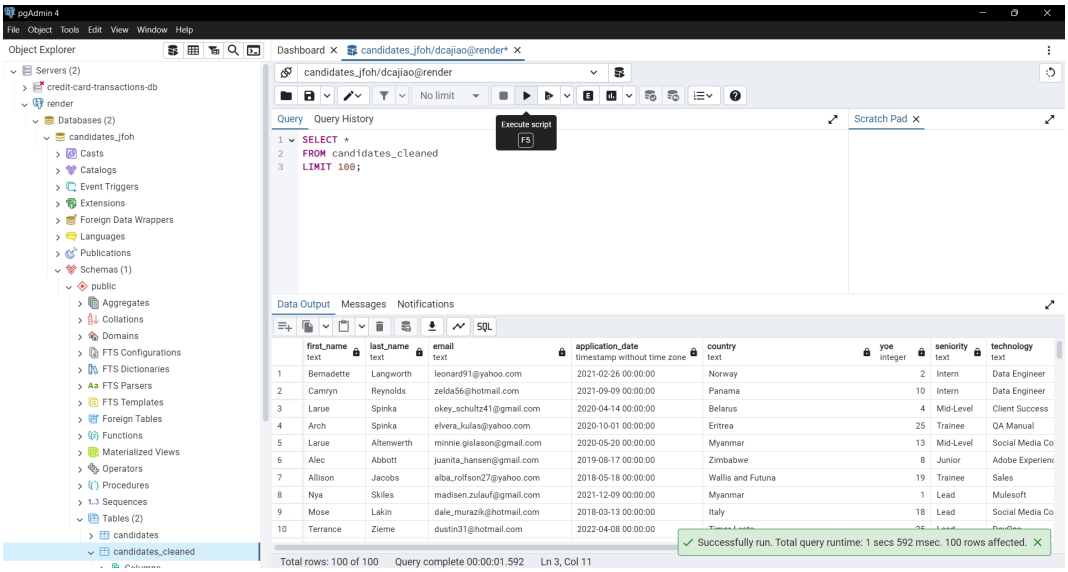
3. Data Cleaning (`02_data_cleaning.ipynb`)

What was done?

Data cleaning was an essential step to ensure the quality and usability of the dataset for analysis. The process involved the following steps:

- **Standardization of column names**
- **Checking for inconsistencies in data types**
- **Identifying which records will be recruited based on scores**

Afterward, we proceeded with the Database Update: After cleaning, the data was updated in the database in the `candidates_cleaned` table, ensuring that the database reflected the cleanest and most prepared version of the data.



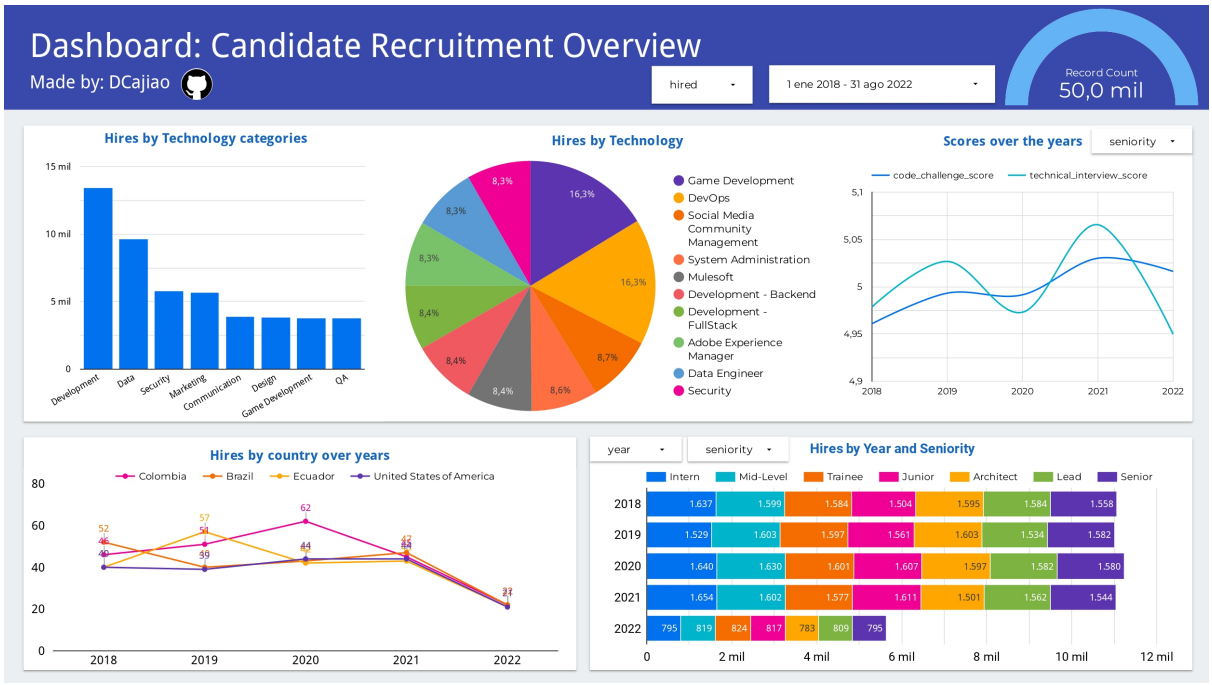
---

4. Data Visualization (`03_visualization.ipynb`)

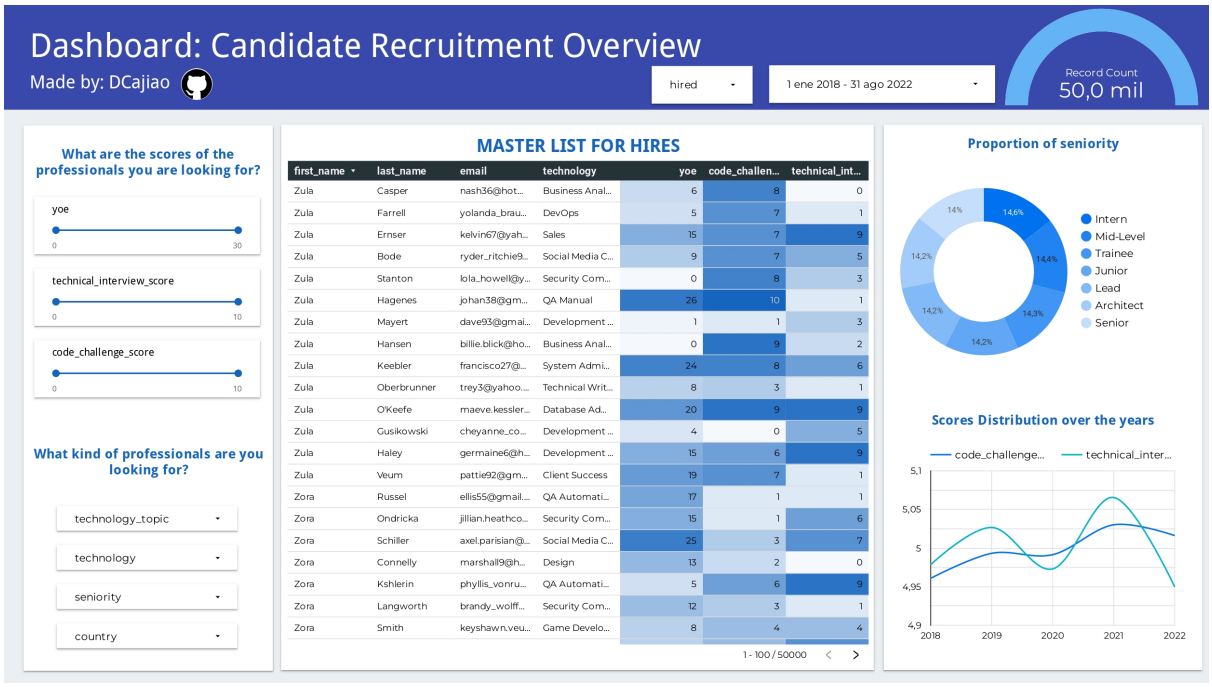
What was done?

The final step involved creating visualizations to meaningfully present the cleaned and processed data. These visualizations were designed to uncover trends, patterns, and insights that could guide decision-making.

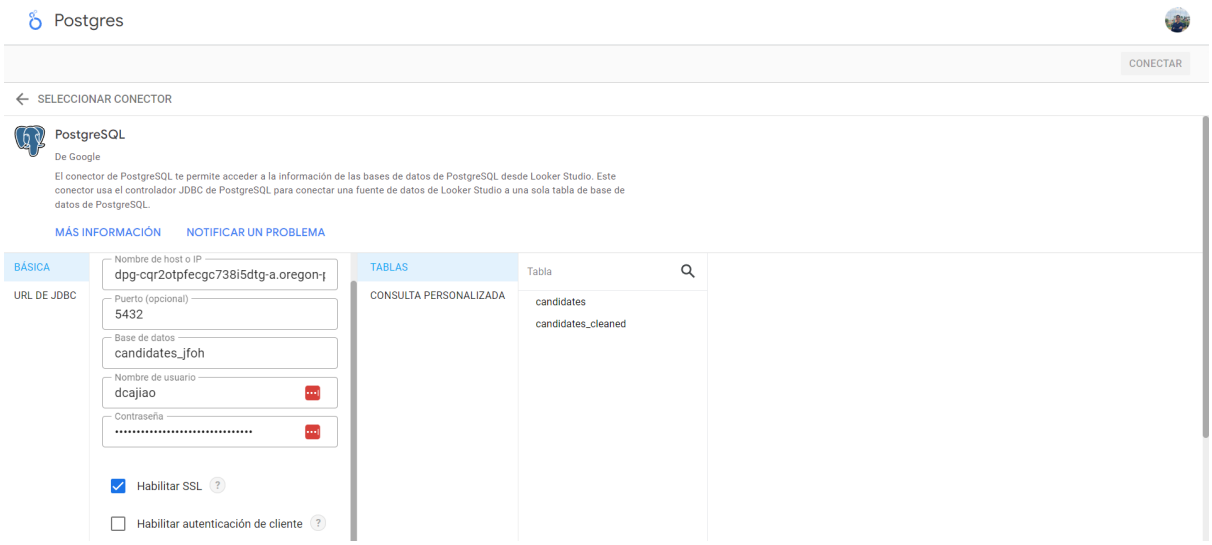
- **Creation of Visualizations:** Matplotlib and Seaborn were used to create a series of visualizations, including bar charts, histograms, and scatter plots, which showed the distribution of candidates based on various metrics.
- **Enhancement of Charts in Looker:** These visualizations were later enhanced in **Looker Studio**, where a more interactive and visually appealing dashboard was built.



- **Development of a Recruiter Dashboard:** In addition to the requested dashboard, a specific dashboard for recruiters was created, allowing users to search for detailed information on candidates based on various criteria.



- **Automatic Connection with the Cloud Database:** The advantage of using a cloud database, such as the instance on Render, allowed Looker to connect automatically, facilitating real-time updates to the dashboards as data was processed.



5. Additional Relevant Aspects

PySQLSchema (`pysqlschema.py`)

One of the most notable components of this project was the use of `pysqlschema.py`, a custom script developed to programmatically manage SQL schemas using Python. This script was essential in ensuring that the database schema remained consistent throughout the project.

- **Schema Management:** `pysqlschema.py` enabled the dynamic creation, validation, and migration of database schemas. This was particularly useful during the data loading and cleaning phases, where the schema needed to be adjusted based on the characteristics of the data.
- **Automation:** The script automated various aspects of schema management, reducing the potential for human error and ensuring that schema changes were systematically applied across the entire database.

Relevance

- **Agile Development:** Thanks to the use of `pysqlschema.py`, the project benefited from an agile development process, where schema changes could be quickly implemented and tested without manual intervention.
- **Consistency and Integrity:** The script played a crucial role in maintaining the consistency and integrity of the database, ensuring that all data transformations adhered to the defined schema.