Tableau to Fabric Query Converter

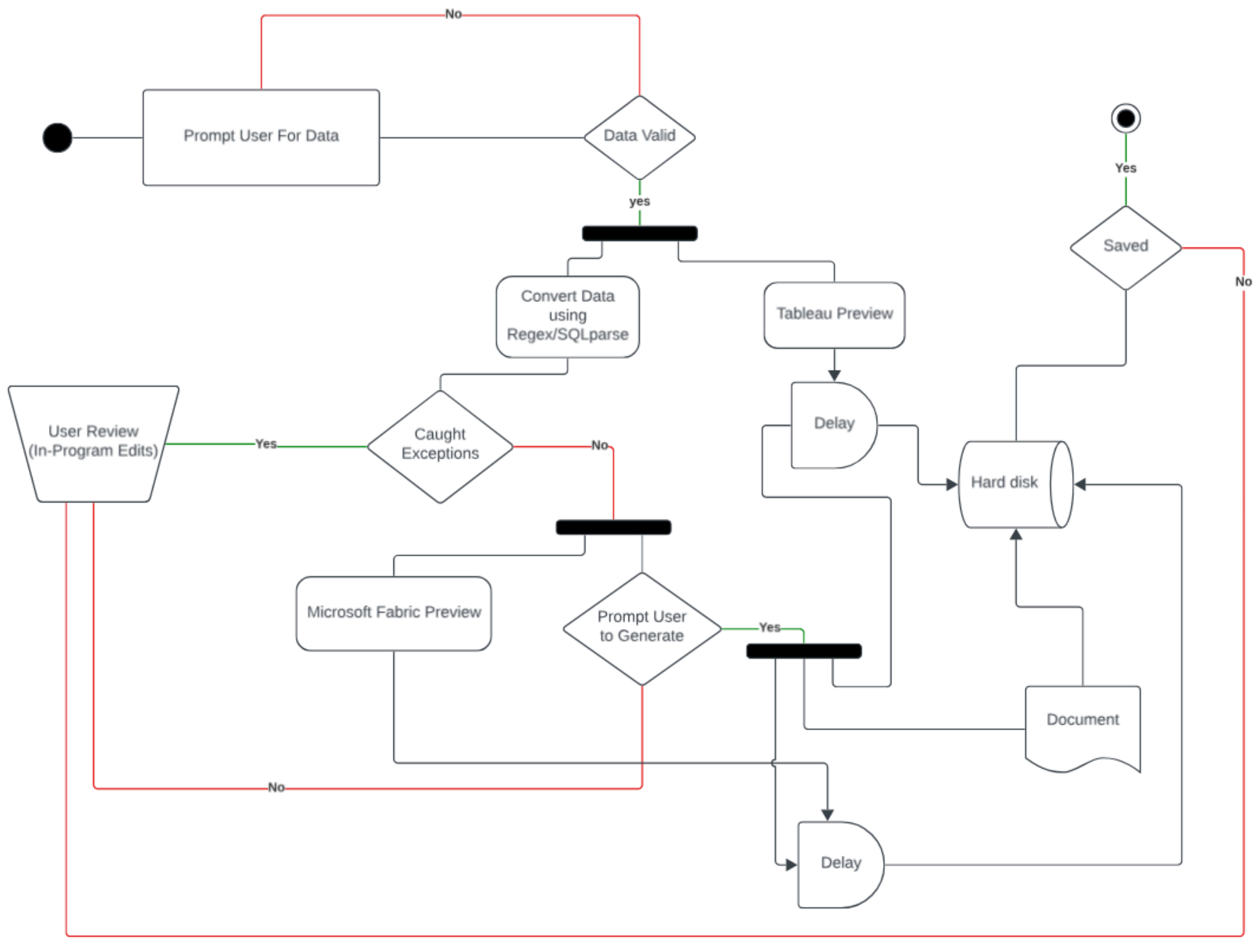
Members:

- Syed Ahmad Shah
- James Sessions

As our organization migrates analytics dashboards from Tableau to Power BI (Microsoft Fabric), teams are spending significant time and resources manually converting Tableau SQL queries to Fabric-compatible SQL. This manual process is repetitive, error-prone, and diverts valuable resources from higher-impact work. There is a clear need for an automated solution to streamline this migration and reduce the burden on our teams.

Our solution to this problem involves creating a Python application that automatically translates Tableau SQL scripts as input into its Microsoft Fabric equivalent. We plan to rely on regex using the **"re"** library, and predefined statement mappings to help facilitate this conversion. In addition to core libraries, we will incorporate **Pandas** to clean and structure input queries for processing and **Matplotlib** to visualize conversion rate, error counts, and performance metrics. Utilizing **Tkinter** for the User Interface, and **Pyinstaller** to package the .py script into a Windows .exe. Finally, the **OS** library to handle file paths and pytest for testing. As the app's intended audience includes those not too accustomed with such tools, we aim at making our application as simple and seamless as possible. We will ensure to include docstrings on each file, explaining the purpose, alongside detailed comments within each block of code, detailing flow and intent of the code.

As the main purpose of our program is to tackle a real engineering problem for a company, we cannot provide the datasets, instead we will create our own dataset to perform testing and verification for the submission. Our Pytest tests will compare our generated output to our verified expected output. We plan on utilizing Hash Maps (dictionaries) and comprehension to store and manipulate the Tableau SQL keywords mapped to their Fabric equivalent. Additionally, utilizing mapping to strip inconsistencies in our Input data (whitespaces, tabs, etc..). Operator Overloading will be necessary for combining SQL queries to form entire statements, with generator functions reading line-by-line without storing all files into memory (which will allow us to easily detect erroneous input lines for user review). A detailed README file will accompany the project, providing setup instructions, usage details, and examples to ensure that users can easily run and test the program. Provided below is a diagram displaying the intended flow of data processing and user actions.



No
Prompt User For Data
Data Valid
yes
Yes
Saved
No
Convert Data using Regex/SQLparse
Tableau Preview
User Review (In-Program Edits)
Yes
Caught Exceptions
No
Delay
Hard disk
Microsoft Fabric Preview
Prompt User to Generate
Yes
Document
No
Delay