**SQL Code Analysis Dashboard**

This repository comprises Java code for a SQL Code Analysis Dashboard created by Syed Muhammad Ali Hassan (SID: 2248952). The dashboard offers functionalities to authenticate users, assess SQL health, view alerts, and apply fixes to SQL files using the SQLFluff linting tool.

**Features**

**Main Class (Main.java)**

***Welcome Message and Options***: Presents a welcome message and provides options to either log in or exit the program.

***Login Handling***: Utilizes the Login class to authenticate users. [Credentials located in userdb.txt within the src folder of the code]

[**Redgate email**: abc@red-gate.com], [**password**: pass]

**Login Class (Login.java)**

***User Authentication***: Reads user credentials, validates them against stored credentials, and authenticates users.

***Login Execution***: Executes the login process, guiding users to the dashboard upon successful login.

**Dashboard Class (Dashboard.java)**

***Options Display***: Presents a menu of dashboard options and handles user selections.

***Functionality Handling***: Guides users to different functionalities—SQL Health Checker, Viewing Alerts, Applying Fixes, and Logout—based on their selections.

**SQLHealthChecker Class (SQLHealthChecker.java)**

***Database Health Display***: Analyzes the health of selected databases by counting linting errors using SQLFluff.

***Error Evaluation***: Calculates error density and health score, presenting the information in a report format.

**ViewAlert Class (ViewAlert.java)**

***Alert Viewing***: Fetches linting alerts from SQL files using SQLFluff and displays them in a formatted table.

***Alert Details***: Allows users to view detailed information about selected alerts.

**Alert Class (Alert.java)**

***Alert Representation***: Represents an alert with its details, including serial number, code, description, file path, line number, and line position.

***Alert Information Display***: Provides methods for displaying alert information and managing alert data.

**ViewAndApplyFixes Class (ViewAndApplyFixes.java)**

***Alert Processing***: Fetches linting alerts from SQL files and provides options to view automated fixes or manually edit the code.

***Auto Fix Application***: Offers functionality to apply suggested fixes automatically to SQL files.

**Libraries and Tools Used**

**Gson Library [com.google.gson.\*]:**

Used for parsing JSON data obtained from SQLFluff's output.

**Other Libraries used:**

* java.io.\*
* java.nio.file.\*
* java.util.\*
* java.util.stream.Collectors
* java.util.stream.Stream

**Dialect and SQLFluff:**

The code interacts with SQLFluff, a SQL linting tool, using the 'ansi' dialect for detecting linting errors and suggested fixes.

**Python:**

Invoked via ProcessBuilder to execute SQLFluff commands in the background.

[Ensure Python is installed on your system]

**Usage**

**To use this SQL Code Analysis Dashboard:**

* Ensure you have Java installed on your system.
* Make sure Python is installed.
* Set up and understand how to use the Gson library.
* Configure the required dependencies and SQLFluff tool.
* Set the filepath for the database directory according to your system. Databases will be provided in the "Databases" folder inside src folder.
* Compile and run the Main.java file to initiate the dashboard.
* Follow the prompts to authenticate and utilize the various functionalities.

**Note**

* Ensure proper configurations for SQLFluff and file access permissions are set up before running the program.
* Two databases [Database1.sql (good SQL code) and Database2.sql (poor SQL code)] are already provided in the "Databases" folder inside src.
* This code, while functional, may not be perfect. There is room for enhancement and refinement to improve its efficiency and expand its capabilities. Your contributions and suggestions to enhance this project are welcomed and encouraged.
* If you face any issues feel free to contact me.