Technical Running Document for the Provided Python Code

Overview

This Python script is designed to generate and format a report in Excel based on data fetched from a SQL Server database. The report includes several sections, each focusing on specific demographics and statistics related to initial referrals within a school year.

The order of running:

Report_1_4.py

Report_5_7.py

Report_8.py

Report_8a.py

Report_9.py

Report_10.py

Report_11.py

Report_12.py

Report_13.py

Report_14.py

Report_15.py

Report_15a.py

Report_16.py

Report_17.py

Prerequisites

1. **Python Environment** : Ensure that Python 3.x is installed on your machine.
2. **Required Libraries** :

- `openpyxl`: For working with Excel files.
- `pyodbc`: For connecting to and querying a SQL Server database.

  Install the required libraries using pip:

```
pip install openpyxl pyodbc
```

Technical Running Document for the Provided Python Code

Overview

This Python script is designed to generate and format a report in Excel based on data fetched from a SQL Server database. The report includes several sections, each focusing on specific demographics and statistics related to initial referrals within a school year.

Prerequisites

1. **Python Environment** : Ensure that Python 3.x is installed on your machine.
2. **Required Libraries** :

- `openpyxl`: For working with Excel files.
- `pyodbc`: For connecting to and querying a SQL Server database.

  Install the required libraries using pip:

```
pip install openpyxl pyodbc
```

1. **Database Access** : The script connects to a SQL Server database (`SEO_MART`) and requires access to execute stored procedures and fetch data.
2. **Excel Template** : The script expects an Excel template located at `C:\Users\Ywang36\OneDrive - NYCDOE\Desktop\CityCouncil\Non-Redacted Annual Special Education Data Report.xlsx`. Ensure this file exists, or modify the script to point to the correct path.

Script Structure

1. **Initialization** :

- The script starts by importing necessary libraries and defining the `Solution` class.
- The `Solution` class contains methods for formatting headers, creating an Excel report template, connecting to a database, fetching data, and writing data to the Excel sheet.

1. **Key Methods** :

- `__init__` : Initializes the object with a school year (`SY 2022-23`).
- `get_column_index_from_string` : Converts an Excel column letter to an index.
- `format_header` : Formats the headers in the Excel sheet.

- `create_excel_report_template` : Creates a new Excel worksheet and applies initial formatting.
- `create_border_styles` : Defines various border styles used in the Excel sheet.
- `connect_to_database` : Establishes a connection to the SQL Server database and executes a stored procedure.
- `fetch_data_by_*` : Various methods for fetching data categorized by different demographics (e.g., race, district, meal status).
- `write_data_to_excel` : Writes fetched data into the Excel worksheet, applying formatting and alignment.

1. **Main Workflow** :

- `main_Reports_1_4_Initials` : The core method orchestrating the report generation process:
    1. Creates the Excel report template with titles and subtitles.
    2. Connects to the database and fetches data for each demographic category.
    3. Writes the fetched data into the appropriate sections of the Excel worksheet.
    4. Applies necessary formatting to the data.
    5. Saves the completed Excel file to the specified path.
    6. Closes the database connection.

Running the Script

1. **Modify Parameters** :

- Update the `save_path` in the `main_Reports_1_4_Initials` method if needed to reflect the correct save location for the output Excel file.

1. **Execution** :

- Open a terminal or command prompt.
- Navigate to the directory containing the script.
- Run the script using:

    ```
    pip install openpyxl pyodbc
    ```

    Replace `script_name.py` with the actual filename of the Python script.

1. **Expected Output** :

- The script will create a new Excel sheet titled "Reports 1-4 = Initials" in the specified Excel file.
- It will populate this sheet with formatted data for different categories (district, race/ethnicity, meal status, etc.).
- The final Excel file will be saved to `C:\Users\Ywang36\OneDrive - NYCDOE\Desktop\CityCouncil\Non-Redacted Annual Special Education Data Report.xlsx`.

Customization

- **Column Adjustments** : Modify `columns` and `column_letters` in the `format_header` method if the structure of the report changes.
- **Database Queries** : Adjust SQL queries in the `fetch_data_by_*` methods if the database schema or requirements change.
- **Additional Formatting** : Further customize the `write_data_to_excel` method to handle different data types or specific formatting needs.

Debugging and Logging

- Print statements are included in some methods (e.g., `print(col + cell_number)` in `format_header`) for debugging purposes. These can be removed or commented out if not needed.
- If the script fails to connect to the database, ensure the `conn_str` in `connect_to_database` is correctly configured.

By following this guide, you should be able to execute the script, generate the Excel report, and customize it according to your requirements.

Technical Running Document for the Provided Python Code

Overview

This Python script is designed to generate and format a report in Excel based on data fetched from a SQL Server database. The report includes several sections, each

focusing on specific demographics and statistics related to initial referrals within a school year.

Prerequisites

1. **Python Environment** : Ensure that Python 3.x is installed on your machine.
2. **Required Libraries** :

- `openpyxl`: For working with Excel files.
- `pyodbc`: For connecting to and querying a SQL Server database.

  Install the required libraries using pip:

```
pip install openpyxl pyodbc
```

1. **Database Access** : The script connects to a SQL Server database (`SEO_MART`) and requires access to execute stored procedures and fetch data.
2. **Excel Template** : The script expects an Excel template located at `C:\Users\Ywang36\OneDrive - NYCDOE\Desktop\CityCouncil\Non-Redacted Annual Special Education Data Report.xlsx`. Ensure this file exists, or modify the script to point to the correct path.

Script Structure

1. **Initialization** :

- The script starts by importing necessary libraries and defining the `Solution` class.
- The `Solution` class contains methods for formatting headers, creating an Excel report template, connecting to a database, fetching data, and writing data to the Excel sheet.

1. **Key Methods** :

- `__init__` : Initializes the object with a school year (`SY 2022-23`).
- `get_column_index_from_string` : Converts an Excel column letter to an index.
- `format_header` : Formats the headers in the Excel sheet.
- `create_excel_report_template` : Creates a new Excel worksheet and applies initial formatting.
- `create_border_styles` : Defines various border styles used in the Excel sheet.

- `connect_to_database` : Establishes a connection to the SQL Server database and executes a stored procedure.
- `fetch_data_by_*` : Various methods for fetching data categorized by different demographics (e.g., race, district, meal status).
- `write_data_to_excel` : Writes fetched data into the Excel worksheet, applying formatting and alignment.

1. **Main Workflow** :

- `main_Reports_1_4_Initials` : The core method orchestrating the report generation process:
    1. Creates the Excel report template with titles and subtitles.
    2. Connects to the database and fetches data for each demographic category.
    3. Writes the fetched data into the appropriate sections of the Excel worksheet.
    4. Applies necessary formatting to the data.
    5. Saves the completed Excel file to the specified path.
    6. Closes the database connection.

Running the Script

1. **Modify Parameters** :

- Update the `save_path` in the `main_Reports_1_4_Initials` method if needed to reflect the correct save location for the output Excel file.

1. **Execution** :

- Open a terminal or command prompt.
- Navigate to the directory containing the script.
- Run the script using:

```
pip install openpyxl pyodbc
```

Replace `script_name.py` with the actual filename of the Python script.

1. **Expected Output** :

- The script will create a new Excel sheet titled "Reports 1-4 = Initials" in the specified Excel file.

- It will populate this sheet with formatted data for different categories (district, race/ethnicity, meal status, etc.).
- The final Excel file will be saved to `C:\Users\Ywang36\OneDrive - NYCDOE\Desktop\CityCouncil\Non-Redacted Annual Special Education Data Report.xlsx`.

Customization

- **Column Adjustments** : Modify `columns` and `column_letters` in the `format_header` method if the structure of the report changes.
- **Database Queries** : Adjust SQL queries in the `fetch_data_by_*` methods if the database schema or requirements change.
- **Additional Formatting** : Further customize the `write_data_to_excel` method to handle different data types or specific formatting needs.

Debugging and Logging

- Print statements are included in some methods (e.g., `print(col + cell_number)` in `format_header`) for debugging purposes. These can be removed or commented out if not needed.
- If the script fails to connect to the database, ensure the `conn_str` in `connect_to_database` is correctly configured.

By following this guide, you should be able to execute the script, generate the Excel report, and customize it according to your requirements.

1. **Database Access** : The script connects to a SQL Server database (`SEO_MART`) and requires access to execute stored procedures and fetch data.
2. **Excel Template** : The script expects an Excel template located at `C:\Users\Ywang36\OneDrive - NYCDOE\Desktop\CityCouncil\Non-Redacted Annual Special Education Data Report.xlsx`. Ensure this file exists, or modify the script to point to the correct path.

Script Structure

1. **Initialization** :

- The script starts by importing necessary libraries and defining the `Solution` class.
- The `Solution` class contains methods for formatting headers, creating an Excel report template, connecting to a database, fetching data, and writing data to the

Excel sheet.

1. **Key Methods** :

- `__init__` : Initializes the object with a school year (`SY 2022-23`).
- `get_column_index_from_string` : Converts an Excel column letter to an index.
- `format_header` : Formats the headers in the Excel sheet.
- `create_excel_report_template` : Creates a new Excel worksheet and applies initial formatting.
- `create_border_styles` : Defines various border styles used in the Excel sheet.
- `connect_to_database` : Establishes a connection to the SQL Server database and executes a stored procedure.
- `fetch_data_by_*` : Various methods for fetching data categorized by different demographics (e.g., race, district, meal status).
- `write_data_to_excel` : Writes fetched data into the Excel worksheet, applying formatting and alignment.

1. **Main Workflow** :

- `main_Reports_1_4_Initials` : The core method orchestrating the report generation process:
    1. Creates the Excel report template with titles and subtitles.
    2. Connects to the database and fetches data for each demographic category.
    3. Writes the fetched data into the appropriate sections of the Excel worksheet.
    4. Applies necessary formatting to the data.
    5. Saves the completed Excel file to the specified path.
    6. Closes the database connection.

Running the Script

1. **Modify Parameters** :

- Update the `save_path` in the `main_Reports_1_4_Initials` method if needed to reflect the correct save location for the output Excel file.

1. **Execution** :

- Open a terminal or command prompt.

- Navigate to the directory containing the script.

- Run the script using:

  ```
  Python script_name.py
  ```

  Replace `script_name.py` with the actual filename of the Python script.

1. **Expected Output** :

- The script will create a new Excel sheet titled "Reports 1-4 = Initials" in the specified Excel file.
- It will populate this sheet with formatted data for different categories (district, race/ethnicity, meal status, etc.).
- The final Excel file will be saved to `C:\Users\Ywang36\OneDrive - NYCDOE\Desktop\CityCouncil\Non-Redacted Annual Special Education Data Report.xlsx`.

Customization

- **Column Adjustments** : Modify `columns` and `column_letters` in the `format_header` method if the structure of the report changes.
- **Database Queries** : Adjust SQL queries in the `fetch_data_by_*` methods if the database schema or requirements change.
- **Additional Formatting** : Further customize the `write_data_to_excel` method to handle different data types or specific formatting needs.

Debugging and Logging

- Print statements are included in some methods (e.g., `print(col + cell_number)` in `format_header`) for debugging purposes. These can be removed or commented out if not needed.
- If the script fails to connect to the database, ensure the `conn_str` in `connect_to_database` is correctly configured.

By following this guide, you should be able to execute the script, generate the Excel report, and customize it according to your requirements.