|  |  |  |
| --- | --- | --- |
| **Sl no** | **Table of Content** | **Page No** |
| 1 | Objective | 2 |
| 2 | Software Requirements | 2 |
| 3 | Python Package Requirements | 3 |
| 4 | Folder Structure for the program | 3 |
| 5 | Running Python File | 4 |
| 6 | Program-Execution | 6 |
| 7 | Reading the source data | 7 |
| 8 | Entity Relationship Diagram (ERD) | 8 |
| 9 | Report generation | 9 |
| 10 | Sending mail | 11 |
| 11 | Client Addition/Deletion | 12 |
| 12 | Software Requirements | 42 |
| 13 | Processed Reports | 19 |
| 14 | End of Document | 20 |
|  |  |  |

**1. Objective:**

* A code-based solution using Python and SQL that will automate the process of importing data, preparing reports, and delivering reports on/at scheduled days/times, to designated stakeholders.

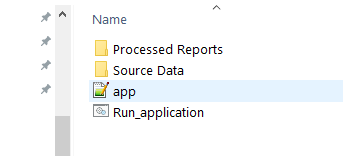
**2. Software Requirements:**

|  |  |
| --- | --- |
|  | Software Requirements |
| ****Operating System**** | Windows 10 64 bit |
| Database | MS SQL SERVER 2019 |
| Database Management Tool | SQL Server Management Studio 18 (SSMS)  https://www.microsoft.com/en-in/sql-server/sql-server-downloads?rtc=1 |
| Python | Python – Version 3.9.13 |
| VS Code | (version 1.53) and above |
| Chromelessdriver | <https://chromedriver.chromium.org/downloads>  (Download driver based on chrome browser version) |

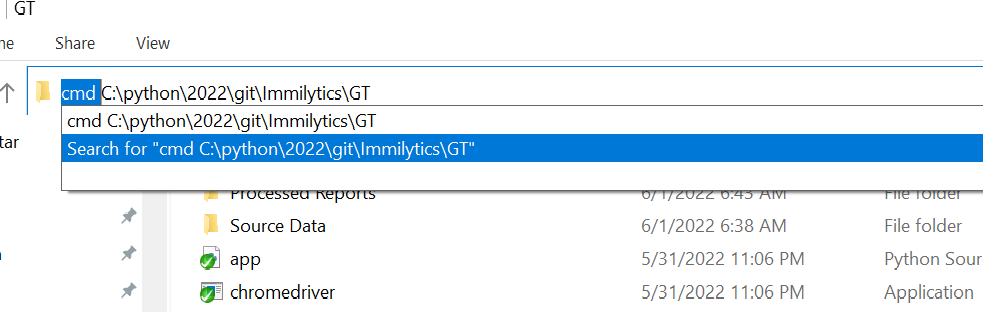
**3.Python Package Requirements:**

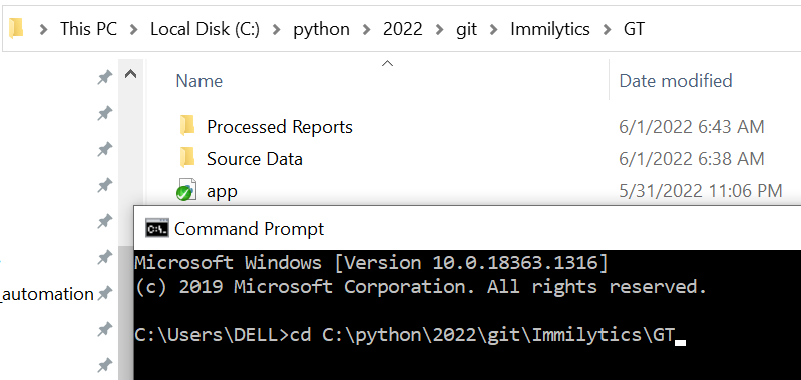
|  |  |
| --- | --- |
|  | List of required External python packages |
| Chardet 3.0.4 | For character detection while reading source excel |
| Openpyxl-0.10 | for excel formatting |
| Pandas-1.4.3 | To retrieve and manipulate source file data |
| Pyodbc-0.32 | To establish connection between Python code and DBMS |
| XlsxWriter-.0.3 | to write data from data frames to external excel files |

* + **Folder Structure for the program flow:**
  + The following folders will be manually created at the path, where the python code will be placed for the ease of flow of program
  + Processed Reports
  + Source Data



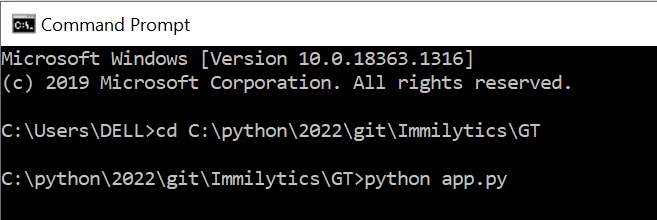
* **Processed Reports folder** - The path that will hold the processed output files
* **Source data folder** - The path that will hold the Source input files which are going to be processed.
* **app.py** - This is the executable Python file that we have to run it in the command prompt.
* **Run\_application.bat** – This is the batch file that is either double-clicked or mapped to any task scheduler to invoke the actual python script **app.py**
* **Running Python File:**
* In Real-working environment the program is supposed to run automatically using any task scheduler app which can trigger the “Run\_application” batch file in our folder structure in any defined time and date.
* Alternatively the program can also be run manually in case of any fails or for any other reasons through manual method.
* To try running the program manually, Go to the respective folder and type **cmd** in the address bar and click enter.
* It will open a command prompt or else you can open command prompt from the run command (Win Key + R) and move to the respective folder by using **cmd command.**





* Then run python **app** file by running following command:

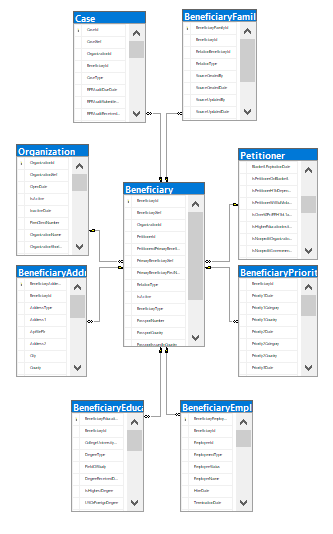
python **app.py**



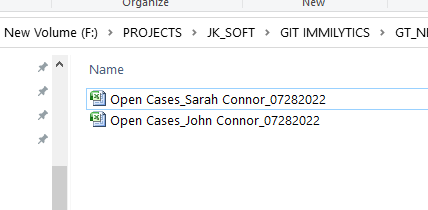
* And Then Press Enter to start Program Execution.
* **Program-Execution:**
* The Program executes in two cycles with two different data sets (source files) and correspondingly creates different set of reports in each cycle.
* **Cycle-one Source files**
* **Reports Automation\_Beneficiary Data\_(client-name)\_mmddyyyy**
* **Reports Automation\_Case Data\_(client-name)\_mmddyyy.**
* **Cycle-one processed reports:**
* **Document Expiration Report\_07282022**
* **Alcon\_Status Report\_07282022**

* **Cycle-two Source files**
* **Reports Automation\_Open Cases\_INSZoom\_07282022**
* **Cycle-two processed reports:**
* **Open Cases \_07282022**
* **Reads the source files for data (Beneficiary File):**
* Thus the acquired source files namely Reports Automation - Beneficiary, Reports Automation - Case Data will be renamed as per the below conventions and will be placed in the source file folder.
* **Reports Automation\_Beneficiary Data\_(client-name)\_mmddyyyy**
* **Reports Automation\_Case Data\_(client-name)\_mmddyyy.**
* Now the code searches and picks for the Beneficiary data file in the source folder that contains the current date in its filename.
* This uses pandas framework to read the content of the file and stores within as data frames
* Now the code reads the number of rows the dataframe has and loops repetitive for ‘n’ number of rows the dataframe holds.
* During each looping it reads the data of all the columns of the each respective rows and assigns to the respective columns as defined by the mappings and pushes/adds the data of each rows to the Beneficiary table as denoted in the respective SQL code.
* The loop ends once feeding data of all the rows in dataframe that is extracted from the source beneficiary file.

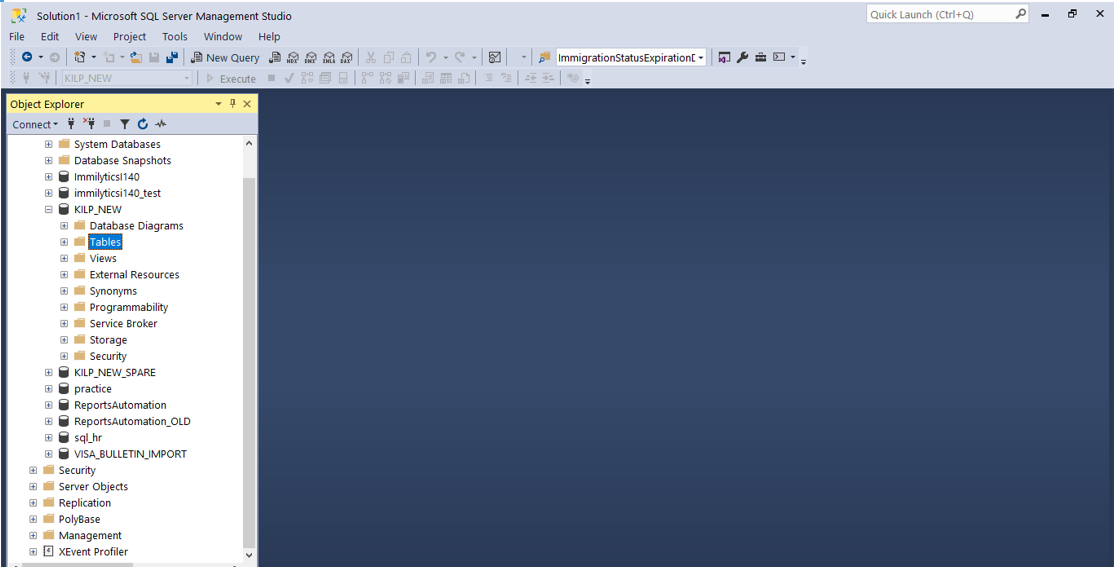
**Entity Relationship Diagram (ERD) For DataBase**

****

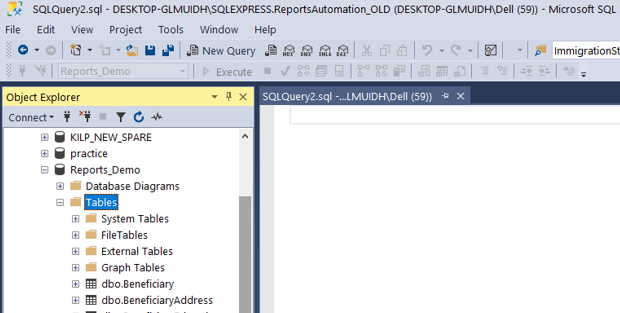
* **Reads the source files for data (Case data File):**
* Once the insertion of data to the DBMS from the beneficiary data is completed the same identical process is restarted for case file as elaborated below
* The code now searches and picks for the Case data file in the source folder that contains the current date in its filename.
* This uses pandas framework to read the content of the file and stores within as data frames.
* Now the code reads the number of rows the dataframe has and loops repetitive for ‘n’ number of rows the the dataframe holds.
* During each looping it reads the data of all the columns of the each respective rows and assigns to the respective columns as defined by the mappings and pushes/adds the data of each rows to Case table as denoted in the respective SQL code with the respective beneficiaryXref as ForeignKey.
* The loop ends once feeding data of all the rows in dataframe that is extracted from the source case file.
* **Report generation:**
* Now once all the data from the beneficiary files are available in the DBMS it’s time to query all the tables as per the requirements to create an output file as required.
* The code queries the Client data table in the database for records, that has a definite non-empty delivery date in it.
* And creates the Output file with the following tabs:
* Open Cases -
* Filed & Pending Cases
* Approved & Closed Cases
* Cases on Hold
* Priority Date Report
* Active Beneficiary List
* Further it proceeds to create Document Expiration Report with the below tab name.
* Doc Exp Report - 8 Months Out
* The sheet enlists all active beneficiaries whose any one of the below criteria did not yet expire a given certain date which is 8months from the day 1 of the month the program is exceuted.
* I797ExpirationDate
* FinalNivDate
* VisaPedDate
* EadExpirationDate
* AdvanceParoleExpirationDate
* Ds2019ExpirationDate
* ReEntryPermitExpirationDate
* GreenCardExpirationDate
* MostRecentPassportExpirationDate
* VisaExpirationDate
* **Sending mail:**
* Once all the reports are ready it’s time to mail the respective clients
* The code checks for availability of the recipient mail id’s either from ClientsDetails table in warehouse or static recipient MAIL-ID’s within the code, when the SEND\_MAIL option is enabled/disabled
* Initially, the code gets the mail ID’s of the primary recipient and secondary recipient from the pre-assigned static Mail-ID’s within the code.
* And only in an account of **SEND\_MAIL\_TO\_CLIENT** is too enabled the code checks for availability of the recipient mail id’s from ClientsDetails table in warehouse.
* An **send\_exp\_report** option is set to provide an option to whether to include are not the “Document Expiration Report” in the mailing attachments.
* Once the report being sent the “**report\_sent\_on**” column of clients details table are updated accordingly.
* **Cycle-2-Execution:**
* Thus now Cycle-1 ends and the program initiates the 2nd Cycle of operation.
* As mentioned earlier in this cycle the code considers only the respective source file (Reports Automation\_Open Cases\_INSZoom\_07282022) as its source and any other files are ignored.
* The same process that is carried out throughout 1st cycle loops for the second time except the only differences are the Source files and the generated reports
* With Successful completion of 2nd cycle of operation a processed report as shown below can be seen in Processed Reports folder.
* The number of reports generated depends on the number of clients who could have matching records in th source file

****

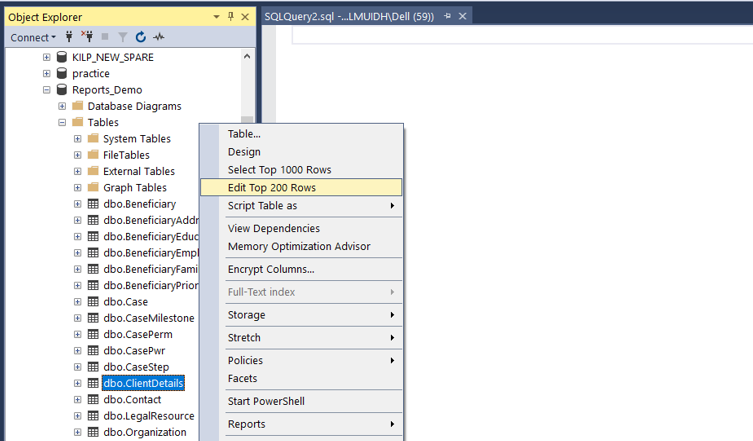
* **Client Addition/Deletion :**
* In Real-time Project execution, client’s mailing list, details are fetched from the warehouse database table, hence it is critical to know client addition/Deletions process.
* The below illustrations explains Addition/Deletions of client data in Warehouse in an non-technical user friendlier way.
* Step1: On your open SQL Server Management Studio 18 from start menu as below



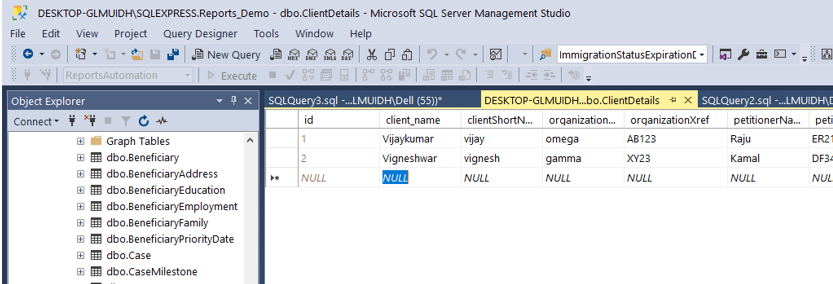
* Step2: On the Object Explorer on your left, you can see a list of created/available databases, In this case we will use **Reports\_Demo** as an example further, double Click on the Reports\_Demo DB to see the list of table it contains.



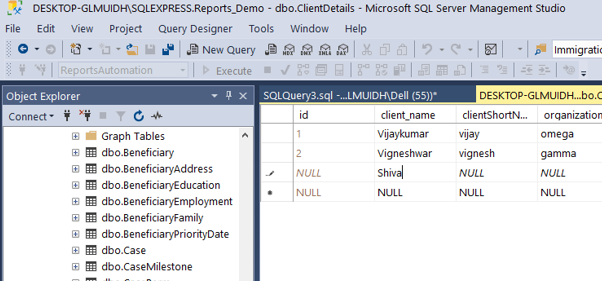
* Step3: In the list of available Table we can see **dbo.ClientDetails** which is going to contain the client details
* Right Click on **dbo.ClientDetails** = > and Click **Edit Top 200 Rows**



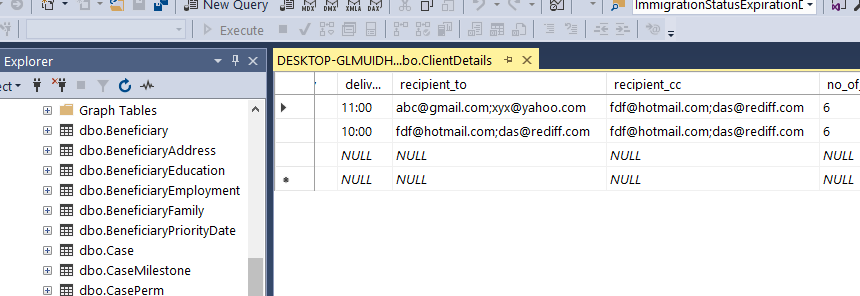
* Step4: Once **Edit Top 200 Rows** option is clicked we can see the below table that contains the existing client data as below



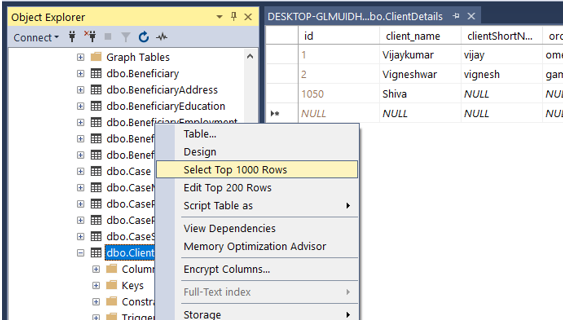
* Step5: By default NULL values can be seen in the last row which the record meant for new records addition.
* You can simply place your cursor there and add values in each columns of the record and keep adding records on the go as shown below.



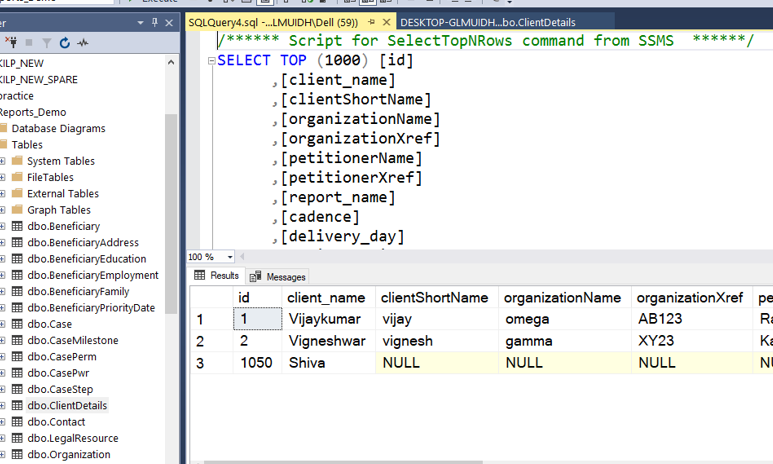
* **NOTE**: In the Mail-ID columns when there is a need to add multiple Emails in a same cell, you can simply use a “;” to separate each mails as shown below.



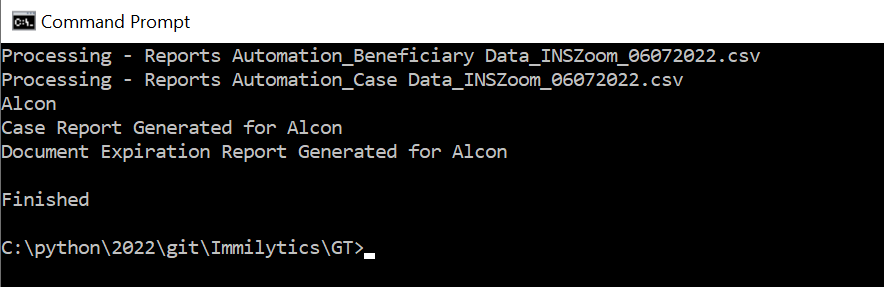
* Step6: Once Entering all the required values in each cell, place your cursor anywhere outside the table now the records gets auto-saved and the active tab can be closed. Now to ensure the addition of record right click on **dbo.ClientDetails** table as did in Step3.
* Now you can see “Select Top 1000 Rows ” option as shown below, Click it.



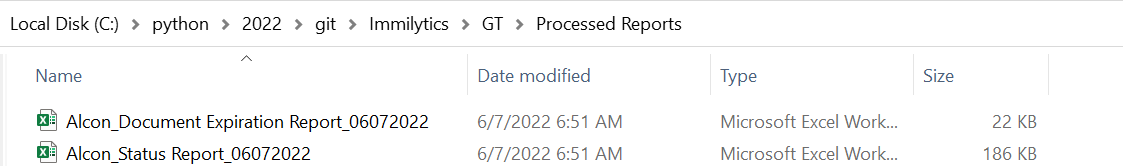
* Now you can see the newly added record as shown below. Please notice that, unlike the image shown below all columns are required to be added in the record for the program to run successfully.



* **Processed Reports:**
* Once Processed the command prompt will show ‘Finished’ status, upon which we shall check reports in the processed report folder.



* The Processed Report will be created at Processed Folder with the respective date of report generation.



---------------------------END OF DOCUMENT------------------------------