Assignment Documentation

Arbour Education - Data Engineer position technical Assessment

GOAL:

1. Automate the process of uploading data to Postgres db, which includes
    a. Fetching data from Explore education statistics
    b. Transforming the data if necessary
    c. Uploading the data as table to Postgres db
    d. Validation of data
2. Python code to filter data from existing table according to user selection

DATASET USED：

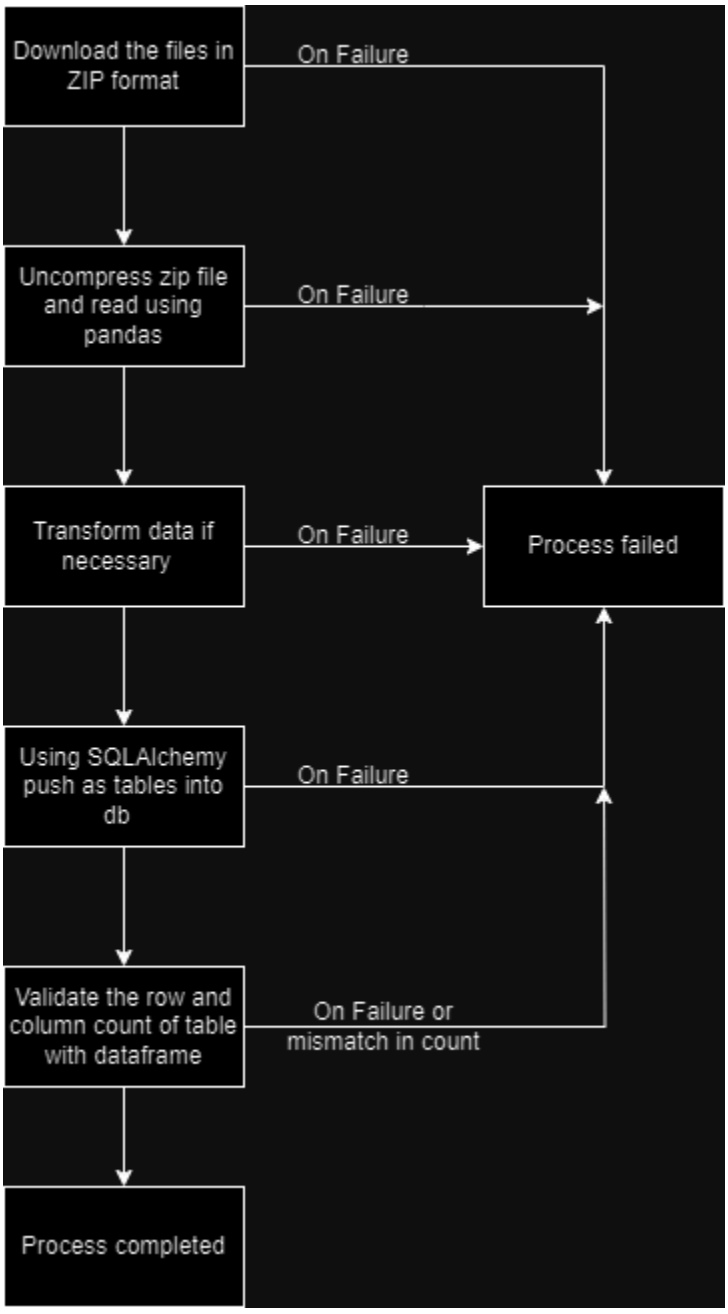
➔ Academic year 2022/23 School capacity
https://explore-education-statistics.service.gov.uk/find-statistics/school-capacity

WORKFLOW:

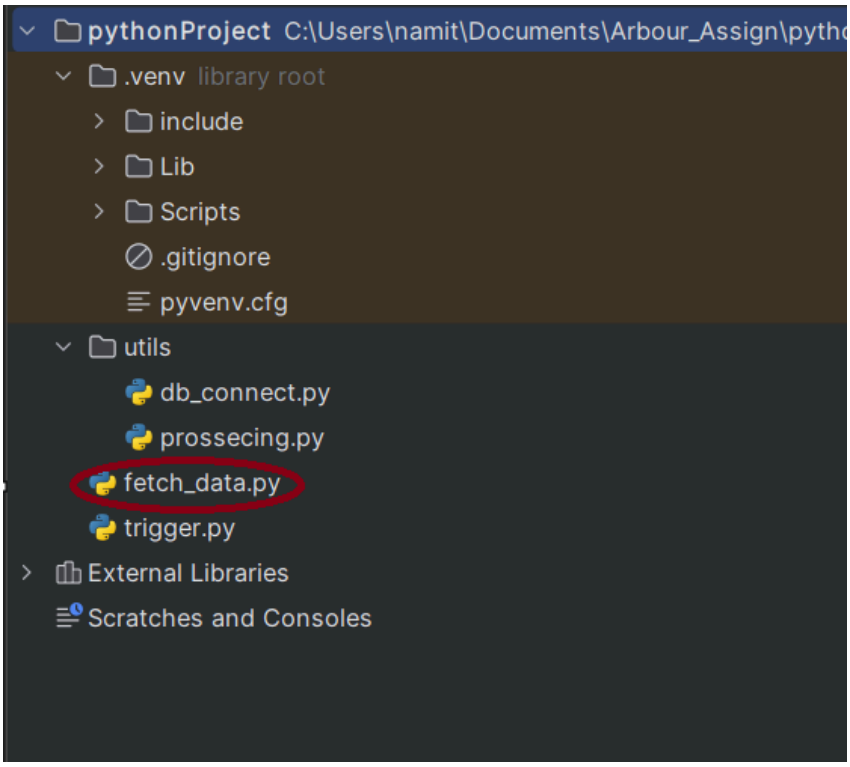
- Automate the process of uploading data to Postgres db
    - **Prerequisite**
        - **URL** from which the zip folder can be downloaded
        - Local directory to store the downloaded files
        - Postgres Db connection details
        - Prior knowledge on column data types

(Scroll down to view code flow)

- **Code Flow**

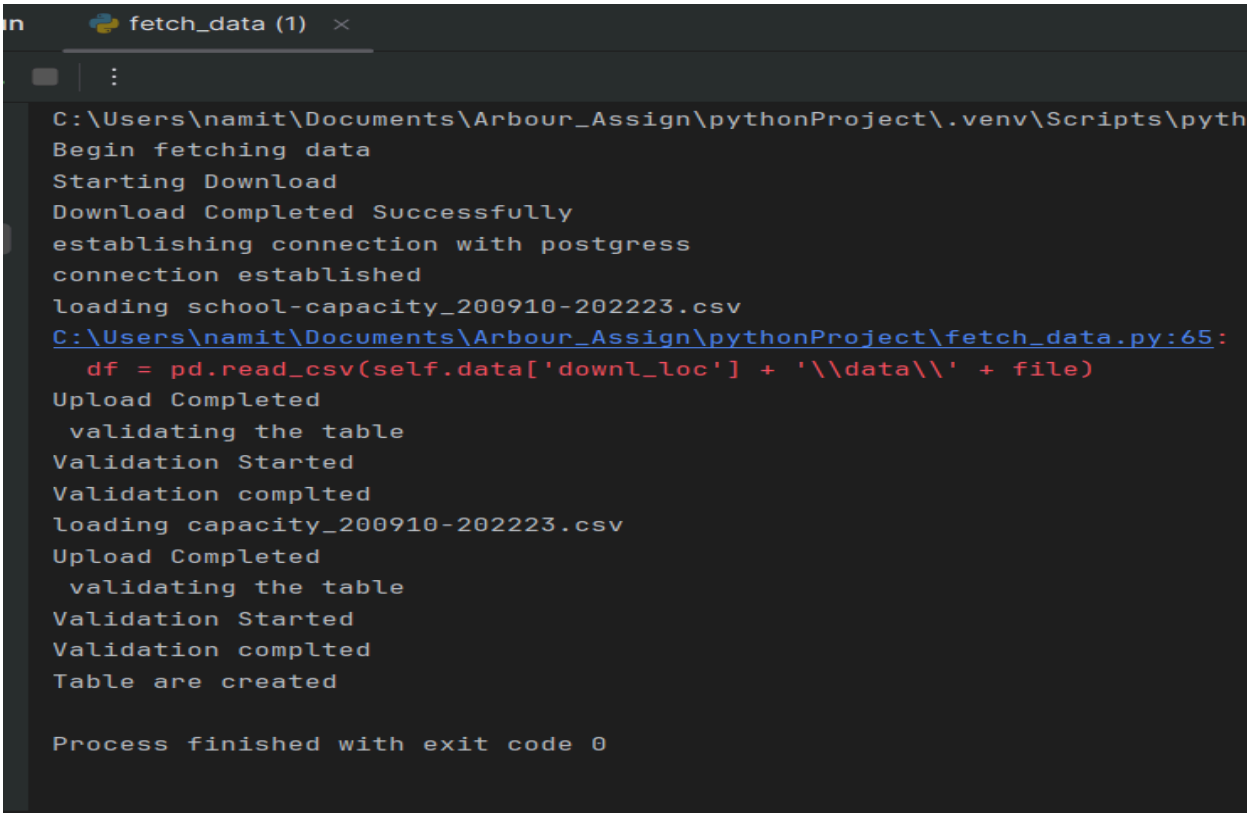

Download the files in ZIP format
On Failure
Uncompress zip file and read using pandas
On Failure
Transform data if necessary
On Failure
Process failed
Using SQLAlchemy push as tables into db
On Failure
Validate the row and column count of table with dataframe
On Failure or mismatch in count
Process completed

- **Application Modules**

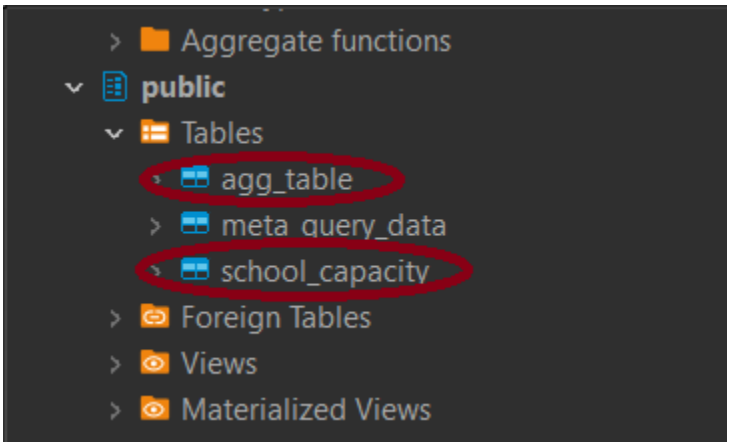


Fetch_data.py is the entry point for the code (no Arguments Required)

- **Code was able to achieve its goal**



- **Table created**



- Python code to filter data from existing table according to user selection
  - **Prerequisite**
    - Table meta_query_data should be created for holding the selection details.The result from code will depend on the selection criteria

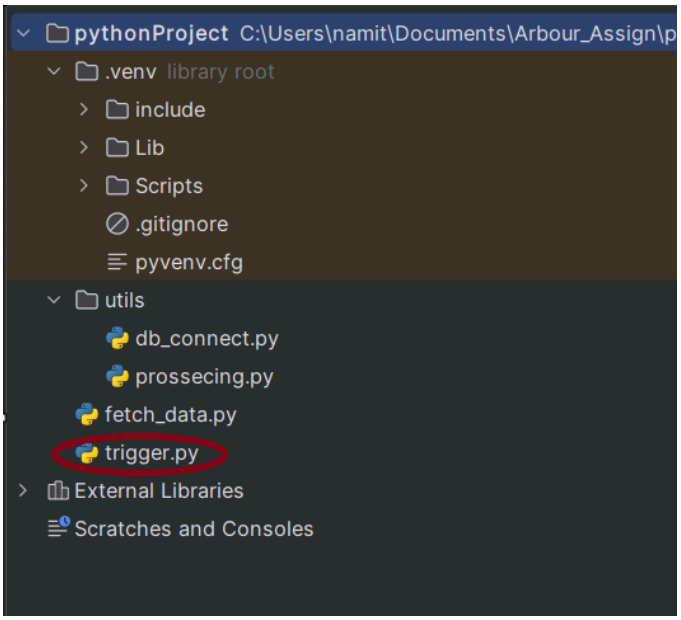
  - **Selection**
    - **Max_Hier** : The max Hierarchy from the given option
      - National
      - Region Code
      - LA code
      - School

        Suppose Max_hier is chosen to be Region Code and in choice option you specify a particular region code eg 'E12000001' Then the result will contain all the data of schools of that particular selection only
    - **Choise** : Specify the value of Max_hier
    - **Agg :** if this is set to be true an additional query will also be created with a result of 39 rows containing total ,primary and secondary school data over the whole time frame.

      Aggregation will be based on the hier of **Max_Hier (**Agg is not possible when max_hier is School as it is the lowest aggregation)
    - **Create table :** if the users want the query result in the form of a table setting this true will fetch the query result in a table

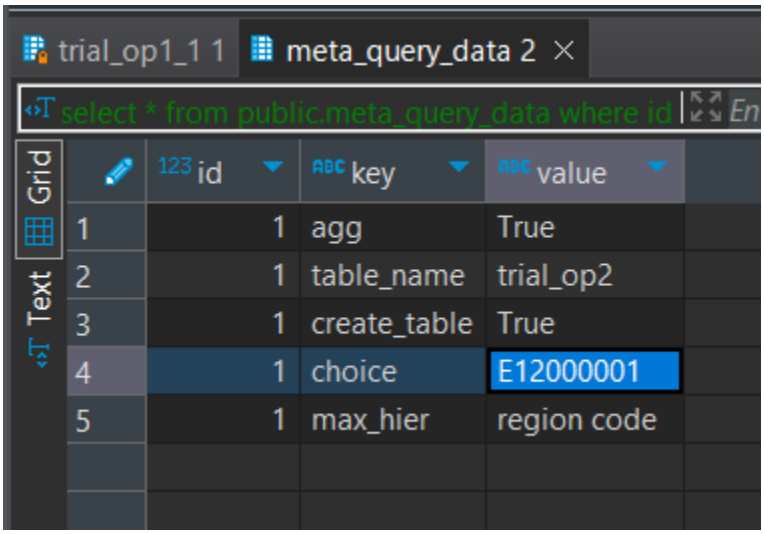
      If create table is set as false the query will be save in a txt file and can be used later
    - **Table name:** if the above option is set to true then table will be created with the name provided here

- **Application Module**

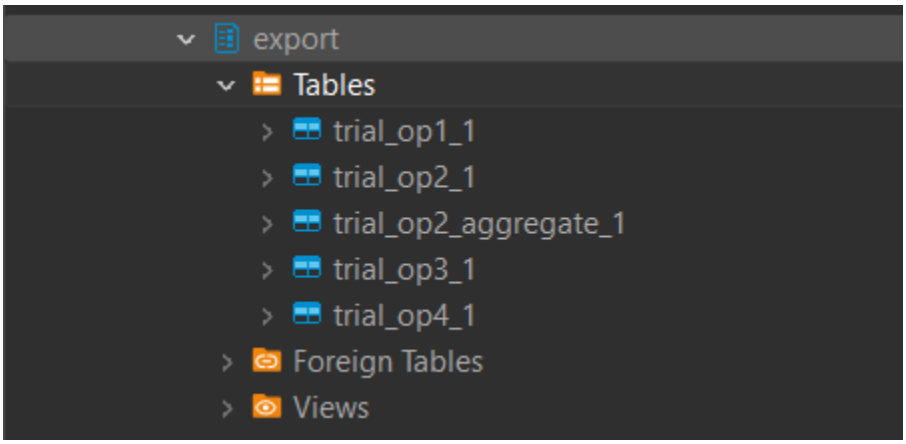


trigger.py is the entrypoint it takes one arg that is task id

- **Metadata table**



- **When create_table set to true**



- **When create_table set to false**