**Here is a document that describe the design, implementation as well as running of the code below.**

The ETL process is written in python to extract data from a csv file and data in rows are transformed to upper case ( first\_name, Last\_name while id is the primary key).

Note: Path to csv file(SRDataEngineerChallenge\_DATASET.csv) must be changed to where file is located at for ETL\_Script.py to run.

A postgres database is provisioned using a docker compose file using postgres 14.1 image. Table Name, ‘employee\_data’ is created with fields;

* ID
* First\_name
* Last\_name
* Email
* Gender
* Ip\_address

To run the code:

* Navigate to folder containing “docker-compose.yml” file and “ETL\_Script.py”, ensure docker is installed on your computer then;
  + Run “docker-compose up –build” in a terminal (this gets postgres image defined in docker-compose file running)
  + Run “python ETL\_Script.py” in a separate terminal. ( running is not necessary as data has been loaded to the table hence, you will get an error [duplicate key value violates unique constraint "employeedata\_pkey"] because table does not allow duplicates.
  + Run “Docker ps” (in a separate terminal to get container id to interact with the shell)
    - Run “docker exec -it [container id] psql -U postgres -d postgres”
      * Opens a postgres interactive shell
    - Run “\dt” to see table (employeedata) created by “ETL\_Script.py”
  + Run “select count (\*) from employeedata;” to count rows of data which matches our csv file then run,
    - select \* from employeedata; to see data loaded.
    - screenshot of table with data loaded
      * A screen shot of a computer

        Description automatically generated