Princess Sumaya University for Technology

Data Engineering Course

Assignment 1

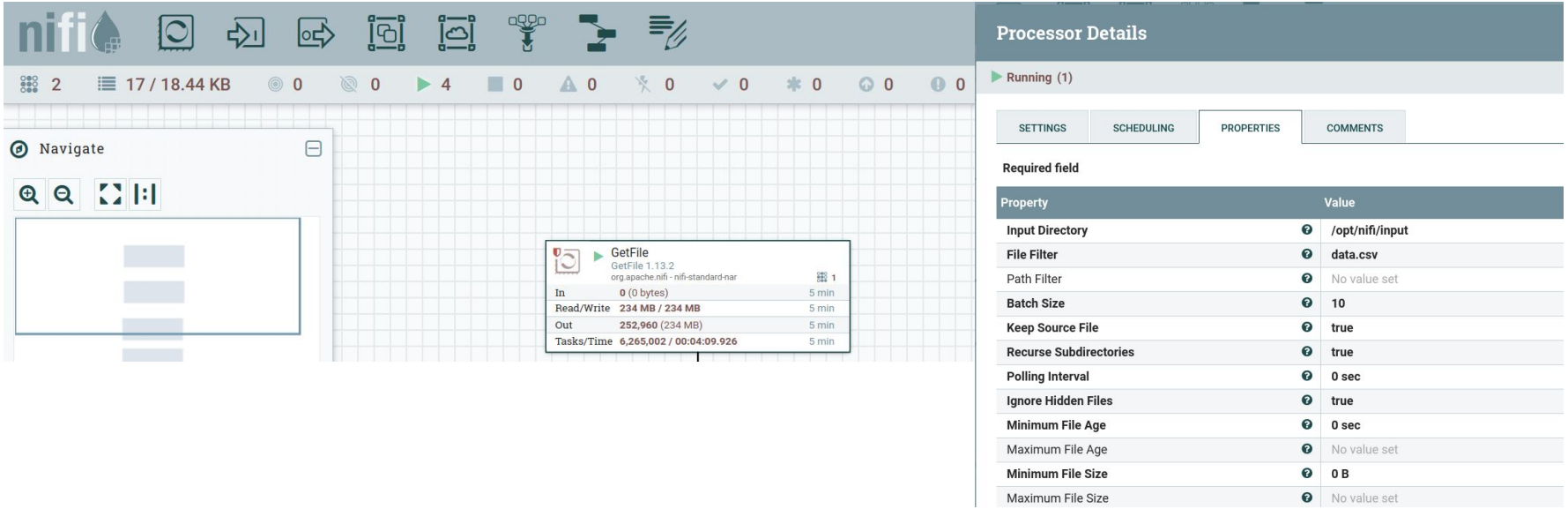
Dr Ibrahim Abu Alhaol

By Waed Alsawarieh , 20208020

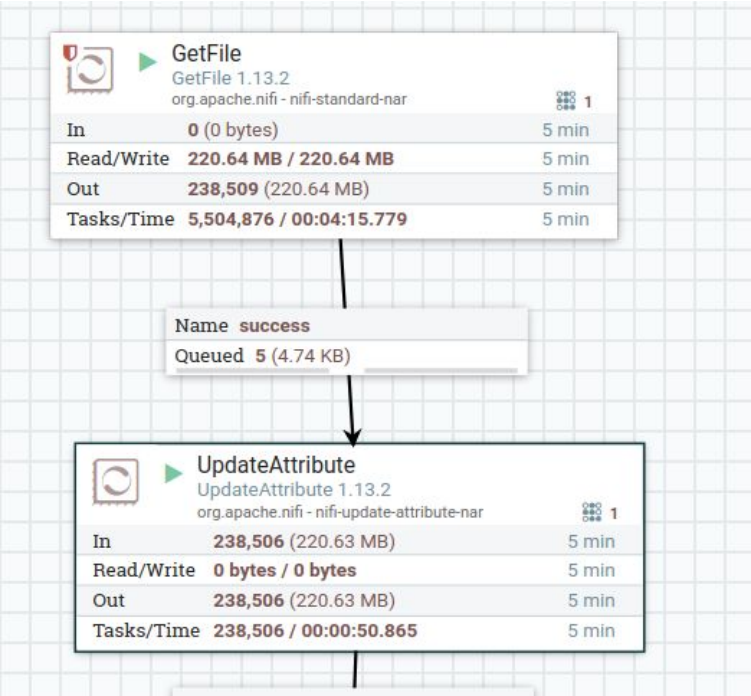
Question 1

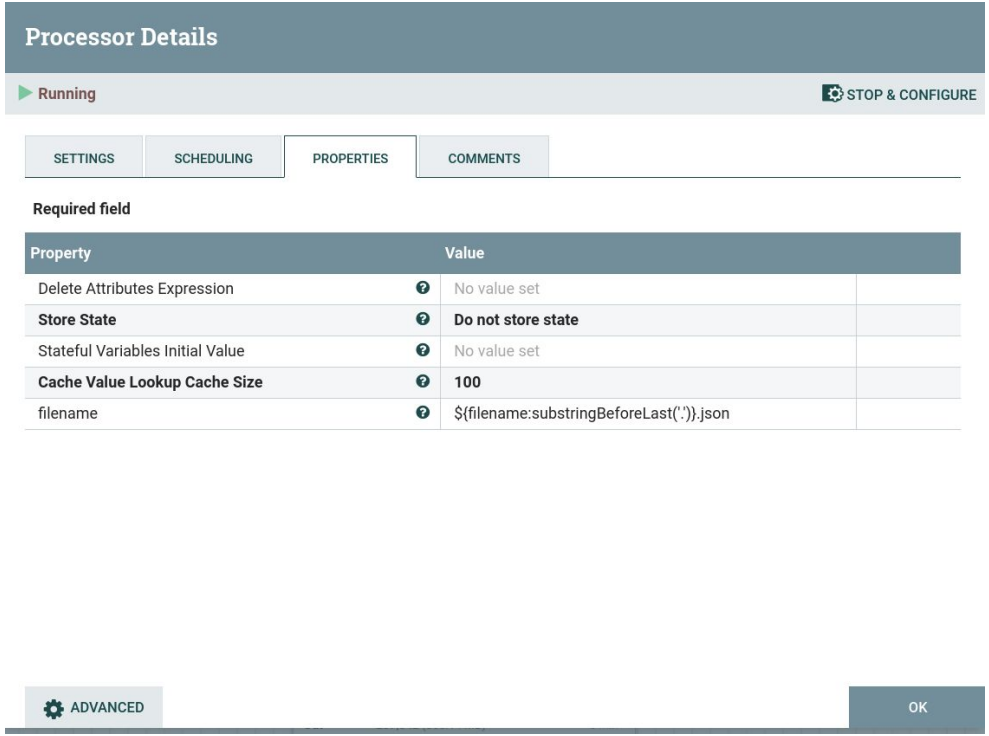
Q1: Provide similar to AirFlow implementation but using NiFi and provide the GitHub repo with all dependencies and detailed REAME.MD and PPT presentation on how to run your workflow.

1.1 Get Data CSV From Input Directory in NIFI Container



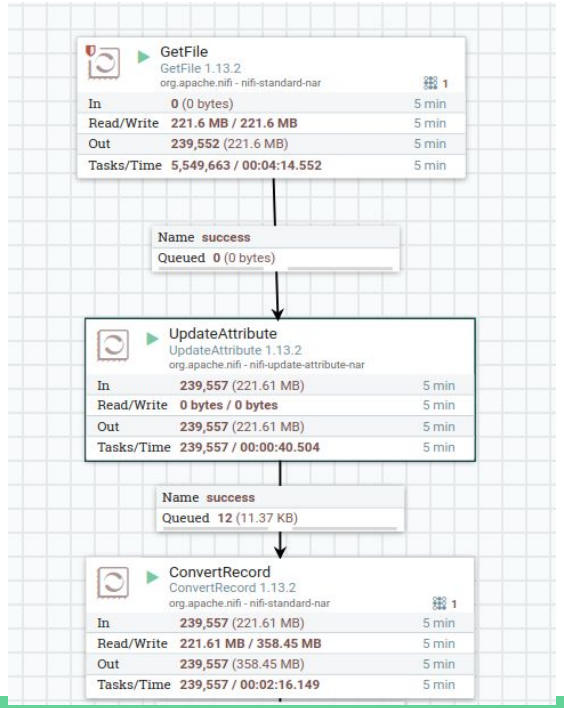
1.2 UpdateAttribute Processor to update attribute from data.csv to data.json

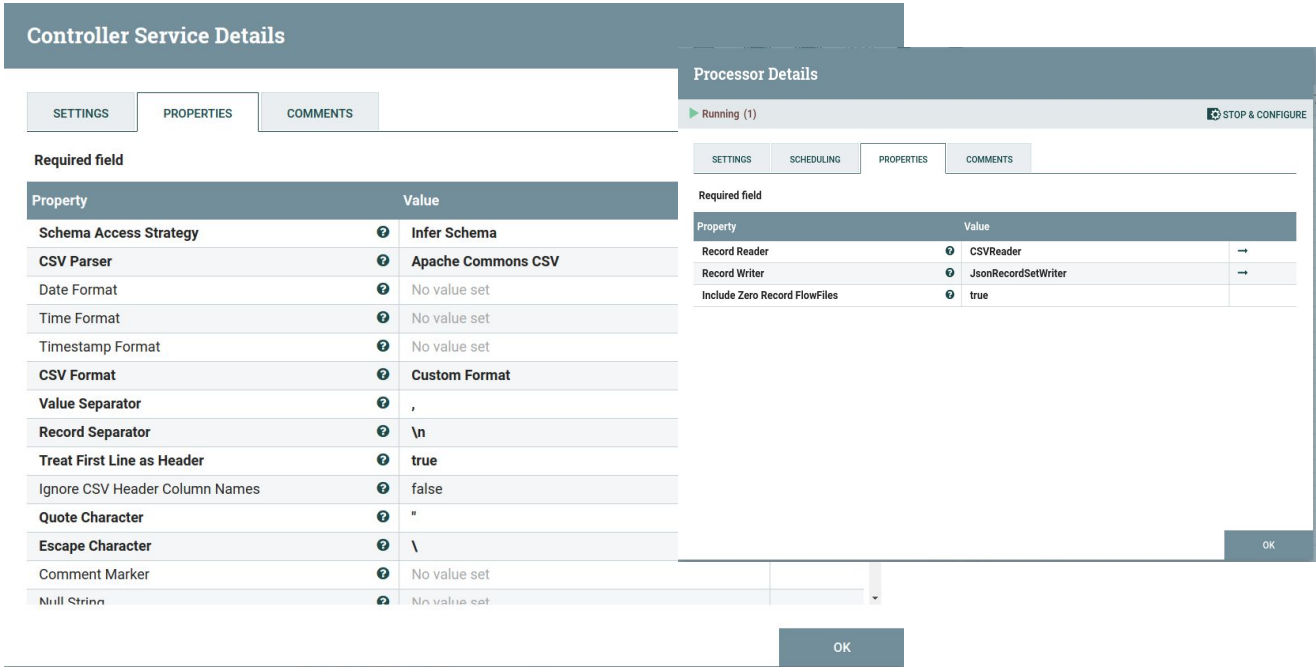




1.3 ConvertRecord Processor

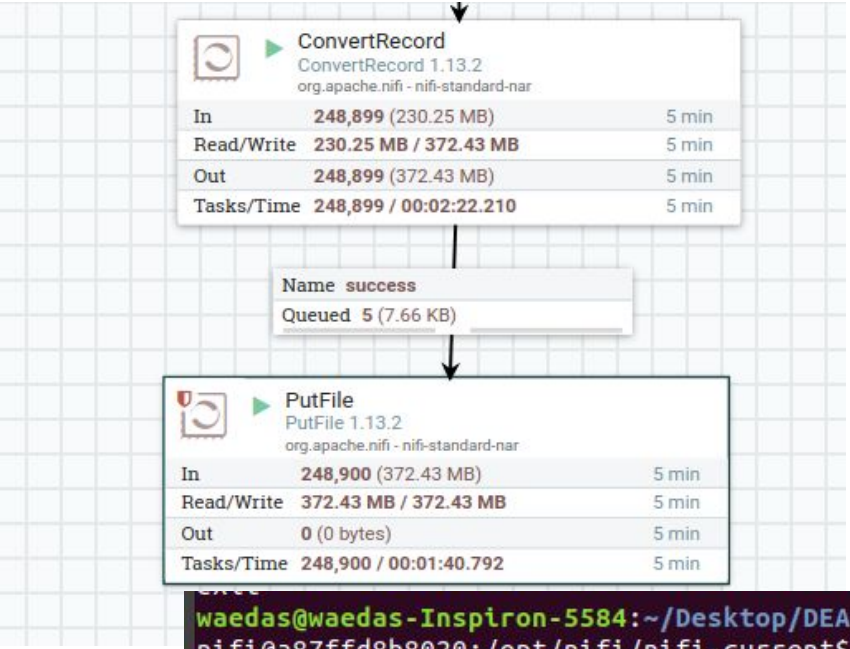
Converts records from one data format to another using configured Record Reader and Record Write Controller Services.

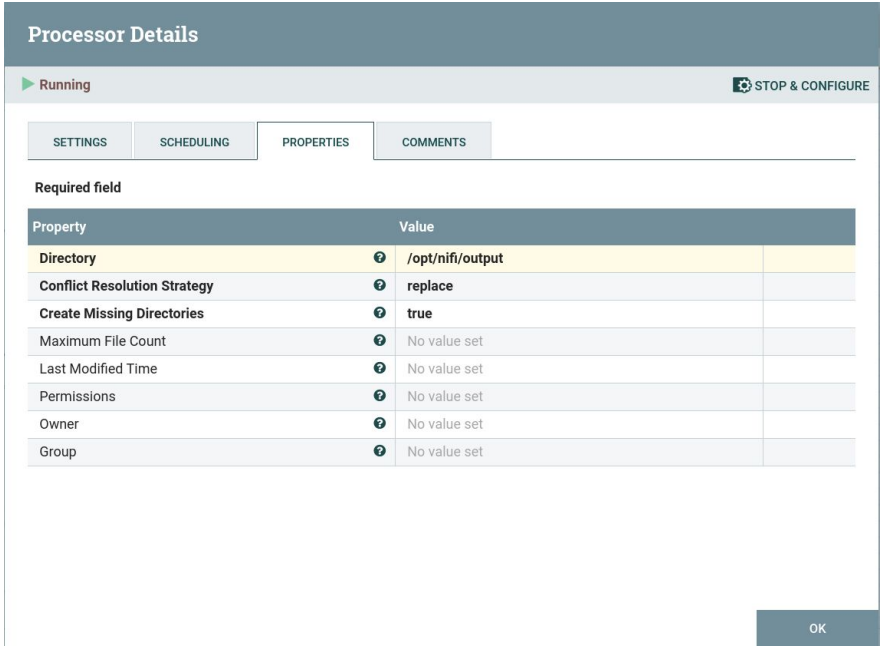




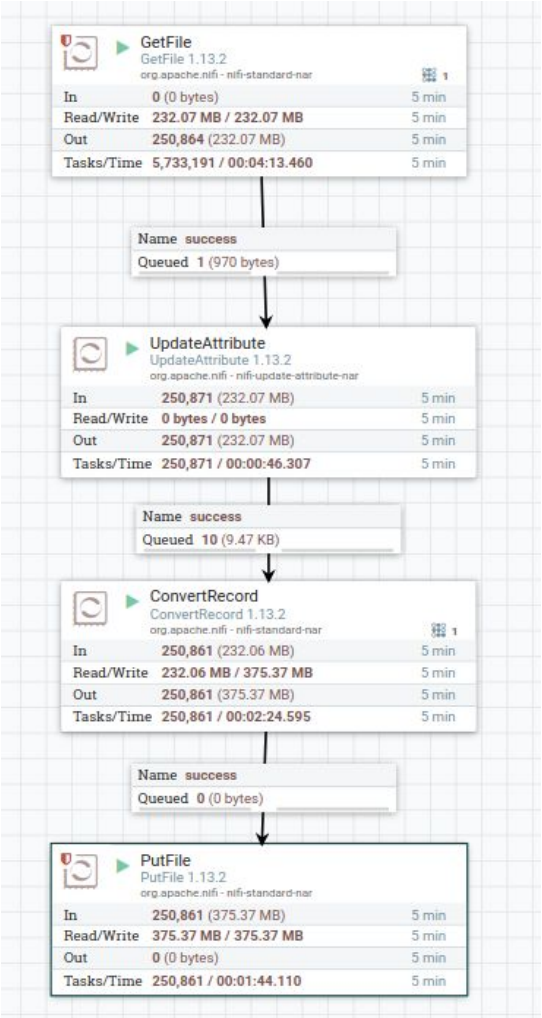
1.4 PutFile Processor

Writes the contents of a FlowFile to the local file system






```
waedas@waedas-Inspiron-5584:~/Desktop/DEAssignment1/nifi$ docker exec -it a87ffd8b8020 bash
nifi@a87ffd8b8020:/opt/nifi/nifi-current$ cd ../output/
nifi@a87ffd8b8020:/opt/nifi/output$ ls
data.json
nifi@a87ffd8b8020:/opt/nifi/output$
```



Validate data.json

Question 2

Q2: Provide Similar to Airflow implementation but with csv file is extracted from Postgresql table and the produced json file is pushed to MongoDB database. Provide Github repo with all dependencies and detailed REAME.MD and PPT presentation how to run your workflow.

Docker Compose File- Services

1. **Apache Airflow**

2. **Postgresql**

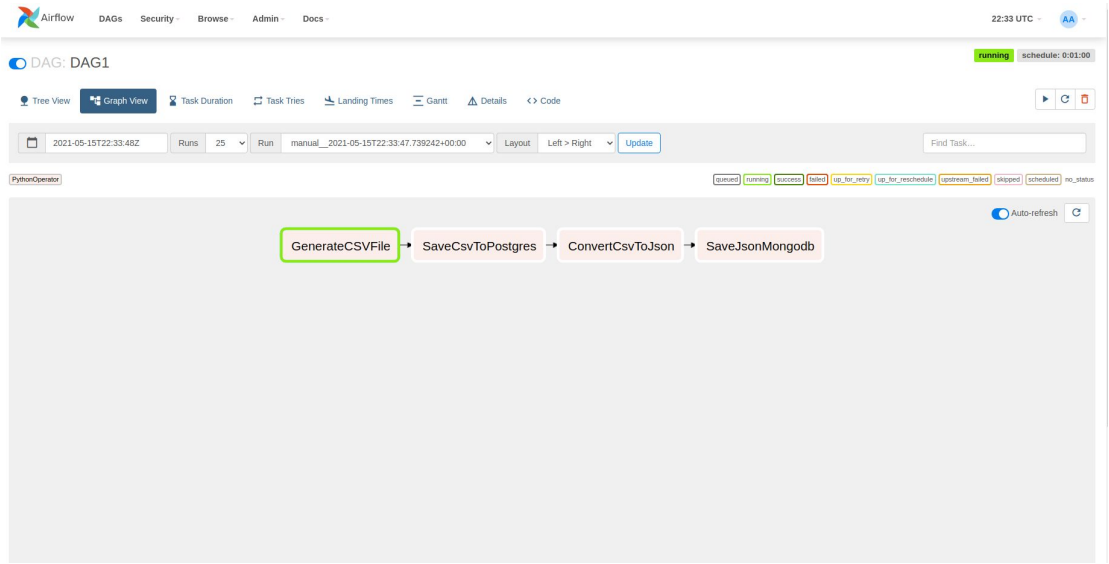
3. **pgAdmin**

4. **mongo-express**: web-based **MongoDB** admin interface.

5. **mongo** : MongoDB document database.

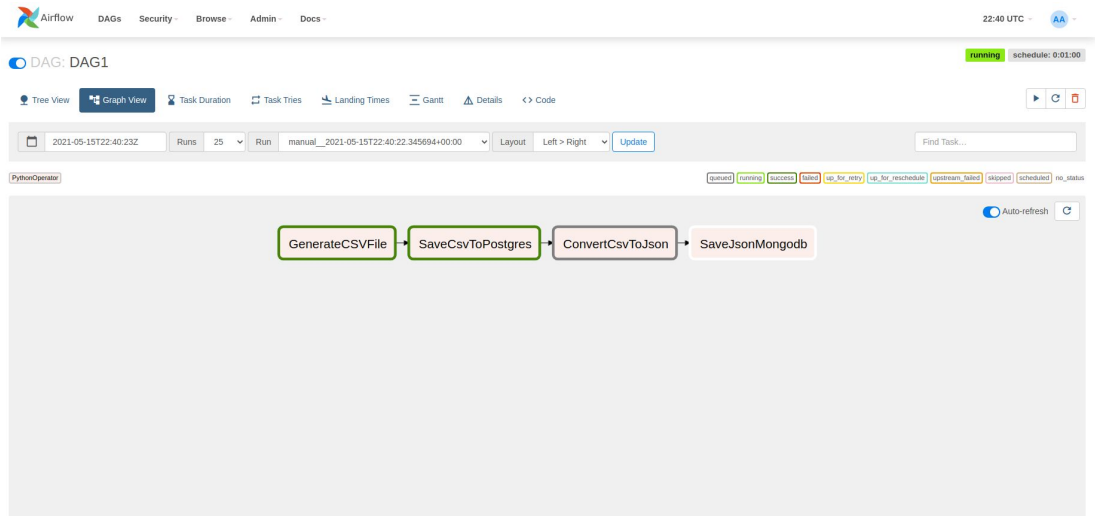
2.1 Generate CSV file from faker

```python
def GenerateCSV():
    output = open('data.csv', 'w')
    fake = Faker()
    header = ['name', 'age', 'street', 'city', 'state', 'zip', 'lng', 'lat']
    mywriter = csv.writer(output)
    mywriter.writerow(header)
    for r in range(10):
        row = [fake.name(), fake.random_int(min=18, max=80, step=1),
               fake.street_address(), fake.city(), fake.state(),
               fake.zipcode(), fake.longitude(), fake.latitude()]
        print(row)
        mywriter.writerow(row)

    output.close()
    DF = pd.read_csv('data.csv')
    DF.to_csv(AIRFLOW_HOME + '/dags/dataframe.csv', index=False)
```

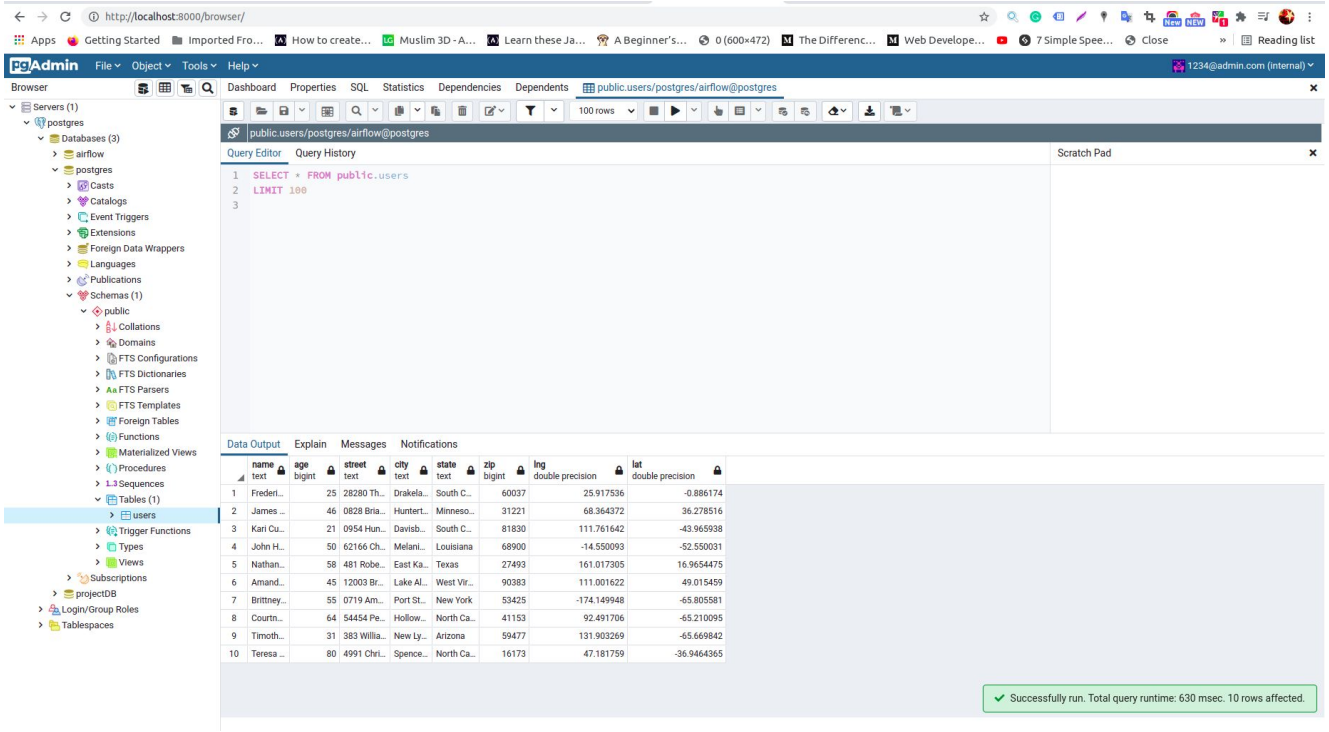


2.2 Save CSV in Postgresql Database

```python
# config varabiles
host = Variable.set("host", "postgres")
user = Variable.set("user", "airflow")
password = Variable.set("password", "airflow")
port = Variable.set("port", '5432')
database = Variable.set("database", 'postgres')
AIRFLOW_HOME = os.getenv('AIRFLOW_HOME')

def SaveCsvToPostgres():
    host = Variable.get('host')
    user = Variable.get('user')
    password = Variable.get('password')
    port = Variable.get('port')
    database = Variable.get('database')
    engine = create_engine(
        f'postgresql://{user}:{password}@{host}:{port}/{database}')
    print("Airflow Database Tables :- ", engine.table_names())
    DF = pd.read_csv(AIRFLOW_HOME + '/dags/dataframe.csv')
    # push table
    DF.to_sql('users', engine, if_exists='replace', index=False)
```



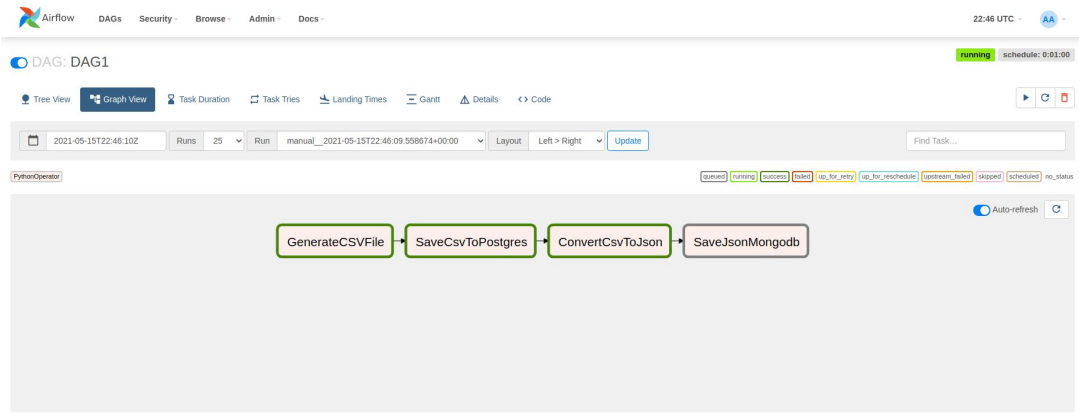
2.3 Save CSV in Postgresql Database



	name text	age bigint	street text	city text	state text	zip bigint	lng double precision	lat double precision
1	Frederi...	25	28280 Th...	Drakela...	South C...	60037	25.917536	-0.886174
2	James ...	46	0828 Bria...	Huntert...	Minneso...	31221	68.364372	36.278516
3	Kari Cu...	21	0954 Hun...	Davisb...	South C...	81830	111.761642	-43.965938
4	John H...	50	62166 Ch...	Melani...	Louisiana	68900	-14.550093	-52.550031
5	Nathan...	58	481 Robe...	East Ka...	Texas	27493	161.017305	16.9654475
6	Amand...	45	12003 Br...	Lake Al...	West Vir...	90383	111.001622	49.015459
7	Brittney...	55	0719 Am...	Port St...	New York	53425	-174.149948	-65.805581
8	Courtn...	64	54454 Pe...	Hollow...	North Ca...	41153	92.491706	-65.210095
9	Timoth...	31	383 Willia...	New Ly...	Arizona	59477	131.903269	-65.669842
10	Teresa ...	80	4991 Chri...	Spence...	North Ca...	16173	47.181759	-36.9464365

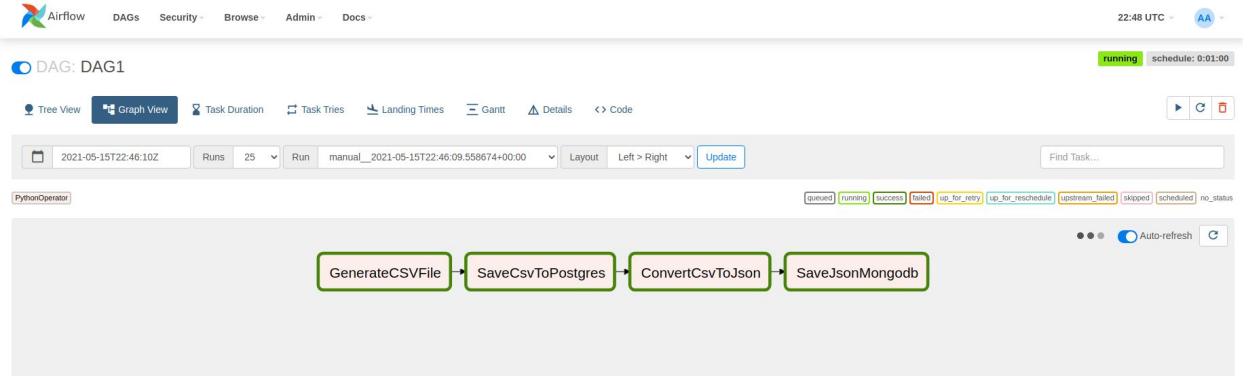
2.4 Convert To JSON

```python
def ConvertCsvToJson():
    # read from postgres
    host = Variable.get('host')
    user = Variable.get('user')
    password = Variable.get('password')
    port = Variable.get('port')
    database = Variable.get('database')
    engine = create_engine(
        f'postgresql://{user}:{password}@{host}:{port}/{database}')
    DF2 = pd.read_sql("SELECT * FROM users", engine)

    for i, r in DF2.iterrows():
        print(r['name'])

    DF2.to_json(AIRFLOW_HOME + '/dags/fromAirflow.json', orient='records')
```

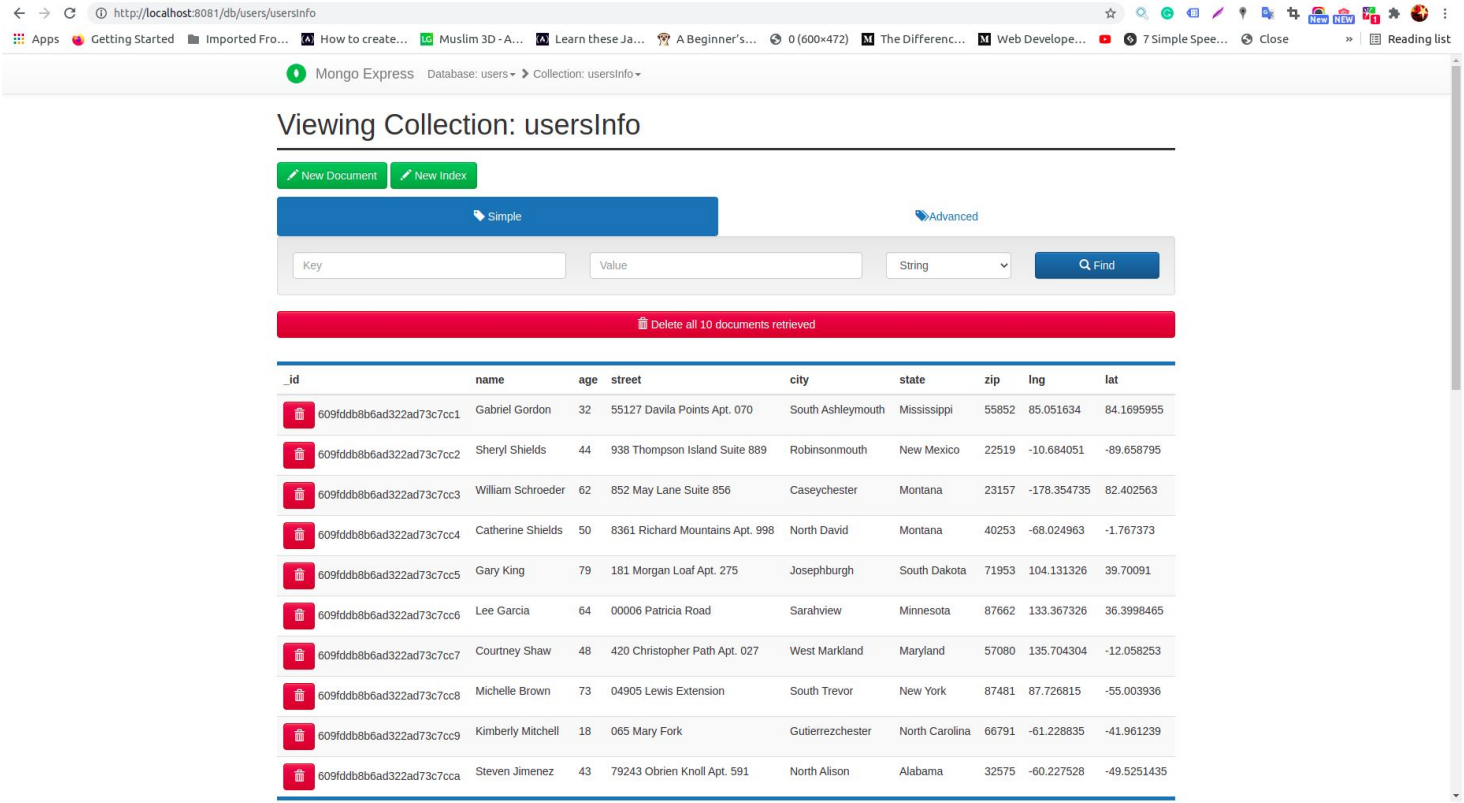


2.4 Save JSON file in MongoDB

```python
def SaveJsonMongodb():
    from pymongo import MongoClient
    client = MongoClient('mongo:27017',
                         username='root',
                         password='example')
    db = client['users']
    # Create Collection
    usersInfo = db.usersInfo
    with open(AIRFLOW_HOME + '/dags/fromAirflow.json') as f:
        users = json.load(f)
    # Push documents to collection
    for key in users:
        usersInfo.insert_one(key)
```



2.4 Save Json file in MongoDB



_id	name	age	street	city	state	zip	lng	lat
609fddb8b6ad322ad73c7cc1	Gabriel Gordon	32	55127 Davila Points Apt. 070	South Ashleymouth	Mississippi	55852	85.051634	84.1695955
609fddb8b6ad322ad73c7cc2	Sheryl Shields	44	938 Thompson Island Suite 889	Robinsonmouth	New Mexico	22519	-10.684051	-89.658795
609fddb8b6ad322ad73c7cc3	William Schroeder	62	852 May Lane Suite 856	Caseychester	Montana	23157	-178.354735	82.402563
609fddb8b6ad322ad73c7cc4	Catherine Shields	50	8361 Richard Mountains Apt. 998	North David	Montana	40253	-68.024963	-1.767373
609fddb8b6ad322ad73c7cc5	Gary King	79	181 Morgan Loaf Apt. 275	Josephburgh	South Dakota	71953	104.131326	39.70091
609fddb8b6ad322ad73c7cc6	Lee Garcia	64	00006 Patricia Road	Sarahview	Minnesota	87662	133.367326	36.3998465
609fddb8b6ad322ad73c7cc7	Courtney Shaw	48	420 Christopher Path Apt. 027	West Markland	Maryland	57080	135.704304	-12.058253
609fddb8b6ad322ad73c7cc8	Michelle Brown	73	04905 Lewis Extension	South Trevor	New York	87481	87.726815	-55.003936
609fddb8b6ad322ad73c7cc9	Kimberly Mitchell	18	065 Mary Fork	Gutierrezchester	North Carolina	66791	-61.228835	-41.961239
609fddb8b6ad322ad73c7cca	Steven Jimenez	43	79243 Obrien Knoll Apt. 591	North Alison	Alabama	32575	-60.227528	-49.5251435