In [122…
```
#
# Author : Wish MKN
#

import requests
```

In [79]:
```
url = "https://randomuser.me/api/"
```

https://randomuser.me/api/?results=300&nat=de,dk,fr,gb&inc=id,gender,name,location,email,dob,picture,nat

URL Broken down to understand it better

In [80]:
```
params = {'seed': 'flightright',
          'results': 300,
          'inc': 'id,gender,name,location,email,dob,picture,nat',
          'nat':'de,dk,fr,gb'}
```

In [81]:
```
request = requests.request("GET",url, params = params)
```

In [82]:
```
request = request.json()
```

Download the request as a json file

In [83]:
```
import pandas as pd
```

download and normalize the results part to a data frame

In [93]:
```
df = pd.json_normalize(request['results'])
```

downloaded raw data as dataframe

now we can rename the colums after checking it in DF info so that we can create a like-able Database schema

Creating a dictornary below based on existing column names to Clean column names

can keep or change it to whatever name required in database

In [97]:
```
rename_dict = {
    'gender':'Gender',
    'email':'Email',
    'nat':'Nat',
    'name.title':'Title',
    'name.first' : 'First_Name',
    'name.last':'Last_Name',
    'location.street.number':'Str_Number',
    'location.street.name':'Str_Name',
    'location.city':'City',
```

```
    'location.state':'State',
    'location.country':'Country',
    'location.postcode':'Postcode',
    'location.coordinates.latitude':'Latitude',
    'location.coordinates.longitude':'Longitude',
    'location.timezone.offset':'Time_Zone_Offset',
    'location.timezone.description':'Time_Zone_Discription',
    'dob.date':'DOB',
    'dob.age':'Age',
    'id.name':'ID_Name',
    'id.value':'ID_Value',
    'picture.large':'Pic_Large',
    'picture.medium':'Pic_Medium',
    'picture.thumbnail':'Pic_Thumbnail',
}
```

In [98]:
```
df = df.rename(columns=rename_dict)
```

In [103…
```
df.head(2)  # just to visualize our renamed columns
```

Out[103…

	Gender	Email	Nat	Title	First_Name	Last_Name	Str_Number	Str_N
0	male	florian.pierre@example.com	FR	Mr	Florian	Pierre	6120	Rue de (
1	female	gabriele.rupprecht@example.com	DE	Miss	Gabriele	Rupprecht	80	Gartenst

In [101…
```
df = df.drop(['Pic_Large', 'Pic_Medium','Pic_Thumbnail'], axis = 1)
```

Removed the picture columns that were above for now ,

note: we can keep that data seprately but for now removed them

Now exporting data generated as a dataframe to a CSV file

In [105…
```
df.to_csv('Results.csv', index = False)
```

In [107…
```
import pyodbc
```

open connection to sql odbc

In [116…
```
conn = pyodbc.connect("driver={ODBC Driver 17 for SQL Server};server= MSI;UID=W_NEW;
```

Using microsoft sql server management studio for SQL stuff.

1. Create connection to sql server management studio
2. Create a database

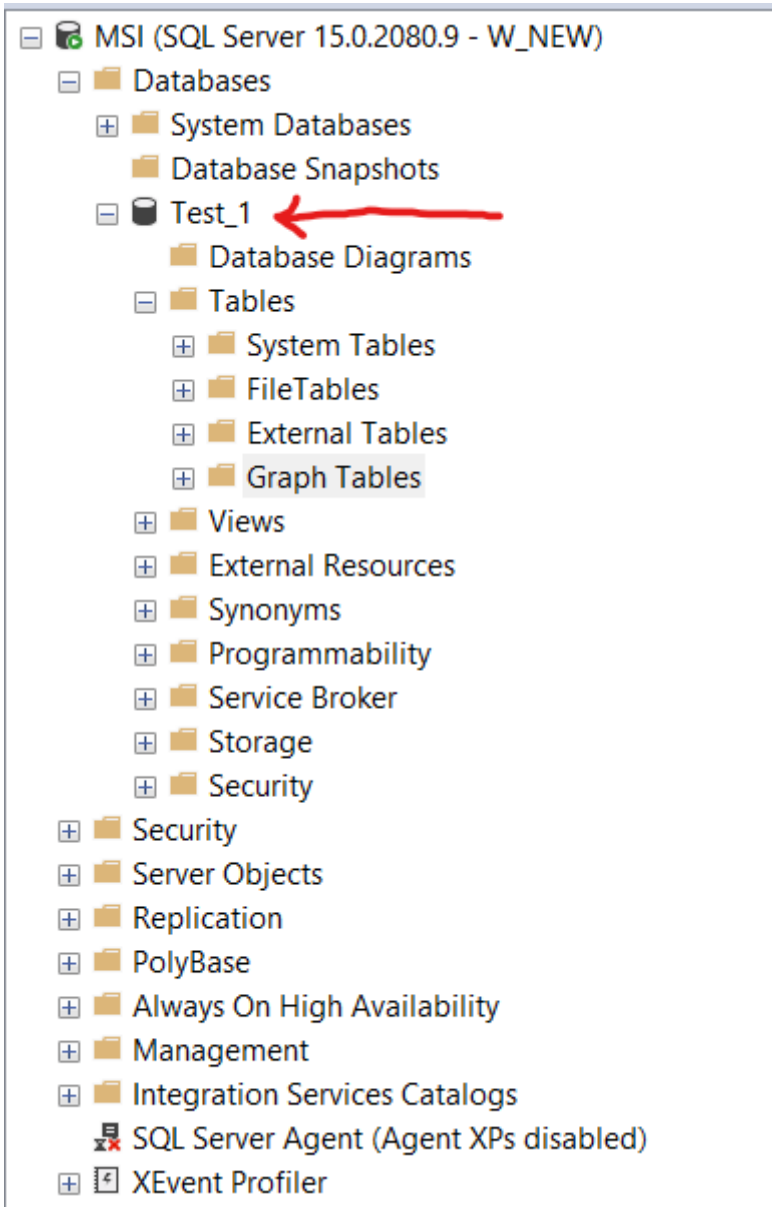
In [117…
```
cursor = conn.cursor()
```

In [118…
```
sq_cmd_1 = """
```

```
        CREATE DATABASE Test_1
"""
```

In [ ]:
```
cursor.execute(sq_cmd_1)
```

Test_1 Database created



created a database to work forward with for us

This part below is made in a simple way can add more things here

connecting to the sepecific database currently it is Test_1 and creating a table with our csv file generated

1. Navigating and opening connection to created database
2. Created table with data
3. Populated table with data
4. Running simple stastic queries on table to view text output
5. Autocommit set to true , easy commits to avoid manual commit again and again
6. Give correct usr id and paswrd for sql server login else code will not run

In [121…
```
conn = pyodbc.connect("driver={ODBC Driver 17 for SQL Server};server= MSI;database=T
```

In [123…
```
cursor = conn.cursor() # overrites and now it works in the Test_1 database area
```

In [155…
```
# Create table query
# for now kept it as simple as possible

sq_cmd_2 = """
        CREATE TABLE flightright_data
        (
        Gender varchar(10),
        Email varchar(250),
        Nat varchar(5),
        Title varchar(10),
        First_Name varchar(250),
        Last_Name varchar(250),
        Str_Number varchar(250),
        Str_Name varchar(250),
        City varchar(250),
        State varchar(250),
        Country varchar(250),
        Postcode varchar(250),
        Latitude varchar(250),
        Longitude varchar(250),
        Time_Zone_Offset varchar(250),
        Time_Zone_Discription varchar(250),
        DOB varchar(250),
        Age varchar(10),
        ID_Name varchar(250),
        ID_Value varchar(250)
        )
"""
```

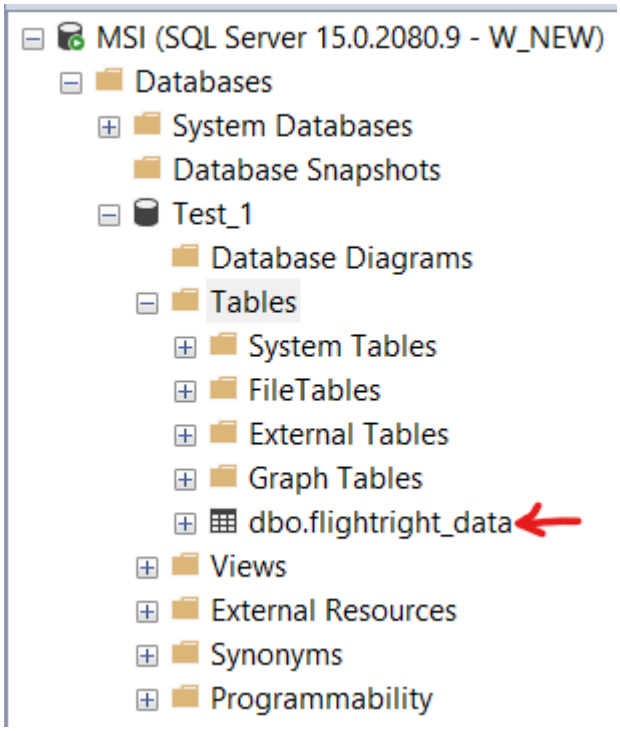
In [156…
```
# Data insert query

sq_cmd_3 = """
        BULK INSERT flightright_data
        FROM 'C:\\Users\\Wish\\Saved Games\\Results.csv'
        WITH ( FORMAT = 'CSV' )
"""
```

In [157…
```
cursor.execute(sq_cmd_2) # run table creation
```

Out[157… <pyodbc.Cursor at 0x2cb7f9fbcb0>

Table created in Test_1 Data Base



In [158...
```
cursor.execute(sq_cmd_3) # run data input to table
```

Out[158... <pyodbc.Cursor at 0x2cb7f9fbcb0>

data inserted into table and query execution for simple stats

In [159...
```
# delete 1st row as it is just the column name again

sq_cmd_4 = """
    DELETE FROM flightright_data
    WHERE Country = 'Country' AND Title = 'Title' AND Gender = 'Gender'
"""
cursor.execute(sq_cmd_4)
```

Out[159... <pyodbc.Cursor at 0x2cb7f9fbcb0>

In [161...
```
cursor.execute("ALTER TABLE flightright_data ALTER COLUMN Age int")  # changing colu
```

Out[161... <pyodbc.Cursor at 0x2cb7f9fbcb0>

In [162...
```
# get list of entries in country/ No of entries per country

cursor.execute("Select Country, COUNT(Country) AS COUNT_ENTRIES from flightright_dat
data = cursor.fetchall()

for x in data:
    print(x)
```

```
('Denmark', 64)
('France', 74)
('Germany', 82)
('United Kingdom', 80)
```

In [163...
```
# get avg age of each country
```

```
cursor.execute("Select Country, AVG(Age) AS AVG_AGE from flightright_data group by C
data = cursor.fetchall()

for x in data:
    print(x)
```

```
('Denmark', 52)
('France', 50)
('Germany', 50)
('United Kingdom', 52)
```

In [164…

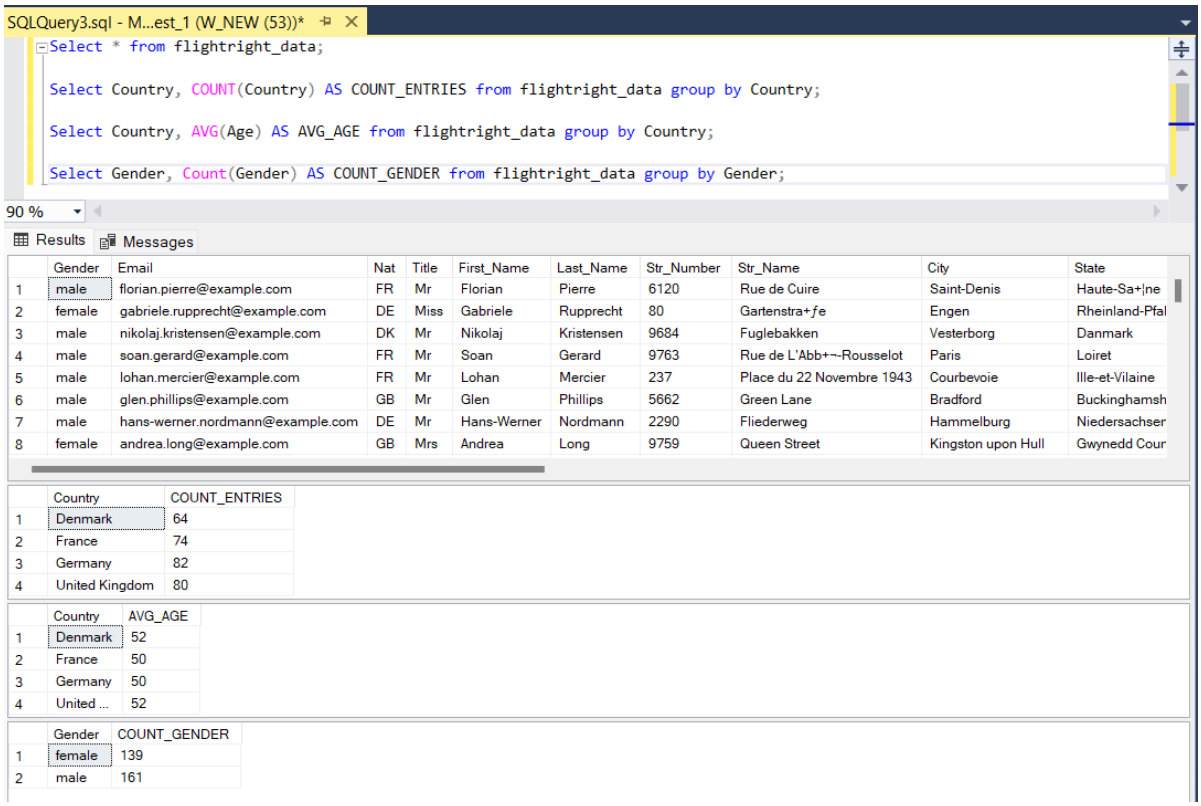
```
# get count of gender

cursor.execute("Select Gender, Count(Gender) AS COUNT_GENDER from flightright_data g
data = cursor.fetchall()

for x in data:
    print(x)
```

```
('female', 139)
('male', 161)
```

above results match to what we have in data base



	Gender	Email	Nat	Title	First_Name	Last_Name	Str_Number	Str_Name	City	State
1	male	florian.pierre@example.com	FR	Mr	Florian	Pierre	6120	Rue de Cuire	Saint-Denis	Haute-Sa+¦ne
2	female	gabriele.rupprecht@example.com	DE	Miss	Gabriele	Rupprecht	80	Gartenstra+ſe	Engen	Rheinland-Pfal
3	male	nikolaj.kristensen@example.com	DK	Mr	Nikolaj	Kristensen	9684	Fuglebakken	Vesterborg	Danmark
4	male	soan.gerard@example.com	FR	Mr	Soan	Gerard	9763	Rue de L'Abb+¬-Rousselot	Paris	Loiret
5	male	lohan.mercier@example.com	FR	Mr	Lohan	Mercier	237	Place du 22 Novembre 1943	Courbevoie	Ille-et-Vilaine
6	male	glen.phillips@example.com	GB	Mr	Glen	Phillips	5662	Green Lane	Bradford	Buckinghamsh
7	male	hans-werner.nordmann@example.com	DE	Mr	Hans-Werner	Nordmann	2290	Fliederweg	Hammelburg	Niedersachser
8	female	andrea.long@example.com	GB	Mrs	Andrea	Long	9759	Queen Street	Kingston upon Hull	Gwynedd Cour

	Country	COUNT_ENTRIES
1	Denmark	64
2	France	74
3	Germany	82
4	United Kingdom	80

	Country	AVG_AGE
1	Denmark	52
2	France	50
3	Germany	50
4	United ...	52

	Gender	COUNT_GENDER
1	female	139
2	male	161

In [ ]: