

Build a Data Pipeline for Scraping and Storing FDIC Data

THE SITUATION	<p>Created a data pipeline to scrap, process and store FDIC data from a Federal Deposit Insurance Corporation website. The pipeline will transform and load the data with Python into a PostgreSQL.</p> <p>Tools required: Python (Pandas), PostgreSQL, DBdiagram.io</p>
----------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

THE STEPS	<p>Used DBdiagram.io:</p> <ul style="list-style-type: none">• To design a dimensional model for FDIC Database. <p>Used PostgreSQL to:</p> <ul style="list-style-type: none">• Create a database title FDIC.• Define the database schema under the database (FDIC): <p>Used Python to:</p> <ul style="list-style-type: none">• Use pandas library to load the movies data from the Federal Deposit Insurance Corporation website into pandas dataframe.• Use pandas to clean and transform the data as needed.• Use the psycopg2 and to connect to the PostgreSQL database.• Use pandas, psycopg2 to load the cleaned and transformed data into the database.• Runed some test queries on the data in the database to ensure the data pipeline is working correctly.
------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

SQL SCRIPTS FOR FCID

```
create table bank_details(
```

```
    bank varchar(200),
```

```
    city varchar(200),
```

```
    state varchar(200),
```

```
    cert varchar(200),
```

```
    ai varchar(200),
```

```
    closing date,
```

```
    fund int
```

```
);
```

bank_details

bank varchar(200)

city varchar(200)

state varchar(200)

cert varchar(200)

ai varchar(200)

closing date

fund int

Scrap Data with Selenium

```
In [1]: pip install selenium
Requirement already satisfied: selenium in c:\users\lenovo\anaconda3\lib\site-packages (4.8.3)
Requirement already satisfied: trio~=0.17 in c:\users\lenovo\anaconda3\lib\site-packages (from selenium) (0.22.0)
Requirement already satisfied: urllib3[socks]~=1.26 in c:\users\lenovo\anaconda3\lib\site-packages (from selenium) (1.26.9)
Requirement already satisfied: trio-websocket~=0.9 in c:\users\lenovo\anaconda3\lib\site-packages (from selenium) (0.10.2)
Requirement already satisfied: certifi>=2021.10.8 in c:\users\lenovo\anaconda3\lib\site-packages (from selenium) (2021.10.8)
Requirement already satisfied: cffi>=1.14 in c:\users\lenovo\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.15.0)

Requirement already satisfied: outcome in c:\users\lenovo\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.2.0)
Requirement already satisfied: attrs>=19.2.0 in c:\users\lenovo\anaconda3\lib\site-packages (from trio~=0.17->selenium) (21.4.0)
Requirement already satisfied: idna in c:\users\lenovo\anaconda3\lib\site-packages (from trio~=0.17->selenium) (3.3)
Requirement already satisfied: exceptiongroup>=1.0.0rc9 in c:\users\lenovo\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.1.1)
Requirement already satisfied: sniffio in c:\users\lenovo\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.2.0)
Requirement already satisfied: async-generator>=1.9 in c:\users\lenovo\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.10)
Requirement already satisfied: sortedcontainers in c:\users\lenovo\anaconda3\lib\site-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: pycparser in c:\users\lenovo\anaconda3\lib\site-packages (from cffi>=1.14->trio~=0.17->selenium) (2.21)
Requirement already satisfied: wsproto>=0.14 in c:\users\lenovo\anaconda3\lib\site-packages (from trio-websocket~=0.9->selenium) (1.2.0)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\users\lenovo\anaconda3\lib\site-packages (from urllib3[socks]~=1.26->selenium) (1.7.1)
Requirement already satisfied: h11<1,>=0.9.0 in c:\users\lenovo\anaconda3\lib\site-packages (from wsproto>=0.14->trio-websocket~=0.9->selenium) (0.14.0)
```

```
In [2]: from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.support.select import Select
import pandas as pd

s = Service(r'C:\Users\Lenovo\Downloads\Software Application\chromedriver_win32\chromedriver.exe')
driver = webdriver.Chrome(service = s)
url = 'https://www.fdic.gov/resources/resolutions/bank-failures/failed-bank-list/'
driver.get(url)
dropdown = driver.find_element(By.XPATH, "//*[@id='DataTables_Table_0_length']/label/select")
radius = Select(dropdown)
radius.select_by_visible_text('All')
Bank_Details = driver.find_elements(By.XPATH, "//*[@id='DataTables_Table_0']/tbody/tr")

bank = []
city = []
state = []
cert = []
AI = []
closing = []
Fund = []

for bank_detail in Bank_Details:
    bank.append(bank_detail.find_element(By.XPATH, ".//td[1]").text)
    city.append(bank_detail.find_element(By.XPATH, ".//td[2]").text)
    state.append(bank_detail.find_element(By.XPATH, ".//td[3]").text)
    cert.append(bank_detail.find_element(By.XPATH, ".//td[4]").text)
    AI.append(bank_detail.find_element(By.XPATH, ".//td[5]").text)
    closing.append(bank_detail.find_element(By.XPATH, ".//td[6]").text)
    Fund.append(bank_detail.find_element(By.XPATH, ".//td[7]").text)

    Details = {'bank' : bank, 'city' : city, 'state' : state, 'cert' : cert, 'ai' : AI, 'closing' : closing, 'fund' : Fund}

NewData = pd.DataFrame(Details)
NewData
```

```
Out[2]:
```

	bank	city	state	cert	ai	closing	fund
0	First Republic Bank	San Francisco	CA	59017	JPMorgan Chase Bank, N.A.	May 1, 2023	10543
1	Signature Bank	New York	NY	57053	Flagstar Bank, N.A.	March 12, 2023	10540
2	Silicon Valley Bank	Santa Clara	CA	24735	First-Citizens Bank & Trust Company	March 10, 2023	10539
3	Almena State Bank	Almena	KS	15426	Equity Bank	October 23, 2020	10538
4	First City Bank of Florida	Fort Walton Beach	FL	16748	United Fidelity Bank, fsb	October 16, 2020	10537
...
561	Superior Bank, FSB	Hinsdale	IL	32646	Superior Federal, FSB	July 27, 2001	6004
562	Malta National Bank	Malta	OH	6629	North Valley Bank	May 3, 2001	4648
563	First Alliance Bank & Trust Co.	Manchester	NH	34264	Southern New Hampshire Bank & Trust	February 2, 2001	4647
564	National State Bank of Metropolis	Metropolis	IL	3815	Banterra Bank of Marion	December 14, 2000	4646
565	Bank of Honolulu	Honolulu	HI	21029	Bank of the Orient	October 13, 2000	4645

566 rows × 7 columns

Load Data in PostgreSQL

```
In [3]: pip install sqlalchemy
Requirement already satisfied: sqlalchemy in c:\users\lenovo\anaconda3\lib\site-packages (1.4.32)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\lenovo\anaconda3\lib\site-packages (from sqlalchemy) (1.1.1)
Note: you may need to restart the kernel to use updated packages.

In [4]: import pandas as pd

In [5]: import sqlalchemy

In [6]: engine = sqlalchemy.create_engine("postgresql+psycopg2://Uche:diamond@localhost/FDIC")

In [7]: NewData.to_sql(name = 'bank_details', con = engine, index = False, if_exists = 'replace')
```

Out[7]: 566

```
In [8]: df = pd.read_sql_table("bank_details", engine)
df
```

```
Out[8]:
```

	bank	city	state	cert	ai	closing	fund
0	First Republic Bank	San Francisco	CA	59017	JPMorgan Chase Bank, N.A.	May 1, 2023	10543
1	Signature Bank	New York	NY	57053	Flagstar Bank, N.A.	March 12, 2023	10540
2	Silicon Valley Bank	Santa Clara	CA	24735	First-Citizens Bank & Trust Company	March 10, 2023	10539
3	Almena State Bank	Almena	KS	15426	Equity Bank	October 23, 2020	10538
4	First City Bank of Florida	Fort Walton Beach	FL	16748	United Fidelity Bank, fsb	October 16, 2020	10537
...
561	Superior Bank, FSB	Hinsdale	IL	32646	Superior Federal, FSB	July 27, 2001	6004
562	Malta National Bank	Malta	OH	6629	North Valley Bank	May 3, 2001	4648
563	First Alliance Bank & Trust Co.	Manchester	NH	34264	Southern New Hampshire Bank & Trust	February 2, 2001	4647
564	National State Bank of Metropolis	Metropolis	IL	3815	Banterra Bank of Marion	December 14, 2000	4646
565	Bank of Honolulu	Honolulu	HI	21029	Bank of the Orient	October 13, 2000	4645

566 rows × 7 columns

```
In [ ]:
```

