**Hands-on Lab: Acquiring and Processing Information on the World's Largest Banks**

A logo with text on it

Description automatically generated

**Estimated Time: 60 mins**

In this project, you will put all the skills acquired throughout the course and your knowledge of basic Python to test. You will work on real-world data and perform the operations of Extraction, Transformation, and Loading (ETL) as required.

**Disclaimer:**

*Cloud IDE is not a persistent platform, and you will lose your progress every time you restart this lab. We recommend saving a copy of your file on your local machine as a protective measure against data loss.*

**Project Scenario:**

You have been hired as a data engineer by research organization. Your boss has asked you to create a code that can be used to compile the list of the top 10 largest banks in the world ranked by market capitalization in billion USD. Further, the data needs to be transformed and stored in GBP, EUR and INR as well, in accordance with the exchange rate information that has been made available to you as a CSV file. The processed information table is to be saved locally in a CSV format and as a database table.

Your job is to create an automated system to generate this information so that the same can be executed in every financial quarter to prepare the report.

Particulars of the code to be made have been shared below.

| **Parameter** | **Value** |
| --- | --- |
| Code name | banks\_project.py |
| Data URL | https://web.archive.org/web/20230908091635 /https://en.wikipedia.org/wiki/List\_of\_largest\_banks |
| Exchange rate CSV path | <https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMSkillsNetwork-PY0221EN-Coursera/labs/v2/exchange_rate.csv> |
| Table Attributes (upon Extraction only) | Name, MC\_USD\_Billion |
| Table Attributes (final) | Name, MC\_USD\_Billion, MC\_GBP\_Billion, MC\_EUR\_Billion, MC\_INR\_Billion |
| Output CSV Path | ./Largest\_banks\_data.csv |
| Database name | Banks.db |
| Table name | Largest\_banks |
| Log file | code\_log.txt |

**Project tasks**

**Task 1:**  
Write a function log\_progress() to log the progress of the code at different stages in a file code\_log.txt. Use the list of log points provided to create log entries as every stage of the code.

**Task 2:**  
Extract the tabular information from the given URL under the heading 'By market capitalization' and save it to a dataframe.  
a. Inspect the webpage and identify the position and pattern of the tabular information in the HTML code  
b. Write the code for a function extract() to perform the required data extraction.  
c. Execute a function call to extract() to verify the output.

**Task 3:**  
Transform the dataframe by adding columns for Market Capitalization in GBP, EUR and INR, rounded to 2 decimal places, based on the exchange rate information shared as a CSV file.  
a. Write the code for a function transform() to perform the said task.  
b. Execute a function call to transform() and verify the output.

**Task 4:**  
Load the transformed dataframe to an output CSV file. Write a function load\_to\_csv(), execute a function call and verify the output.

**Task 5:**  
Load the transformed dataframe to an SQL database server as a table. Write a function load\_to\_db(), execute a function call and verify the output.

**Task 6:**  
Run queries on the database table. Write a function load\_to\_db(), execute a given set of queries and verify the output.

**Task 7:**  
Verify that the log entries have been completed at all stages by checking the contents of the file code\_log.txt.

**Preliminaries: Installing libraries and downloading data**

Before building the code, you need to install the required libraries.

The libraries needed for the code are:

requests - The library used for accessing the information from the URL.

bs4 - The library containing the BeautifulSoup function used for webscraping.

pandas - The library used for processing the extracted data, storing it in required formats, and communicating with the databases.

sqlite3 - The library required to create a database server connection.

numpy - The library required for the mathematical rounding operations.

datetime - The library containing the function datetime used for extracting the timestamp for logging purposes.

Install the required libraries from the terminal window. The command syntax is:

1. 1
2. python3.11 -m pip install <library\_name>

Copied!

Also, download the required exchange rate file using the terminal command:

1. 1
2. wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMSk

**Task 1: Logging function**

Write the function to log the progress of the code, log\_progress(). The function accepts the message to be logged and enters it to a text file code\_log.txt.

The format to be used for logging must have the syntax:

1. 1
2. <time\_stamp> : <message>

Copied!

Each log entry must happen in the next line in the text file.

You must associate the correct log entries with each of the executed function calls. Use the following table to note the logging message at the end of each function call that follows.

| **Task** | **Log message on completion** |
| --- | --- |
| Declaring known values | Preliminaries complete. Initiating ETL process |
| Call extract() function | Data extraction complete. Initiating Transformation process |
| Call transform() function | Data transformation complete. Initiating Loading process |
| Call load\_to\_csv() | Data saved to CSV file |
| Initiate SQLite3 connection | SQL Connection initiated |
| Call load\_to\_db() | Data loaded to Database as a table, Executing queries |
| Call run\_query() | Process Complete |
| Close SQLite3 connection | Server Connection closed |

At this stage, you should now make the first log entry from the table above.

**Peer graded assignment prompt:**

Take a screenshot of the code, as created for the log\_progress() function and save it to your local machine as Task\_1\_log\_function.png

Previous

# Task 2 : Extraction of data

Analyze the webpage on the given URL:

1. 1
2. https://web.archive.org/web/20230908091635/https://en.wikipedia.org/wiki/List\_of\_largest\_banks

Copied!

Identify the position of the required table under the heading By market capitalization. Write the function extract() to retrieve the information of the table to a Pandas data frame.

Note: Remember to remove the last character from the Market Cap column contents, like, '\n', and typecast the value to float format.

Write a function call for extract() and print the returning data frame.

Make the relevant log entry.

Execute the code using the command:

1. 1
2. python3.11 banks\_project.py

Copied!Executed!

#### Quiz question prompt:

While inspecting the webpage, note the attributes of the data in the first row. There will be a quiz questions based on these attributes.

#### Peer graded assignment prompt:

Take a screenshot of the html code of the table, as obtained by inspecting the webpage. Make sure that the contents of at least the first row of the table, as entered in the HTML code, are completely visible. Save this screenhot to your local machine as Task\_2a\_extract.png

Take a screenshot of the code, as created for the extract() function and save it to your local machine as Task\_2b\_extract.png.

Take a screenshot of the output, as obtained upon execution in the terminal, and save it to your local machine as Task\_2c\_extract.png.

# Task 2 : Extraction of data

Analyze the webpage on the given URL:

1. 1
2. https://web.archive.org/web/20230908091635/https://en.wikipedia.org/wiki/List\_of\_largest\_banks

Copied!

Identify the position of the required table under the heading By market capitalization. Write the function extract() to retrieve the information of the table to a Pandas data frame.

Note: Remember to remove the last character from the Market Cap column contents, like, '\n', and typecast the value to float format.

Write a function call for extract() and print the returning data frame.

Make the relevant log entry.

Execute the code using the command:

1. 1
2. python3.11 banks\_project.py

Copied!Executed!

#### Quiz question prompt:

While inspecting the webpage, note the attributes of the data in the first row. There will be a quiz questions based on these attributes.

#### Peer graded assignment prompt:

Take a screenshot of the html code of the table, as obtained by inspecting the webpage. Make sure that the contents of at least the first row of the table, as entered in the HTML code, are completely visible. Save this screenhot to your local machine as Task\_2a\_extract.png

Take a screenshot of the code, as created for the extract() function and save it to your local machine as Task\_2b\_extract.png.

Take a screenshot of the output, as obtained upon execution in the terminal, and save it to your local machine as Task\_2c\_extract.png.

**Task 3 : Transformation of data**

The Transform function needs to perform the following tasks:

1. Read the exchange rate CSV file and convert the contents to a dictionary so that the contents of the first columns are the keys to the dictionary and the contents of the second column are the corresponding values.

Click here for hint

1. Add 3 different columns to the dataframe, viz. MC\_GBP\_Billion, MC\_EUR\_Billion and MC\_INR\_Billion, each containing the content of MC\_USD\_Billion scaled by the corresponding exchange rate factor. Remember to round the resulting data to 2 decimal places.

A sample statement is being provided for adding the MC\_GBP\_Billion column. You can use this to add the other two statements on your own.

1. 1
2. df['MC\_GBP\_Billion'] = [np.round(x\*exchange\_rate['GBP'],2) for x in df['MC\_USD\_Billion']]

Copied!

Write the function call for transform() and print the contents of the returning data frame. Comment out all previous print statements.

Make the relevant log entry and execute the code.

**Quiz question prompt:**

1. Experiment with the statement provided for adding the transformed columns to the dataframe. There will be a question on this in the quiz.
2. Print the contents of df['MC\_EUR\_Billion'][4], which is the market capitalization of the 5th largest bank in billion EUR. Note this value, as it will be the answer to a question in the final quiz.

**Peer graded assignment prompt:**

Take a screenshot of the code, as created for the transform() function, and save it to your local machine as Task\_3a\_transform.png.

Table of Contents

+-

1Hands-on Lab: Acquiring and Processing Information on the World's Largest Banks2Preliminaries: Installing libraries and downloading data3Code Structure4Task 1: Logging function5Task 2 : Extraction of data6Task 3 : Transformation of data7Task 4: Loading to CSV8Task 5: Loading to Database9Task 6: Function to Run queries on Database10Task 7: Verify log entries11Conclusion

**Task 3 : Transformation of data**

The Transform function needs to perform the following tasks:

1. Read the exchange rate CSV file and convert the contents to a dictionary so that the contents of the first columns are the keys to the dictionary and the contents of the second column are the corresponding values.

Click here for hintYou can use the below-mentioned syntax to achieve this. Remember to modify the statement as per your code.

1. 1
2. dict = dataframe.set\_index('Col\_1\_header').to\_dict()['Col\_2\_header']

Copied!

1. Add 3 different columns to the dataframe, viz. MC\_GBP\_Billion, MC\_EUR\_Billion and MC\_INR\_Billion, each containing the content of MC\_USD\_Billion scaled by the corresponding exchange rate factor. Remember to round the resulting data to 2 decimal places.

A sample statement is being provided for adding the MC\_GBP\_Billion column. You can use this to add the other two statements on your own.

1. 1
2. df['MC\_GBP\_Billion'] = [np.round(x\*exchange\_rate['GBP'],2) for x in df['MC\_USD\_Billion']]

Copied!

Write the function call for transform() and print the contents of the returning data frame. Comment out all previous print statements.

Make the relevant log entry and execute the code.

**Quiz question prompt:**

1. Experiment with the statement provided for adding the transformed columns to the dataframe. There will be a question on this in the quiz.
2. Print the contents of df['MC\_EUR\_Billion'][4], which is the market capitalization of the 5th largest bank in billion EUR. Note this value, as it will be the answer to a question in the final quiz.

**Peer graded assignment prompt:**

Take a screenshot of the code, as created for the transform() function, and save it to your local machine as Task\_3a\_transform.png.

Take a snapshot of the output and save it as Task\_3b\_tranform.png.

# Task 4: Loading to CSV

Write the function to load the transformed data frame to a CSV file, like load\_to\_csv(), in the path mentioned in the project scenario.

Make the relevant log entry.

#### Peer graded assignment prompt:

Double-click the created CSV file in the Explorer tab on the left ribbon of the programming pane in Cloud IDE. Note that its contents are displayed on the editor screen. Take a snapshot of this screen and save it as Task\_4\_CSV.png.

# Task 5: Loading to Database

Write the function to load the transformed data frame to an SQL database, like, load\_to\_db(). Use the database and table names as mentioned in the project scenario.

Before calling this function, initiate the connection to the SQLite3 database server with the name Banks.db. Pass this connection object, along with the required table name Largest\_banks and the transformed data frame, to the load\_to\_db() function in the function call.

Make the relevant log entry.

Upon successful function call, you will have loaded the contents of the table with the required data and the file Banks.db will be visible in the Explorer tab of the IDE under the project folder.

#### Peer graded assignment prompt:

Take a single screenshot of the code, as created for load\_to\_csv() and load\_to\_db() functions, and save it to your local machine as Task\_4\_5\_save\_file.png.

**Task 6: Function to Run queries on Database**

Write the function run\_queries() that accepts the query statement, and the SQLite3 Connection object, and generates the output of the query. The query statement should be printed along with the query output.

Execute 3 function calls using the queries as mentioned below.

1. Print the contents of the entire table

Query statement:

1. 1
2. SELECT \* FROM Largest\_banks

Copied!

1. Print the average market capitalization of all the banks in Billion USD.

Query statement:

1. 1
2. SELECT AVG(MC\_GBP\_Billion) FROM Largest\_banks

Copied!

1. Print only the names of the top 5 banks

Query statement:

1. 1
2. SELECT Name from Largest\_banks LIMIT 5

Copied!

Make the relevant log entry.

**Peer graded assignment prompt:**

Take the snapshot of the output and save it as Task\_6\_SQL.png. Please adjust the size of the terminal prompt in order to take a single screenshot that captures all three outputs together.

**Quiz question prompt:**

There will be a quiz question on the output of these queries.

**Task 7: Verify log entries**

After updating all the log\_progress() function calls, you have to run the code for a final execution. However, you will first have to remove the code\_log.txt file, that would have been created and updated throughout the multiple executions of the code in this lab. You may remove the file using the following command on a terminal.

1. 1
2. rm code\_log.txt

Copied!Executed!

Once the existing file is removed, now run the final execution. Upon successful completion of execution, open the code\_log.txt file by clicking on it in the Explorer tab of the toolbar on left side of the programming pane of the IDE, under the project folder. You should see all the relevant entries made in the text file in relation to the stages of code execution.

**Peer graded assignment prompt:**

Take the snapshot of the file contents and save it as Task\_7\_log\_content.png.

Previous

# Conclusion

Congratulations on completing this project!

With this, you are now trained to perform ETL operations on real-world data and make the processed information available for further use in different formats.

You should now be able to:

* Use Webscraping techniques to extract information from any website as per requirement.
* Use Pandas data frames and dictionaries to transform data as per requirement.
* Load the processed information to CSV files and as Database tables
* Query the database tables using SQLite3 and pandas libraries
* Log the progress of the code properly

## Author(s)

[Abhishek Gagneja](https://www.coursera.org/instructor/~129186572)

## Changelog

| **Date** | **Version** | **Changed by** | **Change Description** |
| --- | --- | --- | --- |
| 2023-09-19 | 0.3 | Abhishek Gagneja | Instruction revision |
| 2023-09-19 | 0.2 | Mary Stenberg | QA pass with edits |
| 2023-09-08 | 0.1 | Abhishek Gagneja | Initial version created |

### © IBM Corporation 2023. All rights reserved.