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 as required. Throughout the project, you will note some outputs you need to answer questions on the graded quiz. You will also take snapshots, which you will upload in the peergraded assignment.

Project Scenario

A multi-national firm has hired you as a data engineer. Your job is to access and process data as per requirements.

Your boss asked you to compile the list of the top 10 largest banks in the world ranked by market capitalization in billion USD. Further, you need to transform the data and store it in USD, GBP, EUR, and INR per the exchange rate information made available to you as a CSV file. You should save the processed information table locally in a CSV format and as a database table. Managers from different countries will query the database table to extract the list and note the market capitalization value in their own currency.

Directions

- 1. Write a function to extract the tabular information from the given URL under the heading *By Market Capitalization*, and save it to a data frame.
- 2. Write a function to transform the data frame 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.
- 3. Write a function to load the transformed data frame to an output CSV file.
- 4. Write a function to load the transformed data frame to an SQL database server as a table.
- 5. Write a function to run queries on the database table.

- 6. Run the following queries on the database table:a. Extract the information for the London office, that is *Name* and *MC_GBP_Billion*b. Extract the information for the Berlin office, that is *Name* and *MC_EUR_Billion*c. Extract the information for New Delhi office, that is *Name* and *MC_INR_Billion*
- 7. Write a function to log the progress of the code.
- 8. While executing the data initialization commands and function calls, maintain appropriate log entries.

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



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.

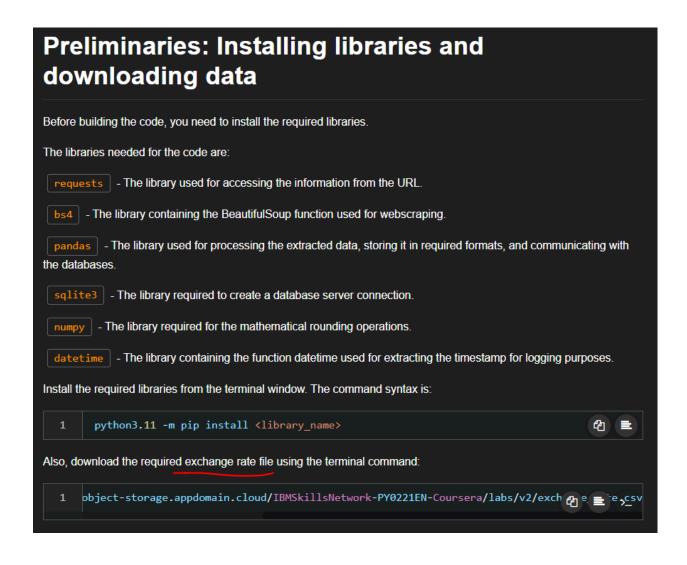
Project Scenario:

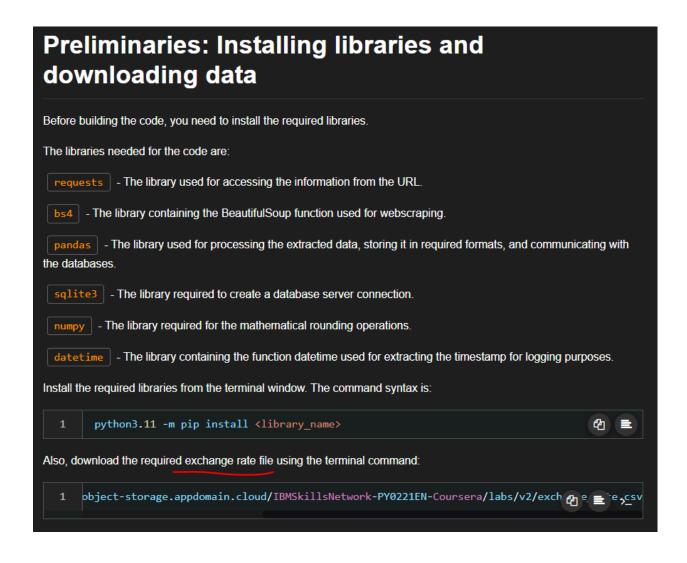
You have been hired as a data engineer by research organization. Your boss has asked you to create a code that can be used to compile the list of the top 10 largest banks in the world ranked by market capitalization in billion USD. Further, the data needs to be transformed and stored in GBP, EUR and 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.

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 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 \[\text{transform()} \] 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 [run_queries()], 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 | .





```
Code Structure
Create the file banks_project.py in the path \home\project\ . Copy and paste the following code structure to the file:
                                                                                                        42 ≡
         def log_progress(message):
             ''' This function logs the mentioned message of a given stage of the
             code execution to a log file. Function returns nothing'''
         def extract(url, table_attribs):
  10
             information from the website and save it to a data frame. The
             function returns the data frame for further processing. '''
             return df
         def transform(df, csv_path):
  16
             "This function accesses the CSV file for exchange rate
             information, and adds three columns to the data frame, each
  18
             containing the transformed version of Market Cap column to
  20
             return df
         def load to csv(df, output path):
  27
         def load to db(df, sql connection, table name):
             "" This function saves the final data frame to a database
             table with the provided name. Function returns nothing.'''
  30
```

```
def run_query(query_statement, sql_connection):

''' This function runs the query on the database table and

prints the output on the terminal. Function returns nothing. '''

This function runs the query on the database table and

prints the output on the terminal. Function returns nothing. '''

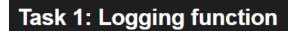
There, you define the required entities and call the relevant

functions in the correct order to complete the project. Note that this

portion is not inside any function.'''

At this stage, import the required libraries at the space mentioned in the code structure. Save the file using Ctrl+S.

Also, initialize all the known variables as shared in the project scenario.
```



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 <time_stamp> : <message>

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

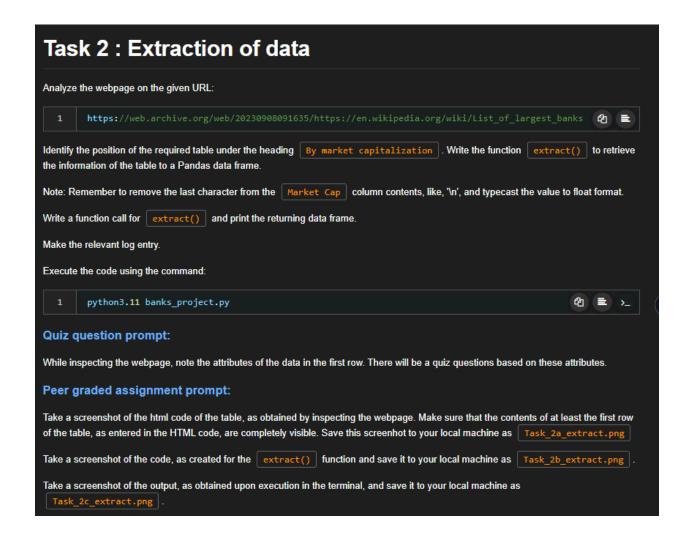
You must associate the correct log entries with each of the executed function calls. Use the following table to note the logging message at the end of each function call that follows.

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:

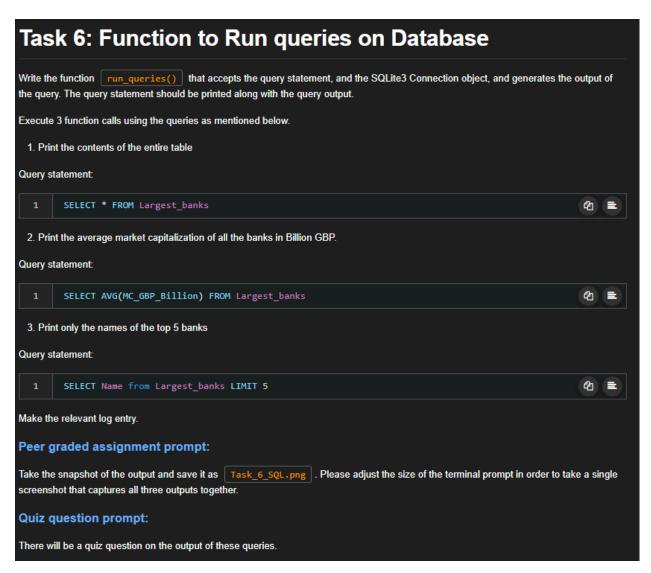
Take a screenshot of the code, as created for the log_progress() function and save it to your local machine as Task_1_log_function.png



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 2. 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 df['MC_GBP_Billion'] = [np.round(x*exchange_rate['GBP'],2) for x in df['MC_USD_Billion']] 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 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 <code>load_to_csv()</code>, 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 <code>Explorer</code> 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 <code>Task_4_CSV.png</code>.

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 load_to_db(). Pass this connection object, along with the required table name 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 load_to_db() will be visible in the load_to_db() function to the load_to_db() function in the function call. 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 load_to_page.



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