Laboratório Prático: Aquisição e Processamento de Informações sobre os Maiores Bancos do Mundo

Tempo Estimado: 60 mins

Neste projeto, você colocará todas as habilidades adquiridas ao longo do curso e seu conhecimento básico de Python à prova. Você trabalhará com dados do mundo real e realizará as operações de Extração, Transformação e Carregamento (ETL) conforme necessário.

Aviso:

> O IDE em nuvem não é uma plataforma persistente, e você perderá seu progresso toda vez que reiniciar este laboratório. Recomendamos salvar uma cópia do seu arquivo em sua máquina local como uma medida de proteção contra perda de dados.

Cenário do Projeto:

Você foi contratado como engenheiro de dados por uma organização de pesquisa. Seu chefe pediu que você criasse um código que possa ser usado para compilar a lista dos 10 maiores bancos do mundo classificados por capitalização de mercado em bilhões de USD. Além disso, os dados precisam ser transformados e armazenados em GBP, EUR e INR também, de acordo com as informações da taxa de câmbio que foram disponibilizadas para você como um arquivo CSV. A tabela de informações processadas deve ser salva localmente em formato CSV e como uma tabela de banco de dados.

Seu trabalho é criar um sistema automatizado para gerar essas informações para que o mesmo possa ser executado em cada trimestre financeiro para preparar o relatório.

Particularidades do código a ser criado foram compartilhadas abaixo.

Parâmetro	**Valor**
Nome do código	`banks_project.py`
URL dos dados	`https://web.archive.org/web/20230908091635 /https://en.wikipedia.org/wiki/List_of_largest_banks`
Caminho do CSV da taxa de câmbio	https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMSkillsNetwork-PY0221EN-Coursera/labs/v2/exchange_rate.csv
Atributos da Tabela (apenas na Extração)	`Name, MC_USD_Billion`
Atributos da Tabela (final)	`Name, MC_USD_Billion, MC_GBP_Billion, MC_EUR_Billion, MC_INR_Billion`
Caminho do CSV de Saída	`./Largest_banks_data.csv`
Nome do banco de dados	`Banks.db`
Nome da tabela	`Largest_banks`
Arquivo de log	`code_log.txt`

Tarefas do Projeto

Tarefa 1:
Escreva uma função `log_progress()` para registrar o progresso do código em diferentes estágios em um arquivo `code_log.txt`. Use a lista de pontos de log fornecida para criar entradas de log em cada estágio do código.

Tarefa 2:
Extraia as informações tabulares da URL fornecida sob o título 'Por capitalização de mercado' e salve em um dataframe.
a. Inspecione a página da web e identifique a posição e o padrão das informações tabulares no código HTML
b. Escreva o código para uma função `extract()` para realizar a extração de dados necessária.
c. Execute uma chamada de função para `extract()` para verificar a saída.

Tarefa 3:
Transforme o dataframe adicionando colunas para a Capitalização de Mercado em GBP, EUR e INR, arredondadas para 2 casas decimais, com base nas informações da taxa de câmbio compartilhadas como um arquivo CSV.
a. Escreva o código para uma função `transform()` para realizar a tarefa mencionada.
b. Execute uma chamada de função para `transform()` e verifique a saída.

Tarefa 4:
Carregue o dataframe transformado em um arquivo CSV de saída. Escreva uma função `load_to_csv()`, execute uma chamada de função e verifique a saída.

Tarefa 5:
Carregue o dataframe transformado em um servidor de banco de dados SQL como uma tabela. Escreva uma função `load_to_db()`, execute uma chamada de função e verifique a saída.

Tarefa 6:
Execute consultas na tabela do banco de dados. Escreva uma função `load_to_db()`, execute um conjunto de consultas fornecidas e verifique a saída.

Tarefa 7:
Verifique se as entradas de log foram completadas em todas as etapas, verificando o conteúdo do arquivo `code_log.txt`.

Preliminares: Instalando bibliotecas e baixando dados

Antes de construir o código, você precisa instalar as bibliotecas necessárias.

As bibliotecas necessárias para o código são:

`requests` - A biblioteca usada para acessar as informações da URL.

`bs4` - A biblioteca que contém a função BeautifulSoup usada para webscraping.

`pandas` - A biblioteca usada para processar os dados extraídos, armazená-los nos formatos necessários e se comunicar com os bancos de dados.

`sqlite3` - A biblioteca necessária para criar uma conexão com o servidor de banco de dados.

`numpy` - A biblioteca necessária para as operações de arredondamento matemático.

`datetime` - A biblioteca que contém a função datetime usada para extrair o timestamp para fins de registro.

Instale as bibliotecas necessárias a partir da janela do terminal. A sintaxe do comando é:

```
python3.11 -m pip install <library_name>
```

Além disso, baixe o arquivo de taxa de câmbio necessário usando o comando do terminal:

```
wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMSkillsNetwork-PY0221EN-Coursera/labs/v2/exchange_rate.csv
```

Estrutura do Código

Crie o arquivo `banks_project.py` no caminho `\home\project\`. Copie e cole a seguinte estrutura de código no arquivo:

```
# Code for ETL operations on Country-GDP data
# Importing the required libraries
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):
    ''' This function aims to extract the required
    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):
    ''' This function accesses the CSV file for exchange rate
    information, and adds three columns to the data frame, each
    containing the transformed version of Market Cap column to
    respective currencies'''
    return df
def load_to_csv(df, output_path):
    ''' This function saves the final data frame as a CSV file in
    the provided path. Function returns nothing.'''
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.'''
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. '''
```

```
''' Here, 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.'''
```

Nesta fase, importe as bibliotecas necessárias no espaço mencionado na estrutura do código. Salve o arquivo usando `Ctrl+S`.

Além disso, inicialize todas as variáveis conhecidas conforme compartilhado no cenário do projeto.

Tarefa 1: Função de registro

Escreva a função para registrar o progresso do código, `log_progress()`. A função aceita a mensagem a ser registrada e a insere em um arquivo de texto `code_log.txt`.

O formato a ser utilizado para o registro deve ter a sintaxe:

```
<time_stamp> : <message>
```

Cada entrada de log deve ocorrer na próxima linha no arquivo de texto.

Você deve associar as entradas de log corretas com cada uma das chamadas de função executadas. Use a tabela a seguir para anotar a mensagem de log ao final de cada chamada de função que se segue.

Tarefa	**Mensagem de log ao completar**
Declarar valores conhecidos	Preliminares concluídas. Iniciando o processo ETL
Chamar a função extract()	Extração de dados concluída. Iniciando o processo de Transformação
Chamar a função transform()	Transformação de dados concluída. Iniciando o processo de Carga
Chamar load_to_csv()	Dados salvos no arquivo CSV
Iniciar conexão SQLite3	Conexão SQL iniciada
Chamar load_to_db()	Dados carregados no Banco de Dados como uma tabela, Executando consultas
Chamar run_query()	Processo completo
Fechar conexão SQLite3	Conexão com o servidor fechada

Neste estágio, você deve agora fazer a primeira entrada de log da tabela acima.

Prompt da tarefa avaliada por pares:

Tire uma captura de tela do código, conforme criado para a função `log_progress()` e salve-o em sua máquina local como `Task_1_log_function.png`

Tarefa 2 : Extração de dados

Analise a página da web no URL fornecido:

```
https://web.archive.org/web/20230908091635/https://en.wikipedia.org/wiki/List_of_largest_banks
```

Identifique a posição da tabela necessária sob o título `Por capitalização de mercado`. Escreva a função `extract()` para recuperar as informações da tabela para um dataframe do Pandas.

Nota: Lembre-se de remover o último caractere do conteúdo da coluna `Market Cap`, como `\n`, e converter o valor para o formato float.

Escreva uma chamada de função para `extract()` e imprima o dataframe retornado.

Faça a entrada de log relevante.

Execute o código usando o comando:

```
python3.11 banks_project.py
```

Pergunta do quiz:

Ao inspecionar a página da web, observe os atributos dos dados na primeira linha. Haverá uma pergunta do quiz baseada nesses atributos.

Prompt da tarefa avaliada por pares:

Tire uma captura de tela do código HTML da tabela, obtido ao inspecionar a página da web. Certifique-se de que o conteúdo de pelo menos a primeira linha da tabela, conforme inserido no código HTML, esteja completamente visível. Salve esta captura de tela em sua máquina local como `Task_2a_extract.png`.

Tire uma captura de tela do código, conforme criado para a função `extract()` e salve-a em sua máquina local como `Task_2b_extract.png`.

Tire uma captura de tela da saída, conforme obtida na execução no terminal, e salve-a em sua máquina local como `Task_2c_extract.png`.

Tarefa 3 : Transformação de dados

A função Transform precisa realizar as seguintes tarefas:

1. Ler o arquivo CSV da taxa de câmbio e converter o conteúdo em um dicionário, de modo que o conteúdo da primeira coluna seja a chave do dicionário e o conteúdo da segunda coluna sejam os valores correspondentes.

▶ Clique aqui para dica

2. Adicione 3 colunas diferentes ao dataframe, a saber, `MC_GBP_Billion`, `MC_EUR_Billion` e `MC_INR_Billion`, cada uma contendo o conteúdo de `MC_USD_Billion` escalado pelo fator de taxa de câmbio correspondente. Lembre-se de arredondar os dados resultantes para 2 casas decimais.

Uma declaração de exemplo está sendo fornecida para adicionar a coluna `MC_GBP_Billion`. Você pode usar isso para adicionar as outras duas declarações por conta própria.

```
df['MC_GBP_Billion'] = [np.round(x*exchange_rate['GBP'],2) for x in df['MC_USD_Billion']]
```

Escreva a chamada da função para `transform()` e imprima o conteúdo do dataframe retornado. Comente todas as instruções de impressão anteriores.

Faça a entrada de log relevante e execute o código.

Pergunta do quiz:

1. Experimente a instrução fornecida para adicionar as colunas transformadas ao dataframe. Haverá uma pergunta sobre isso no quiz.
2. Imprima o conteúdo de `df['MC_EUR_Billion'][4]`, que é a capitalização de mercado do 5º maior banco em bilhões de EUR. Observe este valor, pois será a resposta para uma pergunta no quiz final.

Prompt da tarefa avaliada por pares:

Tire uma captura de tela do código, conforme criado para a função `transform()`, e salve-o em sua máquina local como `Task_3a_transform.png`.

Tire um instantâneo da saída e salve-o como `Task_3b_tranform.png`.

Tarefa 4: Carregando para CSV

Escreva a função para carregar o DataFrame transformado em um arquivo CSV, como `load_to_csv()`, no caminho mencionado no cenário do projeto.

Faça a entrada de log relevante.

Prompt da tarefa avaliada por pares:

Dê um clique duplo no arquivo CSV criado na aba `Explorer` na barra lateral esquerda do painel de programação no Cloud IDE. Observe que seu conteúdo é exibido na tela do editor. Tire uma captura de tela dessa tela e salve-a como `Task_4_CSV.png`.

Tarefa 5: Carregando para o Banco de Dados

Escreva a função para carregar o DataFrame transformado em um banco de dados SQL, como `load_to_db()`. Use os nomes do banco de dados e da tabela conforme mencionado no cenário do projeto.

Antes de chamar essa função, inicie a conexão com o servidor de banco de dados SQLite3 com o nome `Banks.db`. Passe este objeto de conexão, juntamente com o nome da tabela necessária `Largest_banks` e o DataFrame transformado, para a função `load_to_db()` na chamada da função.

Faça a entrada de log relevante.

Após a chamada bem-sucedida da função, você terá carregado o conteúdo da tabela com os dados necessários e o arquivo `Banks.db` estará visível na aba `Explorer` do IDE sob a pasta `project`.

Prompt de tarefa corrigida por pares:

Tire uma única captura de tela do código, conforme criado para as funções `load_to_csv()` e `load_to_db()`, e salve-a em sua máquina local como `Task_4_5_save_file.png`.

Tarefa 6: Função para Executar consultas no Banco de Dados

Escreva a função `run_queries()` que aceita a instrução de consulta e o objeto de Conexão SQLite3, e gera a saída da consulta. A instrução de consulta deve ser impressa juntamente com a saída da consulta.

Execute 3 chamadas de função usando as consultas conforme mencionado abaixo.

1. Imprima o conteúdo de toda a tabela

Instrução de consulta:

```
SELECT * FROM Largest_banks
```

2. Imprima a capitalização de mercado média de todos os bancos em bilhões de USD.

Declaração da consulta:

```
SELECT AVG(MC_GBP_Billion) FROM Largest_banks
```

```
SELECT name FROM banks ORDER BY ranking LIMIT 5;
```

```
SELECT Name from Largest_banks LIMIT 5
```

Registre a entrada de log relevante.

Prompt de tarefa avaliada por colegas:

Tire uma captura de tela da saída e salve-a como `Task_6_SQL.png`. Por favor, ajuste o tamanho do prompt do terminal para tirar uma única captura de tela que capture todas as três saídas juntas.

Prompt da pergunta do quiz:

Haverá uma pergunta no quiz sobre a saída dessas consultas.

Tarefa 7: Verificar entradas de log

Após atualizar todas as chamadas da função `log_progress()`, você deve executar o código para uma execução final. No entanto, você primeiro terá que remover o arquivo `code_log.txt`, que teria sido criado e atualizado ao longo das múltiplas execuções do código neste laboratório. Você pode remover o arquivo usando o seguinte comando em um terminal.

```
rm code_log.txt
```

Uma vez que o arquivo existente foi removido, agora execute a execução final. Após a conclusão bem-sucedida da execução, abra o arquivo `code_log.txt` clicando nele na aba `Explorer` da barra de ferramentas no lado esquerdo do painel de programação do IDE, sob a pasta `project`. Você deve ver todas as entradas relevantes feitas no arquivo de texto em relação às etapas da execução do código.

Prompt de tarefa avaliada por pares:

Tire uma captura de tela do conteúdo do arquivo e salve-a como `Task_7_log_content.png`.

Conclusão

Parabéns por concluir este projeto!

Com isso, você agora está treinado para realizar operações de ETL em dados do mundo real e tornar as informações processadas disponíveis para uso posterior em diferentes formatos.

Agora você deve ser capaz de:

- Usar técnicas de Webscraping para extrair informações de qualquer site conforme necessário.
- Usar data frames e dicionários do Pandas para transformar dados conforme necessário.
- Carregar as informações processadas em arquivos CSV e como tabelas de banco de dados.
- Consultar as tabelas do banco de dados usando as bibliotecas SQLite3 e pandas.
- Registrar o progresso do código adequadamente.

Author(s)

Abhishek Gagneja