IDC 409 Project-1

Web scraping and RDBMS

Submitted by - Rimjhim Goel(MS18133) and Chhavi Chahar(MS18136)
Contribution - MS18133(50%) and MS18136(50%)

Aim - To fetch data from a website and push it into a MySQL database.

Source of data - In this project, we have made a database of Goodreads' horror book reccommendations for Halloween. There are 22 books in that list and the data includes title of the book, the author, average rating of the book and the total number of ratings.
The URL is:
"https://www.goodreads.com/list/show/167306.New_horror_for_your_Halloween_reading"

Approach to the problem: The problem is divided in two steps-

1. To scrape relevant data for each book from the URL.
2. To dump that data in a MySQL table under specific column names

To scrape relevant data for each book from the URL-

We have used 'requests' for making the HTTP requests and 'BeautifulSoup' for parsing the data.

By inspection of the page source, we know that the the information regarding each books is stored in <tr> tags.

An example is shown below for the book 'Revelator'

Thus, we fetch all the <tr> tags from the HTML.

For storing the data for each book, we created a dictionary named bookList with 3 keys: Title, Author, Average rating and total ratings.

By inspection of the page we then found out the subtags, class and attributes under which these 3 values are stored for each book and appended them in our dictionary.

We then created a pandas dataframe form our dictionary s that it is easier to push the data to MySQL.

<u>To dump that data in a MySQL table under specific column names:</u>

We used sqlalchemy to establish a connection between MySQL and python by providing the host, user and password. sqlalchemy created an engine between python and a database *'mydatabase'* already present in MySQL.

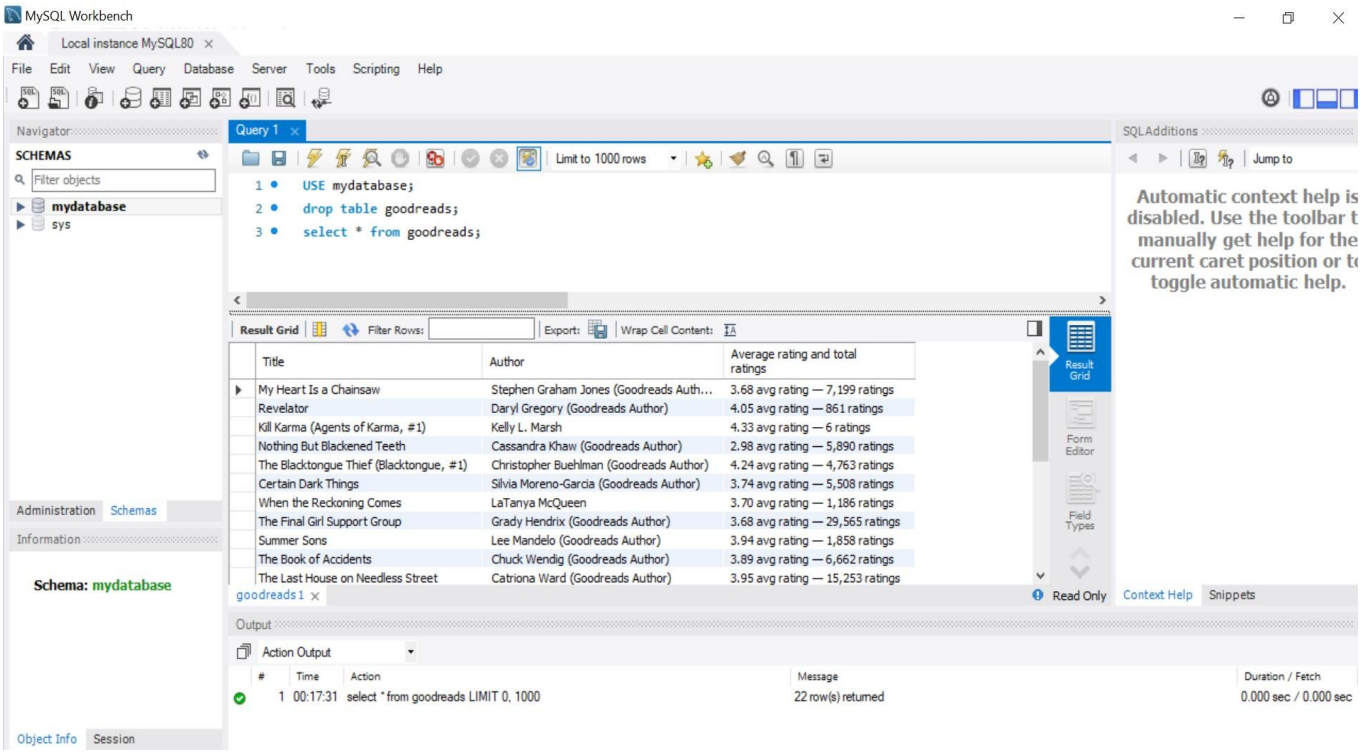
Then we pushed the pandas dataframe into a MySQL table called *'goodreads'*.

Results:

The python code for the problem is as follows -

```
import requests
# requests - library for making HTTP requests
from bs4 import BeautifulSoup
#BeautifulSoup - for parsing the data
import pandas as pd
# pandas - to convert the data scraped from the website into a dataframe
from sqlalchemy import create_engine
#sqlalchemy - to push the scraped data to mysql

URL = 'https://www.goodreads.com/list/show/167306.New_horror_for_your_Halloween_reading'
# URL from which data is to be fetched.
p = requests.get(URL)
# requests.get sends HTTP get request to the URL and recieves data
s = BeautifulSoup(p.content, "html.parser")
#passes p as p.content for decoding, the parser used is HTML parser
books = s.find_all("tr")
#Inspection of the page source tells that the information of each book is stored under tr tag
#find_all fetches information from tr tags from the page.

bookList = {'Title':[], 'Author':[], 'Average rating and total ratings':[]}
# we create a dictionary to store the relevant information from the HTML.
# We have 3 keys(=3 columns in the database): Title, Author, Average rating and total ratings
for book in books:
    bookList['Title'].append(book.find("span", {"itemprop":"name"}).text)
    bookList['Author'].append(book.find("span", {"itemprop":"author"}).text)
    bookList['Average rating and total ratings'].append(book.find("span", class_ = "minirating").text)
# for each book on the webpage,by inspection, we find the tags under which the title, author and ratings are present
# and append the information in the respective keys

dataFrame = pd.DataFrame.from_dict(data=bookList)
# create the pandas dataframe from the  bookList dictionary

connection=create_engine("mysql+mysqldb://root:123ds@localhost/mydatabase")
# create an engine to connect the 'mydatabase' database in MySQL to python
dataFrame.to_sql(con=connection, name='goodreads', if_exists='replace', index=False)
# a table named 'goodreads' is created in mydatabase
# the pandas dataframe is pushed to 'goodreads' table
# if exists = 'replace' replaces data previously present in the table,if any.
```

We ran a query on MySQL workbench to see the results -



The table from MySQL workbench was exported as an Excel spreadsheet -

	Title	Author	Average rating and total ratings
2	My Heart Is a Chainsaw	Stephen Graham Jones (Goodreads Author)	3.68 avg rating — 7,199 ratings
3	Revelator	Daryl Gregory (Goodreads Author)	4.05 avg rating — 861 ratings
4	Kill Karma (Agents of Karma, #1)	Kelly L. Marsh	4.33 avg rating — 6 ratings
5	Nothing But Blackened Teeth	Cassandra Khaw (Goodreads Author)	2.98 avg rating — 5,890 ratings
6	The Blacktongue Thief (Blacktongue, #1)	Christopher Buehlman (Goodreads Author)	4.24 avg rating — 4,763 ratings
7	Certain Dark Things	Silvia Moreno-Garcia (Goodreads Author)	3.74 avg rating — 5,508 ratings
8	When the Reckoning Comes	LaTanya McQueen	3.70 avg rating — 1,186 ratings
9	The Final Girl Support Group	Grady Hendrix (Goodreads Author)	3.68 avg rating — 29,565 ratings
10	Summer Sons	Lee Mandelo (Goodreads Author)	3.94 avg rating — 1,858 ratings
11	The Book of Accidents	Chuck Wendig (Goodreads Author)	3.89 avg rating — 6,662 ratings
12	The Last House on Needless Street	Catriona Ward (Goodreads Author)	3.95 avg rating — 15,253 ratings
13	All's Well	Mona Awad (Goodreads Author)	3.64 avg rating — 3,766 ratings
14	The Death of Jane Lawrence	Caitlin Starling (Goodreads Author)	3.34 avg rating — 4,025 ratings
15	Later	Stephen King (Goodreads Author)	4.04 avg rating — 70,384 ratings
16	A Dowry of Blood	S.T. Gibson (Goodreads Author)	4.25 avg rating — 5,559 ratings
17	Chasing the Boogeyman	Richard Chizmar (Goodreads Author)	4.07 avg rating — 6,078 ratings
18	The Taking of Jake Livingston	Ryan Douglass (Goodreads Author)	3.49 avg rating — 5,111 ratings
19	The Woods Are Always Watching	Stephanie Perkins (Goodreads Author)	3.12 avg rating — 2,614 ratings
20	White Smoke	Tiffany D. Jackson (Goodreads Author)	3.91 avg rating — 4,654 ratings
21	House of Hollow	Krystal Sutherland (Goodreads Author)	4.10 avg rating — 22,156 ratings
22	The Nameless Ones (Charlie Parker #19)	John Connolly (Goodreads Author)	4.32 avg rating — 2,027 ratings
23	The House of Dust	Noah Broyles (Goodreads Author)	3.49 avg rating — 165 ratings

Interpretation: We could successfully create a MySQL database from a website. Pushing the data to MySQL can be very helpful in managing large amount of tabular data.

Files attached:

1. Python code
2. Output Excel file exported from MySQL database