Assignment #3

CSCI 5408 (Data Management, Warehousing, Analytics)
Faculty of Computer Science, Dalhousie University

Date Given: Jun 22, 2022
Due Date: Jul 9, 2022 at 11:59 pm

Late Submissions are not accepted and will result in a late penalty of 10% deductions / day in the assignment.

Disclaimer: This assignment requires students to work on Spark framework for unstructured data processing, MongoDb for data storing, and Neo4j graph database for visualization. Submissions related to this assignment will not be used for commercial purposes.

Objective:

- The objective of this assignment is to understand Big Data processing problems, and NoSQL database (document, and graph).

Plagiarism Policy:

- This assignment is an individual task. Collaboration of any type amounts to a violation of the academic integrity policy and will be reported to the AIO.
- Content cannot be copied verbatim from any source(s). Please understand the concept and write in your own words. In addition, cite the actual source. Failing to do so will be considered as plagiarism and/or cheating.
- The Dalhousie Academic Integrity policy applies to all material submitted as part of this course. Please understand the policy, which is available at: https://www.dal.ca/dept/university_secretariat/academic-integrity.html

Assignment Rubric

	Excellent (25%)	Proficient (15%)	Marginal (5%)	Unacceptable (0%)	This Rubric Applied to
Completeness including Citation	All required tasks are completed	Submission highlights tasks completion. However, missed some tasks in between, which created a disconnection	Some tasks are completed, which are disjoint in nature.	Incorrect and irrelevant	Problem #1 Problem #2
Correctness	All parts of the given tasks are correct	Most of the given tasks are correct However, some portions need	Most of the given tasks are incorrect. The submission	Incorrect and unacceptable	Problem #1 Problem #2

		minor modifications	requires major modifications.		
Novelty	The submission contains novel contribution in key segments, which is a clear indication of application knowledge	The submission lacks novel contributions. There are some evidences of novelty, however, it is not significant	The submission does not contain novel contributions. However, there is an evidence of some effort	There is no novelty	Problem #1 Problem #2
Clarity	The written or graphical materials, and developed applications provide a clear picture of the concept, and highlights the clarity	The written or graphical materials and developed applications do not show clear picture of the concept. There is room for improvement	The written or graphical materials, and developed applications fail to prove the clarity. Background knowledge is needed	Failed to prove the clarity. Need proper background knowledge to perform the tasks	Problem #1 Problem #2

Citation:
McKinney, B. (2018). The impact of program-wide discussion board grading rubrics on students' and faculty satisfaction. Online Learning, 22(2), 289-299.

This assignment requires you to submit programming codes on gitlab, and a PDF file(s) on Brightspace.

Problem #1: This problem requires you to develop Data Extraction and Filtration Engine using Java (core Java and libraries mentioned in given API Documentation) and Store data in NoSQL.

Data Extraction and Preprocessing Engine: These following tasks are identical in terms of complexity. Code can be reused partially for all 3 problems.
Sources – (1) NEWS API (2) Open Movie Database (OMDb) (3) Reuter News Articles

(1) News Article Data Extraction & Transformation:

- Visit the news API https://newsapi.org/
- Create a developer account
- Write a Java program "***NewsExtraction***" (you may use the client library given in the API documentation) to search the following keyword (**case insensitive**)
  - "Canada", "University", "Dalhousie University", "Halifax", "Canada Education", "Moncton", "Toronto", "oil", "inflation".
  - Using your extraction engine code, write the returned result for each search term in a new text file. Note: If a keyword does not return any result, then you may just create an empty text file for that keyword.

- For this problem, you need to write one more Java program, "***NewsFiltration***", which will automatically clean and transform the data stored in the files, and then upload each record to new MongodB database **myMongoNews**
  - For cleaning and transformation -Remove special characters, URLs, emoticons etc.
  - Write your own regular expression logic. **You cannot use libraries such as, jsoup, JTidy etc.**

- You need to **include a flowchart/algorithm of your NEWS cleaning/transformation program on the PDF file.**

(2) Movie Data Extraction & Transformation:

- Visit the news API http://www.omdbapi.com/
- Request Free API key
- Write a Java program "***MovieDbExtraction***", and search the following keywords (case insensitive)
  - "Canada", "University", "Moncton", "Halifax", "Toronto", "Vancouver", "Alberta", "Niagara".
  - Using your extraction engine code, write the returned result for each search term in a new text file. **Note**: If a keyword does not return any result, then you may just create an empty text file for that keyword.

- For this problem, you need to write one more Java program, "***MovieDbFiltration***", which will automatically clean and transform the data stored in the files, and then upload each record to new MongodB database **myMongoMovieDb**
  - For cleaning and transformation -Remove special characters, URLs, emoticons etc.
  - Write your own regular expression logic. **You cannot use libraries such as, jsoup, JTidy etc.**

- You need to **include a flowchart/algorithm of your MovieDb cleaning/transformation program on the PDF file.**

(3) Reuter News Data Reading & Transformation:

- From the two given news files (reut2-009.sgm, and reut2-014.sgm), create MongoDb Database – **ReuterDb**, where each **Document** contains a news article. The task must be done using a Java Program "***ReutersRead***".
  - To perform this operation, you need to write a Java code to scan the required texts between two <REUTERS></ REUTERS > tags, <TEXT></ TEXT> tags, and <TITLE></ TITLE > tags.
  - In the **ReuterDb**, you may consider each news as a document. You can also include nested or sub-document.

```
{
        title: "",
        text: ""
}
```

- You need to **include a flowchart/algorithm of your Reuters Data cleaning/transformation program on the PDF file.**

Problem #2: This problem contains two tasks.

Task 1: Data Processing using Spark – MapReduce to perform count

- Using your GCP cloud account, configure and initialize Apache Spark cluster. (Follow the tutorials provided in Lab session).
- Create a flowchart or **write ½ page explanation** on how you completed the task, include this part in your PDF file.

  Note: If for some reason, you fail to work on GCP cloud account (valid reasons required), you need to create local standalone Hadoop/Spark cluster to perform the next set of operations.

- Write a MapReduce program **(WordCounter Engine**) to count (frequency count) the following substrings or words. Your MapReduce should perform the frequency count on the stored raw NEWS API files that you created for **Problem #1 – (1)**
  - **Keywords**: "Canada", "University", "Dalhousie University", "Halifax", "Canada Education", "Moncton", "Toronto", "oil", "inflation".
  - You need to include a flowchart/algorithm of your MapReduce program on the PDF file.
  - In your PDF file, report the words that have highest and lowest frequencies.

Task 2: Neo4J Database

- Explore the Neo4j graph database, understand the concept, and learn the cypher query (how to create a node, build relationship, add properties, and use find query etc.).
- Write a summary within 1-page - explaining what you learned, and how you will use Neo4j for any of your future project(s). To highlight your learning, you may also wish to create a small example graph using Neo4j.

Assignment 3 Submission Format:

1) Compress all your reports/files into a single .zip file and give it a meaningful name or ask your TA for naming format.

2) Submit your reports only in PDF format.

Please avoid submitting .doc/.docx and submit only the PDF version. You can merge all the reports into a single PDF or keep them separate. **You should also include output (if any) and test cases (if any) in the PDF file.**

3) Your Java code needs to be submitted on gitlab