Homework Assignment #2

Big Data Systems (2023A)

Objectives

- Basic CQL programming
- Basic Cassandra modeling

Submission requirements

Please read carefully and submit the exact required format.

You should create 1 zip file named **hw2.zip** which contains the following files:

- students.txt - a text file containing your names and ids
- astradb - a folder with the following files
  - GeneratedToken.csv - Application token ("Administrator User") generated from AstraDB
  - astradb.zip - "Secure Connect Bundle" from AstraDB (rename the file to astradb.zip)
- src - a folder contains all the JAVA source files
- runme.jar - an executable of your implementation that starts HW2CLI

You should also provide Nadav (nadavmagar@mail.tau.ac.il) Administrator user access

Assignment requirements

The objective is to ingest a dataset contains user reviews scrapped from Amazon.com and to support some basic queries.

To simplify things, you are not required to create any UI. Instead, you would need to implement some functions (`HW2API`) that would be executed by a CLI which is also provided.

Please note:

- You should only change the class `HW2StudentAsnwer.java` under the `bigdatacourse.hw2.studentcode` package.
- You can add more classes under that package but it is not required
- **DO NOT change any of the code under** `bigdatacourse.hw2` (holds the API and the CLI logic)
- The first two functions (`connect()` and `close()`) are already implemented
- You do not need to save all the parsable data - only what is needed to implement the APIs

AstraDB

Please create a free account for AstraDB as shown in class for this submission.

You should NOT enter any credit card information. Please use the free tier!

When creating a “Database”:

- set “bigdatacourse” as the keyspace
- select GCP (Google Cloud) Belgium region as the provider

For this assignment you would need to download 2 files from the AstraDB website:

- "Secure Connect Bundle” (a zip file you should rename to astradb.zip). It is available via the “Connect“ tab in Astra DB Dashboard
- Application token (select and “Administrator User” role) It is available via the “Token Management” under the “Organization Settings”.
  Note - you should download a “CSV” file. If you are downloading a JSON file, create the token from the “Organization Settings” and not from the “Create Database” template.

Reviews Dataset

The dataset consist of a set of office products available for purchase at Amazon.com and their matching reviews given by amazon’s users.

To stay with the limits of the free AstraDB version, you are given a subset of the original dataset.

The files are easily parsed with JSON. It is recommended to view some examples from each file before you start programming.

Some of the attributes are missing from the data. In such cases, please enter the “NOT_AVAILABLE_VALUE” const found in the source code. Do NOT enter “null”s.

General notes and tips

- Before you start coding, make sure you set your environment (Eclipse) correctly. This means you should create a project, copy the src and lib files, link them, and run successfully the code examples shown in class (`CassandraExample.java`)

- HW2CLI requires 2 parameters to be passed:
  - (1) the location of astradb folder
    (see “submission requirement” section for more info
  - (2) the location of ml-1m folder.
    (this is the directory of the extracted dataset files)
  - When developing, you can set the program’s argument in Eclipse.
    Make sure the CLI starts before you start coding

- Before you create the tables, understand the query requirements. The design should be based on them.

- **Read the comments in the code - it contains more requirements (for example, the print format required)**
- AstraDB has a rate limit of about 4k operations per second.
  For ingesting the data, you cannot just use “executeAsync” in a loop as you would hit the rate limit. NOTE - you won’t see the errors as the request are async, and the data won’t be added.
  Instead, you would need to throttle your ingestion.
  - Tip (1) - assuming the datacenter is in Europe which the ping is about 70ms, you can use 250 threads without hitting the limit.
  - Tip (2) - you should use the JSON.org package to parse the json strings easily. The source files are provided and also a JSONExamples class with the necessary examples.
  - Tip (3) - for sanity check, you can run “select count(*) from …” in CQLSH (although this is an anti pattern to use ‘select count’ on all partitions) to verify you have 134837 items entires and 1243186 reviews.
    - With high probability the “select count(*)” will failed due to timeout. This is expected as even with this small data set Cassandra has too many partition to evaluate. In order to make it work, you can “divide” the query into several smaller queries by the token range of the partition key (this is the “hash range” of Dynamo.
    - Assuming Cassandra uses 64bit to define the range, we get possible tokens from -2^63 (-9223372036854775808) to 2^63 -1 (9223372036854775807)
    - Using the data model we defined, it is sufficient to “cut” the range into 2 parts. That is we ran 2 queries:
      ```
      select count(*) from <table> where token(<partition attr>) >=0
      select count(*) from <table> where token(<partition attr>) <0
      ```
- Finally, the generated JAR (runme.jar) should include all the lib files
  (“Extract required libraries into generated JAR” option when exporting the JAR)
  Please test and make sure the runme.jar works.

Good luck :)