HW1 – Due by 11:59pm on Friday, September 27, 2019

PART 1 - READING ASSIGNMENT
[https://www.safaribooksonline.com/library/view/mongodb-in-action/9781617291609/]
Chapter 3. Writing programs using MongoDB
Chapter 4. Document-oriented data

PART 2 - READING ASSIGNMENT (Papers Attached)
Read the following papers and provide a short summary for each paper.
- Application of NoSQL Database in Web Crawling
- Comparing NoSQL MongoDB to an SQL
- Data Aggregation System

PART 3 - PROGRAMMING ASSIGNMENT
Create a collection called 'games'. We're going to put some games in it.
Add 5 games to the database. Give each document the following properties: name, genre, rating (out of 100)
If you make some mistakes and want to clean it out, use remove() on your collection.
Write a query that returns all the games.
Write a query to find one of your games by name without using limit().
Use the findOne method.
Look how much nicer it's formatted!
Write a query that returns the 3 highest rated games.
Update your two favorite games to have two achievements called 'Game Master' and 'Speed Demon', each under a single key.
Show two ways to do this.
Do the first using update() and do the second using save().
Hint: for save, you might want to query the object and store it in a variable first.
Write a query that returns all the games that have both the 'Game Master' and the 'Speed Demon' achievements.
Write a query that returns only games that have achievements.
Not all of your games should have achievements, obviously.
You could take the screenshots by pressing ALT + PRT SCRN or Snipping Tool every time you execute a command, and paste into a word document.
You could then submit this document.

PART 4 - PROGRAMMING ASSIGNMENT
Write a Java (could be a console app - will only run once to import the data into MongoDB) program to read the following file, and insert into 3 different collections (movies, ratings, tags).
- MovieLens 10M
Stable benchmark dataset. 10 million ratings and 100,000 tag applications applied to 10,000 movies by 72,000 users.
http://grouplens.org/datasets/movielens/
Once the data are inserted into MongoDB, do the followings using MapReduce:
Write a MapReduce to do the followings:
- Number of Movies released per year (Movies Collection)
- Number of Movies per genre (Movies Collection)
- Number of Movies per rating (Ratings Collection)
- Number of times each movie was tagged (Tags Collection)

PART 5- PROGRAMMING ASSIGNMENT
Write a Java (could be a console app - will only run once to import the data into MongoDB) program to read the **access.log** file (attached), and insert into access collection.
Once the data are inserted into MongoDB, do the followings using MapReduce:
- Number of times any webpage was visited by the same IP address.
- Number of times any webpage was visited each month

PART 6 - PROGRAMMING ASSIGNMENT
Execute 5 commands of your choice from each of the following groups, and paste the screenshots in a word document.

mongo> help	[5 commands]
mongo> db.help()	[5 commands]
mongo> db.mycoll.help()	[10 commands]

PART 7 - PROGRAMMING ASSIGNMENT
Write a .bat (for Windows) or .sh (for MacOS) to import the entire NYSE dataset (stocks A to Z) into MongoDB.
NYSE Dataset Link: http://msis.neu.edu/nyse/nyse.zip