CS 327E Final Project: Milestone 1

Prerequisites:

1. Completed Lab 3.
2. Continuing to work with your partner.
3. Installed Cygwin with open-ssh package if you are a Windows user.

Step 1. Create a new folder in your local git repository called **final-project**. All the work that your team will do for the Final Project of this course will go into this folder.

Step 2. Download the **movielens** dataset to your laptop from S3. The download link is: http://cs327e-fall2017-final-project.s3.amazonaws.com/movielens_dataset_with_header.zip. Open the files and become familiar with the structure of this dataset. This dataset will be used in the first two milestones of the final project.

Step 3. Download the **movielens_ratings.py** file to your laptop from our snippets repo. The download link is: https://github.com/cs327e-fall2017/snippets/blob/master/movielens_ratings.py. This is the Spark program that we will work with in this milestone. Open the program in an editor of your choice and read through the code. Look up the RDD operation that you don't understand by consulting our lecture slides and notes, text, and the Spark Programming Guide [3].

Step 4. Open a **psql** session and connect to your Postgres RDS. Add a new column to the `Title_Ratings` table in your IMDB database. The column should be named `movielens_rating` and of type `numeric(3,1)`. Do not create any constraints on this column. The `ALTER TABLE` documentation [4] in the Postgres manual has an example on how to add a column to an existing table.

Step 5. Follow the Spark guides to create, connect to, and configure an EMR cluster:

I. https://github.com/cs327e-fall2017/snippets/wiki/Creating-First-EMR-Cluster
II. https://github.com/cs327e-fall2017/snippets/wiki/Connect-and-Configure-EMR-Cluster

Step 6. Follow the Spark guide to run the `movie_ratings.py` program on your EMR cluster: https://github.com/cs327e-fall2017/snippets/wiki/Running-Spark-Scripts

Step 7. Debug and fix any errors encountered during program execution. The Spark program should run without any errors if your EMR cluster is configured correctly.

Step 8. Connect to your IMDB database and find the number of records in the `Title_Ratings` table that have a non-NULL `movielens_rating` value. Copy the SQL query and output into a text file. Save the text file as `movielens_rating.out` and add it to your git repo.

Step 9. Go through the Spark program and add a comment above each RDD operation. The comment should describe the behavior of the RDD operation in a single line. If you are unsure of what a particular RDD operation is doing, you can print out the input RDD and output RDD using the `print_rdd()` function provided and examine the printed values. The print function writes out the contents of an RDD to a log file under `/home/hadoop/logs` directory. Once you have commented the RDD operations in the code, add your commented `movie_ratings.py` to your git repo.

Step 10. Write an aggregate query that accesses the newly populated `movielens_rating` column in some interesting way and wrap this query inside a view. Name this view `v_movielens_rating`. The view should be virtual if it executes in under 10 seconds or materialized if it runs for 10 seconds or longer. Create the view in your IMDB database and add the view definition to a file. Name the file `v_movielens_rating.sql` and add it to your git repo .

Step 11. Create a QuickSight analysis that visualizes the output from your view. Create a dashboard for the analysis and share it with the IAM admin user. Also, take a screenshot of the analysis and save it as a png or jpg format. Add the screenshot to your git repo.

Step 12. Locate the commit id that you will be using for your submission. This is a long 40-character that shows up on your main GitHub repo page next to the heading "Latest commit" (e.g. commit 6ca6f695bca36f7fc2c33485d1080ae30f8b9928). Locate the link to your GitHub repo (e.g. https://github.com/cs327e-fall2017/xyz.git where xyz is your repo name). Go back to your existing Stache entry and locate the read-only API endpoint and read key.

Replace the commit id, repo link, API endpoint, and read key in the json string below with your own:

```
{
  "repository-link": "https://github.com/cs327e-spring2017/xyz.git",
  "commit-id": "6ca6f695bca36f7fc2c33485d1080ae30f8b9928",
  "stache-endpoint": "/api/v1/item/read/61515",
  "stache-read-key": "b2eacb0387a919e33b27e7c03a6c5d84b71234795732be33eb28711ec16f0e21"
}
```

Create a submission.json file that contains your modified json string. Click on the Final Project Milestone 1 in Canvas and upload submission.json. **Do not add submission.json to your git repo**.

This submission is due by **Friday, 11/03 at 11:59pm**. If it's late, there will be a 10% grade reduction per late day. This late policy is also documented in the syllabus.

Additional Notes:

- The EMR service is not covered by the free tier and it costs $8/hour for a single-node cluster on an m3.xlarge instance. The charges apply even when the cluster is idle and there is no option to stop and restart an EMR cluster. Therefore, you should always **terminate** your EMR cluster when it is not

in use. Follow the steps in this guide to terminate your EMR cluster: https://github.com/cs327e-fall2017/snippets/wiki/Terminating-EMR-cluster.

- When you need to create a new EMR cluster, you can simply **clone** a previously terminated cluster in order to speed up the cluster creation process. Follow the steps in this guide to clone an EMR cluster: https://github.com/cs327e-fall2017/snippets/wiki/Clone-EMR-Cluster

References and Additional Resources:

[1] Snippets Wiki: https://github.com/cs327e-fall2017/snippets/wiki
[2] Final Project Milestone 1 Grading Rubric: http://www.cs.utexas.edu/~scohen/projects/m1-rubric.pdf
[3] Spark Programming Guide: https://spark.apache.org/docs/2.2.0/rdd-programming-guide.html
[4] Postgres `ALTER TABLE` documentation: https://www.postgresql.org/docs/9.6/static/sql-altertable.html