School of Computer Science

COMP47470

Project 1
Command Line Interface and Data Management Systems (for Big Data)

Teaching Assistant:	Leandro Almeida
Coordinator:	Dr Anthony Ventresque
Date:	Thursday 21st February, 2019
Total Number of Pages:	3

General Instructions

- This project consists in using Bash scripts to write Big Data jobs and manage database management systems (MySQL and MongoDB). You can use the DBMS set up seen in lab 3 and lab 4 (through csserver.ucd.ie) or your own machine.
- You are encouraged to collaborate with your peers on this project, but all written work must be your own. In particular we expect you to be able to explain every aspect of your solution if asked.
- We ask you to hand in an archive (zip or tar.gz) of your solution: code/scripts and a 5-10 page pdf report of your work.
- The report should include the following sections:
  1. a short introduction
  2. a requirement section that answers the question *what* is the system supposed to do
  3. an architecture/design section that answers the question *how* your solution has been designed to address the requirements described in the previous section
  4. a series of sections that describe the different challenges you faced and your solutions. For instance, take one of the script, describe the difficulty you faced and your solution. These sections can be short – the objective here is to show how you crafted the solutions with the tools you have learnt so far.
  5. a short conclusion
- The breakdown of marks for the project will be as follows:
  - question 1: 35%
  - question 2: 40%
  - Report: 25%
- **Due date: 7/03/2019**

1 Big Data jobs in Bash

Download the following 6 books:

- `http://www.gutenberg.org/files/1524/1524-0.txt`
- `http://www.gutenberg.org/files/1112/1112.txt`
- `http://www.gutenberg.org/files/2267/2267.txt`
- `http://www.gutenberg.org/files/2253/2253.txt`
- `http://www.gutenberg.org/files/1513/1513-0.txt`
- `http://www.gutenberg.org/files/1120/1120.txt`

1. parse the text of these files to remove the punctuation; save this new version in a file (the command `tr` could be useful).
2. list the 20 most frequent words from your text
3. remove the stop words (i.e., the words with little or no real meaning); use a list of stop words from the web (e.g., one of these lists).
4. list the 20 most frequent words excluding stop words

2 Performance Testing of the Two Database Systems

We will use the database found here: https://dev.mysql.com/doc/employee/en/employees-introduction.html. It's a relatively large database (4 millon records, 160MB).

1. download the GitHub repo using the git command

```
git clone https://github.com/datacharmer/test_db
```

2. modify the employee.sql file to change the name of the database to a the name of a database you can create yourself (remember what Leandro said in the practicals) - use sed for it.
3. Then run

```
mysql < employees.sql
```

4. check the database.

Now we want to create a similar database in MongoDB. Remember the SQL to MongoDB correspondences (more info here: https://docs.mongodb.com/manual/reference/sql-comparison/).

1. write a Bash script that would read the stucture of the MySQL database and create a MongoDB collection for each table in the MySQL server
2. for each of these collections, read the content of the corresponding table in MySQL and populate the MongoDB collection.

Now we want to do some performance comparison of the two database systems.

1. write a simple and a complex (e.g., joint query) queries in SQL and MongoDB. Test them on your databases.
2. write a Bash script that will execute each query and measure the time taken by each query - you can use the following bash variables to measure time:

```
START_TIME=$SECONDS
dosomething
ELAPSED_TIME=$(($SECONDS - $START_TIME))
echo $ELAPSED_TIME
```

3. What do you observe?
4. Do you think this was a good way to compare the performance of these two data management systems?