**Big Data Systems CA1**

For this CA I would like you to compare and contrast the two databases that we have covered so far – Mongo and PostgreSQL. In order to do this, you should insert data into a Mongo collection using steps 1 – 7 detailed below. Then you should detail how you would store this data if you were to use PostgreSQL. You will have to create the tables and schema necessary to store the data in PostgreSQL. Once you have stored the data using both you should reflect on the pros and cons of each for this data.

1. **Getting started**

Connect to a running mongo instance, use a database named `bigdata\_ca1\_firstname\_lastname`. Use mongodump to submit a copy of your database when you have finished the assignment.

Document all your queries in a javascript file for submission. Where you are asked a question with a number document your answers to these questions as well as the code needed to find the answer.

1. **Insert Documents**

Insert the following documents into a `movies` collection.

title : Fight Club

writer : Chuck Palahniuk

year : 1999

actors : [

Brad Pitt

Edward Norton

]

title : Pulp Fiction

writer : Quentin Tarantino

year : 1994

actors : [

John Travolta

Uma Thurman

]

title : Inglorious Basterds

writer : Quentin Tarantino

year : 2009

actors : [

Brad Pitt

Diane Kruger

Eli Roth

]

title : The Hobbit: An Unexpected Journey

writer : J.R.R. Tolkein

year : 2012

franchise : The Hobbit

title : The Hobbit: The Desolation of Smaug

writer : J.R.R. Tolkein

year : 2013

franchise : The Hobbit

title : The Hobbit: The Battle of the Five Armies

writer : J.R.R. Tolkein

year : 2012

franchise : The Hobbit

synopsis : Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.

title : Pee Wee Herman's Big Adventure

title : Avatar

1. **Query / Find Documents**

query the `movies` collection to

1. get all documents

2. get all documents with `writer` set to "Quentin Tarantino"

3. get all documents where `actors` include "Brad Pitt"

4. get all documents with `franchise` set to "The Hobbit"

5. get all movies released in the 90s

6. get all movies released before the year 2000 or after 2010

1. **Update Documents**

1. add a synopsis to "The Hobbit: An Unexpected Journey" : "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."

2. add a synopsis to "The Hobbit: The Desolation of Smaug" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."

3. add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"

1. **Text Search**

1. find all movies that have a synopsis that contains the word "Bilbo"

2. find all movies that have a synopsis that contains the word "Gandalf"

3. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf" 4. find all movies that have a synopsis that contains the word "dwarves" or "hobbit"

5. find all movies that have a synopsis that contains the word "gold" and "dragon"

1. **Delete Documents**

1. delete the movie "Pee Wee Herman's Big Adventure"

2. delete the movie "Avatar"

1. **Relationships**

Insert the following documents into a `users` collection

username : GoodGuyGreg

first\_name : "Good Guy"

last\_name : "Greg"

username : ScumbagSteve

full\_name :

first : "Scumbag"

last : "Steve"

Insert the following documents into a `posts` collection

username : GoodGuyGreg

title : Passes out at party

body : Wakes up early and cleans house

username : GoodGuyGreg

title : Steals your identity

body : Raises your credit score

username : GoodGuyGreg

title : Reports a bug in your code

body : Sends you a Pull Request

username : ScumbagSteve

title : Borrows something

body : Sells it

username : ScumbagSteve

title : Borrows everything

body : The end

username : ScumbagSteve

title : Forks your repo on github

body : Sets to private

Insert the following documents into a `comments` collection

username : GoodGuyGreg

comment : Hope you got a good deal!

post : [post\_obj\_id]

where [post\_obj\_id] is the ObjectId of the `posts` document: "Borrows something"

username : GoodGuyGreg

comment : What's mine is yours!

post : [post\_obj\_id]

where [post\_obj\_id] is the ObjectId of the `posts` document: "Borrows everything"

username : GoodGuyGreg

comment : Don't violate the licensing agreement!

post : [post\_obj\_id]

where [post\_obj\_id] is the ObjectId of the `posts` document: "Forks your repo on github"

username : ScumbagSteve

comment : It still isn't clean

post : [post\_obj\_id]

where [post\_obj\_id] is the ObjectId of the `posts` document: "Passes out at party"

username : ScumbagSteve

comment : Denied your PR cause I found a hack

post : [post\_obj\_id]

where [post\_obj\_id] is the ObjectId of the `posts` document: "Reports a bug in your code"

1. **Querying related collections**

1. find all users

2. find all posts

3. find all posts that was authored by "GoodGuyGreg"

4. find all posts that was authored by "ScumbagSteve"

5. find all comments

6. find all comments that was authored by "GoodGuyGreg"

7. find all comments that was authored by "ScumbagSteve"

8. find all comments belonging to the post "Reports a bug in your code"

1. **PostgreSQL**

Now you should detail how you would carry out steps 2-8 if you were to use PostgreSQL. You should create the necessary tables, insert, find, update, delete the data and add relationships between tables.

Once you have done this reflect on the pros and cons of Mongo and PostgreSQL for this scenario.

1. **Screencast**

Produce a screencast detailing the work you have done in this CA. You can use https://obsproject.com/ to produce the screencast. Upload the final screencast to Youtube and make sure that you have the permissions set to unlisted or public so that I can view it. The sceeencast should be 5-10 minutes in length. If you exceed the 15 minutes it is possible

to do a second screencast and send me both links. In the screencast you should cover the following:

1. Show all collections in the database.

2. Discuss the queries necessary to answer all of the question and insert data into the collections. 3. Discuss the answers to all numbered questions.

4. Discuss the schema and queries necessary to store and access this data in PostgreSQL.

5. Discuss the differences between MongoDB and PostgreSQL.

**9. Submission**

Place all of your work in a folder called ca1\_yourfirstname\_yourlastname, zip all of the files together into a zip file called ca1\_yourfirstname\_yourlastname.zip and submit on Moodle. The files that need to be in the folder are:

1. A dump of your databases

2. A javascript file with all queries used in the assignment

3. A file containing the answers to all questions in the assignment. Include SQL queries for the PostgreSQL section here.

4. A link to your screencast in a file called screencasturl.txt and a link to a Bitbucket code repo with all of the files that you submit

This CA is due Friday November 10th at 23:55 and is worth 15% of your grade for this course.