Comp 5338 – Assignment: Polyglot Persistence with NoSQL Systems

# Section 1 – Introduction

This report will briefly introduce the schema designs for both MongoDB and neo4j storage systems for this assignment, then followed by the query design and execution using command line tool to fetch the correct result given simple queries and analytics queries, at last, this report will compare these two storage systems with their pros and cons regarding the query performances, data schema design etc. You can also find the source code in the following github link:

<https://github.com/MingxuanLi/comp5338-polygot-persistence-systems>

# Section 2 – Schema Design

The following diagram briefly describe the data model schema and their relationships in both MongoDB and neo4j.

## Section 2.1 – MongoDB Schema Design

[[Insert the MongoDb diagram]]

For MongoDB part, we use mongoose to explicitly design the schema in the javascript files, the schemas are under ***src/mongo-schemas***.

## Section 2.2 – neo4j Schema Design

[[Insert the neo4j diagram]]

For neo4j part, all the schemas for loading the data are in ***src/neo4j-helper.js*** file.

Section 3 – Query Design and Execution

## Section 3.1 – Query Design

[[description to query design]]

## Section 3.2 – Query Execution

There are two major execution steps:

* Loading data
* Executing Queries

First, you need to use Git to check this repo using the following command:

***git clone*** [***https://github.com/MingxuanLi/comp5338-polygot-persistence-systems.git***](https://github.com/MingxuanLi/comp5338-polygot-persistence-systems.git)

After that you need to install the library dependencies we use to import and query data from MongoDB and neo4j, run the following:

***npm install***

After the installing finish, you can try loading the data into MongoDB by running following:

***npm run load -- --db=mongodb***

And neo4j loading command is this:

***npm run generate*** – (this does some conversion for the unix date format for neo4j)

***npm run load -- --db=neo4j***

After the loading is finished, then you can use the following query to fetch result.

For MongoDB: ***npm run query -- --db=mongodb --query={$name}***

For neo4j: ***npm run query -- --db=neo4j --query={$name}***

The variable $name can be **‘sq1’, ‘sq2’, ‘aq1’, ‘aq2’, ‘aq3’, ‘aq4’, ‘aq5\_1’, ‘aq5\_2’, ‘aq6’**.

sq – simple query, aq – analytic query.

The result of the query should be printed in the terminal console.

Section 4 – Comparison and Summary

References:

<https://docs.mongodb.com/manual/applications/data-models/>

<https://www.mongodb.com/blog/post/6-rules-of-thumb-for-mongodb-schema-design-part-1>

<https://www.mongodb.com/blog/post/6-rules-of-thumb-for-mongodb-schema-design-part-2>

<https://www.mongodb.com/blog/post/6-rules-of-thumb-for-mongodb-schema-design-part-3>

<https://neo4j.com/developer/javascript/>

<https://neo4j.com/developer/guide-data-modeling/>