CS 336 Mini-project

MongoDB Trip Advisor

Fall 2018

The purpose of this assignment is to get familiar with a NoSQL database called MongoDB, store data in MongoDB, and query them by using MongoDB’s query language. At the end you will do the same in MySQL and you will write down your experience from using these two diﬀerent in philosophy databases.

1 Logistics

The project is individual (no groups) and is due on Wednesday, December 12 at 11:55 pm on Sakai. You should submit a single document showing what you did and the results you obtained. You should be able to run a local instance of MongoDB and do all analysis locally, so no hosting, website, etc. should be necessary.

2 MongoDB

First download MongoDB:

https://www.mongodb.com/download-center/community

Then follow the installation instructions for your OS:

MacOS https://docs.mongodb.com/manual/tutorial/install-mongodb-on-os-x/ Linux https://docs.mongodb.com/manual/administration/install-on-linux/ Windows https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/

After you have successfully installed and started MongoDB you will be able to use it. Some commands that may be useful:

show dbs # Show all database instances in the database use cs336 # Create a new database instance

Mind that after executing these you are under the cs336 database instance. Under the database instance, create two collections:

db.createCollection("reviews") db.createCollection("test")

If you use the command show collections, you can ﬁnd two new collections have been created. You can check those two collections are all empty by using either the find() or count() command. The find() command will return all contents in the collection and count() will return the number of documents in the collection. For example: db.reviews.count() will return how many documents you have inside your reviews collection and if you replace count() with find() you will print the documents that exist in the collection. If you use the command pretty() after the find() command, you will see your json documents prettiﬁed. The results are as follows:

> db.createCollection("reviews") { "ok" : 1 } > db.createCollection("test") { "ok" : 1 } > db.reviews.count() 0 > db.reviews.find().pretty() >

After you have completed the installation of MongoDB, you be should be able to import the reviews ﬁle of tripAdvisor data.

3 Trip Advisor data

Download the data from Google Drive. The data has been cleaned (to an extent) for you, so you do not have to worry about it. The code used to clean up some parts is as below, you might want to modify it to get clean some other parts of the data 1 db . reviews . find () . forEach ( function (doc) { 2 doc . Reviews . forEach ( function ( review ) { 3 var newOverall = review . Ratings . Overall ; 4 var OverallFloat = parseFloat ( newOverall ) . toFixed (1) ; 5 review . Ratings . Overall = parseFloat ( OverallFloat ) ;

6 7 i f ( review . Ratings . Service ) { 8 var newService = review . Ratings . Service ; 9 review . Ratings . Service = parseInt ( newService ) ; 10 } 11 12 i f ( review . Ratings . Value) { 13 var newValue = review . Ratings . Value ; 14 review . Ratings . Value = parseInt (newValue) ; 15 }

2

16 17 i f ( review . Date) { 18 var newDate = review . Date ; 19 review . Date = new Date(newDate) 20 } 21 }) 22 db . reviews . save (doc) 23 })

Unzip the data reviews.tar.bz2 and import it into your MongoDB instance:

mongoimport --db mydb --collection reviews --file reviews.json

The review document will have the following format:

1 { 2 ” id ” : string 3 ”Reviews” : [{ 4 ”Ratings” : { 5 ” Service ” ( optional ) : numeric , 6 ” Cleanliness ” ( optional ) : numeric , 7 ”Overall” : numeric , 8 ”Value” ( optional ) : numeric , 9 ”Sleep Quality” ( optional ) : numeric , 10 ”Rooms” ( optional ) : numeric , 11 ”Location” ( optional ) : numeric 12 }, 13 ”AuthorLocation” : string , 14 ” Title ” : string , 15 ”Author” : string , 16 ”ReviewID” : string , 17 ”Content” : string , 18 ”Date” : ISODate () 19 }] , 20 ”HotelInfo” : { 21 ”Name” : string , 22 ”HotelURL” : string , 23 ”Price” : string , 24 ”Address” : string , 25 ”HotelID” : string , 26 ”ImgURL” : string 27 } 28 }

4 Review patterns

After you have successfully downloaded and stored reviews in your database you are ready to write some useful queries in order to ﬁnd interesting patterns in your reviews. You can provide graphs for your patterns if you want. You should have these queries (and more):

3

1. Find all reviews for a hotel ‘Desert Rose Resort’.

2. Number of ratings for each hotel. Sort the results.

3. Average overall ratings for each hotel. Sort the results

4. Show hotels with number of 5.0 overall ratings that they recieved.

5. Number of ratings given out per month/day of week.

6. Number of reviews per author.

You might notice that there are multiple hotels with the name ‘Desert Rose Resort’. In that case use the HotelID of ﬁrst ‘Desert Rose Resort’. Alternatively, you might also combine the 2 hotels into one - MongoDB provides an operator to do this. You can think of many more queries like that. What we expect from you is to submit a number of MongoDB queries you have written in order to ﬁnd interesting patterns in your data. You can just submit your queries you have written and the result you have gotten