

MongoDB Assignment

Chatzimoschou Angeliki

BAPTXXXX

Answers

Part 1

Use stu

load("/data/db/prep.js")

1) db.students.find({"courses.course\_status": "In Progress"}).count()

2) db.students.aggregate({$group : { \_id : "$home\_city", num\_of\_students : {$sum : 1}}})

3)db.students.aggregate(  
{$unwind: "$hobbies"},  
 {$group :{ \_id:"$hobbies", most\_popular:{$sum:1}}},  
 {$sort:{most\_popular:-1}},  
{$limit:1},{$project:{hobbies:1}})

4) db.students.aggregate(

{$unwind: "$courses"},

{$match:{"courses.course\_status" : "Complete"}},

{$group : {\_id : "$\_id", GPA : {$avg : "$courses.grade"}}},{$sort : {GPA : -1}},

{$project : {\_id:0, GPA : 1}},{$limit: 1})

5) db.students.aggregate(  
{$unwind: "$courses"},  
{$match: {"courses.grade" : 10 }},  
{$group : {\_id : "$\_id", grade : {$sum : "$courses.grade"}}},  
{$sort : {"grade" :-1}},   
{$project : {\_id:0, student : "$\_id", grade:1}},{$limit: 4})

6) db.students.aggregate(  
{$unwind: "$courses"},  
{$group : {\_id : "$courses.course\_code", avggrade : {$avg : "$courses.grade"}}},  
{$sort : {"avggrade" : -1}},   
{$project : {\_id:0, class: "$\_id"}},{$limit: 1})

7) db.students.aggregate(  
{$unwind: "$courses"},  
{$match: { "courses.course\_status" : "Dropped" }},  
{$group : {\_id : "$courses.course\_code", dropped : {$sum : 1}}},  
{$sort : {"dropped" : -1}},  
{$project : {\_id:0, class: "$\_id"}},{$limit: 1})

8) db.students.aggregate(  
 {$unwind:"$courses"},  
{$match: {'courses.course\_status':"Complete"}},   
{$group:{\_id: { ctype : { $substr: [ "$courses.course\_code", 0, 1 ] }}, classcount :{$sum:1}}},  
{$project: {\_id: 0, classInfo: "$\_id", classcount:1 }})

9) db.students.aggregate({ $project : {\_id : 1 , hobbies:1, hobbyist : { $gt: [ { $size: "$hobbies" }, 3]}}})

10)db.students.aggregate(  
{$unwind:"$courses"},

{$match: {'courses.course\_status':"Complete"}},

{$group:{\_id: "$\_id", count :{$sum:1}}},

{$project: {count : 1}})

11) db.students.aggregate(  
{$unwind: "$courses"},

{$group : {\_id: "$\_id",GPA : {$avg : "$courses.grade"},

classesInProgress: {$sum: { $cond: [{$eq: ["$courses.course\_status", "In Progress"]},1,0]}},

droppedClasses: {$sum: { $cond: [{$eq: ["$courses.course\_status", "Dropped"]},1,0]}},}},

{$project: {\_id:1, GPA:1,classesInProgress:1,droppedClasses:1}}).pretty()

12)db.students.aggregate(

{$unwind: "$courses"},

{$group : {\_id: "$courses.course\_code", course\_title: {$first: "$courses.course\_title"},

numberOfDropouts: {$sum: { $cond: [{$eq: ["$courses.course\_status", "Dropped"]},1,0]}},

numberOfTimesCompleted: {$sum: { $cond: [{$eq: ["$courses.course\_status", "Complete"]},1,0]}},

currentlyRegistered: {$addToSet: "$\_id"},

maxGrade: {$max: "$courses.grade"},

minGrade: {$min: "$courses.grade"},

avgGrade: {$avg: "$courses.grade"}}},

{$project: {course\_title:1,numberOfDropouts:1, numberOfTimesCompleted:1,

currentlyRegistered:1,maxGrade:1,minGrade:1,avgGrade:1}},{ $out : "classesInfo" })

Part 2

1) install.packages("rmongodb")

library(rmongodb)

2) mongo<- mongo.create()

3) mongo.is.connected(mongo)

4)(namespace <- paste("r","lab2",sep = "."))

5) JSON\_Cristiano<- '{"name":"Cristiano", "language":"Portuguese"}'

6)BSON\_Cristiano<-mongo.bson.from.JSON(JSON\_Cristiano)

ok<- mongo.insert(mongo,namespace,BSON\_Cristiano)

7) BSON\_Ioanna<-mongo.bson.from.list(list(name="Ioanna",language="English",age=34))

BSON\_Dimitris<-mongo.bson.from.list(list(name="Dimitris",language="Greek",age=29))

8) ok2<-mongo.insert.batch(mongo,namespace,

list(BSON\_Ioanna,BSON\_Dimitris))

ok2

*Commands in mongo*  
use r

db.lab2.find()

#*Results*  
{"\_id" : ObjectId("572b8bf2b1966d7894e2c66d"),"name" : "Cristiano","language" : "Portuguese"}

/\* 2 \*/

{ "\_id" : ObjectId("572b8c18b1966d7894e2c66e"), "name" : "Ioanna", "language" : "English",

"age" : 34}

/\* 3 \*/

{ "\_id" : ObjectId("572b8c18b1966d7894e2c66f"), "name" : "Dimitris", "language" : "Greek",

"age" : 29 }

9) ok3<-mongo.update(mongo,namespace, '{"name":"Cristiano"}',  
'{"name":"Cristiano", "language":"Portuguese","age":26}')

ok3

*Commands in mongo*

db.lab2.find({"name":"Cristiano"})

#*Results*  
{"\_id" : ObjectId("572b8bf2b1966d7894e2c66d"),"name" : "Cristiano","language" : "Portuguese",  
"age" : 26}

10) ok4<-mongo.remove(mongo,namespace,'{"name":"Dimitris"}')

ok4

*Commands in mongo*  
use r

db.lab2.find()

#*Results*  
{"\_id" : ObjectId("572b8bf2b1966d7894e2c66d"),"name" : "Cristiano","language" : "Portuguese"}

/\* 2 \*/

{ "\_id" : ObjectId("572b8c18b1966d7894e2c66e"), "name" : "Ioanna", "language" : "English",

"age" : 34}

11) persons<-list(mongo.bson.from.list(list(name="TamakoMisota", language="Japanese", age="30")),

mongo.bson.from.list(list(name="YuraKirkorov", language="Russian", age="26")))

ok5 <- mongo.insert.batch(mongo,namespace,persons)

cursor<- mongo.find(mongo, namespace)

current\_row\_number<- 0

name\_1<- NULL

language\_1 <- NULL

age\_1 <- NULL

while(mongo.cursor.next(cursor)) {

current\_row\_number<- current\_row\_number+1

current\_row<- mongo.cursor.value(cursor)

name\_1<- rbind(name\_1, mongo.bson.value(current\_row,'name'))

language\_1<- rbind(language\_1, mongo.bson.value(current\_row,'language'))

age\_1<- rbind(age\_1, mongo.bson.value(current\_row, 'age'))  
}  
Persons\_dataset<- data.frame(name = name\_1,language=language\_1,age = age\_1)  
  
View(Persons\_dataset)  
  
12)library(rjson)

heart<-read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/heart/heart.dat",  
sep="",header=FALSE)

namespace2 <- paste("r","heart",sep = ".")

heartlist<- lapply(split(heart, 1:nrow(heart)), function(x) mongo.bson.from.JSON(toJSON(x)))

ok6<-mongo.insert.batch(mongo,namespace2,heartlist)

13) mongo.destroy(mongo)

Part 3  
  
i) var map = function() {

var courses = this.courses;

var course;

varcourseWords;

varstopWords = new Array('time','person','year','way','day','thing','man','world','life',

'hand','part','child','eye','woman','place','work','week','case','point','government','company',

'number','group','problem','fact’,’be','have','do','say','get','make','go','know','take','see',

'come','think','look','want','give','use','find','tell','ask','work','seem','feel','try','leave','call','good',

'new','first','last','long','great','little','own','other','old','right','big','high','different','small','large',

'next','early','young','important','few','public','bad','same','able','to','of','in','for','on','with','at','by',

'from','up','about','into','over','after','beneath','under','above','the','and','a','that','I','it','not','he','as',  
  
'you','this','but','his','they','her','she','or','an','will','my','one','all','would','there','their')  
 if (courses) {

for (varidx = 0; idx<=courses.length-1; idx++){  
course= courses[idx].course\_title

courseWords = course.toLowerCase().split(" ");

for (vari = 0; i<=courseWords.length-1; i++){

if ( stopWords.indexOf(courseWords[i]) == -1 ) {

emit(courseWords[i], 1);

}}}}};  
  
var reduce = function( key, values ) {var count = 0;  
values.forEach(function(v) {  
count +=v;});  
return count;}  
db.students.mapReduce(map, reduce, {out: "word\_count\_courses"})

ii) var map = function() {

var courses = this.courses;

var course;

var key;

if (courses) {

for (var j = 0; j <=courses.length-1; j++){course= courses[j]

if (course.course\_status==="Complete") {key = {homeCity:this.home\_city, courseType:course.course\_title.charAt(0)};

emit(key, course.grade);}}}};

var reduce = function( key, values ) {  
 return (Array.sum(values)/values.length);  
}

db.students.mapReduce(map, reduce, {out: "GPA"})