MongoDB – CRUD Operations

Advanced Databases

Submitted To:
Mr. Busca

Submitted By:
Romain Brisse

05/11/2018

LaTeX

Contents

0.1 Design

Within MongoDB, the different components of the E/R diagram will be changed like so:

- The collections are Employee & Mission. Dept will be integrated to employee
- The documents are the rows of each table.
- The columns become the fields, and since the IDs are already defined within the company.sql database, and have a signification, I will use them instead of using MongoDB's autogeneration function.
- These imported ID will still be the primary keys.

Here is also an example of the insertion of a document in each collection I defined.

```
db.EMP.insertOne({
    EID:1,
    ENAME: "John Smith",
    JOB: "Manager",
    MGR: 1,
    HIRED: "2006/08/12",
    SAL: 7000,
    COMM: 500,
    DNAME: "Accounting",
    DLOC:"New York"
})
```

```
db.MISSION.insertOne({
    MID: 2029,
    EID: 7654,
    CNAME: "BMW",
    MLOC: "BERLIN",
    ENDD: "2011/02/09"
})
```

0.2 Population

```
> db.EMP.insertOne({_id:7839, ENAME:"KING",    JOB:"PRESIDENT",    MGR: null, HIRED:"1981-11-17", SAL
:5000.00,   COMM: null,   DNAME: null, DLOC: null});
{ "acknowledged" : true, "insertedId" : 7839 }
```

Figure 1: inserting an employee

```
1          db.products.insertMany( [
2              { item: "card", qty: 15 },
3              { item: "envelope", qty: 20},
4              { item: "stamps" , qty: 30 }
5     ] );
```

After that, I used the given file on campus to create the database according to your requirements.

0.3 Queries

0.3.1 all the employees

db.employees.find().pretty();

0.3.2 the number of employees

db.employees.count().pretty();

0.3.3 one of the employees, with pretty printing (2 methods)

db.employees.findOne();

db.employees.limit(1).find();

0.3.4 the name and salary of all the employees (without $_id$)

db.employees.find(,_id:0).pretty();

0.3.5 the name and salary of the employees with a salary in the range [1,000 , 2,500[

db.employees.find(salary:$gte:1000,$lte:2500,$_id : 0).pretty();$

0.3.6 the name and salary of the clerks with a salary in the range [1,000 , 1,500[(2 methods)

db.employees.find(salary:$gte:1000,$lte:1500,job:"clerk",$_id : 0).pretty();$

0.3.7 the name, job and salary of the employees, sorted first by job (ascending) and then by salary (descending)

db.employees.find().sort(job:1,salary:-1).pretty()

0.3.8 the employee names that begins with 'S' and ends with 't'

db.employees.find(name:/$^S.*t\$/).pretty();$

0.3.9 the employees whose name contains a double 'l'

db.employees.find(name:/$\cdot * ll. * \$/).pretty();$

0.3.10 the employees who don't have a manager

db.employees.find(manager:null).pretty();

0.3.11 the employees who work in Dallas (2 methods)

db.employees.find("department.location":"Dallas").pretty()

db.employees.find("department.location":$eq:"Dallas").pretty()

0.3.12 the employees who don't work in Chicago (2 methods)

```
db.employees.find("department.location":$ne:"Chicago").pretty()
```

```
db.employees.find("department.location":$not:$eq:"Chicago")
```

0.3.13 the employees who did a mission in Chicago

```
db.employees.find("missions.location":"Chicago").pretty()
```

0.3.14 the employees who did a mission in Chicago or Dallas

```
db.employees.find($or:["missions.location":"Chicago","missions.location":"Dallas"]).pretty()
```

0.3.15 the employees who did a mission in Chicago and Dallas

```
db.employees.find($and:["missions.location":"Chicago","missions.location":"Dallas"]).pretty()
```

0.3.16 the employees who did a mission for IBM in Chicago

```
db.employees.find($and:["missions.name":"IBM","missions.location":"Chicago"]).pretty()
```

0.3.17 the employees who did their first mission for IBM

```
db.employees.find("missions.company":"IBM","missions.location":"Chicago").pretty();
```

0.3.18 the employees who did exactly two missions

```
db.employees.find("missions":$size:2)
```

0.3.19 the name of the departments

```
db.employees.distinct("department.name")
```

0.3.20 the cities where the missions took place

```
db.employees.distinct("missions.location")
```

0.4 Updates

- 1. db.employees.updateMany(job:"clerk",$inc: salary:300)
- 2. db.employees.updateMany(manager:null)
- 3. db.employees.updateMany("department.name":"Accouting",$set: "department.location":"Dallas") "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 3
- 4. db.employees.updateMany("department.location":"New-York",$unset:missions:"");