CSCI235 Database Systems

MongoDB Query Language

Dr Janusz R. Getta

School of Computing and Information Technology -
University of Wollongong

MongoDB: Query language

Outline

MongoDB query language

MongoDB query language is based on a concept of pattern matching

A query is expressed as a BSON pattern and all documents that match the pattern are included in an answer

A method `find()` is used to match a pattern with the documents in a collection

```
db.department.find({"age":25})
```
find()

Matching of an empty pattern `{}` with a collection returns the returns an entire collection

```
db.department.find({})
```
find()

Finding the first n documents in a collection

```
db.department.find({}).limit(1)
```
find()

MongoDB query language

A sample document

Sample document

```
db.department.insert(
  { "name":"School of Computing and Information Technology",
    "code":"SCIT",
    "total_staff_number":30,
    "budget":1000000,
    "address":{"street":"Northfields Ave",
               "bldg":3,
               "city":"Wollongong",
               "country":"Australia"},
   "courses":[ {"code":"CSCI835",
                "title":"Database Systems",
                "credits":6},
               {"code":"CSIT115",
                "title":"Data Management and Security",
                "credits":6},
               {"code":"CSCI317",
                "title":"Database Performance Tuning",
                "credits":6},
               {"code":"CSIT321",
                "title":"Software Project",
                "credits":12}
             ]
    }
);
```

MongoDB: Query language

Outline

Simple queries

Find total number of documents in a collection

```
db.department.count()
```
count()

Find all departments whose code is `SCIT`

```
db.department.find({"code":"SCIT"})
```
find()

Find total number of departments whose code is `SOPH`

```
db.department.find({"code":"SOPH"}).count()
```
find()

```
db.department.count({"code":"SOPH"})
```
count()

Find all departments whose name is `School of Physics` and whose code is `SOPH`

```
db.department.find({"name":"School of Physics", "code":"SOPH"})
```
find()

MongoDB: Query language

Outline

Queries with Boolean operations

Find all departments whose name is `School of Physics` or whose code is `SCIT`

find()

```
db.department.find({$or:[{"name":"School of Physics"},{"code":"SCIT"}]})
```

Find all departments whose name is `School of Physics` and whose code is `SOPH`

find()

```
db.department.find({$and:[{"name":"School of Physics"},{"code":"SOPH"}]})
```

Find all departments whose code is either `SCIT` or `SOPH`

find()

```
db.department.find({"code":{$in:["SCIT", "SOPH"]}})
```

Find all departments where `budget > 10000`

find()

```
db.department.find({"budget":{$gt:10000}})
```

MongoDB: Query language

Outline

Boolean expressions

Comparison "key"="value"

```
{"key":"value"}
```
Boolean expression

```
{"key": {$eq:"value"}}
```
Boolean expression

Comparison "key" > "value"

```
{"key": {$gt:"value"}}
```
Boolean expression

Disjunction ("key1"="value1") or ("key2"="value2")

```
{$or: [{"key1":"value1"},{"key2":"value2"}]}
```
Boolean expression

Conjunction ("key1"="value1") and ("key2"="value2")

```
{$and: [{"key1":"value1"},{"key2":"value2"}]}
```
Boolean expression

Boolean expressions

Boolean expression `(("key1"="value1") or ("key2"="value2")) and ("key3"="value3")`

Boolean expression

```
{$and: [{$or: [{"key1":"value1"},{"key2":"value2"}]},{"key3":"value3"}]}
```

Negation of a comparison `"key" not ="value"`

Boolean expression

```
{"key": {$not: {$eq:"value"}}}
```

Negation of an expression `not (("key1"="value1") or ("key2"="value2"))`

Boolean expression

```
{$nor: [{"key1":"value1"},{"key2":"value2"}]}
```

Negation `not ("key1"="value1")`

Boolean expression

```
{$nor: [{"key1":"value1"}]}
```

MongoDB: Query language

Outline

Queries on nested documents

Find all departments located in Wollongong

find()

```
db.department.find({"address.city":"Wollongong"})
```

Find all departments that offer a course that has a code `SOA101`, title `Astronomy for Kids` and it is worth `3` credits

find()

```
db.department.find({"courses":{"code":"SOA101",
                               "title":"Astronomy for Kids",
                               "credits":3}})
```

Find all departments that offer the courses `SOA101`, `SOA201`, `SOA301`, with titles `Astronomy for Kids`, `Black Holes`, `Dark Matter`, and credits `3`, `6`, and `12` respectively

find()

```
db.department.find({"courses":[{"code":"SOA101","title":"Astronomy for Kids", "credits":3},
                               {"code":"SOA201", "title":"Black Holes", "credits":6},
                               {"code":"SOA301", "title":"Dark Matter", "credits":12}]})
```

Queries on nested documents

Find all departments such that the second offered course has code `SOA201`, title `Black Holes` and it is worth `6` credits

find()

```
db.department.find({"courses.1":{"code":"SOA201","title":"Black Holes","credits":6}})
```

Find all departments that offer courses worth more than 12 and less than 18 credits

find()

```
db.department.find({"courses.credits": {$gt:12, $lt:18}})
```

Find all departments such that the first offered course is worth 6 credits

find()

```
db.department.find({"courses.0.credits":6})
```

Queries on nested documents

Find all departments such that the first offered course is worth less than 6 credits

find()

```
db.department.find({"courses.0.credits":{$lt:6}})
```

Find all departments such that any offered course is worth less than 6 credits

find()

```
db.department.find({"courses.credits":{$lt:6}})
```

Find all departments such that offer Quantum Mechanics course worth 6 credits

find()

```
db.department.find({"courses.credits":6,"courses.title":"Quantum Mechanics"})
```

MongoDB: Query language

Outline

Query operators on array

Array equal to `[1,2,3,4,5]`

```
{"array":{$all:[1,2,3,4,5]}}
```
find()

Array includes an element that satisfies a condition

```
{"array":{"$elemMatch":{$eq:2}}}
```
find()

```
{"array":{"$elemMatch":{$gt:2,$lt:4}}}
```
find()

Size of an array

```
{"array":{$size:5}}
```
find()

MongoDB: Query language

Outline

Projections

Find name and code of each department

```
db.department.find({ },{"name":1,"code":1})
```
find()

find() name and code of each department and do not include document identifier `_id` into an answer

```
db.department.find({ },{"name":1,"code":1, "_id":0})
```
find()

Find all key:value pairs describing each department except `name`, `code` and `_id`

```
db.department.find({ },{"name":0,"code":0,"_id":0})
```
find()

Find all key:value pairs describing each department except course `credits`, course `code`, and `_id`

```
db.department.find({ },{"courses.credits":0,"courses.code":0,"_id":0})
```
find()

MongoDB: Query language

Outline

Queries about nulls and missing keys

Find all departments that have no `budget`

```
db.department.find({"budget":null})
```
find()

Find all departments that have a `budget`

```
db.department.find({"budget":{$not: {$eq:null}}})
```
find()

Find all departments that have no key `name` in their description

```
db.department.find({"name":{$exists:false}})
```
find()

Find all departments that have key "name" in their description

```
db.department.find({"name":{$exists: true }})
```
find()

Queries about nulls and missing keys

Find all departments that have no `code` of a `course`

```
db.department.find({"courses.code":{$exists:false}})
```
find()

```
db.department.find({"courses.code":{$not:{$exists:false}}})
```
find()

MongoDB: Query language

Outline

Iterations over a cursor

Display the first 20 departments

```
db.department.find({})
```
find()

Display the next 20 departments

```
it
```
cursor

Create a cursor and display all documents in a cursor

```
var cursor = db.department.find({})
cursor
```
cursor

```
var cursor = db.department.find({})
while(cursor.hasNext()) { print(tojson(cursor.next())); }
```
cursor

```
var cursor = db.department.find({})
cursor.forEach(printjson)
```
cursor

Iterations over a cursor

Use a cursor to list names and budgets of all departments whose budget is greater than 1000

cursor

```
var cdept=db.department.find()
cdept.forEach(function(x)
  { if (x.budget > 1000 ) {print(x.name, x.budget)}});
```

Saving the results in an array

cursor

```
var cursor = db.department.find({})
var CursorArray = cursor.toArray();
var document = CursorArray[2];
printjson(document)
```

References

Chodorow K. MongoDB The Definitive Guide, O'Reilly, 2013, chapter 2

Banker K., Bakkum P., Verch S., Garret D., Hawkins T., MongoDB in Action, 2nd ed., Manning Publishers, 2016

MongoDB Reference, Operators, Query and Projection Operators