CSCI235 Database Systems

MongoDB Databases, Collections, Documents

Dr Janusz R. Getta

School of Computing and Information Technology -
University of Wollongong

MongoDB Databases, Collections, Documents

Outline

Basics

MongoDB is a database system that belong to a class of so called NoSQL database systems based on a data model different from the relational model and data definition, manipulation, retrievel, and administration languages different from SQL

MongoDB data model (BSON) is based a on concept of key:value pairs grouped into documents and arrays

MongoDB database system operates on a number of databases

A MongoDB database is a set of collections

A MongoDB collection is a set of documents

A MongoDB document is a set of key:value pairs

A MongoDB value is either an atomic value or a document or an array

A MongoDB atomic value is of one of the types included BSON specification like number, string, date, etc

A MongoDB array is a sequence of values

Basics

Each MongoDB key:value pair must have a unique key within in a document

Each MongoDB document must have a unique identifier within a collection

Each MongoDB collection must have a unique name within a database

A sample BSON document

```
{"_id": ObjectId(),
 "full name": {"first name":"James",
               "initials":null,
               "last name":"Bond"},
 "employee number":"007",
 "skills":["cooking", "painting", "gardening"],
 "cars owned":[ {"rego":"007-1",
                 "made":"Porsche"},
                {"rego":"007-2",
                 "made":"Ferrari"} ],
 "secret codes":[ [1,2,3,4],
                  [9,8,7,5] ],
 "date of birth":new Date("1960-01-01")
}
```

MongoDB Databases, Collections, Documents

Outline

Architecture

MongoDB flexible storage architecture automatically manages the movement of data between storage engine technologies using native replication

MongoDB stores data as documents in a binary representation called BSON (Binary JSON)

MongoDB query model is implemented as methods or functions within the API of a specific programming language, as opposed to a completely separate language like SQL

MongoDB provides horizontal scale-out for databases on low cost, commodity hardware or cloud infrastructure using a technique called sharding, which is transparent to applications

In-Memory storage engine enables performance advantages of in-memory computing for operational and real-time analytics workloads

MongoDB Enterprise Advanced provides extensive capabilities to defend, detect, and control access to data (data security)

Architecture

MongoDB Ops Manager makes easy for operations teams to deploy, monitor, backup and scale the system (system management)

MongoDB Atlas provides all of the features of Database as a Service cloud computing model

MongoDB Databases, Collections, Documents

Outline

Server

Starting MongoDB server with options `--dbpath`, `--port`, and `--bind_ip`

Starting MongoDB server
```
mongod --dbpath data --port 4000 --bind_ip 10.0.2.100
```

Starting MongoDB server with options `--dbpath`, `--port`, and server running on a `localhost`

Starting MongoDB server
```
mongod --dbpath data --port 4000 --bind_ip localhost
```

or simply ...

Starting MongoDB server
```
mongod --dbpath data --port 4000
```

Starting MongoDB command based shell

Starting MongoDB command client
```
mongo --port 4000
```

Server

Getting the first help from MongoDB shell

Getting MongoDB help

```
help
```

MongoDB help messages

```
db.help()        help on db methods
show dbs         show database names
show collections show collections in current database
use db_name      set current database
... ... ...      ... ... ...
```

MongoDB Databases, Collections, Documents

Outline

Databases

Setting a default database

```
Setting 'database-name' as a default database
use database-name
```

For example using a database `local`

```
Setting 'local' as a default database
use local
```

Creating and switching to a new database `mydb`

```
Setting 'mydb' as a default databas
use mydb
```

Listing the databases

```
Listing all databases
show dbs
```

```
Databases
local 0.000GB
mydb 0.000GB
```

MongoDB Databases, Collections, Documents

Outline

Collections

Creating a new collection with an empty document

Creating a new collection mycol

```
db.mycol.insert({})
```

Listing the contents of a collection

Listing a collection mycol

```
db.mycol.find()
```

```
{ "_id" : ObjectId("57e385f8ffc660a351b58010") }
```

Listing the collections

Listing the names of collections

```
show collections
```

```
mycol
```

MongoDB Databases, Collections, Documents

Outline

Documents

Creating a new non empty document

```
Inserting a document into a collection mycol
db.mycol.insert({"one":"1", "many ones":[1,1,1,1]})
```

Listing the contents of a collection

```
MongoDB Shell
db.mycol.find()
```

```
{ "_id" : ObjectId("57e385f8ffc660a351b58010") }
{ "_id" : ObjectId("57e38cbeffc660a351b58012"),
  "one" : "1", "many ones" : [ 1, 1, 1, 1 ] }
```

MongoDB Databases, Collections, Documents

Outline

Formatting

Listing the nicely formatted contents of a collection

Listing a nicely formatted collection

```
db.mycol.find().pretty()
```

```
{ "_id" : ObjectId("57e385f8ffc660a351b58010") }
{ "_id" : ObjectId("57e38cbeffc660a351b58012"),
  "one" : "1",
  "many ones" : [ 1,
                  1,
                  1,
                  1
                ]
}
```

MongoDB Databases, Collections, Documents

Outline

DDL

Removing all documents from a collection

```
Removing all documents from a collection
db.mycol.remove({})
```

Removing a collection

```
Dropping a collection
db.mycol.drop()
```

Removing a database

```
Dropping a database
db.dropDatabase()
```

MongoDB Databases, Collections, Documents

Outline

DML

Let a file `dbload.js` contains the following `insert` methods

Inserting documents into a collection mycol

```
db.mycol.insert({"CITY": {"name":"Wollongong",
                          "population":"80K",
                          "country":"Australia",
                          "state":"New South Wales"} });
db.mycol.insert({"EMPLOYEE":{"enum":1234567,
                             "full-name":"Janusz R.Getta",
                             "salary": "200K",
                             "hobbies":["cooking",
                                        "painting",
                                        "gardening"]} });
```

Processing a script inserts two documents into a collection `mycol`

Processing a script dbload.js

```
load("dbload.js")
```

Listing a collection `mycol`

Listing a nicely formatted collection mycol

```
db.mycol.find().pretty()
```

DML

```
{
      "_id" : ObjectId("57e3c817fe6a1bfd5105022a"),
      "CITY" : {
                 "name" : "Wollongong",
                 "population" : "80K",
                 "country" : "Australia",
                 "state" : "New South Wales"
                }
}
{
      "_id" : ObjectId("57e3c817fe6a1bfd5105022b"),
      "EMPLOYEE" : {
                     "enum" : 1234567,
                     "full-name" : "Janusz R.Getta",
                     "salary" : "200K",
                     "hobbies" : [
                                    "cooking",
                                    "painting",
                                    "gardening"
                                 ]
                    }
}
```

MongoDB Databases, Collections, Documents

Outline

Query language

MongoDB query language is based on a concept of pattern matching

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

A method `find()` is used to match a query pattern {"age":25}against the documents in a collection

```
Finding all documents that have a pair "age":25 at the topmost nesting level
db.department.find({"age":25})
```

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

```
Finding all documents in a collection
db.department.find({})
```

Query language

The first 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}
            ]
    }
);
```

Query language

The second document

```
db.department.insert(
  { "name":"School of Astronomy",
    "code":"SOA",
    "total_staff_number":5,
    "budget":10000,
    "address":{"street":"Franz Josef Str",
               "bldg":4,
               "city":"Vienna",
               "country":"Austria"},
   "courses":[ {"code":"SOA101",
                "title":"Astronomy for Kids",
                "credits":3},
               {"code":"SOA201",
                "title":"Black Holes",
                "credits":6},
               {"code":"SOA301",
                "title":"Dark Matter",
                "credits":12}
             ]
  }
);
```

Query language

The third document

```
db.department.insert(
  { "name":"School of Physics",
    "code":"SOPH",
    "total_staff_number":25,
    "budget":100000,
    "address":{"street":"Victoria St",
               "bldg":25,
               "city":"Cambridge",
               "country":"UK"},
   "courses":[ {"code":"SOPH101",
                "title":"Special Relativity",
                "credits":6},
               {"code":"SOPH102",
                "title":"General Relativity",
                "credits":12},
               {"code":"SOPH103",
                "title":"Quantum Mechanics",
                "credits":18}
             ]
  }
);
```

Query language

Find total number of documents in a collection

```
                                                  Counting the documents
db.department.count()
```

Find all departments whose code is `SCIT`

```
                                   Finding the documents that match a given search pattern
db.department.find({"code":"SCIT"})
```

Find total number of departments whose code is `SOPH`

```
                                  Counting the documents that match a given search pattern
db.department.find({"code":"SOPH"}).count()
```

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

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

```
                                   Finding the documents that match a given search pattern
db.department.find({"name":"School of Physics", "code":"SOPH"})
```

Query language

Comparison `"key"="value"`

```
Search pattern
{"key":"value"}
```

```
Search pattern
{"key": {$eq:"value"}}
```

Comparison `"key" > "value"`

```
Search pattern
{"key": {$gt:"value"}}
```

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

```
Search pattern
{$or: [{"key1":"value1"},{"key2":"value2"}]}
```

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

```
Search pattern
{$and: [{"key1":"value1"},{"key2":"value2"}]}
```

Query Language

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

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

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

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

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

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

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

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

Query language

A sample nested 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}
            ]
    }
);
```

Query Language

Find all departments located in Wollongong

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

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

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

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

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

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

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

References

MongoDB Architecture, `https://www.mongodb.com/mongodb-architecture`

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