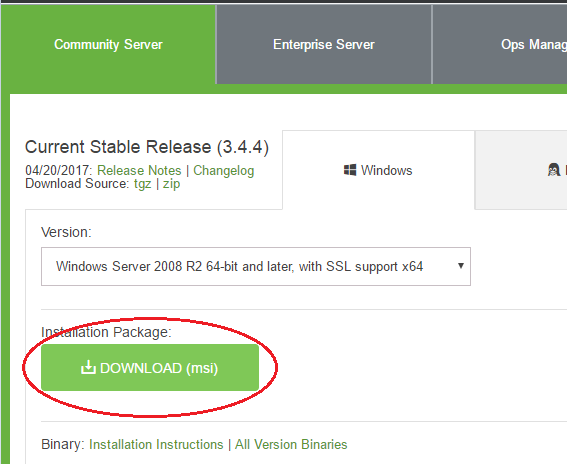
# Data Centric RAD

## Lab 5 MongoDB

### Part 1 Install MongoDB

Part 1 only needs to be done if you are using your own laptop.

1. Go to [https://www.mongodb.com/download-center#community](https://www.mongodb.com/download-center" \t "_blank) and download Current Stable Release (3.4.4)



1. Click on the file just downloaded which opens the installer.
2. Click *Next*.
3. Accept Terms & Conditions and click *Next*.
4. Choose *Complete*Setup.
5. Click *Install*.
6. Click *Finish*.
7. Create the following folder: C:\data\db\

### Setup

1. Open the Windows command prompt and navigate to the following folder:

C:\Program Files\MongoDB\Server\3.4\bin

1. Start the mongo daemon as follows:

mongod

1. In a different Windows command prompt window navigate to the following folder:

C:\Program Files\MongoDB\Server\3.4\bin

1. Start mongoDB as follows:

mongo

### Exercises

1. Create a mongodb database called usersdb.

use userdb

switched to db userdb

1. Create a collection called users which has documents with the following attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **\_id** | **fname** | **surname** | **age** | **email** | **carReg** |
| 100 | John | Smith | 33 | [jsmith@gmail.com](mailto:jsmith@gmail.com) | 131-G-101 |
| 101 | Sean | Murphy | 21 | [seanmurph@yahoo.com](mailto:seanmurph@yahoo.com) | 04-WH-235 |
| 102 | Aine | Browne | 23 | [abrowne@gmail.com](mailto:abrowne@gmail.com) |  |
| 103 | Alan | Murphy | 24 | [murpha@hotmail.com](mailto:murpha@hotmail.com) | 07-RN-9988 |
| 104 | Sarah | Doyle | 23 | [sarah@gmail.com](mailto:sarah@gmail.com) | 142-G-2343 |
| 105 | Bill | Mulligan | 19 | [billy123@gmail.com](mailto:billy123@gmail.com) |  |
| 106 | Shane | Kelly | 24 | [sk998@yahoo.com](mailto:sk998@yahoo.com) |  |
| 107 | Will | Doyle | 19 | [doyler123@gmail.com](mailto:doyler123@gmail.com) | 10-G-2353 |

db.user2.save({\_id:101,fname:"Sean",surname:"Murphy",age:21,email:"seanmurph@yahoo.com",carReg:"04-WH235"})

db.user2.save({\_id:102,fname:"Anie",surname:"Brown",age:23,email:"abrowne@gmail"})

db.user2.save({\_id:103,fname:"Alan",surname:"Murphy",age:24,carReg:"07-RN-9988"})

db.user2.save({\_id:104,fname:"Sarah",surname:"Doyle",age:23,email:"sarah@gmail.com",carReg:"142-G-2343"})

db.user2.save({\_id:105,fname:"Bill",surname:"Mulligan",age:19,email:"billy123@gmail.com"})

db.user2.save({\_id:106,fname:"Shane",surname:"Kelly",age:24,email:"sk998@yahoo.com"})

db.user2.save({\_id:107,fname:"Will",surname:"Doyle",age:19,email:"doyler123@yahoo.com",carReg:"10-G-2353"})

1. Give the command to find all documents in the users collection.

db.user2.find()

{ "\_id" : 100, "fname" : "John", "surname" : "Smith", "age" : 33, "email" : "jsmith@gmail.com", "carReg" : "131-G-101" }

{ "\_id" : 101, "fname" : "Sean", "surname" : "Murphy", "age" : "21", "email" : "seanmurph@yahoo.com", "carReg" : "04-WH235" }

{ "\_id" : 102, "fname" : "Anie", "surname" : "Brown", "age" : 23, "email" : "abrowne@gmail" }

{ "\_id" : 103, "fname" : "Alan", "surname" : "Murphy", "age" : 24, "carReg" : "07-RN-9988" }

{ "\_id" : 104, "fname" : "Sarah", "surname" : "Doyle", "age" : "23", "email" : "sarah@gmail.com", "carReg" : "142-G-2343" }

{ "\_id" : 105, "fname" : "Bill", "surname" : "Mulligan", "age" : "19", "email" : "billy123@gmail.com" }

{ "\_id" : 106, "fname" : "Shane", "surname" : "Kelly", "age" : "24", "email" : "sk998@yahoo.com" }

{ "\_id" : 107, "fname" : "Will", "surname" : "Doyle", "age" : 19, "email" : "doyler123@yahoo.com", "carReg" : "10-G-2353" }

1. Give the command to find all documents in the users collection where the age is 19.

db.user2.find({age:19})

{ "\_id" : 107, "fname" : "Will", "surname" : "Doyle", "age" : 19, "email" : "doyler123@yahoo.com", "carReg" : "10-G-2353" }

1. Give the command to find all documents in the users collection where the age greater than 19.

db.user2.find({age:{$gt:19}})

{ "\_id" : 100, "fname" : "John", "surname" : "Smith", "age" : 33, "email" : "jsmith@gmail.com", "carReg" : "131-G-101" }

{ "\_id" : 102, "fname" : "Anie", "surname" : "Brown", "age" : 23, "email" : "abrowne@gmail" }

{ "\_id" : 103, "fname" : "Alan", "surname" : "Murphy", "age" : 24, "carReg" : "07-RN-9988" }

1. Give the command to find all documents in the users collection where the age greater than 19 and the user has a car.

db.user2.find({$and:[{carReg:{$exists:true}},{age:{$gt:19}}]})

{ "\_id" : 100, "fname" : "John", "surname" : "Smith", "age" : 33, "email" : "jsmith@gmail.com", "carReg" : "131-G-101" }

{ "\_id" : 103, "fname" : "Alan", "surname" : "Murphy", "age" : 24, "carReg" : "07-RN-9988" }

Db

1. Give the command to find all documents in the users collection where \_id is greater than 104 and age is greater than 20.

db.user2.find({$and:[{\_id:{$gt:104}},{age:{$gt:20}}]})

{ "\_id" : 106, "fname" : "Shane", "surname" : "Kelly", "age" : 24, "email" : "sk998@yahoo.com" }

1. Give the command to find the first document in the users collection where the user has a car.

First 相当于only用findOne

db.user2.findOne({carReg:{$exists:true}})

{

"\_id" : 100,

"fname" : "John",

"surname" : "Smith",

"age" : 33,

"email" : "jsmith@gmail.com",

"carReg" : "131-G-101"

}

1. User 106 - Shane has bought a car with reg 12-G-1234. The following command was run to update the user’s document:

db.users.save({\_id:106, carReg:"12-G-1234"})

What does the document look like now and why?

There are only \_id and carReg.

The second document was saved over the first one, because the \_id specified in the second document already existed in the collection.

{ "\_id" : 106, "carReg" : "12-G-1234" }

Because save() takes as a parameter, the document to update. If the document \_id exists, the existing document is replaced. If the document \_id does not exist, a new document is created.

1. User 102 - Sean has bought a car with reg 10-G-9876. The following command was run to update the user’s document:

db.users.update({\_id:102}, {carReg:"10-G-9876"})

What does the document look like now and why?

Because update() takes as its second parameter, the document to update.

The document to update replaces the existing document if the query is found, if the query is not found, nothing is updated.

{ "\_id" : 102, "carReg" : "10-G-9876" }

* The entire document (except the \_id attribute) is replaced with the *update* parameter of the update method.
* In the case where you want to update more than one document, make sure the [multi](https://docs.mongodb.com/manual/reference/method/db.collection.update/) parameter is set.

1. User 105 – Bill’s document is as follows:

{ "\_id" : 105, "fname" : "Bill", "surname" : "Mulligan", "age" : 19, "email" : "billy123@gmail.com" }

Bill has bought a car with reg 161-MO-4. Give the command so that Bill’s document now looks as follows:

{ "\_id" : 105, "fname" : "Bill", "surname" : "Mulligan", "age" : 19, "email" : "billy123@gmail.com", "carReg" : "161-MO-4" }

$set用来指定一个键的值.如果这个键存在,就修改它;不存在,就创建它

$unset用来删除键。也可以修改普通文档和内嵌文档

$inc只能用于整数、长整数和双精度浮点数

其他类型应该使用$set修改器或其他修改器

db.user2.update({\_id:105},{$set:{carReg:"10-G-9876"}})

1. User 106’s document now looks as follows:

{ "\_id" : 106, "carReg" : "12-G-1234" }

Give the command to add the original fields:

db.user2.save({\_id:106,fname:"Shane",surname:"Kelly",age:24,email:"sk998@yahoo.com"})

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **\_id** | **fname** | **surname** | **Age** | **email** |
| 106 | Shane | Kelly | 24 | [sk998@yahoo.com](mailto:sk998@yahoo.com) |

back to the document in a single command so that the document now looks as follows:

db.user2.update({\_id:106},{$set:{carReg:"12-G-1234"}})

OR

db.users.update({\_id:106},{$set:{fname:"Shane",surname:"Kelly",age:24,email:"sk998@yahoo.com"}})

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **\_id** | **fname** | **surname** | **Age** | **email** | **carReg** |
| 106 | Shane | Kelly | 24 | [sk998@yahoo.com](mailto:sk998@yahoo.com) | 12-G-1234 |

1. Give the mongodb command to add 1 to each user’s age.

db.user2.update({},{$inc:{age:1}},{multi:true})

1. Add a new attribute *sex*, to each document as follows:

|  |  |
| --- | --- |
| **\_id** | **Sex** |
| 100 | M |
| 101 | M |
| 103 | M |
| 104 | F |
| 105 | M |
| 106 | M |
| 107 | M |

{$ne:...},{$set:{sex:”M”}}: 除了...这行之外的都加一个sex:M

db.user2.update({age:{$exists:true}},{$inc:{age:1}},{multi:true})

db.user2.update({age:{$exists:true}},{$inc:{age:1}},{multi:true})

OR

db.users.update({\_id:{$ne:104}},{$set:{sex:"M"}},{multi:true})

db.users.update({\_id:104},{$set:{sex:"F"}})

1. Add a new attribute *title* with the value *Mr.*, to each document where the *sex* is M, and the age is greater than 20:

db.user2.update({$and:[{sex:"M"},{age:{$gt:20}}]},{$set:{title:"Mr"}},{multi:true})

1. Users 101 – Sean, 103 – Alan and 107 – Will have sold their cars, update the collection to remove the *carReg* attribute from these documents.

db.user2.update({$or:[{\_id:{$eq:101}},{\_id:{$eq:103}},{\_id:{$eq:107}}]}, {$unset:{carReg:1}},{multi:true})

1. Give the mongodb command to list/show **only** the *fname*, *surname*, *age* and *sex* attributes of documents where the \_id is between 101 and 107 inclusive.

db.user2.find({$and:[{\_id:{$gt:100}},{\_id:{$lt:108}}]}, {fname:1,surname:1,age:1,sex:1,\_id:0})

注意：db.users.find({1-所需要找的条件},{2-最后输出的要求})

1和2的位置不能调换!