Advanced Database Programming Portfolio 2 - MongoDB

Niall Stack T00174406

2016

# Q1 - Implementing My Own MongoDB Database

## Introduction:

For the first question of my Advanced Database Programming project part two, I was asked to document the creation of a MongoDB database. I had used CouchDB in the first part so I had gotten used to using a NoSQL database although there was no simple web interface like futon for MongoDB, so everything had to be done via the command line(Although it is possible to download some interfaces they have to be installed and set up unlike futon). I used Mongo to create a car registration database that could be used to store all the cars in a country.

## Vision:

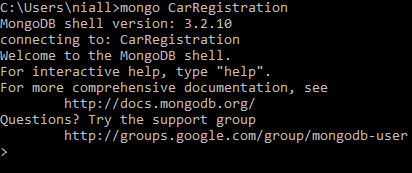
MongoDB is very good at handling big data so I wanted to pick a project that would suit this ability. I decided on choosing a car registration database that could be used for both a relatively small scale (such as for a town) and a large scale (like a national database), taking advantage of Mongo’s scalability. Using Mongo also gives the ability to have fields that are in some document and no others, such as if I needed different fields for an electric car because Mongo does not enforce a schema. MongoDB is also very efficient at ad hoc queries so it is fast at retrieving documents which is very important for a large scale document based application like a car registration database.

## Creating the Database:

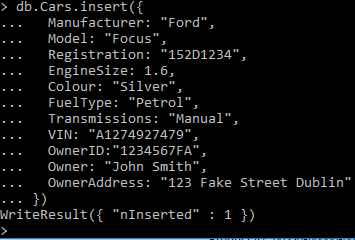
For the car registration database I initially decided to have 12 fields, although I knew it was likely I was going to add more but luckily that is very simple to do with the schema-less MongoDB:

* \_id
* Manufacturer
* Model
* Registration
* EngineSize
* Colour
* FuelType
* Transmissions
* VIN
* OwnerID
* Owner
* OwnerAddress

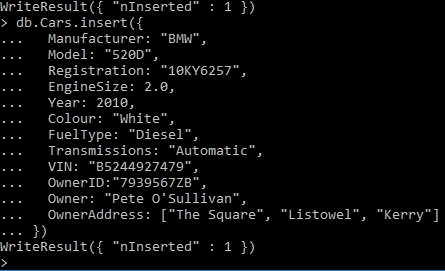
To create the Database, the same command is used as to connect to it which either connects if the database exists or creates a new database with the chosen name and connects to it.



Creating a collection also uses the same command as is used to add a document to a collection which like the above command to connect creates a collection if it does not already exist. The command creates the collection called “Cars” as there was no collection of Cars previously in the database and also inserts the document into it.



I inserted another document but instead of having a single string for address I created an array.

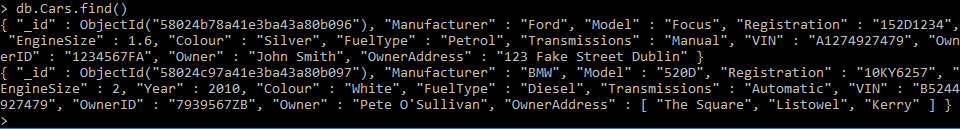


## Reading From the Database:

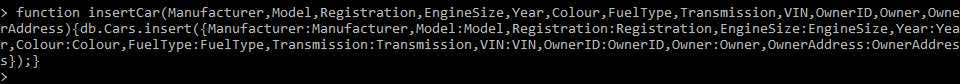
To view all collections in the database just use the simple command – “show collections”.



Then to view all the documents in a collection.



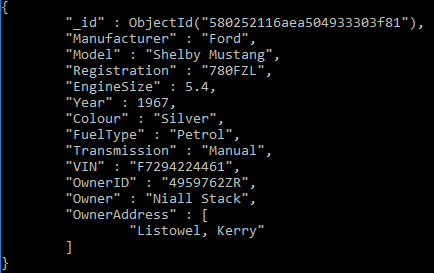
MongoDB also allows for functions to be created that can speed up the population process which is very useful for a big data application.



Using this function I can add a document to the collection with a more simplified insert.



Which could then be seen in the collection.



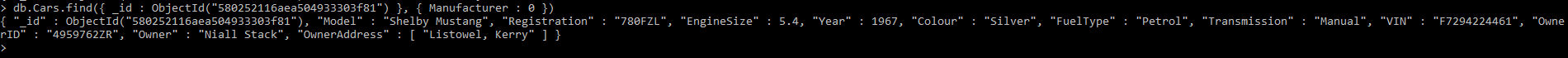
It is also possible make it easier to read add by adding to the find command.



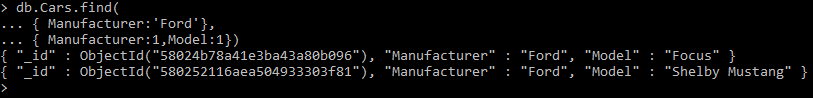
Mongo allows to view just one field along with the ID by setting the specific field to 1.



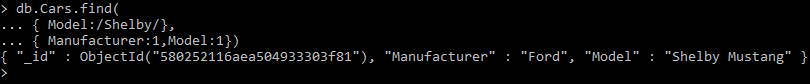
Then by turning changing it to zero you get all fields except that one.



You can also search for exact values in a field.



Also by partial values in a field using “/…/”.

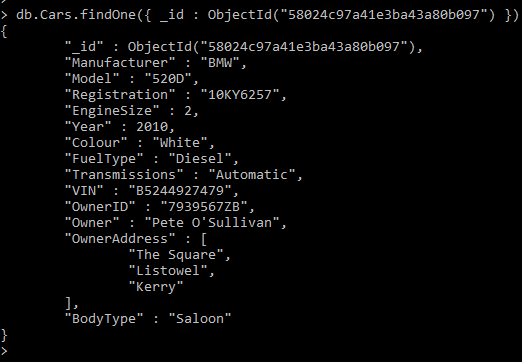


## Updating the Database

In Mongo it is simple to update an existing document for example by adding a field you just need the \_id and then use the $set command to alter the document.



We can check this with the findOne command



Not only can we use $set we can use $inc to increment a number, like the year the car was manufactured in.

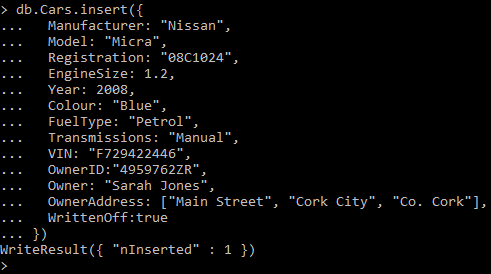


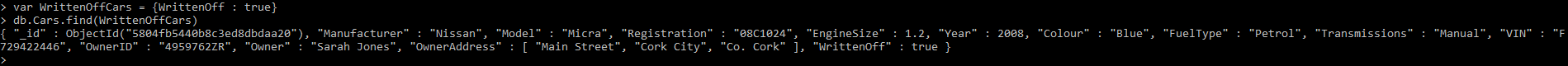
You can do many other operations with the update directives even removing with unset which I used after accidently creating a field called 2010 when I was meant to increment the Year.



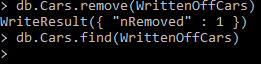
## Deleting from the Database

To delete a document from mongo there is a simple remove function but we can use an $elemMatch to search for certain queries and then remove them based on that. For this I decided to add a field about if a car was written off in a crash when I added a new vehicle.



I then created a “WrittenOffCars” variable that I saved all cars where the field “WrittenOff = true”.

Then to remove all cars that are written off I used the “WrittenOffCars” variable with the remove command.



## Conclusion:

Using Mongo’s CRUD commands I was able to create the document I had envisioned with relative ease and was able to do it quickly enough considering I was doing it all through the command line. Although I didn’t discuss is in the first question as I was not asked to, using map reduces was also very simple and would be of good help when creating a GUI to connect with the car registration database that I created here. MongoDB would definitely be an option I would consider when choosing a database for any future projects.

# Q2 - Implementing the Database into a GUI

## Introduction: