Lab 2: Data Manipulation

This lab is about the essentials of data manipulation with MongoDB - the typical and infamous CRUD stuff… You will learn how to create databases, collections & documents, how to query those documents, how to update and delete them.

We will be using the example of the simple blog system introduced in the lecture. While we already learned and talked about collection `user`, now we will start with the articles in this lab.

1. If not already done, install MongoDB by grabbing the appropriate version from the files provided (or from http://www.mongodb.org/downloads)

   - MongoDB 2.4.7 64-bit 2008R2+ for Windows 7 or greater
   - MongoDB 2.4.7 64-bit for all other Windows versions
   - MongoDB 2.4.7 64-bit for Linux or Mac OS X

   The installation instructions provided (or available at
   http://docs.mongodb.org/manual/installation/
   might be helpful

2. Copy all files from the provided folder `00_data_for_labs` into your MongoDB installation directory

3. Now open a shell and start MongoDB as a standalone server (using the default port 27017). To reduce the overhead on our local system we are using the parameter `--smallFiles` and `--oplogSize 50` which is not recommended for productive systems.

   ```
   > mongod --dbpath data --smallfiles --oplogSize 50
   ```

4. Now open another shell and connect to MongoDB (preloading the functions needed for this lab) :

   ```
   > mongo --shell lab_02.js
   ```

5. Now you are at the Mongo Shell and can start interacting with MongoDB. Let's start and have a look at the available data bases:

   ```
   > show dbs
   ```

6. There should be no data base and first thing is we switch to the data base `blog`:

```
> use blog
```

7. Now have again a look the databases:

```
> show dbs
```

   Uh, still no data base created? Remember data bases (and collections) are automatically created once you insert the first document into a collection.

8. Have a look at the collections of the data base `blog`:

```
> show collections
```

9. You should not see any collections yet. Thus we start with inserting the first blog article into our collection:

```
> db.articles.insert(
        {
              name : "Welcome",
              text : "Welcome to the MongoDB Blog!",
              author : "Sheldon"
        }
)
```

10. Now let's again have a look at the data bases and collections:

```
> show dbs
> show collections
```

   Ah, now they finally have been created!

11. In the next step we add some additional blog articles (This is done by a user defined function and the reason why you had too preload the file `lab_02.js`)

```
> initArticles()
```

12. Let's start with querying documents. Please answer the following questions:

    - How many documents are now in the collection `articles`?

      Answer: _____

    - How many articles did Sheldon write?

Answer: _____

- How many articles did Sheldon or Amy write?

  Answer: _____

- What is the name of the article Bernadette wrote?

  Answer: _____

- How many articles contain the word “MongoDB“?

  Answer: _____

- How many articles were written by the girls?

  Answer: _____

13. Let’s continue with the updating of documents. Please solve the following tasks:

(Notice: If you mess something totally up, please delete all documents with `db.articles.drop()` and restore the original data with the user defined function `restoreArticles()`):

- Oh no, there is a typo! Instead of the correct author `Raj` one article has the author `Ray`. Please correct that typo!

- Please add to the article called `Isolation` a new simple field `category` mit with the value `write operations`.

- Please also add to the article named `Isolation` a new field `tags` as an array with the two initial values als `mongodb` and `operations`. The updated document should look like this:

```
{
   …
   "name" : "Isolation",
   "tags" : [
         "mongodb",
         "operations"
    ],
    …
}
```

- Now please add to the article `Isolation` an additional tag `atomic`.
- Finally we want to add some more information about the author. Please enhance the field `author` of the document `Isolation` in such a way that the author now is a subdocument:

```
"author" :
        {
              first_name : "Bernadette",
              last_name : "Rostenkowski-Wolowitz"
        }
```

14. Ah, maybe all those enhancements were not such a good idea at all. Thus, please delete the article with the author Bernadette completely from the collection.

15. And while you are at it, please also remove all articles where the `text` starts with `"When …"`.