MongoDB Introduction

The Getting Started[1] guide is highly recommended!

1   Installing MongoDB

Download the "Community Edition" for your platform. Uncompress/install, as is appropriate. The Installation guide[2] has more detail.

2   Accessing MongoDB via the Shell

You will first run the MongoDB daemon (i.e. server), then connect via the shell. First, create a folder in which the server will store data. Then, run the daemon providing the path to this directory:

```
mongod --dbpath <path to data directory>
```

In a separate command line, now simply run the mongo shell:

```
mongo
```

3   Data Organization

A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays, and arrays of documents.

MongoDB stores documents in collections. Collections are analogous to tables in relational databases. Unlike a table, however, a collection does not require its documents to have the same schema. In MongoDB, documents stored in a collection must have a unique `_id` field that acts as a primary key. Finally, collections belong to a database.

To see all databases: `show dbs`

To see the current database: `db`

To see all collections in the current database: `show collections`

---

[1]`https://docs.mongodb.com/getting-started/shell/`
[2]`https://docs.mongodb.com/getting-started/shell/installation/`

4 CRUD Examples

Let's create and manipulate a database:

```
use mydb
db.cars.insert([{"make":"Toyota", "model":"Camry"},
                {"make":"Tesla", "model":"Model 3", "year":2017}])
```

The first command tells the shell to use "mydb" as the current database, which will work even if that database doesn't exist (after the first insert, the database will be created). The second command inserts two documents into the "cars" collection – notice that the fields don't perfectly match up, and that an "_id" field has been added automatically. To see the contents of the cars collection:

```
db.cars.find()
db.cars.count()
```

To find a particular car . . .

```
db.cars.find({"year":2017})
```

To get some field information . . .

```
db.cars.distinct("make")
```

To update a car . . .

```
db.cars.updateMany({}, { $set:{"colors":["black"]} })
db.cars.updateMany({"make":"Tesla", "model":"Model 3"}, { $push: {"colors":"red"} })
```

To remove a car . . .

```
db.cars.remove({"colors":"red"})
```

4.1 Importing Data

In a separate command line (fill in the appropriate path to the `tours.json` file) . . .

```
mongoimport --db mydb --collection tours --drop --file tours.json
```

Now back in the shell, try . . .

```
db.tours.find()
db.tours.find({"tourTags":"whale watching"})
db.tours.find({"tourTags":"whale watching"}).count()
db.tours.find({"tourTags":"whale watching", {"tourName":1, "tourPrice":1})
db.tours.find({"tourTags":"whale watching", "tourPrice":{$lt: 200}},
              {"tourName":1, "tourPrice":1})
```