MongoDB Tutorial: Installing and Importing Data

CompSci 516 Staff

1 Overview

This document will show you:

- How to install MongoDB Community Edition on Ubuntu 18.04 machine.
- How to import data into MongoDB
- How to run sample queries

2 Prerequisite

2.1 Virtual machine running Ubuntu 18.04

In order to avoid any incompatibility issue, we require you to reserve a virtual machine running Ubuntu 18.04 from Duke Virtual Computing Manager if you do not currently have one: `https://vcm.duke.edu` We will not provide any technical support if you choose to install it elsewhere.

2.2 Movielens dataset

After you successfully reserved your virtual machine, ssh into the machine and download the dataset from our course website:

```
$ cd ~
$ curl www2.cs.duke.edu/courses/fall19/compsci516/DataForClass/movielens.tsv
--output movielens.tsv
```

3 Installation

Import the public key used by the package management system:

```
$ sudo apt-get install gnupg
$ wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key add -
```

Create a list file for MongoDB:

```
$ echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.2
multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.2.list
```

Reload local package database:

```
$ sudo apt-get update
```

Install the MongoDB packages:

```
$ sudo apt-get install -y mongodb-org
```

Start/stop/restart MongoDB:

```
$ sudo service mongod start
$ sudo service mongod stop
$ sudo service mongod restart
```

Start a mongo shell:

```
$ mongo
```

4 Importing dataset

Exit mongo shell and import dataset into MongoDB:

```
$ cd ~
$ mongoimport -d movielens -c movies --type tsv --file movielens.tsv --headerline
--columnsHaveTypes
```

5 Sample Queries

Enter mongo shell and select movielens:

```
$ mongo
> use movielens
```

5.1 Sample query 1

Find the number of action movies that users in Duke love. Same movie can be repeated many times. The zip code of Duke is 27708. If the rating of a movie is greater or equal to 4, we say the user loves the movie. By action movies we mean any movie with the "Action" field set to true.

```
> db.movies.find({zipcode: 27708, rating: {$gte: 4}, "genre.Action": true}).count()
```

output: 10

5.2 Sample query 2

Find all user_ids over the age of 68 (age>68) who gave a rating that is 3.0 or above to a movie that was released in 1997. Return both the user_id and the movie_title. Sort your result by user_id(ascending) and movie_title(descending). Only display the first 20 rows.

```
> db.movies.aggregate([{$match: {age: {$gt: 68}, rating: {$gte: 3},
release_date: /.*1997.*/ }}, {$group: {_id: {user_id: "$user_id",
movie_title: "$movie_title"}}}, {$project: {_id:0, user_id: "$_id.user_id",
movie_title:"$_id.movie_title"}}, {$sort: {user_id:1, movie_title:-1}},
{$limit:20}])
```

output:

```
{ "user_id" : 481, "movie_title" : "Volcano (1997)" }
{ "user_id" : 481, "movie_title" : "Titanic (1997)" }
{ "user_id" : 481, "movie_title" : "Speed 2: Cruise Control (1997)" }
{ "user_id" : 481, "movie_title" : "Return of the Jedi (1983)" }
{ "user_id" : 481, "movie_title" : "Murder at 1600 (1997)" }
{ "user_id" : 481, "movie_title" : "Lost World: Jurassic Park, The (1997)" }
{ "user_id" : 481, "movie_title" : "Hamlet (1996)" }
{ "user_id" : 481, "movie_title" : "Fargo (1996)" }
{ "user_id" : 559, "movie_title" : "Wings of the Dove, The (1997)" }
{ "user_id" : 559, "movie_title" : "Murder at 1600 (1997)" }
{ "user_id" : 559, "movie_title" : "Men in Black (1997)" }
{ "user_id" : 559, "movie_title" : "McHale's Navy (1997)" }
{ "user_id" : 559, "movie_title" : "George of the Jungle (1997)" }
{ "user_id" : 559, "movie_title" : "Boot, Das (1981)" }
{ "user_id" : 559, "movie_title" : "Big Lebowski, The (1998)" }
{ "user_id" : 559, "movie_title" : "Air Force One (1997)" }
{ "user_id" : 559, "movie_title" : "Air Bud (1997)" }
{ "user_id" : 585, "movie_title" : "Titanic (1997)" }
{ "user_id" : 585, "movie_title" : "Prisoner of the Mountains (Kavkazsky Plennik) (1996)"
{ "user_id" : 767, "movie_title" : "Kolya (1996)" }
```

5.3 Sample query 3

Add a binary-valued column "satisfaction" in the movies table. When the rating of the movie is greater than or equals three, the value of the column should be 'Positive', otherwise, 'Negative'.

```
> db.movies.updateMany({rating: {$gte:3}}, {$set:{"satisfaction":"positive"}})
> db.movies.updateMany({rating: {$lt:3}}, {$set:{"satisfaction":"negative"}})
```

6 Additional Resources

- Complete MongoDB installation instructions:
  `https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/`
- MongoDB official document: `https://docs.mongodb.com/manual/`
- SQL to MongoDB mapping chart:
  `https://docs.mongodb.com/manual/reference/sql-comparison/`
- SQL to aggregation mapping chart:
  `https://docs.mongodb.com/manual/reference/sql-aggregation-comparison/`