

mongodb-yelp

Summary

Brief description of all the steps needed to configure the database. If you have done already one of these steps only follow the next in the sequence. - Install the MongoDB program

```
bash mongo_install.bash
```

- Configure MongoDB after installation

```
bash mongo_config.bash
```

- Populate database

```
# Using data already existent and creating extra data  
bash populate_mongodb.bash
```

As the proposed database have a size of 3G even compressed we propose a script that automatically download it from an url and after finish unzip the database and import it to the MongoDB database

To take a look more in deep about all the scripts you can see the next sections

Mongodb setup

Requirements

- Ubuntu 18.04

Steps to install mongodb

- Import the public key

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv  
9DA31620334BD75D9DCB49F368818C72E52529D4
```

- Add the repository to the sources

```
echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.0  
multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.0.list
```

- Reload local package database

```
sudo apt-get update
```

- Install mongodb

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

Configure mongodb after the installation

- Launch a linux console and create a local folder to save the databases

```
mkdir dbmongo
```

- Launch MongoDB server

```
mongod --shardsvr --dbpath dbmongo --port 27021 &
```

- **Open another linux console**
- Launch MongoDB instance

```
mongo --host localhost:27021
```

- If you want to stop the server then execute

```
mongod --shutdown --dbpath dbmongo
```

To load a database

```
mongoimport --db <databasename> --collection <collectionName> --drop --file  
<datafile> --port 27021  
# The option --drop erase and create the collection with the same name if exist yet  
if not only create one.
```

We import the five collections we have obtained: - business: Contain the data related to a particular business like the id, name, the category and other useful information. - Checkin: Contain the day of the week with the count of the people that did a checkin in a business. - Review: Contain a qualification of the business with a stars rating and commentary. - Tip: useful information posted by an user about one business. - User: Description of an user like name and mail.

To import that information we used the next commands:

```
mongoimport --db yelpdb --collection business --drop --file  
yelp_academic_dataset_business.json --port 27021  
mongoimport --db yelpdb --collection checkin --drop --file
```

```
yelp_academic_dataset_checkin.json --port 27021
mongoimport --db yelpdb --collection review --drop --file
yelp_academic_dataset_review.json --port 27021
mongoimport --db yelpdb --collection tip --drop --file
yelp_academic_dataset_tip.json --port 27021
mongoimport --db yelpdb --collection user --drop --file
yelp_academic_dataset_user.json --port 27021

#data imported from these json files
#- yelp_academic_dataset_business.json
#- yelp_academic_dataset_checkin.json
#- yelp_academic_dataset_review.json
#- yelp_academic_dataset_tip.json
#- yelp_academic_dataset_user.json
```

Creation of extra collections to assure the 1:1, 1:N, N:N relationships

To avoid this step execute the script ***populate_mongodb.bash***

To do that we created two collections: - Ceo: Can be a user or not and have associated one to many business, in the same way a business can have many ceo's. - BankAccount: A ceo have a unique bank account, and a bank account can be linked with only one Ceo.

The script used to generate the data can be found in the folder by the name ***generate_collections.py***. To execute the script you have to have installed the pymongo library and the Faker library.

It's not mandatory to execute the script until you want to create new data. Because we already executed it and exported the generated data to JSON files provided in the folder ***extra_data***

Also another script to link the Ceo collection with the business collection should be executed mandatory for some queries to work ***associate_collections.py***

Other manipulations

The date in the ***tip*** collection is a string. To prepare the data to be ready for the queries we include a script that change the type of the field from string to date. That's done in the script ***date_conversion.js*** that's executed at the end of the ***populate_mongodb.bash*** script