MongoDB basics notes:

Create database: use DATABASE\_NAME eg use nyxdb

Drop database: db.dropDatabase()

Create collection(table): db.createCollection(‘TABLENAME’)

Or you can directly insert without creating the collection, mongo automatically creates that collection

Code:

> db.users.insert([

...{

... title: "MongoDB Overview",

... description: "MongoDB is no sql database",

... by: "tutorials point",

... url: "http://www.tutorialspoint.com",

... tags: ['mongodb', 'database', 'NoSQL'],

... likes: 100

... },

...{

... title: "MS SQL Overview",

... description: "MS SQL is sql database",

... by: "tutorials point",

... url: "http://www.tutorialspoint.com",

... tags: ['mongodb', 'database', 'NoSQL'],

... likes: 100

... }

...])

*The above code creates a user collection and inserts two rows inside it. Note that rows are comma separated and inside a list. Also note the format in which data is stored and retrieved is JSON for mongoDB*

For insert there are also insertOne and insertMany which means by the naming convention

Query the records (like select statement): we use the find() function like this -> db.users.find() if you want to print with proper spacings then it’s db.users.find().pretty()

By default, all rows have a unique id created by mongoDB

Some filtering done in mongo:

* db.users.find({ “title” : “MongoDB Overview”}).pretty() #returns records having title = ‘MongoDB Overview’
* db.users.find({“likes” : { $lt : 200 }}).pretty() #returns records having less than 200 likes
* $lte – less than or equal to
* $gt – greater than
* $gte – greater than or equal to
* $ne – not equals
* db.users.find({"name":{$in:["Raj", "Ram", "Raghu"]}}).pretty() //finds if any one from the list matches ($in)
* $nin – not in list
* db.users.find({ $and : [ {‘title’: ‘MongoDB Overview} ,{ ‘name’ : ‘Raj’}]}).pretty //And operation checks if title and name match
* $or – or operation
* $not – not operation

Update in mongo:

Code:

>db.users.update({'title':'MongoDB Overview'},{$set:{'title':'New MongoDB Tutorial'}})

The below code finds one record and updates it , similarly there are also updateOne and updateMany

db.empDetails.findOneAndUpdate(

{First\_Name: 'Radhika'},

{ $set: { Age: '30',e\_mail: 'radhika\_newemail@gmail.com'}}

)

Remove/Deleting a row in mongo:

>db.users.remove({'title':'MongoDB Overview'}) #removes all rows having that title

>db.users.remove({'title':'MongoDB Overview',1}) #removes only one row even if there are more rows with same title . Note the second parameter that is 1

For mongo connection in python :

From pymongo import MongoClient #import

mongo = MongoClient(‘mongoclient://0.0.0.0.:27017/nyxdb’) #connection

db = mongo.nyxdb # connect to db

reports = db.reports # connect to reports table in that db

for k in reports.find(“YOUR QUERY”):

# play with k