---------------------------------------------------------------------------------------------------------------------------

TO get sample JSON data go to…  
[www.json-generator.com](http://www.json-generator.com)----  
-----------------------------------------------------------------------------------------------------------------------

 NoSQL (originally referring to "non SQL" or "non relational") database provides a mechanism for storage and retrieval of data which is modelled in means other than the tabular relations used in relational databases.  
  
**\*\*relational database** - a database structured in which there exists relations between stored items of information.

**mongoDB**

>> NoSQL database

>> cd C:\Program Files\MongoDB\Server\3.2\bin

**It has two important files**  
>> mongo  
(for writing commands and stuff)  
>> mongod  
(running at the background as a server, as a database)

IF C:\data\db is not found…  
1. Manually create it…  
OR  
2. >> mkdir \data\db  
  
---------------------------------------------------------------------------------------------------------------------------

Instead of going every time to C:\Program Files\MongoDB\Server\3.2\bin, follow this procedure

>>control panel \ system and security \ system  
>> advanced system settings\ advanced\ environment variable  
>> in PATH, add C:\Program Files\MongoDB\Server\3.2\bin  
  
---------------------------------------------------------------------------------------------------------------------------

>>command prompt1--->mongod  
>>command prompt2--->mongo  
  
THEN  
>>show dbs  
(to show all the databases)  
  
>> use dbName  
(creates as well as switch to the database)

>> db  
(tells which db you are currently working on)  
  
>> db.getCollectionNames()  
(to get names of all the collections in that particular db)

>> show collections  
(to view all collections)

>> db.dropDatabase()  
(to drop database)

>> db.players.insert(

|  |
| --- |
| { |
|  | "position":"Right Wing", |
|  | "id":8465166, |
|  | "weight":200, |
|  | "height":"6' 0\"", |
|  | "imageUrl":"http://1.cdn.nhle.com/photos/mugs/8465166.jpg", |
|  | "birthplace":"Seria, BRN", |
|  | "age":37, |
|  | "name":"Craig Adams", |
|  | "birthdate":"April 26, 1977", |
|  | "number":27 |
|  | } |

)  
IF “players” collection doesn’t exist, it’s gonna create a “players” collection and then insert data in it.  
  
  
  
To insert an array  
>> db.players.insert([   
{ },  
{ },  
{ }  
 ])

>> db.players.find()  
(to view all objects in players collection)  
  
  
>> db.players.find().pretty()

MONGO adds a new ID to every object you insert in your database to keep it UNIQUE

>> db.players.remove(

{ "\_id" : ObjectId("56a3e0c67d9c03b6145a68d8") }

)  
  
  
>> db.players.update(  
{ "\_id" : ObjectId("56a3e0c67d9c03b6145a68d8")},

|  |
| --- |
| { |
|  | "position":"LEFT Wing", |
|  | "id":8471887, |
|  | "weight":189, |
|  | "height":"5' 11\"", |
|  | "imageUrl":"http://1.cdn.nhle.com/photos/mugs/8471887.jpg", |
|  | "birthplace":"Sollentuna, SWE", |
|  | "age":28, |
|  | "name":"Patric Hornqvist", |
|  | "birthdate":"January 01, 1987", |
|  | "number":72 |
|  | } |

)

**Basically Syntax for update is**  
  
>> db.players.update(  
{object ID},  
{new details}  
)  
  
  
NOTE : the {new details} should contain all the details of that object, details which are getting modified and also the details which are gonna remain the same.  
  
  
>> db.players.drop()  
(to drop a collection)

>> db.players.find( {} )  
(to find specific data)

>> db.players.find( {a,b,c} )  
(DOING AND for conditions a,b,c)  
  
Example:

>> db.players.find(  
{ “age” : 30,   
 “name” : “peter” })

>> db.players.find( { $or:[ {a},{b} ] } )  
(DOING OR for conditions a,b)  
  
Example:   
  
>>db.players.find({$or:[{"position":"Left Wing"},{"position":"Right Wing"}]})  
  
  
  
$gt, $gte, $lt, $lte, $ne  
  
gt : greater than  
lt : less than  
ne: not equal  
gte: greater than equal  
  
  
  
  
>> db.players.find({“age”: {$gt:30}})

IF you don’t want to return all the attributes/columns, then   
>> db.players.find(  
{“age”:30},  
{“name”:1, \_id:0}

)

\*\*NOTE  
(For attributes name, position, age etc. you will have to mention “attributeName”:1 for it to appear and nothing for it to disappear.   
BUT for object id you need to explicitly mention “\_id”:0 for it to disappear)

>> db.players.find({“age”:30}).limit(2)  
(to limit rows)

>> db.players.find({“age”:30}).skip(2)  
(to skip rows)

>> use bank  
>> db.users.insert([ {},{},{},{}])  
  
INDEXING- for efficiency when dealing with large sites  
  
//ignored as of now, read if you’re doing project with large database.  
  
  
  
>> db.users.aggregate({

$group :

{

\_id : "$eyeColor",

total : {$sum:1}

}} )  
  
(It groups the data about people having same eye color and then count the number of items in each group)  
  
  
  
  
>>   
db.users.aggregate({

$group :

{

\_id : "$gender",

avgAge : {$avg : “$age”}

}} )  
  
(It groups the data about different genders and calculate the average of the attribute “age” in each group)  
  
  
  
>>

db.users.aggregate({

$group :

{

\_id : "$gender",

richest : {$max : “$balance”}

}} )

---------------------------------------------------------------------------------------------------------------------------

TO get sample JSON data go to…  
[www.json-generator.com](http://www.json-generator.com)----  
-----------------------------------------------------------------------------------------------------------------------