CLASS-2

Few Commands to test after connections

Command	Notes
show dbs	All Databases are shown
use db	Create and use db
db	View current database
db.dropDatabase()	To drop(delete) the database
db.createCollection(name)	To create a new collection
show collections	Shows collection
db.(collection name).insert(document)	To insert document in collection
db.(collection_name).drop()	To drop the selected collection
db.(collection name).batchInsert([{"_id" : 0}, {"_id" : 1}, {"_id" : 2}])	To insert more than one document
db(name).find()	Print all rows
db.(name).remove()	Remove (name) table

Documents, Database, Collections:

Databases, collections, and documents are important parts of MongoDB; without them you are not able to store data on the MongoDB server. A Database contains a collection, and a collection contains documents and the documents contain data, they are related to each other.

Database - In MongoDB, a database contains the collections of documents. One can create multiple databases on the MongoDB server.

Collections- Collections are just like tables in relational databases, they also store data, but in the form of documents. A single database is allowed to store multiple collections.

Naming restrictions for collections:

Before creating a collection you should first learn about the naming restrictions for collections:

- Collection name must start with an underscore or a character.
- Collection name does not contain $, empty string, null character and does not begin with system prefix.
- The maximum length of the collection name is 120 bytes.

Documents: At the heart of MongoDB is the document: an ordered set of keys with associated values.

The representation of a document varies by programming language, but most languages have a data structure that is a natural fit, such as a map, hash, or dictionary.

{"greeting" : "Hello, world!"}

Datatypes:

- **String** − This is the most commonly used datatype to store the data. The string in MongoDB must be UTF-8 valid.
- **Integer** − This type is used to store a numerical value. Integer can be 32-bit or 64-bit depending upon your server.
- **Boolean** − This type is used to store a boolean (true/ false) value.
- **Double** − This type is used to store floating point values.
- **Min/ Max keys** − This type is used to compare a value against the lowest and highest BSON elements.
- **Arrays** − This type is used to store arrays or lists or multiple values into one key.
- **Timestamp** − This can be handy for recording when a document has been modified or added.
- **Object** − This datatype is used for embedded documents.
- **Null** − This type is used to store a Null value.
- **Symbol** − This datatype is used identically to a string; however, it's generally reserved for languages that use a specific symbol type.
- **Date** − This datatype stores the current date or time in UNIX time format. You can specify your own date time by creating object of Date and passing day, month, year into it.
- **Object ID** − This datatype is used to store the document's ID.
- **Binary data** − This datatype is used to store binary data.
- **Code** − This datatype is used to store JavaScript code into the document.
- **Regular expression** − This datatype is used to store regular expression.

Loading the document

- Download the student csv from the link
- Import the data to the collection created-

  MongoDB Compass can import and export data to and from collections. Compass supports import and export for both **JSON** and **CSV** files. link
- You should be able to see the uploaded data in mongo compass