Mongo db

class-2:Add,update,delete

Commands:

Command	Expected Output	Notes
show dbs	admin 40.00 KiB config 72.00 KiB db 128.00 KiB local 40.00 KiB	All Databases are shown
use db	switched to db db	Connect and use db
show collections	Students	Show all tables
db.foo.insert({"bar" : "baz"})		Insert a record to collection. Create Collection if not exists

Show dbs command description:

The show dbs shell command corresponds to the list Databases admin command, which shows the size on disk for each database. The size for sharded collections is the sum of one member for each shard, and does not include redundant storage usage across all members of the shard replica set.

Use db command description:

The MongoDB command interface provides access to all non CRUD database operations. Fetching server statistics, initializing a replica set, and running an aggregation pipeline or map-reduce job are all accomplished with commands.

Show collections command description:

For users with the required access, show collections lists the non-system collections for the database. For users without the required access, show collections lists only the collections for which the users has privileges.

Command	**Notes**
db.foo.batchInsert([{"_id" : 0}, {"_id" : 1}, {"_id" : 2}])	Insert more than one document
db.foo.find()	Print all rows
db.foo.remove()	Remove foo table

➢ Documents,collections,database:

Document:

A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays, and arrays of documents.

Collections:

MongoDB stores data records as documents (specifically BSON documents) which are gathered together in collections. A database stores one or more collections of documents. You can manage MongoDB databases and collections in the UI for deployments hosted in MongoDB Atlas.

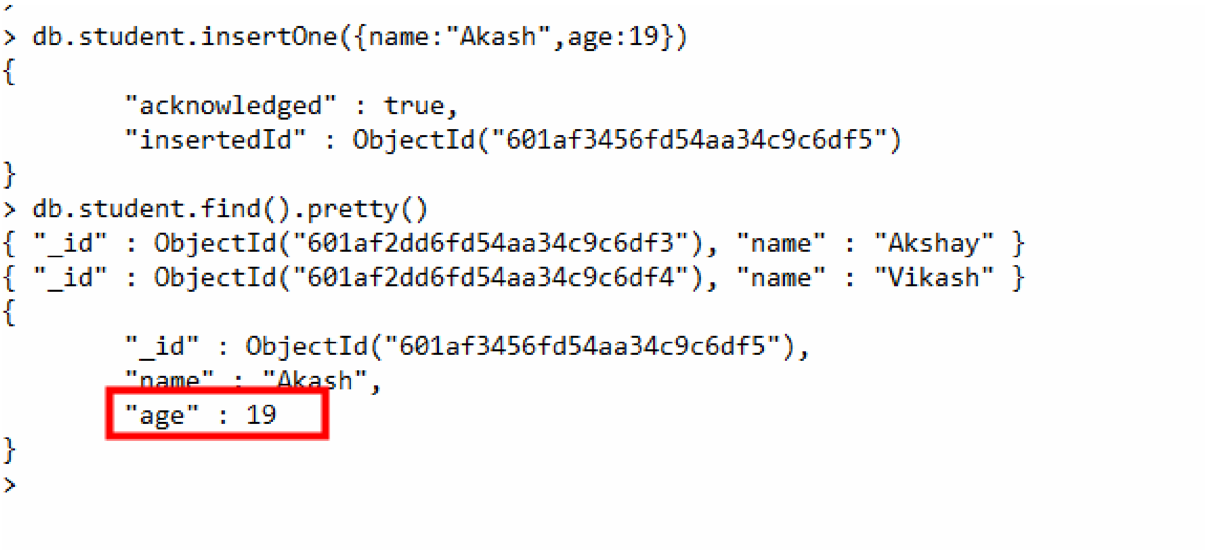
Database:

MongoDB is a non-relational document database that provides support for JSON-like storage. The MongoDB database has a flexible data model that enables you to store unstructured data, and it provides full indexing support, and replication with rich and intuitive APIs.

Datatype:

In MongoDB, the integer data type is used to store an integer value. We can store integer data type in two forms 32 -bit signed integer and 64 – bit signed integer.

Example: In the following example we are storing the age of the student in the student collection:



```
> db.student.insertOne({name:"Akash",age:19})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("601af3456fd54aa34c9c6df5")
}
> db.student.find().pretty()
{ "_id" : ObjectId("601af2dd6fd54aa34c9c6df3"), "name" : "Akshay" }
{ "_id" : ObjectId("601af2dd6fd54aa34c9c6df4"), "name" : "Vikash" }
{
        "_id" : ObjectId("601af3456fd54aa34c9c6df5"),
        "name" : "Akash",
        "age" : 19
}
>
```