Mongo Db Commands

mongod –version: for checking version

mongosh: connect to mongo db shell

show dbs: display all the databases

show collections: display all the tables

use company: creates a database named company

db.createCollection(“emp”): creates a table/collection and makes the database active

db.Employee.insertOne({ID: 1, Name:”Pratyush”}): creates a table Employee and inserts data in it

db.Employee.find(): retrieves all the records

db.Employee.insertMany([{}]): insert multiple records

db.Employee.updateOne({name: “Prayushi”},{$set:{department:”Finance”}}): update one record

db.Employee.updateMany({},{$set:{Hobby:"Procrastinating"}}): add column with common value for all collections

db.Employee.updateOne({Name:'Aaditya'},{$set:{idcards:{hasPanCard:true, hasAadharCard:true}}}): for nested document

db.Employee.find({'idcards.hasAadharCard':true}): will display only those entry which have hasAadharCard as true

db.Employee.find().count(): gives the total number of records

db.Employee.find({Salary:{$gt:60000}}): displays records having salary greater than 60000.  
\* For the above query we can also use $lt, $gte, $lte, $eq, $ne \*

db.Employee.updateMany({},{$inc:{id:1}}): increments id by 1 everytime we create a new record

db.Employee.find({Name: {$in:['Aaditya', 'Santosh']}}): similar to in operator in sql

db.Employee.find({Name: {$nin:['Aaditya', 'Santosh']}}): similar to not in operator in sql

db.Employee.find().skip(2).limit(2): similar to offset and limit in sql

db.Employee.find().sort({Salary: 1}): similar to order by ASC

db.Employee.find().sort({Salary: -1}): similar to order by DESC

db.Employee.deleteOne({Name:'Krushna'}): delete 1 record

db.Employee.aggregate([{$group:{\_id:'$Hobby', count:{$sum:1}}}]): similar to group by clause in sql