1. **To connect MongoDB shell to your Atlas Cluster(movieDetails Database)-->**  
   mongo "mongodb+srv://cluster0-f2zzi.mongodb.net/test" --username m001-student --password m001-mongodb-basics
2. **To Connect MongoDB shell to your MongoDB university Database(movies, 100yearWeather)-->**mongo "mongodb://cluster0-shard-00-00-jxeqq.mongodb.net:27017,cluster0-shard-00-01-jxeqq.mongodb.net:27017,cluster0-shard-00-02-jxeqq.mongodb.net:27017/100YWeatherSmall?replicaSet=Cluster0-shard-0" --authenticationDatabase admin --ssl --username m001-student --password m001-mongodb-basics
3. **To insert many records in a document**db.moviesScratch.insert(  
    [  
    {  
    “title”:”Yadein”,  
    “year” : “2052”,  
    “type”: “documentary”   
    },  
    {  
    “title”:”Zameer”,  
    “year”:”1947”,  
    “type”: “documentary”  
    }  
    ]  
   );

**If there is any problem with one document while inserting the mongo stops from the error part, to remove this we need to add options field and add “ordered”:false**

PTO

db.moviesScratch.insertMany(

[

{ "title":"Zohraah", "year" : "2051", "type": "documentary" },

{ "title":"Yahhah", "year":"1948", "type": "documentary" }

],

{

"ordered":false

}

);

1. **In Movies Collection Find Documents where mpaaRating is “PG-13”**  
   db.movies.find({mpaaRating: "PG-13"}).pretty();
2. **In Movies Collection Find Documents where mpaaRating is “PG-13” and year is 2009**db.movies.find({mpaaRating: "PG-13", year:2009}).pretty();
3. **In 100YWeatherSmall Find Documents where angle of direction of wind is 290**   
   db.data.find({"wind.direction.angle": 290}).pretty();
4. **In Movies Collection Find Documents where First Cast is Jeff Bridges.**db.movies.find({"cast.0": "Jeff Bridges"}).pretty();
5. **In Movies Collection Find Documents where One of the Cast is Jeff Bridges.**db.movies.find({"cast": "Jeff Bridges"}).pretty();
6. **How many movies in the movieDetails collection list "Family" among its genres?**db.movieDetails.find({"genres":"Family"}).count();
7. **How many movies in the movieDetails collection list "Western" second among its genres?**db.movieDetails.find({"genres.1":"Western"}).count();
8. **How can I view only title of the movies in the movieDetails collection list "Western" second among its genres?Just the title**Using Projections.  
   db.movieDetails.find({"genres.1":"Western"},{"title":1, “\_id”:0}).pretty();
9. **Update The Martian movie where the poster field is missing**For the update we will use a **unique identifier**(title:’**The Martian’**) to specify the document  
   For **Update operator** we will use **$set.**

db.movieDetails.update(

{title: 'The Martian'},

{

$set: {

poster:'https://ih0.redbubble.net/image.400643184.3136/flat,550x550,075,f.u2.jpg'

}

});

1. **Update the Tomato rating by 1.2 and Tomato review by 3000 in The Martian movie**db.movieDetails.update(

{title: 'The Martian'},

{

$inc: {

"tomato.reviews": 3000,

"tomato.rating": 1.1

}

}

);

1. **Add document to an array**let rvTxt1 = [

"A",

"great",

"film"

].join()  
  
  
db.movieDetails.update({

"title": "The Martian"

}, {

$push: {

"reviews": {

"rating": 9,

"reviewer": "John",

"text": rvTxt1

}

}

}

);

1. **Add multiple document to an array as arr[0], arr[1], arr[2]**let txt1 = [

"Awesome movie",

"Must Watch For Everyone",

"Great Cast and Direction"

].join()

let txt2 = [

"Aamazaing",

"Must Watch For Everyone",

"Great Cast and Direction"

].join()

let txt3 = [

"WOoohooooo. Gonna watch it again",

"Must Watch For Everyone",

"Great Cast and Direction"

].join()

db.movieDetails.update(

{

"title": "The Martian"

},

{

$push: {

"reviews": {

$each: [

{

"rating": 7.5,

"reviewer": "Katy",

"text": txt1

},

{

"rating": 5.5,

"reviewer": "Paul",

"text": txt2

},

{

"rating": 7.9,

"reviewer": "Scholes",

"text": txt3

},

]

}

}

});