**Create database EMP and Make Collection With name "EMPL" and Follow Queries**  
  
**=>Created Database**  
> use emp  
 switched to DB emp  
  
**=>Create Collection**  
> db.createCollection("empl")  
 **=>Insert Records Into EMPL Collection**  
> db.empl.insert([  
 {no:1,name:"ST",salary:2000,role:"OB"},  
 {no:2,name:"MSD",salary:1500,role:"WK"},  
 {no:3,name:"YS",salary:1000,role:"ALR"},  
 {no:4,name:"RD",salary:1000,role:"MOB"},  
 {no:5,name:"RS",salary:500,role:"OB"},  
 {no:6,name:"BK",salary:500,role:"MOB"},  
 {no:7,name:"VK",salary:300,role:"BW"},  
 {no:8,name:"JB",salary:400,role:"BW"},  
 {no:9,name:"HP",salary:400,role:"ALR"},  
 {no:10,name:"VS",salary:300,role:"OB"}])  
 **=>Display Data in Proper Format**   
>db.empl.find().pretty()  
 **=>Update Salary Of Employee where Name is "ST" by +8000**  
>db.empl.update({name:"ST"},{[**$inc:**](https://codingonly4u.blogspot.com/2018/07/mongodb-exercises-practice-solution.html){salary:8000}})  
 **=>Update Salary Of All Employee by giving an increment of +4000 each**  
>db.empl.update({},{**$inc:**{salary:4000}},{multi:true})  
 **=>update role of "MSD" as "C and WK"**  
>db.empl.update({name:"MSD"},{**$set:**{role:"c and WK"}})  
 **=>Add a New Field remark to document with name "RS" set Remark as WC**  
>db.emp.update({name:"RS"},{**$set:**{remark:"WC"}})  
 **=>Add a New Field As Number 11,name AK,Salary 10000,role coch without using insert statement. But for Doing So You should have a Record Added woth number 11.**  
>db.empl.update({no:11},{**$set:**{no:11,name:"AK",salary:10000,role:"coch"}},{upsert:true})  
 **=>remove added New Field**  
>db.empl.update({name:"RD"},{**$unset:**{remark:"WC"}})  
 **=>Update the Field "RD" by multiplying with salary by 2**  
>db.empl.update({name:"RD"},{**$mul:**{salary:2}})  
  
**=>To Find Document From the empl collection where name begins with S**  
>db.empl.find({name:**/^**S**/**})  
  
**=>To Find Document From the empl collection where name begins with R**  
>db.empl.find({name:**/^**R**/**})  
  
**=>To Find Document From the empl collection where name ends with K**  
>db.empl.find({name:**/**K**$/**})  
  
**=>To Find Document From the empl collection where name ends with D**  
>db.empl.find({name:**/**S**$/**})  
  
**=>To Find Document From the empl collection where name has S in any position**  
>db.empl.find({name:**/**S**/**})  
ㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡ  
**Regular Expression**  
ㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡ **(Note: Use Case sensitive allow For that write in Option: "i")**  
  
**=> To Find Document From the empl collection where name begins with S**  
>db.empl.find({name:{**$regex:**"^S"}})  
  
**=> To Find Document From the empl collection where name begins with S**  
>db.empl.find({name:{**$regex**:"S",**$options**:"i"}})  
ㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡ  
 **Use of $in and $nin (**[**in and notin**](https://codingonly4u.blogspot.com/2018/07/mongodb-exercises-practice-solution.html)**)**  
ㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡ

### (Note: There will not use {} braces in that $in and $nin)

**=> Display Documents where in empl collection Field have OB,MOB**  
>db.empl.find({role:{**$in**:["OB","MOB"]}})  
  
**=> Display Documents where in empl collection Field not have OB,MOB**  
>db.empl.find({role:{**$nin**:["OB","MOB"]}})

**MongoDB Example: Insert Multiple Documents in collection**

var beginners =

[

{

"StudentId" : 1001,

"StudentName" : "Steve",

"age": 30

},

{

"StudentId" : 1002,

"StudentName" : "Negan",

"age": 42

},

{

"StudentId" : 3333,

"StudentName" : "Rick",

"age": 35

},

];

db.students.insert(beginners);

## Insert multiple documents:

## Create a products

**[ // array of documents**  
{  
"name": "iPhone 11",  
"description": "Cell with lot of cameras",  
"price": 400  
},  
{  
"name": "samsung galaxy s10",  
"description": "A good option with mid range budget",  
"price": 350  
},  
{  
"name": "Huawei mate 20 pro",  
"description": "Worth the money",  
"price": 300  
}  
**]**

**Practice question**

Insert the following documents into a movies collection.

title : Fight Club

writer : Chuck Palahniuk

year : 1999

actors : [

Brad Pitt

Edward Norton

]

title : Pulp Fiction

writer : Quentin Tarantino

year : 1994

actors : [

John Travolta

Uma Thurman

]

title : Inglorious Basterds

writer : Quentin Tarantino

year : 2009

actors : [

Brad Pitt

Diane Kruger

Eli Roth

]

title : The Hobbit: An Unexpected Journey

writer : J.R.R. Tolkein

year : 2012

franchise : The Hobbit

title : The Hobbit: The Desolation of Smaug

writer : J.R.R. Tolkein

year : 2013

franchise : The Hobbit

title : The Hobbit: The Battle of the Five Armies

writer : J.R.R. Tolkein

year : 2012

franchise : The Hobbit

synopsis : Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.

title : Pee Wee Herman's Big Adventure

title : Avatar

**Query / Find Documents**

query the movies collection to

1. get all documents
2. get all documents with writer set to "Quentin Tarantino"
3. get all documents where actors include "Brad Pitt"
4. get all documents with franchise set to "The Hobbit"
5. get all movies released in the 90s
6. get all movies released before the year 2000 or after 2010

**Delete Documents**

1. delete the movie "Pee Wee Herman's Big Adventure"
2. delete the movie "Avatar"