

# BDA LAB 1 : CODE(Command Prompt)

```
Command Prompt - mongo
Microsoft Windows [Version 10.0.18362.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\AJAY>cd C:\Program Files\MongoDB\Server\4.0\bin

C:\Program Files\MongoDB\Server\4.0\bin>mongo
MongoDB shell version v4.0.20-36-gf8006ee
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongod
Implicit session: session { "id" : UUID("db69073a-865e-4a11-bff6-3f66bda4ca28") }
MongoDB server version: 4.0.20-36-gf8006ee
Server has startup warnings:
2020-09-24T10:02:14.339+0530 I CONTROL [initandlisten]
2020-09-24T10:02:14.339+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-09-24T10:02:14.339+0530 I CONTROL [initandlisten] **      Read and write access to data and configuration is unrestricted.
2020-09-24T10:02:14.340+0530 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> use myDB;
switched to db myDB
> db;
myDB
> show dbs;
admin    0.000GB
config  0.000GB
local    0.000GB
> db.createCollection("Student");
{ "ok" : 1 }
> db.createCollection("Faculty");
{ "ok" : 1 }
> db.createCollection("COE");
{ "ok" : 1 }
> db.createCollection("Library");
{ "ok" : 1 }
> db.createCollection("Admission");
```

```
Command Prompt - mongo
{ "ok" : 1 }
> db.createCollection("Admission");
{ "ok" : 1 }
> db.Student.insert({_id:1,name:"Rounak",sem:7,dept:"CSE"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:2,name:"Sai",sem:7});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:3,name:"Saif"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:4,name:"Rohit",dept:"CSE"});
WriteResult({ "nInserted" : 1 })
>
>
> db.Student.find({name:"Rounak"});
{ "_id" : ObjectId("5f6c2a5c2aa9d272e6295036"), "id" : 1, "name" : "Rounak", "sem" : 7, "dept" : "CSE" }
> db.Student.find({name:"Rohit"});
{ "_id" : ObjectId("5f6c2aae2aa9d272e6295039"), "id" : 4, "name" : "Rohit", "dept" : "CSE" }
> db.Student.find({name:"Saif"});
{ "_id" : ObjectId("5f6c2a782aa9d272e6295038"), "id" : 3, "name" : "Saif" }
> db.Student.find({name:"Sai"});
{ "_id" : ObjectId("5f6c2a6d2aa9d272e6295037"), "id" : 2, "name" : "Sai", "sem" : 7 }
>
>
> db.Faculty.insert({_id:1,name:"Rahul",dept:"CSE",sal:10000});
WriteResult({ "nInserted" : 1 })
> db.Faculty.insert({_id:2,name:"Ritesh",dept:"CSE"});
WriteResult({ "nInserted" : 1 })
> db.Faculty.insert({_id:3,name:"Ridhi",dept:"ISE"});
WriteResult({ "nInserted" : 1 })
> db.Faculty.insert({_id:4,name:"Ridhika"});
WriteResult({ "nInserted" : 1 })
>
> db.Faculty.find({name:"Rahul"});
{ "_id" : 1, "name" : "Rahul", "dept" : "CSE", "sal" : 10000 }
> db.Faculty.find({name:"Ritesh"});
{ "_id" : 2, "name" : "Ritesh", "dept" : "CSE" }
> db.Faculty.find({name:"Ridhika"});
{ "_id" : 4, "name" : "Ridhika" }
> db.Faculty.find({name:"Ridhi"});
{ "_id" : 3, "name" : "Ridhi", "dept" : "ISE" }
>
>
> db.COE.insert({_id:1,emp:"Abhi",sal:10000});
WriteResult({ "nInserted" : 1 })
```

Command Prompt - mongo

```
> db.COE.insert({_id:2,emp:"Abhishek",sal:10000});
WriteResult({ "nInserted" : 1 })
> db.COE.insert({_id:3,emp:"Tushar",sal:1000});
WriteResult({ "nInserted" : 1 })
> db.COE.insert({_id:4,emp:"Ramesh"});
WriteResult({ "nInserted" : 1 })
>
> db.COE.find(_id:1);
2020-09-24T10:52:33.699+0530 E QUERY    [js] SyntaxError: missing ) after argument list @(shell):1:15
> db.COE.find({_id:1});
{ "_id" : 1, "emp" : "Abhi", "sal" : 10000 }
> db.COE.find({_id:2});
{ "_id" : 2, "emp" : "Abhishek", "sal" : 10000 }
> db.COE.find({_id:3});
{ "_id" : 3, "emp" : "Tushar", "sal" : 1000 }
> db.COE.find({_id:4});
{ "_id" : 4, "emp" : "Ramesh" }
>
>
>
> db.Library.insert({_id:1,book_name:"Panchatantra",cost:1000});
WriteResult({ "nInserted" : 1 })
> db.Library.insert({_id:2,book_name:"Ramayana",cost:100});
WriteResult({ "nInserted" : 1 })
> db.Library.insert({_id:2,book_name:"Mahabharat",cost:500});
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "E11000 duplicate key error collection: myDB.Library index: _id_dup key: { : 2.0 }"
  }
})
> db.Library.insert({_id:3,book_name:"Mahabharat",cost:500});
WriteResult({ "nInserted" : 1 })
> db.Library.insert({_id:4,book_name:"Gita",cost:5090});
WriteResult({ "nInserted" : 1 })
>
>
>
> db.Library.find({_id:1});
```

Command Prompt - mongo

```
> db.Library.find({_id:1});
{ "_id" : 1, "book_name" : "Panchatantra", "cost" : 1000 }
> db.Library.find({_id:2});
{ "_id" : 2, "book_name" : "Ramayana", "cost" : 100 }
> db.Library.find({_id:3});
{ "_id" : 3, "book_name" : "Mahabharat", "cost" : 500 }
> db.Library.find({_id:4});
{ "_id" : 4, "book_name" : "Gita", "cost" : 5000 }
>
> db.Library.find();
{ "_id" : 1, "book_name" : "Panchatantra", "cost" : 1000 }
{ "_id" : 2, "book_name" : "Ramayana", "cost" : 100 }
{ "_id" : 3, "book_name" : "Mahabharat", "cost" : 500 }
{ "_id" : 4, "book_name" : "Gita", "cost" : 5000 }
>
>
> db.Admission.insert({_id:1,name:"Rounak",fee:30000,dept:"CSE"});
WriteResult({ "nInserted" : 1 })
> db.Admission.insert({_id:2,name:"Sai",fee:30000});
WriteResult({ "nInserted" : 1 })
> db.Admission.insert({_id:3,name:"Saif",fee:130000});
WriteResult({ "nInserted" : 1 })
> db.Admission.insert({_id:4,name:"Rohit",fee:13000});
WriteResult({ "nInserted" : 1 })
>
> db.Admission.find();
{ "_id" : 1, "name" : "Rounak", "fee" : 30000, "dept" : "CSE" }
{ "_id" : 2, "name" : "Sai", "fee" : 30000 }
{ "_id" : 3, "name" : "Saif", "fee" : 130000 }
{ "_id" : 4, "name" : "Rohit", "fee" : 13000 }
> db.Admission.find();
{ "_id" : 1, "name" : "Rounak", "fee" : 30000, "dept" : "CSE" }
{ "_id" : 2, "name" : "Sai", "fee" : 30000 }
{ "_id" : 3, "name" : "Saif", "fee" : 130000 }
{ "_id" : 4, "name" : "Rohit", "fee" : 13000 }
>
> db.College-festival.insert({_id:1,name:"UTSAV",cost:100000,no_of_students:1000});
```

Command Prompt - mongo

```
>
> db.College-festival.insert({_id:1,name:"UTSAV",cost:100000,no_of_students:1000});
2020-09-24T11:00:55.451+0530 E QUERY [js] ReferenceError: festival is not defined :
@(<shell>):1:1
> db.CollegeFestival.insert({_id:1,name:"UTSAV",cost:100000,no_of_students:1000});
WriteResult({ "nInserted" : 1 })
> db.CollegeFestival.insert({_id:1,name:"INFIX",cost:1000,no_of_students:100});
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "E11000 duplicate key error collection: myDB.CollegeFestival index: _id_ dup key: { : 1.0 }"
  }
})
> db.CollegeFestival.insert({_id:2,name:"INFIX",cost:1000,no_of_students:100});
WriteResult({ "nInserted" : 1 })
> db.CollegeFestival.insert({_id:3,name:"INFITIQ",cost:100,no_of_students:500});
WriteResult({ "nInserted" : 1 })
> db.CollegeFestival.insert({_id:4,name:"PHASE SHIFT",cost:1008});
WriteResult({ "nInserted" : 1 })
>
> db.CollegeFestival.find();
{ "_id" : 1, "name" : "UTSAV", "cost" : 100000, "no_of_students" : 1000 }
{ "_id" : 2, "name" : "INFIX", "cost" : 1000, "no_of_students" : 100 }
{ "_id" : 3, "name" : "INFITIQ", "cost" : 100, "no_of_students" : 500 }
{ "_id" : 4, "name" : "PHASE SHIFT", "cost" : 1008 }
```

# BDA LAB 1 : OUTPUT(Command Prompt)

```
Command Prompt - mongo
> show collections;
Admission
COE
CollegeFestival
Faculty
Library
Student

> db.Admission.find();
{"_id" : 1, "name" : "Rounak", "fee" : 30000, "dept" : "CSE" }
{"_id" : 2, "name" : "Sai", "fee" : 30000 }
{"_id" : 3, "name" : "Saif", "fee" : 130000 }
{"_id" : 4, "name" : "Rohit", "fee" : 13000 }

> db.COE.find();
{"_id" : 1, "emp" : "Abhi", "sal" : 10000 }
{"_id" : 2, "emp" : "Abhishek", "sal" : 10000 }
{"_id" : 3, "emp" : "Tushar", "sal" : 1000 }
{"_id" : 4, "emp" : "Ramesh" }

> db.CollegeFestival.find();
{"_id" : 1, "name" : "UTSAV", "cost" : 100000, "no_of_students" : 1000 }
{"_id" : 2, "name" : "INFIX", "cost" : 1000, "no_of_students" : 100 }
{"_id" : 3, "name" : "INFITIQU", "cost" : 100, "no_of_students" : 500 }
{"_id" : 4, "name" : "PHASE SHIFT", "cost" : 1008 }

> db.Faculty.find();
{"_id" : 1, "name" : "Rahul", "dept" : "CSE", "sal" : 10000 }
{"_id" : 2, "name" : "Ritesh", "dept" : "CSE" }
{"_id" : 3, "name" : "Ridhi", "dept" : "ISE" }
{"_id" : 4, "name" : "Ridhika" }

> db.Library.find();
{"_id" : 1, "book_name" : "Panchatantra", "cost" : 1000 }
{"_id" : 2, "book_name" : "Ramayana", "cost" : 100 }
{"_id" : 3, "book_name" : "Mahabharat", "cost" : 500 }
{"_id" : 4, "book_name" : "Gita", "cost" : 5090 }

> db.Student.find();
{"_id" : ObjectId("5f6c2a5c2aa9d272e6295036"), "id" : 1, "name" : "Rounak", "sem" : 7, "dept" : "CSE" }
{"_id" : ObjectId("5f6c2a6d2aa9d272e6295037"), "id" : 2, "name" : "Sai", "sem" : 7 }
{"_id" : ObjectId("5f6c2a782aa9d272e6295038"), "id" : 3, "name" : "Saif" }
{"_id" : ObjectId("5f6c2aae2aa9d272e6295039"), "id" : 4, "name" : "Rohit", "dept" : "CSE" }
```