

CSLR 51 : DBMS LAB-10

Roll no. : **106119100**

Name : **Rajneesh Pandey**

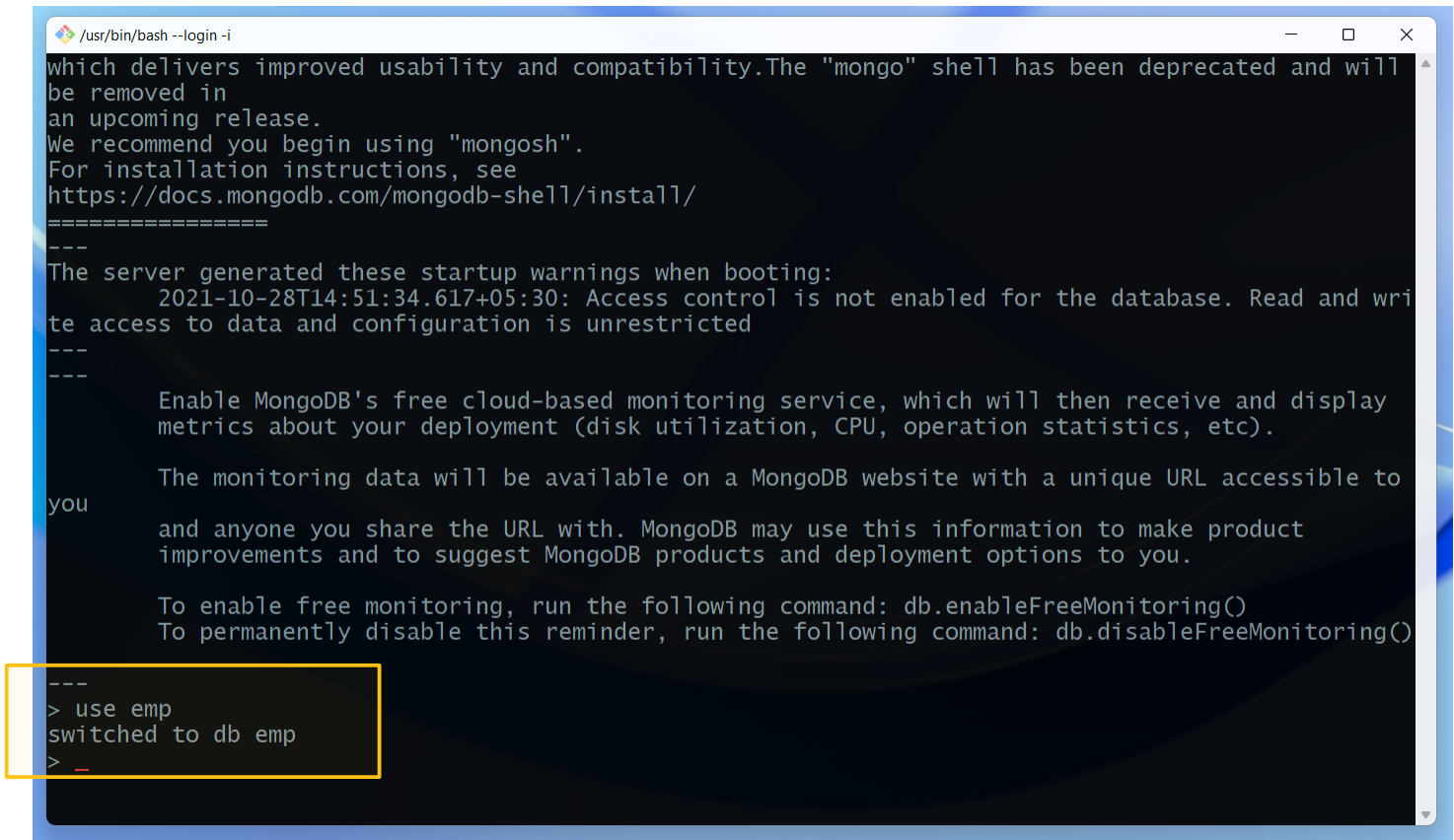
Section : **CSE-B**

-- Create database EMP and Make Collection With name "EMPL" and Follow Queries

--Created Database

use emp

switched to DB emp

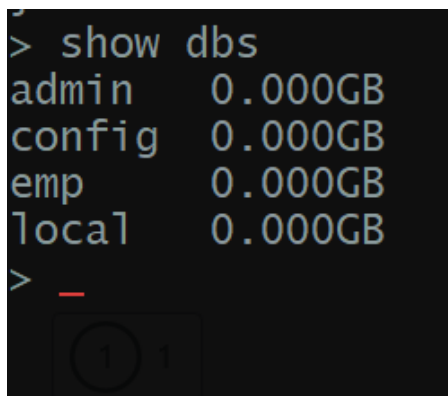


```
/usr/bin/bash --login -i
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will
be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
---
The server generated these startup warnings when booting:
  2021-10-28T14:51:34.617+05:30: Access control is not enabled for the database. Read and write
access to data and configuration is unrestricted
---
---
  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 emp
switched to db emp
> _
```

Database created :



```
> show dbs
admin      0.000GB
config     0.000GB
emp        0.000GB
local      0.000GB
> _
```

```
--Create Collection
db.createCollection("empl")
```

```
> db.createCollection("empl")
{ "ok" : 1 }
> _
```

```
--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"}])
```

```
> 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"}])
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 10,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
```

--Display Data in Proper Format

```
db.empl.find().pretty()
```

```
> db.empl.find().pretty()
{"_id": ObjectId("617a76b1ebdd290968913317"),
  "no": 1,
  "name": "ST",
  "salary": 2000,
  "role": "OB"
}

{"_id": ObjectId("617a76b1ebdd290968913318"),
  "no": 2,
  "name": "MSD",
  "salary": 1500,
  "role": "WK"
}

{"_id": ObjectId("617a76b1ebdd290968913319"),
  "no": 3,
  "name": "YS",
  "salary": 1000,
  "role": "ALR"
}

{"_id": ObjectId("617a76b1ebdd29096891331a"),
  "no": 4,
  "name": "RD",
  "salary": 1000,
  "role": "MOB"
}

{"_id": ObjectId("617a76b1ebdd29096891331b"),
  "no": 5,
  "name": "RS",
  "salary": 500,
  "role": "OB"
}
```

```
{
  "_id" : ObjectId("617a76b1ebdd29096891331c"),
  "no" : 6,
  "name" : "BK",
  "salary" : 500,
  "role" : "MOB"
}

{
  "_id" : ObjectId("617a76b1ebdd29096891331d"),
  "no" : 7,
  "name" : "VK",
  "salary" : 300,
  "role" : "BW"
}

{
  "_id" : ObjectId("617a76b1ebdd29096891331e"),
  "no" : 8,
  "name" : "JB",
  "salary" : 400,
  "role" : "BW"
}

{
  "_id" : ObjectId("617a76b1ebdd29096891331f"),
  "no" : 9,
  "name" : "HP",
  "salary" : 400,
  "role" : "ALR"
}

{
  "_id" : ObjectId("617a76b1ebdd290968913320"),
  "no" : 10,
  "name" : "VS",
  "salary" : 300,
  "role" : "OB"
}

--Update Salary Of Employee where Name
db.empl.update({name:"ST"},{$inc:{sala

--Update Salary Of All Employee by giv
db.empl.update({},{$inc:{salary:4000}})
```


--Update Salary Of Employee where Name is "ST" by +8000

```
db.empl.update({name:"ST"},{$inc:{salary:8000}})
```

```
> db.empl.update({name:"ST"},{$inc:{salary:8000}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> _
```

--Update Salary Of All Employee by giving an increment of +4000 each

```
db.empl.update({},{$inc:{salary:4000}},{multi:true})
```

```
> db.empl.update({},{$inc:{salary:4000}},{multi:true})
WriteResult({ "nMatched" : 10, "nUpserted" : 0, "nModified" : 10 })
> _
```

--update role of "MSD" as "C and WK"

```
db.empl.update({name:"MSD"},{$set:{role:"c and WK"}})
```

```
> db.empl.update({name:"MSD"},{$set:{role:"c and WK"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> _
```

--Add a New Field remark to document with name "RS" set Remark as WC

```
db.empl.update({name:"RS"},{$set:{remark:"WC"}})
```

```
> db.empl.update({name:"RS"},{$set:{remark:"WC"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> _
```

--Add a New Field As Number 11,name AK,Salary 10000,role coach without using insert statement. But for Doing So You should have a Record Added with number 11.

```
db.empl.update({no:11},{$set:{no:11,name:"AK",salary:10000,role:"coach"}},{upsert:true})
```

```
> db.empl.update({no:11},{$set:{no:11,name:"AK",salary:10000,role:"coch"}},{upsert:true})
WriteResult({
  "nMatched" : 0,
  "nUpserted" : 1,
  "nModified" : 0,
  "_id" : ObjectId("617a7867eb8bec42355c9b0d")
})
--remove added New Field
> db.empl.update({name:"RD"},{$unset:{remark:"WC"}})
```

--remove added New Field

```
db.empl.update({name:"RS"},{$unset:{remark:"WC"}})
```

```
> db.empl.update({name:"RS"},{$unset:{remark:"WC"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> _
```

--Update the Field "RD" by multiplying with salary by 2

```
db.empl.update({name:"RD"},{$mul:{salary:2}})
```

```
> db.empl.update({name:"RD"},{$mul:{salary:2}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> _
```

--To Find Document From the empl collection where name begins with S

```
db.empl.find({name:/^S/})
```

```
> db.empl.find({name:/^S/})
{ "_id" : ObjectId("617a76b1ebdd290968913317"), "no" : 1, "name" : "ST", "salary" : 14000, "role" : "OB" }
> _
```

Database After all the query

```
> db.empl.find().pretty()
{
  "_id" : ObjectId("617a883a71b2159c0e8fe3a6"),
  "no" : 1,
  "name" : "ST",
  "salary" : 14000,
  "role" : "OB"
}

{
  "_id" : ObjectId("617a883a71b2159c0e8fe3a7"),
  "no" : 2,
  "name" : "MSD",
  "salary" : 5500,
  "role" : "c and WK"
}
```

```

{
  "_id" : ObjectId("617a883a71b2159c0e8fe3a8"),
  "no" : 3,
  "name" : "YS",
  "salary" : 5000,
  "role" : "ALR"
}

{
  "_id" : ObjectId("617a883a71b2159c0e8fe3a9"),
  "no" : 4,
  "name" : "RD",
  "salary" : 10000,
  "role" : "MOB"
}

{
  "_id" : ObjectId("617a883a71b2159c0e8fe3aa"),
  "no" : 5,
  "name" : "RS",
  "salary" : 4500,
  "role" : "OB"
}

{
  "_id" : ObjectId("617a883a71b2159c0e8fe3ab"),
  "no" : 6,
  "name" : "BK",
  "salary" : 4500,
  "_id" : ObjectId("617a883a71b2159c0e8fe3ad"),
  "no" : 8,
  "name" : "JB",
  "salary" : 4400,
  "role" : "BW"
}

{
  "_id" : ObjectId("617a883a71b2159c0e8fe3ae"),
  "no" : 9,
  "name" : "HP",
  "salary" : 4400,
  "role" : "ALR"
}

{
  "_id" : ObjectId("617a883a71b2159c0e8fe3af"),
  "no" : 10,
  "name" : "VS",
  "salary" : 4300,
  "role" : "OB"
}

{
  "_id" : ObjectId("617a897aeb8bec42355c9b7b"),
  "no" : 11,
  "name" : "AK",
  "role" : "coach",
  "salary" : 10000
}

```