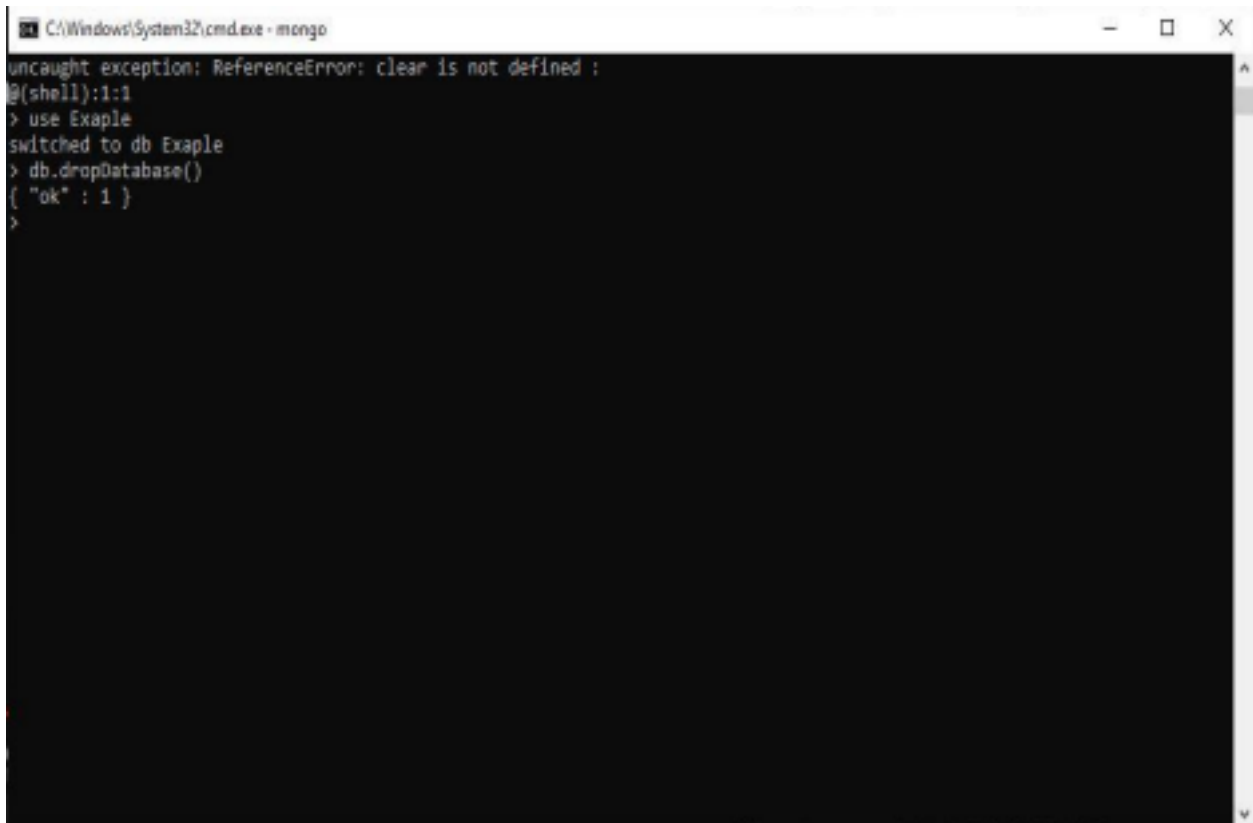


3
4
5

Practical No. 2:

Aim: Write a MongoDB query to create and drop databases. **Output:**



```
C:\Windows\System32\cmd.exe - mongo
uncaught exception: ReferenceError: clear is not defined :
@(shell):1:1
> use Exaple
switched to db Exaple
> db.dropDatabase()
{ "ok" : 1 }
>
```

6

Practical No. 3:

Aim: Write a MongoDB query to create, display and drop Collections.

Output:

```
C:\Windows\System32\cmd.exe - mongo
---
---
  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 college
switched to db college
> db.createCollection("students")
{ "ok" : 1 }
> show collections
students
> db.students.drop()
true
>
```

7

Practical No. 4:

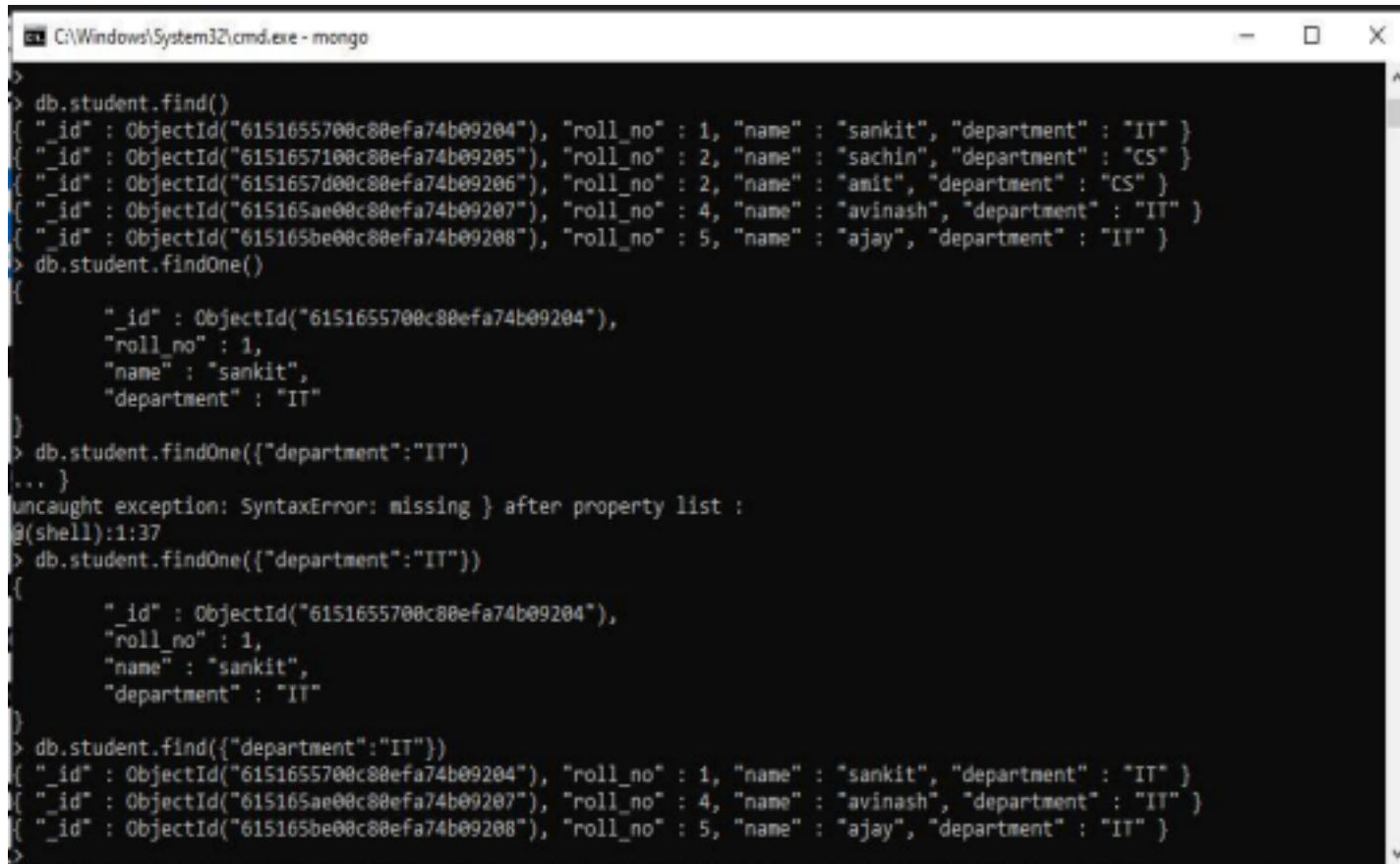
Aim: Write a MongoDB query to insert, query, update and delete Document.

Output:

Practical No. 5:

Aim: Simple query with MongoDB.

Output:



```
C:\Windows\System32\cmd.exe - mongo
> db.student.find()
{ "_id" : ObjectId("6151655700c80efa74b09204"), "roll_no" : 1, "name" : "sankit", "department" : "IT" }
{ "_id" : ObjectId("6151657100c80efa74b09205"), "roll_no" : 2, "name" : "sachin", "department" : "CS" }
{ "_id" : ObjectId("6151657d00c80efa74b09206"), "roll_no" : 2, "name" : "amit", "department" : "CS" }
{ "_id" : ObjectId("615165ae00c80efa74b09207"), "roll_no" : 4, "name" : "avinash", "department" : "IT" }
{ "_id" : ObjectId("615165be00c80efa74b09208"), "roll_no" : 5, "name" : "ajay", "department" : "IT" }
> db.student.findOne()
{
  "_id" : ObjectId("6151655700c80efa74b09204"),
  "roll_no" : 1,
  "name" : "sankit",
  "department" : "IT"
}
> db.student.findOne({"department":"IT"})
... }
uncaught exception: SyntaxError: missing } after property list :
@shell):1:37
> db.student.findOne({"department":"IT"})
{
  "_id" : ObjectId("6151655700c80efa74b09204"),
  "roll_no" : 1,
  "name" : "sankit",
  "department" : "IT"
}
> db.student.find({"department":"IT"})
{ "_id" : ObjectId("6151655700c80efa74b09204"), "roll_no" : 1, "name" : "sankit", "department" : "IT" }
{ "_id" : ObjectId("615165ae00c80efa74b09207"), "roll_no" : 4, "name" : "avinash", "department" : "IT" }
{ "_id" : ObjectId("615165be00c80efa74b09208"), "roll_no" : 5, "name" : "ajay", "department" : "IT" }
```

```
C:\Windows\System32\cmd.exe - mongo
> db.student.find({'$or':[{'name':'sachin','department':'IT'}]})
> db.student.find({'$or':[{'name':'sachin','name':'sankit'}]})
{ "_id" : ObjectId("6151655700c80efa74b09204"), "roll_no" : 1, "name" : "sankit", "department" : "IT" }
> db.student.find().count()
5
> db.student.find({"name":"sankit"},{"student_no":1,"Age":21});
{ "_id" : ObjectId("6151655700c80efa74b09204") }
> db.student.find({},{"name":"sankit"}).sort({"name":1})
{ "_id" : ObjectId("615165be00c80efa74b09208"), "name" : "sankit" }
{ "_id" : ObjectId("6151657d00c80efa74b09206"), "name" : "sankit" }
{ "_id" : ObjectId("615165ae00c80efa74b09207"), "name" : "sankit" }
{ "_id" : ObjectId("6151657100c80efa74b09205"), "name" : "sankit" }
{ "_id" : ObjectId("6151655700c80efa74b09204"), "name" : "sankit" }
> db.student.find({"name":"sankit"},{"student_no":1,"Age":21}).limit(4)
{ "_id" : ObjectId("6151655700c80efa74b09204") }
> db.student.find({"department":"IT"},{"student_no":1,"Age":21}).limit(4)
{ "_id" : ObjectId("6151655700c80efa74b09204") }
{ "_id" : ObjectId("615165ae00c80efa74b09207") }
{ "_id" : ObjectId("615165be00c80efa74b09208") }
> var c=db.student.find({"department":"IT"});
> while(c.hasNext()) printjson(c.next());
{
  "_id" : ObjectId("6151655700c80efa74b09204"),
  "roll_no" : 1,
  "name" : "sankit",
  "department" : "IT"
}
{
  "_id" : ObjectId("615165ae00c80efa74b09207"),
  "roll_no" : 4,
  "name" : "avinash",
  "department" : "IT"
}
{
  "_id" : ObjectId("615165be00c80efa74b09208"),
  "roll_no" : 5,
  "name" : "ajay",
  "department" : "IT"
}
>
```

10

Practical No. 6:

Aim: Write a MongoDB query to use sum, avg, min and max expressions.

Output:

```
Select C:\Windows\System32\cmd.exe - mongo
db.student.find({"name":/([A|B])/i})
{"_id": ObjectId("6151655700c80efa74b09204"), "roll_no": 1, "name": "sankit", "department": "IT" }
{"_id": ObjectId("6151657100c80efa74b09205"), "roll_no": 2, "name": "sachin", "department": "CS" }
{"_id": ObjectId("6151657d00c80efa74b09206"), "roll_no": 2, "name": "amit", "department": "CS" }
{"_id": ObjectId("615165ae00c80efa74b09207"), "roll_no": 4, "name": "avinash", "department": "IT" }
{"_id": ObjectId("615165be00c80efa74b09208"), "roll_no": 5, "name": "ajay", "department": "IT" }
{"_id": ObjectId("61516b400c80efa74b09209"), "name": "sankit", "department": "IT", "age": 21 }
{"_id": ObjectId("61516b5300c80efa74b0920a"), "name": "sachin", "department": "IT", "age": 24 }
{"_id": ObjectId("61516b5e00c80efa74b0920b"), "name": "amit", "department": "IT", "age": 23 }
{"_id": ObjectId("61516b6f00c80efa74b0920c"), "name": "viabhav", "department": "IT", "age": 25 }
{"_id": ObjectId("61516b8100c80efa74b0920d"), "name": "akash", "department": "IT", "age": 20 }
```

11

Practical No. 7:

Aim: Write a MongoDB query to use first and last expressions.

Output:

```
C:\Windows\System32\cmd.exe - mongo
{"_id": ObjectId("61516b8100c80efa74b0920d"), "name": "akash", "department": "IT", "age": 20 }
db.student.find({"age":{$lt:23}})
{"_id": ObjectId("61516b400c80efa74b09209"), "name": "sankit", "department": "IT", "age": 21 }
{"_id": ObjectId("61516b8100c80efa74b0920d"), "name": "akash", "department": "IT", "age": 20 }
db.student.find({"age":{$lte:23}})
{"_id": ObjectId("61516b400c80efa74b09209"), "name": "sankit", "department": "IT", "age": 21 }
{"_id": ObjectId("61516b5e00c80efa74b0920b"), "name": "amit", "department": "IT", "age": 23 }
{"_id": ObjectId("61516b8100c80efa74b0920d"), "name": "akash", "department": "IT", "age": 20 }
db.student.find({"age":{$gt:23}})
{"_id": ObjectId("61516b5300c80efa74b0920a"), "name": "sachin", "department": "IT", "age": 24 }
{"_id": ObjectId("61516b6f00c80efa74b0920c"), "name": "viabhav", "department": "IT", "age": 25 }
db.student.find({"age":{$gte:23}})
{"_id": ObjectId("61516b5300c80efa74b0920a"), "name": "sachin", "department": "IT", "age": 24 }
{"_id": ObjectId("61516b5e00c80efa74b0920b"), "name": "amit", "department": "IT", "age": 23 }
{"_id": ObjectId("61516b6f00c80efa74b0920c"), "name": "viabhav", "department": "IT", "age": 25 }
db.student.find({"name":{$in:["sankit","sachin"]}})
{"_id": ObjectId("6151655700c80efa74b09204"), "roll_no": 1, "name": "sankit", "department": "IT" }
{"_id": ObjectId("6151657100c80efa74b09205"), "roll_no": 2, "name": "sachin", "department": "CS" }
{"_id": ObjectId("61516b400c80efa74b09209"), "name": "sankit", "department": "IT", "age": 21 }
{"_id": ObjectId("61516b5300c80efa74b0920a"), "name": "sachin", "department": "IT", "age": 24 }
```

Practical No. 8:

Aim: Write a MongoDB query to use indexes.

Output:

