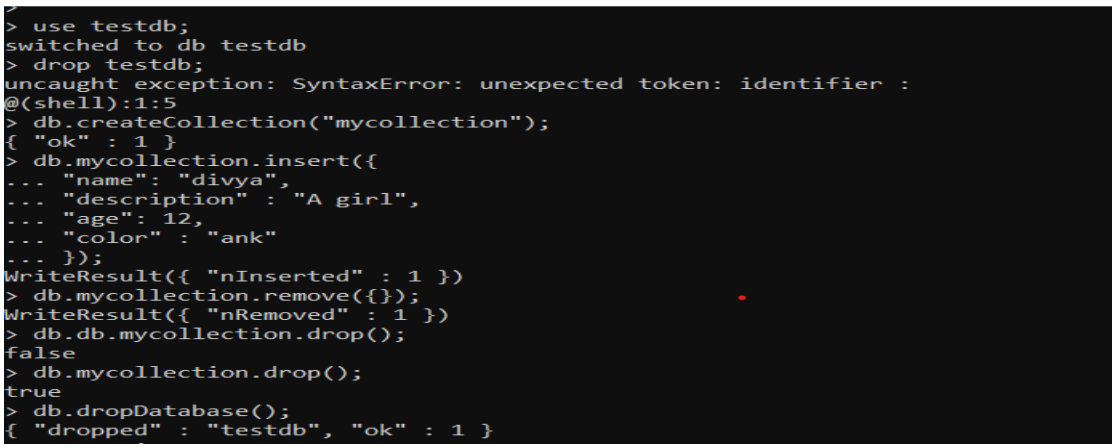
MONGODB:

1. **DROPPING COLLECTIONS AND DBS** : Find out how to delete all documents of a collection, drop a collection, and drop a database. Answer: Explore drop() method of collection, and dropDatabase() of database.



2.DATA TYPES: Create a moviesDB with a movies collection. The movie lists out year of release (int32), box office collection (this has to be stores as a long int as it can exceed maximum value of int32), a release date (a date object), and a timestamp when document as added.
Use db.stats() to check average document size. Change data types for some value using an update query. See how it affects the size.

```
> try{
... db.products.insertMany([
... {_id:1, x:1},
... {_id:2, x:2}], {ordered:false});
... }catch(e){
... print(e);
... }
{ "acknowledged" : true, "insertedIds" : [ 1, 2 ] }
> try{
... db.products.insertMany([
... {_id:3, x=3},
... {_id:4, x=4},
... {_id:5, x=5}],{ordered:true});
... }catch(e){
... print(e);
... }
uncaught exception: SyntaxError: missing : after property id :
0(shell):3:9
> try{ db.products.insertMany([ {_id:3, x:3}, {_id:4, x:4}, {_id:5, x:5}],{ordered:true}); }catch(e){ print(e); }
{ "acknowledged" : true, "insertedIds" : [ 3, 4, 5 ] }
> db.products().find();
uncaught exception: TypeError: db.products is not a function :
0(shell):1:1
> db.products.find();
{ "_id" : 1, "x" : 1 }
{ "_id" : 2, "x" : 2 }
{ "_id" : 3, "x" : 3 }
{ "_id" : 4, "x" : 4 }
{ "_id" : 5, "x" : 5 }
```

3.CRUD OPERATIONS: Create a moviesDB with an albums collection. The albums lists out year of release, music director, lyricist and a list of songs. Every song has details of singers (names), length of song (string).

a) Insert 3 documents with at least 1 song entry each

b) Update album data for an album. Change the name of the lyricist, and also an entirely replaced history entry.

c) Find all albums released before 2019

Remove all documents that have a particular music director / lyricist as value

```
> use musicDB;
switched to db musicDB
> db.createCollection("albums");
{ "ok" : 1 }
> db.albums.insert({
... "yearOfrelease":1200,
... "director": "divya",
... "lyricist":"abcd",
... "songs": [{
... "singername":"xyzw", "lines":100}]
... });
WriteResult({ "nInserted" : 1 })
> db.albums.insert({
... "yearOfrelease":1300,
... "director":"pqrs",
... "lyricist":"abcd",
... "songs":[{
... "singername":"divya", "lines":100}]
... });
WriteResult({ "nInserted" : 1 })
> db.albums.insert({
... "yearOfrelease":1400,
... "director":"divya",
... "lyricist":"qwer",
... "songs":[{
... "singername": "qwer", "lines":145}]
... });
WriteResult({ "nInserted" : 1 })
> db.albums.updateMany(
... {"lyricist":"abcd"},
... {$set:{"lyricist":"divyanagare"}}
... );
{ "acknowledged" : true, "matchedCount" : 2, "modifiedCount" : 2 }
> db.albums.find();
{ "_id" : ObjectId("5f4fd022fd16719e97546b88"), "yearOfrelease" : 1200, "director" : "divya", "lyricist" : "divyanagare", "songs" : [ { "singername" : "xyzw", "lines" : 100 } ] }
{ "_id" : ObjectId("5f4fd090fd16719e97546b89"), "yearOfrelease" : 1300, "director" : "pqrs", "lyricist" : "divyanagare", "songs" : [ { "singername" : "divya", "lines" : 100 } ] }
{ "_id" : ObjectId("5f4fd0e0fd16719e97546b8a"), "yearOfrelease" : 1400, "director" : "divya", "lyricist" : "qwer", "songs" : [ { "singername" : "qwer", "lines" : 145 } ] }
> db.albums.find({"yearOfrelease":{$get:2019}});
Error: error: {
        "ok" : 0,
        "errmsg" : "unknown operator: $get",
        "code" : 2,
        "codeName" : "BadValue"
}
> db.albums.find({"yearOfrelease":{$gt:2019}});
> db.albums.find({"yearOfrelease":{$gt:2019}})
> db.albums.find({"yearOfrelease":{$lt:2019}});
```

```
        "code" : 2,
        "codeName" : "BadValue"
}
> db.albums.find({"yearOfrelease":{$gt:2019}});
> db.albums.find({"yearOfrelease":{$gt:2019}})
> db.albums.find({"yearOfrelease":{$lt:2019}});
{ "_id" : ObjectId("5f4fd022fd16719e97546b88"), "yearOfrelease" : 1200, "director" : "divya", "lyricist" : "divyanagare", "songs" : [ { "singername" : "xyzw", "lines" : 100 } ] }
{ "_id" : ObjectId("5f4fd090fd16719e97546b89"), "yearOfrelease" : 1300, "director" : "pqrs", "lyricist" : "divyanagare", "songs" : [ { "singername" : "divya", "lines" : 100 } ] }
{ "_id" : ObjectId("5f4fd0e0fd16719e97546b8a"), "yearOfrelease" : 1400, "director" : "divya", "lyricist" : "qwer", "songs" : [ { "singername" : "qwer", "lines" : 145 } ] }
> db.albums.remove({ "director":"pqrs"});
WriteResult({ "nRemoved" : 1 })
> db.albums.find();
{ "_id" : ObjectId("5f4fd022fd16719e97546b88"), "yearOfrelease" : 1200, "director" : "divya", "lyricist" : "divyanagare", "songs" : [ { "singername" : "xyzw", "lines" : 100 } ] }
{ "_id" : ObjectId("5f4fd0e0fd16719e97546b8a"), "yearOfrelease" : 1400, "director" : "divya", "lyricist" : "qwer", "songs" : [ { "singername" : "qwer", "lines" : 145 } ] }
>
```