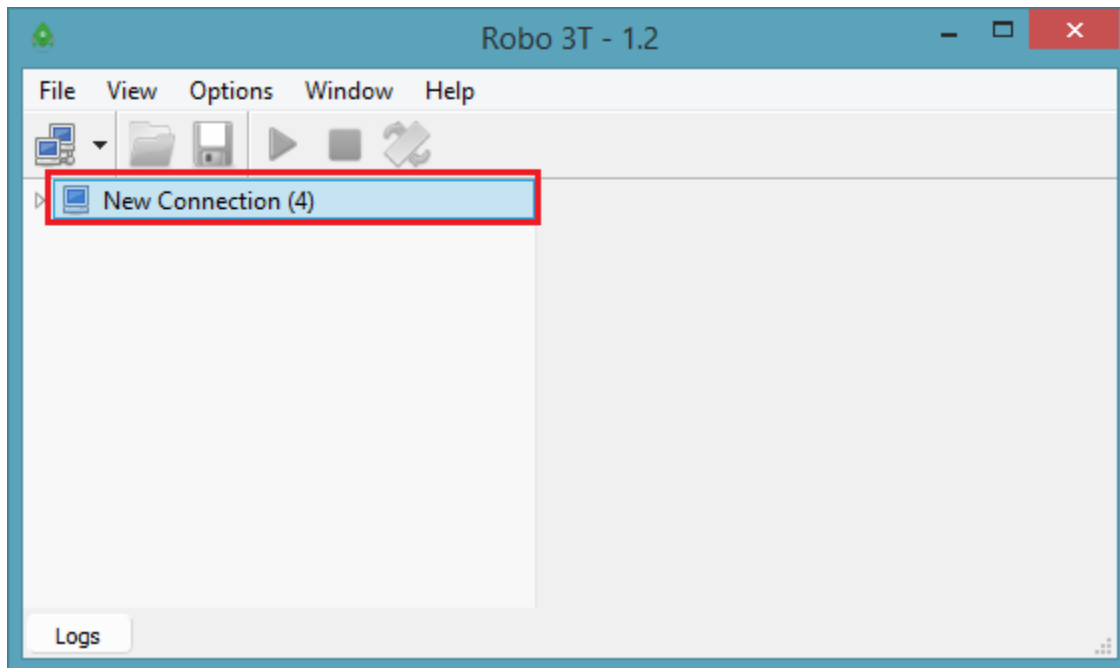


MongoDB

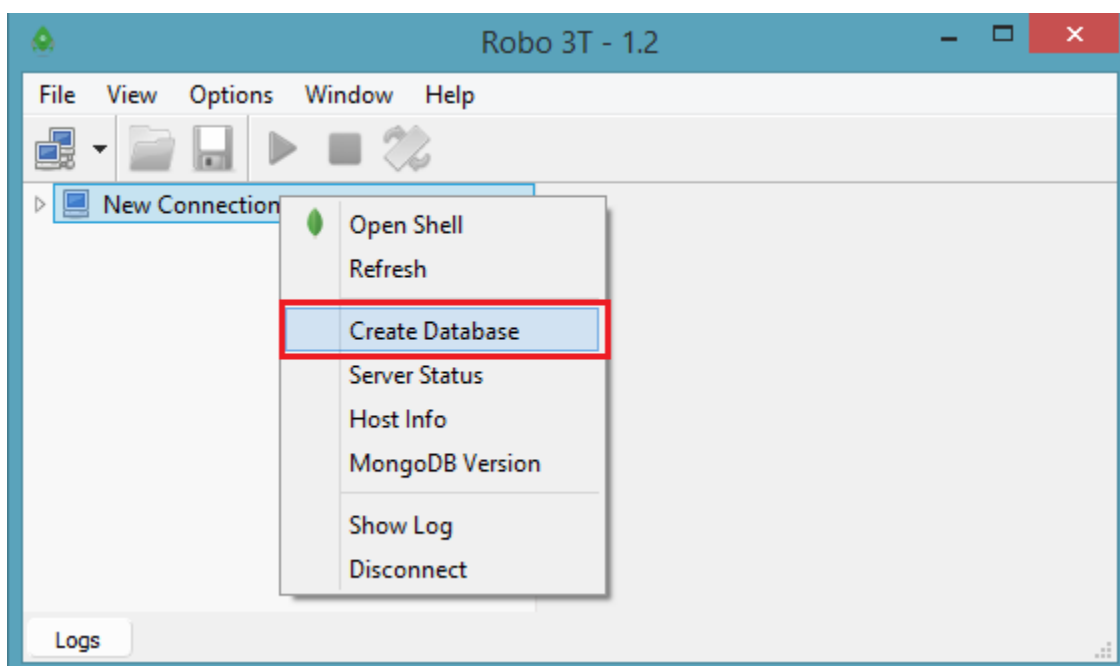
Steps to create MongoDB database:

The Robo 3T tool is used to setup MongoDB, download Robo 3T from <https://robomongo.org/>.

i] Right click on New Connection



ii] Click on Create Database



iii] Enter the Database Name MongoDB and click on Create

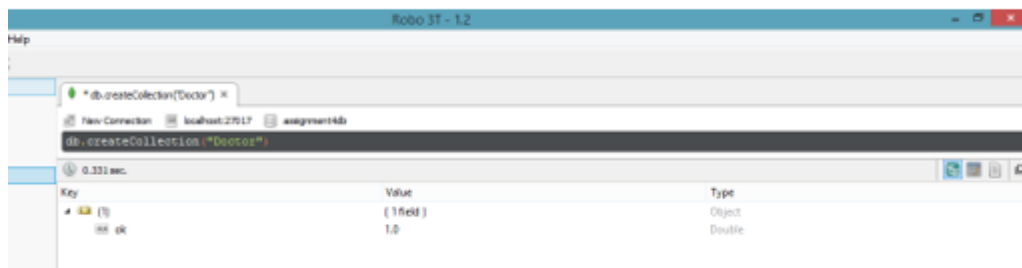
iv] Finally database MongoDB is created.

Collection create code and
screenshot:

Code:

```
db.createCollection("Doctor ")
```

Screenshot:



Populate Collection with data:

Code:

```
db.Doctor.insertMany([
  { _id: 1, name: "Mark Steven", speciality: "Nephrologist", sex: "Male", phonenumber: "9192345678", },
  { _id: 2, name: "Jane Philip", speciality: "Cadiologist", sex: "Female", phonenumber: "919098567", },
  { _id: 3, name: "Ronit Ray", speciality: "Dermatologist", sex: "Male", phonenumber: "9195789023", },
  { _id: 4, name: "Jude Thomas", speciality: "Oncologist", sex: "Male", phonenumber: "9197864356", },
  { _id: 5, name: "Joan Thomas", speciality: "Neurologist", sex: "Female", phonenumber: "9199160428", }
])
```

Screenshot:

The screenshot shows the Robo 3T - 1.2 application window. The top toolbar includes a 'New Connection' button, the current connection 'localhost:27017', and the database 'assignment4db'. The main editor displays the following JavaScript code:

```
db.Doctor.insertMany([
  { _id: 1, name: "Mark Steven", speciality: "Nephrologist", sex: "Male", phonenumber: "9192345678", },
  { _id: 2, name: "Jane Philip", speciality: "Cadiologist", sex: "Female", phonenumber: "919098567", },
  { _id: 3, name: "Ronit Ray", speciality: "Dermatologist", sex: "Male", phonenumber: "9195789023", },
  { _id: 4, name: "Jude Thomas", speciality: "Oncologist", sex: "Male", phonenumber: "9197864356", },
  { _id: 5, name: "Joan Thomas", speciality: "Neurologist", sex: "Female", phonenumber: "9199160428", }
])
```

Below the editor, the execution results are shown. The execution time is 0.005 sec. The results table has three columns: Key, Value, and Type.

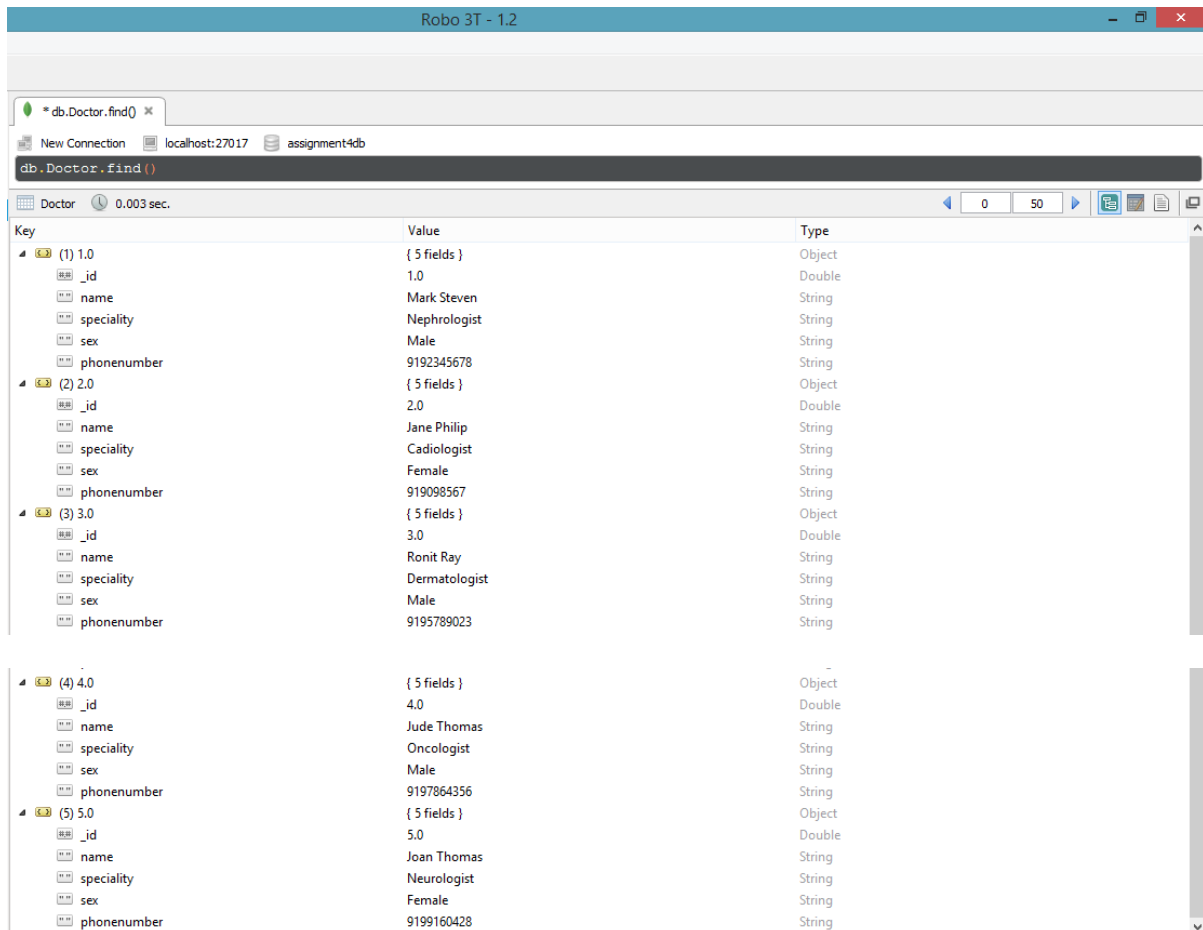
Key	Value	Type
(1)	{ 2 fields }	Object
acknowledged	true	Boolean
insertedIds	[5 elements]	Array
[0]	1.0	Double
[1]	2.0	Double
[2]	3.0	Double
[3]	4.0	Double
[4]	5.0	Double

Queries

1]

```
db.Doctor.find()
```

Screenshot:



The screenshot shows the Robo 3T - 1.2 application window. The top bar indicates the connection to localhost:27017, database assignment4db. The query db.Doctor.find() is entered in the command field. The results are displayed in a table with 5 rows, each representing a doctor record. The table has three main columns: Key, Value, and Type. The Key column shows the document ID (e.g., (1) 1.0). The Value column shows the document fields (e.g., { 5 fields }). The Type column shows the data type (e.g., Object, Double, String).

Key	Value	Type
(1) 1.0	{ 5 fields }	Object
_id	1.0	Double
name	Mark Steven	String
speciality	Nephrologist	String
sex	Male	String
phonenumber	9192345678	String
(2) 2.0	{ 5 fields }	Object
_id	2.0	Double
name	Jane Philip	String
speciality	Cadiologist	String
sex	Female	String
phonenumber	919098567	String
(3) 3.0	{ 5 fields }	Object
_id	3.0	Double
name	Ronit Ray	String
speciality	Dermatologist	String
sex	Male	String
phonenumber	9195789023	String
(4) 4.0	{ 5 fields }	Object
_id	4.0	Double
name	Jude Thomas	String
speciality	Oncologist	String
sex	Male	String
phonenumber	9197864356	String
(5) 5.0	{ 5 fields }	Object
_id	5.0	Double
name	Joan Thomas	String
speciality	Neurologist	String
sex	Female	String
phonenumber	9199160428	String

2]

```
db.Doctor.find({sex:"Female"})
```

) Screenshot:

Robo 3T - 1.2

* db.Doctor.find({sex:'Female'})

New Connection localhost:27017 assignment4db

db.Doctor.find({sex:"Female"})

Doctor 0.004 sec.

Key	Value	Type
(1) 2.0	{ 5 fields }	Object
_id	2.0	Double
name	Jane Philip	String
speciality	Cadiologist	String
sex	Female	String
phonenumber	919098567	String
(2) 5.0	{ 5 fields }	Object
_id	5.0	Double
name	Joan Thomas	String
speciality	Neurologist	String
sex	Female	String
phonenumber	9199160428	String

3]

db.Doctor.find().count()

Screenshot:

Robo 3T - 1.2

* db.Doctor.find().count()

New Connection localhost:27017 assignment4db

db.Doctor.find().count()

0.002 sec.

5

4]

db.Doctor.find().sort({"name":1})

) Screenshot:

Robo 3T - 1.2

* db.Doctor.find().sort({"name":1})

New Connection localhost:27017 assignment4db

db.Doctor.find().sort({"name":1})

Doctor 0.003 sec.

Key	Value	Type
(1) 2.0	{ 5 fields }	Object
_id	2.0	Double
name	Jane Philip	String
speciality	Cadiologist	String
sex	Female	String
phonenumber	919098567.0	Double
(2) 5.0	{ 5 fields }	Object
_id	5.0	Double
name	Joan Thomas	String
speciality	Neurologist	String
sex	Female	String
phonenumber	9199160428.0	Double
(3) 4.0	{ 5 fields }	Object
_id	4.0	Double
name	Jude Thomas	String
speciality	Oncologist	String
sex	Male	String
phonenumber	9197864356.0	Double

4	1.0	{ 5 fields }	Object
	_id	1.0	Double
	name	Mark Steven	String
	speciality	Nephrologist	String
	sex	Male	String
	phonenumber	9192345678.0	Double
5	3.0	{ 5 fields }	Object
	_id	3.0	Double
	name	Ronit Ray	String
	speciality	Dermatologist	String
	sex	Male	String
	phonenumber	9195789023.0	Double

5] db.Doctor.find({ sex: "Male" }, { name: 1, speciality: 1 }) Screenshot:

Robo 3T - 1.2

Query: `db.Doctor.find({ sex: "Male" }, { name: 1, speciality: 1 })`

Results:

Key	Value	Type
(1) 1.0	{ 3 fields }	Object
_id	1.0	Double
name	Mark Steven	String
speciality	Nephrologist	String
(2) 3.0	{ 3 fields }	Object
_id	3.0	Double
name	Ronit Ray	String
speciality	Dermatologist	String
(3) 4.0	{ 3 fields }	Object
_id	4.0	Double
name	Jude Thomas	String
speciality	Oncologist	String

6] db.Doctor.update({"name": "Jane Philip"}, {\$set: {"phonenumber": 9753028841}}) Screenshot:

Robo 3T - 1.2

Query: `db.Doctor.update({"name": "Jane Philip"}, {$set: {"phonenumber": 9753028841}})`

Execution Time: 0.124 sec.

Status: Updated 1 existing record(s) in 103ms

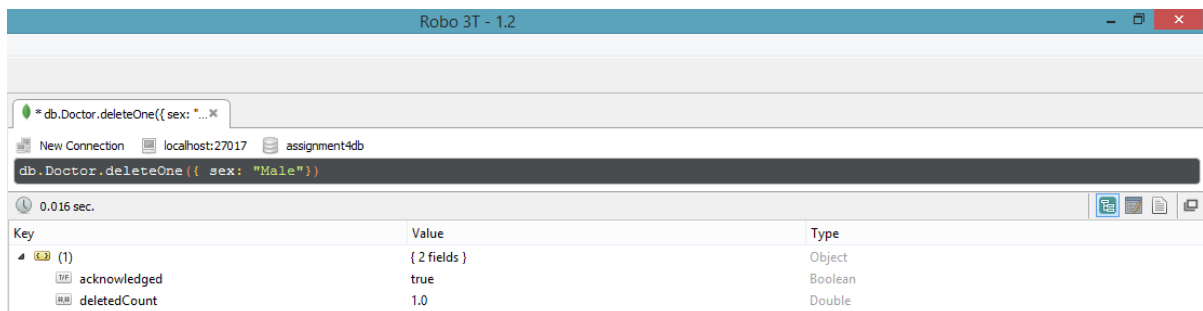
7] db.Doctor.aggregate([
 { \$match: { sex: "Male" } },
 { \$group: { _id: "\$speciality" } }
])

Screenshot:



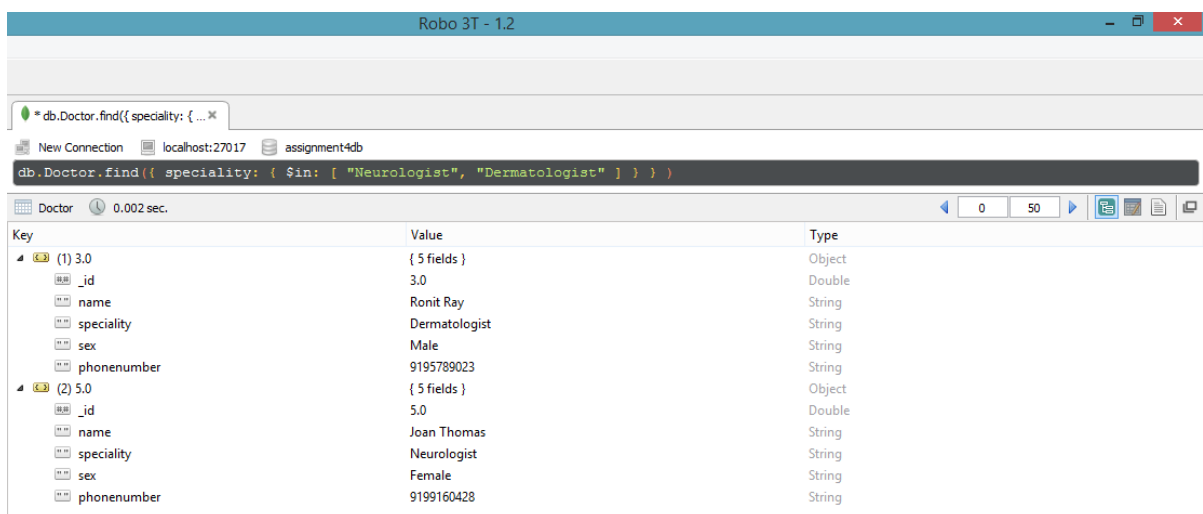
8] db.Doctor.deleteOne({ sex:

"Male"}) Screenshot:



9] db.Doctor.find({ speciality: { \$in: ["Neurologist",

"Dermatologist"] } }) Screenshot:



10]

```

db.Doctor.replaceOne(
  { name: "Joan Thomas" },
  { name: "Roy Mark", speciality: "Neurologist", sex: "Male", phonenummer: "9198448400" }
)

```

Screenshot:

The screenshot shows the Robo 3T - 1.2 application window. At the top, the title bar reads "Robo 3T - 1.2". Below the title bar, there is a toolbar with icons for New Connection, localhost:27017, and assignment4db. The main area displays a MongoDB command: `db.Doctor.replaceOne({name: "Joan Thomas"}, {name: "Roy Mark", speciality: "Neurologist", sex: "Male", phonenumber: "9198448400"})`. Below the command, the execution time is shown as "0.059 sec.". The result is displayed in a table with three columns: Key, Value, and Type.

Key	Value	Type
acknowledged	true	Boolean
matchedCount	1.0	Double
modifiedCount	1.0	Double