

LAB-2

Robo 3T - 1.4

File View Options Window Help

The screenshot shows the Robo 3T interface. On the left, a tree view displays the database structure: 'rishika (5)' contains 'System', 'config', 'employeeDb', and 'test'. 'employeeDb' contains two collections: 'Department' and 'Employee'. The 'Employee' collection is selected. On the right, a console window shows a MongoDB query: `* db.Employee.find({"age..."`. Below the console, a tab shows the 'employeeDb' database with a JavaScript console containing the following code:

```
db.Employee.insert({
  "_id" : 1,
  "empName" : "Harsh",
  "Addr" : "Bangalore",
  "projects" : [
    {
      "projName" : "abc",
      "dur" : "48hrs"
    }
  ]
})
```

Robo 3T - 1.4

File View Options Window Help

The screenshot shows the Robo 3T interface. On the left, a tree view displays the database structure: 'rishika (5)' contains 'System', 'config', 'employeeDb', and 'test'. 'employeeDb' contains two collections: 'Department' and 'Employee'. The 'Employee' collection is selected. On the right, a console window shows a MongoDB query: `* db.Employee.find({"age..."`. Below the console, a tab shows the 'employeeDb' database with a JavaScript console containing the following code:

```
db.Employee.insertOne(
  {
    "empName" : "veena",
    "lastname": "reddy",
    "Addr" : "ooty",
    "projects" : [ "efg", "38hrs" ]
  },
  {
    "acknowledged": true,
    "insertedId": ObjectId("56fc40f9d735c28df206d078")
  }
)
```

Robo 3T - 1.4

File View Options Window Help

The screenshot shows the Robo 3T interface with a MongoDB database connection named 'employeeDb' at 'localhost:27017'. The left sidebar shows the database structure, including a collection named 'Employee'. The main window displays the command `db.Employee.insertMany()` and the resulting data for the 'Employee' collection. The data is an array of three objects, each representing an employee with fields: empName, lastname, Date_Of_Birth, and phone.

```
db.Employee.insertMany([
  {
    "empName": "jessy",
    "lastname": "Sharma",
    "Date_Of_Birth": "1995-04-06",
    "phone": "9000012345"
  },
  {
    "empName": "jay",
    "lastname": "kar",
    "Date_Of_Birth": "1990-02-16",
    "phone": "9000054321"
  },
  {
    "empName": "Fathima",
    "lastname": "Sheik",
    "Date_Of_Birth": "1980-05-18",
    "phone": "9000054321"
  }
])
```

Employee 0.001 sec.

Robo 3T - 1.4

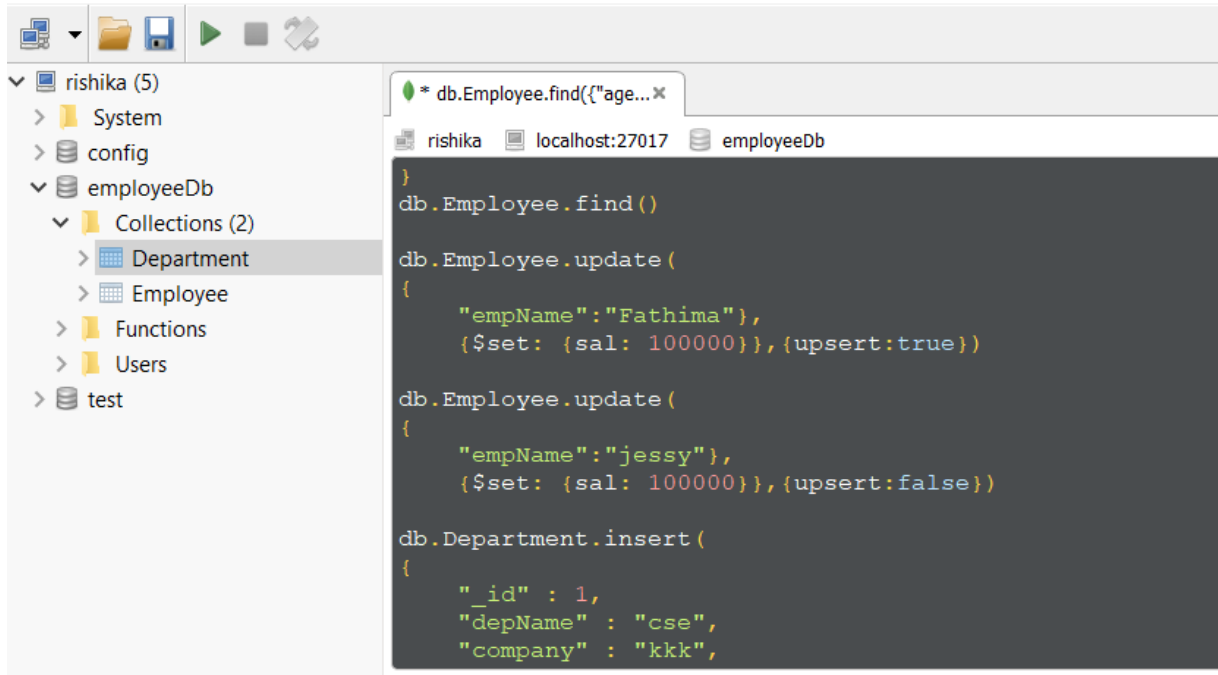
File View Options Window Help

The screenshot shows the Robo 3T interface with a MongoDB database connection named 'employeeDb' at 'localhost:27017'. The left sidebar shows the database structure, including a collection named 'Employee'. The main window displays the command `db.Employee.find()` and the resulting data for the 'Employee' collection. The data is an array of three objects, each representing an employee with fields: empName, lastname, Date_Of_Birth, and phone. The first two objects are the same as in the previous screenshot, but the third object has a different Date_Of_Birth and phone number.

```
db.Employee.find()
[
  {
    "empName": "Fathima",
    "lastname": "Sheik",
    "Date_Of_Birth": "1980-05-18",
    "phone": "9000054321"
  },
  {
    "empName": "jessy",
    "lastname": "Sharma",
    "Date_Of_Birth": "1995-04-06",
    "phone": "9000012345"
  },
  {
    "empName": "jay",
    "lastname": "kar",
    "Date_Of_Birth": "1990-02-16",
    "phone": "9000054321"
  }
]
```

Robo 3T - 1.4

File View Options Window Help

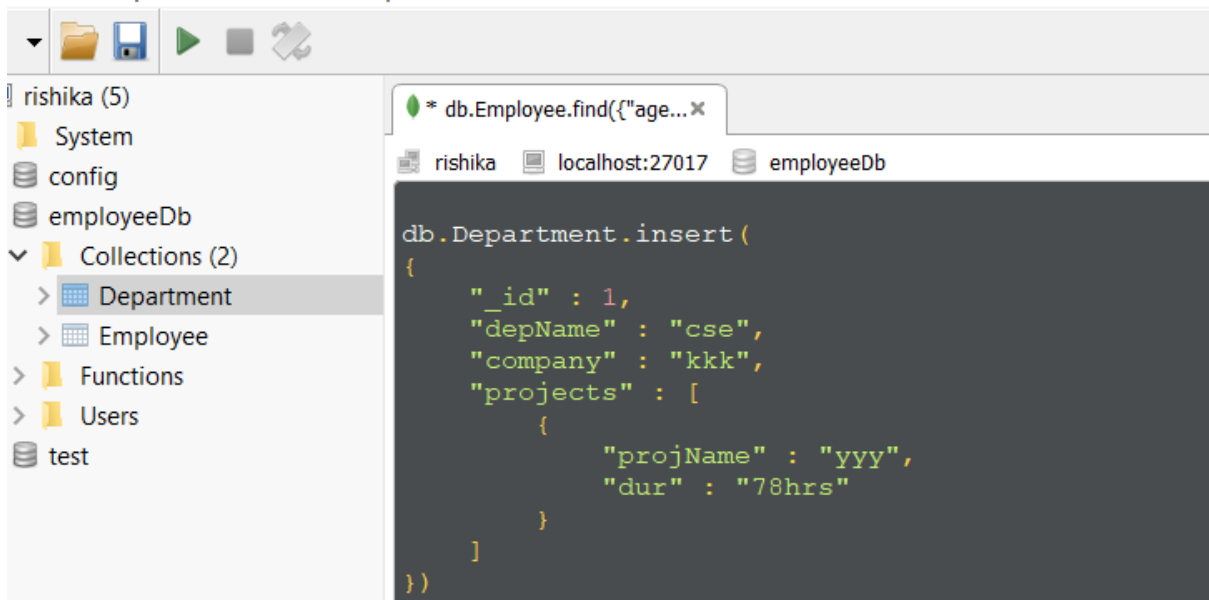


The screenshot shows the Robo 3T interface. On the left, the 'Collections (2)' list is expanded, showing 'Department' and 'Employee'. The main console on the right contains the following JavaScript code:

```
* db.Employee.find({"age...x
rishika localhost:27017 employeeDb
}
db.Employee.find()
db.Employee.update(
{
  "empName": "Fathima",
  {$set: {sal: 100000}}, {upsert: true})
db.Employee.update(
{
  "empName": "jessy",
  {$set: {sal: 100000}}, {upsert: false})
db.Department.insert(
{
  "_id" : 1,
  "depName" : "cse",
  "company" : "kkk",
```

Robo 3T - 1.4

View Options Window Help



The screenshot shows the Robo 3T interface. On the left, the 'Collections (2)' list is expanded, showing 'Department' and 'Employee'. The main console on the right contains the following JavaScript code:

```
* db.Employee.find({"age...x
rishika localhost:27017 employeeDb
db.Department.insert(
{
  "_id" : 1,
  "depName" : "cse",
  "company" : "kkk",
  "projects" : [
    {
      "projName" : "yyy",
      "dur" : "78hrs"
    }
  ]
})
```

Robo 3T - 1.4

File View Options Window Help

The screenshot shows the Robo 3T interface with a MongoDB database connection named 'employeeDb' at 'localhost:27017'. The left sidebar shows a tree view of the database structure, including 'System', 'config', 'employeeDb', 'Collections (2)', 'Department', 'Employee', 'Functions', 'Users', and 'test'. The 'Department' collection is selected. The main editor displays a JavaScript code snippet for inserting a document into the 'Department' collection.

```
* db.Employee.find({"age... ×  
rshika localhost:27017 employeeDb  
})  
db.Department.insertOne(  
  {"depName" : "RnD",  
   "company" : "rrr",  
   "projects" : [ "efg", "38hrs"]},  
  {  
    "acknowledged": true,  
    "insertedId": ObjectId("56fc40f9d735c28df206d078")  
  }  
)
```

Robo 3T - 1.4

File View Options Window Help

The screenshot shows the Robo 3T interface with a MongoDB database connection named 'employeeDb' at 'localhost:27017'. The left sidebar shows a tree view of the database structure, including 'System', 'config', 'employeeDb', 'Collections (2)', 'Department', 'Employee', 'Functions', 'Users', and 'test'. The 'Department' collection is selected. The main editor displays a JavaScript code snippet for inserting multiple documents into the 'Department' collection.

```
* db.Employee.find({"age... ×  
rshika localhost:27017 employeeDb  
)  
db.Department.insertMany(  
  [  
    {  
      "depName" : "mech",  
      "company" : "ppp",  
      "projects" : [ "qq", "38hrs"],  
      "phone": "9000012345"  
    },  
    {  
      "depName" : "eee",  
      "company" : "jjj",  
      "projects" : [ "aa", "38hrs"],  
      "phone": "9000012349"  
    },  
    {  
      "depName" : "IS",  
      "company" : "lll",  
    }  
  ]  
)
```

Robo 3T - 1.4

File View Options Window Help

The screenshot shows the Robo 3T interface with a sidebar on the left displaying a tree view of the database structure. The tree view includes 'rishika (5)' with sub-items: 'System', 'config', 'employeeDb' (expanded), 'Collections (2)' (expanded), 'Department', 'Employee', 'Functions', 'Users', and 'test'. The main editor area on the right has a tab titled '* db.Employee.find({"age...'. Below the tab, there are three buttons: 'rishika', 'localhost:27017', and 'employeeDb'. The code editor contains the following JavaScript code:

```
],
},
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("5dd631f270fb13eec3963bed"),
    ObjectId("5dd631f270fb13eec3963bee"),
    ObjectId("5dd631f270fb13eec3963bef") ]
}
db.Department.find()

db.Department.update (
{
  "depname": "IS",
  {$set: {sal: 200000}}, {upsert: true})

db.Department.update (
```

Robo 3T - 1.4

File View Options Window Help

The screenshot shows the Robo 3T interface with a sidebar on the left displaying a tree view of the database structure. The tree view includes 'rishika (5)' with sub-items: 'System', 'config', 'employeeDb' (expanded), 'Collections (2)' (expanded), 'Department', 'Employee', 'Functions', 'Users', and 'test'. The main editor area on the right has a tab titled '* db.Employee.find({"age...'. Below the tab, there are three buttons: 'rishika', 'localhost:27017', and 'employeeDb'. The code editor contains the following JavaScript code:

```
{
  "depname": "IS",
  {$set: {sal: 200000}}, {upsert: true})

db.Department.update (
{
  "depname": "mech",
  {$set: {sal: 100000}}, {upsert: false})

db.Employee.update ( { "empname": "jessy"}, {$set: {"age": 34}}, {upsert: true})

db.Employee.update ({ "empname": "Harsh"}, {$unset: {"Addr": ""}}, {upsert: true})
db.Employee.find({})
db.Department.find({})
db.Employee.find ({ "empname": "Harsh"})
db.Employee.find ({ "empname": /^H/ })
db.Employee.find ({ "age": {$gt: 20} })
```

Robo 3T - 1.4

File View Options Window Help

risika (5)

- System
- config
- employeeDb
 - Collections (2)
 - Department
 - Employee
 - Functions
 - Users
 - test

* db.Employee.find() x

```
risika localhost:27017 employeeDb
{
  "acknowledged": true,
  "insertedIds": [
    ObjectId("5d4631f270fb13eec3963bed"),
    ObjectId("5d4631f270fb13eec3963bee"),
    ObjectId("5d4631f270fb13eec3963bef")
  ]
}
db.Employee.find()
db.Employee.update(
  {
    "empName": "Pathima",
    ($set: {sal: 100000}), (upsert: true)
  })
db.Employee.update(
  {
    "empName": "jessy",
    ($set: {sal: 100000}), (upsert: false)
  })
```

Employee @ 0.001 sec.

Key	Value	Type
> (1) 1	(4 fields)	Object
> (2) ObjectId("5f75acca4a9f30f15ed3309b")	(5 fields)	Object
> (3) ObjectId("5f75aeba4a9f30f15ed3309c")	(6 fields)	Object
> (4) ObjectId("5f75aeba4a9f30f15ed3309d")	(5 fields)	Object
> (5) ObjectId("5f75aeba4a9f30f15ed3309e")	(6 fields)	Object
> (6) ObjectId("5f75bbb3b193723708f306e8")	(3 fields)	Object
> (7) ObjectId("5f75bcb4b193723708f30709")	(2 fields)	Object

Robo 3T - 1.4

File View Options Window Help

risika (5)

- System
- config
- employeeDb
 - Collections (2)
 - Department
 - Employee
 - Functions
 - Users
 - test

* db.Department.find() x

```
risika localhost:27017 employeeDb
{
  "acknowledged": true,
  "insertedIds": [
    ObjectId("5d4631f270fb13eec3963bed"),
    ObjectId("5d4631f270fb13eec3963bee"),
    ObjectId("5d4631f270fb13eec3963bef")
  ]
}
db.Department.find()
db.Department.update(
  {
    "deptname": "IS",
    ($set: {sal: 200000}), (upsert: true)
  })
db.Department.update(
  {
    "deptname": "IS",
    ($set: {sal: 200000}), (upsert: true)
  })
```

Department @ 0.001 sec.

Key	Value	Type
> (1) 1.0	(4 fields)	Object
> (2) ObjectId("5f75b9024a9f30f15ed3309f")	(4 fields)	Object
> (3) ObjectId("5f75b9064a9f30f15ed330a0")	(4 fields)	Object
> (4) ObjectId("5f75b9f74a9f30f15ed330a1")	(5 fields)	Object
> (5) ObjectId("5f75b9f74a9f30f15ed330a2")	(5 fields)	Object
> (6) ObjectId("5f75b9f74a9f30f15ed330a3")	(5 fields)	Object
> (7) ObjectId("5f75ba05b193723708f306b0")	(3 fields)	Object

Robo 3T - 1.4

File View Options Window Help

risika (5)

- System
- config
- employeeDb
 - Collections (2)
 - Department
 - Employee
 - Functions
 - Users
 - test

```
db.Employee.find("em... X")
{
  "deptname": "IS",
  ($set: {sal: 200000}), (upsert: true)
}
db.Department.update(
{
  "deptname": "mech",
  ($set: {sal: 100000}), (upsert: false)
}
db.Employee.update( { "empname": "jessy"}, {$set: {"age": 34}}, (upsert: true))
db.Employee.update( {"empname": "Harsh"}, {$unset: {"Addr": ""}}, (upsert: true))
db.Employee.find({})
db.Department.find({})
db.Employee.find({"empname": "Harsh"})
db.Employee.find({"empname": "/R/})
db.Employee.find({"age": {$gt: 20}})
```

Employee 0.001 sec.

Key	Value	Type
(1) ObjectId("5f75bcb4b193723708f30709")	{ 2 fields }	Object
_id	ObjectId("5f75bcb4b193723708f30709")	Objectid
empname	Harsh	String

Robo 3T - 1.4

File View Options Window Help

risika (5)

- System
- config
- employeeDb
 - Collections (2)
 - Department
 - Employee
 - Functions
 - Users
 - test

```
db.Employee.find("em... X")
{
  "deptname": "IS",
  ($set: {sal: 200000}), (upsert: true)
}
db.Department.update(
{
  "deptname": "mech",
  ($set: {sal: 100000}), (upsert: false)
}
db.Employee.update( { "empname": "jessy"}, {$set: {"age": 34}}, (upsert: true))
db.Employee.update( {"empname": "Harsh"}, {$unset: {"Addr": ""}}, (upsert: true))
db.Employee.find({})
db.Department.find({})
db.Employee.find({"empname": "Harsh"})
db.Employee.find({"empname": "/R/})
db.Employee.find({"age": {$gt: 20}})
```

Employee 0.001 sec.

Key	Value	Type
(1) ObjectId("5f75bcb4b193723708f30709")	{ 2 fields }	Object
_id	ObjectId("5f75bcb4b193723708f30709")	Objectid
empname	Harsh	String

Employee 0 sec.

Key	Value	Type
(1) ObjectId("5f75bbb3b193723708f306e8")	{ 3 fields }	Object
_id	ObjectId("5f75bbb3b193723708f306e8")	Objectid
empname	jessy	String
age	34.0	Double