**1)Insert Operation in a Nested Document**

input : db.students.insertMany ([

{id: 3, Name: "Priya", Address: {

Address1: "Vadapalani",

Address2: "Koyembedu",

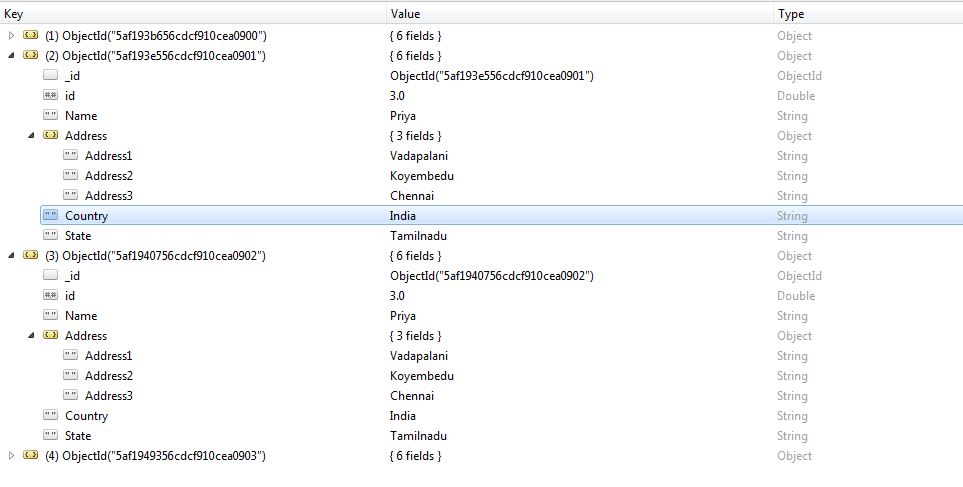
Address3: "Chennai"

},

Country: "India", State: "Tamilnadu"

}

]);

output : 

**2) Update Operation in a Nested Document**

input : db.students.update(

{

"id" :3,

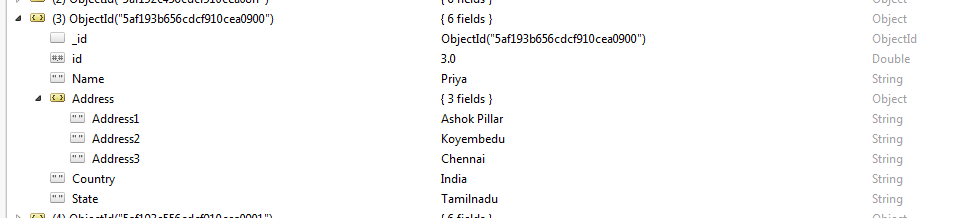
"Address.Address1" : "Vadapalani" },

{

$set : { "Address.Address1" : "Ashok Pillar" }

}

)

output : 

**3) Insert Array of Documents**

input : var sample = [

{

"\_id" : 1,

"name" : "Dave Gahan",

"medications" : [

{ "id" : 23,

"name" : "Dilaudid",

"type" : "Rx",

"prescriptions" : [

{ "id" : 13,

"quantity" : 60,

"started" : 2009-01-01

},

{ "id" : 77,

"quantity" : 45,

"started" : 2009-02-01

}

]

},

{ "id" : 41,

"name" : "Oxycodone",

"type" : "Rx"

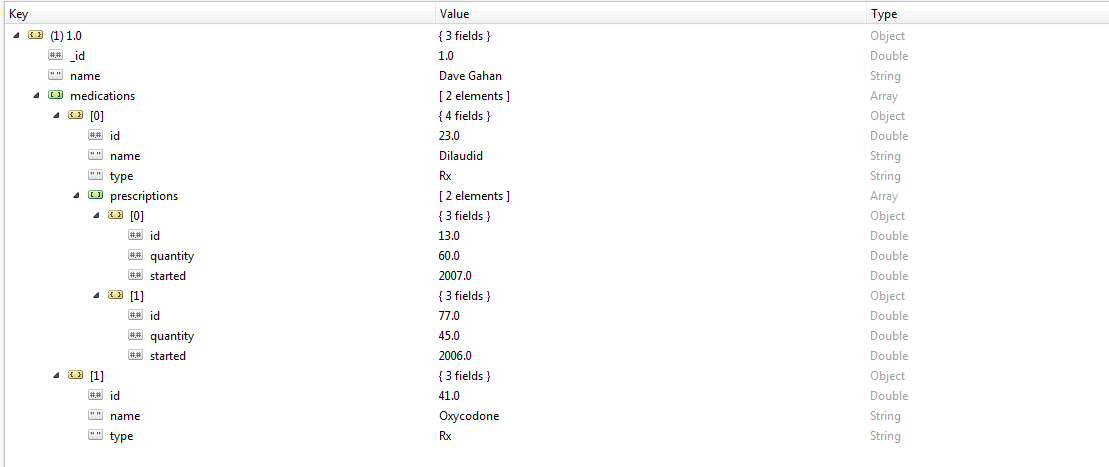
}

]

}

]

db.Medicines.insert(sample)

output: 

**4)Update Array of Documents**

input : db.Medicines.update(

{

"\_id" : 1,

"medications.id" : 23,

"medications.prescriptions.id" : 77 },

{

$set : { "medications.prescriptions.$.quantity" : 30 }

},

)

**5) findAndModify Operation**

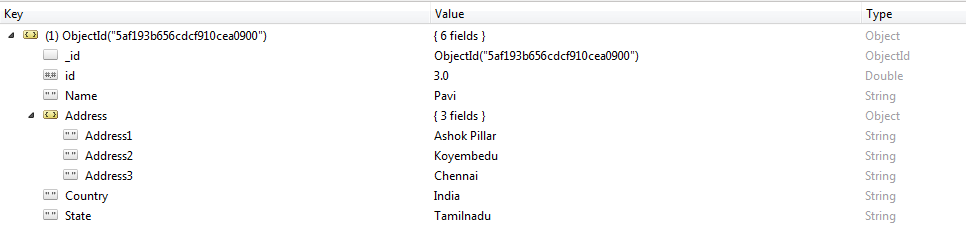
input : db.students.findAndModify({

query : { id : 3},

update : { $set : { Name : "Pavi" } },

upsert : true

})

output : 

Upsert method : Upsert : true option for the update operation to either update a matching document or, if no matching document exists, create a new document.

**6)Adding another row to the Document**

input : db.Medicines.update (

{ \_id : 1 },

{

$push: {

medications : {

$each: [ { id: 24, name: "Priya", "type": "YAZ"}]

}

}

}

)

output : 

**7) Updating a field in the array of the Document**

input : db.Medicines.updateOne(

{

\_id : 1,

"medications.id" : 24

},

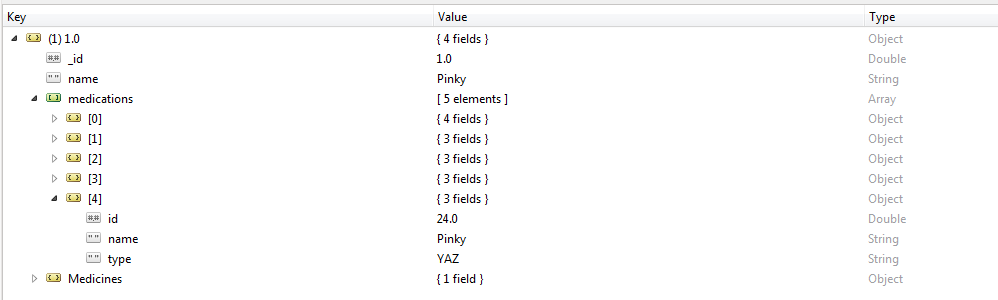
{

$set : { "medications.$.name" : "Pinky"

}

}

)

output: 

**8) Deleting one column from the Document**

input : db.Medicines.update(

{ "medications.prescriptions.id" : [77.0]

},

{

$pull: {"medications.$.prescriptions":

{ id:[77.0] }

}

})

output : 

**9)Adding a New field to a existing Document** input : db.Medicines.update (

{ "medications.prescriptions.id" : [77.0]

},

{

$push : {"medications.prescriptions" : {"isprescribedBy" : ["Doctor"] }

}

})

**10)Joins**

input : db.donor.aggregate([

{ $lookup: {

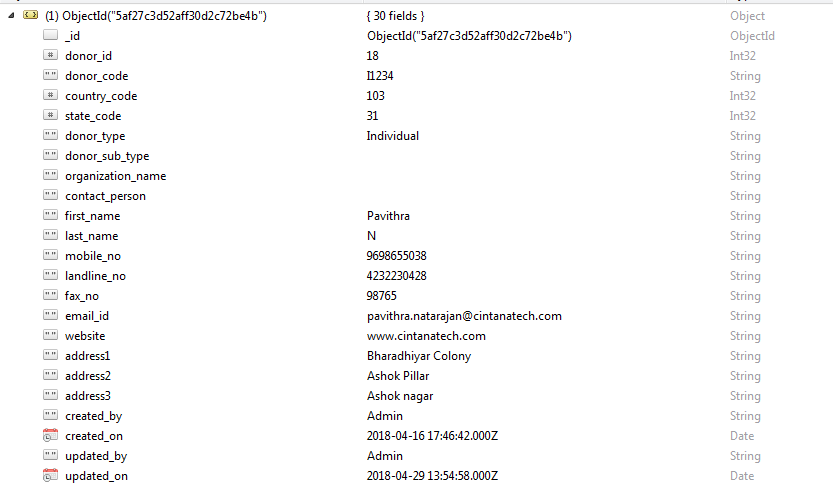
from : "annual\_maintenance",

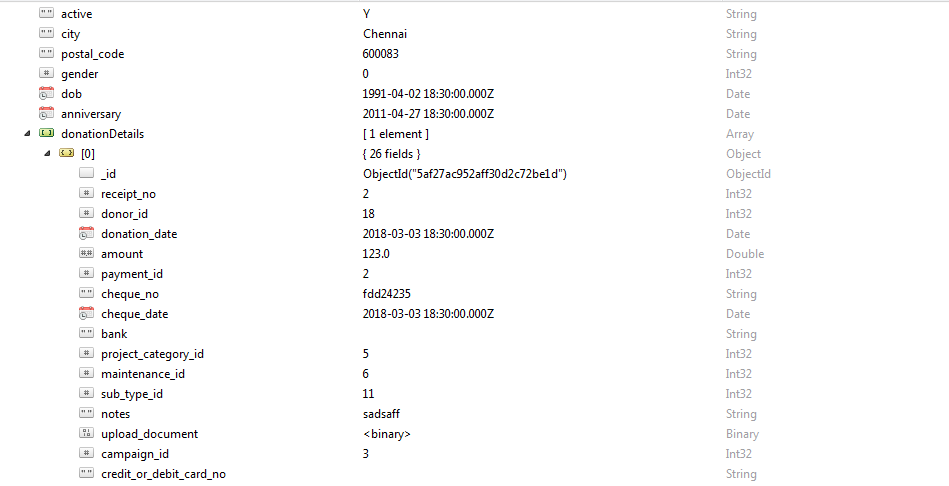
localField : "donor\_id",

foreignField : "donor\_id",

as : "donationDetails" }}

])

output : 

  
**11) Fetching Records between Two Dates**

input : db.annual\_maintenance.find({

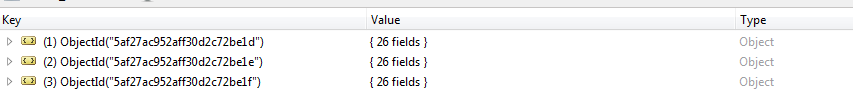
donation\_date : {

$gte: ISODate("2018-03-03"),

$lte: ISODate("2018-06-03")

}

})

output : 

**19) Fetching Records from multiple collections and Matching the Value**

input : db.donor.aggregate([

{ $lookup: {

from: "country\_master",

localField: "country\_code",

foreignField: "country\_code",

as : "countryDetails" }

},

{

$lookup: {

from: "state\_master",

localField: "state\_code",

foreignField: "state\_code",

as : "stateDetails" }

},

{

$lookup: {

from: "annual\_maintenance",

localField: "donor\_id",

foreignField: "donor\_id",

as : "annual" }

},

{

$lookup: {

from: "campaign\_master",

localField: "campaign\_id",

foreignField: "annual.campaign\_id",

as : "campaign" }

},

{ $match: {"donor\_id":31} },{"$project":{

"countryDetails.country":1,

"stateDetails.state":1,

"annual.campaign\_name":1,

"donor\_code":1,

"first\_name":1,

"last\_name":1,

"mobile\_no":1,

"email\_id":1,

"annual.start\_date":1,

"annual.end\_date":1,

}}

])

output : 

**20)Inserting Records in a Nested Array**

var array = [

{

"provider\_id": 1,

"provider\_name": "Ram",

"provider\_category": [

{ "id" : 101,

"category\_name" :"Painting",

"provider\_sub\_category" : [

{ "id": 001,

"sub\_category\_name" : "Interior Painting",

"sub\_categories" :[

{ "name" : "Hall" },

{ "name" : "Kitchen" },

{ "name" : "Bedroom" }

]

},

{ "id" : 002,

"sub\_category\_name" : "Exterior Painting",

"sub\_categories" :[

{ "name" : "Lawn" },

{ "name" : "Visitors\_room" }

]

}

]

},

{

"id" : 102,

"category\_name" :"Electrical and Computers",

"provider\_sub\_category" : [

{ "id": 001,

"sub\_category\_name" : "Computers",

"sub\_categories" :[

{ "name" : "CPU" },

{ "name" : "Motherboard" },

{ "name" : "Battery" }

]

},

{ "id": 002,

"sub\_category\_name" : "Mobile",

"sub\_categories" :[

{ "name" : "Charger" },

{ "name" : "Battery" },

{ "name" : "Headphones" }

]

}

]

}

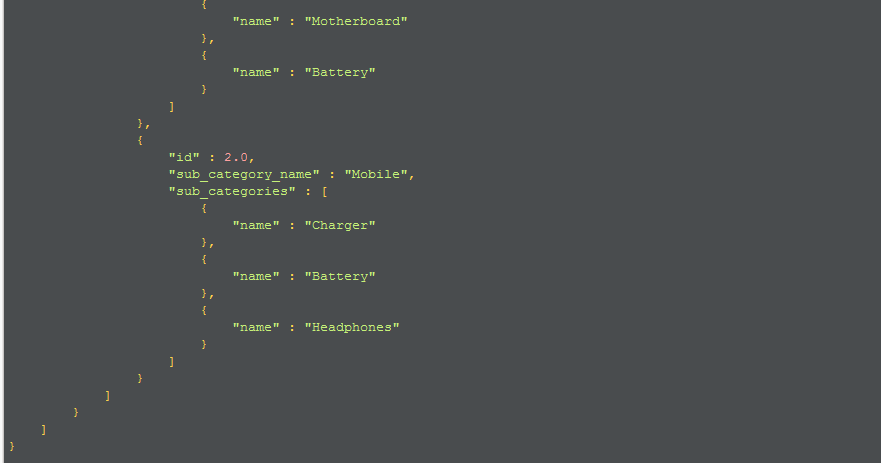
]

}]

db.service\_provider.insert(array)

output : 





**21) Find the Records based on a particular Id.**

input : db. service\_provider.find(

{

"provider\_category" : {

"$elemMatch" : {

"category\_name" : "Painting",

"provider\_sub\_category" : {

"$elemMatch" : {

"sub\_category\_name" : "Interior Painting"

}

}

}

}

})

**22) Update a Record in Nested Document**

input : db.service\_provider.update(

{

'provider\_id' : 1

},

{

'provider\_category.id' : 101

},

{ '$set' :

{ 'provider\_sub\_category.sub\_category\_name' : 'Interior'

}

});