

MongoDB Schema and Relations Report

By Arjun Raj



Schema and relations operations

- One to One relation
- One to Many relation
- Many to Many relation
- Lookup
- Validation schema
- Write concern

One to One Relation

```
Command Prompt - mongo
> use book
switched to db book
> db.books.insertOne({Name: "Famous Fice",Year: 2020})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5fd257869f1fdb96366b296d")
}
> db.books.insertOne({Name: "The Secret seven",Year: 2004})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5fd257fb9f1fdb96366b296e")
}
> db.author.insertOne({Name:"Enid Blyton",age:33,BookWritten: ObjectId("5fd257fb9f1fdb96366b296e")})
...
> db.author.insertOne({Name:"Enid Blyton",age:33,BookWritten: ObjectId("5fd257fb9f1fdb96366b296e")})
uncaught exception: ReferenceError: ObjectId is not defined :
@ (shell):1:48
> db.author.insertOne({Name:"Enid Blyton",age:33,BookWritten: ObjectId("5fd257fb9f1fdb96366b296e")})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5fd258c89f1fdb96366b296f")
}
> db.books.find().pretty()
{
  "_id" : ObjectId("5fd257869f1fdb96366b296d"),
  "Name" : "Famous Fice",
  "Year" : 2020
}
{
  "_id" : ObjectId("5fd257fb9f1fdb96366b296e"),
  "Name" : "The Secret seven",
  "Year" : 2004
}
> db.author.find().pretty()
{
  "_id" : ObjectId("5fd258c89f1fdb96366b296f"),
  "Name" : "Enid Blyton",
  "age" : 33,
  "BookWritten" : ObjectId("5fd257fb9f1fdb96366b296e")
}
>
```

One to Many Relation

```
Command Prompt - mongo
> use cityData
switched to db cityData
> db.cities.insertOne({name: "Kottayam",coordinates:{lat:21,lng:55}})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5fd259e99f1fdb96366b2970")
}
> db.cities.findOne()
{
  "_id" : ObjectId("5fd259e99f1fdb96366b2970"),
  "name" : "Kottayam",
  "coordinates" : {
    "lat" : 21,
    "lng" : 55
  }
}
> db.citizens.insertmany([
  {name: "Akshay",cityId : ObjectId("5fd259e99f1fdb96366b2970")},
  ... {name:"Arya",cityId: ObjectId("5fd259e99f1fdb96366b2970")}]])
uncaught exception: ReferenceError: Objectid is not defined :
@shell):2:14
> db.citizens.insertmany([
  {name: "Akshay",cityId : ObjectId("5fd259e99f1fdb96366b2970")},
  {name:"Arya",cityId: ObjectId("5fd259e99f1fdb96366b2970")}]])
uncaught exception: TypeError: db.citizens.insertmany is not a function :
@shell):1:1
> db.citizens.insertMany([
  {name: "Akshay",cityId : ObjectId("5fd259e99f1fdb96366b2970")},
  {name:"Arya",cityId: ObjectId("5fd259e99f1fdb96366b2970")}]])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("5fd25a849f1fdb96366b2971"),
    ObjectId("5fd25a849f1fdb96366b2972")
  ]
}
> db.citizens.find().pretty()
{
  "_id" : ObjectId("5fd25a849f1fdb96366b2971"),
  "name" : "Akshay",
  "cityId" : ObjectId("5fd259e99f1fdb96366b2970")
}
{
  "_id" : ObjectId("5fd25a849f1fdb96366b2972"),
  "name" : "Arya",
  "cityId" : ObjectId("5fd259e99f1fdb96366b2970")
}
```

Many to Many Relation

```
Command Prompt - mongo
> use shop
switched to db shop
> db.products.insertOne({title: "Abook",price: 12.99})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5fd25b579f1fdb96366b2973")
}
> db.customers.insertOne({name: "Arjun",age: 22})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5fd25b719f1fdb96366b2974")
}
> db.orders.insertOne({productID: ObjectId("5fd25b579f1fdb96366b2973"),customerID : ObjectId("5fd25b719f1fdb96366b2974")})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5fd25bb79f1fdb96366b2975")
}
> db.orders.drop()
true
> db.customers.updateOne({},{$set :{orders : [{productID: ObjectId("5fd25b579f1fdb96366b2973"),quantity: 2}]}})
...
...
> db.customers.updateOne({},{$set :{orders : [{productID: ObjectId("5fd25b579f1fdb96366b2973"),quantity: 2}]}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.customers.findOne()
{
  "_id" : ObjectId("5fd25b719f1fdb96366b2974"),
  "name" : "Arjun",
  "age" : 22,
  "orders" : [
    {
      "productID" : ObjectId("5fd25b579f1fdb96366b2973"),
      "quantity" : 2
    }
  ]
}
> db.products.findOne()
{ "_id" : ObjectId("5fd07a93021c50b4ca8524f5"), "name" : "JS BOOK" }
>
```

Writeconcern


Command Prompt - mongo

```
> db.persons.insertOne({name:"Arjun",age:22},{writeConcern: {w:1}})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5fd25d3c9f1fdb96366b2976")
}
> db.persons.insertOne({name:"Akshay",age:22},{writeConcern: {w:1,j:false}})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5fd25d599f1fdb96366b2977")
}
> db.persons.insertOne({name:"Akshay",age:22},{writeConcern: {w:1,j:true}})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5fd25d6d9f1fdb96366b2978")
}
> db.persons.insertOne({name:"Akhil",age:21},{writeConcern: {w:1,j:true,wtimeout: 2000}})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5fd25d909f1fdb96366b2979")
}
> db.persons.insertOne({name:"Akhil",age:21},{writeConcern: {w:1,j:true,wtimeout: 1}})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5fd25da79f1fdb96366b297a")
}
>
```

\$lookup operation

```
db.book.updateOne({},{$set :{authors:[ObjectId("5fd24fc19f1fdb96366b2969"),ObjectId("5fd2509e9f1fdb96366b296a")]]}})
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
db.book.updateOne({},{$set :{authors:[ObjectId("5fd24fc19f1fdb96366b2969"),ObjectId("5fd2509e9f1fdb96366b296a")]]}})
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
db.book.updateOne({},{$set :{authors:[ObjectId("5fd2509e9f1fdb96366b296a"),ObjectId("5fd24fc19f1fdb96366b2969")]]}})
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
db.books.findOne()
{
  "_id" : ObjectId("5fd24fc19f1fdb96366b2969"),
  "name" : "My favorite Book",
  "authors" : [
    {
      "name" : "venkat",
      "age" : 32
    },
    {
      "name" : "Sajith",
      "age" : 29
    }
  ]
}
db.books.aggregate([{$lookup: {from:"authors",localField: "authors",foreignField: "_id",as:"creators"}}])
{ "_id" : ObjectId("5fd24fc19f1fdb96366b2969"), "name" : "My favorite Book", "authors" : [ { "name" : "venkat", "age" : 32 }, { "name" : "Sajith", "age" : 29 } ], "creators" : [ ] }
db.books.aggregate([{$lookup: {from:"authors",localField: "authors",foreignField: "_id",as:"creators"}}]).pretty()
{
  "_id" : ObjectId("5fd24fc19f1fdb96366b2969"),
  "name" : "My favorite Book",
  "authors" : [
    {
      "name" : "venkat",
      "age" : 32
    },
    {
      "name" : "Sajith",
      "age" : 29
    }
  ],
  "creators" : [ ]
}
```

Validation Schema



```
> db.createCollection("posts", {
...   validator: {
...     $jsonSchema: {
...       bsonType: "object",
...       required: ['title', 'text', 'creator', 'comments'],
...       properties: {
...         title: {
...           bsonType: 'string',
...           description: 'must be a string and it is required'
...         },
...         text: {
...           bsonType: 'string',
...           description: "must be a string and it is required"
...         },
...         creator: {
...           bsonType: 'objectId',
...           description: "Must be an object id and this is required"
...         },
...         comments: {
...           bsonType: 'array',
...           required: ['text', 'author'],
...           properties: {
...             text: {
...               bsonType: 'string',
...               description: "must be a string and required"
...             },
...             author: {
...               bsonType: 'objectId',
...               description: "must be an objectId and required"
...             }
...           }
...         }
...       }
...     }
...   }
... })
{ "ok" : 1 }
```




Thank You!