MongoDB Relations

ONE TO ONE

```
> show dbs
admin         0.000GB
blog          0.000GB
bookRegistry  0.000GB
carData       0.000GB
cityData      0.000GB
companyData   0.000GB
config        0.000GB
flightData    0.000GB
local         0.000GB
movies        0.000GB
posts         0.000GB
shop          0.000GB
ustglobal     0.000GB
> use demo
switched to db demo
> db.user.insertOne({name:"Akshay",hobby:"learning"})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd246f0a34e1b3475c96780")
}
> db.post.insertOne(
... {title:"Comments",description:"this is text"}
... )
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd247cda34e1b3475c96781")
}
> db.post.updateOne({_id: ObjectId("5fd247cda34e1b3475c96781")},{$set:{commentby: ObjectId("5fd246f0a34e1b3475c96780")}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.post.findOne()
{
        "_id" : ObjectId("5fd247cda34e1b3475c96781"),
        "title" : "Comments",
        "description" : "this is text",
        "commentby" : ObjectId("5fd246f0a34e1b3475c96780")
}
>
```

ONE TO MANY

```
> use cityData
switched to db cityData
> db.city.insertOne({name:"alappuzha" ,citytype:"urban",coordinates:{lat:444,long:555}})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd24e5b9027a43537b56b11")
}
> db.person.insertMany({name:"AKSHAY",cityID:ObjectId("5fd24e5b9027a43537b56b11")},{name:"ANILKUMAR",cityID:ObjectId("5fd24e5b9027a43537b56b11")})
uncaught exception: TypeError: documents.map is not a function :
DBCollection.prototype.insertMany@src/mongo/shell/crud_api.js:307:17
@(shell):1:1
> db.person.insertMany([{name:"AKSHAY",cityID:ObjectId("5fd24e5b9027a43537b56b11")},{name:"ANILKUMAR",cityID:ObjectId("5fd24e5b9027a43537b56b11")}])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5fd24f0d9027a43537b56b12"),
                ObjectId("5fd24f0d9027a43537b56b13")
        ]
}
> db.person.find().pretty()
{
        "_id" : ObjectId("5fd24f0d9027a43537b56b12"),
        "name" : "AKSHAY",
        "cityID" : ObjectId("5fd24e5b9027a43537b56b11")
}
{
        "_id" : ObjectId("5fd24f0d9027a43537b56b13"),
        "name" : "ANILKUMAR",
        "cityID" : ObjectId("5fd24e5b9027a43537b56b11")
}
>
```

```
> use game
switched to db game
> db.user.insertOne({name:"akshay",age:23})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd250f79027a43537b56b14")
}
> db.games.insertOne({gameName:"Apex legends",genre:"Battle royale"})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd251469027a43537b56b15")
}
> db.user.updateOne({},{$set:{Copies:[{gameID:ObjectId("5fd251469027a43537b56b15"),quantity:5}]}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.user.find().pretty()
{
        "_id" : ObjectId("5fd250f79027a43537b56b14"),
        "name" : "akshay",
        "age" : 23,
        "Copies" : [
                {
                        "gameID" : ObjectId("5fd251469027a43537b56b15"),
                        "quantity" : 5
                }
        ]
}
>
```

```
> use gameData
switched to db gameData
> db.game.insertMany([{_id:"g1" ,name:"APEX" },{_id:"g2",name:"LEGENDS" }])
{ "acknowledged" : true, "insertedIds" : [ "g1", "g2" ] }
> db.game.findOne()
{ "_id" : "g1", "name" : "APEX" }
> db.game.find()
{ "_id" : "g1", "name" : "APEX" }
{ "_id" : "g2", "name" : "LEGENDS" }
> db.user.insertOne({name:"AKSHAY", games:["g1","g2"]})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd24bdf9027a43537b56b10")
}
> db.user.find().pretty()
{
        "_id" : ObjectId("5fd24bdf9027a43537b56b10"),
        "name" : "AKSHAY",
        "games" : [
                "g1",
                "g2"
        ]
}
>
```

MANY TO MANY

```
> use warehouse
switched to db warehouse
> db.products.insertOne({name:"fruits and veg",price:400,owners:[{name:"akshay"},{name:"anil"}]})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd256959027a43537b56b16")
}
> db.owners.insertMany([{name:"akshay"},{name:"anil"}])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5fd256e69027a43537b56b17"),
                ObjectId("5fd256e69027a43537b56b18")
        ]
}
> db.products.updateOne({},{$set:{owners:[ ObjectId("5fd256e69027a43537b56b17"),
...                 ObjectId("5fd256e69027a43537b56b18")]}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.products.pretty()
uncaught exception: TypeError: db.products.pretty is not a function :
@(shell):1:1
> db.products.find().pretty()
{
        "_id" : ObjectId("5fd256959027a43537b56b16"),
        "name" : "fruits and veg",
        "price" : 400,
        "owners" : [
                ObjectId("5fd256e69027a43537b56b17"),
                ObjectId("5fd256e69027a43537b56b18")
        ]
}
>
```

Join using $lookup

```
> db.products.find().pretty()
{
        "_id" : ObjectId("5fd256959027a43537b56b16"),
        "name" : "fruits and veg",
        "price" : 400,
        "owners" : [
                ObjectId("5fd256e69027a43537b56b17"),
                ObjectId("5fd256e69027a43537b56b18")
        ]
}
> db.products.aggregate([{$lookup:{ from: "owners",localField: "owners",foreignField:"_id",as: "owners"}}]).pretty()
{
        "_id" : ObjectId("5fd256959027a43537b56b16"),
        "name" : "fruits and veg",
        "price" : 400,
        "owners" : [
                {
                        "_id" : ObjectId("5fd256e69027a43537b56b17"),
                        "name" : "akshay"
                },
                {
                        "_id" : ObjectId("5fd256e69027a43537b56b18"),
                        "name" : "anil"
                }
        ]
```

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 a object and this is required"
...             },
...             comments: {
...                 bsonType: 'array',
...                 required: ['text', 'author'],
...                 properties: {
...                     text: {
...                         bsonType: 'string',
...                 description: "must be a string and it is required"
...
...                             },
...                 author : {
...                     bsonType: 'objectId',
...                     description: "must be a object and this is required"
...                 }
...                     }
...                 }
...             }
...             }
...         }} )
{ "ok" : 1 }
```

WriteConcern

```
> db.persons.insertOne({name:"akshay",age:23},{writeConcern: {w:1}})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd25eb69027a43537b56b19")
}
> db.persons.insertOne({name:"malu",age:22},{writeConcern: {w:1,j:false}})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd25ee29027a43537b56b1a")
}
> db.persons.insertOne({name:"nassrin",age:22},{writeConcern: {w:1,j:true}})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd25eff9027a43537b56b1b")
}
> db.persons.insertOne({name:"nassrin",age:22},{writeConcern: {w:1,j:true,wtimeout:2000}})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd25f2a9027a43537b56b1c")
}
> db.persons.insertOne({name:"nassrin",age:22},{writeConcern: {w:1,j:true,wtimeout:1}})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd25f339027a43537b56b1d")
}
>
```