Relations IN MONGODB

BY

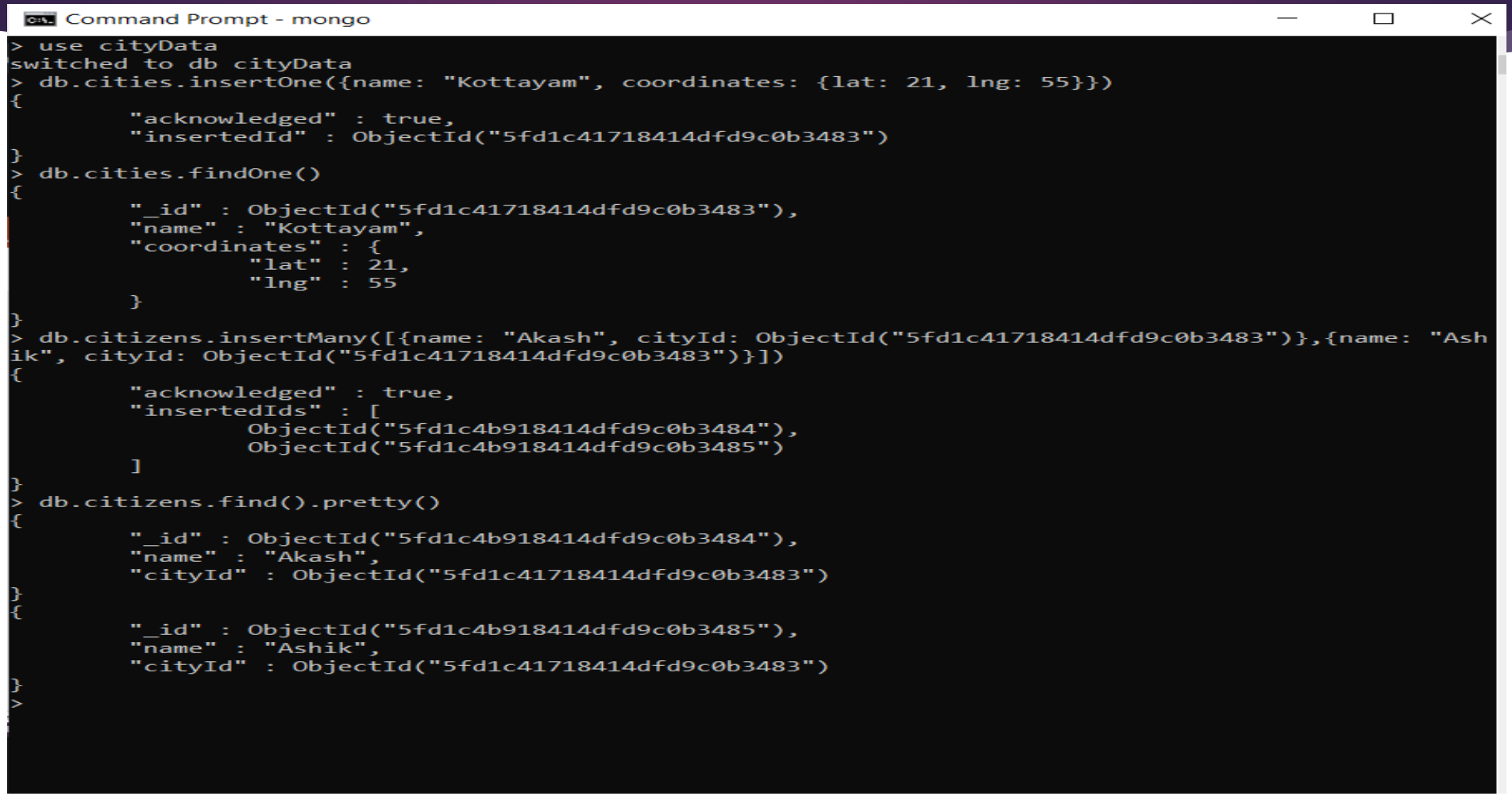
AKHIL M ANIL

ONE to ONE Relation

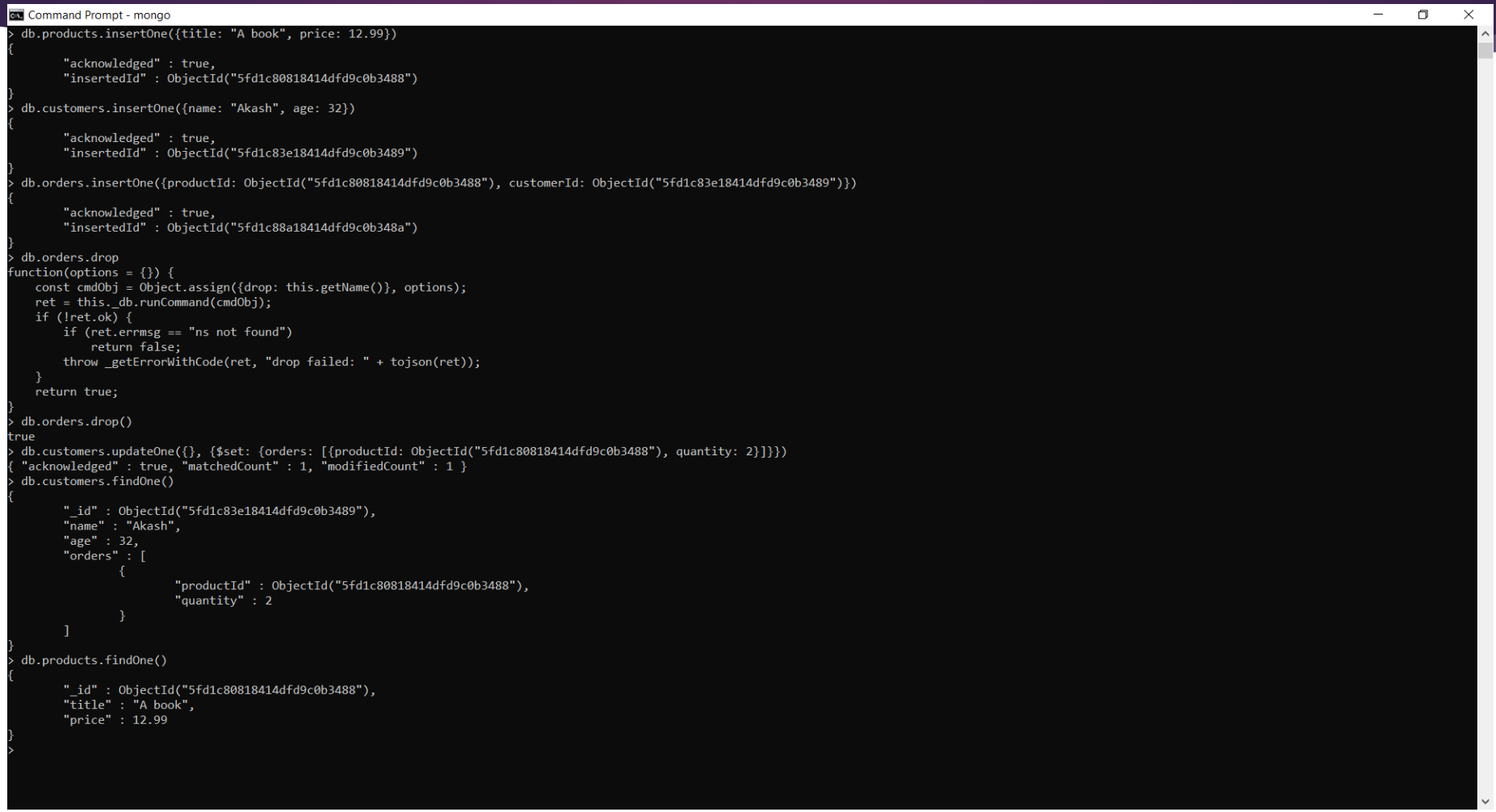
Command Prompt - mongo

```
> use book
switched to db book
> db.books.insertOne({Name: "A book", Year: 2020})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd24875f4064862fca225e0")
}
> db.author.insertOne({Name: "Akhil", age: 22, BookWritten: "ObjectId("5fd24875f4064862fca225e0")"})
uncaught exception: SyntaxError: identifier starts immediately after numeric literal :
@(shell):1:69
> db.author.insertOne({Name: "Akhil", age: 22, BookWritten: ObjectId("5fd24875f4064862fca225e0")})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd24923f4064862fca225e1")
}
> db.books.find().pretty()
{
        "_id" : ObjectId("5fd24875f4064862fca225e0"),
        "Name" : "A book",
        "Year" : 2020
}
> db.author.find().pretty()
{
        "_id" : ObjectId("5fd24923f4064862fca225e1"),
        "Name" : "Akhil",
        "age" : 22,
        "BookWritten" : ObjectId("5fd24875f4064862fca225e0")
}
>
```

ONE to MANY Relation



MANY to MANY Relation



Joining using $lookup

```
> db.books.aggregate( [ { $lookup : { from : "authors" , localField : "Authors" , foreignField : "_id" , as: "Authors" }}]).pretty()
{
        "_id" : ObjectId("5fd1cee75151e8a105d39f43"),
        "Title" : "A book",
        "Price" : 230,
        "Authors" : [
                {
                        "_id" : ObjectId("5fd1cf145151e8a105d39f44"),
                        "Name" : "Asad"
                },
                {
                        "_id" : ObjectId("5fd1cf145151e8a105d39f45"),
                        "Name" : "Dev"
                }
        ]
}
```

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',
...                          description:'Must Be an array',
...                          required: ['text', 'author'],
...                          items: {
...                               properties: {
...                               text: {
...                                    bsonType: 'string',
...                                    description: "must be a string and required"
...                               },
...                               author: {
...                                    bsonType: 'objectId',
...                                    description: "must be an objectId and required"
...                               }
...                          }
...                          }
...                     }
...                }
...           }
...      }
... })
{ "ok" : 1 }
```

Validation Schema Test

```
> db.posts.insertOne({title:"My first post", text:"MongoDB is a NO Sql Database",creator:ObjectId("5fd216655151e8a105d39f50"),comments:[{text:"I will try MongoDB",autho
r:123}]})
WriteError({
        "index" : 0,
        "code" : 121,
        "errmsg" : "Document failed validation",
        "op" : {
                "_id" : ObjectId("5fd217a25151e8a105d39f52"),
                "title" : "My first post",
                "text" : "MongoDB is a NO Sql Database",
                "creator" : ObjectId("5fd216655151e8a105d39f50"),
                "comments" : [
                        {
                                "text" : "I will try MongoDB",
                                "author" : 123
                        }
                ]
        }
}) :
WriteError({
        "index" : 0,
        "code" : 121,
        "errmsg" : "Document failed validation",
        "op" : {
                "_id" : ObjectId("5fd217a25151e8a105d39f52"),
                "title" : "My first post",
                "text" : "MongoDB is a NO Sql Database",
                "creator" : ObjectId("5fd216655151e8a105d39f50"),
                "comments" : [
                        {
                                "text" : "I will try MongoDB",
                                "author" : 123
                        }
                ]
        }
})
```

writeConcern