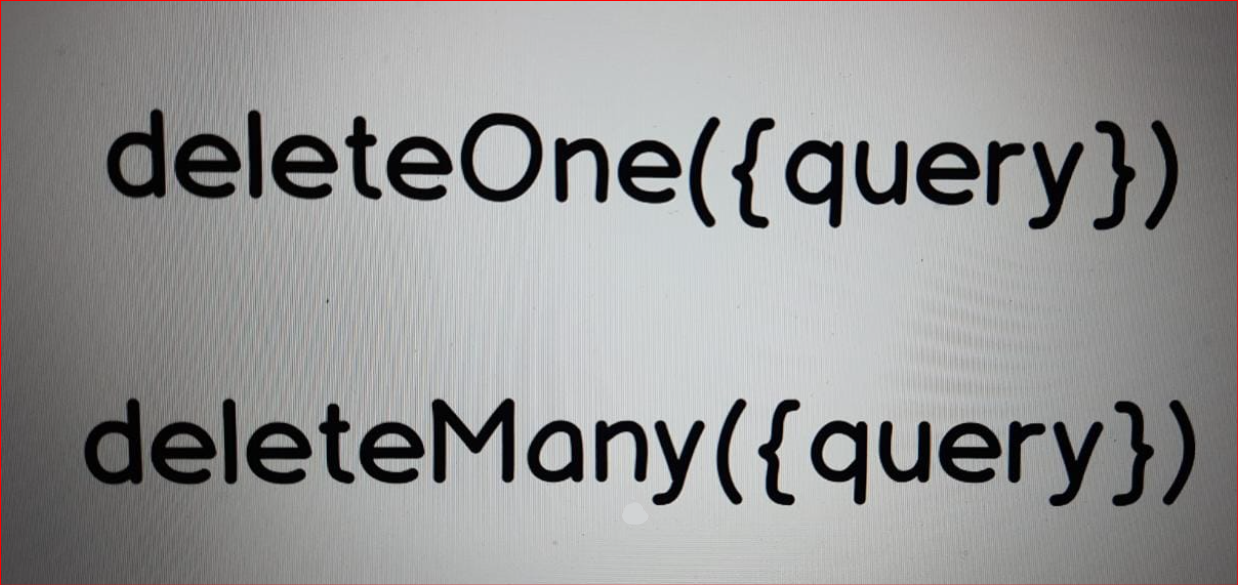
**32 - Delete Methods deleteOne() and deleteMany()**

In this section we will talk about delete methods.

They are very simple and you need to pass only query as an argument here.

"deleteOne" will delete just one document.

"deleteMany" will delete all documents that match query.



**33. PRACTICE - Deleting Documents**

* Let's now try "deleteOne" and "deleteMany" methods in action.

And in order to do that let me insert a couple of new documents into the "posts" collection.

**// insert dummy documents first**

**// insertOne({postId: NumberInt(1111)})**

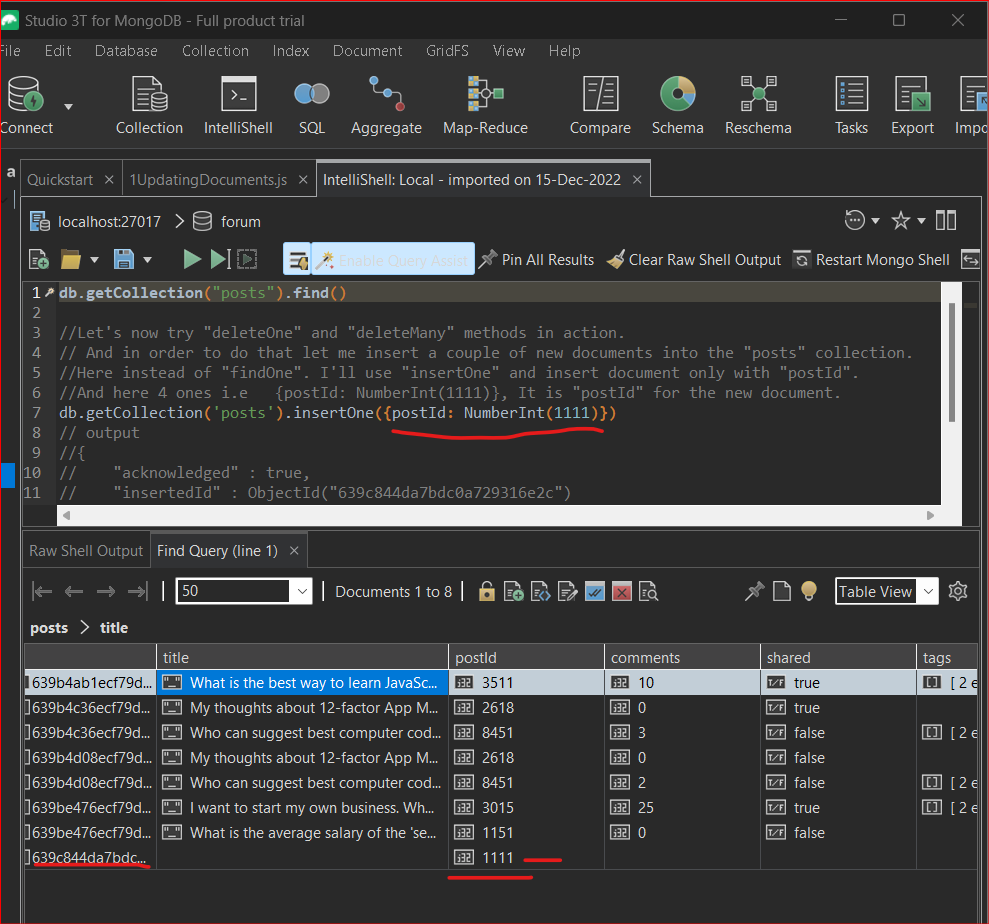
Here instead of "findOne"

I'll use "insertOne" and insert document only with "postId".

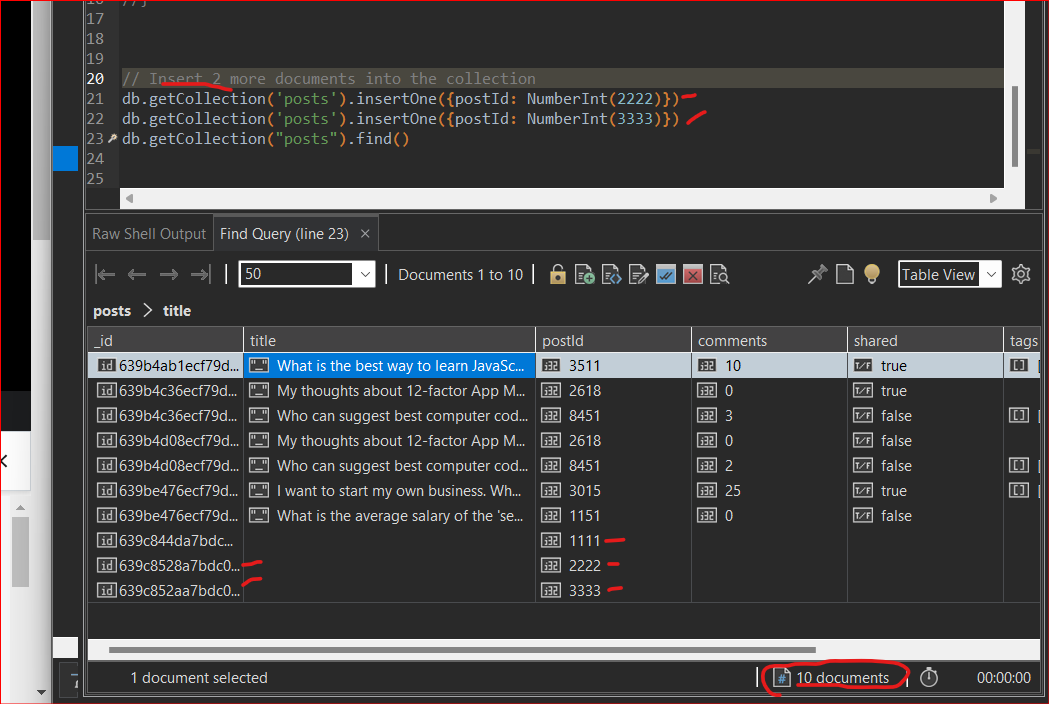
{postId: NumberInt()}

And here 4 ones i.e postId: NumberInt(1111)}

It is "postId" for the new document.



Now insert one more document with "postId" equal to NumberInt(2222) and 3333 as third "postId".



Now in my collection there will be 10 documents.

**Let's first delete this one with this "postId": 1111**

**// deleteOne()**

//Let's first delete this one with this "postId": 1111

db.getCollection('posts').deleteOne({postId: 1111})

// O/p:

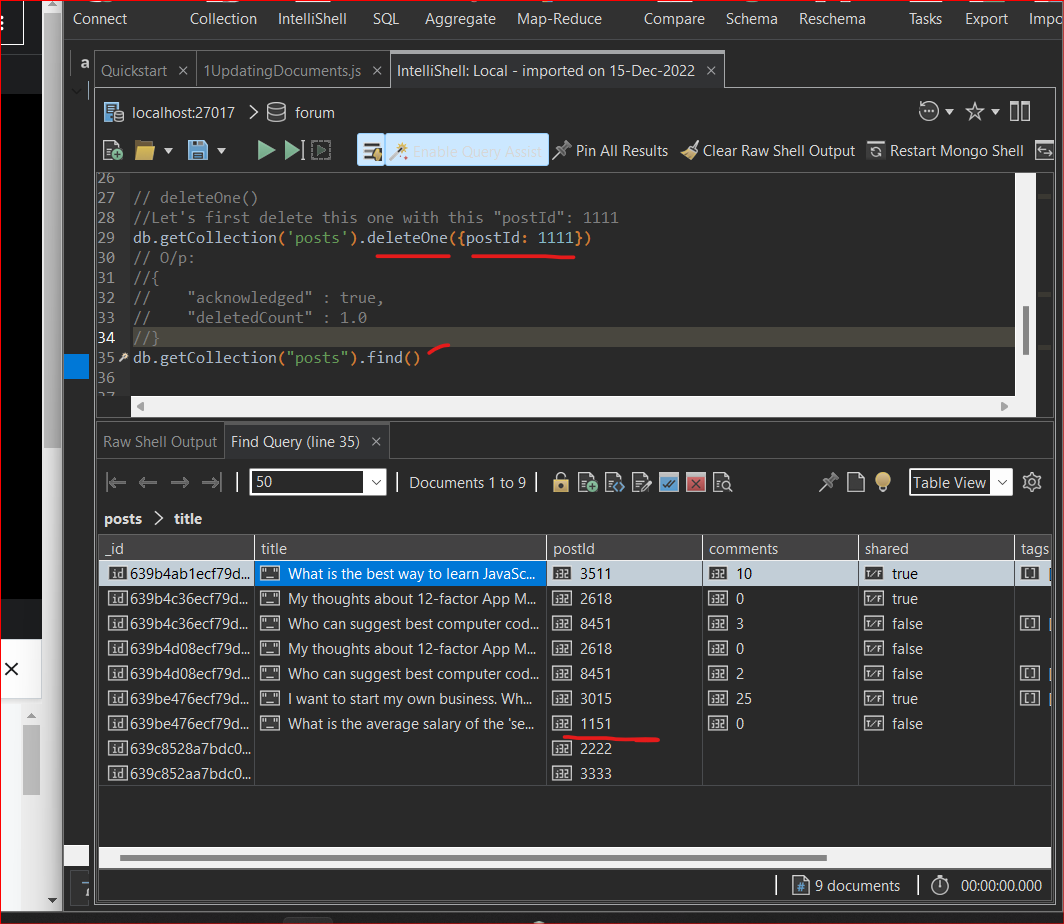
//{

// "acknowledged" : true,

// "deletedCount" : 1.0

//}

db.getCollection("posts").find()



One document was deleted and now if I'll double click on the "posts" collection I will not see this document.

**// deleteMany() we need to use $exists operator**

**Now let's delete those two documents. With postId : 2222 and 3333**

**Use "deleteMany" method and delete all documents that don't have "title" field.**

This task was relatively challenging because I have asked you to find documents that don't have specific field and for this case you need to use $exists operator and we have not discussed this operator in the section.

Let's use "deleteMany".

**// find the documents**

db.getCollection('posts').find({title: {$exists: false}})

//{

// "\_id" : ObjectId("639c8528a7bdc0a729316e2d"),

// "postId" : NumberInt(2222)

//}

//{

// "\_id" : ObjectId("639c852aa7bdc0a729316e2e"),

// "postId" : NumberInt(3333)

//}

**// Now delete the above 2 documents**

db.getCollection('posts').deleteMany(

{title: {$exists: false}}

)

// {

// "acknowledged" : true,

// "deletedCount" : 2.0

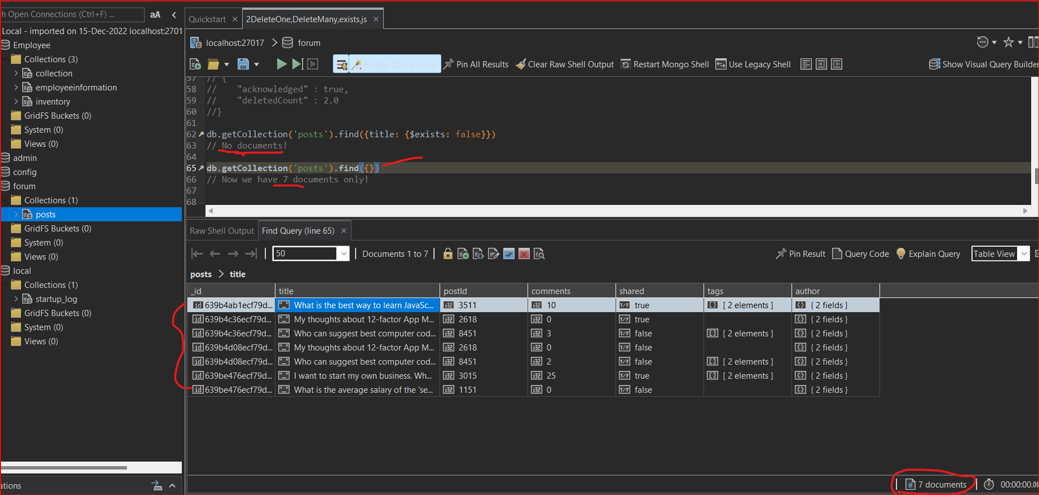
//}

db.getCollection('posts').find({title: {$exists: false}})

// No documents!

db.getCollection('posts').find({})

// Now we have 7 documents only!



**OK let's move on to the last section in this section where I will shortly overview some advanced MongoDB features.**