**Downloading Studio 3T (According to MongoDB we need to download ‘Studio 3T’ instead of Robo3T**

**For MongDB**

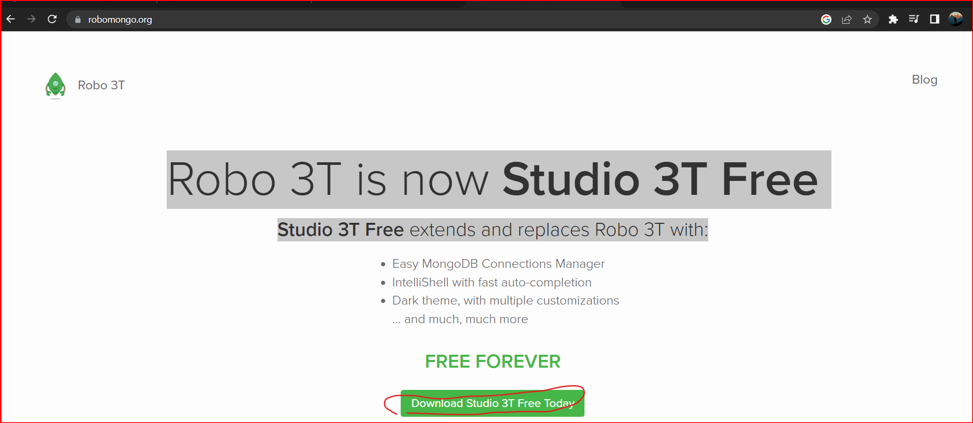
**-> Studio3T is the Client Application**

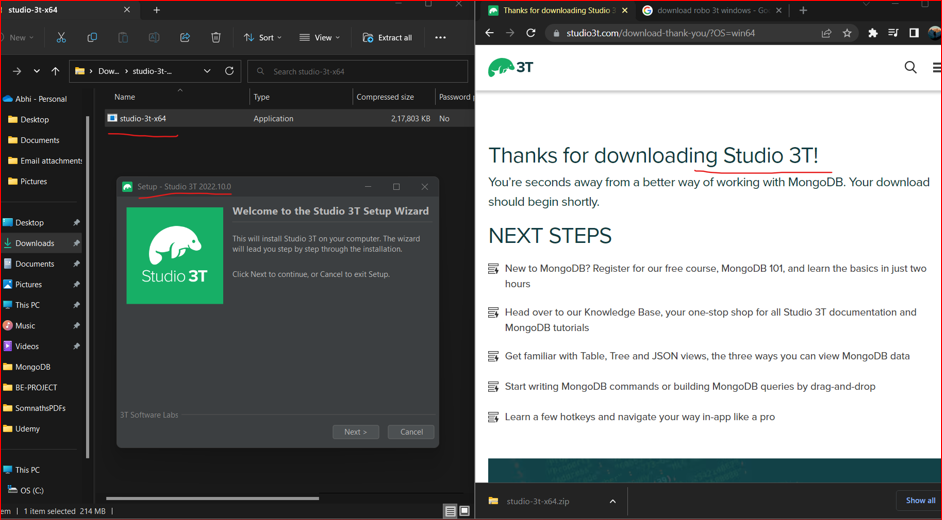
**//It is the MongoDB connection Manager (Just like a MySQL workbench or postgreSQL 'pgAdmin4' interface) Robo 3T is now Studio 3T**

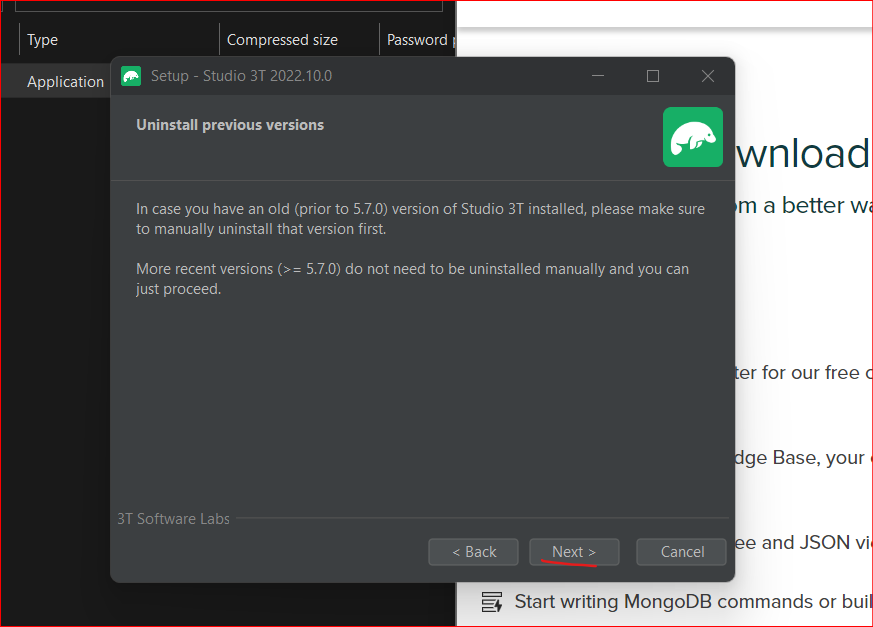
**--> Inside that we use "IntelliShell on Upper Tab" //Standard interface for Querying and Displaying the o/p (Just like a Query Tool in pgAdming4)**

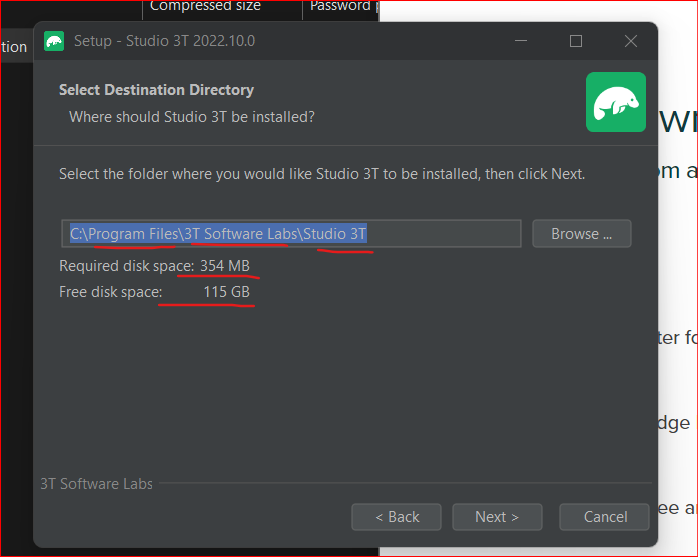
Navigate to: **robomongo.org**

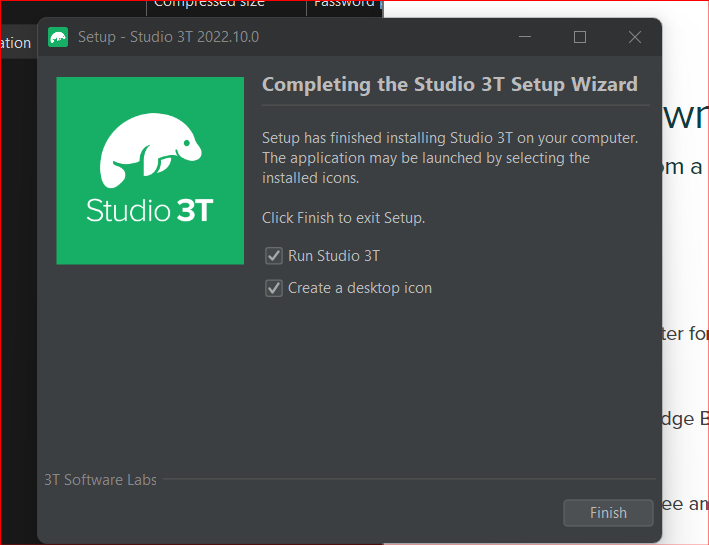
**Robo 3T is now Studio 3T Free, Studio 3T Free extends and replaces Robo 3T with:**

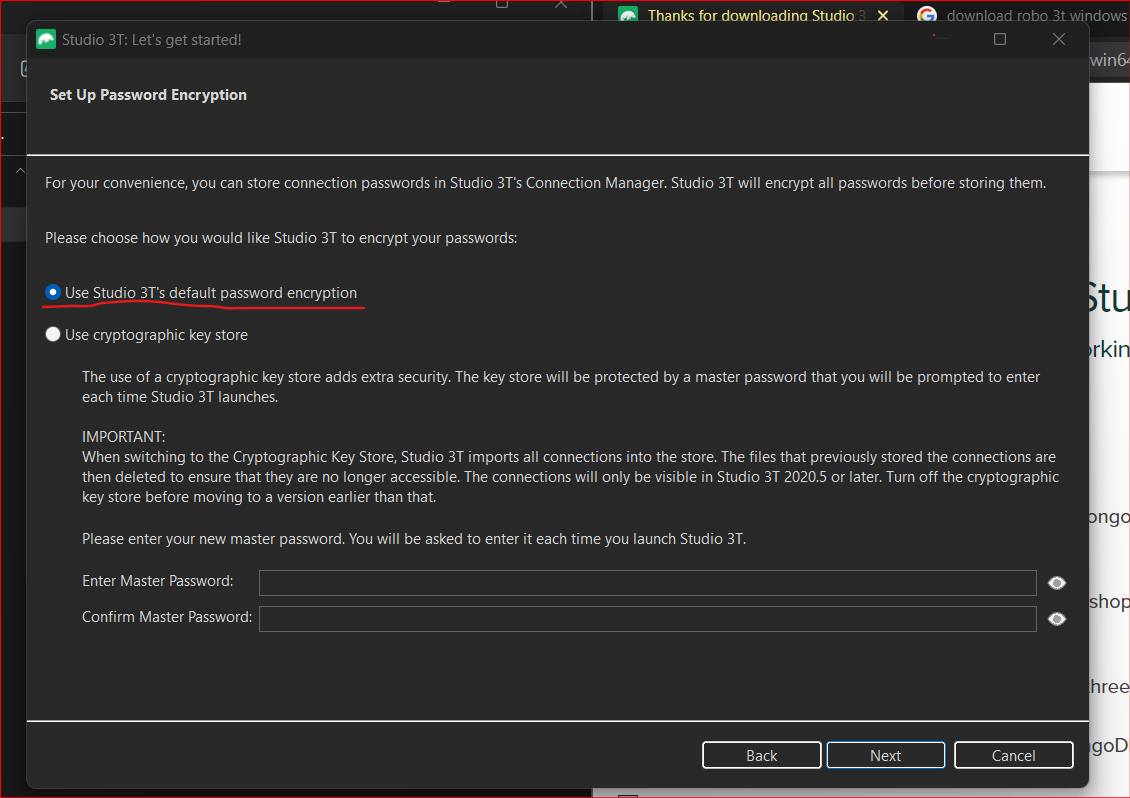
****

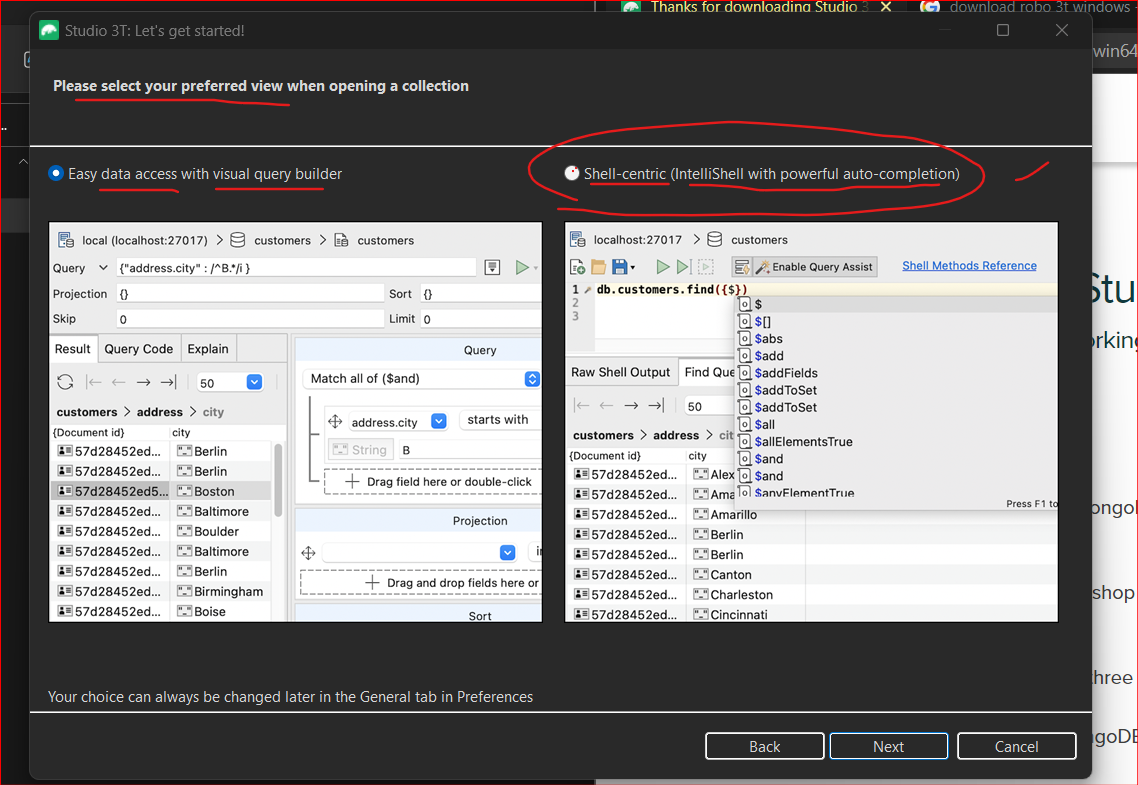


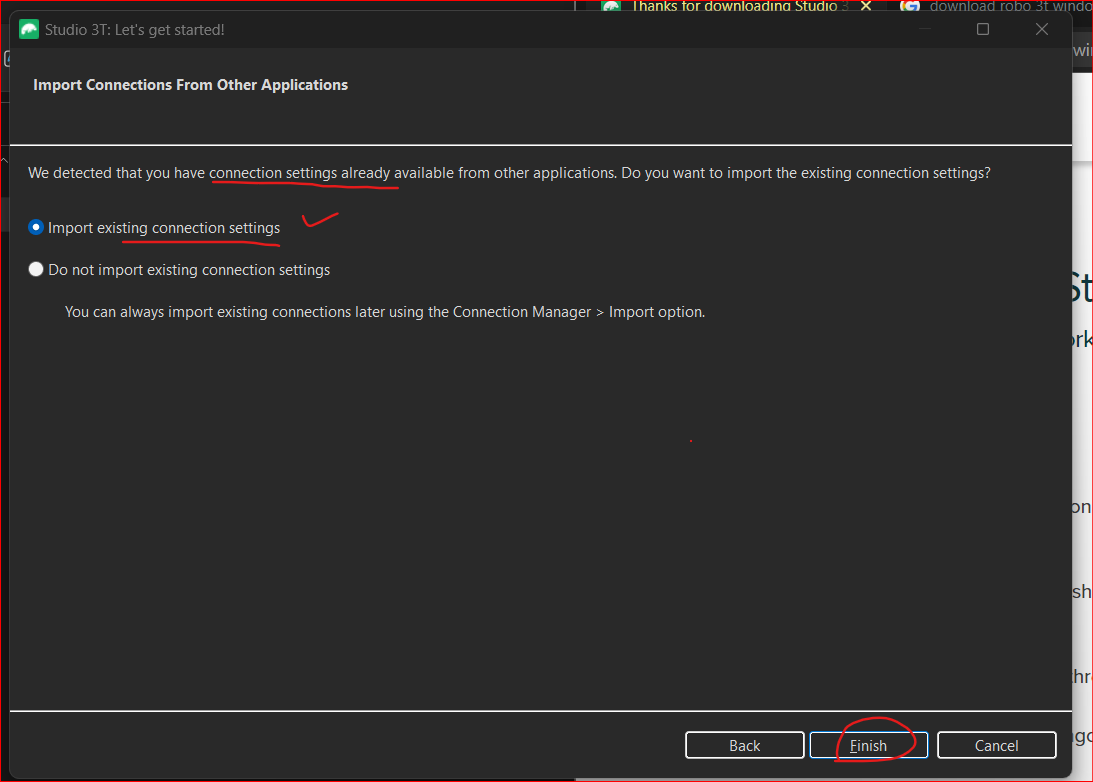


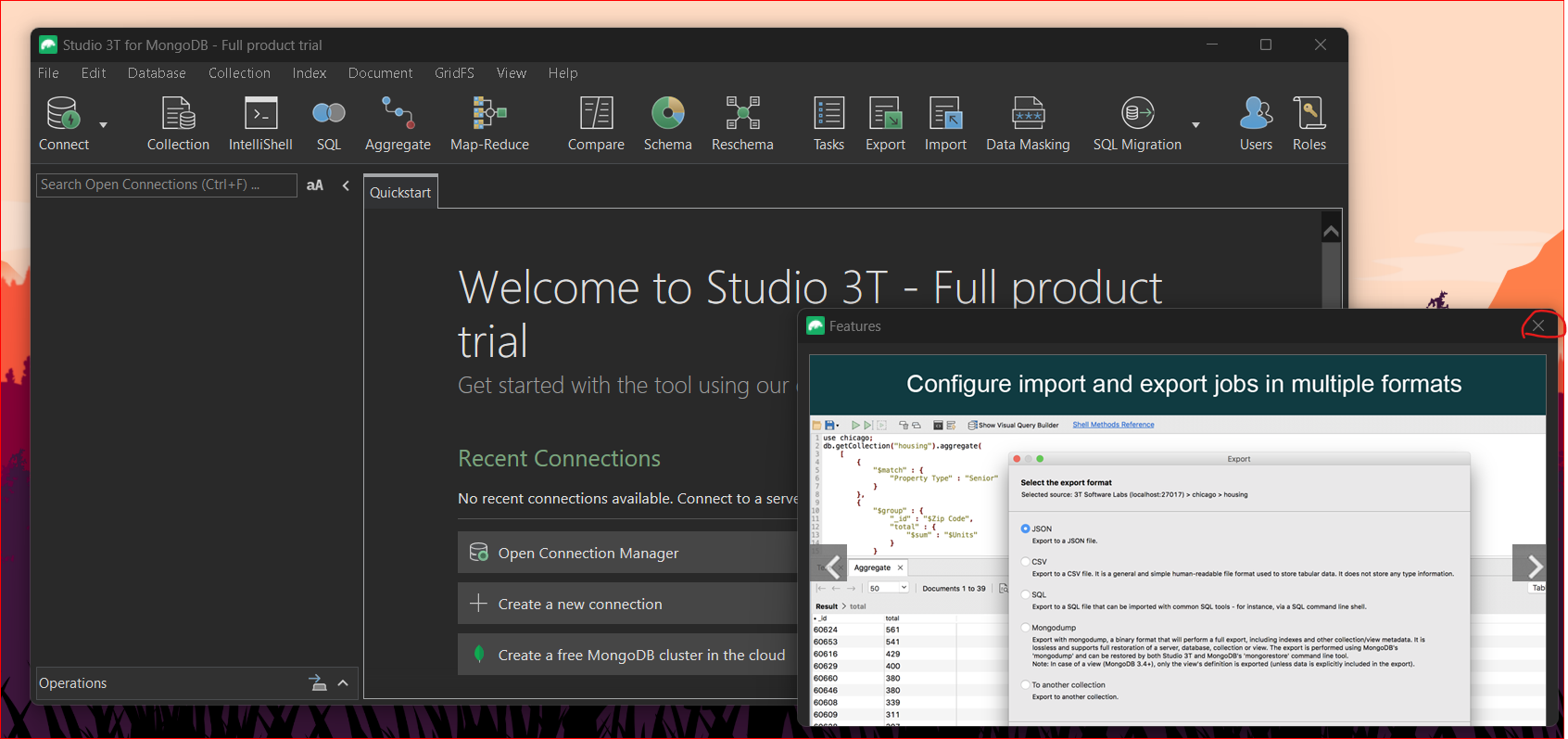


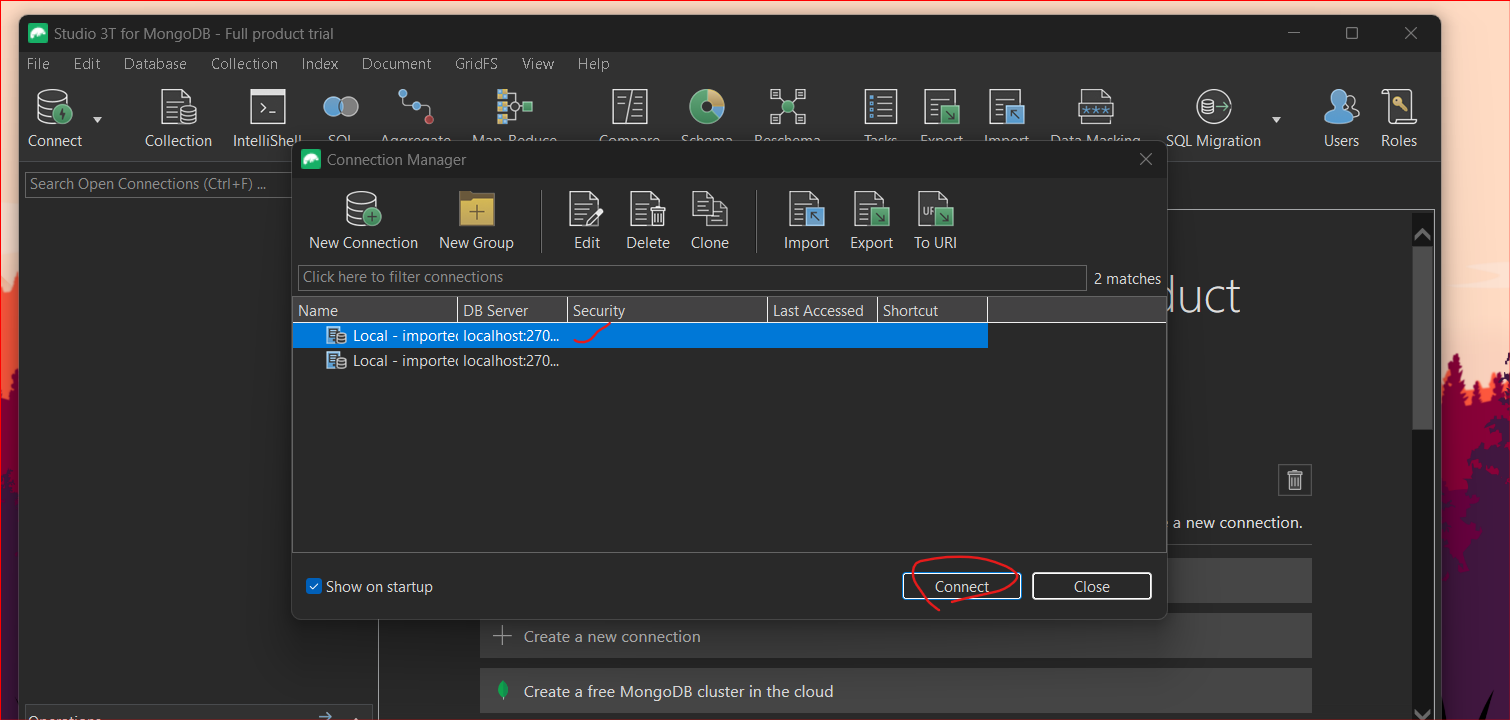


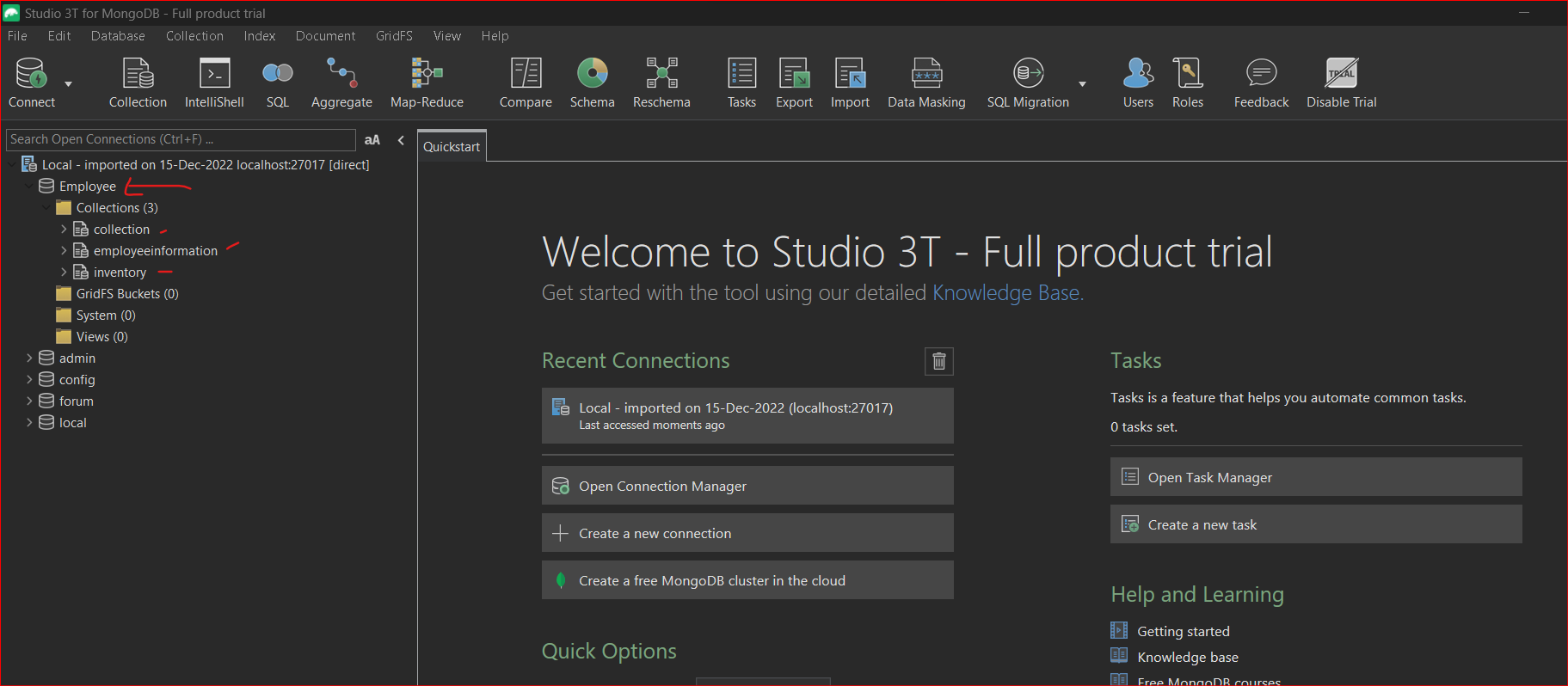








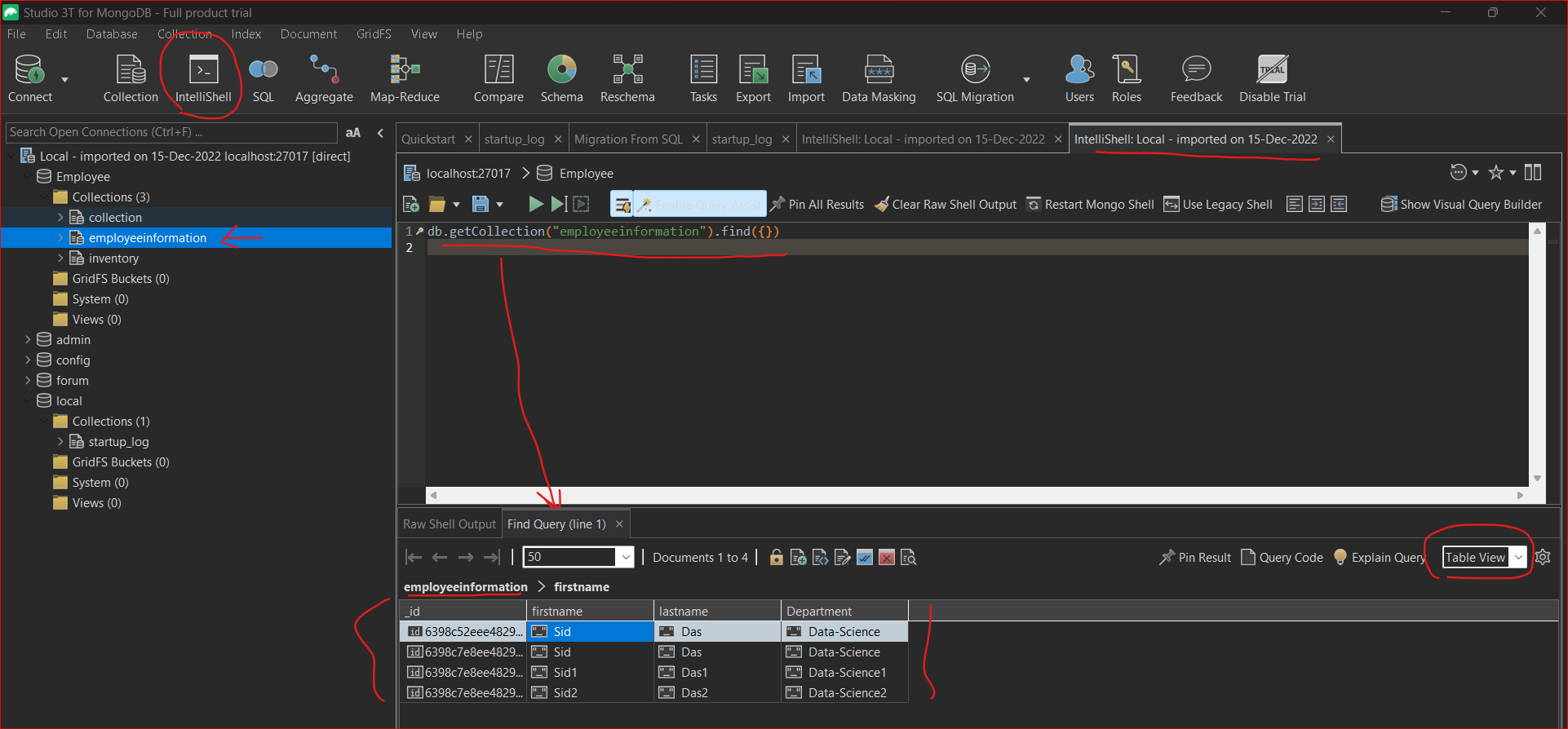




**Press on IntelliShell on Upper Tab**

**IntelliShell is the Standard interface for Querying and Displaying the o/p on Screen**

**And from LHS Tab select your Collections**



**CLTR + ENTER to Execute the Mongo commands**

**// are used for comments**

**CLTR + / for comment**

db.version();

see the Below **Raw Shell** for output of Your Query and **Find Query** where you can see the Table view/JSON view/ Tree view of the results

**Easy MongoDB Connections Manager**

IntelliShell with fast auto-completion

Dark theme, with multiple customizations

… and much,

**22. PRACTICE - Insert Sample Documents into the Collection**

**// This is byDefault Query comes up when you Double click on the databases on LHS tab**

db.getCollection("posts").find({}); // empty table

**// CLTR + Enter (To Run the Queries) and CTRL + / (to Comment down the lines)**

**// All the documents in posts(Section5).txt are Javascript objects: JavaScript object is variable type and JSON is a string. JavaScript does Not surround each key by double quotes.**

**// Document Insertion into the Collection (posts)**

**//1) Using insertOne() Method:**

**// Creating a new variable and passing the 1 JavaScript Object**

const doc = {

title: "What is the best way to learn JavaScript from the ground up?",

postId: NumberInt(3511),

comments: 10,

shared: true,

tags: [

"JavaScript",

"programming"

],

author: {

name: "Mike Forester",

nickname: "mikef"

}

};

**// passing doc as an argument**

db.posts.**insertOne**(doc);

**// Output/Result:**

//{

// "acknowledged" : true,

// "insertedId" : ObjectId("639b4ab1ecf79d9136c315c1")

//}

**//2) Using insertMany() Method**

**// It requires an array , that consist of several objects**

**// Creating a new variable and passing the 2 JavaScript Objects**

const **docArray** = [{

title: "My thoughts about 12-factor App Methodology",

postId: NumberInt(2618),

comments: 0,

shared: false,

tags: [],

author: {

name: "Emily Watson",

nickname: "emily23"

}

},

{

title: "Who can suggest best computer coding book for beginners?",

postId: NumberInt(8451),

comments: 2,

shared: false,

tags: [

"programming",

"coding"

],

author: {

name: "Emily Watson",

nickname: "emily23"

}

}];

db.posts.**insertMany**(docArray);

**//Output/Result:**

//{

// acknowledged: true,

// insertedIds: {

// '0': ObjectId("639b4c36ecf79d9136c315c2"),

// '1': ObjectId("639b4c36ecf79d9136c315c3")

// }

//}

**// 3) passing the remaining 4th and 5th javascript objects in the above same collection**

**// Note here we are using the same docArray document , it is incorrect you will get the duplicated entries of above 2 documents if you try to run any documents again in docArray**

**// Solution: To insert New Documents you need to give Unique Name in like : docArray1**

const **docArray1** = [{

title: "I want to start my own business. What I need to do first?",

postId: NumberInt(3015),

comments: 25,

shared: true,

tags: [

"business",

"money"

],

author: {

name: "Bob Hutchinson",

nickname: "bob1995"

}

},

{

title: "What is the average salary of the junior frontend developer?",

postId: NumberInt(1151),

comments: 0,

shared: false,

author: {

name: "Mike Forester",

nickname: "mikef"

}

}];

db.posts.insertMany(docArray1);

**//Output:**

//{

// acknowledged: true,

// insertedIds: {

// '0': ObjectId("639be476ecf79d9136c315c6"),

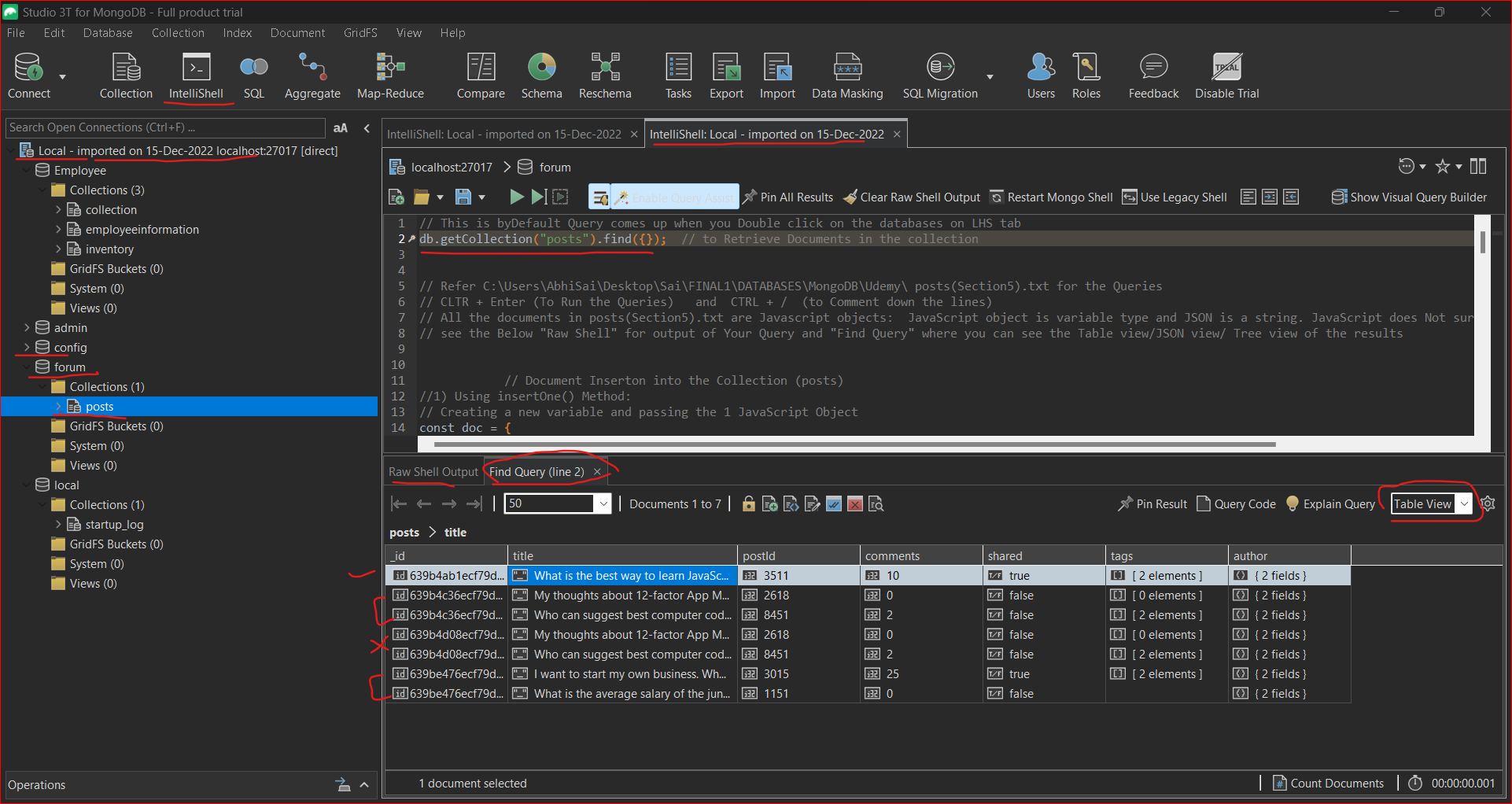
// '1': ObjectId("639be476ecf79d9136c315c7")

// }

//}

**See the Results in Table View**

**db.getCollection("posts").find({});** // to Retrieve Documents in the collection



**And in the next sections we will use find, update and delete operations and next section will be dedicated**

**to "find" and "findOne" methods.**

And you may notice that when you double click on the name of the collection immediately "find" method is executed.

And you may notice that empty object is passed to the "find" method.

So, let's talk about read operations in the next section.