**Assignment-4**

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Name** | **Email** |
| **8800149** | **Basavraj Jaliminche** | **Bjaliminche0149@econestogac.on.ca** |
| **8800060** | **Zarana Gohil** | **Zgohil0060@conestogac.on.ca** |
| **8807575** | **Dharti Patel** | **Dpatel757@conestogac.on.ca** |

**ASSIGNMENT 4**

1. **Recall our study of XML. Referencing your notes, look back at the section, how to Use, and study the Books.xml data set. Rewrite the data definitions using JSON and write a detailed, step-by-step explanation of how you completed the task.**

**Answer:**

An XML file is a type of data file that contains hierarchical parts. Custom tags, which specify the type of element, can be used by computer systems to access data stored in XML files.

XML is used in many aspects of web development.and often used to separate data from presentation.

The root element is “catalog”. Under the root element we have 12 child elements “book” and corresponding to the further child elements

**Book.xml**

<?xml version="1.0" encoding="UTF-8"?>

<bookstore>

<book category="COOKING">

<title lang="en">Everyday Italian</title>

<author>Giada De Laurentiis</author>

<year>2005</year>

<price>30.00</price>

</book>

<book category="CHILDREN">

<title lang="en">Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

<book category="WEB">

<title lang="en">XQuery Kick Start</title>

<author>James McGovern</author>

<author>Per Bothner</author>

<author>Kurt Cagle</author>

<author>James Linn</author>

<author>Vaidyanathan Nagarajan</author>

<year>2003</year>

<price>49.99</price>

</book>

<book category="WEB">

<title lang="en">Learning XML</title>

<author>Erik T. Ray</author>

<year>2003</year>

<price>39.95</price>

</book>

</bookstore>

## To open this xml file doc() function is used and the syntax is:

doc("books.xml")

## To navigate through the elements in an XML document, XQuery uses path expressions as shown below

doc("books.xml")/bookstore/book/year

## The above query returns following output:

<year>2005</year>

<year>2005</year>

<year>2003</year>

<year>2003</year>

**Process to convert XML into JSON data**

There are many ways to convert the XML data into JSON format like by using SQL Server, using C#, convert online and so on.

Here one of such process is explained to convert XML into JSON data with the help of console application in .NET using C# and the code related to this is as per below:

string xml = @""; //assign XML value here

XmlDocument doc = new XmlDocument(); doc.LoadXml(xml);

string json = JsonConvert.SerializeXmlNode(doc); Console.WriteLine(json);

## The output of the above XML with this code is as below:

[

{

"@category": "COOKING", "title": {

"@lang": "en",

"#text": "Everyday Italian"

},

"author": "Giada De Laurentiis", "year": "2005",

"price": "30.00"

},

{

"@category": "CHILDREN", "title": {

"@lang": "en", "#text": "Harry Potter"

},

"author": "J K. Rowling",

"year": "2005",

"price": "29.99"

},

{

"@category": "WEB", "title": {

"@lang": "en",

"#text": "XQuery Kick Start"

},

"author": [

"James McGovern", "Per Bothner", "Kurt Cagle", "James Linn",

"Vaidyanathan Nagarajan"

],

"year": "2003",

"price": "49.99"

},

{

"@category": "WEB", "title": {

"@lang": "en", "#text": "Learning XML"

},

"author": "Erik T. Ray",

"year": "2003",

"price": "39.95"

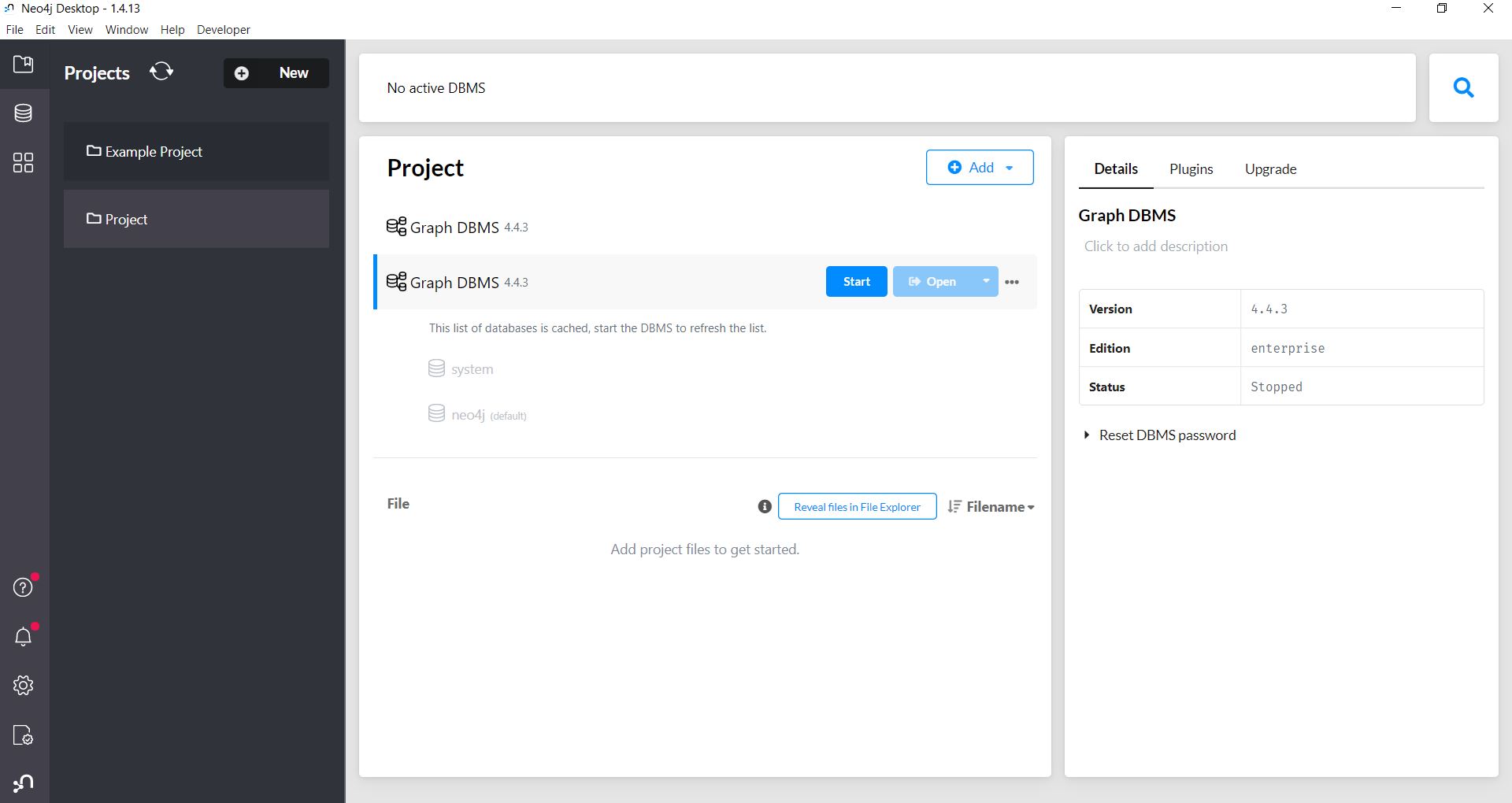
}

]

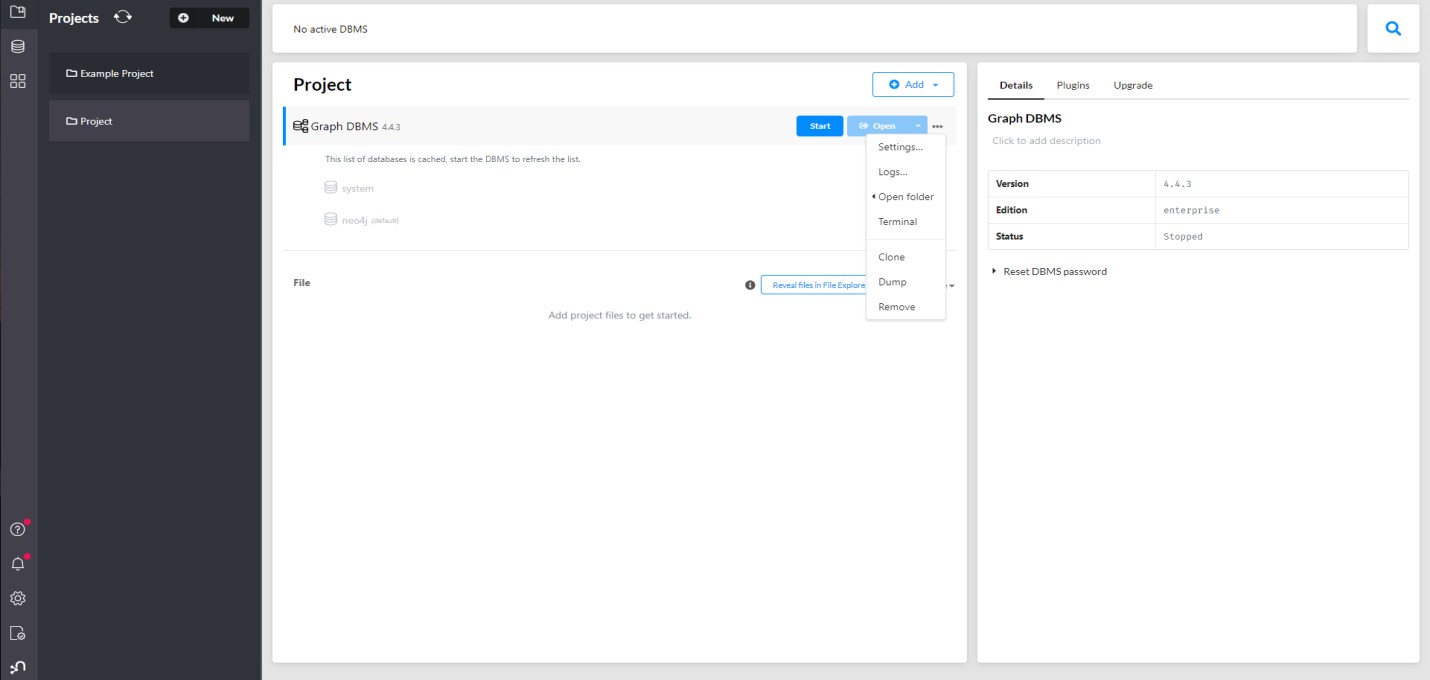
**2.Continuing with the previous task, convert the books data set to a graph database. Write a detailed, a step-by-step explanation of how you completed the work.**

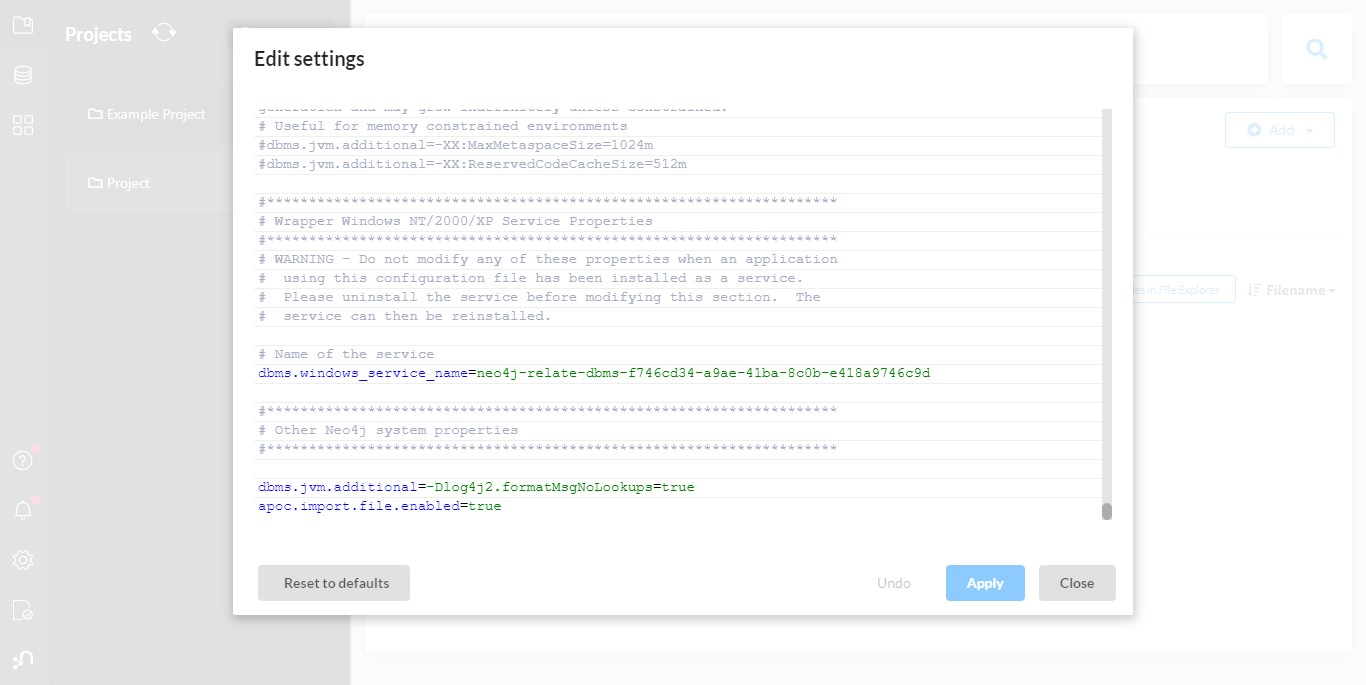
**Answer:**

Step 1: Open the Neo4J and create a project which shows the active database.



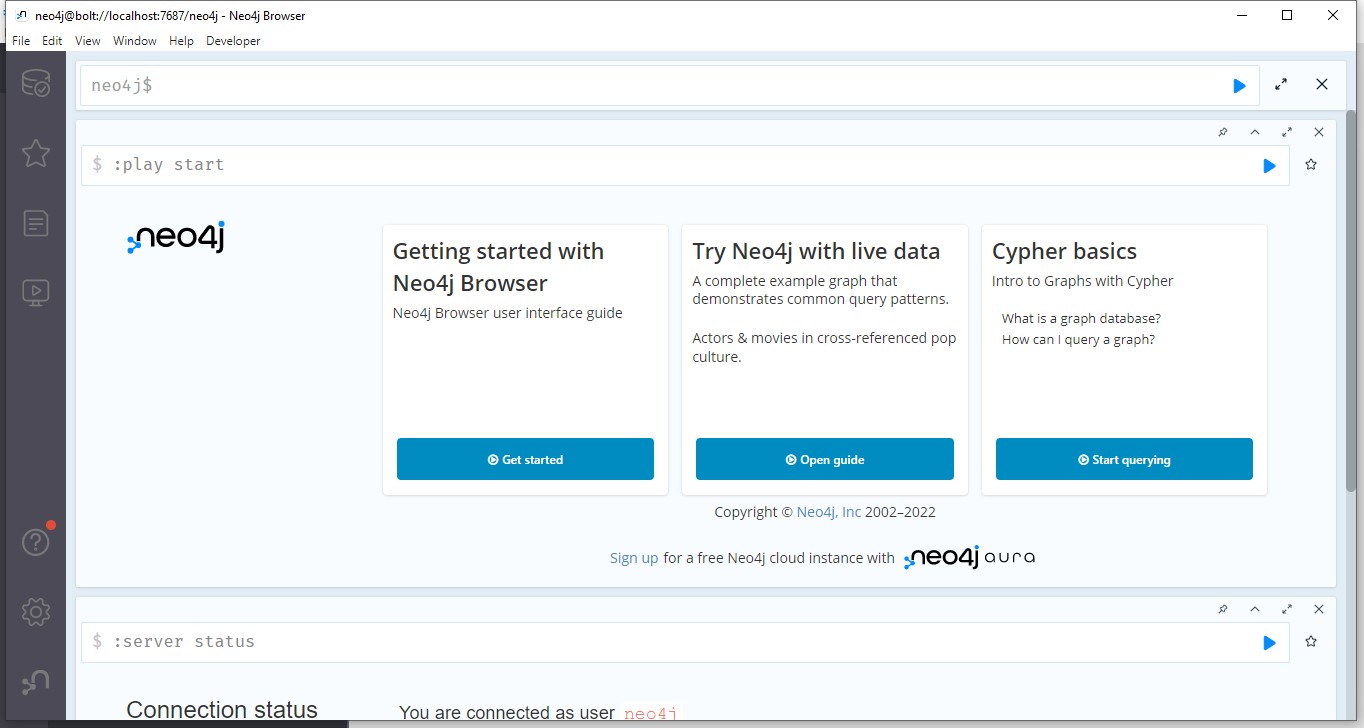
Step 2: After that click on setting to add install APOC plug-in.



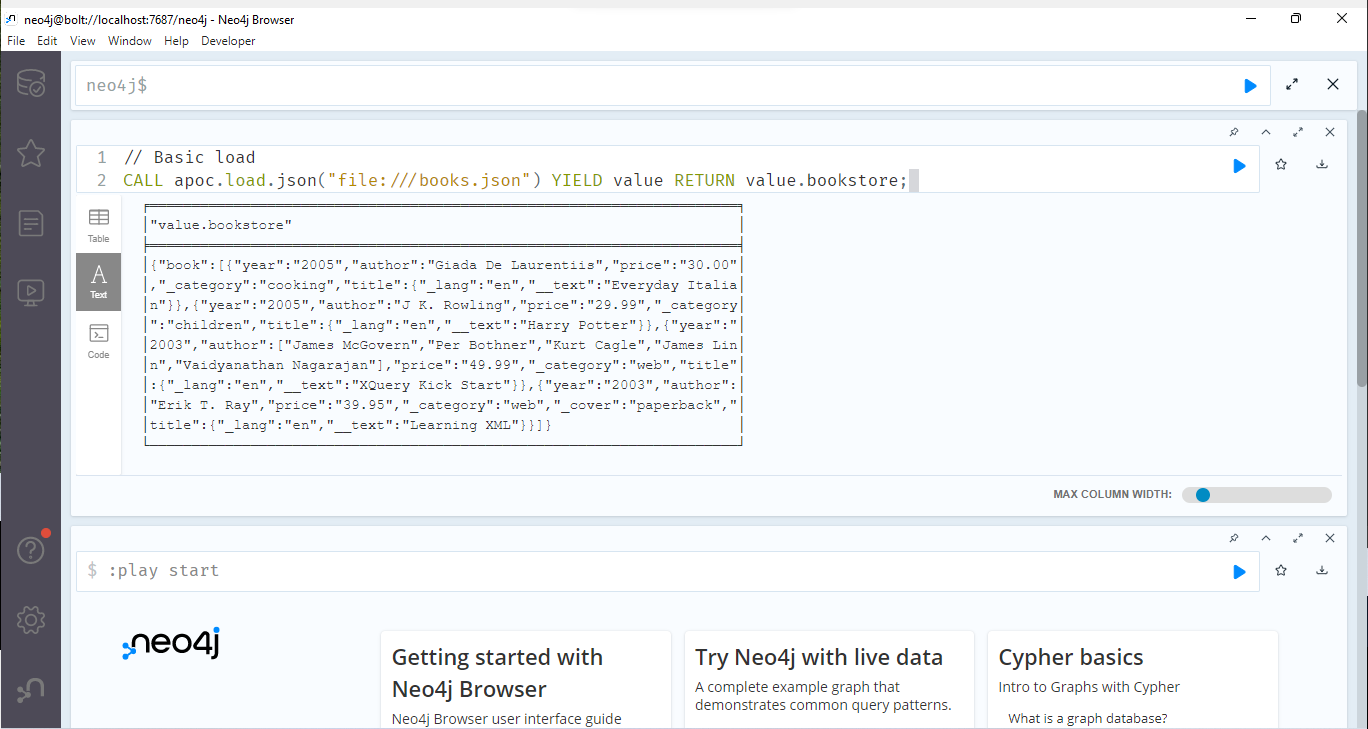
 Step 3: Add **apoc.import.file.enabled=true** configuration to add files from the device.

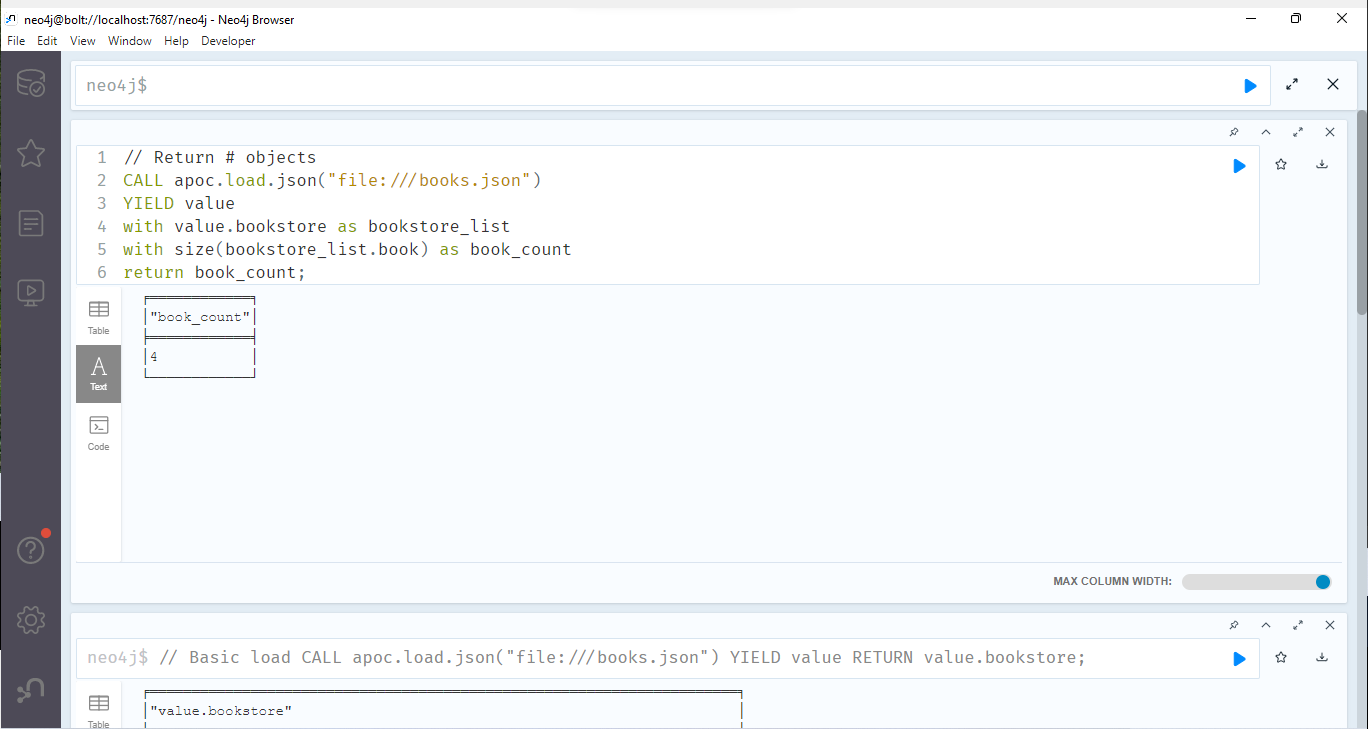
## 

## Step 4: After that open the db with Neo4j browser.

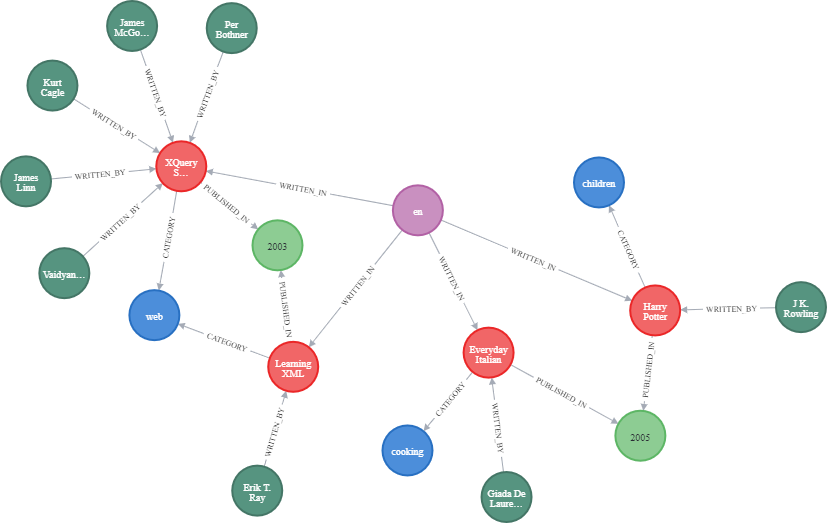


Step 5: Use call function to load Book.json file as shown below.



 Step 6: After loading the JSON data, it will return the number of records in JSON

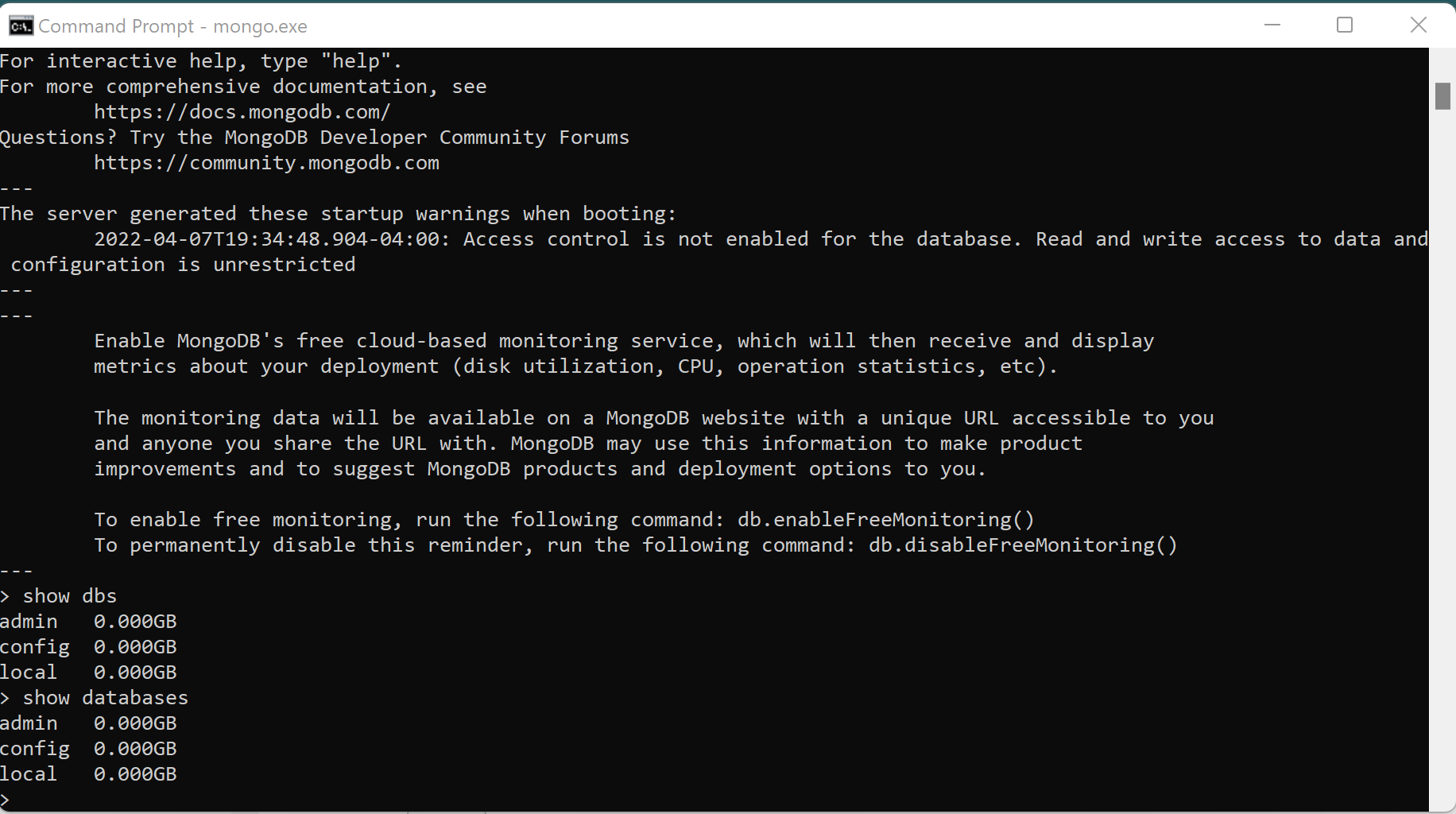
Step 7: To show he data we can return every record in table form

 Step 8: Write a query to import all the records and create a graph with the help of connections and nodes.

**3.A couple of weeks ago you were asked to complete Laboratory Exercise 6. Review the work  
that you completed for the exercise and write a short tutorial/learning object that documents  
the steps are taken to complete the work. Support your solution with a detailed set of screenshots.**

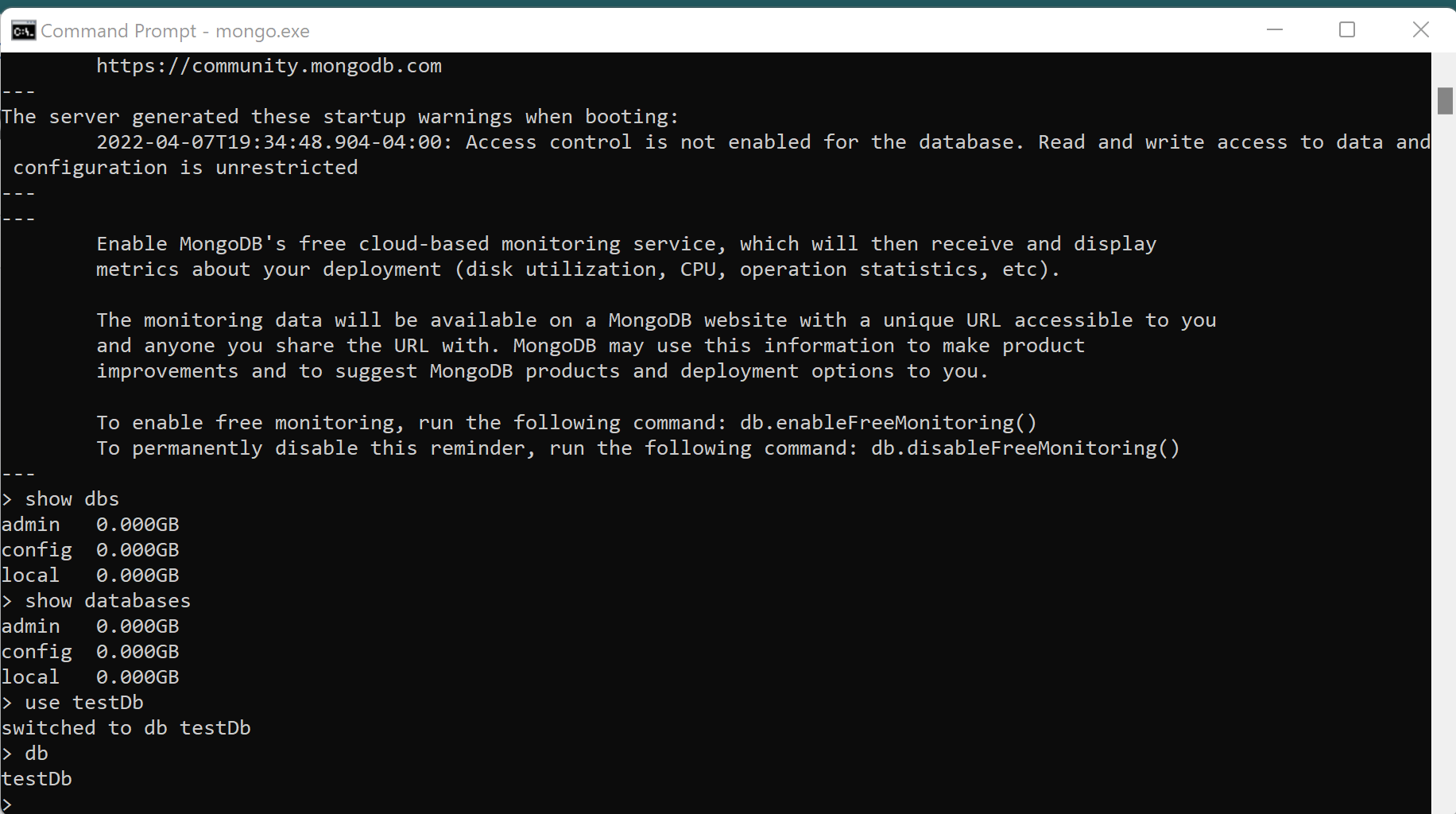
Answer:

* Verify that testDb is the current database being used.

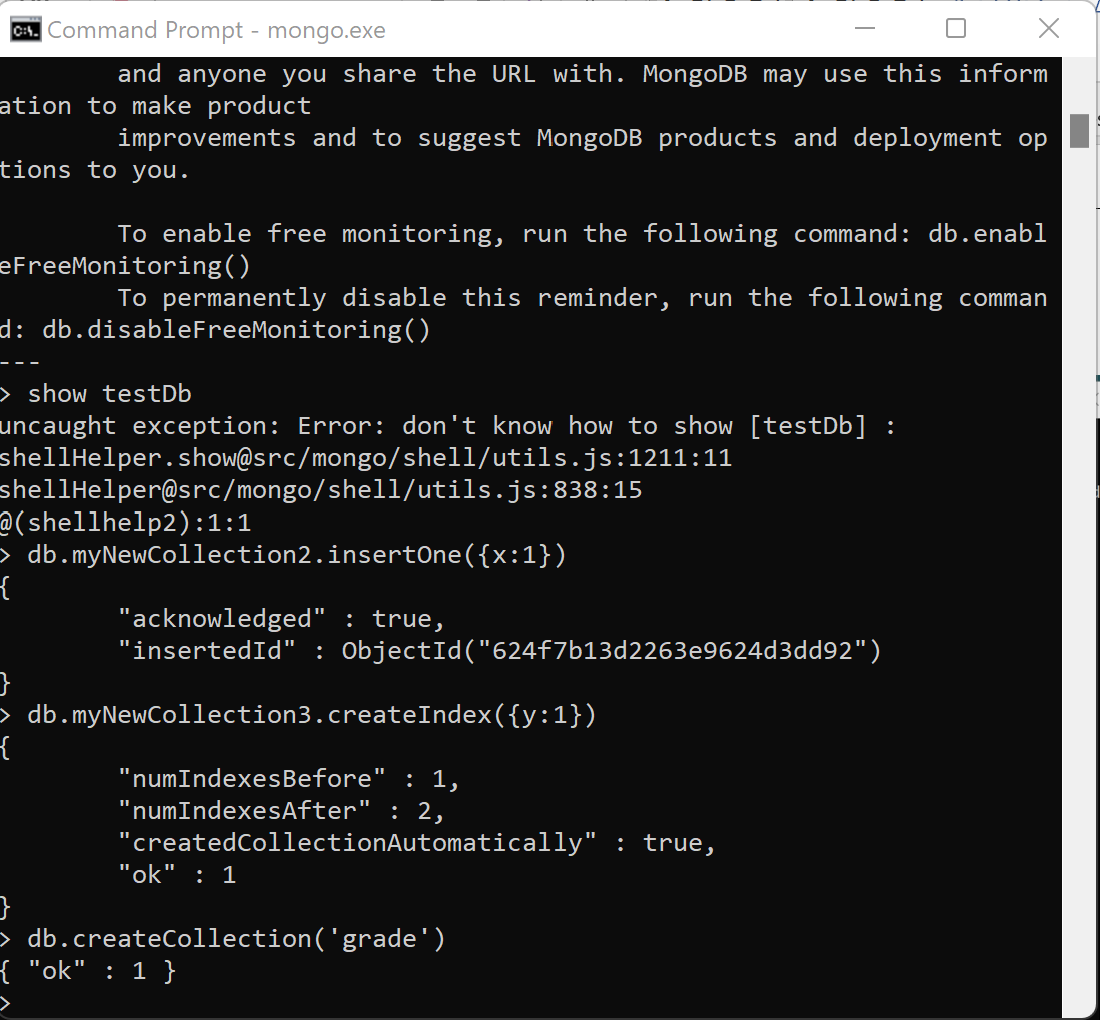


.

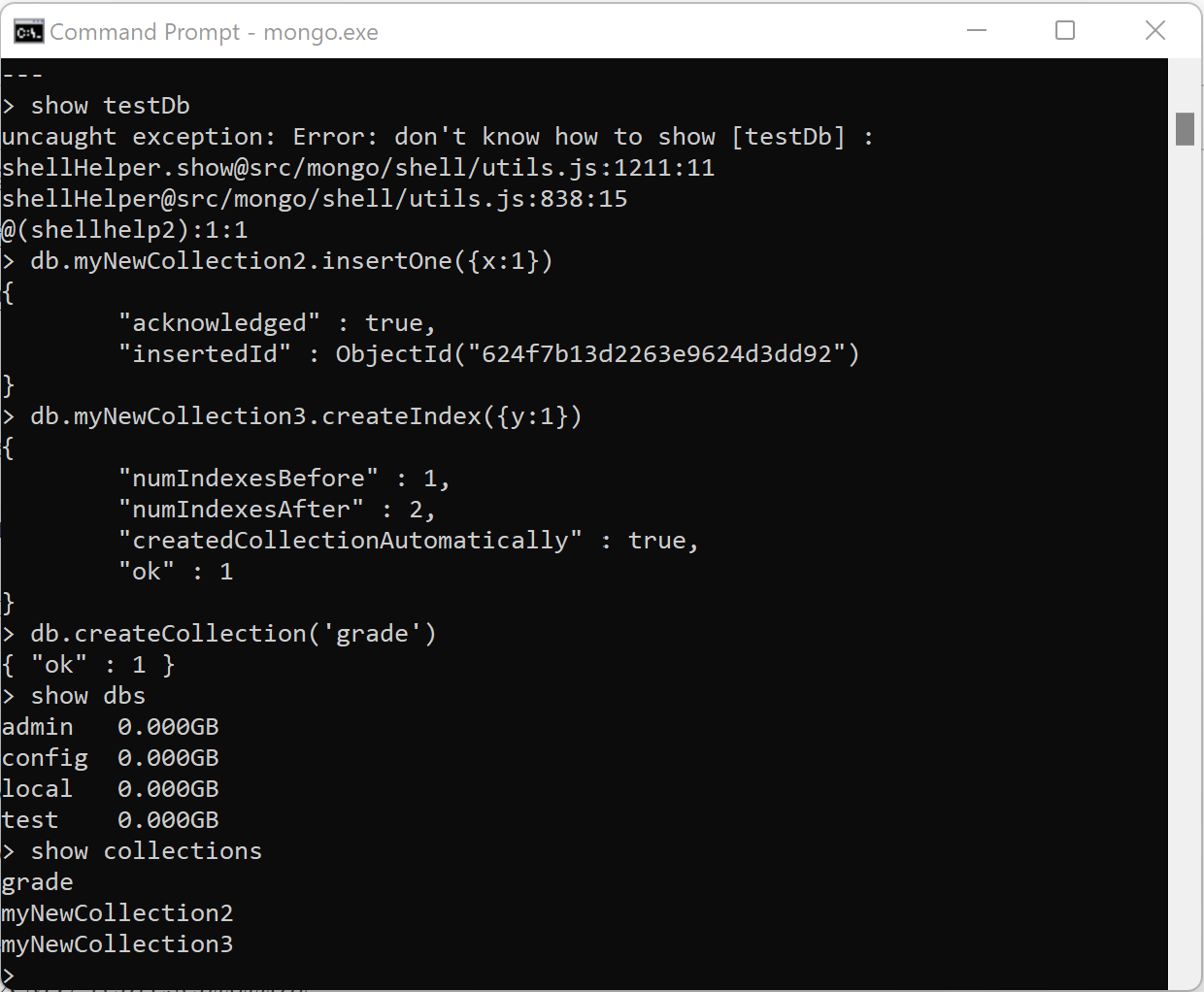
* show dbs to verify that testDb is in the list of available databases



* Type the following commands:  
  db.myNewCollection2.insertOne( { x : 1 } )  
  db.myNewCollection3.createIndex( { y : 1 } )  
  db.createCollection(‘grade’)



* To display a listing of all data collections, type the command: show collections



* Input the command: db.getCollectionInfos()
* This command retrieves the data collection method and it results in in JSON form.



* How would the data from the previous step be represented in XML?  
  Write an equivalent XML representation.

<?xml version="1.0" encoding="UTF-8" ?>

<root>

<0>

<name>grade</name>

<type>collection</type>

<options></options>

<info>

<readOnly>false</readOnly>

<uuid>UUID(8dbc8932-0338-48f3-8ef4-fe7524997a59)</uuid>

</info>

<idIndex>

<v>2</v>

<key>

<\_id>1</\_id>

</key>

<name>\_id\_</name>

</idIndex>

</0>

<1>

<name>myNewCollection2</name>

<type>collection</type>

<options></options>

<info>

<readOnly>false</readOnly>

<uuid> UUID(95cf0c37-7d0d-4f93-a34a-c218d7f99f37)</uuid>

</info>

<idIndex>

<v>2</v>

<key>

<\_id>1</\_id>

</key>

<name>\_id\_</name>

</idIndex>

</1>

<2>

<name>myNewCollection3</name>

<type>collection</type>

<options></options>

<info>

<readOnly>false</readOnly>

<uuid>UUID(135aff69-79b4-4d63-a243-7b351e774fc7)</uuid>

</info>

<idIndex>

<v>2</v>

<key>

<\_id>1</\_id>

</key>

<name>\_id\_</name>

</idIndex>

</2>

</root>