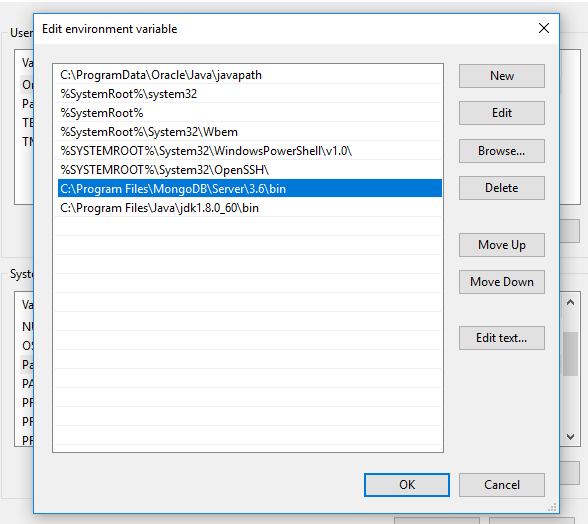
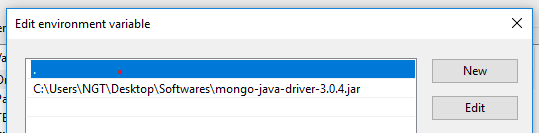
|  |  |  |  |
| --- | --- | --- | --- |
| **Srno** | **Date** | **Topic** | **Sign** |
| **5** |  | **Java and MongoDB** |  |

a. Connecting java with MongoDB and inserting, retrieving, updating and deleting.

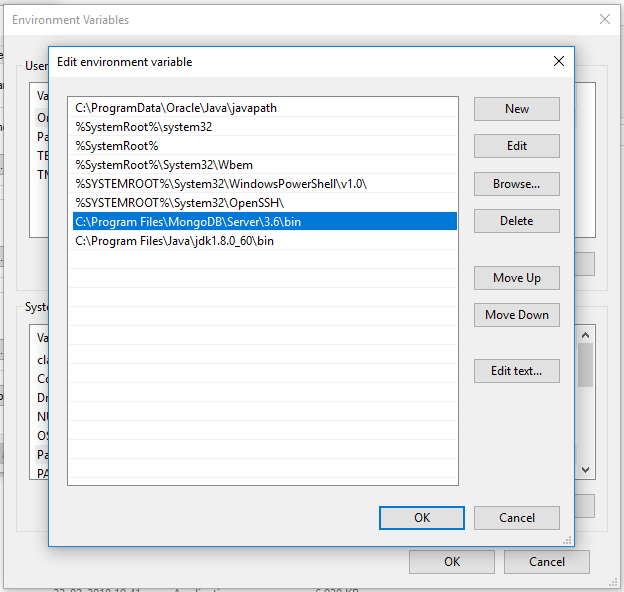
Step 1 : Set the MongoDB server path.



Step 2: We require a jar file to connect java with MongoDB. For this we have to create a new variable for classpath whose value will be the address of this jar file in the system variable of environment variable.



Step 3: We also have to put the path of java till bin folder in the path variable of the system variables of the environment variable.



Step 4: Write the java code and open the Mongo shell.

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

import com.mongodb.MongoClient;

import org.bson.Document;

public class insert

{

public static void main(String[] args)

{

MongoClient mongo = new MongoClient("localhost",27017);

System.out.println("Connected to the database successfully");

MongoDatabase database = mongo.getDatabase("TYITDB259664");

MongoCollection<Document> collection = database.getCollection("myCol");

System.out.println("Collection myCol selected successfully");

Document document = new Document();

document.append("id",1);

document.append("Name" , "Simran");

document.append("Roll no",259664);

document.append("Age",20);

document.append("College" , "Mcc");

collection.insertOne(document);

System.out.println("Document inserted successfully");

}

}

Step 5: Compile the java code and execute it.

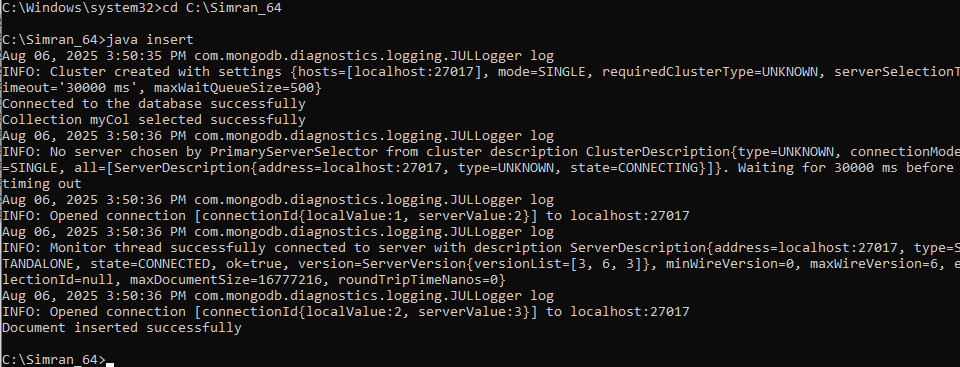
Step 6: To check the output we need to type the following commands in a Mongo shell.

6.1 show dbs : To check the database is created or not.

6.2 If the database is created use the database by using **use databasename.**

6.3 : Now we need to check collection is created or not using **show collections**.

6.4: To check document write **db.collection\_name. find().**

****

