**BIG DATA ANALYTICS**

**PRACTICAL - VIII**

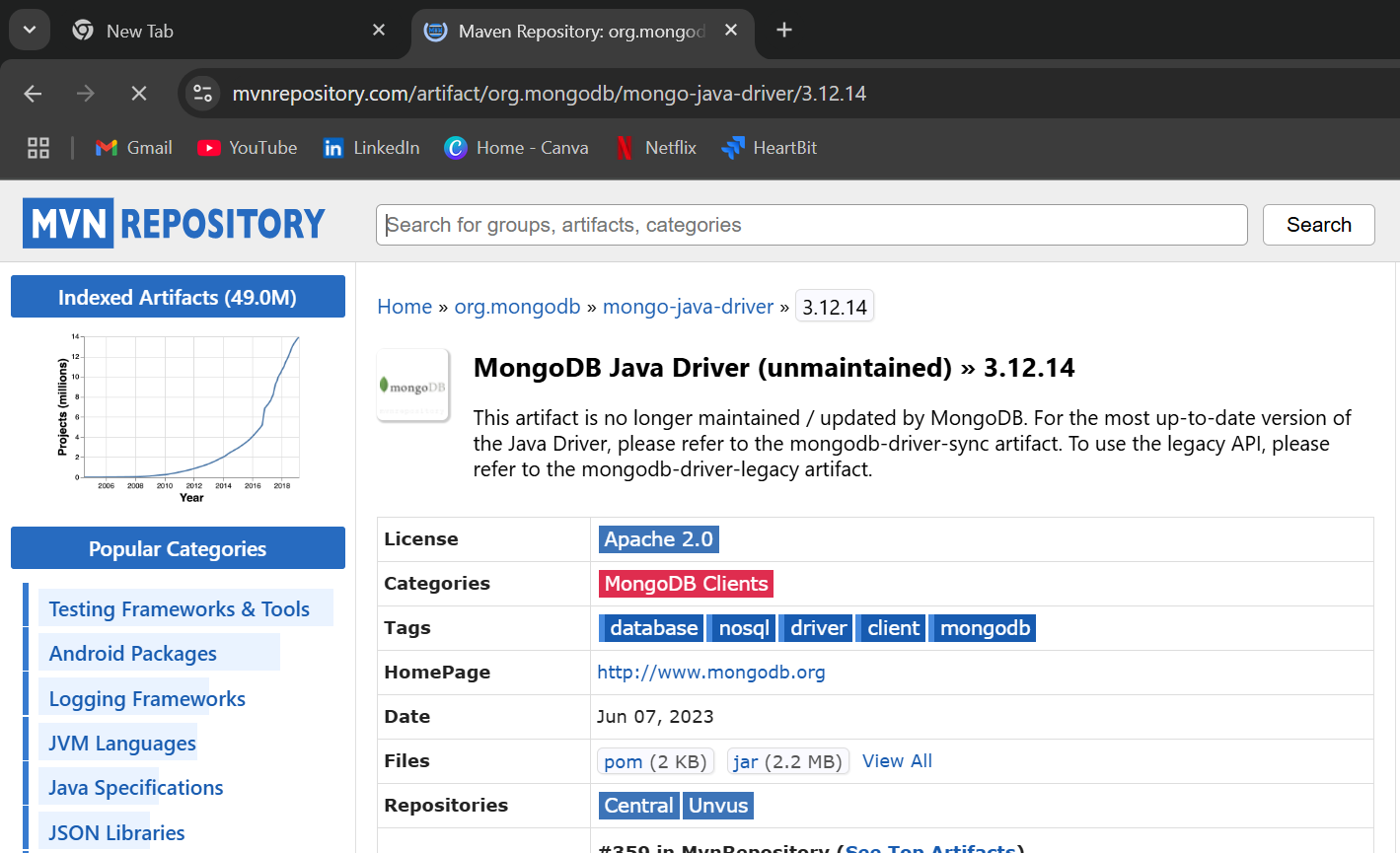
**NAME: MAHEK BUKELIYA ROLL NO: 008**

**DATE: 02/01/2025**

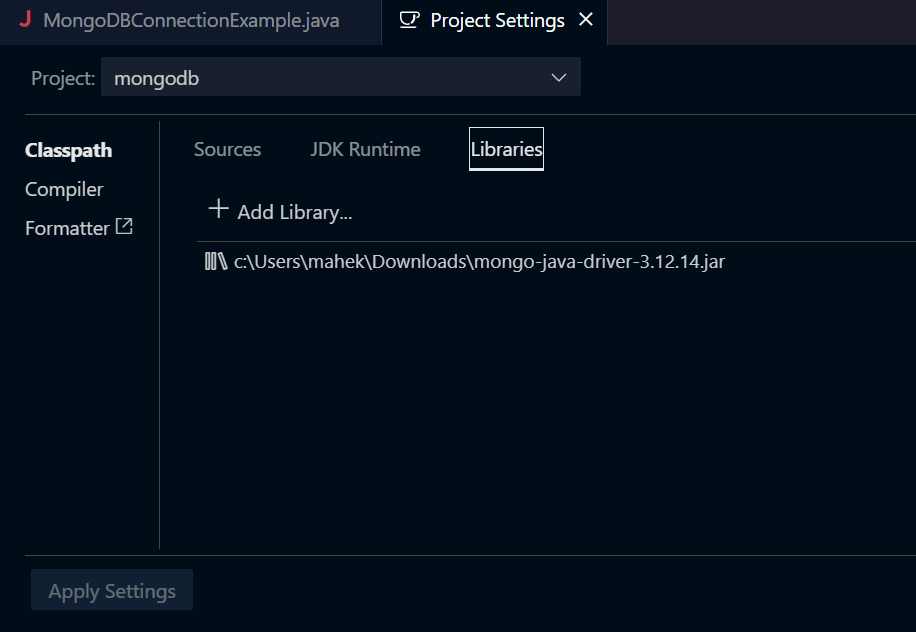
**TOPIC: MONDODB IN JAVA**

**SETTING UP MONGODB DRIVER FOR MONGODB IN JAVA**

**STEP I : *Install mongo-java-driver-3.12.14.jar file.***



**STEP II: *Import the jar file in the project classpath.***



**Q1] Make a connection in Mongodb using Java**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoDatabase;

import com.mongodb.client.MongoClient;

public class MongoDBConnectionExample {

public static void main(String[] args) {

String uri = "mongodb://localhost:27017";

// Create MongoClient instance using MongoClients

try (MongoClient mongoClient = MongoClients.create(uri)) {

// Access database

MongoDatabase database = mongoClient.getDatabase("mydb");

System.out.println("Connected to the database: " + database.getName());

} catch (Exception e) {

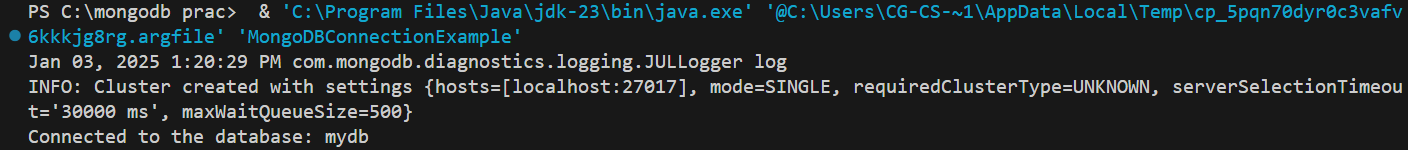
e.printStackTrace();

}

}

}

**OUTPUT:**



**Q2] Create new database ‘mynewdatabse’ and collection ‘books’ with fields book\_id, book\_title, author\_name, isbn\_no, no\_of\_pages, price and insert a document in it.**

**CODE:**

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

collection.insertOne(new Document("book\_id", 1)

.append("book\_title", "The White Tiger")

.append("author\_name", "Aravind Adiga")

.append("isbn\_no", "9788172238476")

.append("no\_of\_pages", 320)

.append("price\_in\_inr", 399.00));

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

} catch (Exception e) {

e.printStackTrace();

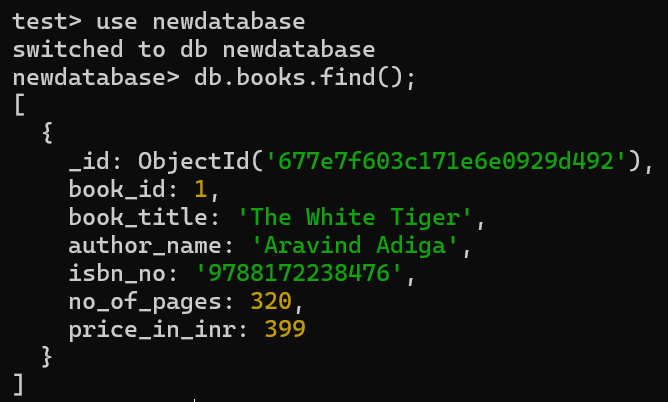
}

}

}

**OUTPUT:**





**Q3] Insert 10 documents in the collection books.**

**CODE:**

import com.mongodb.client.MongoClients;

import org.bson.Document;

import java.util.Arrays;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

// Create 10 book records

var books = Arrays.asList(

new Document("book\_id", 2)

.append("book\_title", "1984")

.append("author\_name", "George Orwell")

.append("isbn\_no", "9780451524935")

.append("no\_of\_pages", 328)

.append("price\_in\_inr", 1299.00),

new Document("book\_id", 3)

.append("book\_title", "To Kill a Mockingbird")

.append("author\_name", "Harper Lee")

.append("isbn\_no", "9780061120084")

.append("no\_of\_pages", 281)

.append("price\_in\_inr", 899.00),

new Document("book\_id", 4)

.append("book\_title", "The Catcher in the Rye")

.append("author\_name", "J.D. Salinger")

.append("isbn\_no", "9780316769488")

.append("no\_of\_pages", 277)

.append("price\_in\_inr", 599.00),

new Document("book\_id", 5)

.append("book\_title", "Moby-Dick")

.append("author\_name", "Herman Melville")

.append("isbn\_no", "9781503280786")

.append("no\_of\_pages", 585)

.append("price\_in\_inr", 1099.00),

new Document("book\_id", 6)

.append("book\_title", "The Alchemist")

.append("author\_name", "Paulo Coelho")

.append("isbn\_no", "9780061122415")

.append("no\_of\_pages", 208)

.append("price\_in\_inr", 499.00),

new Document("book\_id", 7)

.append("book\_title", "Shantaram")

.append("author\_name", "Gregory David Roberts")

.append("isbn\_no", "9780312330538")

.append("no\_of\_pages", 936)

.append("price\_in\_inr", 1399.00),

new Document("book\_id", 8)

.append("book\_title", "The Namesake")

.append("author\_name", "Jhumpa Lahiri")

.append("isbn\_no", "9780618485222")

.append("no\_of\_pages", 304)

.append("price\_in\_inr", 799.00),

new Document("book\_id", 9)

.append("book\_title", "Midnight’s Children")

.append("author\_name", "Salman Rushdie")

.append("isbn\_no", "9780099578512")

.append("no\_of\_pages", 463)

.append("price\_in\_inr", 999.00),

new Document("book\_id", 10)

.append("book\_title", "Train to Pakistan")

.append("author\_name", "Khushwant Singh")

.append("isbn\_no", "9780143065883")

.append("no\_of\_pages", 181)

.append("price\_in\_inr", 499.00)

);

// Insert the book records

collection.insertMany(books);

System.out.println("Multiple records inserted into the collection.");

} catch (Exception e) {

e.printStackTrace();

}

}

}

**OUTPUT:**



**Q4] Update a document in collection and set price as 599 where author name is ‘Khushwant Singh’**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.model.Filters;

import com.mongodb.client.model.Updates;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

// Update the document where author\_name is "Khushwant Singh"

var result = collection.updateOne(

Filters.eq("author\_name", "Khushwant Singh"),

Updates.set("price\_in\_inr", 599.00)

);

// Display the result of the update operation

if (result.getMatchedCount() > 0) {

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

} else {

System.out.println("No matching document found.");

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

**OUTPUT:**





**Q5] Update many documents of collection books increment price by Rs.50 where book title starts with T.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.model.Filters;

import com.mongodb.client.model.Updates;

public class MongoDBConnectionExample {

    public static *void* main(String[] *args*) {

        try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

            var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

            var result = collection.updateMany(

                Filters.regex("book\_title", "^T"),

                Updates.inc("price\_in\_inr", 50)

            );

            System.out.println(result.getModifiedCount() + " documents updated.");

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}

**OUTPUT:**



**Q6] Display all the documents in collection books.**

**CODE:**

import com.mongodb.client.MongoClients;

public class MongoDBConnectionExample {

        public static *void* main(String[] *args*) {

            try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

                var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

                for (var document : collection.find()) {

                    System.out.println(document.toJson());

                }

            } catch (Exception e) {

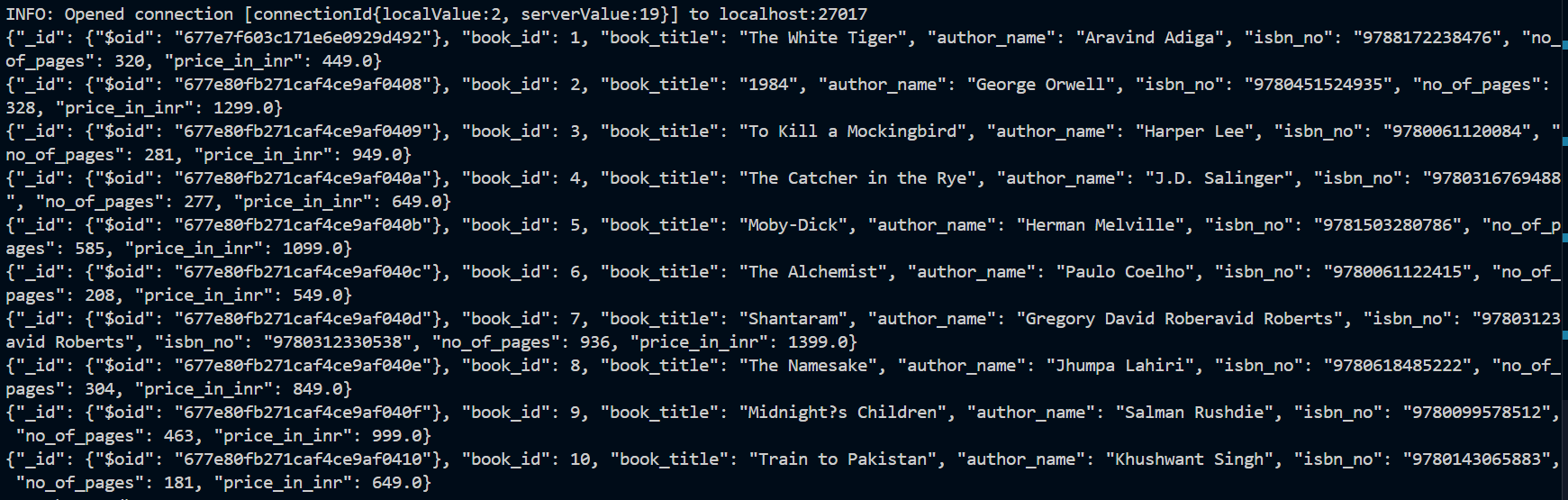
                e.printStackTrace();

            }

        }

    }

**OUTPUT:**



**Q7] Display only documents whose book title starts with letter ‘T’.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.model.Filters;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

for (var document : collection.find(Filters.regex("book\_title", "^T"))) {

System.out.println(document.toJson());

}

} catch (Exception e) {

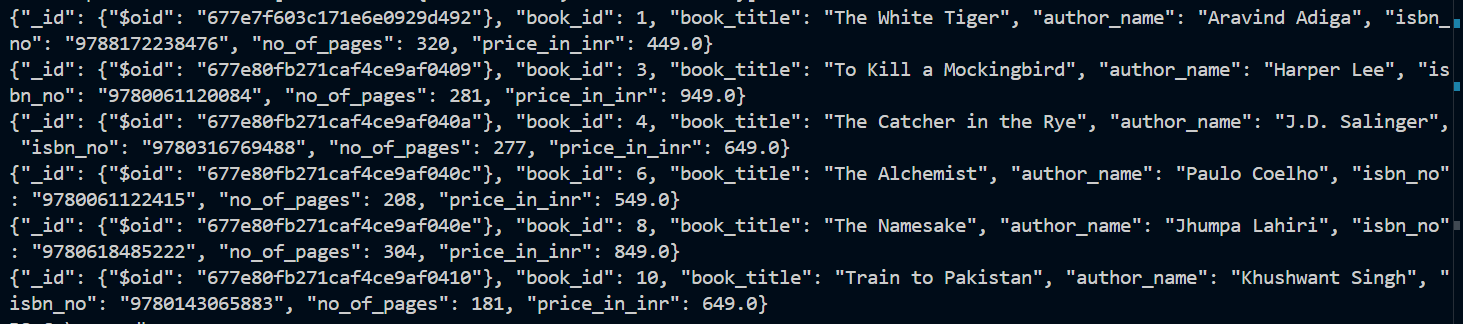
e.printStackTrace();

}

}

}

**OUTPUT:**

****

**Q8] Delete a document where the book pages are 181.**

CODE:  
import com.mongodb.client.MongoClients;

import com.mongodb.client.model.Filters;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

var result = collection.deleteOne(Filters.eq("no\_of\_pages", 181));

System.out.println(result.getDeletedCount() + " document(s) deleted.");

} catch (Exception e) {

e.printStackTrace();

}

}

}

**OUTPUT:**



**Q9] Delete multiple documents in the collection books where price between 100-400.**

**CODE:**import com.mongodb.client.MongoClients;

import com.mongodb.client.model.Filters;

import org.bson.Document;

public class MongoDBConnectionExample {

    public static *void* main(String[] *args*) {

        try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

            var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

            Document filter = new Document("price\_in\_inr",

                new Document("$gt", 100).append("$lt", 500));

            var result = collection.deleteMany(filter);

            System.out.println(result.getDeletedCount() + " document(s) deleted.");

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}

**OUTPUT:**



**Q10] Skip 2 document in collection books and display others documents.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoIterable;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

MongoIterable<org.bson.Document> documents = collection.find().skip(2);

for (var document : documents) {

System.out.println(document.toJson());

}

} catch (Exception e) {

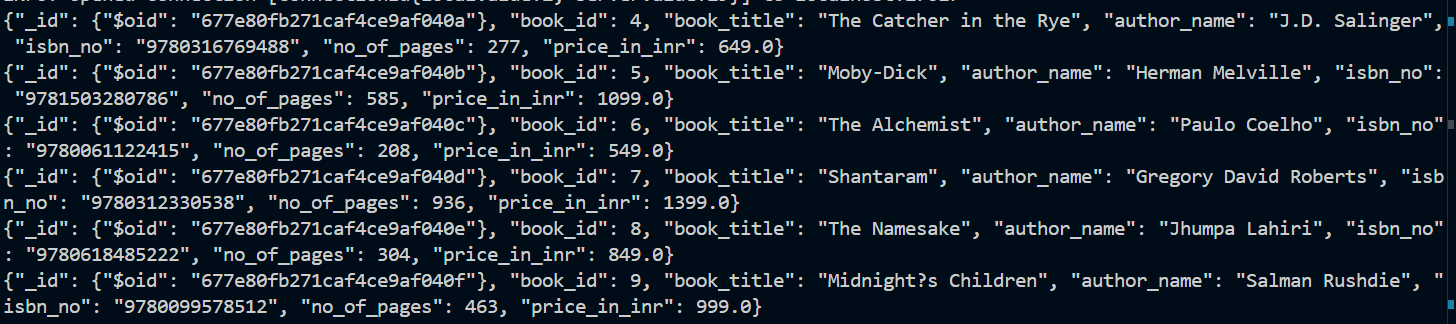
e.printStackTrace();

}

}

}

**OUTPUT:**



**Q11] Display the documents of book having price greater than 500.**

**CODE:**

import com.mongodb.client.MongoClients;

import org.bson.Document;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

Document filter = new Document("price\_in\_inr", new Document("$gt", 1000));

var documents = collection.find(filter);

for (var document : documents) {

System.out.println(document.toJson());

}

} catch (Exception e) {

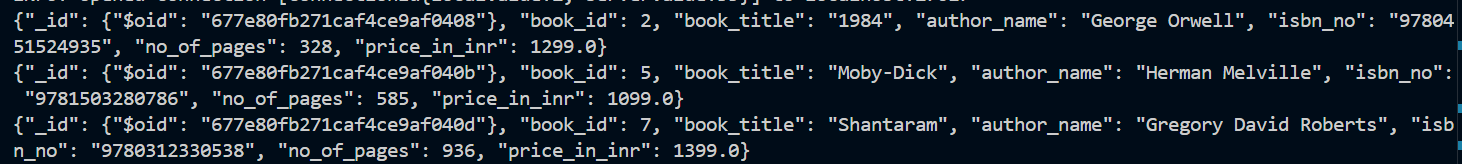
e.printStackTrace();

}

}

}

**OUTPUT:**



**Q12] Display the documents of book having price lesser than 500.**

**CODE:**

import com.mongodb.client.MongoClients;

import org.bson.Document;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

Document filter = new Document("price\_in\_inr", new Document("$lt", 800));

var documents = collection.find(filter);

for (var document : documents) {

System.out.println(document.toJson());

}

} catch (Exception e) {

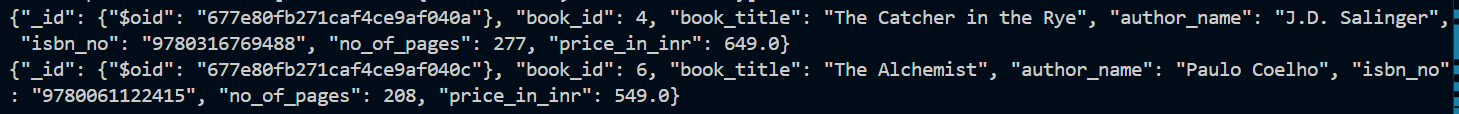
e.printStackTrace();

}

}

}

**OUTPUT:**



**Q13 Display only first 3 documents of collection book using limit.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.model.Filters;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

var documents = collection.find().limit(3);

for (var document : documents) {

System.out.println(document.toJson());

}

} catch (Exception e) {

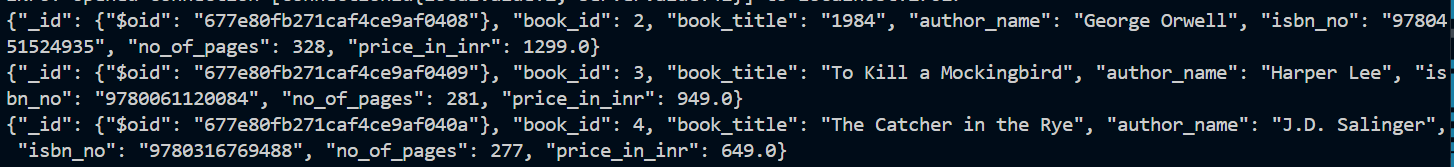
e.printStackTrace();

}

}

}

**OUTPUT:**



**Q14] Count the total number of documents in collection books.**

**CODE:**

import com.mongodb.client.MongoClients;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

long count = collection.countDocuments();

System.out.println("Total number of documents: " + count);

} catch (Exception e) {

e.printStackTrace();

}

}

}

**OUTPUT:**



**Q15] Count the total number of documents in collection books where book title starts with letter ‘M’.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.model.Filters;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

long count = collection.countDocuments(Filters.regex("book\_title", "^M"));

System.out.println("Total number of documents with title starting with 'M': " + count);

} catch (Exception e) {

e.printStackTrace();

}

}

}

**OUTPUT:**



**Q16] Check whether the document with book title ‘Midnight’s Children’ exists or not and display it.**

**CODE:**

import com.mongodb.client.MongoClients;

import com.mongodb.client.model.Filters;

public class MongoDBConnectionExample {

public static void main(String[] args) {

try (var mongoClient = MongoClients.create("mongodb://localhost:27017")) {

var collection = mongoClient.getDatabase("newdatabase").getCollection("books");

var filter = Filters.and(

Filters.exists("book\_title", true),

Filters.eq("book\_title", "Midnight’s Children")

);

var documents = collection.find(filter);

for (var document : documents) {

System.out.println(document.toJson());

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

**OUTPUT:**

