

```

import com.mongodb.client.*;
import com.mongodb.client.model.Updates;
import com.mongodb.client.result.DeleteResult;
import com.mongodb.client.result.InsertOneResult;
import com.mongodb.client.result.UpdateResult;
import org.bson.Document;
import org.bson.json.JsonWriterSettings;

import java.util.Arrays;
import java.util.Scanner;

import static com.mongodb.client.model.Filters.and;
import static com.mongodb.client.model.Filters.eq;

public class App {
    private static final MongoClient mongoClient;
    private static final MongoDBDatabase db;
    private static final MongoCollection<Document> collection;
    private static final JsonWriterSettings settings;
    private static final Scanner scanner;

    static {
        mongoClient = MongoClient.create("mongodb://localhost:27017");
        db = mongoClient.getDatabase("MockC01");
        collection = db.getCollection("MovieReview");
        settings = JsonWriterSettings.builder().indent(true).build();
        scanner = new Scanner(System.in);
    }

    public static void main(String[] args) {

        System.out.print("\nOperation: ");
        String choice = scanner.nextLine().toLowerCase();

        while (!choice.equalsIgnoreCase("exit")){

            switch (choice){
                case "insert":
                    insert();
                    break;
                case "update":
                    update();
                    break;
                case "delete":
                    delete();
                    break;
                case "find":
                    find();
            }
        }
    }

```

```

        break;
    default:
        System.out.println("Invalid Action");
    }

    System.out.print("\nOperation: ");
    choice = scanner.nextLine().toLowerCase();

}

}

private static void insert(){
    String name, movie, genre;
    int rating;
    System.out.print("Name: ");
    name = scanner.nextLine();
    System.out.print("Movie: ");
    movie = scanner.nextLine();
    System.out.print("Genre: ");
    genre = scanner.nextLine();
    System.out.print("Rating: ");
    rating = Integer.parseInt(scanner.nextLine());

    Document article = new Document();
    article.append("Name",name);
    article.append("Movie",movie);
    article.append("Genre",genre);
    article.append("Rating",rating);

    InsertOneResult result = collection.insertOne(article);
    System.out.println("result: "+result);

}

private static void update(){
    System.out.print("Name: ");
    String name = scanner.nextLine();

    System.out.print("Movie: ");
    String movie = scanner.nextLine();

    System.out.print("Genre: ");
    String genre = scanner.nextLine();

    System.out.print("New Rating: ");
    int rating = Integer.parseInt(scanner.nextLine());

    UpdateResult result = collection.updateOne(
        and(eq("Name",name),eq("Movie",movie),eq("Genre",genre)),

```

```

        Updates.set("Rating",rating)
    );
    System.out.println("result: "+result);
}

private static void delete(){

    System.out.print("Name: ");
    String name = scanner.nextLine();

    DeleteResult result = collection.deleteMany(
        eq("Name",name)
    );
    System.out.println("result: "+result);
}

private static void find(){
    for (Document document : collection.find()) {
        System.out.println(document.toJson(settings));
    }
}
}

```

OUTPUT

```

Operation: delete
Name: pratt
result: AcknowledgedDeleteResult{deletedCount=1}

Operation: Gold
Invalid Action

Operation: find

Operation: insert
Name: Prathamesh
Movie: Gold
Genre: Jingolism
Rating: 8
result: AcknowledgedInsertOneResult{insertedId=BsonObjectId{value=5fc73d45e859c26882fd9dda}}

Operation: update
Name: Prathamesh
Movie: Gold
Genre: Jingolism
New Rating: 4
result: AcknowledgedUpdateResult{matchedCount=1, modifiedCount=1, upsertedId=null}

Operation: find
{
  "_id": {
    "$oid": "5fc73d45e859c26882fd9dda"
  },
  "Name": "Prathamesh",
  "Movie": "Gold",
  "Genre": "Jingolism",
  "Rating": 4
}

Operation: delete
Name: Prathamesh
result: AcknowledgedDeleteResult{deletedCount=1}

Operation:

```