Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра системотехніки

Дисципліна: «СПЕЦІАЛІЗОВАНІ БАЗИ ДАНИХ (NOSQL)»

**ПРАКТИЧНЕ ЗАНЯТТЯ № 4**

**«МІГРАЦІЯ ДАНИХ МІЖ СУБД MONGODB І MYSQL»,**

|  |  |  |
| --- | --- | --- |
| Виконав:  ст. гр. ІТКН-18-2  Гуляев Д.С. |  | Прийняв:  Доц. каф. Хряпкін О.В. |

Харків 2020

1. **Мета роботи**

Розробити і реалізувати алгоритм міграції (перенесення) даних між СУБД MySQL і MongoDB. Отримати практичні навички: аналізу реляційної моделі даних і приведення її до документо-орієнтованого виду, і навпаки, створення на основі шаблону проектування DAO модуля конвертації даних між СУБД MySQL і MongoDB.

**2. Постановка задачі**

1. Розробити на основі шаблону проектування DAO набір методів взаємодії клієнтського програмного забезпечення і вибраних СУБД (MySQL і MongoDB).

2. Розробити метод (методи), який, на основі існуючого DAO, дозволяє повністю переносити дані між двома СУБД, в обидві сторони.

3. Вибрати в якості основної одну з СУБД і заповнити БД в ній тестовими даними (не менше 5 записів по кожній сутності).

4. Використовуючи розроблені методи міграції (перенесення) даних, здійснити конвертацію в другу СУБД.

5. Повторити конвертацію даних в зворотному напрямку і порівняти отримані результати.

**3. Міграція даних**

**Код:**

**migrationFromMySqlToMongo**

package Migration;  
  
import Models.Model;  
import Models.Monitor;  
import Models.Type;  
import com.mongodb.client.MongoClient;  
import com.mongodb.client.MongoClients;  
import com.mongodb.client.MongoCollection;  
import com.mongodb.client.MongoDatabase;  
import org.bson.Document;  
import org.bson.types.ObjectId;  
  
import java.sql.\*;  
import java.util.ArrayList;  
import java.util.List;  
  
public class migrationFromMySqlToMongo {  
  
 public void MigrationTypeFromMySqlToMongo(){  
 try{  
 Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/Monitors\_nosql?serverTimezone=UTC","root","Rocstar256157");  
 try{  
 PreparedStatement ps = connection.prepareStatement("SELECT \* FROM Type");  
 ResultSet rs = ps.executeQuery();  
 List<Document> lec = new ArrayList<>();  
 while (rs.next()){  
 Type type = new Type(  
 rs.getString("Name\_type"),  
 rs.getString("Features")  
 );  
 Document type\_lec = new Document("\_id", new ObjectId());  
 type\_lec.append("Name\_type", type.getName\_type())  
 .append("Features", type.getFeatures());  
 lec.add(type\_lec);  
 }  
 rs.close();  
  
 MongoClient mongoClient = MongoClients.*create*("mongodb://127.0.0.1:27017");  
 MongoDatabase DB = mongoClient.getDatabase("Monitors");  
 MongoCollection<Document> Collection = DB.getCollection("Type");  
  
 Collection.insertMany(lec);  
 }  
 catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void MigrationModelFromMySqlToMongo(){  
 try{  
 Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/Monitors\_nosql?serverTimezone=UTC","root","Rocstar256157");  
 try{  
 PreparedStatement ps = connection.prepareStatement("SELECT \* FROM Model");  
 ResultSet rs = ps.executeQuery();  
 List<Document> lec = new ArrayList<>();  
 while (rs.next()){  
 Model model = new Model(  
 rs.getString("Name\_company"),  
 rs.getString("Country"),  
 rs.getString("Name\_model")  
 );  
 Document model\_lec = new Document("\_id", new ObjectId());  
 model\_lec.append("Name\_company", model.getName\_company())  
 .append("Country", model.getCountry())  
 .append("Name\_model", model.getName\_model());  
 lec.add(model\_lec);  
 }  
 rs.close();  
  
 MongoClient mongoClient = MongoClients.*create*("mongodb://127.0.0.1:27017");  
 MongoDatabase DB = mongoClient.getDatabase("Monitors");  
 MongoCollection<Document> Collection = DB.getCollection("Model");  
  
 Collection.insertMany(lec);  
 }  
 catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void MigrationMonitorFromMySqlToMongo(){  
 try{  
 Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/Monitors\_nosql?serverTimezone=UTC","root","Rocstar256157");  
 try{  
 PreparedStatement ps = connection.prepareStatement("SELECT \* FROM Monitor, Model, Type WHERE Monitor.id\_type = Type.id AND Monitor.id\_model = Model.id");  
 ResultSet rs = ps.executeQuery();  
 List<Document> lec = new ArrayList<>();  
 while (rs.next()){  
 Type type = new Type(  
 rs.getString("Name\_type"),  
 rs.getString("Features")  
 );  
 Model model = new Model(  
 rs.getString("Name\_company"),  
 rs.getString("Country"),  
 rs.getString("Name\_model")  
 );  
 Monitor monitor = new Monitor(  
 rs.getString("Name"),  
 rs.getString("Matrix\_type"),  
 rs.getString("Aspect\_ratio"),  
 rs.getInt("Diagonal\_matrix"),  
 rs.getInt("Screen\_resolution"),  
 rs.getInt("Screen\_refresh\_rate"),  
 type,  
 model  
 );  
  
 Document setData\_Type = new Document("\_id", new ObjectId());  
 setData\_Type.append("Name\_type", monitor.Type.getName\_type())  
 .append("Features", monitor.Type.getFeatures());  
  
 Document setData\_Model = new Document("\_id", new ObjectId());  
 setData\_Model.append("Name\_company", monitor.Model.getName\_company())  
 .append("Country", monitor.Model.getCountry())  
 .append("Name\_model", monitor.Model.getName\_model());  
  
 Document monitor\_lec = new Document("\_id", new ObjectId());  
 monitor\_lec.append("Name", monitor.getName())  
 .append("Matrix\_type", monitor.getMatrix\_type())  
 .append("Aspect\_ratio", monitor.getAspect\_ratio())  
 .append("Diagonal\_matrix", monitor.getDiagonal\_matrix())  
 .append("Screen\_resolution", monitor.getScreen\_resolution())  
 .append("Screen\_refresh\_rate", monitor.getScreen\_refresh\_rate())  
 .append("Type", setData\_Type)  
 .append("Model", setData\_Model);  
  
 lec.add(monitor\_lec);  
 }  
 rs.close();  
  
 MongoClient mongoClient = MongoClients.*create*("mongodb://127.0.0.1:27017");  
 MongoDatabase DB = mongoClient.getDatabase("Monitors");  
 MongoCollection<Document> Collection = DB.getCollection("Monitor");  
  
 Collection.insertMany(lec);  
 }  
 catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
}

**migrationFromMongoToSql**

package Migration;  
  
import Models.Type;  
import com.mongodb.client.MongoClient;  
import com.mongodb.client.MongoClients;  
import com.mongodb.client.MongoCollection;  
import com.mongodb.client.MongoDatabase;  
import org.bson.Document;  
import org.bson.types.ObjectId;  
  
import java.sql.\*;  
import java.util.ArrayList;  
import java.util.List;  
  
public class migrationFromMongoToSql {  
  
 public void MigrationTypeFromMongoToMySql(){  
 try {  
 MongoClient mongoClient = MongoClients.*create*("mongodb://127.0.0.1:27017");  
 MongoDatabase DB = mongoClient.getDatabase("Monitors");  
 MongoCollection<Document> Collection = DB.getCollection("Type");  
 List<Document> typeList = Collection.find().into(new ArrayList<>());  
 for (Document type : typeList) {  
 try {  
 Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/Monitors\_nosql?serverTimezone=UTC","root","Rocstar256157");  
 PreparedStatement ps = connection.prepareStatement("INSERT INTO Type (Name\_type, Features) VALUES (?, ?)");  
 ps.setString(1, type.getString("Name\_type"));  
 ps.setString(2,type.getString("Features"));  
 ps.executeUpdate();  
 }  
 catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void MigrationModelFromMongoToMySql(){  
 try {  
 MongoClient mongoClient = MongoClients.*create*("mongodb://127.0.0.1:27017");  
 MongoDatabase DB = mongoClient.getDatabase("Monitors");  
 MongoCollection<Document> Collection = DB.getCollection("Model");  
 List<Document> modelList = Collection.find().into(new ArrayList<>());  
 for (Document model : modelList) {  
 try {  
 Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/Monitors\_nosql?serverTimezone=UTC","root","Rocstar256157");  
 PreparedStatement ps = connection.prepareStatement("INSERT INTO Model (Name\_company, Country, Name\_model) VALUES (?, ?, ?)");  
 ps.setString(1, model.getString("Name\_company"));  
 ps.setString(2,model.getString("Country"));  
 ps.setString(3,model.getString("Name\_model"));  
 ps.executeUpdate();  
 }  
 catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void MigrationMonitorFromMongoToMySql(){  
 try {  
 MongoClient mongoClient = MongoClients.*create*("mongodb://127.0.0.1:27017");  
 MongoDatabase DB = mongoClient.getDatabase("Monitors");  
 MongoCollection<Document> Collection = DB.getCollection("Monitor");  
 List<Document> monitorList = Collection.find().into(new ArrayList<>());  
 for (Document monitor : monitorList) {  
 try {  
 Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/Monitors\_nosql?serverTimezone=UTC","root","Rocstar256157");  
 PreparedStatement ps = connection.prepareStatement("INSERT INTO Monitor (Name, Matrix\_type, Aspect\_ratio, Diagonal\_matrix, Screen\_resolution, Screen\_refresh\_rate, id\_type, id\_model) VALUES (?, ?, ?, ?, ?, ?, ?, ?)");  
 PreparedStatement ps1 = connection.prepareStatement("SELECT \* FROM Type WHERE Type.Name\_type " + " = ?");  
 ps1.setString(1, monitor.getString("Name\_type"));  
 ResultSet rs1 = ps1.executeQuery();  
 PreparedStatement ps2 = connection.prepareStatement("SELECT \* FROM Model WHERE Model.Name\_model " + " = ?");  
 ps2.setString(1, monitor.getString("Name\_model"));  
 ResultSet rs2 = ps2.executeQuery();  
 ps.setString(1, monitor.getString("Name"));  
 ps.setString(2, monitor.getString("Matrix\_type"));  
 ps.setString(3, monitor.getString("Aspect\_ratio"));  
 ps.setInt(4, monitor.getInteger("Diagonal\_matrix"));  
 ps.setInt(5, monitor.getInteger("Screen\_resolution"));  
 ps.setInt(6, monitor.getInteger("Screen\_refresh\_rate"));  
 ps.setInt(7, rs1.getInt("id"));  
 ps.setInt(8, rs2.getInt("id"));  
 ps.executeUpdate();  
 }  
 catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
}

**Main**

//MigrationFromMySqlToMongoDB  
new Migration.migrationFromMySqlToMongo().MigrationTypeFromMySqlToMongo();  
new Migration.migrationFromMySqlToMongo().MigrationModelFromMySqlToMongo();  
new Migration.migrationFromMySqlToMongo().MigrationMonitorFromMySqlToMongo();  
  
//MigrationFromMongoDbToMySql  
new Migration.migrationFromMongoToSql().MigrationTypeFromMongoToMySql();  
new Migration.migrationFromMongoToSql().MigrationModelFromMongoToMySql();  
new Migration.migrationFromMongoToSql().MigrationMonitorFromMongoToMySql();

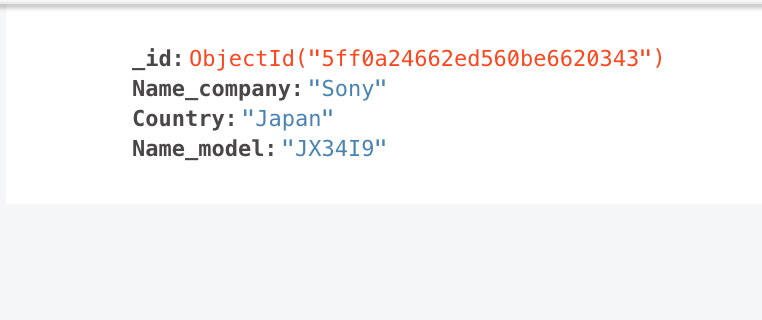


Рисунок 1 – Стан сутності «Model» до міграції

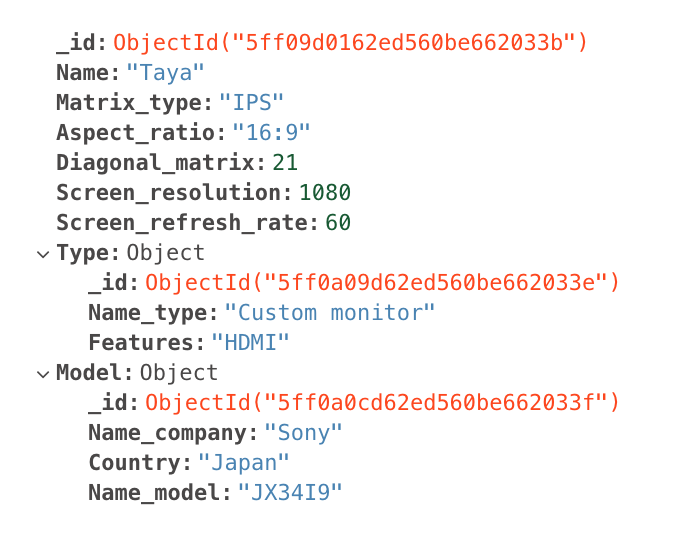


Рисунок 2 – Стан сутності «Monitor» до міграції



Рисунок 3 – Стан сутності «Type» до міграції

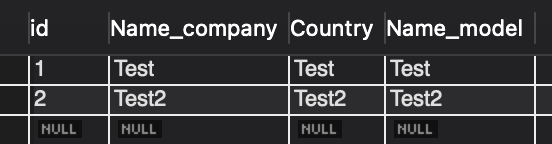


Рисунок 4 – Стан сутності «Model» до міграції

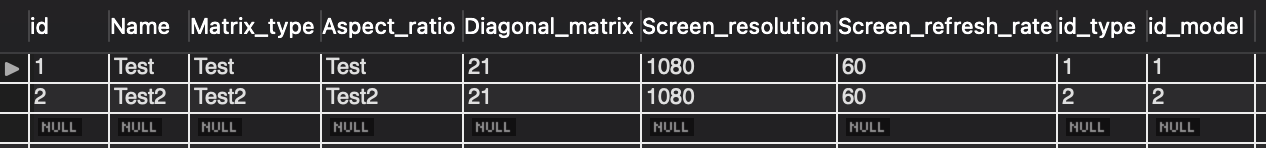


Рисунок 5 – Стан сутності «Monitor» до міграції

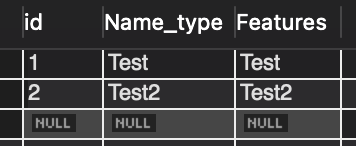


Рисунок 6 – Стан сутності «Type» до міграції

Стан сутностей після міграції даних.



Рисунок 7 – Стан сутності «Model» після міграції

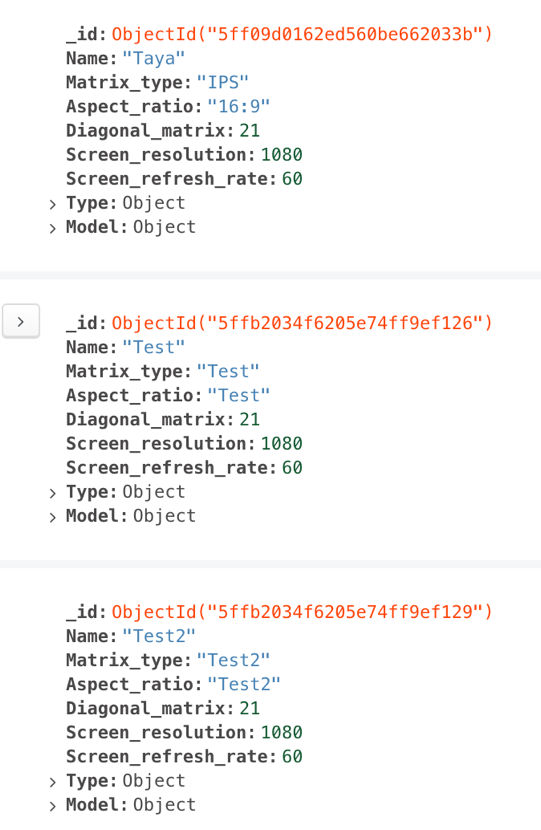


Рисунок 8 – Стан сутності «Monitor» після міграції



Рисунок 9 – Стан сутності «Type» до міграції



Рисунок 10 – Стан сутності «Model» після міграції

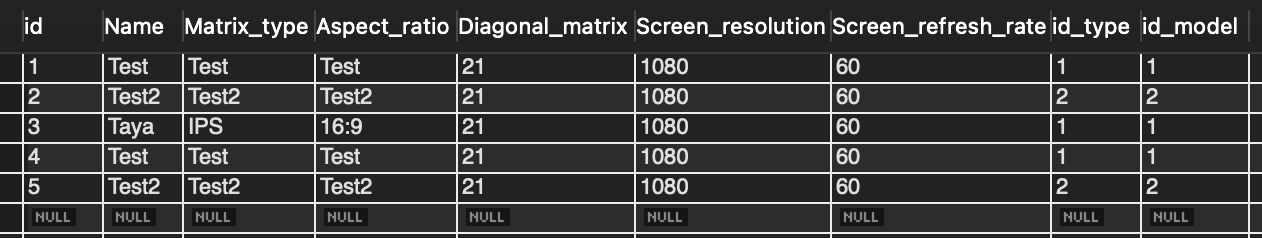


Рисунок 11 – Стан сутності «Monitor» після міграції

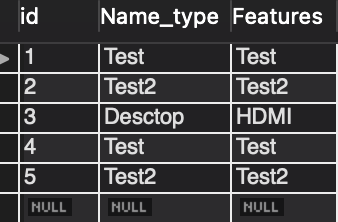


Рисунок 12 – Стан сутності «Type» після міграції

**5. Висновки**

В ході виконання лабораторної роботи був реалізований програмний засіб, що містить алгоритм міграції даних між СУБД MySQL і MongoDB. Для цього було проаналізовано моделі даних, та створено функції для конвертації даних з одного типу на інший та навпаки.