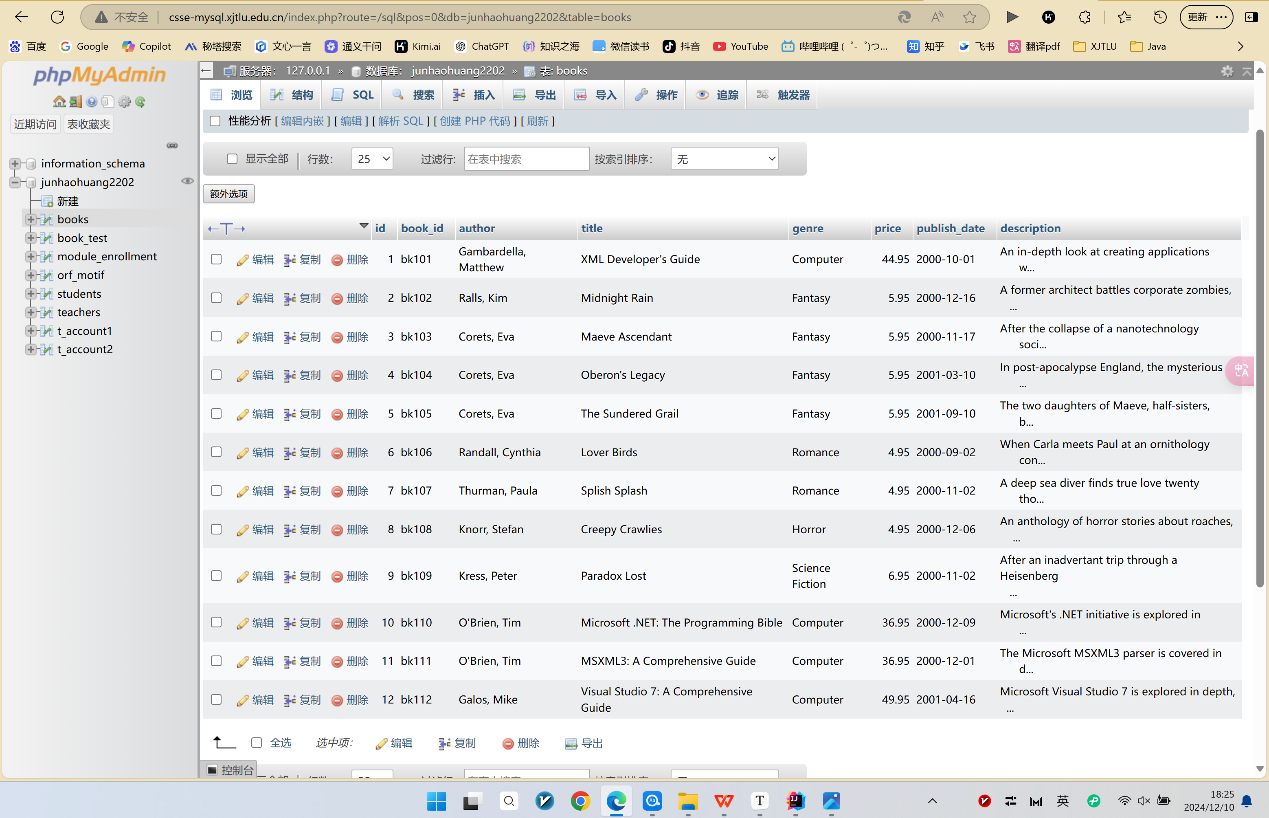
The result of the database



The code can successfully get the result above:

package Lab6;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import java.io.File;

import java.sql.\*;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

import java.util.ArrayList;

import java.util.LinkedHashMap;

public class LabSix {

// initiate database information

static final String JDBC\_DRIVER = "com.mysql.cj.jdbc.Driver";

//replace the following three parameter values with your own ones

static final String DB\_URL = "jdbc:mysql://10.7.1.127/junhaohuang2202";

static final String USER = "JunhaoHuang2202";

static final String PASS = "123";

public ArrayList<LinkedHashMap<String, Object>> extractInfo(String filepath) {

ArrayList<LinkedHashMap<String, Object>> list = new ArrayList<>();

try {

File inputFile = new File("src/Lab6/books.xml");

DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();

DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

Document doc = dBuilder.parse(inputFile);

doc.getDocumentElement().normalize();

System.out.println("Root element :" + doc.getDocumentElement().getNodeName());

NodeList nList = doc.getElementsByTagName("book");

// get each book info

for (int temp = 0; temp < nList.getLength(); temp++) {

LinkedHashMap<String, Object> entity = new LinkedHashMap<>();

Node nNode = nList.item(temp);

// print book

// System.out.println("\nCurrent Element:" + nNode.getNodeName() + temp);

if (nNode.getNodeType() == Node.ELEMENT\_NODE) {

Element eElement = (Element) nNode;

// get book\_id

String book\_id = eElement.getAttribute("id");

// get author

String author = eElement.getElementsByTagName("author").item(0).getTextContent();

// get title

String title = eElement.getElementsByTagName("title").item(0).getTextContent();

// get genre

String genre = eElement.getElementsByTagName("genre").item(0).getTextContent();

// get price

String price = eElement.getElementsByTagName("price").item(0).getTextContent();

// get date

String date = eElement.getElementsByTagName("publish\_date").item(0).getTextContent();

// get description

String description = eElement.getElementsByTagName("description").item(0).getTextContent();

// add to map

entity.put("book\_id", book\_id);

entity.put("author", author);

entity.put("title", title);

entity.put("genre", genre);

entity.put("price", price);

entity.put("date", date);

entity.put("description", description);

// 讲当前的entity添加到arraylist当中

list.add(entity);

}

}

} catch (Exception e) {

e.printStackTrace();

}

return list;

}

public void databaseOperation(ArrayList<LinkedHashMap<String, Object>> list){

Connection conn;

Statement stmt;

ResultSet rs;

String sql;

String first;

long begin;

long end;

try {

Class.forName(JDBC\_DRIVER);

conn = DriverManager.getConnection(DB\_URL, USER, PASS);

if (!conn.isClosed()) {

System.out.println("Succeeded connecting to the Database.");

}

stmt = conn.createStatement();

//prepare query without an index

System.out.println("Start to insert values into the database");

for (int i = 0; i < list.size(); i++) {

LinkedHashMap<String, Object> entity = list.get(i);

try {

String priceStr = (String) entity.get("price");

String dateStr = (String) entity.get("date");

String bookId = (String) entity.get("book\_id");

String author = (String) entity.get("author");

String title = (String) entity.get("title");

String genre = (String) entity.get("genre");

String description = (String) entity.get("description");

// check null value

if (priceStr == null || dateStr == null || bookId == null || author == null || title == null || genre == null || description == null) {

System.out.println("Skipping incomplete data: " + entity);

continue;

}

// convert float and date

float price = Float.parseFloat(priceStr);

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");

LocalDate localDate = LocalDate.parse(dateStr, formatter);

java.sql.Date sqlDate = java.sql.Date.valueOf(localDate);

// 构建 SQL

sql = "INSERT INTO books (book\_id, author, title, genre, price, publish\_date, description) VALUES (?, ?, ?, ?, ?, ?, ?)";

try (PreparedStatement pstmt = conn.prepareStatement(sql)) {

pstmt.setString(1, bookId);

pstmt.setString(2, author);

pstmt.setString(3, title);

pstmt.setString(4, genre);

pstmt.setFloat(5, price);

pstmt.setDate(6, sqlDate);

pstmt.setString(7, description);

int affectedRows = pstmt.executeUpdate();

System.out.println("Rows affected: " + affectedRows);

}

} catch (SQLException e) {

System.err.println("SQL error: " + e.getMessage());

e.printStackTrace();

} catch (Exception e) {

System.err.println("Error: " + e.getMessage());

e.printStackTrace();

}

}

System.out.println("End query.");

conn.close();

//close database connections.

conn.close();

} catch (ClassNotFoundException e) {

System.out.println("Cannot find JDBC Driver!");

e.printStackTrace();

} catch (SQLException e) {

e.printStackTrace();

} catch (Exception e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

LabSix labSix = new LabSix();

ArrayList<LinkedHashMap<String, Object>> result = labSix.extractInfo("books.xml");

for (LinkedHashMap<String, Object> entity : result) {

System.out.printf((String) entity.get("book\_id"));

}

labSix.databaseOperation(result);

}

}

The screen shot of terminal output

