《数据库概论》实验三 使用高级程序设计语言访问数据库 实验报告

刘潇远 161220083 161220083@smail.nju.edu.cn/courseralxy@qq.com

实验环境

操作系统：Windows10

数据库版本：MySQL8.0.12

IDE：Eclipse 2018-09

mysql-connector-java-5.1.47

commons-pools-2.6.0

commons-logging-1.2

commons-dbcp2-2.5.0

实验过程

**实验内容1**

**import** java.sql.\*;

**public** **class** DBCon {

//输出表格

**public** **static** **void** output(ResultSet worths) **throws** SQLException {

ResultSetMetaData data= worths.getMetaData();

String output1=**new** String();

**for**(**int** i = 1; i <= data.getColumnCount();i++)

output1+=data.getColumnName(i)+"\t";

System.***out***.println(output1);

String output2=**new** String();

**while**(worths.next()) {

**for**(**int** i = 1; i <= data.getColumnCount();i++)

output2+=worths.getString(i)+"\t";

output2+="\n";

}

System.***out***.println(output2);

}

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

String driverName="com.mysql.jdbc.Driver";

String dbURL="jdbc:mysql://localhost:3306/lab2";

String userName="root";

String userPwd="1234";

Connection dbConn = **null**;

**try** {

Class.*forName*(driverName);

dbConn=DriverManager.*getConnection*(dbURL,userName,userPwd);

System.***out***.println("连接数据库成功");

//dbConn.close();

} **catch**(Exception e) {

e.printStackTrace();

System.***out***.print("连接失败");

}

System.***out***.println("初始状态");

Statement execStat = dbConn.createStatement();

//查询并输出

ResultSet worths = execStat.executeQuery("SELECT \* FROM STUDENT");

DBCon.*output*(worths);

//插入操作，输出结果

System.***out***.println("插入操作");

execStat.executeUpdate(

"INSERT INTO "

+ "STUDENT(NAME,SEX,ID,AGE,DID) "

+ "VALUES "

+ "('乌蝇哥','男',0,40,2),"

+ "('非凡哥','男',1,42,3),"

+ "('醒哥','男',2,35,4);");

worths = execStat.executeQuery("SELECT \* FROM STUDENT");

DBCon.*output*(worths);

//修改操作，输出结果

System.***out***.println("修改操作");

execStat.executeUpdate(

"UPDATE "

+ "STUDENT "

+ "SET "

+ "NAME = '刘醒' "

+ "WHERE NAME = '醒哥';");

execStat.executeUpdate(

"UPDATE "

+ "STUDENT "

+ "SET "

+ "NAME = '梁非凡' "

+ "WHERE NAME = '非凡哥';");

worths = execStat.executeQuery("SELECT \* FROM STUDENT");

DBCon.*output*(worths);

//删除操作，输出结果

System.***out***.println("删除操作");

execStat.executeUpdate(

"DELETE FROM "

+ "STUDENT "

+ "WHERE "

+ "ID = 0 OR "

+ "ID = 1 OR "

+ "ID = 2;");

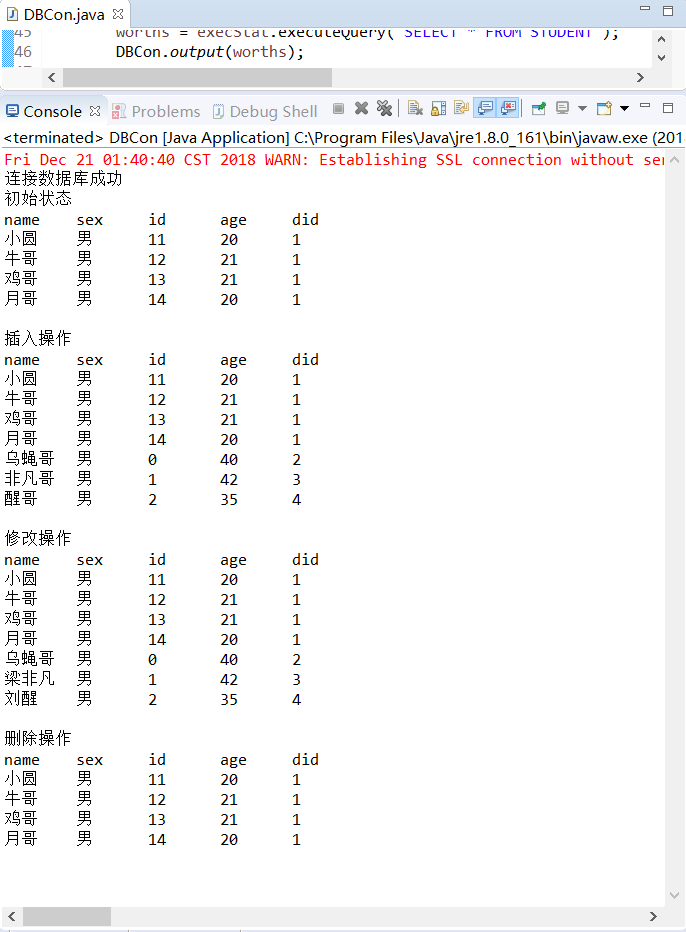
worths = execStat.executeQuery("SELECT \* FROM STUDENT");

DBCon.*output*(worths);

}

}

**程序运行截图**



**实验内容2**

实验2中有三个java文件，分别是**DBConn.java**，**DBCPConn.java**和**DBCPTest.java**。前两个分别用来将建立直接连接模式连接和连接池模式连接，最后一个用来进行测试。

在本实验中，仍然使用实验内容1中使用的数据库和表，通过一个for循环插入两千条数据。另外考虑到直接连接模式可能会有两种情况，一种是插入数据的过程始终使用同一条连接，另一种是每次插入数据都使用一条新连接，因此一共有三个对比测试。

**DBConn.java文件内容如下**。之所以注释掉实验内容1中的输出是因为输入输出占用了所需时间较长，会影响直接连接模式的运行时间/

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**public** **class** DBConn {

**static** String *driverName*="com.mysql.jdbc.Driver";

**static** String *dbURL*="jdbc:mysql://localhost:3306/lab2";

**static** String *userName*="root";

**static** String *userPwd*="1234";

**private** **static** Connection *dbConn* = **null**;

//获取一个数据库连接

**public** **static** Connection getConnection() {

**try** {

Class.*forName*(*driverName*);

*dbConn*=DriverManager.*getConnection*(*dbURL*,*userName*,*userPwd*);

//System.out.println("连接数据库成功");

//dbConn.close();

} **catch**(Exception e) {

e.printStackTrace();

//System.out.print("连接失败");

}

**return** *dbConn*;

}

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

*getConnection*();

}

}

DBCPConn.java内容如下

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.SQLException;

**import** java.util.Properties;

**import** javax.sql.DataSource;

**import** org.apache.commons.dbcp2.BasicDataSourceFactory;

**public** **class** DBCPConn {

**private** **static** Properties *properties* = **new** Properties();

**private** **static** DataSource *dataSource*;

//加载DBCP配置文件。static在forName时执行

**static**{

**try**{

FileInputStream is = **new** FileInputStream("config/dbcp.properties");

*properties*.load(is);

}**catch**(IOException e){

e.printStackTrace();

}

**try**{

*dataSource* = BasicDataSourceFactory.*createDataSource*(*properties*);

}**catch**(Exception e){

e.printStackTrace();

}

}

//从连接池中获取一个连接

**public** **static** Connection getConnection(){

Connection connection = **null**;

**try**{

connection = *dataSource*.getConnection();

}**catch**(SQLException e){

e.printStackTrace();

}

**try** {

connection.setAutoCommit(**false**);

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** connection;

}

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

*getConnection*();

System.***out***.println("Finish!");

}

}

在这个文件中，加载了**配置文件dbcp.properties**，里面描述了连接池的配置参数，内容如下

########DBCP配置文件##########

#驱动名

driverClassName=com.mysql.jdbc.Driver

#url

url=jdbc:mysql://localhost:3306/lab2

#用户名

username=root

#密码

password=1234

#初试连接数

initialSize=30

#最大活跃数

maxTotal=30

#最大idle数

maxIdle=10

#最小idle数

minIdle=5

#最长等待时间(毫秒)

maxWaitMillis=1000

#程序中的连接不使用后是否被连接池回收(该版本要使用removeAbandonedOnMaintenance和removeAbandonedOnBorrow)

#removeAbandoned=true

removeAbandonedOnMaintenance=true

removeAbandonedOnBorrow=true

#连接在所指定的秒数内未使用才会被删除(秒)(为配合测试程序才配置为1秒)

removeAbandonedTimeout=1

**DECPTest.java内容如下**

**import** java.sql.Connection;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** DBCPTest {

//清空数据表.在每次测试前执行

**public** **void** initialize() **throws** Exception{

**for**(**int** i = 0; i < 2000; i++){

initialize(i);

}

System.***out***.println("DONE");

}

**public** **void** initialize(**int** id) {

String sql = "INSERT INTO "

+ "STUDENT "

+ "values "

+ "('测试',"

+ "'男',"

+ id

+",99,1)";

Connection conn = DBConn.*getConnection*();

**try**{

Statement stat = conn.createStatement();

stat.executeUpdate(sql);

}**catch**(Exception e){

e.printStackTrace() ;

}**finally**{

**try** {

conn.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

//直接连接模式，每次写数据都需要创建新连接

**public** **void** DBConnWrite1Test() **throws** Exception{

**for**(**int** i = 0; i < 2000; i++){

DBConnWrite1(i);

}

System.***out***.println("DONE");

}

**public** **void** DBConnWrite1(**int** id){

//String sql = "insert into dbcp values (" + id + ")";

String sql = "INSERT INTO "

+ "STUDENT "

+ "values "

+ "('测试',"

+ "'男',"

+ id

+",99,1)";

Connection conn = DBConn.*getConnection*();

**try**{

Statement stat = conn.createStatement();

stat.executeUpdate(sql);

}**catch**(Exception e){

e.printStackTrace() ;

}**finally**{

**try** {

conn.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

//连接池模式

**public** **void** DBCPConnWriteTest() **throws** Exception{

**for**(**int** i = 0; i < 2000; i++){

DBCPConnWrite(i);

}

System.***out***.println("DONE");

}

**public** **void** DBCPConnWrite(**int** id){

String sql = "INSERT INTO "

+ "STUDENT "

+ "values "

+ "('测试',"

+ "'男',"

+ id

+",99,1)";

**try** {

Connection conn = DBCPConn.*getConnection*();

Statement stat = conn.createStatement();

stat.executeUpdate(sql);

conn.commit();

conn.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

//直接连接模式，所有数据插入都使用同一条连接

**public** **void** DBConnWrite2Test() **throws** Exception{

Connection conn = DBConn.*getConnection*();

Statement stat = conn.createStatement();

**for**(**int** i = 0; i < 2000; i++){

DBConnWrite2(i, stat);

}

conn.close();

System.***out***.println("DONE");

}

**public** **void** DBConnWrite2(**int** id, Statement stat){

String sql = "INSERT INTO "

+ "STUDENT "

+ "values "

+ "('测试',"

+ "'男',"

+ id

+",99,1)";

**try**{

stat.executeUpdate(sql);

}**catch**(Exception e){

e.printStackTrace() ;

}

}

**public** **void** DBCPDelete() {

}

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

DBCPTest dbt = **new** DBCPTest();

**try** {

dbt.initialize();

} **catch** (Exception e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

**long** startMili1=System.*currentTimeMillis*();// 开始时间1

**try** {

dbt.DBCPConnWriteTest();

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**long** endMili1=System.*currentTimeMillis*();// 结束时间1

**try** {

dbt.initialize();

} **catch** (Exception e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

**long** startMili2=System.*currentTimeMillis*();// 开始时间2

**try** {

dbt.DBConnWrite1Test();

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**long** endMili2=System.*currentTimeMillis*();// 结束时间2

**try** {

dbt.initialize();

} **catch** (Exception e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

**long** startMili3=System.*currentTimeMillis*();// 开始时间3

**try** {

dbt.DBConnWrite2Test();

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**long** endMili3=System.*currentTimeMillis*();//结束时间3

**try** {

dbt.initialize();

} **catch** (Exception e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

System.***out***.println("连接池模式总耗时为："+(endMili1-startMili1)+"毫秒");

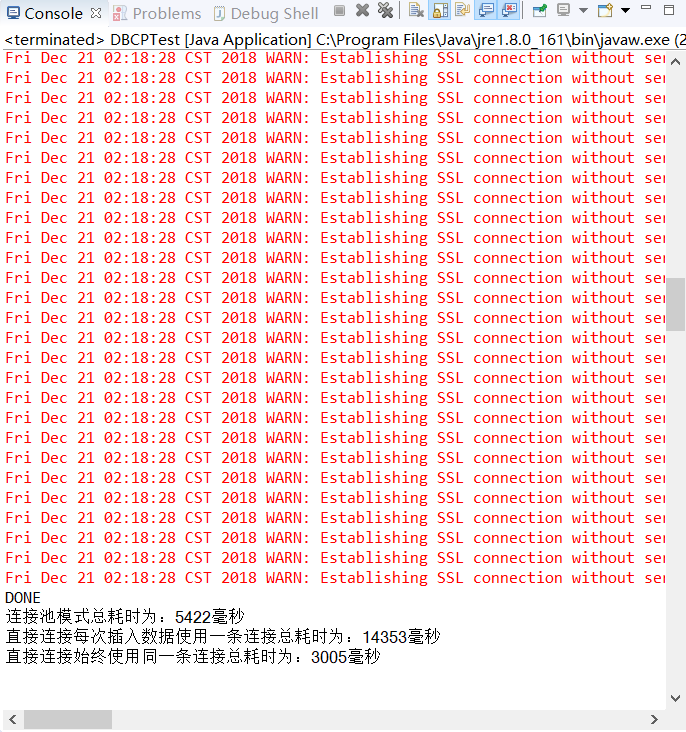
System.***out***.println("直接连接每次插入数据使用一条连接总耗时为："+(endMili2-startMili2)+"毫秒");

System.***out***.println("直接连接始终使用同一条连接总耗时为："+(endMili3-startMili3)+"毫秒");

}

}

**程序运行结果如下**



**实验结论**

可见连接池模式的效率不如只使用一条连接的直接连接模式，由于建立连接需要时间，所以每次插入数据都使用一条连接的直接连接模式用时是最长的。

实验中遇到的困难及解决办法

实验的困难主要在于所需文件的下载，官网界面并不是很友好。。。尤其是connector下载的时候，windows属于others下载，搞得我找来找去一直在链接里兜圈子。另外的难点在于上次实验的数据库中，有几个表之间主键副键相互依赖，不能方便地进行插入，所以重新写了一个表STUDENT进行本次实验。最后的难点当然就是自学这些工具的使用了，好在书上和网上教程还是挺多的。

参考文献及致谢

《数据库系统基础教程》 机械工业出版社 （美）Jeffrey D.UUllman Jennifer Widom著 岳丽华 金培权 万寿红 等译