《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

汤聪 151220098 1195882464@qq.com

实验环境

机房电脑配置，ide是eclipse

实验过程

1. 在高级程序设计语言中通过数据库连接API直接访问SQL Server。
2. 使用高级程序设计语言执行输入的SQL查询语句，输出查询结果。

Java代码：

Statement state=(Statement) dbConn.createStatement();

String sql="select \* from depart"; //查询语句 ResultSet rs=((java.sql.Statement) state).executeQuery(sql); //将sql语句传至数据库，返回的值为一个字符集用一个变量接收

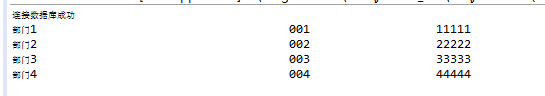
while(rs.next()){ //next（）获取里面的内容

System.out.println(rs.getString(1)+" "+rs.getString(2)+" "+rs.getString(3));}

dbConn.close();

}

实验截图：



1. 使用高级程序设计语言执行输入的SQL插入、删除、更新语句。
2. 添加成员：

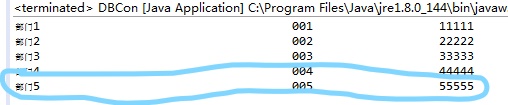
Java代码：

Statement state=(Statement) dbConn.createStatement(); //容器

String sql="insert into depart values('5','005','55555')"; //SQL语句

state.executeUpdate(sql);

实验截图：



1. 删除成员：

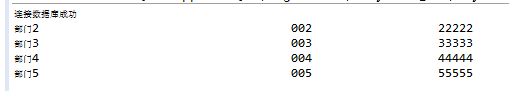
Java代码：

Statement state=(Statement) dbConn.createStatement(); //容器

String sql="delete from DEPART where dno='001'";

state.executeUpdate(sql);

实验截图：



1. Update1：

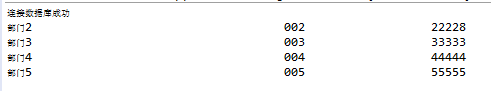
Java代码：

Statement state=(Statement) dbConn.createStatement();

String sql="update depart set dmanage='22228' where dno='002'";

state.executeUpdate(sql);

实验截图：



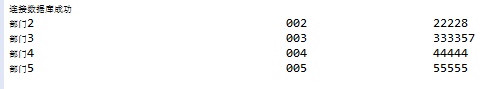
1. Update2：

Java代码：

String sql="update depart set dmanage='333357' where dname='部门3'";

state.executeUpdate(sql);

实验截图：



1. 在高级程序设计语言中通过数据库连接池来管理连接。
2. 请你利用DBCP创建并配置BasicDataSource对象，之后通过BasicDataSource对象的getConnection()方法获得与数据库的连接，并尝试执行SQL查询语句。

Java代码：

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Properties;

import javax.sql.DataSource;

import org.apache.commons.dbcp2.BasicDataSource;

import org.apache.commons.dbcp2.BasicDataSourceFactory;

public static void main(String[] args) {

String driverName="com.microsoft.sqlserver.jdbc.SQLServerDriver";

String dbURL="jdbc:sqlserver://localhost:1433;DatabaseName=tcpaul";

String userName="worker";

String userPwd="111";

BasicDataSource pool = new BasicDataSource();

pool.setUsername(userName);

pool.setPassword(userPwd);

pool.setDriverClassName(driverName);

pool.setUrl(dbURL);

try {

Connection con = pool.getConnection();

Statement state=(Statement) con.createStatement();

String sql1="select \* from depart"; //sql语句

ResultSet rs=((java.sql.Statement)state).executeQuery(sql1); //将sql语句传至数据库，返回的值为一个字符集用一个变量接收

con.close();

while(rs.next()){ //next（）获取里面的内容

System.out.println(rs.getString(1)+" "+rs.getString(2)+" "+rs.getString(3));}

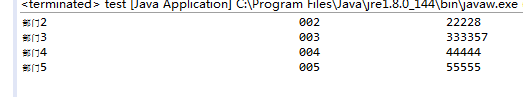
}catch(Exception e) {

e.printStackTrace();

System.out.print("连接失败");

}

实验截图：



1. 请你设计实验对比直接连接模式和连接池模式在打开连接、执行SQL、关闭连接上的效率（例如，分别使用直接连接模式和连接池模式打开连接、执行查询、关闭连接2000次，比较两者消耗的时间）。

直接连接模式：

Java代码：

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

{

try {

Class.forName(driverName);

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

// System.out.println("连接数据库成功");}

/\*查询数据库\*/

/\*Statement state=(Statement) dbConn.createStatement();

String sql="select \* from depart"; //sql语句

ResultSet rs=((java.sql.Statement) state).executeQuery(sql);

long endTime=System.currentTimeMillis(); //获取结束时间

System.out.println("程序运行时间： "+(endTime-startTime)+"ms");

实验截图：

C:\Users\汤聪\AppData\Local\Temp\360zip$Temp\360$10\2.2.PNG

连接池模式：

Java代码：

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

{\*/

try {

Connection con = pool.getConnection();

Statement state=(Statement) con.createStatement();

String sql1="select \* from depart"; //sql语句

ResultSet rs=((java.sql.Statement) state).executeQuery(sql1); //将sql语句传至数据库，返回的值为一个字符集用一个变量接收

// con.close();

long endTime=System.currentTimeMillis(); //获取结束时间

System.out.println("程序运行时间： "+(endTime-startTime)+"ms");\*

实验截图：

C:\Users\汤聪\AppData\Local\Temp\360zip$Temp\360$11\2.1.PNG

结论：连接池模式下打开连接，执行SQL，关闭连接上效率高很多。

实验中遇到的困难及解决办法

主要是没有接触过如何在java中操控数据库，在网上一些博客和论坛的帮助下，总体完成的比较顺利。

参考文献及致谢

[www.baidu.com](http://www.baidu.com)

[www.csdn.com](http://www.csdn.com)