**Федеральное агентство по образованию**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Институт информационных технологий и управления**

**Кафедра компьютерных систем и программных технологий**

**ОТЧЕТ**

**по лабораторной работе №8**

# «Разработка клиентского приложения»

**Студент гр.43501/1: Русаков Р.В.**

**Преподаватель: Мяснов А.В.**

**Санкт-Петербург**

**2014**

1. **Цели работы**

Необходимо создать консольное приложение выполняющее следующие функции:

* + соединение с БД, выполнение фиксированного SQL-запроса и получение результатов запроса
  + добавление данных в одну из таблиц БД
  + выполнение хранимой процедуры

1. **Выполнение работы**

Диалог с пользователем реализован через консоль, для выбора нужного действия необходимо ввести с клавиатуры его номер :

* + 1. В качестве фиксированного запроса выступает select \* FROM genres, выводящий список имеющихся жанров
    2. Добавление данных ведётся в ту же таблицу жанров, на вход подаются идентификатор и название нового жанра.
    3. Вызывается хранимая процедура LAB5, на вход которой подаётся номер заказа, а на выходе выводится его сумма.
    4. Завершение работы программы

Код программы:

package app;

import java.sql.\*;

import java.io.PrintStream;

import java.util.Scanner;

public class App {

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

String base\_addr = "jdbc:firebirdsql://localhost/C://db/book\_store2.fdb";

Class.forName("org.firebirdsql.jdbc.FBDriver").newInstance();

PrintStream printStream = new PrintStream(System.out, true, "cp866");

Scanner sc = new Scanner(System.in);

Connection conn=null;

conn = DriverManager.getConnection(base\_addr,"SYSDBA", "masterkey");

if (conn==null)

{

System.err.println("Could not connect to database");

}

Statement st = conn.createStatement();

ResultSet rs;

int t=0;

while(t!=4)

{

printStream.println("1. Show genres");

printStream.println("2. Add genres");

printStream.println("3. Execute procedure");

printStream.println("4. Exit");

if(sc.hasNextInt())

t=sc.nextInt();

else

{

sc.next();

continue;

}

if(t==1)

{

rs = st.executeQuery("select \* FROM genres order by genre\_id;");

while(rs.next())

{

System.out.println();

printStream.println(rs.getString(1)+ "." + rs.getString(2) );

}

printStream.println(" ");

continue;

}

if(t==2)

{

int kol=0,cnt=0;

printStream.println("Insert genre\_name");

String tstr=sc.next();

rs = st.executeQuery("select count(genre\_id) FROM genres where name ='" + tstr + "';");

while(rs.next())

{

cnt=rs.getInt(1);

}

if(cnt!=0)

{

printStream.println("This genre already exist");

continue;

}

rs = st.executeQuery("select first 1 genre\_id FROM genres order by genre\_id desc ;");

while(rs.next())

{

kol=rs.getInt(1);

}

kol++;

tstr="INSERT INTO genres (genre\_id,name) VALUES ("+String.valueOf(kol)+",'"+tstr+"')";

try

{

st.executeUpdate(tstr);

}

catch(org.firebirdsql.jdbc.FBSQLException e)

{

printStream.println("Wrong genre\_id, genre was not inserted");

continue;

}

continue;

}

if(t==3)

{

printStream.println("Insert genre\_id");

int t1=0;

if(sc.hasNextInt())

{

t1=sc.nextInt();

rs = st.executeQuery("select genre\_id FROM genres where genre\_id=" + Integer.toString(t1) + ";");

if (!rs.isBeforeFirst() ) {

printStream.println("This genre is not exist");

continue;

}

}

else

{

sc.next();

printStream.println("Wrong value");

continue;

}

Date s1,s2;

String tstr1=sc.next();

try

{

s1=Date.valueOf(tstr1);

}

catch(java.lang.IllegalArgumentException e)

{

printStream.println("Wrong date");

continue;

}

String tstr2=sc.next();

try

{

s2=Date.valueOf(tstr2);

}

catch(java.lang.IllegalArgumentException e)

{

printStream.println("Wrong date");

continue;

}

CallableStatement call\_stmt=conn.prepareCall("{call lab4\_1(?,?,?)}");

call\_stmt.setInt(1, t1);

call\_stmt.setDate(2, s1);

call\_stmt.setDate(3, s2);

call\_stmt.registerOutParameter(1, Types.FLOAT);

call\_stmt.execute();

float result = call\_stmt.getFloat(1);

System.out.println("Cost: "+result);

}

}

st.close();

conn.close();

}

}

1. **Вывод**

В ходе работы на языке Java было написано клиентское приложение, связывающееся с базой данных по стандарту JDBC. В нём реализованы: простой SQL запрос, команда на добавление данных в таблицу и запуск хранимой процедуры. К плюсам JDBC относится то, что к любой базе можно присоединиться посредством URL, нет необходимости устанавливать клиентскую программу, код является универсальным.