Санкт-Петербургский политехнический университет Петра Великого

Институт информационных технологий и управления

Кафедра компьютерных систем и программных технологий

**Отчёт лабораторной работе №8**

**Дисциплина**: Базы данных

**Тема**: Хранимые процедуры

Выполнил студент гр. 43501/1 С.И.Бочкова

(подпись)

Руководитель А.В. Мяснов

(подпись)

“ ” 2015 г.

Санкт - Петербург

2015

1. **Цель работы**

Разработать консольное приложение на языке Java, 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. **Проведение работы**

Код приложения:

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package films;  import java.io.UnsupportedEncodingException;  import java.sql.\*;  import java.util.Scanner;  import java.util.Vector;  /\*\*  /\*\*  \*  \* @author Sony  \*/  public class Films {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) throws ClassNotFoundException, SQLException, InstantiationException, IllegalAccessException {  // TODO code application logic here  //System.setOut(new java.io.PrintStream(System.out, true, "Cp866")); //Cp866 UTF-8    Connection connect = null;  // объект класса, с помощью которого будут выполняться SQL запросы  java.sql.Statement stmt = null;  // создаем класс для ввода с консоли  Scanner sc = new Scanner(System.in);  int numb\_table = 0; // Номер введенной таблицы  Vector<String> vec\_tab = new Vector<String>();  String temp2 = null; // Временные строковые  String [] temp3 = null;  int var = 0; // переменная = выбранный пункт меню  int argz; // аргументы запроса insert (атрибуты таблицы)  String argz2;  int cnt\_col = 0;  // создаем класс для хранения результата запроса  ResultSet res;  // инициализируем Firebird JDBC driver  Class.forName("org.firebirdsql.jdbc.FBDriver").newInstance();  //путь к базе данных  String strPath = "jdbc:firebirdsql://localhost/C:/Users/Sony/Documents/Bd/Films.fdb";  Class.forName("org.firebirdsql.jdbc.FBDriver").newInstance();  // создаём подключение к базе данных  connect = DriverManager.getConnection(strPath, "SYSDBA", "masterkey");  if (connect == null) {  System.err.println("Could not connect to database.");}  // создаём класс, с помощью которого будут выполняться SQL запросы  stmt = connect.createStatement();  System.out.println("Подключение к БД успешно выполнено.");  // Получаем список всех таблиц БД:  DatabaseMetaData metaData = connect.getMetaData();  ResultSet temp=metaData.getTables(temp2, temp2, temp2, temp3);  while(temp.next())  {  temp2=temp.getString(3);  if(!temp2.contains("$"))  vec\_tab.add(temp2);  }  // цикл для вывода меню и выполнения операций с консоли  while (var != 4) {  // вывод меню  System.out.println("-------------------------------------------");  System.out.println("МЕНЮ:");  System.out.println("1.Вывести список таблиц.");  System.out.println("2.Добавить запись в таблицу Company.");  System.out.println("3.Выполнить хранимую процедуру Top\_10\_dir.");  System.out.println("4.Выход");  // считывание номера пункта меню  System.out.println("Выберите пункт меню:");  try{  var = Integer.parseInt(sc.nextLine());  }catch(NumberFormatException e){  System.err.println("Ошибка! Вводите только цифры.");  continue;  }  if (var > 4)  System.err.println("Ошибка! Пункт меню с таким номером отсутствует.");  if (var == 1)  {  System.out.println("Список таблиц:");  for(int i=1;i<=vec\_tab.size();i++)  {  System.out.printf("%d. %s\n",i,vec\_tab.elementAt(i-1));  }  System.out.println("Введите номер таблицы для отображения ее содержимого или "  + "\n0 для возврата в основное меню:");  try{  numb\_table=Integer.parseInt(sc.nextLine());  }catch(NumberFormatException e){  System.err.println("Ошибка! Номер должен быть числом!");  continue;  }  if((numb\_table > vec\_tab.size()) || (numb\_table < 0)){  System.err.println("Ошибка! Таблица с таким номером отсутсвует.");  continue;  }  if(numb\_table == 0){  continue;  }  System.out.println();  //Выполняем SQL запрос:  res = stmt.executeQuery("SELECT \* from "+ vec\_tab.elementAt(numb\_table-1));  // Выводим результат:  cnt\_col = res.getMetaData().getColumnCount();  // Выводим содержание таблицы:  // Сначала имена столбцов:  for(int i = 1; i < cnt\_col + 1; i++){  System.out.print(res.getMetaData().getColumnName(i)+  " | ");  }  // Затем сами записи в таблице:  while(res.next())  {  System.out.println();  for (int i = 1;i < cnt\_col + 1;i++)  {  Object obj = res.getObject(i);  if (obj!=null)  {  System.out.print(obj+" \t ");  }  }  }  System.out.println();  continue;  }  if (var == 2 )  {  if(connect == null) {  System.err.println("Соединение с БД не установлено.");  continue;  }  // вводим аргументы insert  System.out.println("Введите уникальный идентификатор для компании.");  try{  argz=Integer.parseInt(sc.nextLine());  }catch(NumberFormatException e){  System.err.println("Ошибка! Номер не является числом или превышает 9 символов");  continue;  }  if (argz <= 0)  {  System.err.println("Ошибка! Номер не может быть отрицательным или равным нулю.");  continue;  }  System.out.println("Введите название кинокомпании.");  argz2 = sc.nextLine();  if (argz2.length()>25)  {  System.err.println("Ошибка! Длина названия кинокомпании не должна превышать 25 символов.");  continue;  }    System.out.println("Введите год основания.");  String argz3 = sc.nextLine();  if (argz3.length()<0)  {  System.err.println("Ошибка! Год сонования не может быть отрицательным.");  continue;  }    System.out.println("Введите название прокатной компании.");  String argz4 = sc.nextLine();  if (argz4.length()>25)  {  System.err.println("Ошибка! Длина названия кинокомпании не должна превышать 25 символов.");  continue;  }    System.out.println("Введите,сколько денег собрала в прокате данная компания.");  String argz5 = sc.nextLine();  if (argz5.length()>25)  {  System.err.println("Ошибка! Длина не должна превышать 25 символов.");  continue;  }    stmt.executeUpdate("insert into Company values (" + String.valueOf(argz) + ",'"+argz2+"','"+argz3+"','"+argz4+"','"+argz5+"');");  System.out.println("Запись добавлена в таблицу.");  continue;  }  if (var == 3)  {  if(connect == null) {  System.err.println("Соединение с БД не установлено.");  continue;  }  System.out.println("Хранимая процедура выводит 10 режиссеров с наилучшим соотношением суммарных сборов от прокатов фильмов к суммарным гонорарам актеров. \n"  );  PreparedStatement pstmt = connect.prepareStatement("{call Top\_10\_dir}");  pstmt.execute();  System.out.println("\nХранимая процедура Top\_10\_dir выполнена.");  pstmt.close();  continue;    }  if (var == 4)  {  System.out.println("Выход (exit)");  continue;  }  }  System.exit(0);  }    } |

Тестирование приложения:

|  |
| --- |
| run:  Подключение к БД успешно выполнено.  -------------------------------------------  МЕНЮ:  1.Вывести список таблиц.  2.Добавить запись в таблицу Company.  3.Выполнить хранимую процедуру Top\_10\_dir.  4.Выход  Выберите пункт меню:  1  Список таблиц:  1. CAREER  2. COMPANY  3. COUNTRY  4. FILM  5. GENRE  6. MADE\_IN  7. MAN  8. OWNER  9. PART  10. PRODUCTION  11. SOUNDTRACK  12. STYLE  Введите номер таблицы для отображения ее содержимого или  0 для возврата в основное меню:  5  ID | NAME |  1 action  2 biography  3 western  4 crime  5 documentary  6 drama  7 comedy  8 melodrama  9 animation  10 musical  11 thriller  12 horror  13 fantasy  14 adventure  15 mystery  16 romance  -------------------------------------------  МЕНЮ:  1.Вывести список таблиц.  2.Добавить запись в таблицу Company.  3.Выполнить хранимую процедуру Top\_10\_dir.  4.Выход  Выберите пункт меню:  2  Введите уникальный идентификатор для компании.  10002  Введите название кинокомпании.  Warner  Введите год основания.  1963  Введите название прокатной компании.  FilmsP  Введите,сколько денег собрала в прокате данная компания.  54648121354  Запись добавлена в таблицу.  -------------------------------------------  МЕНЮ:  1.Вывести список таблиц.  2.Добавить запись в таблицу Company.  3.Выполнить хранимую процедуру Top\_10\_dir.  4.Выход  Выберите пункт меню:  3  Хранимая процедура выводит 10 режиссеров с наилучшим соотношением суммарных сборов от прокатов фильмов к суммарным гонорарам актеров.  Хранимая процедура Top\_10\_dir выполнена.  -------------------------------------------  МЕНЮ:  1.Вывести список таблиц.  2.Добавить запись в таблицу Company.  3.Выполнить хранимую процедуру Top\_10\_dir.  4.Выход  Выберите пункт меню:  4  Выход (exit) |

**3.Вывод:**

Трехзвенная архитектура— архитектурная модель программного комплекса, предполагающая наличие в нём трёх компонентов: клиента, сервера приложений (к которому подключено клиентское приложение) и сервера баз данных (с которым работает сервер приложений).

Клиент — это интерфейсный компонент комплекса, предоставляемый конечному пользователю. Этот уровень не должен иметь прямых связей с базой данных (по требованиям безопасности и масштабируемости), быть нагруженным основной бизнес-логикой (по требованиям масштабируемости) и хранить состояние приложения (по требованиям надёжности). На этот уровень обычно выносится только простейшая бизнес-логика.

Сервер приложений (средний слой, связующий слой) располагается на втором уровне, на нём сосредоточена большая часть бизнес-логики. Вне его остаются только фрагменты, экспортируемые на клиента (терминалы), а также элементы логики, погруженные в базу данных (хранимые процедуры и триггеры).

Сервер баз данных (слой данных) обеспечивает хранение данных.

Основные плюсы:

-Высокая безопасность;

- Высокая надежность;

- Балансировка нагрузки;

- Увеличение скорости работы;

- Простое обновление;

Минусы:

- Разработка приложений для трёхуровневых программных комплексов сложнее, чем для клиент-серверных приложений;

- Наличие дополнительного связующего программного обеспечения может налагать дополнительные издержки в администрировании таких комплексов