НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА
Институт радиоэлектроники и информационных технологий
Кафедра “Прикладная математика”

Лабораторная работа №1 по курсу «Базы данных»
«SQL + JDBC »

Выполнила:
Горенкова А. В.
Группа 16-ПМ
Проверил:
Моисеев А.Е.

НИЖНИЙ НОВГОРОД
2018 г.

Оглавление

Введение

SQL (*structured query language* — «язык структурированных запросов») — декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

SQL обладает как недостатками, так и преимуществами. Так из основных плюсов можно выделить *декларативность* языка, т. е. Программист описывает только то, какие данные нужно извлечь или модифицировать, и *наличие стандартов*, а так же большинство текстов SQL-запросов можно перенести из одной системы управления базой данных в другую с минимальными изменениями.

Из недостатков можно выделить сложность работы для обычного пользователя, а не программиста, большой объём стандарта языка, а так же различные компании часто отходят от стандарта, тем самым переносимость между СУБД становится невозможной. К тому же в языке SQL не предусмотрена работа с рекурсиями, циклами и пользовательскими функциями.

На данный момент SQL — один из самых популярных языков для работы с базами данных, который можно применить практически к любой СУБД.

В данной работе мы будем использовать язык SQL совместно с JDBC, стандартом взаимодействия приложения с различными СУБД. JDBC основан на концепции драйверов, позволяющей получать соединение с базой данных по специальному URL. Интерфейс позволяет взаимодействовать Java-приложению с базами данных с

помощью запросов на языке SQL. Такой подход имеет несколько преимуществ: для разработки не нужно знать всей специфики базы данных, с которой предстоит работать, а так же нет необходимости в громоздкой клиентской программе и к любой базе можно подсоединиться через легко описываемый URL.

Задание

- Описать базу данных мебели на языке Java с использованием JDBC-интерфейса.
- Сделать несколько SQL-запросов к таблице.

Решение

1. Импортируем необходимые для работы библиотеки:

```
import java.sql.*;
```

2. Подключаемся к базе данных и создаем `Statement`:

```
static final String JDBC_DRIVER = "org.h2.Driver";
static final String DB_URL = "jdbc:h2:file:~/base";
Class.forName(JDBC_DRIVER).newInstance();
Connection con = DriverManager.getConnection(DB_URL);
Statement stm = con.createStatement();
```

3. Создаём и заполняем таблицу

```
stm.executeUpdate("CREATE TABLE Furniture" +
                        "(id INTEGER NOT NULL AUTO_INCREMENT, " +
                        " name VARCHAR(255), " +
                        " color VARCHAR(255), " +
                        " price INTEGER, " +
                        " PRIMARY KEY ( id ))");
stm.executeUpdate("INSERT INTO Furniture (name, color, price) VALUES ('Table',
'White', 3200)");
stm.executeUpdate("INSERT INTO Furniture (name, color, price) VALUES ('Table',
'Black', 1000)");
...
```

4. Выполняем SQL-запрос и получаем результат

```
ResultSet result;
result = stm.executeQuery("SELECT * FROM Furniture");
```

Результаты работы

1. Получаем из базы данных всю мебель:

```
ResultSet result;
result = stm.executeQuery("SELECT * FROM Furniture");
```

```
Hello
Table, White, 3200;
Table, Black, 1000;
Cupboard, Blue, 9000;
Char, Brown, 2000;
kirpitch, red, 400;
Chair, Green, 400;
Shelf, White, 8000;
Cupboard, Brown, 20000;
Table, Red, 9000;
Shelf, Brown, 200;
Chair, White, 500;
Cupboard, Brown, 5000;
Shelf, Green, 1000;
Cupboard, Green, 8000;
Chair, Brown, 800;
Shelf, Red, 800;
Table, Green, 9000;
Стол, Белый, 900;
```

2. Получаем из базы данных всю мебель коричневого цвета:

```
result = stm.executeQuery("SELECT * FROM Furniture WHERE color = 'Brown'");
```

```
Hello
Char, Brown, 2000;
Cupboard, Brown, 20000;
Shelf, Brown, 200;
Cupboard, Brown, 5000;
Chair, Brown, 800;
```

3. Получаем из базы данных всю мебель, цена которой меньше 1000:

```
Hello
kirpitch, red, 400;
Chair, Green, 400;
Shelf, Brown, 200;
Chair, White, 500;
Chair, Brown, 800;
Shelf, Red, 800;
Стол, Белый, 900;
```

Листинг

Main.java:

```
import java.sql.*;

public class Main {
	static final String JDBC_DRIVER = "org.h2.Driver";
	static final String DB_URL = "jdbc:h2:file:~/base";

	public static void main(String[] args) {
		System.out.println("Hello");
		try {
			Class.forName(JDBC_DRIVER).newInstance();
			Connection con = DriverManager.getConnection(DB_URL);
			Statement stm = con.createStatement();
			stm.executeUpdate("CREATE TABLE Furniture" +
				"(id INTEGER NOT NULL AUTO_INCREMENT, " +
				" name VARCHAR(255), " +
				" color VARCHAR(255), " +
				" price INTEGER, " +
				" PRIMARY KEY ( id ))");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Table', 'White', 3200)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Table', 'Black', 1000)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Cupboard', 'Blue', 9000)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Chair', 'Blue', 500)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Char', 'Brown', 2000)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Char', 'Green', 1800)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Chair', 'Green', 400)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Table', 'Black', 10000)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Shelf', 'White', 8000)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Cupboard', 'Brown', 20000)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Table', 'Red', 9000)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Shelf', 'Brown', 200)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Chair', 'White', 500)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Cupboard', 'Brown', 5000)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Shelf', 'Green', 1000)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Cupboard', 'Green', 8000)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Chair', 'Brown', 800)");
			stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Shelf', 'Red', 800)");
```

```
                  stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Table', 'Green', 9000)");
                  stm.executeUpdate("INSERT INTO Furniture (name, color, price)
VALUES ('Chair', 'White', 500)");
                  ResultSet result;
//                result = stm.executeQuery("SELECT * FROM Furniture");
//                result = stm.executeQuery("SELECT * FROM Furniture WHERE color
= 'Brown'");
//                result = stm.executeQuery("SELECT name, color FROM
Furniture");
                  result = stm.executeQuery("SELECT * FROM Furniture WHERE price
< 1000");
                  while (result.next()){
                        System.out.println(result.getString("name") + ", " +
                                          result.getString("color")+ ", " +
                                                result.getString("price")+ ";" );
                  }
            } catch(Exception e) {
                  e.printStackTrace();
            }
      }
}
```