ЛР10_1. Основы работы с базой данных SQLight

Цель работы: изучить основы работы с базой данных в SQLight

Задание

1. Создать новое Java приложение.
2. В MainActivity создать класс DBHelper для подключения к базе данных из одной таблицы и объект класса DBHelper. Структура таблицы должна соответствовать структуре класса своего варианта из предыдущих ЛР.
3. В MainActivity создать объекты для отображения данных из базы данных, добавления данных в базу данных, удаления всех записей из базы данных.
4. Создать ссылки на объекты для рабрты с базой данных и связать их с объектами.
5. Создать код добавления информации в базу данных. Запустить приложение.
6. Создать код для отображения данных из базы данных. Запустить приложение.
7. Создать код для удаления всех данных из базы данных. Запустить приложение.
8. Оформить отчет по лабораторной работе. В отчете должен быть программный код, листиги xml-файлов и скрины экранов с кратким описанием.

Ход работы:

1. Создаем новое Java приложение с шаблоном Empty Views Activity.

2. В MainActivity создаем класс DBHelper и объект класса DBHelper для подключения к базе данных к своей таблице. Структура таблицы должна соответствовать структуре класса своего варианта из предыдущих ЛР.

В соответствии с кодом своего класса **Film** создаем класс **DBHelper**. ***Если мы хотим создавать объекты класса DBHelper в других активити, то лучше создать этот класс в отдельном файле и сделать его public.***

Рассмотрим фрагмент кода нашего класса Film:

```
public class Film {
   private String genre;
   private String nameFilm;
   private int  filmYear;
    ...
```

Создаем класс DBHelper в конце файла SQLightActivity.java:

```
class DBHelper extends SQLiteOpenHelper {
    public DBHelper(Context context) {
// конструктор суперкласса
        super(context, "myDB", null, 1);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
    //  Log.d(LOG_TAG, "--- onCreate database ---");
    // создаем таблицу с полями
    db.execSQL("create table mytable (" + "id integer primary key autoincrement," + " genre text," + " nameFilm text, "+ " filmYear text" + ");");}
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    }
```

```
}
```

Не забудьте импортировать нужные классы в MainActivity:

```
import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.os.Bundle;
```

В начале класса MainActivity создаем ссылку для подключения к базе данных:

```
DBHelper dbHelper;
```

В методе onCreate класса MainActivity создаем объект для подключения к базе данных и управления версиями:

```
dbHelper = new DBHelper(this);
```

В результате имеем:

```
public class MainActivity extends AppCompatActivity {
   DBHelper dbHelper;
   @Override
   public void onCreate(Bundle savedInstanceState) {
  ...
     // создаем объект для создания и управления версиями БД
     dbHelper = new DBHelper(this);
  }
```

3. В MainActivity создаем

- текстовое поле для отображения данных из таблицы БД,
- компоненты для добавления данных в базу данных:
  - текстовое поле и выпадающий список для выбора жанра,
  - два текстовых поля и 2 поля ввода для ввода названия фильма и года выпуска,
- кнопку для добавления данных,
- кнопку для отображения данных,
- кнопку для удаления всех данных из базы данных.

В файле макета это может выглядеть примерно так:

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="0dp"
    android:layout_height="300dp"
    android:layout_marginStart="10dp"
    android:text="@string/textview"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
  <TextView
    android:id="@+id/text_g"
    android:layout_width="wrap_content"
    android:layout_height="45dp"
    android:layout_marginStart="10dp"
    android:layout_marginTop="20dp"
    android:paddingTop="15dp"
```

```
    android:text="@string/text_g"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView3" />
<Spinner
    android:id="@+id/film"
    android:layout_width="fill_parent"
    android:layout_height="50dp"
    android:layout_marginStart="80dp"
    android:entries="@array/films"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/text_g" />
<TextView
    android:id="@+id/text_f"
    android:layout_width="wrap_content"
    android:layout_height="45dp"
    android:layout_marginStart="10dp"
    android:layout_marginTop="70dp"
    android:paddingTop="15dp"
    android:text="@string/text_f"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView3" />
<EditText
    android:id="@+id/etext_f"
    android:layout_width="fill_parent"
    android:layout_height="100dp"
    android:layout_marginStart="80dp"
    android:text="@string/text_f"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/text_f" />
<TextView
    android:id="@+id/text_y"
    android:layout_width="wrap_content"
    android:layout_height="45dp"
    android:layout_marginStart="10dp"
    android:layout_marginTop="170dp"
    android:paddingTop="15dp"
    android:text="@string/text_y"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView3" />
<EditText
    android:id="@+id/etext_y"
    android:layout_width="200dp"
    android:layout_height="50dp"
    android:layout_marginStart="80dp"
    android:text="@string/text_y"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/text_y" />
<Button
    android:id="@+id/buttonAdd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
```

```
        android:layout_marginTop="220dp"
        android:onClick="onClickAdd"
        android:text="@string/buttonAdd"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView3" />
    <Button
        android:id="@+id/buttonSelect"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="220dp"
        android:onClick="onClickSelect"
        android:text="@string/buttonSel"
        app:layout_constraintStart_toEndOf="@+id/buttonAdd"
        app:layout_constraintTop_toBottomOf="@+id/textView3" />
    <Button
        android:id="@+id/buttonDelete"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="220dp"
        android:onClick="onClickDelete"
        android:text="@string/buttondel"
        app:layout_constraintStart_toEndOf="@+id/buttonSelect"
        app:layout_constraintTop_toBottomOf="@+id/textView3" />
```

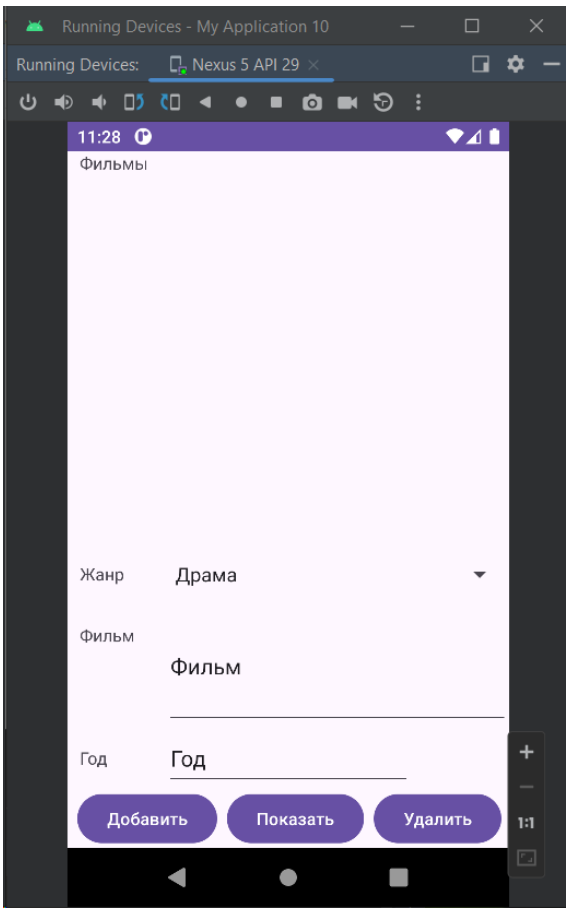
Не забудьте добавить в ресурсы в файл strings.xml текст для элементов макета

```
<string-array name="films">
    <item>Драма</item>
    <item>Мультфильм</item>
    <item>Комедия</item>
    <item>Приключения</item>
</string-array>
<string name="buttonAdd">Добавить</string>
<string name="buttonSel">Показать</string>
<string name="buttondel">Удалить</string>
<string name="text_g">Жанр</string>
<string name="text_f">Фильм</string>
<string name="text_y">Год</string>
<string name="textview">Фильмы</string>
```

Создаем «пустые» процедуры добавления, отображения, удаления данных:

```
public void onClickAdd(View view) {   }
public void onClickSelect(View view) {   }
public void onClickDelete(View view) {   }
```

Запускаем приложение:



4. В начале класса MainActivity создаем ссылки для подключения компонент к активити:

```
private TextView text1;
private Spinner filmSpinner;
private EditText etextF;
private EditText etextY;
```

В методе onCreate класса SQLightActivity создаем код подключения компонент к активити:

```
text1 = (TextView) findViewById(R.id.textView3);
filmSpinner = (Spinner) findViewById(R.id.film);
etextF = (EditText) findViewById(R.id.etext_f);
etextY = (EditText) findViewById(R.id.etext_y);
```

5. Программируем кнопку добавления информацию в базу данных.

```
public void onClickAdd(View view) {
// создаем объект для данных
    ContentValues cv = new ContentValues();
// получаем данные из полей ввода
    String genre = String.valueOf(filmSpinner.getSelectedItem());
    String nameFilm = etextF.getText().toString();
    String filmYear = etextY.getText().toString();
// подключаемся к БД
    SQLiteDatabase db = dbHelper.getWritableDatabase();
//подготовим данные для вставки в виде пар: наименование столбца - значение
        cv.put("genre", genre);
        cv.put("nameFilm", nameFilm);
```

```
            cv.put("filmYear", filmYear);
        // вставляем запись и получаем ее ID
            long rowID = db.insert("mytable", null, cv);
            text1.setText("Добавлена запись с кодом "+rowID);
        // закрываем соединение с БД
            dbHelper.close();
        }
```

Запускаем приложение. Если появляющаяся клавиатура виртуального устройства не содержит символов кириллицы, вводите текст латинскими символами или используйте буфер обмена. Нажимаем кнопку «Добавить».

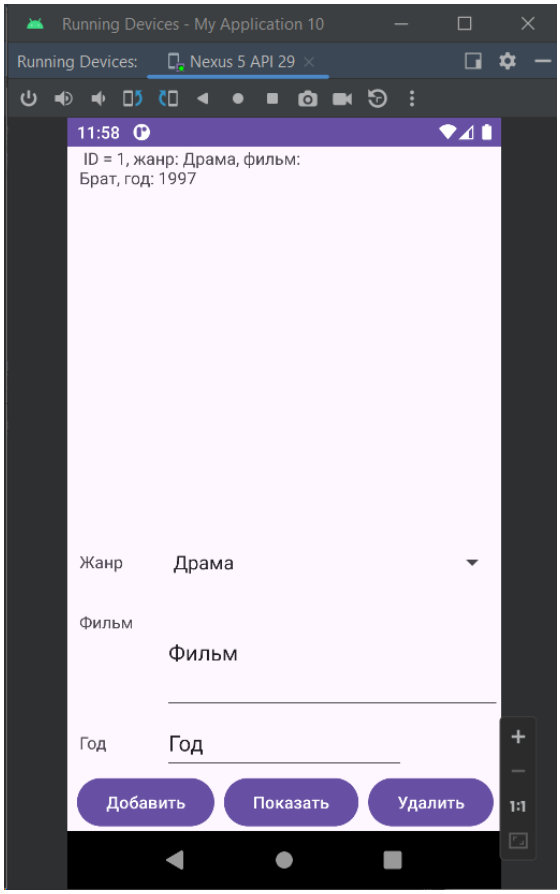


6. Программируем кнопку отображения информации из базы данных:

```
public void onClickSelect(View view) {
    // подключаемся к БД
        SQLiteDatabase db = dbHelper.getWritableDatabase();
        text1.setText("");
     // делаем запрос всех данных из таблицы mytable, получаем Cursor
        Cursor c = db.query("mytable", null, null, null, null, null, null);
    // ставим позицию курсора на первую строку выборки если в выборке нет строк, вернется false
       if (c.moveToFirst()) { // определяем номера столбцов по имени в выборке
            int idColIndex = c.getColumnIndex("id");
            int genreColIndex = c.getColumnIndex("genre");
            int nameFilmColIndex = c.getColumnIndex("nameFilm");
            int filmYearColIndex = c.getColumnIndex("filmYear");
           do {// получаем значения по номерам столбцов и пишем все в текстовое поле
```

```
                text1.append(" ID = " + c.getInt(idColIndex) + ", жанр: " +
c.getString(genreColIndex) + ", фильм: " + c.getString(nameFilmColIndex)+", год: " +
c.getString(filmYearColIndex)+"\n" );
      // переход на следующую строку, если следующей нет, то false - выход из цикла
             } while (c.moveToNext());
           } else  text1.append("0 rows");
           dbHelper.close();
  }
```

Запускаем приложение. Нажимаем кнопку «Показать»:



Добавляем еще несколько записей в базу данных

7. Программируем кнопку удаления всех данных из базы данных.

```
    public void onClickDelete(View view) {
          SQLiteDatabase db = dbHelper.getWritableDatabase();
        // удаляем все записи
          int clearCount = db.delete("mytable", null, null);
          text1.setText( "Удалено записей: " + clearCount);
          dbHelper.close();
       }
```

Запускаем приложение:



8. Оформляем отчет по лабораторной работе. В отчете должен быть программный код, листиги xml-файлов и скрины экранов с кратким описанием.