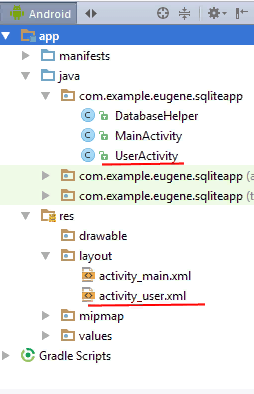
**Добавление, удаление и обновление данных в SQLite**

Продолжим работу с проектом из прошлой темы, где мы получаем данные. Теперь добавим в него стандартную CRUD-логику (создание, обновление, удаление).

Чтобы не нагромождать форму с главной activity, все остальные действия по работе с данными будут происходить на другом экране. Добавим в проект новый класс activity, который назовем UserActivity:



В файле **activity\_user.xml** определим универсальую форму для добавления/обновления/удаления данных:

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"      android:orientation="vertical"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:padding="16dp">      <EditText          android:id="@+id/name"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:hint="Введите имя"/>      <EditText          android:id="@+id/year"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:hint="Введите год рождения"/>      <LinearLayout          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:orientation="horizontal">            <Button              android:id="@+id/saveButton"              android:layout\_width="0dp"              android:layout\_weight="1"              android:layout\_height="wrap\_content"              android:text="Сохранить"              android:onClick="save"/>          <Button              android:id="@+id/deleteButton"              android:layout\_width="0dp"              android:layout\_weight="1"              android:layout\_height="wrap\_content"              android:text="Удалить"              android:onClick="delete"/>      </LinearLayout>  </LinearLayout> |

И также изменим код **UserActivity**:

package com.example.eugene.sqliteapp;

import android.content.ContentValues;

import android.content.Intent;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

public class UserActivity extends AppCompatActivity {

    EditText nameBox;

    EditText yearBox;

    Button delButton;

    Button saveButton;

    DatabaseHelper sqlHelper;

    SQLiteDatabase db;

    Cursor userCursor;

    long userId=0;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_user);

        nameBox = (EditText) findViewById(R.id.name);

        yearBox = (EditText) findViewById(R.id.year);

        delButton = (Button) findViewById(R.id.deleteButton);

        saveButton = (Button) findViewById(R.id.saveButton);

        sqlHelper = new DatabaseHelper(this);

        db = sqlHelper.getWritableDatabase();

        Bundle extras = getIntent().getExtras();

        if (extras != null) {

            userId = extras.getLong("id");

        }

        // если 0, то добавление

        if (userId > 0) {

            // получаем элемент по id из бд

            userCursor = db.rawQuery("select \* from " + DatabaseHelper.TABLE + " where " +

                    DatabaseHelper.COLUMN\_ID + "=?", new String[]{String.valueOf(userId)});

            userCursor.moveToFirst();

            nameBox.setText(userCursor.getString(1));

            yearBox.setText(String.valueOf(userCursor.getInt(2)));

            userCursor.close();

        } else {

            // скрываем кнопку удаления

            delButton.setVisibility(View.GONE);

        }

    }

    public void save(View view){

        ContentValues cv = new ContentValues();

        cv.put(DatabaseHelper.COLUMN\_NAME, nameBox.getText().toString());

        cv.put(DatabaseHelper.COLUMN\_YEAR, Integer.parseInt(yearBox.getText().toString()));

        if (userId > 0) {

            db.update(DatabaseHelper.TABLE, cv, DatabaseHelper.COLUMN\_ID + "=" + String.valueOf(userId), null);

        } else {

            db.insert(DatabaseHelper.TABLE, null, cv);

        }

        goHome();

    }

    public void delete(View view){

        db.delete(DatabaseHelper.TABLE, "\_id = ?", new String[]{String.valueOf(userId)});

        goHome();

    }

    private void goHome(){

        // закрываем подключение

        db.close();

        // переход к главной activity

        Intent intent = new Intent(this, MainActivity.class);

        intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_SINGLE\_TOP);

        startActivity(intent);

    }

}

При обновлении или удалении объекта из списка из главной activity в UserActivity будет передаваться id объекта:

|  |
| --- |
| long userId=0;  //.................  Bundle extras = getIntent().getExtras();  if (extras != null) {      userId = extras.getLong("id");  } |

Если из MainActivity не было передано id, то устанавливаем его значение 0, следовательно, у нас будет добавление, а не редактирование/удаление

Если id определен, то получаем по нему из базы данных объект для редактирования/удаления:

if (id < 0) {

    userCursor = db.rawQuery("select \* from " + DatabaseHelper.TABLE + " where " + DatabaseHelper.COLUMN\_ID + "=?", new String[]{String.valueOf(id)});

    userCursor.moveToFirst();

    nameBox.setText(userCursor.getString(1));

    yearBox.setText(String.valueOf(userCursor.getInt(2)));

    userCursor.close();

}

Иначе просто скрываем кнопку удаления.

Для выполнения операций по вставке, обновлению и удалению данных SQLiteDatabase имеет методы insert(), update() и delete(). Эти методы вызываются в обработчиках кнопок:

|  |  |
| --- | --- |
| 1 | db.delete(DatabaseHelper.TABLE, "\_id = ?", new String[]{String.valueOf(id)}); |

В метод delete() передается название таблицы, а также столбец, по которому происходит удаление, и его значение. В качестве критерия можно выбрать несколько столбцов, поэтому третьим параметром идет массив. Знак вопроса ? обозначает параметр, вместо которого подставляется значение из третьего параметра.

**ContentValues**

Для добавления или обновления нам надо создать объект **ContentValues**. Данный объект представляет словарь, который содержит набор пар "ключ-значение". Для добавления в этот словарь нового объекта применяется метод **put**. Первый параметр метода - это ключ, а второй - значение, например:

|  |  |
| --- | --- |
| 1  2  3 | ContentValues cv = new ContentValues();  cv.put("NAME", "Tom");  cv.put("YEAR", 30); |

В качестве значений в метод put можно передавать строки, целые числа, числа с плавающей точкой

В данном же случае добавляются введенные в текстовое поля значения:

|  |  |
| --- | --- |
| 1  2  3 | ContentValues cv = new ContentValues();  cv.put(DatabaseHelper.COLUMN\_NAME, nameBox.getText().toString());  cv.put(DatabaseHelper.COLUMN\_YEAR, Integer.parseInt(yearBox.getText().toString())); |

При обновлении в метод update() передается название таблицы, объект ContentValues и критерий, по которому происходит обновление (в данном случае столбец id):

|  |  |
| --- | --- |
| 1 | db.update(DatabaseHelper.TABLE, cv, DatabaseHelper.COLUMN\_ID + "=" + String.valueOf(id), null); |

Метод insert() принимает название таблицы, объект ContentValues с добавляемыми значениями. Второй параметр является необязательным: он передает столбец, в который надо добавить значение NULL:

|  |  |
| --- | --- |
| 1 | db.insert(DatabaseHelper.TABLE, null, cv); |

Вместо этих методов, как в прошлой теме, можно использовать метод execSQL() с точным указанием выполняемого sql-выражения. В то же время методы delete/insert/update имеют преимущество - они возвращают id измененной записи, по которому мы можем узнать об успешности операции, или -1 в случае неудачной операции:

|  |  |
| --- | --- |
| 1  2  3  4  5 | long result = db.insert(DatabaseHelper.TABLE, null, cv);  if(result>0){        // действия  } |

После каждой операции выполняется метод goHome(), который возвращает на главную activity.

После этого нам надо исправить код MainActivity, чтобы она инициировала выполнение кода в UserActivity. Для этого изменим код**activity\_main.xml**:

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"      android:orientation="vertical"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent">      <Button          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:text="Добавить"          android:onClick="add"          android:textSize="18dp"/>      <ListView          android:id="@+id/list"          android:layout\_width="match\_parent"          android:layout\_height="match\_parent" />  </LinearLayout> |

В данном случае была добавлена кнопка для вызова UserActivity.

И также изменим код класса **MainActivity**:

package com.example.eugene.sqliteapp;

import android.content.Intent;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ListView;

import android.widget.SimpleCursorAdapter;

public class MainActivity extends AppCompatActivity {

    ListView userList;

    DatabaseHelper databaseHelper;

    SQLiteDatabase db;

    Cursor userCursor;

    SimpleCursorAdapter userAdapter;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        userList = (ListView)findViewById(R.id.list);

        userList.setOnItemClickListener(new AdapterView.OnItemClickListener() {

            @Override

            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

                Intent intent = new Intent(getApplicationContext(), UserActivity.class);

                intent.putExtra("id", id);

                startActivity(intent);

            }

        });

        databaseHelper = new DatabaseHelper(getApplicationContext());

    }

    @Override

    public void onResume() {

        super.onResume();

        // открываем подключение

        db = databaseHelper.getReadableDatabase();

        //получаем данные из бд в виде курсора

        userCursor =  db.rawQuery("select \* from "+ DatabaseHelper.TABLE, null);

        // определяем, какие столбцы из курсора будут выводиться в ListView

        String[] headers = new String[] {DatabaseHelper.COLUMN\_NAME, DatabaseHelper.COLUMN\_YEAR};

        // создаем адаптер, передаем в него курсор

        userAdapter = new SimpleCursorAdapter(this, android.R.layout.two\_line\_list\_item,

                userCursor, headers, new int[]{android.R.id.text1, android.R.id.text2}, 0);

        userList.setAdapter(userAdapter);

    }

    // по нажатию на кнопку запускаем UserActivity для добавления данных

    public void add(View view){

        Intent intent = new Intent(this, UserActivity.class);

        startActivity(intent);

    }

    @Override

    public void onDestroy(){

        super.onDestroy();

        // Закрываем подключение и курсор

        db.close();

        userCursor.close();

    }

}

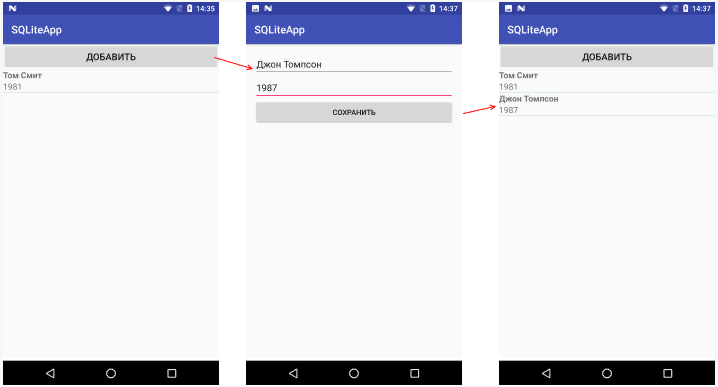
При нажатии на кнопку запускается UserActivity, при этом не передается никакого id, то есть в UserActivity id будет равен нулю, значит будет идти добавление данных:

|  |
| --- |
| public void add(View view){      Intent intent = new Intent(this, UserActivity.class);      startActivity(intent);  } |

Другую ситуацию представляет обработчик нажатия на элемент списка - при нажатии также будет запускаться UserActivity, но теперь будет передаваться id выбранной записи:

|  |
| --- |
| public void onItemClick(AdapterView<?> parent, View view, int position, long id) {      Intent intent = new Intent(getApplicationContext(), UserActivity.class);      intent.putExtra("id", id);      startActivity(intent);  } |

Запустим приложение и нажмем на кнопку, которая должен перенаправлять на UserActivity:



При нажатии в MainActivity на элемент списка этот элемент попадет на UserActivity, где его можно будет удалить или подредактировать:

