**Руководство Android SQLite Database**

**View more Tutorials:**

* [Программирование на Android](https://o7planning.org/ru/11007/android)

1- Что такое SQLite?

2- Содержание примера

3- Шаги для работы с SQLite Database

4- Создать project & сделать дизайн интерфейса

5- Java Code

6- Запуск приложения

2Shares

1- Что такое SQLite?

**SQLite** это реляционная база данных с открытым исходным кодом, построена на операционной системе  **Android**, поэтому вы можете ее использовать в любое время без дополнительных конфигураций.

Обычно, с такими базами данных как  **Oracle**, **MySQL**,.. вам нужна библиотека драйвера (Driver Libary), и подключить к**JDBC**, но с  **SQLite** в этом нет необходимости.

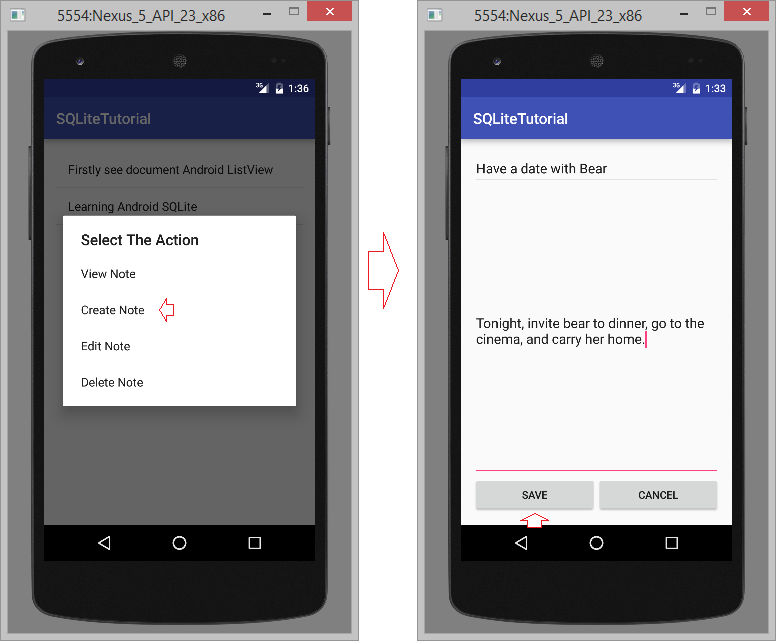
2- Содержание примера

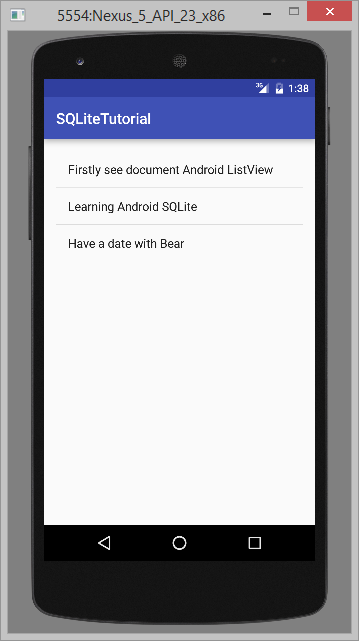
Допустим вы создали приложение работающее на вашем  **Android**, это приложение сохраняет ваши заметки за день (Note). Вам нужна структура таблицы для хранения заметок.

* Название таблицы: **Note**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
| Note\_Id | int | Primary Key | Primary Key |
| Note\_Title | text |  | Note title |
| Note\_Content | text |  | Content |

Пример:





3- Шаги для работы с SQLite Database

Вам нужно создать утилитерный класс для работы с базой данных  **SQLite**, этот класс расширен из класса **SQLiteOpenHelper**. Существует 2 важных метода, которые вам нужно переопределить (override) это  **onCreate()** и**onUpgrade()**.

1. Создать class **MyDatabaseHelper** расширенный из **SQLiteOpenHelper**.
2. После расширения class из **SQLiteOpenHelper** вам нужно переписать два метода ***onCreate()*** и ***onUpgrage()***
   * ***onCreate()*** - Место где вы создаете таблицу. Вызывается (called) при создании базы данных.
   * ***onUpgrade()*** - Этот метод вызывается когда обновляется база данных, как как изменение структуры таблицы, добавление ограничений в базу данных и т.д.

public class MyDatabaseHelper extends SQLiteOpenHelper {

// ....

public MyDatabaseHelper(Context context)  {

    super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

    // Script to create table.

    String script = "CREATE TABLE " + TABLE\_NOTE + "("

            + COLUMN\_NOTE\_ID + " INTEGER PRIMARY KEY," + COLUMN\_NOTE\_TITLE + " TEXT,"

            + COLUMN\_NOTE\_CONTENT + " TEXT" + ")";

    // Execute script.

    db.execSQL(script);

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    // Drop table

    db.execSQL("DROP TABLE IF EXISTS " + TABLE\_NOTE);

    // Recreate

    onCreate(db);

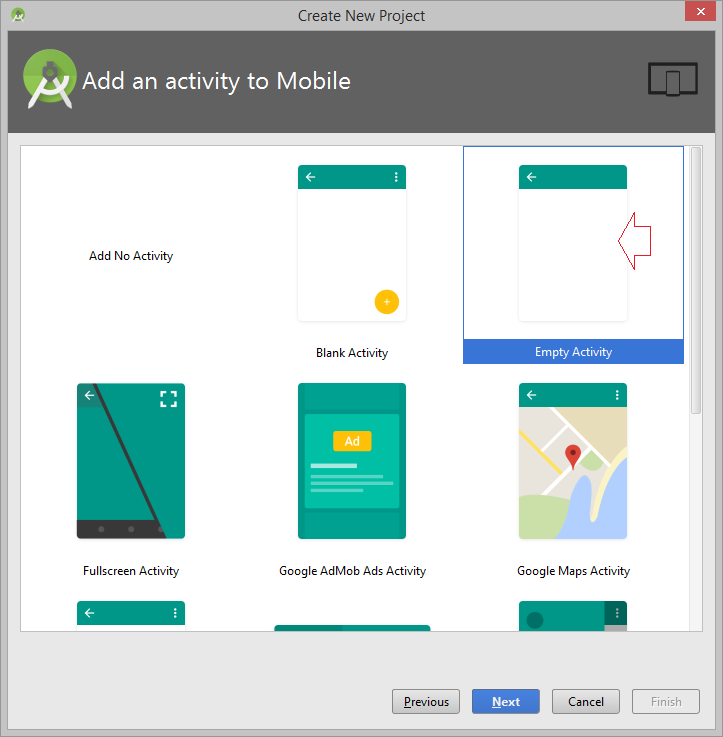
}

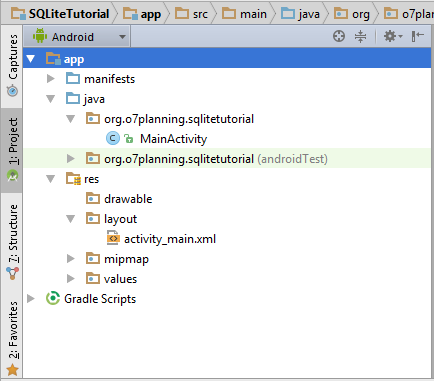
// ...

}

4- Создать project & сделать дизайн интерфейса

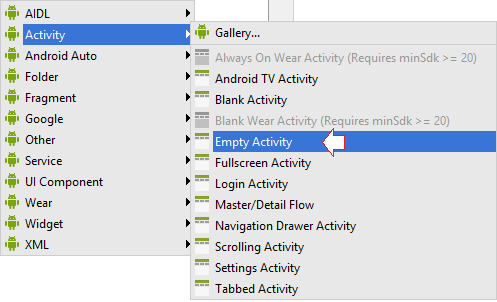
Создайте  **"Empty Activity"**project с названием  **SQLiteTutorial** для выполнения следующих примеров.





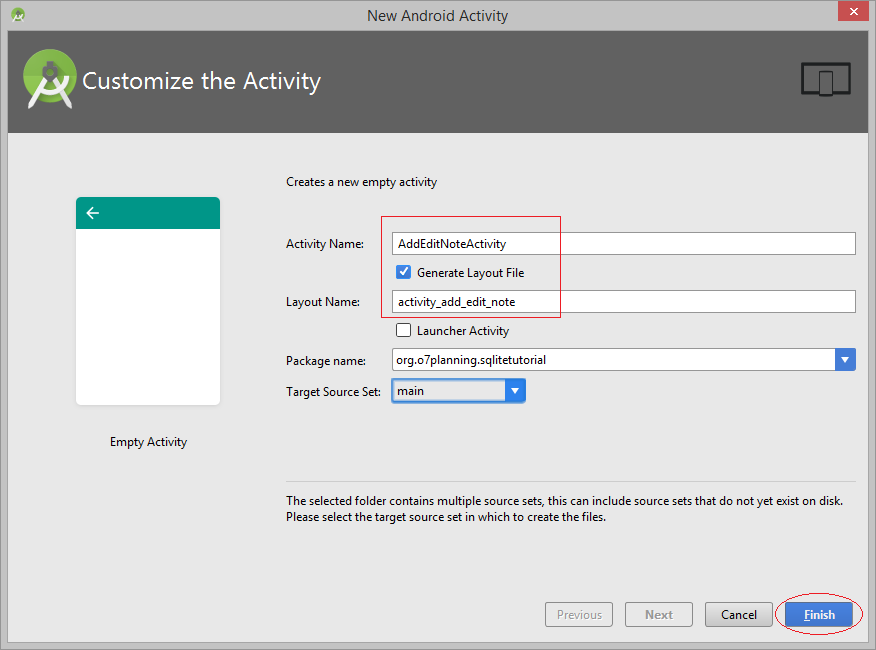
Создать пустой  **Activity**.

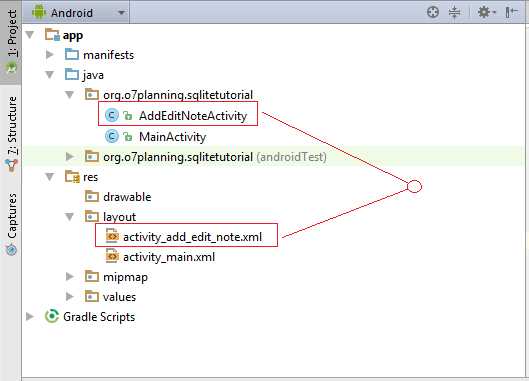
* **File/New/Activity/Empty Activity**



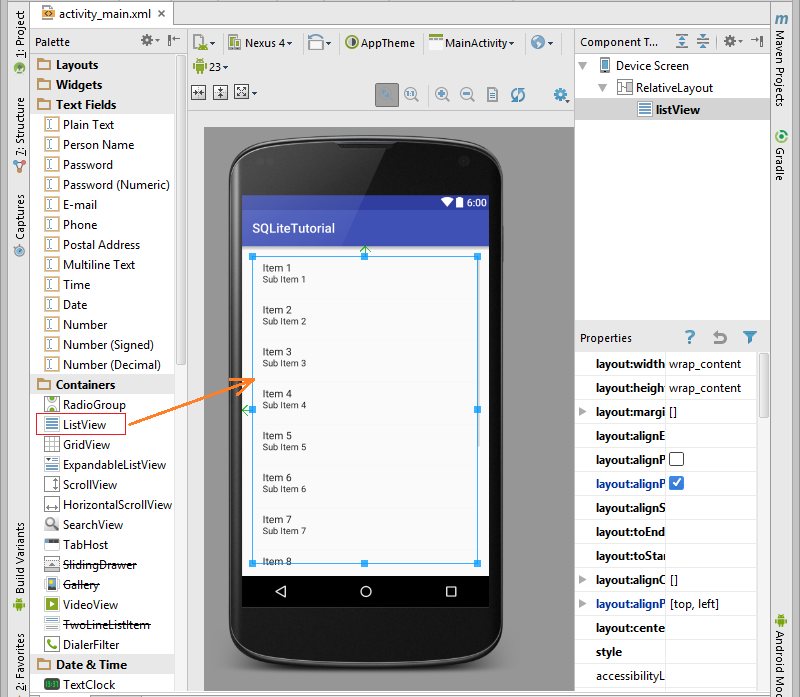
Ввoд:

* **Activity Name**: AddEditNoteActivity
  + (Check Generate Layout File)
* **Layout**: activity\_add\_edit\_note
* **Package name**: org.o7planning.sqlitetutorial





**activity\_main.xml**



**activity\_main.xml**

?

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="<http://schemas.android.com/apk/res/android>"

  xmlns:tools="<http://schemas.android.com/tools>" android:layout\_width="match\_parent"

  android:layout\_height="match\_parent" android:paddingLeft="@dimen/activity\_horizontal\_margin"

  android:paddingRight="@dimen/activity\_horizontal\_margin"

  android:paddingTop="@dimen/activity\_vertical\_margin"

  android:paddingBottom="@dimen/activity\_vertical\_margin" tools:context=".MainActivity">

  <ListView

      android:layout\_width="wrap\_content"

      android:layout\_height="wrap\_content"

      android:id="@+id/listView"

      android:layout\_alignParentTop="true"

      android:layout\_alignParentLeft="true"

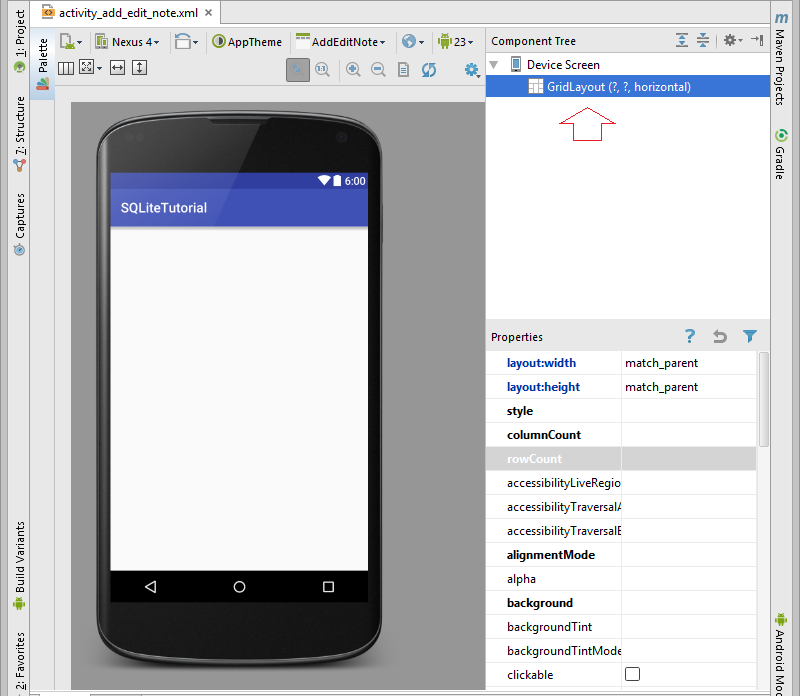
      android:layout\_alignParentStart="true" />

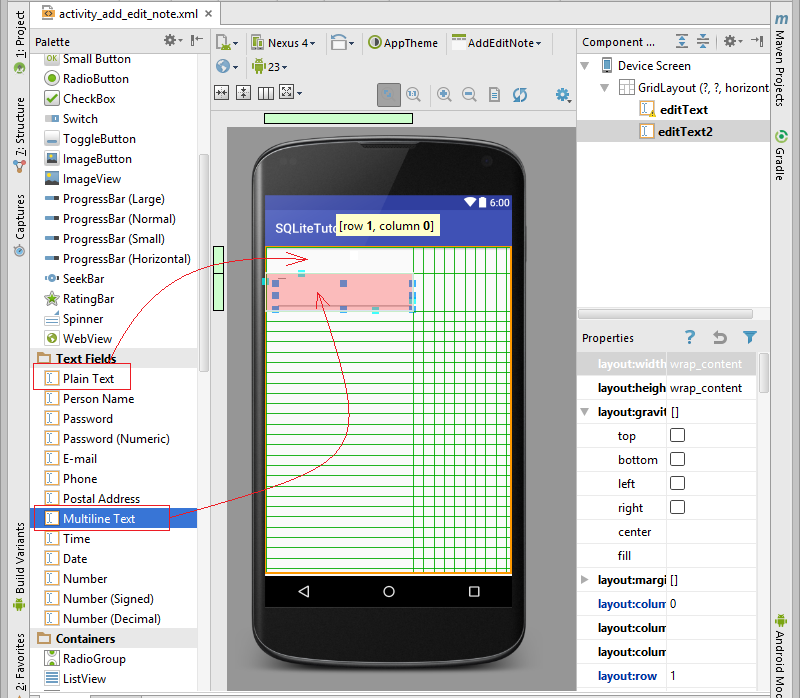
</RelativeLayout>

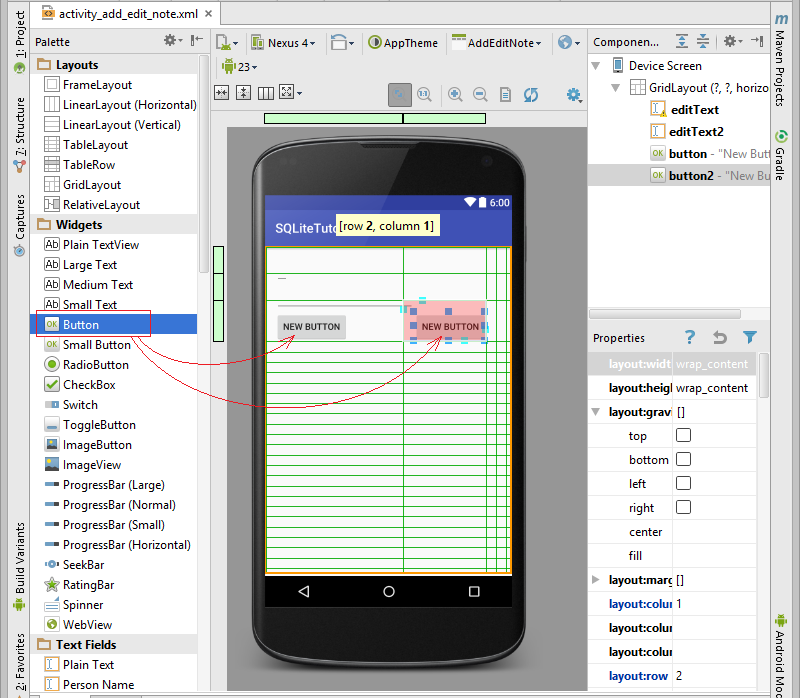
**activity\_add\_edit\_note.xml**

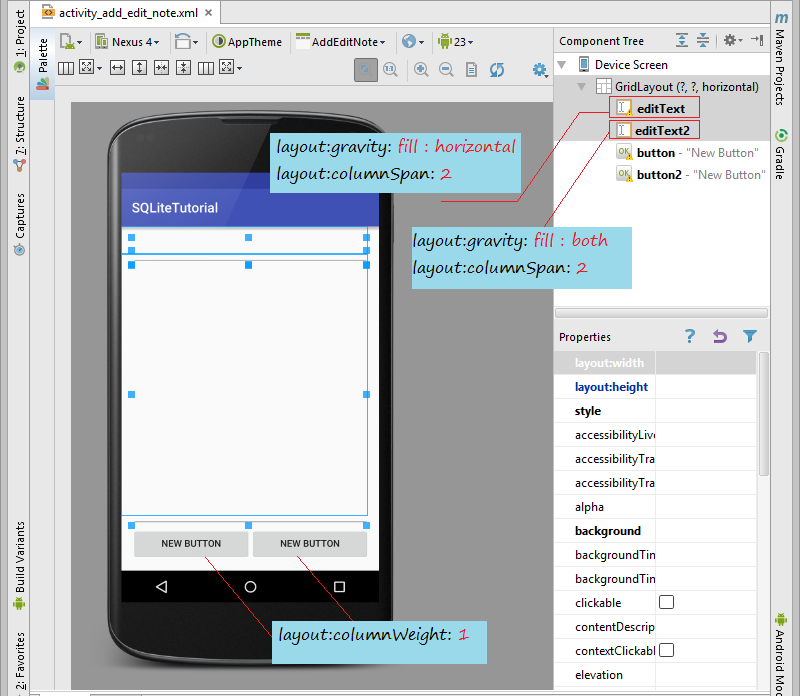
Посмотрите пошаговый **SLIDER**для дизайна интерфейса:

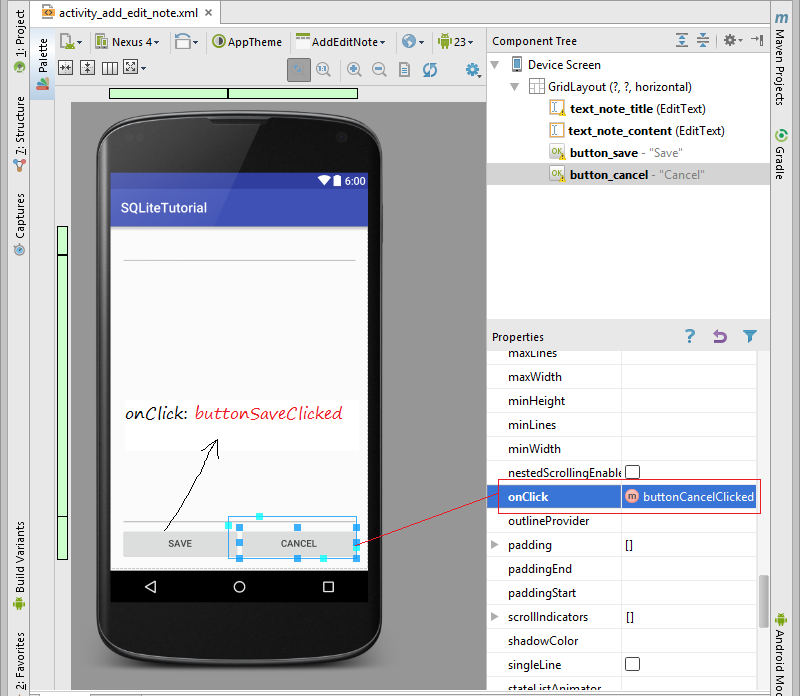
* **SLIDER:**



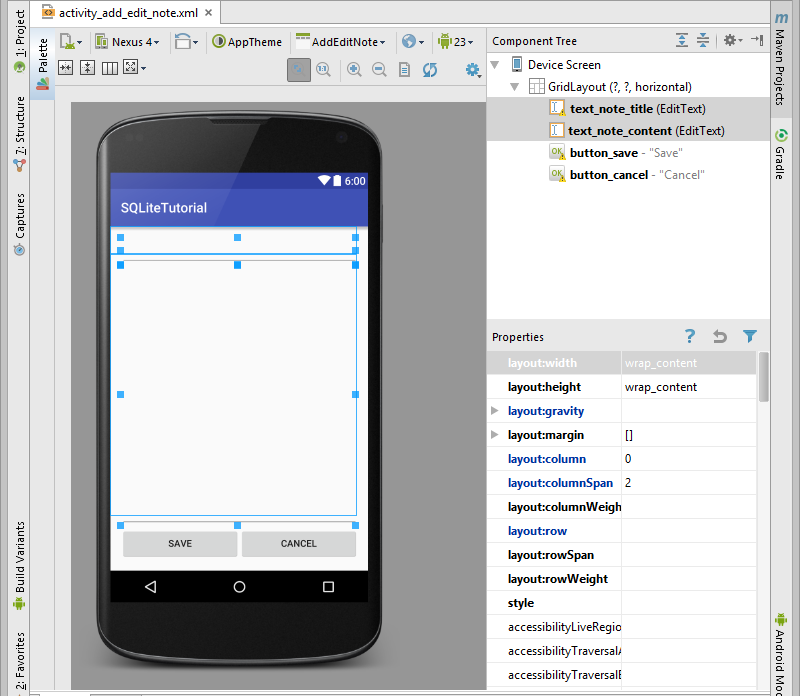








Interface:



**activity\_add\_edit\_note.xml**

?

<?xml version="1.0" encoding="utf-8"?>

<GridLayout xmlns:android="<http://schemas.android.com/apk/res/android>"

   xmlns:tools="<http://schemas.android.com/tools>" android:layout\_width="match\_parent"

   android:layout\_height="match\_parent" android:paddingLeft="@dimen/activity\_horizontal\_margin"

   android:paddingRight="@dimen/activity\_horizontal\_margin"

   android:paddingTop="@dimen/activity\_vertical\_margin"

   android:paddingBottom="@dimen/activity\_vertical\_margin"

   tools:context="org.o7planning.sqlitetutorial.AddEditNoteActivity">

   <EditText

       android:layout\_width="wrap\_content"

       android:layout\_height="wrap\_content"

       android:id="@+id/text\_note\_title"

       android:layout\_row="0"

       android:layout\_column="0"

       android:layout\_gravity="fill\_horizontal"

       android:layout\_columnSpan="2" />

   <EditText

       android:layout\_width="wrap\_content"

       android:layout\_height="wrap\_content"

       android:inputType="textMultiLine"

       android:ems="10"

       android:id="@+id/text\_note\_content"

       android:layout\_row="1"

       android:layout\_column="0"

       android:layout\_gravity="fill"

       android:layout\_columnSpan="2" />

   <Button

       android:layout\_width="wrap\_content"

       android:layout\_height="wrap\_content"

       android:text="Save"

       android:id="@+id/button\_save"

       android:layout\_row="2"

       android:layout\_column="0"

       android:layout\_columnWeight="1"

       android:onClick="buttonSaveClicked" />

   <Button

       android:layout\_width="wrap\_content"

       android:layout\_height="wrap\_content"

       android:text="Cancel"

       android:id="@+id/button\_cancel"

       android:layout\_row="2"

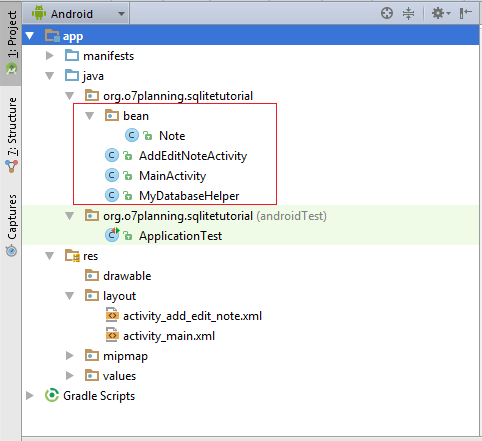
       android:layout\_column="1"

       android:layout\_columnWeight="1"

       android:onClick="buttonCancelClicked" />

</GridLayout>

5- Java Code



**Note.java**

?

package org.o7planning.sqlitetutorial.bean;

import java.io.Serializable;

public class Note implements Serializable {

   private int noteId;

   private String noteTitle;

   private String noteContent;

   public Note()  {

   }

   public Note(  String noteTitle, String noteContent) {

       this.noteTitle= noteTitle;

       this.noteContent= noteContent;

   }

   public Note(int noteId, String noteTitle, String noteContent) {

       this.noteId= noteId;

       this.noteTitle= noteTitle;

       this.noteContent= noteContent;

   }

   public int getNoteId() {

       return noteId;

   }

   public void setNoteId(int noteId) {

       this.noteId = noteId;

   }

   public String getNoteTitle() {

       return noteTitle;

   }

   public void setNoteTitle(String noteTitle) {

       this.noteTitle = noteTitle;

   }

   public String getNoteContent() {

       return noteContent;

   }

   public void setNoteContent(String noteContent) {

       this.noteContent = noteContent;

   }

   @Override

   public String toString()  {

       return this.noteTitle;

   }

}

**MyDatabaseHelper.java**

?

package org.o7planning.sqlitetutorial;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.util.Log;

import org.o7planning.sqlitetutorial.bean.Note;

import java.util.ArrayList;

import java.util.List;

public class MyDatabaseHelper extends SQLiteOpenHelper {

    private static final String TAG = "SQLite";

    // Database Version

    private static final int DATABASE\_VERSION = 1;

    // Database Name

    private static final String DATABASE\_NAME = "Note\_Manager";

    // Table name: Note.

    private static final String TABLE\_NOTE = "Note";

    private static final String COLUMN\_NOTE\_ID ="Note\_Id";

    private static final String COLUMN\_NOTE\_TITLE ="Note\_Title";

    private static final String COLUMN\_NOTE\_CONTENT = "Note\_Content";

    public MyDatabaseHelper(Context context)  {

        super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

    }

    // Create table

    @Override

    public void onCreate(SQLiteDatabase db) {

        Log.i(TAG, "MyDatabaseHelper.onCreate ... ");

        // Script.

        String script = "CREATE TABLE " + TABLE\_NOTE + "("

                + COLUMN\_NOTE\_ID + " INTEGER PRIMARY KEY," + COLUMN\_NOTE\_TITLE + " TEXT,"

                + COLUMN\_NOTE\_CONTENT + " TEXT" + ")";

        // Execute Script.

        db.execSQL(script);

    }

    @Override

    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

        Log.i(TAG, "MyDatabaseHelper.onUpgrade ... ");

        // Drop older table if existed

        db.execSQL("DROP TABLE IF EXISTS " + TABLE\_NOTE);

        // Create tables again

        onCreate(db);

    }

    // If Note table has no data

    // default, Insert 2 records.

    public void createDefaultNotesIfNeed()  {

        int count = this.getNotesCount();

        if(count ==0 ) {

            Note note1 = new Note("Firstly see Android ListView",

                    "See Android ListView Example in o7planning.org");

            Note note2 = new Note("Learning Android SQLite",

                    "See Android SQLite Example in o7planning.org");

            this.addNote(note1);

            this.addNote(note2);

        }

    }

    public void addNote(Note note) {

        Log.i(TAG, "MyDatabaseHelper.addNote ... " + note.getNoteTitle());

        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();

        values.put(COLUMN\_NOTE\_TITLE, note.getNoteTitle());

        values.put(COLUMN\_NOTE\_CONTENT, note.getNoteContent());

        // Inserting Row

        db.insert(TABLE\_NOTE, null, values);

        // Closing database connection

        db.close();

    }

    public Note getNote(int id) {

        Log.i(TAG, "MyDatabaseHelper.getNote ... " + id);

        SQLiteDatabase db = this.getReadableDatabase();

        Cursor cursor = db.query(TABLE\_NOTE, new String[] { COLUMN\_NOTE\_ID,

                        COLUMN\_NOTE\_TITLE, COLUMN\_NOTE\_CONTENT }, COLUMN\_NOTE\_ID + "=?",

                new String[] { String.valueOf(id) }, null, null, null, null);

        if (cursor != null)

            cursor.moveToFirst();

        Note note = new Note(Integer.parseInt(cursor.getString(0)),

                cursor.getString(1), cursor.getString(2));

        // return note

        return note;

    }

    public List<Note> getAllNotes() {

        Log.i(TAG, "MyDatabaseHelper.getAllNotes ... " );

        List<Note> noteList = new ArrayList<Note>();

        // Select All Query

        String selectQuery = "SELECT  \* FROM " + TABLE\_NOTE;

        SQLiteDatabase db = this.getWritableDatabase();

        Cursor cursor = db.rawQuery(selectQuery, null);

        // looping through all rows and adding to list

        if (cursor.moveToFirst()) {

            do {

                Note note = new Note();

                note.setNoteId(Integer.parseInt(cursor.getString(0)));

                note.setNoteTitle(cursor.getString(1));

                note.setNoteContent(cursor.getString(2));

                // Adding note to list

                noteList.add(note);

            } while (cursor.moveToNext());

        }

        // return note list

        return noteList;

    }

    public int getNotesCount() {

        Log.i(TAG, "MyDatabaseHelper.getNotesCount ... " );

        String countQuery = "SELECT  \* FROM " + TABLE\_NOTE;

        SQLiteDatabase db = this.getReadableDatabase();

        Cursor cursor = db.rawQuery(countQuery, null);

        int count = cursor.getCount();

        cursor.close();

        // return count

        return count;

    }

    public int updateNote(Note note) {

        Log.i(TAG, "MyDatabaseHelper.updateNote ... "  + note.getNoteTitle());

        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();

        values.put(COLUMN\_NOTE\_TITLE, note.getNoteTitle());

        values.put(COLUMN\_NOTE\_CONTENT, note.getNoteContent());

        // updating row

        return db.update(TABLE\_NOTE, values, COLUMN\_NOTE\_ID + " = ?",

                new String[]{String.valueOf(note.getNoteId())});

    }

    public void deleteNote(Note note) {

        Log.i(TAG, "MyDatabaseHelper.updateNote ... " + note.getNoteTitle() );

        SQLiteDatabase db = this.getWritableDatabase();

        db.delete(TABLE\_NOTE, COLUMN\_NOTE\_ID + " = ?",

                new String[] { String.valueOf(note.getNoteId()) });

        db.close();

    }

}

**MainActivity.java**

?

package org.o7planning.sqlitetutorial;

import android.app.Activity;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.ContextMenu;

import android.view.MenuItem;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.ListView;

import android.widget.Toast;

import org.o7planning.sqlitetutorial.bean.Note;

import java.util.ArrayList;

import java.util.List;

public class MainActivity extends AppCompatActivity {

    private ListView listView;

    private static final int MENU\_ITEM\_VIEW = 111;

    private static final int MENU\_ITEM\_EDIT = 222;

    private static final int MENU\_ITEM\_CREATE = 333;

    private static final int MENU\_ITEM\_DELETE = 444;

    private static final int MY\_REQUEST\_CODE = 1000;

    private final List<Note> noteList = new ArrayList<Note>();

    private ArrayAdapter<Note> listViewAdapter;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        // Get ListView object from xml

        listView = (ListView) findViewById(R.id.listView);

        MyDatabaseHelper db = new MyDatabaseHelper(this);

        db.createDefaultNotesIfNeed();

        List<Note> list=  db.getAllNotes();

        this.noteList.addAll(list);

        // Define a new Adapter

        // 1 - Context

        // 2 - Layout for the row

        // 3 - ID of the TextView to which the data is written

        // 4 - the List of data

        this.listViewAdapter = new ArrayAdapter<Note>(this,

                android.R.layout.simple\_list\_item\_1, android.R.id.text1, this.noteList);

        // Assign adapter to ListView

        this.listView.setAdapter(this.listViewAdapter);

        // Register the ListView for Context menu

        registerForContextMenu(this.listView);

    }

    @Override

    public void onCreateContextMenu(ContextMenu menu, View view,

                                    ContextMenu.ContextMenuInfo menuInfo)    {

        super.onCreateContextMenu(menu, view, menuInfo);

        menu.setHeaderTitle("Select The Action");

        // groupId, itemId, order, title

        menu.add(0, MENU\_ITEM\_VIEW , 0, "View Note");

        menu.add(0, MENU\_ITEM\_CREATE , 1, "Create Note");

        menu.add(0, MENU\_ITEM\_EDIT , 2, "Edit Note");

        menu.add(0, MENU\_ITEM\_DELETE, 4, "Delete Note");

    }

    @Override

    public boolean onContextItemSelected(MenuItem item){

        AdapterView.AdapterContextMenuInfo

                info = (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();

        final Note selectedNote = (Note) this.listView.getItemAtPosition(info.position);

        if(item.getItemId() == MENU\_ITEM\_VIEW){

            Toast.makeText(getApplicationContext(),selectedNote.getNoteContent(),Toast.LENGTH\_LONG).show();

        }

        else if(item.getItemId() == MENU\_ITEM\_CREATE){

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

            // Start AddEditNoteActivity, (with feedback).

            this.startActivityForResult(intent, MY\_REQUEST\_CODE);

        }

        else if(item.getItemId() == MENU\_ITEM\_EDIT ){

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

            intent.putExtra("note", selectedNote);

            // Start AddEditNoteActivity, (with feedback).

            this.startActivityForResult(intent,MY\_REQUEST\_CODE);

        }

        else if(item.getItemId() == MENU\_ITEM\_DELETE){

            // Ask before deleting.

            new AlertDialog.Builder(this)

                    .setMessage(selectedNote.getNoteTitle()+". Are you sure you want to delete?")

                    .setCancelable(false)

                    .setPositiveButton("Yes", new DialogInterface.OnClickListener() {

                        public void onClick(DialogInterface dialog, int id) {

                            deleteNote(selectedNote);

                        }

                    })

                    .setNegativeButton("No", null)

                    .show();

        }

        else {

            return false;

        }

        return true;

    }

    // Delete a record

    private void deleteNote(Note note)  {

        MyDatabaseHelper db = new MyDatabaseHelper(this);

        db.deleteNote(note);

        this.noteList.remove(note);

        // Refresh ListView.

        this.listViewAdapter.notifyDataSetChanged();

    }

    // When AddEditNoteActivity completed, it sends feedback.

    // (If you start it using startActivityForResult ())

    @Override

    protected void onActivityResult(int requestCode, int resultCode, Intent data) {

        if (resultCode == Activity.RESULT\_OK && requestCode == MY\_REQUEST\_CODE ) {

            boolean needRefresh = data.getBooleanExtra("needRefresh",true);

            // Refresh ListView

            if(needRefresh) {

                this.noteList.clear();

                MyDatabaseHelper db = new MyDatabaseHelper(this);

                List<Note> list=  db.getAllNotes();

                this.noteList.addAll(list);

                // Notify the data change (To refresh the ListView).

                this.listViewAdapter.notifyDataSetChanged();

            }

        }

    }

}

**AddEditNoteActivity.java**

?

package org.o7planning.sqlitetutorial;

import android.app.Activity;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.EditText;

import android.widget.Toast;

import org.o7planning.sqlitetutorial.bean.Note;

public class AddEditNoteActivity extends AppCompatActivity {

    Note note;

    private static final int MODE\_CREATE = 1;

    private static final int MODE\_EDIT = 2;

    private int mode;

    private EditText textTitle;

    private EditText textContent;

    private boolean needRefresh;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_add\_edit\_note);

        this.textTitle = (EditText)this.findViewById(R.id.text\_note\_title);

        this.textContent = (EditText)this.findViewById(R.id.text\_note\_content);

        Intent intent = this.getIntent();

        this.note = (Note) intent.getSerializableExtra("note");

        if(note== null)  {

            this.mode = MODE\_CREATE;

        } else  {

            this.mode = MODE\_EDIT;

            this.textTitle.setText(note.getNoteTitle());

            this.textContent.setText(note.getNoteContent());

        }

    }

    // User Click on the Save button.

    public void buttonSaveClicked(View view)  {

        MyDatabaseHelper db = new MyDatabaseHelper(this);

        String title = this.textTitle.getText().toString();

        String content = this.textContent.getText().toString();

        if(title.equals("") || content.equals("")) {

            Toast.makeText(getApplicationContext(),

                    "Please enter title & content", Toast.LENGTH\_LONG).show();

            return;

        }

        if(mode==MODE\_CREATE ) {

            this.note= new Note(title,content);

            db.addNote(note);

        } else  {

            this.note.setNoteTitle(title);

            this.note.setNoteContent(content);

            db.updateNote(note);

        }

        this.needRefresh = true;

        // Back to MainActivity.

        this.onBackPressed();

    }

    // User Click on the Cancel button.

    public void buttonCancelClicked(View view)  {

        // Do nothing, back MainActivity.

        this.onBackPressed();

    }

    // When completed this Activity,

    // Send feedback to the Activity called it.

    @Override

    public void finish() {

        // Create Intent

        Intent data = new Intent();

        // Request MainActivity refresh its ListView (or not).

        data.putExtra("needRefresh", needRefresh);

        // Set Result

        this.setResult(Activity.RESULT\_OK, data);

        super.finish();

    }

}

6- Запуск приложения

Просмотр **Slider**:

* **SLIDER**

