

# SQLite

## 1. SQLite là gì?

SQLite là một cơ sở dữ liệu quan hệ, mã nguồn mở, nó được tích hợp sẵn trên hệ điều hành Android, vì vậy bạn có thể sử dụng nó bất cứ lúc nào, và không cần phải cấu hình gì thêm.

Thông thường với các cơ sở dữ liệu như Oracle, MySQL,.. bạn cần phải có thư viện điều khiển (Driver library), và tạo kết nối JDBC, tuy nhiên với SQLite điều đó là không cần thiết.

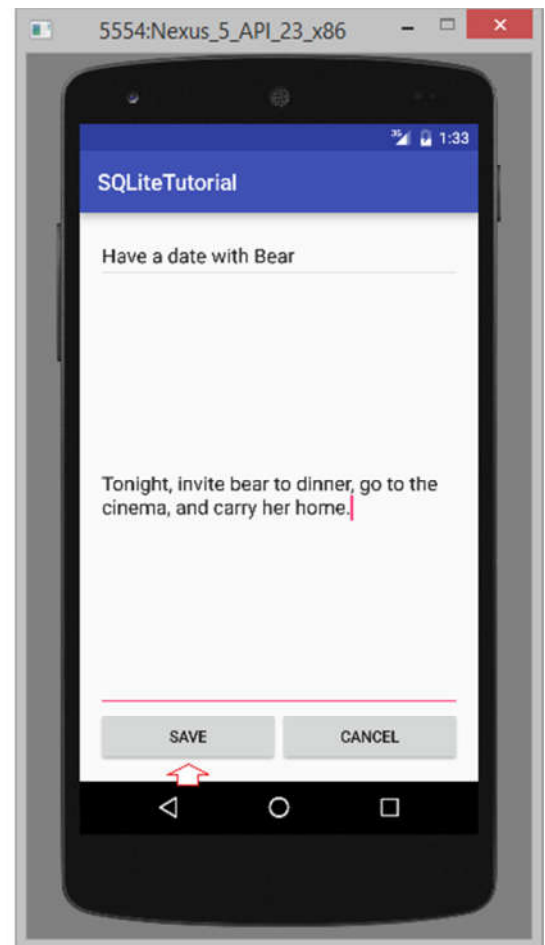
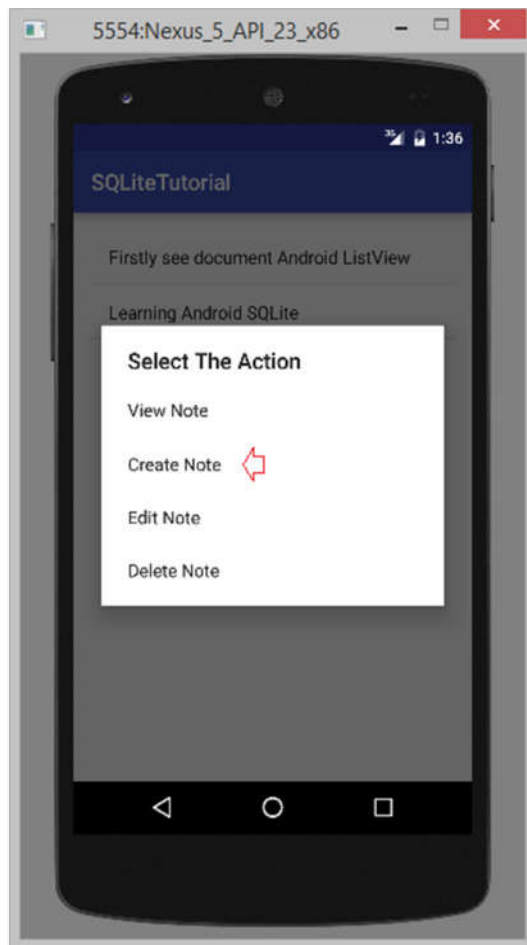
## 2. Ví dụ

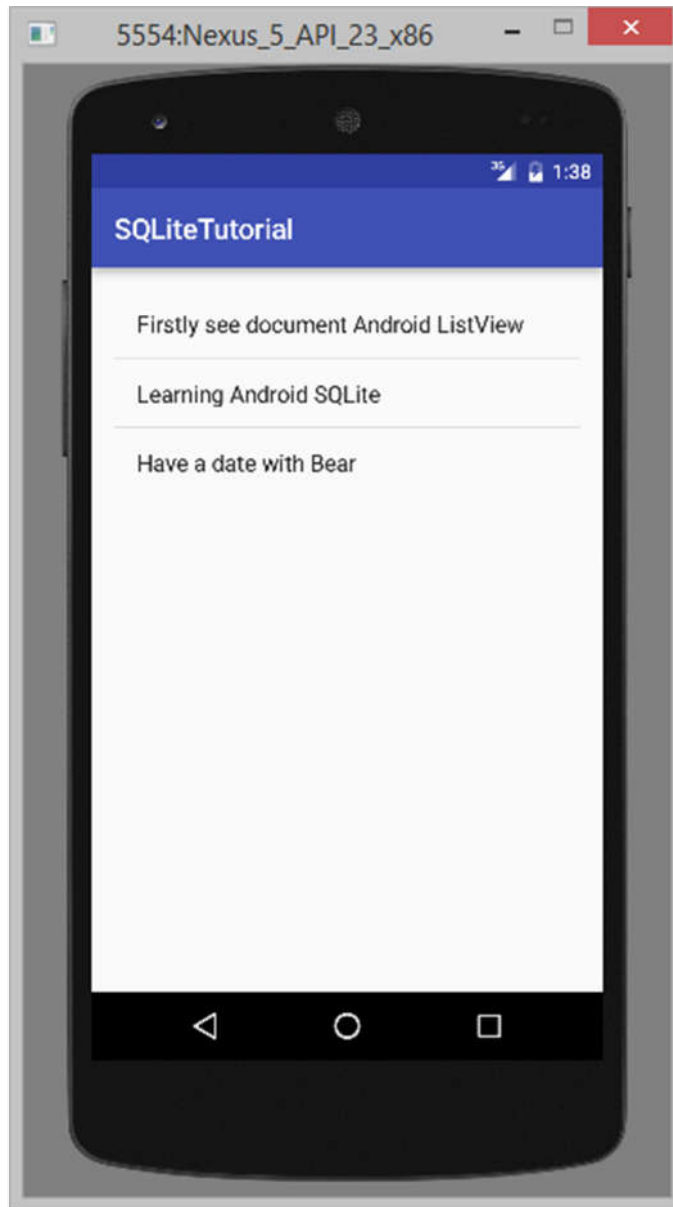
Giả sử rằng bạn tạo ra một ứng dụng chạy trên thiết bị Android của bạn, ứng dụng này ghi chép lại các ghi chú (Note) trong ngày của bạn. Bạn cần một cấu trúc bảng để lưu trữ các ghi chú đó.

- Tên bảng: Note

Tên cột	Kiểu dữ liệu	Giàng buộc	Mô tả
Note_Id	int	Primary Key	Khóa chính
Note_Title	text		Ghi chú ngắn
Note_Content	text		Nội dung ghi chú

Giao diện ứng dụng:





### 3. Các bước thực hiện

Bạn cần tạo một lớp tiện ích để làm việc với cơ sở dữ liệu SQLite, lớp này nên mở rộng từ lớp `SQLiteOpenHelper`. Có 2 phương thức quan trọng mà bạn cần phải ghi đè (override) nó là `onCreate()` và `onUpgrade()`.

Tạo class `MyDatabaseHelper` mở rộng từ `SQLiteOpenHelper`.

Sau khi mở rộng class của bạn từ `SQLiteOpenHelper` bạn cần phải ghi đè lên hai phương thức `onCreate()` và `onUpgrade()`

onCreate() - Những là nơi mà chúng ta cần phải viết để tạo bảng. Nó được gọi (called) khi cơ sở dữ liệu được tạo ra.

onUpgrade() - Phương thức này được gọi khi cơ sở dữ liệu được nâng cấp như thay đổi cấu trúc bảng, thêm giảm buộc cho cơ sở dữ liệu, v..v.

```
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);
    }

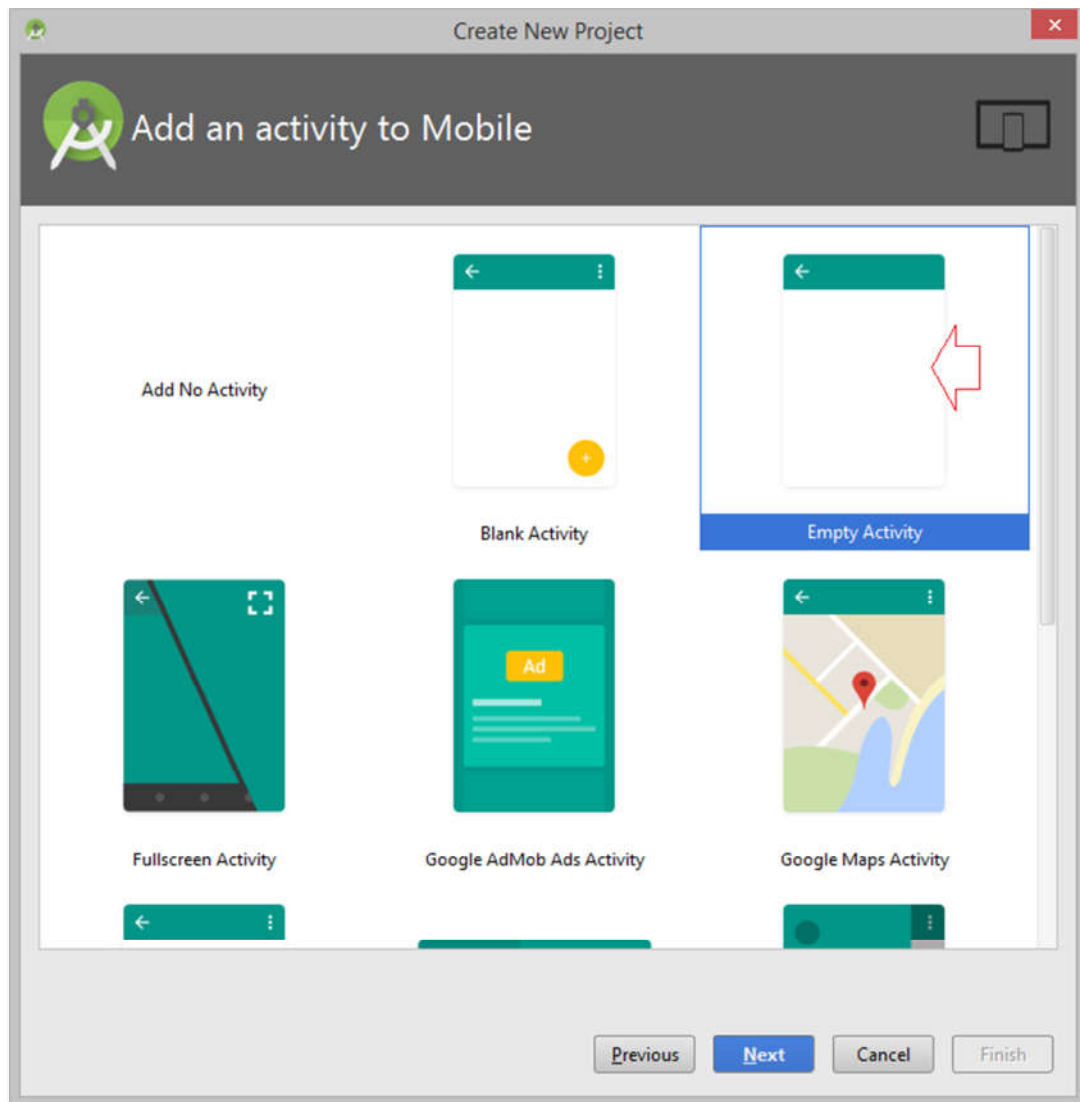
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
        newVersion) {

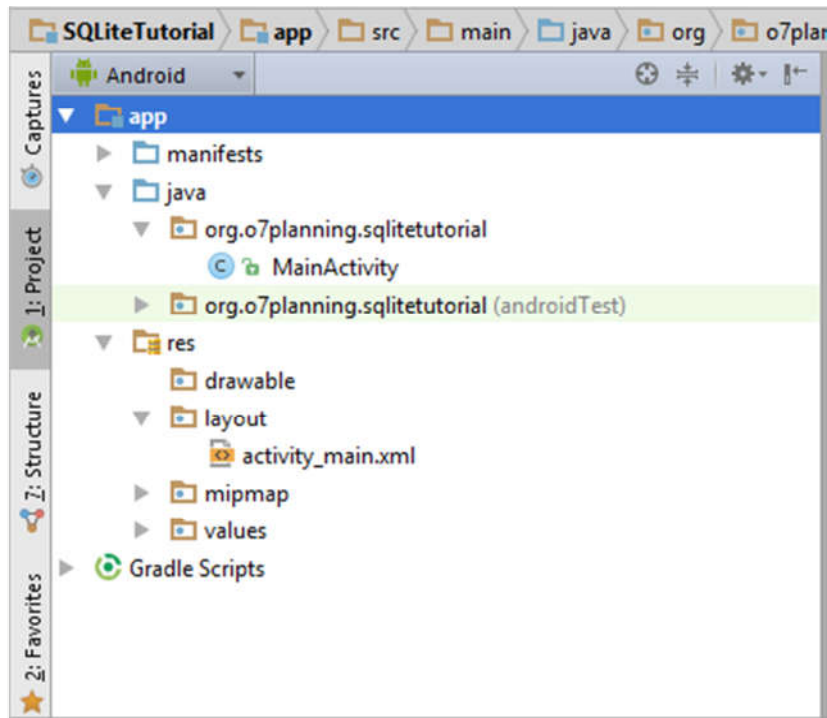
        // Drop table
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NOTE);

        // Recreate
        onCreate(db);
    }
    // ...
}
```

#### 4. Tạo Project

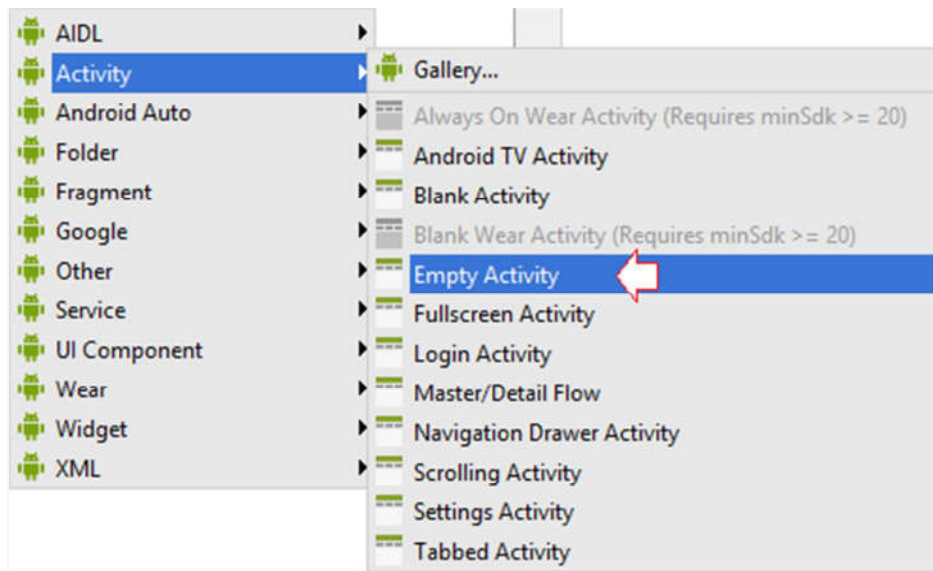
Tạo một "Empty Activity" project có tên SQLiteTutorial để thực hành với các ví dụ.





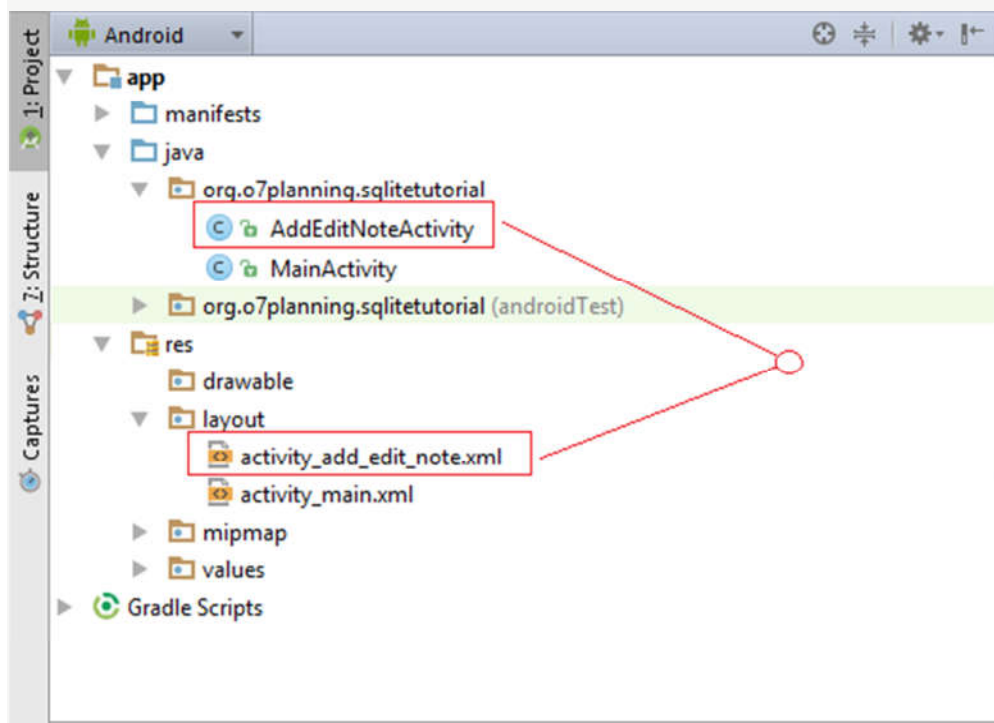
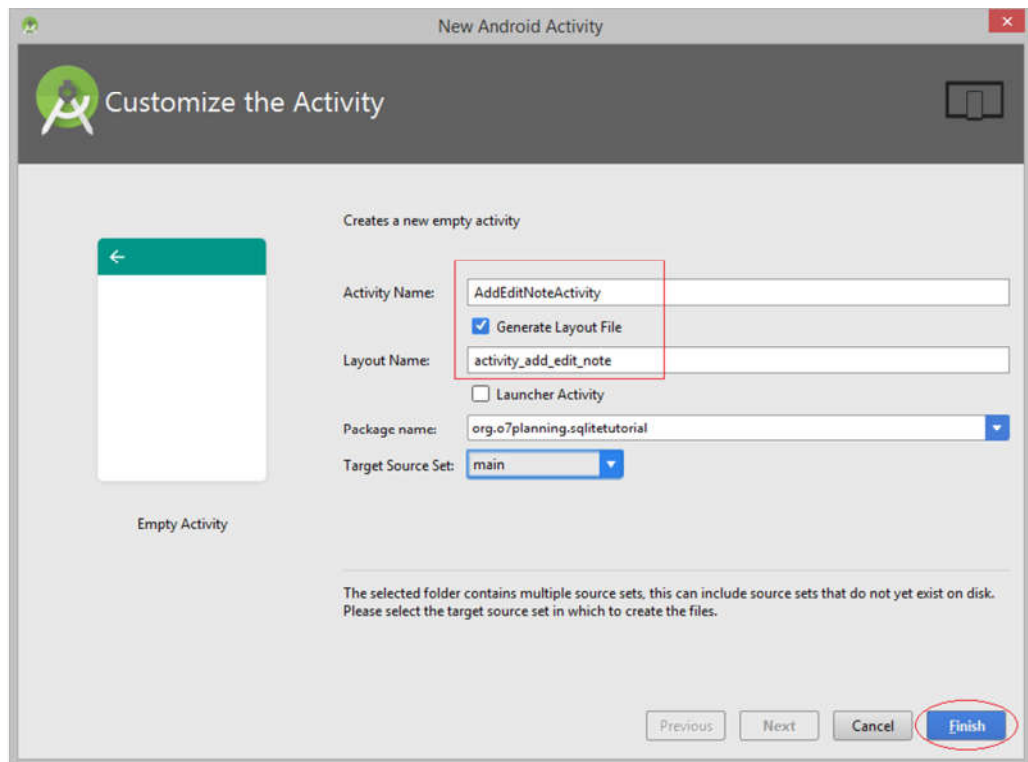
Tạo mới một **Activity** rỗng.

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

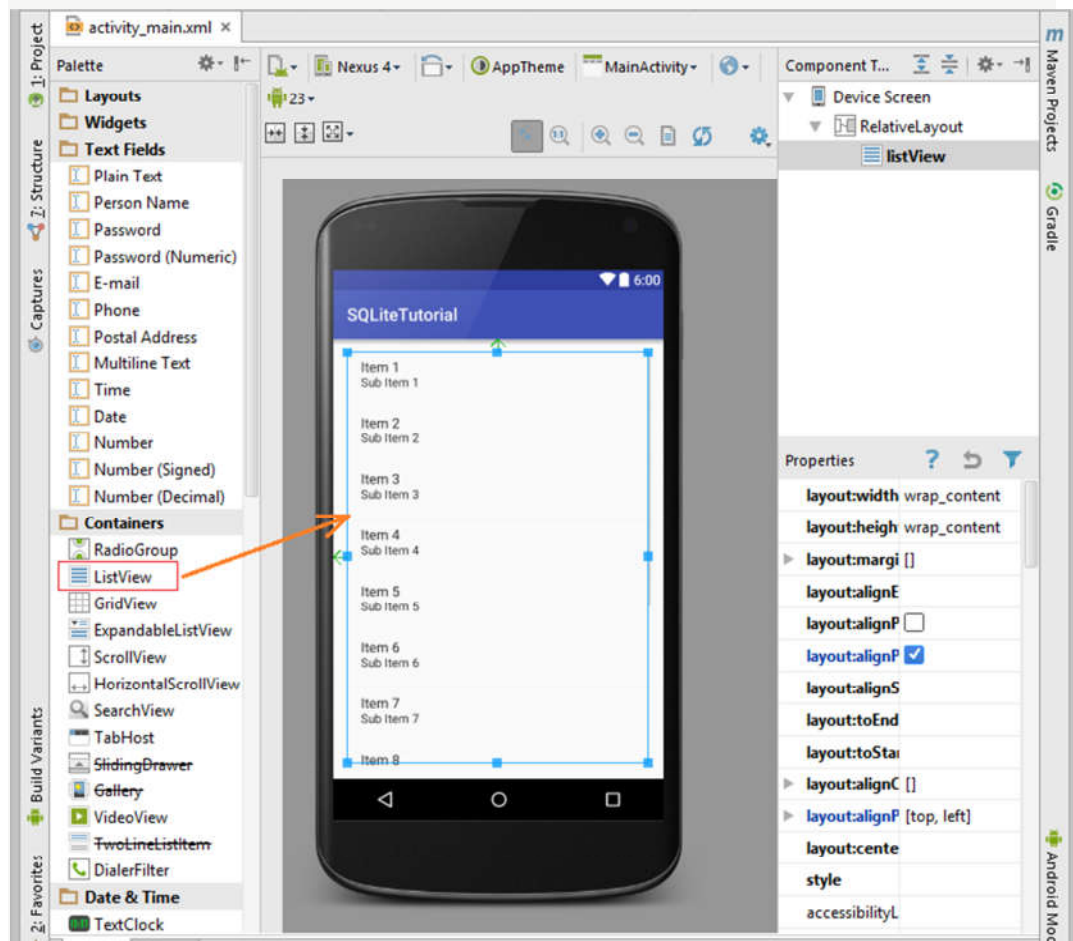


Nhập vào:

- **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" />
```

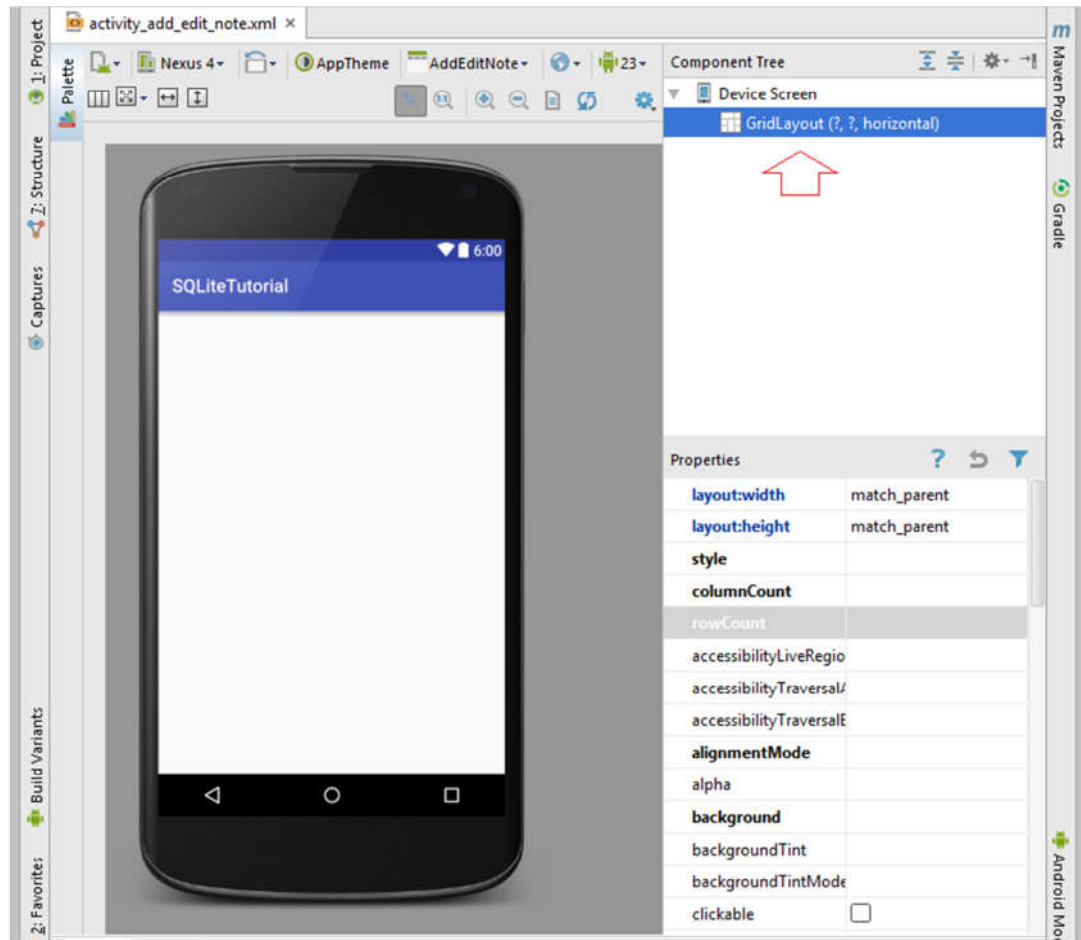


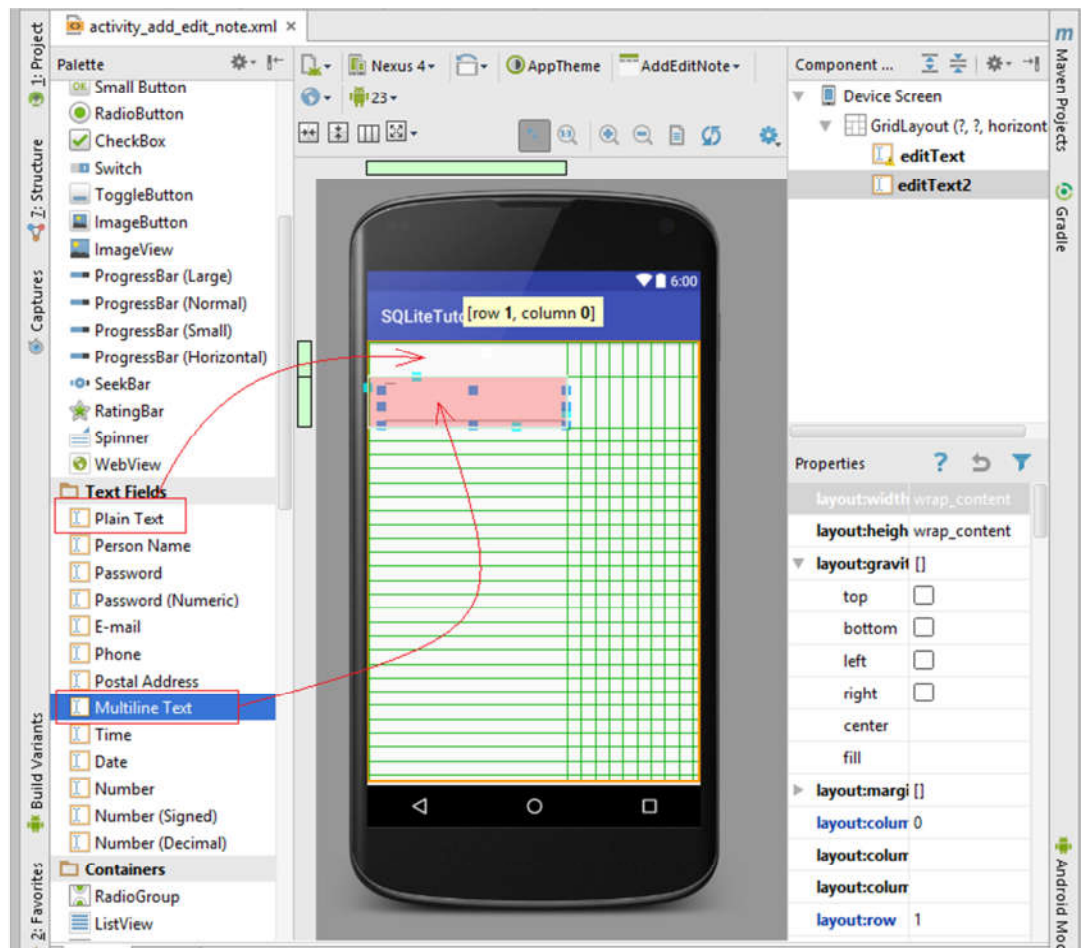
```
</RelativeLayout>
```

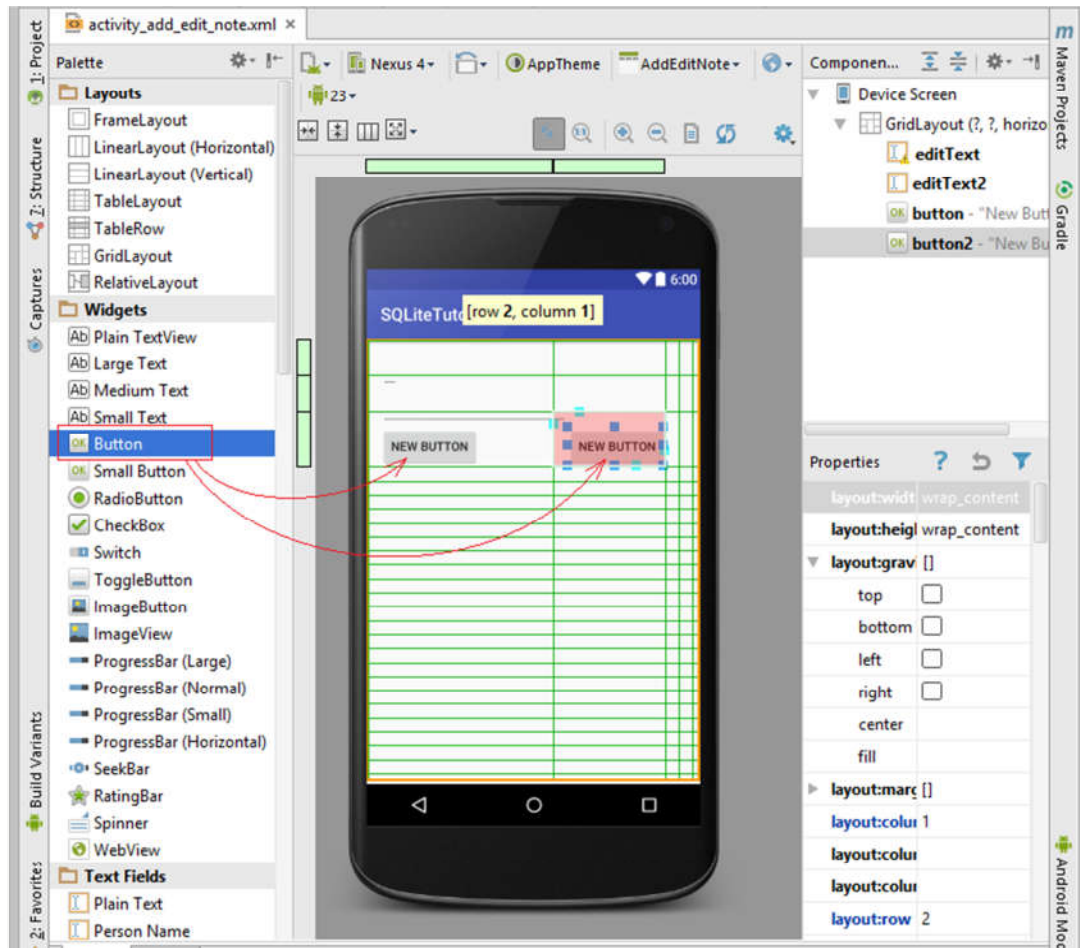
### activity\_add\_edit\_note.xml

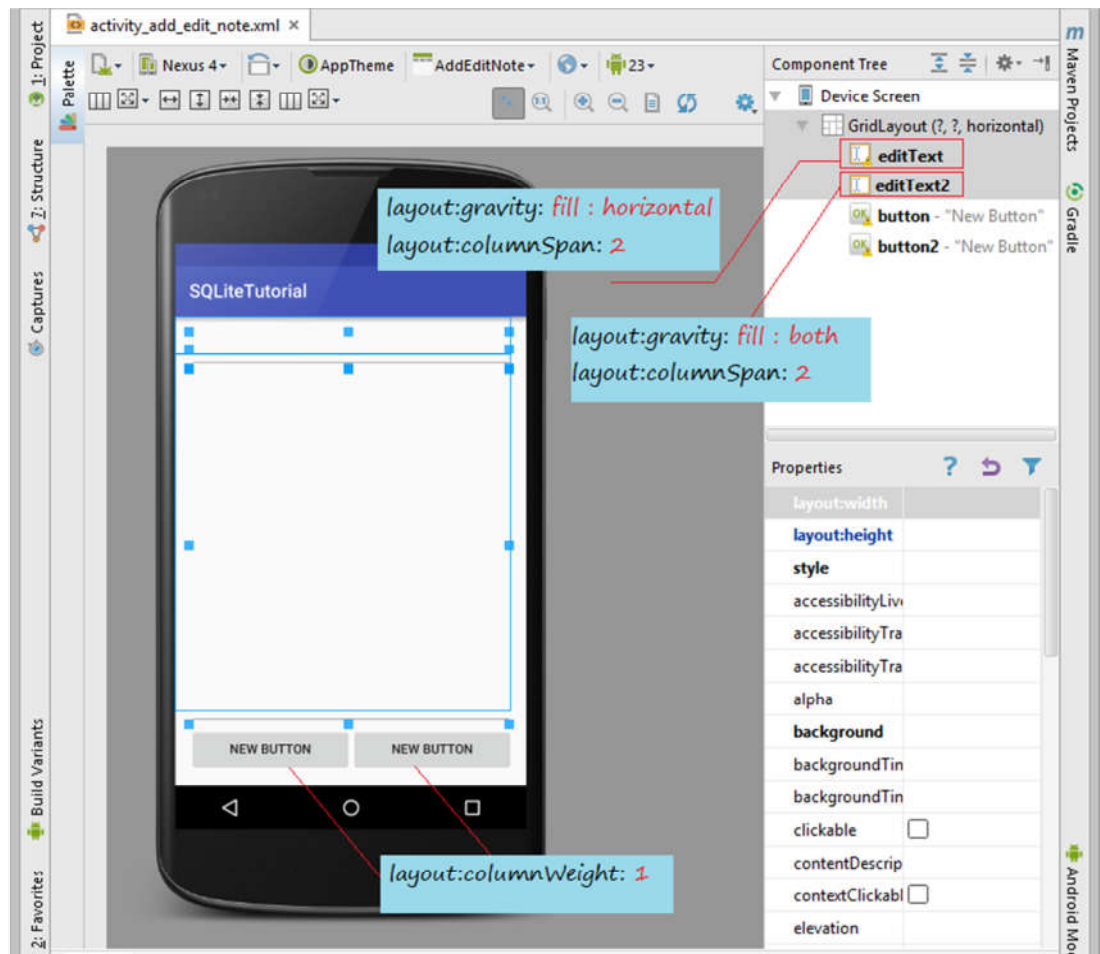
Xem SLIDER các bước thiết kế giao diện:

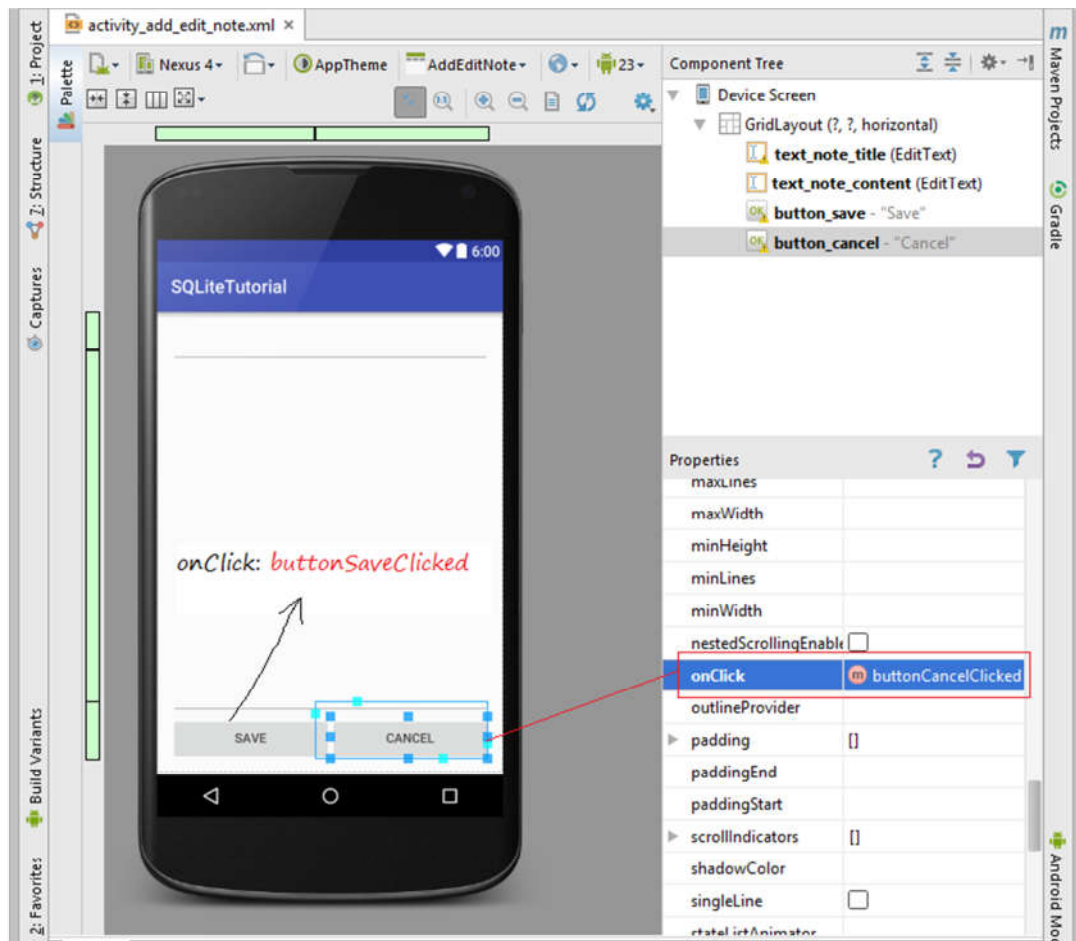
- **SLIDER:**



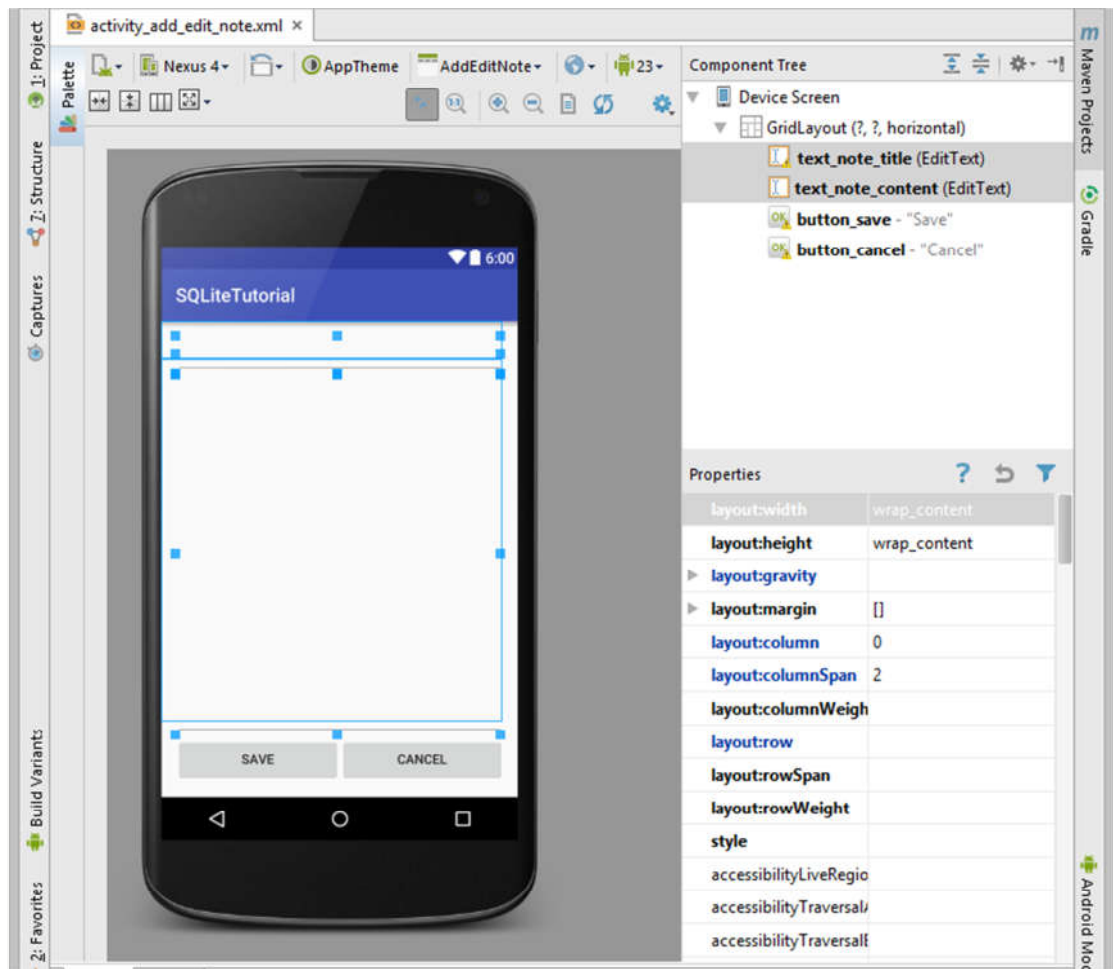








Giao diện:



```
<?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.AddEditNoteAct
ivity">

  <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" />
```

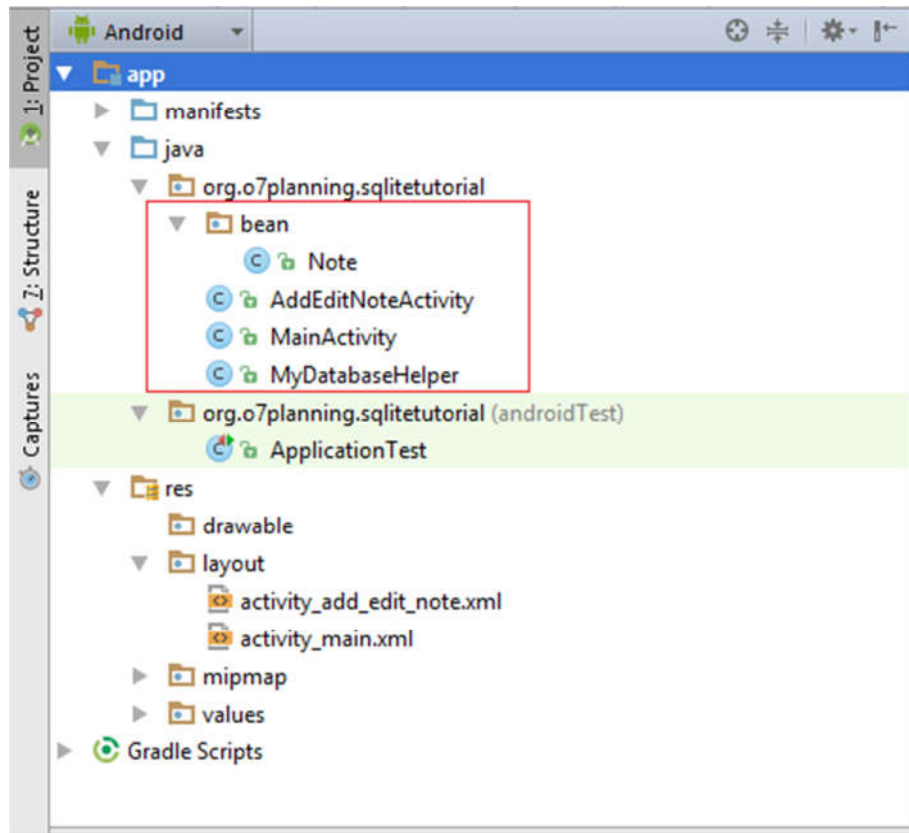
```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="textMultiLine"
    android:ems="10"
    android:id="@+id/text_note_content"
    android:layout_row="1"
    android:layout_column="0"
    android:layout_gravity="fill"
    android:layout_columnSpan="2" />

<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";

```

```

// Phiên bản
private static final int DATABASE_VERSION = 1;

// Tên cơ sở dữ liệu.
private static final String DATABASE_NAME = "Note_Manager";

// Tên bảng: 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);
}

// Tạo các bảng.
@Override
public void onCreate(SQLiteDatabase db) {
    Log.i(TAG, "MyDatabaseHelper.onCreate ... ");
    // Script tạo bảng.
    String script = "CREATE TABLE " + TABLE_NOTE + "("
        + COLUMN_NOTE_ID + " INTEGER PRIMARY KEY, " +
COLUMN_NOTE_TITLE + " TEXT, "
        + COLUMN_NOTE_CONTENT + " TEXT" + ")";
    // Chạy lệnh tạo bảng.
    db.execSQL(script);
}

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

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

    // Hủy (drop) bảng cũ nếu nó đã tồn tại.
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NOTE);

    // Và tạo lại.
    onCreate(db);
}

// Nếu trong bảng Note chưa có dữ liệu,
// Tròn vào mặc định 2 bản ghi.
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());

    // Tràn một dòng dữ liệu vào bảng.
    db.insert(TABLE_NOTE, null, values);

    // Đóng kết nối database.
    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);

        // Duyệt trên con trỏ, và thêm vào danh sách.
        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));

                // Thêm vào danh sách.
                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.AdapterView.OnItemClickListener;
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);

    // Định nghĩa một Adapter.
    // 1 - Context
    // 2 - Layout cho các dòng.
    // 3 - ID của TextView mà dữ liệu sẽ được ghi vào
    // 4 - Danh sách dữ liệu.

    this.listViewAdapter = new ArrayAdapter<Note>(this,
        android.R.layout.simple_list_item_1,
        android.R.id.text1, this.noteList);

    // Đăng ký Adapter cho ListView.
    this.listView.setAdapter(this.listViewAdapter);

    // Đăng ký Context menu cho ListView.
    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, có phản hồi.
            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, có phản hồi.
            this.startActivityForResult(intent,MY_REQUEST_CODE);
        }
        else if(item.getItemId() == MENU_ITEM_DELETE){
            // Hỏi trước khi xóa.
            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;
    }

    // Người dùng đồng ý xóa một Note.
    private void deleteNote(Note note) {
        MyDatabaseHelper db = new MyDatabaseHelper(this);
        db.deleteNote(note);
        this.noteList.remove(note);
        // Refresh ListView.
        this.listViewAdapter.notifyDataSetChanged();
    }

```

```

    }

    // Khi AddEditNoteActivity hoàn thành, nó gửi phản hồi lại.
    // (Nếu bạn đã start nó bằng cách sử dụng
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);
                // Thông báo dữ liệu thay đổi (Để refresh
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());
    }
}

// Người dùng Click vào nút Save.
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;
    // Trở lại MainActivity.
    this.onBackPressed();
}

// Khi người dùng Click vào button Cancel.
public void buttonCancelClicked(View view) {
    // Không làm gì, trở về MainActivity.
    this.onBackPressed();
}

```

```
}

// Khi Activity này hoàn thành,
// có thể cần gửi phản hồi gì đó về cho Activity đã gọi nó.
@Override
public void finish() {

    // Chuẩn bị dữ liệu Intent.
    Intent data = new Intent();
    // Yêu cầu MainActivity refresh lại ListView hoặc không.
    data.putExtra("needRefresh", needRefresh);

    // Activity đã hoàn thành OK, trả về dữ liệu.
    this.setResult(Activity.RESULT_OK, data);
    super.finish();
}
}
```

## 6. Chạy ứng dụng

