Подулючение БД SQLite к Андроид приложению

Есть много уроков о том, как создать и редактировать базы данных SQLite в android приложениях, но как использовать готовую базу данных в приложении, как импортировать созданные базы данных SQLite в android проект?

В самом деле, многие разработчики сталкиваются с проблемами ввода данных, когда речь идет о тестирование существующей базы данных, много людей не имеют времени, чтобы создать новую базу и реализовать все методы CRUD, они просто хотят использовать существующую SQLite базу данных .

В этом уроке вы узнаете, как создать базу данных SQLite с помощью SQLite Software Manager, и импортировать созданную БД в андроид-проект, используя  Android SQLite API.

https://youtu.be/iE\_bsPfB00w

Исходный код проекта:

<LinearLayout 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:orientation="vertical"

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

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

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

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

tools:context=".CopyDbActivity">

<Button

android:id="@+id/button01"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Copy Database"></Button>

</LinearLayout>

import android.database.Cursor;

import android.database.SQLException;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

import java.io.IOException;

public class CopyDbActivity extends AppCompatActivity {

Cursor c = null;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_copy\_db);

((Button) findViewById(R.id.button01)).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

DatabaseHelper myDbHelper = new DatabaseHelper(CopyDbActivity.this);

try {

myDbHelper.createDataBase();

} catch (IOException ioe) {

throw new Error("Unable to create database");

}

try {

myDbHelper.openDataBase();

} catch (SQLException sqle) {

throw sqle;

}

Toast.makeText(CopyDbActivity.this, "Success", Toast.LENGTH\_SHORT).show();

c = myDbHelper.query("EMP\_TABLE", null, null, null, null, null, null);

if (c.moveToFirst()) {

do {

Toast.makeText(CopyDbActivity.this,

"\_id: " + c.getString(0) + "\n" +

"E\_NAME: " + c.getString(1) + "\n" +

"E\_AGE: " + c.getString(2) + "\n" +

"E\_DEPT: " + c.getString(3),

Toast.LENGTH\_LONG).show();

} while (c.moveToNext());

}

}

});

}

}

import android.content.Context;

import android.database.Cursor;

import android.database.SQLException;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteException;

import android.database.sqlite.SQLiteOpenHelper;

import android.util.Log;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

/\*\*

\* Created by Aylar-HP on 12/10/2015.

\*/

public class DatabaseHelper extends SQLiteOpenHelper {

String DB\_PATH = null;

private static String DB\_NAME = "extenalDB";

private SQLiteDatabase myDataBase;

private final Context myContext;

public DatabaseHelper(Context context) {

super(context, DB\_NAME, null, 10);

this.myContext = context;

this.DB\_PATH = "/data/data/" + context.getPackageName() + "/" + "databases/";

Log.e("Path 1", DB\_PATH);

}

public void createDataBase() throws IOException {

boolean dbExist = checkDataBase();

if (dbExist) {

} else {

this.getReadableDatabase();

try {

copyDataBase();

} catch (IOException e) {

throw new Error("Error copying database");

}

}

}

private boolean checkDataBase() {

SQLiteDatabase checkDB = null;

try {

String myPath = DB\_PATH + DB\_NAME;

checkDB = SQLiteDatabase.openDatabase(myPath, null, SQLiteDatabase.OPEN\_READONLY);

} catch (SQLiteException e) {

}

if (checkDB != null) {

checkDB.close();

}

return checkDB != null ? true : false;

}

private void copyDataBase() throws IOException {

InputStream myInput = myContext.getAssets().open(DB\_NAME);

String outFileName = DB\_PATH + DB\_NAME;

OutputStream myOutput = new FileOutputStream(outFileName);

byte[] buffer = new byte[10];

int length;

while ((length = myInput.read(buffer)) > 0) {

myOutput.write(buffer, 0, length);

}

myOutput.flush();

myOutput.close();

myInput.close();

}

public void openDataBase() throws SQLException {

String myPath = DB\_PATH + DB\_NAME;

myDataBase = SQLiteDatabase.openDatabase(myPath, null, SQLiteDatabase.OPEN\_READONLY);

}

@Override

public synchronized void close() {

if (myDataBase != null)

myDataBase.close();

super.close();

}

@Override

public void onCreate(SQLiteDatabase db) {

}

@Override

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

if (newVersion > oldVersion)

try {

copyDataBase();

} catch (IOException e) {

e.printStackTrace();

}

}

public Cursor query(String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy) {

return myDataBase.query("EMP\_TABLE", null, null, null, null, null, null);

}

}