**西 安 邮 电 大 学**

**（计算机学院）**

《Android应用开发》课内实验报告

# 实验六：基于SQLite的通讯录

**专业名称： 计算机科学与技术**

**班 级： 计科1608**

**学 号： 05168066**

**学生姓名： 祁泽宇**

**指导教师： 张庆生**

**起止时间： 2019年5月21日**

1. 课程设计目的
2. 掌握SQLiteOpenHelper类的结构
3. 掌握基于SQLite数据库的开发过程
4. 掌握Content Provider发布数据的方法
5. 掌握Content Prpvider获取数据的方法
6. 实验环境

Jdk:1,8,0\_40

IDE:Android Studio

1. 实验内容

实现基于SQLite数据库的通信录应用，运行结果如图1-3所示。通过单击增加图标打开通信录界面，通过单击通信录中的各条信息可删除选中项。

1. 实验过程及分析
2. 建立数据库操作类DataBaseHelper,实现增删改查

package com.example.sql;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
public class DatabaseHelper extends SQLiteOpenHelper {  
 private static final String DB\_NAME = "MyRelation.db";  
 private static final String TABLE\_NAME = "relation";  
 private static final String CREATE\_TABLE = "create table relation(\_id integer primary key autoincrement,name text,tel text,groupName text);";  
 //字段id为主键，自动增长  
  
 private SQLiteDatabase db; //创建SQLiteDatabase实例  
 DatabaseHelper(Context context)//构造方法  
 {  
 super(context,DB\_NAME,null,2);  
 }  
 public void insert(ContentValues values){//插入  
 SQLiteDatabase db = getWritableDatabase();  
 db.insert(TABLE\_NAME,null,values);  
 db.close();  
 }  
 public void del(int id){//删除  
 if(db==null)  
 {  
 db = getWritableDatabase();  
 db.delete(TABLE\_NAME,"\_id = ?",new String[]{String.valueOf(id)});  
  
 }  
 }  
 public Cursor query(){ //查询  
 SQLiteDatabase db = getWritableDatabase();  
 Cursor cursor = db.query(TABLE\_NAME,null,null,null,null,null,null);  
 return cursor;  
 }  
 public void close(){ //关闭数据库  
 if(db != null) db.close();  
 }  
  
  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 this.db = db;  
 db.execSQL(CREATE\_TABLE);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
  
 }  
}

1. 修改主布局文件如下，并将下列空出代码部分补全

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"  
 android:orientation="vertical"  
 >  
  
 <TextView  
 android:id="@+id/title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginTop="0dp"  
 android:text="通讯录"  
 android:textSize="25dp"  
 android:textStyle="bold" />  
  
 <ListView  
 android:id="@+id/listview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="0dp"  
 android:layout\_marginTop="86dp">  
  
  
 </ListView>  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="0dp"  
 android:layout\_marginTop="35dp"  
 android:onClick="add"  
 android:text="添加" />  
  
</RelativeLayout>

1. 修改主Activity文件MainActivity.class，代码如下所示：

package com.example.sql;  
  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.database.Cursor;  
import android.support.v7.app.AlertDialog;  
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 {  
 private ListView listView;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 listView = (ListView) findViewById(R.id.listview);  
 getRelationFromDB();  
 }  
  
 /\*\*  
 \* 该函数的流程如下  
 \* 首先创建一个DatabaseHelper对象，DatabaseHelper是一个数据库工作类  
 \* 创建一个Cursor用于查询  
 \* 建立适配器，实现数据与对象的绑定  
 \* 将listView与适配器绑定  
 \* 设置listView的点击监听器  
 \* 实现警告对话框的实现，点击确定的话就删除，点取消的话就返回  
 \* 操作完后关闭数据库对象  
 \*/  
 private void getRelationFromDB(){  
 //从数据加载联系人信息到列表视图  
 final DatabaseHelper dbHelper = new DatabaseHelper(this);  
 Cursor cursor = dbHelper.query(); //光标查询？  
 String[] from = {"\_id","name","tel","groupName"}; //数据库字段数组  
 int[] to = {R.id.\_id,R.id.name,R.id.tel,R.id.group}; //显示布局id数组  
 SimpleCursorAdapter scadapter = new SimpleCursorAdapter(this,R.layout.relationlist,cursor,from,to);  
 //建立一个适配器适配器是一个连接数据和AdapterView（ListView就是一个典型的AdapterView，后面还会学习其他的）的桥梁，  
 // 通过它能有效地实现数据与AdapterView的分离设置，使AdapterView与数据的绑定更加简便，修改更加方便  
 listView.setAdapter(scadapter); //给listView设置适配器，用来匹配xml  
 listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id)  
 {  
 final long temp = id;  
 AlertDialog.Builder adBuilder = new AlertDialog.Builder(MainActivity.this);  
 adBuilder.setMessage("确认要删除记录么？").setPositiveButton("确认", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 dbHelper.del((int) temp);  
 Cursor cursor =dbHelper.query();  
 String[] from ={"\_id","name","tel","groupName"};  
 int[] to = {R.id.\_id,R.id.name,R.id.tel,R.id.group};  
 SimpleCursorAdapter scadapter = new SimpleCursorAdapter(getApplicationContext(),R.layout.relationlist,cursor,from,to);  
 MainActivity.this.listView.setAdapter(scadapter);  
 }  
 }).setNegativeButton("取消", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
  
 }  
 });  
 AlertDialog alertDialog = adBuilder.create(); //创建一个警告对话框  
 alertDialog.show(); //显示警告对话框  
  
 }  
 }); dbHelper.close();  
  
 }  
 public void add(View view){ //增加单击按钮  
 Intent intent = new Intent(MainActivity.this,AddRelationActivity.class);  
 startActivityForResult(intent,0x111);  
  
 }  
 @Override  
 protected void onActivityResult(int requestCode,int resultCode,Intent data)  
 {  
 super.onActivityResult(requestCode,resultCode,data);  
 if(requestCode ==0x111&&requestCode==0x111);  
 getRelationFromDB();// 增加新联系人后返回刷新列表框  
 }  
}

1. 创建列表视图文件relationlist.xml，请将下列空出部分代码补全

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/\_id"/>  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/name"/>  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/tel"/>  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/group"/>  
</LinearLayout>

1. 创建增加联系人界面布局文件addrelation.xml，请将下列空出代码补全

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="姓名"  
 android:textSize="25dp"  
 android:textStyle="bold"/>  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/addName"  
 />  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="电话"  
 android:textSize="25dp"  
 android:textStyle="bold"/>  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/addTel"  
 />  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="所属组"  
 android:textSize="25dp"  
 android:textStyle="bold"/>  
 <Spinner  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/addGroup"  
 android:entries="@array/group"  
 />  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/save"  
 android:text="保存"  
 android:onClick="save"/>  
</LinearLayout>

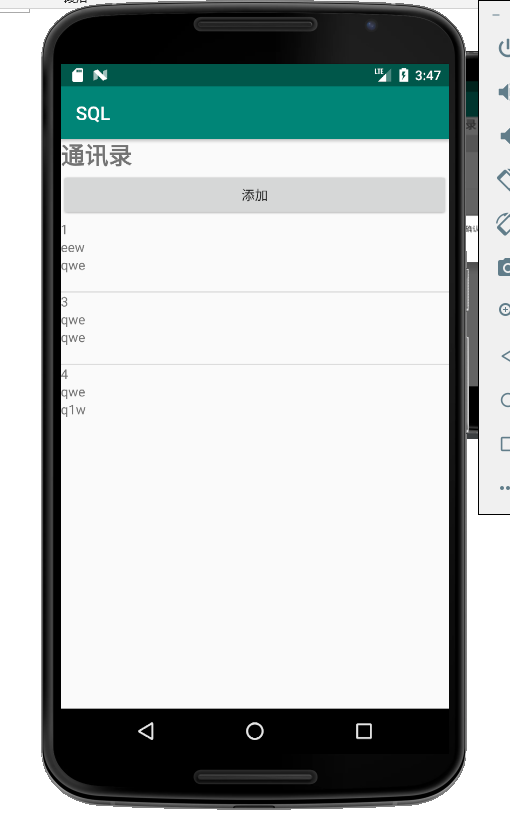
1. 创建增加联系人主界面主Activity文件AddRelationActivity.java，代码如下

package com.example.sql;  
  
import android.content.ContentValues;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
import android.support.v7.app.AlertDialog;  
import android.support.v7.app.AppCompatActivity;  
import android.view.View;  
import android.widget.EditText;  
  
import android.widget.Spinner;  
  
public class AddRelationActivity extends AppCompatActivity {  
 private EditText addName,addTel;  
 private Spinner addGroup;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.addrelation);  
 addName = (EditText) findViewById(R.id.addName);  
 addTel = (EditText) findViewById(R.id.addTel);  
 addGroup = (Spinner) findViewById(R.id.addGroup);  
 }  
 public void save(View view){  
 final ContentValues values = new ContentValues();//ContentValues 和HashTable类似都是一种存储的机制  
 values.put("name",addName.getText().toString());  
 values.put("tel",addTel.getText().toString());  
 final DatabaseHelper dbHelper = new DatabaseHelper(getApplicationContext());  
 final AlertDialog.Builder adBuilder =new AlertDialog.Builder(this);  
 adBuilder.setMessage("确认保存记录么？").setPositiveButton("确认", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 dbHelper.insert(values);  
 Intent intent =getIntent();  
 setResult(0x111,intent);  
 AddRelationActivity.this.finish(); //关闭当前界面，返回首页  
 }  
 }).setNegativeButton("取消", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
  
 }  
 });  
 AlertDialog alertDialog = adBuilder.create(); //创建一个警告对话框  
 alertDialog.show(); //显示警告对话框  
 }  
}

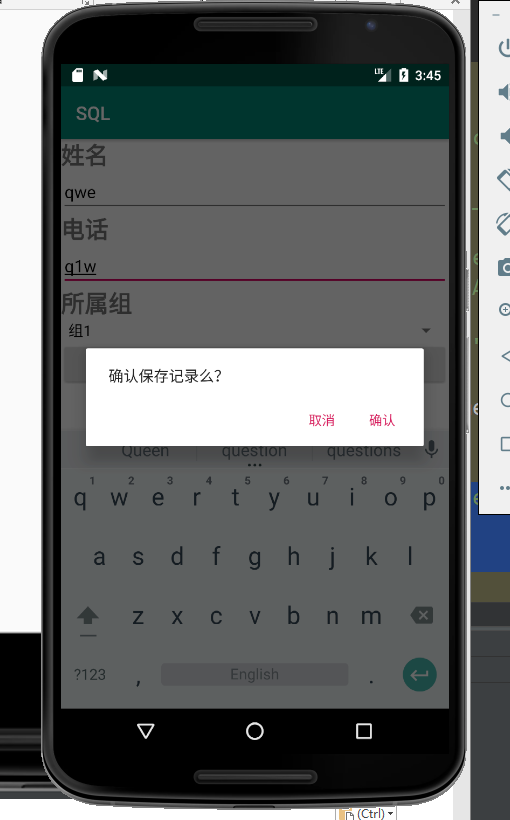
1. 在AndroidManifest.xml文件中添加AddrelationActivity界面应用

<activity android:name=".AddRelationActivity"  
 android:label="@string/app\_name">  
  
</activity>

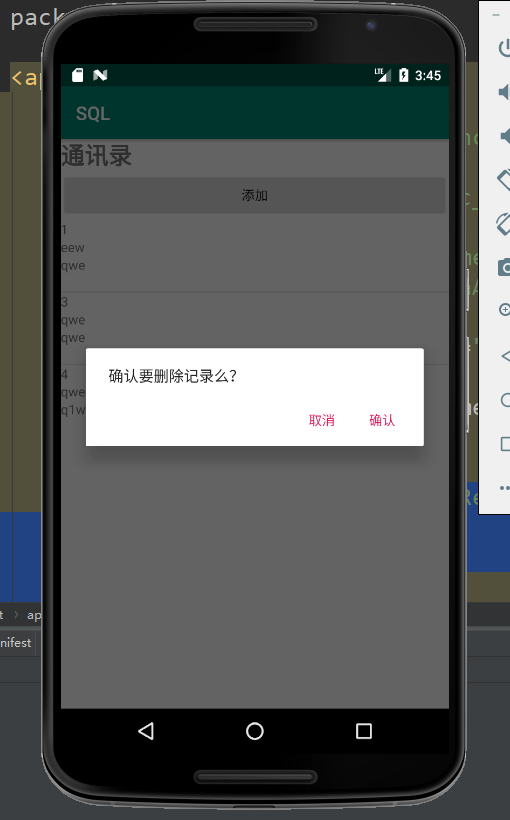
1. 实验结果如图所示



主界面



保存界面



删除界面

1. 总结

本次实验我们学习了基于SQLite的开发，这次实验主要是将数据库与Android结合起来。

SQLite是一款轻量级的关系型数据库，它的运算速度非常快，占用资源很少，通常只需要几百K的内存就足够了，因而特别适合在移动设备上使用。

SQLite的优点如下：

使用 SQLite 只需要带一个动态库，就可以享受它的全部功能，而且那个动态库的尺寸想当小。

SQLite 数据库的核心引擎不需要依赖第三方软件，也不需要所谓的“安装”。

SQLite 数据库中所有的信息（比如表、视图、触发器等）都包含在一个文件夹内，方便管理和维护。

SQLite 目前支持大部分操作系统，不至电脑操作系统更在众多的手机系统也是能够运行，比如：Android和IOS。

SQLite 数据库支持多语言编程接口。

SQLite 数据库通过数据库级上的独占性和共享锁来实现独立事务处理。这意味着多个进程可以在同一时间从同一数据库读取数据，但只能有一个可以写入数据。

SQLite作为一款入门级的数据库，我们只有掌握了它的使用方法才能更好的学习更加专业的数据库。