实验八 数据存储（二）

1、实验目的

- 学习 SQL 数据库的使用
- 学习 ContentProvider 的使用
- 复习 Android 界面编程

2、实验内容

从左至右：初始界面，添加一部分条目，名字不能为空

从左至右：名字不能重复，点击条目显示信息（可修改），长按删除条目。

实现一个生日备忘录，要求实现：

- 使用 SQLite 数据库保存生日的相关信息，并使得每一次运行程序都可以显示出已经存储在数据库里的内容；
- 使用 ContentProvider 来获取手机通讯录中的电话号码。

功能要求：

A. 主界面包含增加生日条目按钮和生日信息列表；

B. 点击“增加条目”按钮，跳转到下一个 Activity 界面，界面中包含三个信息输入框（姓名、生日、礼物）和一个“增加”按钮，**姓名字段不能为空且不能重复**；

C. 在跳转到的界面中，输入生日的相关信息后，点击“增加”按钮返回到主界面，~~**此时，主界面中应更新列表，增加相应的生日信息**~~；

D. 主界面列表点击事件：

- 点击条目：

  弹出对话框，对话框中显示该条目的信息，并允许修改；

  对话框下方显示该寿星电话号码**（如果手机通讯录中有的话，如果没有就显示“无”）**

  点击“保存修改”按钮，更新主界面生日信息列表。

- 长按条目：

  弹出对话框显示是否删除条目；

  点击“是”按钮，删除该条目，并更新主界面生日列表。

3、基础知识

A. SQLite 数据库的使用

第一步：创建数据库的类，**使用 SQLiteOpenHelper 的子类。**

```
public class myDB extends SQLiteOpenHelper {
    private static final String DB_NAME = "Contacts.db";
    private static final String TABLE_NAME = "Contacts";
    private static final int DB_VERSION = 1;
```

第二步：创建数据库，直接执行 SQL 语句即可。

```
public void onCreate(SQLiteDatabase db) {
    String CREATE_TABLE = "create table " + TABLE_NAME
            + " (_id integer primary key , "
            + "name text , "
            + "birth text , "
            + "gift text);";
    db.execSQL(CREATE_TABLE);
}
```

第三步：重写 onUpgrade 函数，虽然本次实验用不到，但是需要重写这个函数才能实例化。

```
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // To Do
}
```

第四步：实现 insert，update 和 delete 三种操作。实现的方法有两个，选择任意一种即可，不作特殊要求。

1） 第一种方法：

使用 execSQL 直接执行对应的增删改 SQ 语句，例如：

```
SQLiteDatabase db = getWritableDatabase();
String insert_sql = "INSERT INTO <表名>(<列 1>,<列 2>,.....) values(<值 1>,<值
2>,.....)";
db.execSQL(insert_sql);
```

2） 第二种方法：

借助 ContentValues 和 Where 语句来实现操作：

Insert：

```
public void insert(String name, String birth, String gift) {
    SQLiteDatabase db = getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("name", name);
    values.put("birth", birth);
    values.put("gift", gift);
    db.insert(TABLE_NAME, null, values);
    db.close();
}
```

Update：

```
public void update(String name, String birth, String gift) {
    SQLiteDatabase db = getWritableDatabase();
    String whereClause = "**** = ?"; // 主键列名 = ?
    String[] whereArgs = { **** };   // 主键的值
    ContentValues values = new ContentValues();
    values.put("name", name);
    values.put("birth", birth);
    values.put("gift", gift);
    db.update(TABLE_NAME, values, whereClause, whereArgs);
    db.close();
}
```

Delete：

```
public void delete(String name) {
    SQLiteDatabase db = getWritableDatabase();
    String whereClause = "**** = ?"; // 主键列名 = ?
    String[] whereArgs = { **** };   // 主键的值

    db.delete(TABLE_NAME, whereClause, whereArgs);
    db.close();
}
```

PS：注意，不管哪一种方法，都要在最开始 getWritableDatabase()一下。

第五步：实现查询操作。同样有两种方法，任选一种即可，不作要求。

1） 第一种方法（示例）

```
Cursor cursor = db.rawQuery("select * from <TABLE_NAME>", null);
```

2） 第二种方法（示例）

```
Cursor cursor = db.query(TABLE_NAME, new String[]{"列 1","列 2",.....},
                         null, null, null, null, null);
```

同样的，**不管哪一种方法，都要在最开始 getWritableDatabase()一下。**

关于 Cursor，可以参考一下这个博客，了解更多的内容。

B. ContentProvider 的使用

第一步：在 AndroidManifest 中声明权限

```
<uses-permission android:name="android.permission.READ_CONTACTS"/>
```

第二步：使用 getContentResolver 方法读取联系人列表

```
Cursor cursor = getContentResolver().query(ContactsContract.Contacts.CONTENT_URI,
null, null, null, null);
```

PS：关于获取通讯录权限，在实际操作中，有些手机（比如：API 19 的）是在程序运行的时候进行询问是否给予权限；有些手机（比如：API 23 的），不会提示，需要你自己到设置界面下开启本应用程序的获取通讯录权限。因此，如果你在点击表单的时候发生闪退，基本就是获取通讯录的权限没有打开。

第三步：判断某条联系人信息中是否有电话号码

```
int isHas = Integer. parseInt
(cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts.
HAS_PHONE_NUMBER )));
```

第四步：取出该条离联系人的信息中的电话号码

```
Cursor phone =
getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI, null,
ContactsContract.CommonDataKinds.Phone.CONTACT_ID + "=" + ContactId, null, null);
String Number = "";
while(phone.moveToNext()) {
    Number +=
phone.getString(phone.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER))
+ " ";
}
```

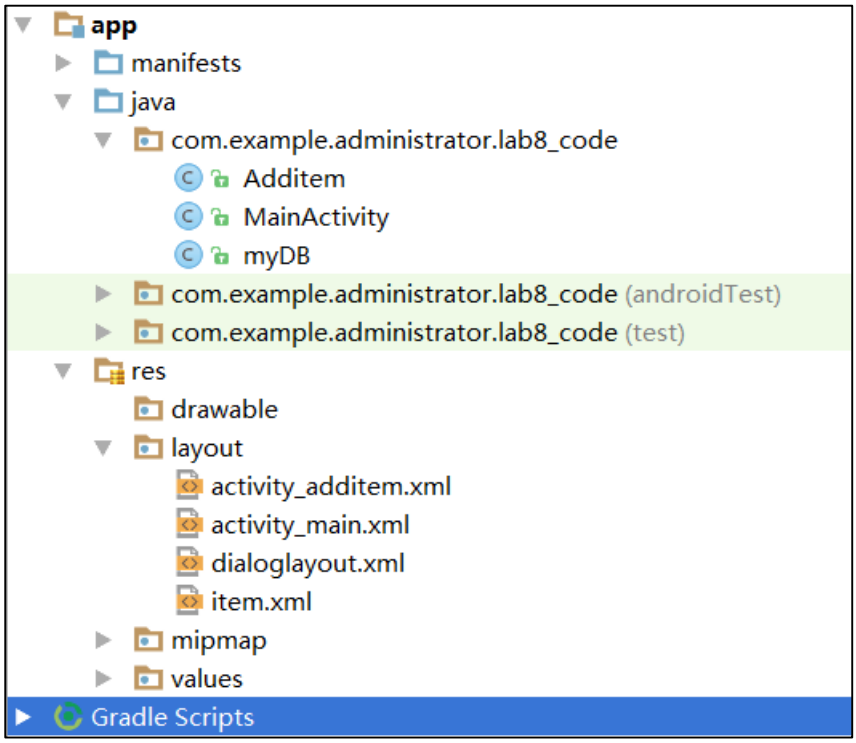
C. 自定义对话框的实现

使用 LayoutInflater 类，如下所示：

```
LayoutInflater factor = LayoutInflater.from(MainActivity.this);
View view_in = factor.inflate(R.layout.dialoglayout, null);
final AlertDialog.Builder alertDialog1 = new AlertDialog.Builder(MainActivity.this);
              alertDialog1.setView(view_in);
```

进行上述操作后，我们就能够将 dialoglayout 这个布局中的内容全部显示在弹出的对话框（alertDialog1）中，并可以对这个布局中的元素进行操作。

4、参考目录



5、检查内容

是否完成实验内容中的所有要求。

6、提交说明

1、 Deadline：下一次实验课前晚上 23:59；

2、 提交作业地址：ftp://edin.sysu.edu.cn

3、 文件命名及格式要求：学号_姓名_labX.zip（姓名中文拼音即可）

4、 目录结构

```
15331111_huashen_lab1 --
                        |
                        --  lab1实验报告.pdf
                        |
                        --  lab1_code（包含项目代码文件）
```

提交之前先 clean！