
第 5 章 SQLite 数据库——应用案例

案例 5-1 RegistUserInfo

一、案例描述

1、考核知识点

030005003: 数据库的常用操作

030005006: ListView 控件的使用

2、练习目标

- 掌握 SQLite 数据库的创建，增删改查操作
- 掌握 ListView 的用法，使用 BaseAdapter 为 ListView 绑定数据

3、需求分析

当一个应用需要将本地数据储存下来时就可以使用 SQLite 数据库，最常见的保存数据情景就是应用的账号和密码，该案例就通过对账号密码的注册、登录、删除、修改功能对 SQLite 数据库做一个详细的讲解，并将数据通过 BaseAdapter 适配器展示在 ListView 上面。

4、设计思路（实现原理）

- 1) 使用 SQLite 储存注册的账号密码和电话号码；
- 2) 点击登录时将输入的信息与数据库中的信息比对，如果存在则跳转至第二个界面；
- 3) 在第二个页面点击所有账户按钮则在按钮上方显示 ListView 展示出注册过的所有账号，并隐藏所有账户按钮显示删除所有账户按钮；
- 4) 点击 ListView 的条目弹出 Toast 显示当前条目的账号和密码信息，点击 ListView 条目的删除按钮则从数据库中删除当前条目的账号和密码；
- 5) 点击修改密码按钮弹出对话框，通过输入的账号密码信息与数据库中比对，如果正确则修改数据库储存的信息。

二、案例实现

(1) 创建第一个界面

创建一个名为“RegistUserInfo”的程序并创建第一个界面 MainActivity，该 Activity 用于账号密码的注册和登录。程序主界面如图 5-1 所示。



图5-1 主界面

MainActivity 界面对应的布局文件（activity_main.xml）如下所示：

```
<?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:background="@drawable/bj"
    tools:context=".MainActivity">
    <LinearLayout
        android:id="@+id/user_psw_ll"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@+id/bottom_ll"
        android:layout_marginBottom="20dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:orientation="vertical">
        <RelativeLayout
            android:id="@+id/user_rl"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@drawable/cz_shade_ll">
            <EditText
                android:id="@+id/name_et"
                style="@style/editStyle"
                android:layout_centerVertical="true"
                android:layout_toRightOf="@+id/userHint_tv"
                android:ems="10" />
        </RelativeLayout>
    </LinearLayout>
</RelativeLayout>
```

```
<TextView
    android:id="@+id/userHint_tv"
    style="@style/textStyle"
    android:layout_centerVertical="true"
    android:text="用户名:" />
</RelativeLayout>
<RelativeLayout
    android:id="@+id/nickname_rl"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:background="@drawable/cz_shade_ll">
    <EditText
        android:id="@+id/password_et"
        style="@style/editStyle"
        android:layout_centerVertical="true"
        android:layout_toRightOf="@+id/passwordHint_tv"
        android:ems="10" />
    <TextView
        android:id="@+id/passwordHint_tv"
        style="@style/textStyle"
        android:text="密  码:" />
</RelativeLayout>
</LinearLayout>
<LinearLayout
    android:id="@+id/bottom_ll"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_marginBottom="10dp"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp"
    android:orientation="horizontal">
    <Button
        android:id="@+id/login_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="39dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:background="@drawable/cz_shade_button"
        android:onClick="login">
```

```

        android:text="登录"
        android:textColor="#FFFFFF"
        android:textSize="18sp" />
    <Button
        android:id="@+id/register_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="38dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:background="@drawable/cz_shade_button"
        android:onClick="register"
        android:text="注册"
        android:textColor="#FFFFFF"
        android:textSize="18sp" />
</LinearLayout>
</RelativeLayout>

```

接下来将 `TextView` 和 `EditText` 的属性抽取出做单独样式，存放在 `res/values` 下的 `styles.xml` 文件内，`TextView` 和 `EditText` 的样式如下所示：

```

<style name="editStyle" parent="android:Widget.Holo.Light.EditText">
    <item name="android:background">@null</item>
    <item name="android:layout_marginRight">5dp</item>
    <item name="android:layout_width">match_parent</item>
    <item name="android:layout_height">wrap_content</item>
</style>
<style name="textStyle" parent="android:Widget.Holo.Light.TextView">
    <item name="android:layout_marginBottom">20dp</item>
    <item name="android:layout_marginLeft">5dp</item>
    <item name="android:layout_marginRight">5dp</item>
    <item name="android:layout_marginTop">20dp</item>
    <item name="android:layout_width">wrap_content</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:layout_centerVertical">>true</item>
    <item name="android:textSize">18sp</item>
</style>

```

在 `drawable` 文件夹下创建两个 XML 文件，分别用于自定义文本框显示效果与按钮背景，具体代码如下所示。

文本框显示效果 (`cz_shade_ll.xml`) 代码如下：

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#FFFFFF" />
    <stroke
        android:width="3dp"

```

```

        android:color="#ff6347" />
    <corners
        android:bottomLeftRadius="5dp"
        android:bottomRightRadius="5dp"
        android:topLeftRadius="5dp"
        android:topRightRadius="5dp" />
</shape>

```

按钮背景（cz_shade_button.xml）代码如下：

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <corners
        android:bottomLeftRadius="5dp"
        android:bottomRightRadius="5dp"
        android:topLeftRadius="5dp"
        android:topRightRadius="5dp" />
    <gradient
        android:angle="270"
        android:centerColor="#2368e4"
        android:centerX="0.5"
        android:centerY="0.5"
        android:endColor="#0c2997"
        android:startColor="#1139b4" />
</shape>

```

（2）点击注册按钮

点击注册按钮弹出对话框，输入账号、密码和电话号码三个数据，点击确定时数据将保存在SQLite数据库中，注册对话框如图5-2所示：



图5-2 注册对话框

注册对话框所对应布局文件（regist_dialog.xml）代码如下所示：

```

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

```

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <TextView
        android:id="@+id/tv_registHint"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="5dp"
        android:text="注册信息"
        android:textColor="#000000"
        android:textSize="20sp" />
    <View
        android:id="@+id/view_wire1"
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/tv_registHint"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginTop="5dp"
        android:background="#000000" />
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/view_wire1"
        android:layout_marginBottom="10dp">
        <TextView
            android:id="@+id/tv_userHint"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_marginLeft="10dp"
            android:layout_marginTop="20dp"
            android:text="用户名："
            android:textSize="18sp" />
        <EditText
            android:id="@+id/editTxt_username_regist"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBaseline="@+id/tv_userHint"
            android:layout_alignBottom="@+id/tv_userHint"
            android:layout_alignLeft="@+id/editTxt_phone_regist"
```

```
        android:background="@null"
        android:ems="10"
        android:hint="请输入用户名" />
<TextView
    android:id="@+id/tv_phoneHint"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/tv_userHint"
    android:layout_below="@+id/editTxt_username_regist"
    android:layout_marginTop="20dp"
    android:text="手机号码 : "
    android:textSize="18sp" />
<EditText
    android:id="@+id/editTxt_phone_regist"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/tv_phoneHint"
    android:layout_toRightOf="@+id/tv_phoneHint"
    android:background="@null"
    android:ems="10"
    android:hint="请输入手机号码"
    android:inputType="phone" />
<EditText
    android:id="@+id/editTxt_psw_regist"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/editTxt_phone_regist"
    android:layout_alignTop="@+id/tv_pswHint"
    android:background="@null"
    android:ems="10"
    android:hint="请输入密码" />
<TextView
    android:id="@+id/tv_pswHint"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/tv_phoneHint"
    android:layout_below="@+id/tv_phoneHint"
    android:layout_marginTop="22dp"
    android:text="密      码 : "
    android:textSize="18sp" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="10dp"
    android:layout_below="@+id/editTxt_psw_regist" />
```

```
</RelativeLayout>
</RelativeLayout>
```

(3) 创建数据库

在编写主页面逻辑之前要先创建数据库，SQLiteHelper.java 代码如下：

```
1 package cn.itcast.registuserinfo;
2 import android.content.Context;
3 import android.database.sqlite.SQLiteDatabase;
4 import android.database.sqlite.SQLiteOpenHelper;
5 public class SQLiteHelper extends SQLiteOpenHelper {
6     public SQLiteHelper(Context context) {
7         super(context, "user.db", null, 1);
8     }
9     @Override
10    public void onCreate(SQLiteDatabase db) {
11        db.execSQL("create table secret (id integer primary key
autoincrement,
12                name varchar(20),password varchar(20),phone
varchar(20))");
13    }
14    @Override
15    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
16    }
17 }
```

上述代码创建了一个 user.db 数据库，在数据库中创建一个名称为 secret 的表，该表中共有三个数据，name、password、phone。

(4) 创建数据库操作类

数据库创建完成后，创建数据库操作类 UserDao.java，该类中主要为数据库的增删改查操作。

```
1 package cn.itcast.registuserinfo;
2 import android.content.ContentValues;
3 import android.content.Context;
4 import android.database.Cursor;
5 import android.database.sqlite.SQLiteDatabase;
6 import java.util.ArrayList;
7 import java.util.List;
8 public class UserDao {
9     private SQLiteHelper helper;
10    public UserDao(Context context) {
11        helper = new SQLiteHelper(context);
12    }
13    public void insert(User psw) {
14        SQLiteDatabase db = helper.getWritableDatabase();
15        ContentValues values = new ContentValues();
```

```
16     values.put("name", psw.getName());
17     values.put("password", psw.getPassword());
18     values.put("phone", psw.getPhone());
19     db.insert("secret", null, values);
20     db.close();
21 }
22 public void delete(String name) {
23     SQLiteDatabase db = helper.getWritableDatabase();
24     db.delete("secret", "name=?", new String[]{name});
25     db.close();
26 }
27 public int update(User password) {
28     SQLiteDatabase db = helper.getWritableDatabase();
29     ContentValues values = new ContentValues();
30     values.put("name", password.getName());
31     values.put("password", password.getPassword());
32     values.put("phone", password.getPhone());
33     int count = db.update("secret", values, "name=?",
34         new String[]{password.getName()});
35     db.close();
36     return count;
37 }
38 public List<User> queryAll() {
39     SQLiteDatabase db = helper.getReadableDatabase();
40     Cursor c = db.query("secret", new String[]{"name", "password",
41         "phone"}, null, null, null, null, null);
42     List<User> list = new ArrayList<User>();
43     while (c.moveToNext()) {
44         String name = c.getString(c.getColumnIndex("name"));
45         String psw = c.getString(c.getColumnIndex("password"));
46         String phone = c.getString(c.getColumnIndex("phone"));
47         User password = new User(name, psw, phone);
48         list.add(password);
49     }
50     c.close();
51     db.close();
52     return list;
53 }
54 public String findData(String name, String data) {
55     String phone = "";
56     SQLiteDatabase db = helper.getReadableDatabase();
57     Cursor c = db.query("secret", null, "name=?", new String[]{name},
58         null, null, null);
59     while (c.moveToNext()) {
```

```
60         phone = c.getString(c.getColumnIndex(data));
61     }
62     c.close();
63     db.close();
64     return phone;
65 }
66 public void deleteAll() {
67     SQLiteDatabase db = helper.getWritableDatabase();
68     String sql = "DELETE FROM secret;";
69     db.execSQL(sql);
70     db.close();
71 }
72 }
```

(5) 创建业务 bean

根据数据结构创建业务 bean，其中包含 name、password、phone 三种属性。User.java 代码如下：

```
1 package cn.itcast.registuserinfo;
2 public class User {
3     private int id;
4     private String name;
5     private String password;
6     private String phone;
7     public User(String name, String password, String phone) {
8         super();
9         this.name = name;
10        this.password = password;
11        this.phone = phone;
12    }
13    public User(String name) {
14        super();
15        this.name = name;
16    }
17    public int getId() {
18        return id;
19    }
20    public void setId(int id) {
21        this.id = id;
22    }
23    public String getPhone() {
24        return phone;
25    }
26    public void setPhone(String phone) {
27        this.phone = phone;
28    }
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/966140213042010212>