

[Android] SQLite数据库之增删改查基础操作

原创 Eastmount 最后发布于2014-11-15 02:02:32 阅读数 4625 ☆ 收藏

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏,采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法...



Eastmount

¥9.90

去订阅

在编程中经常会遇到数据库的操作,而Android系统内置了SQLite,它是一款轻型数据库,遵守事务ACID的关系型数据库管理系统,它占用的资源非常低,能够支持Windows/Linux/Unix等主流操作系统,同时能够跟很多程序语言如C#、PHP、Java等相结合。下面先回顾SQL的基本语句,再讲述Android的基本操作。

一. adb shell回顾SQL语句

首先,我感觉自己整个大学印象最深的几门课就包括《数据库》,所以想先回顾SQL增删改查的基本语句。而在Android SDK中adb是自带的调试工具,它存放在sdk的platform-tools目录下,通过adb shell可以进入设备控制台,操作SQL语句。

```
G:
cd G:\software\Program software\Android\adt-bundle-windows-x86_64-20140321\sdk\platform-tools
adb shell
cd /data/data/com.example.sqliteaction/databases/
sqlite3 StuDatabase.db
.table
.schema
```

如下所示我先创建了SQLiteAction工程,同时在工程中创建了StuDatabase.db数据库。输入adb shell进入设备控制台,调用"sqlite3+数据库名"打开数据库,如果没有db文件则创建。

```
C:\windows\system32\cmd.exe - adb shell
Microsoft Windows [版本 6.2.9200]
(c) 2012 Microsoft Corporation。保留所有权利。

C:\Users\dell>G:
adb是android SDK自带调试工具

G:\>cd G:\software\Program software\Android\adt-bundle-windows-x86_64-20140321\sdk\platform-tools
进入设备控制台

G:\software\Program software\Android\adt-bundle-windows-x86_64-20140321\sdk\platform-tools>adb shell
root@mako:/ # cd /data/data/com.example.sqliteaction/databases/
进入databases目录
cd /data/data/com.example.sqliteaction/databases/
root@mako:/data/data/com.example.sqliteaction/databases # ls
ls
StuDatabase.db
StuDatabase.db-journal
root@mako:/data/data/com.example.sqliteaction/databases # sqlite3 StuDatabase.db
打开数据库
uDatabase.db
SQLite version 3.7.11 2012-03-20 11:35:50
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .table
.table
Student          android_metadata
sqlite>
```

然后如下图所示,可以输入SQL语句执行增删改查。注意很容易写错SQL语句,如忘记")"或结束";"导致cmd中调用出错。

```
-- 创建Teacher表
create table Teacher (id integer primary key, name text);
-- 向表中插入数据
insert into Teacher (id,name) values('10001', 'Mr Wang');
```

```

insert into Teacher (id,name) values('10002', 'Mr Yang'); -- 查询数据
select * from Teacher;
-- 更新数据
update Teacher set name='Yang XZ' where id=10002;
-- 删除数据
delete from Teacher where id=10001;

```

```

C:\windows\system32\cmd.exe - adb shell
sqlite> .table
.table
Student          android_metadata
sqlite> create table Teacher (id integer primary key, name text);
sqlite> .table
.table
Student          Teacher          android_metadata
sqlite> insert into Teacher (id,name) values('10001', 'Mr Wang');
insert into Teacher (id,name) values('10001', 'Mr Wang');
sqlite> insert into Teacher (id,name) values('10002', 'Mr Yang');
insert into Teacher (id,name) values('10002', 'Mr Yang');
sqlite> select * from Teacher;
select * from Teacher;
10001|Mr Wang
10002|Mr Yang
sqlite> update Teacher set name='Yang XZ' where id=10002;
update Teacher set name='Yang XZ' where id=10002;
sqlite> select * from Teacher;
select * from Teacher;
10001|Mr Wang
10002|Yang XZ
sqlite> delete from Teacher where id=10001;
delete from Teacher where id=10001;
sqlite> select * from Teacher;
select * from Teacher;
10002|Yang XZ
sqlite>

```

二. SQLite数据库操作

下面讲解使用SQLite操作数据库:

1. 创建打开数据库

使用openOrCreateDatabase函数实现,它会自动检测是否存在该数据库,如果存在则打开,否则创建一个数据库,并返回一个SQLiteDatabase对象.

2. 创建表

通过定义建表的SQL语句,再调用execSQL方法执行该SQL语句实现建立表.

```

// 创建学生表(学号, 姓名, 电话, 身高) 主键学号
public static final String createTableStu = "create table Student (" +
    "id integer primary key, " +
    "name text, " +
    "tel text, " +
    "height real)";
// SQLiteDatabase定义db变量
db.execSQL(createTableStu);

```

3. 插入数据

使用insert方法添加数据,其实ContentValues就是一个Map,Key字段名称,Value值.

SQLiteDatabase.insert(

String table, //添加数据的表名

String nullColumnHack, //为某些空的列自动复制NULL

ContentValues values //ContentValues的put()方法添加数据

);

//方法一

```
SQLiteDatabase db = sqlHelper.getWritableDatabase();
```

```
ContentValues values = new ContentValues();
```

```
values.put("id", "10001");
```

```
values.put("name", "Eastmount");
```

```
values.put("tel", "15201610000");
```

```
values.put("height", "172.5");
```

```
db.insert("Student", null, values);
```

//方法二

```
public static final String insertData = "insert into Student (" +
```

```
    "id, name, tel, height) values('10002','XiaoMing','110','175')";
```

```
db.execSQL(insertData);
```

4.删除数据

使用delete方法删除表中数据,其中sqlHelper是继承SQLiteDatabase自定义类的实例.

SQLiteDatabase.delete(

String table, //表名

String whereClause, //约束删除行,不指定默认删除所有行

String[] whereArgs //对应数据

);

//方法一 删除身高>175cm

```
SQLiteDatabase db = sqlHelper.getWritableDatabase();
```

```
db.delete("Student", "height > ?", new String[] {"175"});
```

//方法二

```
String deleteData = "DELETE FROM Student WHERE height>175";
```

```
db.execSQL(deleteData);
```

5.更新数据

使用update方法可以修改数据,SQL+execSQL方法就不在叙述.

//小明的身高修改为180

```
SQLiteDatabase db = sqlHelper.getWritableDatabase();
```

```
ContentValues values = new ContentValues();
```

```
values.put("height", "180");
```

```
db.update("Student", values, "name = ?", new String[] {"XiaoMing"});
```

6.其他操作

下面是关于数据库的其他操作,其中包括使用SQL语句执行,而查询数据Query方法由于涉及ListView显示,请见具体实例.

//关闭数据库

```
SQLiteDatabase.close();
```

//删除表 执行SQL语句

```
SQLiteDatabase.execSQL("DROP TABLE Student");
```

//删除数据库

```
this.deleteDatabase("StuDatabase.db");
```

//查询数据

```
SQLiteDatabase.query();
```

三. 数据库操作简单实例

显示效果如下图所示：



首先，添加activity_main.xml文件布局如下：

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.touchimagetest.MainActivity"
    tools:ignore="MergeRootFrame" >
    <!-- 顶部 -->
    <RelativeLayout
        android:id="@+id/MyLayout_top"
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="40dp"
        android:layout_alignParentTop="true" >
        <!-- 标题 -->
        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:gravity="center" >
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:gravity="center"
                android:textSize="20sp"
                android:text="学号" />
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:gravity="center"
                android:textSize="20sp"
                android:text="姓名" />
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:gravity="center"
                android:textSize="20sp"
                android:text="电话" />
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:gravity="center"
                android:textSize="20sp"
                android:text="身高" />
        </LinearLayout>
    </RelativeLayout>
    <TableLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:background="#f0f0f0" >
        <tr>
            <th>学号</th>
            <th>姓名</th>
            <th>电话</th>
            <th>身高</th>
        </tr>
        <tr>
            <td>10001</td>
            <td>Eastmount</td>
            <td>15201610000</td>
            <td>172.5</td>
        </tr>
        <tr>
            <td>10003</td>
            <td>XiaoMing</td>
            <td>15102530000</td>
            <td>180</td>
        </tr>
        <tr>
            <td>10006</td>
            <td>杨幂</td>
            <td>15108080000</td>
            <td>165.5</td>
        </tr>
    </TableLayout>
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:background="#f0f0f0" >
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="创表" />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="插入" />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="删除" />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="更新" />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="查询" />
    </LinearLayout>
</RelativeLayout>
```

```

        android:layout_weight="1"
        android:gravity="center"
        android:textSize="20sp"

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:gravity="center"
            android:textSize="20sp"
            android:text="身高" />
    </LinearLayout>
</RelativeLayout>
<!-- 底部按钮 -->
<RelativeLayout
    android:id="@+id/MyLayout_bottom"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="90dp"
    android:layout_alignParentBottom="true"
    android:gravity="center">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_alignParentBottom="true" >
        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="fill_parent"
            android:layout_height="40dp"
            android:gravity="center" >
            <EditText
                android:id="@+id/edit_id"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:gravity="center"
                android:textSize="20sp"
                android:hint="学号" />
            <EditText
                android:id="@+id/edit_name"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:gravity="center"
                android:textSize="20sp"
                android:hint="姓名" />
            <EditText
                android:id="@+id/edit_tel"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:gravity="center"
                android:textSize="20sp"
                android:hint="电话" />
            <EditText
                android:id="@+id/edit_height"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:gravity="center"

```

```

        android:textSize="20sp"
        android:hint="身高" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >
        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="创表" />
        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="插入" />
        <Button
            android:id="@+id/button3"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="删除" />
        <Button
            android:id="@+id/button4"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="更新" />
        <Button
            android:id="@+id/button5"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="查询" />
    </LinearLayout>
</LinearLayout>
</RelativeLayout>
<!-- 显示列表 -->
<RelativeLayout
    android:id="@+id/Content_Layout"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_above="@id/MyLayout_bottom"
    android:layout_below="@id/MyLayout_top"
    android:background="#EFD9DF" >
    <!-- 显示表内容 -->
    <ListView
        android:id="@+id/listview1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center" >
    </ListView>
</RelativeLayout>
</RelativeLayout>

```

然后是在res/layout中添加ListView显示的stu_item.xml:

```

<?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="horizontal" >
    <TextView
        android:id="@+id/stu_id"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textSize="20sp" />
    <TextView
        android:id="@+id/stu_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textSize="20sp" />
    <TextView
        android:id="@+id/stu_tel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textSize="20sp" />
    <TextView
        android:id="@+id/stu_height"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textSize="20sp" />
</LinearLayout>

```

再次,添加自定义类MySQLiteOpenHelper:

```

// 添加自定义类 继承SQLiteOpenHelper
public class MySQLiteOpenHelper extends SQLiteOpenHelper {

    public Context mContext;

    // 创建学生表(学号, 姓名, 电话, 身高) 主键学号
    public static final String createTableStu = "create table Student (" +
        "id integer primary key, " +
        "name text, " +
        "tel text, " +
        "height real)";

    // 抽象类 必须定义显示的构造函数 重写方法
    public MySQLiteOpenHelper(Context context, String name, CursorFactory factory,
        int version) {
        super(context, name, factory, version);
        mContext = context;
    }

    @Override
    public void onCreate(SQLiteDatabase arg0) {
        // TODO Auto-generated method stub
        arg0.execSQL(createTableStu);
        Toast.makeText(mContext, "Created", Toast.LENGTH_SHORT).show();
    }
}

```

```

@Override
    public void onUpgrade(SQLiteDatabase arg0, int arg1, int arg2) {
        // TODO Auto-generated method stub
        arg0.execSQL("drop table if exists Student");
        onCreate(arg0);
        Toast.makeText(mContext, "Upgraded", Toast.LENGTH_SHORT).show();
    }
}

```

最后是MainActivity.java文件,代码如下:

```

public class MainActivity extends Activity {

    // 继承SQLiteOpenHelper类
    private MySQLiteOpenHelper sqlHelper;
    private ListView listView;
    private EditText edit1;
    private EditText edit2;
    private EditText edit3;
    private EditText edit4;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        sqlHelper = new MySQLiteOpenHelper(this, "StuDatabase.db", null, 2);
        // 建立新表
        Button createBn = (Button) findViewById(R.id.button1);
        createBn.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                sqlHelper.getWritableDatabase();
            }
        });
        // 插入数据
        Button insertBn = (Button) findViewById(R.id.button2);
        edit1 = (EditText) findViewById(R.id.edit_id);
        edit2 = (EditText) findViewById(R.id.edit_name);
        edit3 = (EditText) findViewById(R.id.edit_tel);
        edit4 = (EditText) findViewById(R.id.edit_height);
        insertBn.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                SQLiteDatabase db = sqlHelper.getWritableDatabase();
                ContentValues values = new ContentValues();
                /*
                // 插入第一组数据
                values.put("id", "10001");
                values.put("name", "Eastmount");
                values.put("tel", "15201610000");
                values.put("height", "172.5");
                db.insert("Student", null, values);
                */
                values.put("id", edit1.getText().toString());
                values.put("name", edit2.getText().toString());
                values.put("tel", edit3.getText().toString());
                values.put("height", edit4.getText().toString());
                db.insert("Student", null, values);
                Toast.makeText(MainActivity.this, "数据插入成功", Toast.LENGTH_SHORT).show();
                edit1.setText("");
                edit2.setText("");
            }
        });
    }
}

```



```

        edit3.setText("");
        edit4.setText("");
    }
});
// 删除数据
Button deleteBn = (Button) findViewById(R.id.button3);
deleteBn.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        SQLiteDatabase db = sqlHelper.getWritableDatabase();
        db.delete("Student", "height > ?", new String[] {"180"});
        Toast.makeText(MainActivity.this, "删除数据", Toast.LENGTH_SHORT).show();
    }
});
// 更新数据
Button updateBn = (Button) findViewById(R.id.button4);
updateBn.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        SQLiteDatabase db = sqlHelper.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put("height", "180");
        db.update("Student", values, "name = ?", new String[] {"XiaoMing"});
        Toast.makeText(MainActivity.this, "更新数据", Toast.LENGTH_SHORT).show();
    }
});
// 查询数据
ListView listView = (ListView) findViewById(R.id.listview1);
Button selectBn = (Button) findViewById(R.id.button5);
selectBn.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            SQLiteDatabase db = sqlHelper.getWritableDatabase();
            // 游标查询每条数据
            Cursor cursor = db.query("Student", null, null, null, null, null, null);
            // 定义list存储数据
            List<Map<String, Object>> list = new ArrayList<Map<String, Object>>();
            // 适配器SimpleAdapter数据绑定
            // 错误: 构造函数SimpleAdapter未定义 需把this修改为MainActivity.this
            SimpleAdapter adapter = new SimpleAdapter(MainActivity.this, list,
                R.layout.stu_item,
                new String[]{"id", "name", "tel", "height"},
                new int[]{R.id.stu_id, R.id.stu_name, R.id.stu_tel,
                    R.id.stu_height});

            // 读取数据 游标移动到下一行
            while(cursor.moveToNext()) {
                Map<String, Object> map = new HashMap<String, Object>();
                map.put("id", cursor.getString(cursor.getColumnIndex("id")));
                map.put("name", cursor.getString(cursor.getColumnIndex("name")));
                map.put("tel", cursor.getString(cursor.getColumnIndex("tel")));
                map.put("height", cursor.getString(cursor.getColumnIndex("height")));

                list.add(map);
            }
            listView.setAdapter(adapter);
        } catch (Exception e) {
            Log.i("exception", e.toString());
        }
    }
});
}
}
}

```

PS: 希望文章对大家有所帮助,文章是关于SQLite的基础操作,而且没有涉及到数据库的触发器、存储过程、事务、索引等知识,网上也有很多相关的资料.同时现在有门课程《数据库高级技术与开发》,故作者当个在线笔记及基础讲解吧!这篇文章有一些不足之处,但作为基础文章还是不错的.

下载地址: <http://download.csdn.net/detail/eastmount/8159881>

主要参考:

1.郭霖大神的《第一行代码Android》

2.android中的数据库操作 By:nieweilin

(By:Eastmount 2014-11-15 夜2点 <http://blog.csdn.net/eastmount/> **)**

👍 点赞 5 ☆ 收藏 📄 分享 ...



Eastmount  博客专家

发布了445 篇原创文章 · 获赞 5981 · 访问量 487万+

他的留言板

关注