

实施方案 01 数据库管理 SQLiteOpenHelper

=====任务描述=====

(1) 数据库创建、连接。编写创建数据库事件，创建数据库 db_test，版本号为 1，要求在日志窗口输出数据库声明、创建、打开的状态信息。

(2) 数据库升级与版本管理。编写升级数据库事件，升级数据库 db_test 版本号为 2，要求在日志窗口输出数据库升级时的状态信息。



=====关键技术=====

1. SQLite 数据库可忽略数据类型，是一个轻量级数据库，支持 SQL 语句
 - ✓ 支持类型：null、integer(整数)、real(小数)、text(字符串)、blob(二进制)
 - ✓ SQLite 数据库文件通常存放在 data\data 目录下 (AVD 窗口查看)
 - ✓ 可下载数据库文件使用 SQLiteStudio 小工具查看及修改
2. SQLiteOpenHelper
 - ✓ 数据库辅助类
 - ✓ 用于创建数据库和管理数据库版本
 - ✓ 用户自定义其子类并在构造方法中完成创建表的操作
 - onCreate(): 必须重写，当数据库第一次创建时执行（一般编写创建表的语句及添加初始化数据）
 - onUpgrade(): 必须重写，当数据库更新时调用（数据库的版本号变化时被调用，软件升级时数据库结构的变化通过更新完成）。
 - onOpen(): 可选择重写，当数据库打开时调用。

=====项目实施=====

MyHelper.java

```
package com.jvtc.ad.ys0301;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
```

```

import android.util.Log;

/**
 * SQLiteOpenHelper——数据库辅助类
 * 用于管理数据库：可以定义创建、升级、打开、关闭数据库等操作时要执行的任务
 * Created by Administrator on 2017/3/17.
 */
public class MyHelper extends SQLiteOpenHelper {
    /**
     * 构造方法，创建对象时指定数据库的名字、版本
     * @param context: 上下文
     * @param name: 数据库名
     * @param factory: 自定义查询结果集的游标，一般为 null
     * @param version: 数据库的版本号
     */
    public MyHelper(Context context, String name, SQLiteDatabase.CursorFactory
factory, int version) {
        super(context, name, factory, version);
        Log.i("数据库状态", "数据库被声明：数据库名-" + name + ", 版本号-" + version);
    }
    /**
     * 数据库被创建时执行
     * 一般编写创建表的语句
     */
    @Override
    public void onCreate(SQLiteDatabase db) {
        Log.i("数据库状态", "创建数据库");
    }
    /**
     * 升级数据库时执行
     */
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        Log.i("数据库状态", "升级数据库");
    }
    /**
     * 连接数据库时执行
     */
    @Override
    public void onOpen(SQLiteDatabase db) {
        super.onOpen(db);
        Log.i("数据库状态", "连接数据库");
    }
}

```

MainActivity.java

```
package com.jvtc.ad.yso301;

import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    private Button btnCreate, btnUpdate;           // “声明数据库” 按钮
    private MyHelper dbHelper;                     // 数据库辅助类对象
    private SQLiteDatabase db;                     // 数据库对象
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnCreate=(Button)findViewById(R.id.btn_create);
        btnUpdate=(Button)findViewById(R.id.btn_update);
        btnCreate.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                dbHelper=new MyHelper(MainActivity.this,"db_test",null,1);
                //声明: 数据库名—db_test, 数据库版本—1
                Log.i("数据库状态","MyHelper 对象创建完成");
                db=dbHelper.getWritableDatabase();//获得数据库对象
                //程序安装后首次运行时创建并连接数据库,以后再运行时只连接数据库
            }
        });
        btnUpdate.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                dbHelper=new MyHelper(MainActivity.this,"db_test",null,2);
                //数据库版本号提高——升级数据库
                db=dbHelper.getWritableDatabase();
            }
        });
    }
}
```

关键技术解析 1: 数据库声明

The image shows a code editor with two files: `MyHelper.java` and `MainActivity.java`. The `MyHelper` class extends `SQLiteOpenHelper` and has a constructor with four parameters: `Context context`, `String name`, `SQLiteDatabase.CursorFactory factory`, and `int version`. The `MainActivity` class uses `MyHelper` in its `onCreate` method. Annotations explain the parameters and the log output.

MyHelper.java

```
1 package com.jvtc.ad.y0301;
2
3 import ...
4
5 /**
6  * SQLiteOpenHelper--数据库辅助类
7  * 用于管理数据库: 可以定义创建、升级、打开、关闭数据库等操作时要执行的任务
8  * Created by Administrator on 2017/3/17.
9  */
10
11 public class MyHelper extends SQLiteOpenHelper {
12     /**
13      * 构造方法, 创建对象时指定数据库的名字、版本
14      * @param context: 上下文
15      * @param name: 数据库名
16      * @param factory: 自定义查询结果集的游标, 一般为null
17      * @param version: 数据库的版本号
18      */
19     public MyHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {
20         super(context, name, factory, version);
21         Log.i("数据库状态", "数据库被声明");
22         Log.i("数据库状态", "数据库名: " + name + ", 版本号: " + version);
23     }
24
25     @Override
26     public void onCreate(SQLiteDatabase db) {...}
27
28     @Override
29     public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {...}
30 }
```

MainActivity.java

```
1 package com.jvtc.ad.y0301;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6     private Button btnDb; // "声明数据库" 按钮
7     private MyHelper dbHelper; // 数据库辅助类对象
8
9     @Override
10     protected void onCreate(Bundle savedInstanceState) {
11         super.onCreate(savedInstanceState);
12         setContentView(R.layout.activity_main);
13         btnDb = (Button) findViewById(R.id.btn_db);
14         btnDb.setOnClickListener((v) -> {
15             dbHelper = new MyHelper(MainActivity.this, "db_test", null, 1);
16             // 声明: 数据库名=db_test, 数据库版本=1
17         });
18     }
19 }
```

Android Monitor Log

```
01-01 00:40:40.340 14075-14075/com.jvtc.ad.y0301 I/数据库状态: 数据库被声明
01-01 00:40:40.350 14075-14075/com.jvtc.ad.y0301 I/数据库状态: 数据库名: db_test, 版本号: 1
```

Annotations:

- 参数1: 上下文对象 (points to `Context context`)
- 参数2: 数据库名 (points to `String name`)
- 参数4: 数据库版本号 (用于升级数据库) (points to `int version`)
- 日志窗口输出验证代码功用 (points to the log output)

关键技术解析 2: 数据库创建、升级与连接

The image shows a code editor with two files: `MyHelper.java` and `MainActivity.java`. The `MyHelper` class extends `SQLiteOpenHelper` and implements `onCreate`, `onUpgrade`, and `onOpen` methods. The `MainActivity` class extends `AppCompatActivity` and uses `MyHelper` to manage a database.

Annotations:

- ① 无数据库时 执行创建数据库onCreate()方法**: Points to the `onCreate` method in `MyHelper` and the `btnCreate` button in `MainActivity`.
- 版本号提高, 执行升级数据库方法onUpgrade()**: Points to the `onUpgrade` method in `MyHelper` and the `btnUpdate` button in `MainActivity`.
- ② 执行打开数据库方法onOpen()**: Points to the `onOpen` method in `MyHelper`.

Code Snippets:

MyHelper.java

```
package com.jvtc.ad.y0301;

import ...

/**...*/
public class MyHelper extends SQLiteOpenHelper {
    /**...*/
    public MyHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
        Log.i("数据库状态", "数据库被声明: 数据库名-" + name + ", 版本号-" + version);
    }
    /**...*/
    @Override
    public void onCreate(SQLiteDatabase db) {
        Log.i("数据库状态", "创建数据库");
    }
    /**...*/
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        Log.i("数据库状态", "升级数据库");
    }
    /**...*/
    @Override
    public void onOpen(SQLiteDatabase db) {
        super.onOpen(db);
        Log.i("数据库状态", "连接数据库");
    }
}
```

MainActivity.java

```
package com.jvtc.ad.y0301;

import ...

public class MainActivity extends AppCompatActivity {
    private Button btnCreate, btnUpdate; // "声明数据库" 按钮
    private MyHelper dbHelper; // 数据库辅助类对象
    private SQLiteDatabase db; // 数据库对象

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnCreate = (Button) findViewById(R.id.btn_create);
        btnUpdate = (Button) findViewById(R.id.btn_update);
        btnCreate.setOnClickListener((v) -> {
            dbHelper = new MyHelper(MainActivity.this, "db_test", null, 1);
            // 声明: 数据库名 - db_test, 数据库版本 - 1
            Log.i("数据库状态", "MyHelper对象创建完成");
            db = dbHelper.getWritableDatabase(); // 获得数据库对象
            // 程序安装后首次运行时创建并连接数据库, 以后再运行时只连接数据库
        });
        btnUpdate.setOnClickListener((v) -> {
            dbHelper = new MyHelper(MainActivity.this, "db_test", null, 2);
            // 数据库版本号提高 - 升级数据库
            db = dbHelper.getWritableDatabase();
        });
    }
}
```

关键技术解析 3: 数据库状态解析

