



《移动终端开发技术》 电子教案

第三单元 SQLite 数据库存储

所属专业（教研室）： 计算机软件技术

制定人： 陈媛媛

合作人：

制定时间： 2018年2月

日照职业技术学院



单元标题	SQLite 数据库存储	单元教学学时	6 课时
		在整体设计中的位置	第 15 次
授课班级		上课地点	一体化教室
上课时间	周 月 日第 节		
教学目标	能力目标	知识目标	素质目标
	能够熟练使用 SQLite 创建数据库、对数据进行增删改查操作。	1、掌握创建数据库的方法 2、掌握 adb 工具的使用 3、掌握升级数据库的方法 4、掌握数据的增删改查操作	1、养成积极主动学习意识； 2、养成勤于动手的习惯。
教学重点、难点	教学重点：创建数据库的方法、数据的增删改查操作 教学难点：数据的增删改查操作		
教学方法	采用反转课堂教学模式，课前学生学习微课了解知识点，课上采用教师引导、演示，学生分组练习、讨论等教学方法。 运用多媒体、AndroidStudio 开发环境、实训助手、教学平台等辅助授课。		
课前需掌握的知识点	MySQL 数据库的使用：创建数据库；创建表；对数据增、删、改、查操作。 insert () 、 select () 、 update () 、 delete () 方法的使用。		
教学任务分解	任务一、创建数据库 任务二、adb 工具的使用 任务三、升级数据库 任务四、数据的增删改查操作		
教学总结			



一、情景导入

SharedPreferences 和文件存储适合存储少量数据，存储大量数据时使用 SQLite 数据库。

1、SQLite 数据库特点

SQLite 是一个轻量级数据库，第一个版本诞生于 2000 年 5 月。它最初是为嵌入式设计的，占用资源非常低，在内存中只需要占用几百 KB 的存储空间。这也是 Android 移动设备采用 SQLite 数据库的重要原因之一。

SQLite 是遵守 ACID 关联式的数据库管理系统。这里的 ACID 是指数据库事务正确执行的四个基本要素，即原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability）。同时 SQLite 还支持 SQL 语言、事务处理等功能。

SQLite 没有服务器进程，它通过文件保存数据，该文件是跨平台的，可以放在其他平台中使用。并且在保存数据时，支持 NULL、INTEGER、REAL（浮点数字）、TEXT(字符串文本)和 BLOB(二进制对象)五种数据类型。

2、SQLite 操作 API

老师引导，为了方便使用 SQLite 数据库，Android SDK 提供了一系列对数据库进行操作的类和接口，针对这些类和接口的作用以及 API 进行讲解。

1) SQLiteOpenHelper 类，它是一个抽象类，该类用于创建数据库和数据库版本更新；

2) SQLiteDatabase 类，它是一个数据库访问类，该类封装了一系列数据库操作的 API，可以对数据进行增删改查操作；

3) Cursor 接口，Cursor 是一个游标接口，在数据库操作中作为返回值，相当于结果集 ResultSet；

二、课前测试

在网络课程平台上做以下练习：

1、使用 SQLite 数据库进行查询后，必须要做的操作是（ ）。

- A 关闭数据库
- B 直接退出
- C 关闭 Cursor
- D 使用 quit 函数退出

答案 AC

2、使用 SQLiteOpenHelper 类可以生成一个数据库并可以对数据库版本进行管理的方法是()。

- A getDatabase()
- B getWritableDatabase()
- C getReadableDatabase()
- D getAbleDatabase()



答案 B

3、SQLiteDatabase 提供了 () 方法用于添加数据。

A insert ()

B update ()

C delete ()

D create ()

答案 A

4、创建一个数据库帮助类 TestDBHelper，需要继承父类 SQLiteOpenHelper 后才具有管理数据库版本的功能。 () 答案 对

5、可以使用 adb shell 来对数据库和表的创建情况进行查看。 ()

答案 对

三、知识讲解

任务一、创建数据库

(1) 创建 MyDatabaseHelper.java 继承 SQLiteOpenHelper

```
public class MyDatabaseHelper extends SQLiteOpenHelper {
    public static final String create_book="create table Book (" + "id integer primary
    key autoincrement,"
        + "author varchar(20),"
        + "price double,"
        + "name varchar(20))";

    /*public static final String create_book="create table Author (" + "id integer
    primary key autoincrement,"
        + "name text,"
        + "age integer,"
        + "price real,"
        + "sex text)";*/

    private Context mContext;

    public MyDatabaseHelper(Context context, String name, SQLiteDatabase.CursorFactory
    factory, int version) {
        super(context, name, factory, version);
        mContext=context;
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        sqLiteDatabase.execSQL(create_book);
        Toast.makeText(mContext,"Create succeeded",Toast.LENGTH_LONG).show();
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
```



```
    }  
}  
  
(2) Activity_main.xml  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical">  
  
    <Button  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:text="create database"  
        android:id="@+id/create"/>  
  
</LinearLayout>  
  
(3) MainActivity.java  
public class MainActivity extends AppCompatActivity {  
    private MyDatabaseHelper dbHelper;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        dbHelper=new MyDatabaseHelper(this,"BookStore.db",null,1);  
        Button createDatabase=(Button)findViewById(R.id.create);  
        createDatabase.setOnClickListener(new Button.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                dbHelper.getWritableDatabase();//创建数据库  
            }  
        });  
    }  
}
```

任务二、查看数据库

SQLite 提供了一个访问数据库的工具 `sqlite3.exe`，通过它可以直接操作数据库。

由于 Android 是在运行时 (run-time) 集成了 `sqlite3.exe`，因此在使用 `sqlite3.exe` 工具之前必须要开启模拟器或者连接真实设备。在使用该工具时，首先需要打开 DOS 命令行，依次输入如下命令：

配置环境变量：将 `platform-tools` 目录配置到环境变量中，
D:\Android\Android\jdk\platform-tools

运行 cmd

(1) adb shell (使用 adb 工具)

#表示超级管理员 \$表示普通管理员，使用 su 切换



(2) # cd /data/data/com.example.cyy.databasetest/databases

(3) Ls (查看数据库)

出现 BookStore.db BookStore.db-journal

sqlite3 BookStore.db(使用 sqlite3 操作应用程序下的数据库)

(4) .table (查看表)

出现 Book android_metadata

(5) .schema (查看表结构)

任务三、升级数据库

(1) MyDatabaseHelper 类中的 onUpgrade () 方法

```
public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int il) {  
    sqLiteDatabase.execSQL("drop table if exists Book");  
    sqLiteDatabase.execSQL("drop table if exists Category");  
    onCreate(sqLiteDatabase);  
}
```

(2) MainActivity.java

```
dbHelper=new MyDatabaseHelper(this, "Book1.db", null, 2);  
//创建数据库, 第四个参数大于 1 就能让 onUpgrade () 方法执行
```

任务四、数据的增删改查操作

(1) 插入数据

第一步: getWritableDatabase () 方法返回 SQLiteDatabase 对象, 用于对数据进行增删改查

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

第二步: 创建 ContentValues 对象

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

第三步: 开始组装一条数据

```
values.put("字段名", 值);
```

第四步: 插入记录

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

(2) 更新数据

第一步: getWritableDatabase () 方法返回 SQLiteDatabase 对象, 用于对数据进行增删改查

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

第二步: 创建 ContentValues 对象

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

第三步: 开始组装一条数据

```
values.put("字段名", 值);
```



第四步：调用 update () 方法进行更新

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

有四个参数，第一个参数为表名，第二个参数为 ContentValues 对象，第三个、四个参数用于约束更新某一行或某几行中的数据，不指定就更新所有行。

(3) 删除数据

第一步：getWritableDatabase () 方法返回 SQLiteDatabase 对象，用于对数据进行增删改查

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

第二步：调用 delete () 方法删除数据

```
db.delete("Book", "price>?", new String[]{"100"});
```

delete () 有三个参数，第一个参数是表名，第二个、第三个参数约束删除某一行或某几行的数据，不指定默认删除所有行。

(4) 查询数据

第一步：getWritableDatabase () 方法返回 SQLiteDatabase 对象，用于对数据进行增删改查

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

第二步：调用 query () 方法查询数据，将结果赋给 Cursor 对象。

```
Cursor
```

```
cursor=db.query("Book", null, null, null, null, null, null);
```

第三步：遍历 cursor，读取每个字段。

常用方法：cursor.moveToFirst () 移到第一条记录；getColumnIndex ("字段名") 得到某一列在表中的位置索引；cursor.getString ()；cursor.getInt ()；cursor.getDouble ()；获取数据。

第四步：关闭 cursor

```
cursor.close();
```

四、项目源码

Activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="create database"
        android:id="@+id/create"/>
    <Button
```



```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="add data"
        android:id="@+id/add"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="update data"
    android:id="@+id/update"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="delete data"
    android:id="@+id/delete"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="query data"
    android:id="@+id/query"/>
```

</LinearLayout>

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private MyDatabaseHelper dbHelper;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        dbHelper=new MyDatabaseHelper(this,"Book1.db",null,2);
        //创建数据库, 第四个参数大于1就能让onUpgrade()方法执行
        final Button createDatabase=(Button)findViewById(R.id.create);
        Button addData=(Button)findViewById(R.id.add);
        Button updateData=(Button)findViewById(R.id.update);
        Button deleteData=(Button)findViewById(R.id.delete);
        Button queryData=(Button)findViewById(R.id.query);
        //创建数据库
        createDatabase.setOnClickListener(new Button.OnClickListener() {
            @Override
            public void onClick(View view) {
                dbHelper.getWritableDatabase();//创建数据库
            }
        });
        //插入数据
        addData.setOnClickListener(new Button.OnClickListener() {
            @Override
            public void onClick(View view) {
```




```
        SQLiteDatabase
db=dbHelper.getWritableDatabase();//getWritableDatabase()方法返回 SQLiteDatabase 对
象，用于对数据进行增删改查
        ContentValues values=new ContentValues();//创建 ContentValues 对象
        //开始组装第一条数据
        values.put("name","The First Code");
        values.put("author","GuoLin");
        values.put("pages",570);
        values.put("price",79);
        db.insert("Book",null,values);//插入一条记录
        values.clear();//清空 values 中的数据
        //开始组装第二条数据
        values.put("name","Android");
        values.put("author","LiGang");
        values.put("pages",765);
        values.put("price",108);
        db.insert("Book",null,values);//插入一条记录
    });
    //更新数据
    updateData.setOnClickListener(new Button.OnClickListener() {
        @Override
        public void onClick(View view) {
            SQLiteDatabase
db=dbHelper.getWritableDatabase();//getWritableDatabase()方法返回 SQLiteDatabase 对
象，用于对数据进行增删改查
            ContentValues values=new ContentValues();
            values.put("price",89);
            db.update("Book",values,"name=?",new String[]{"Android"});
            //update()有四个参数，第一个参数是表名，第二个参数是要更新的数据，第
三个、第四个指出具体更新哪几行
        } });
    //删除数据
    deleteData.setOnClickListener(new Button.OnClickListener() {
        @Override
        public void onClick(View view) {
            SQLiteDatabase
db=dbHelper.getWritableDatabase();//getWritableDatabase()方法返回 SQLiteDatabase 对
象，用于对数据进行增删改查
            db.delete("Book","price>?",new String[]{"100"});
            //delete()有三个参数，第一个参数是表名，第二个、第三个参数约束删除某
一行或某几行的数据，不指定默认删除所有行
        } });
    //查询数据
    queryData.setOnClickListener(new Button.OnClickListener() {
```

```
@Override
public void onClick(View view) {
    SQLiteDatabase
db=dbHelper.getWritableDatabase();//getWritableDatabase()方法返回 SQLiteDatabase 对
象，用于对数据进行增删改查
    //查询 Book 表中所有记录
    Cursor cursor=db.query("Book",null,null,null,null,null,null);
    if(cursor.moveToFirst()){
        do{//遍历 Cursor 对象，取出数据并打印
            String
name=cursor.getString(cursor.getColumnIndex("name")); //cursor.getColumnIndex()得到
某一列在表中的位置索引
            String author=cursor.getString(cursor.getColumnIndex("author"));
            int pages=cursor.getInt(cursor.getColumnIndex("pages"));
            double price=cursor.getDouble(cursor.getColumnIndex("price"));
            Log.d("MainActivity","book name is "+name);
            Log.d("mainActivity","book author is "+author);
            Log.d("mainActivity","book pages is "+pages);
            Log.d("mainActivity","book price is "+price);
        }while (cursor.moveToNext());
    }
    cursor.close();//关闭 cursor
}
});}}
```



图 运行结果



	<p>五、演示作品</p> <p>各小组演示汇报作品，教师与其他小组总结评价作品，各小组课后完善作品并提交到教学平台。</p> <p>六、知识巩固</p> <ol style="list-style-type: none">1、总结知识点，使用教学平台中的随堂练习题巩固本所学知识。2、使用教学平台中的测试题给学生布置作业。
<p>拓展 作业</p>	<p>使用 SQLite 创建数据库和表，并对数据进行增删改查操作。</p>
<p>教学 后记</p>	