# 实验3 数据存储与SQLite数据库

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 202231060901 | 姓名 | 刘洋 |
| 专业 | 物联网工程 | 年级 | 物联网2022级 |

特别说明：实验报告必须命名为【学号紧跟姓名\_实验项目名】的格式，不包含括号【】。

例如【202031061234张三\_Android实验3.docx】，不按此命名不得分。

规范命名后，压缩为rar或zip来提交。

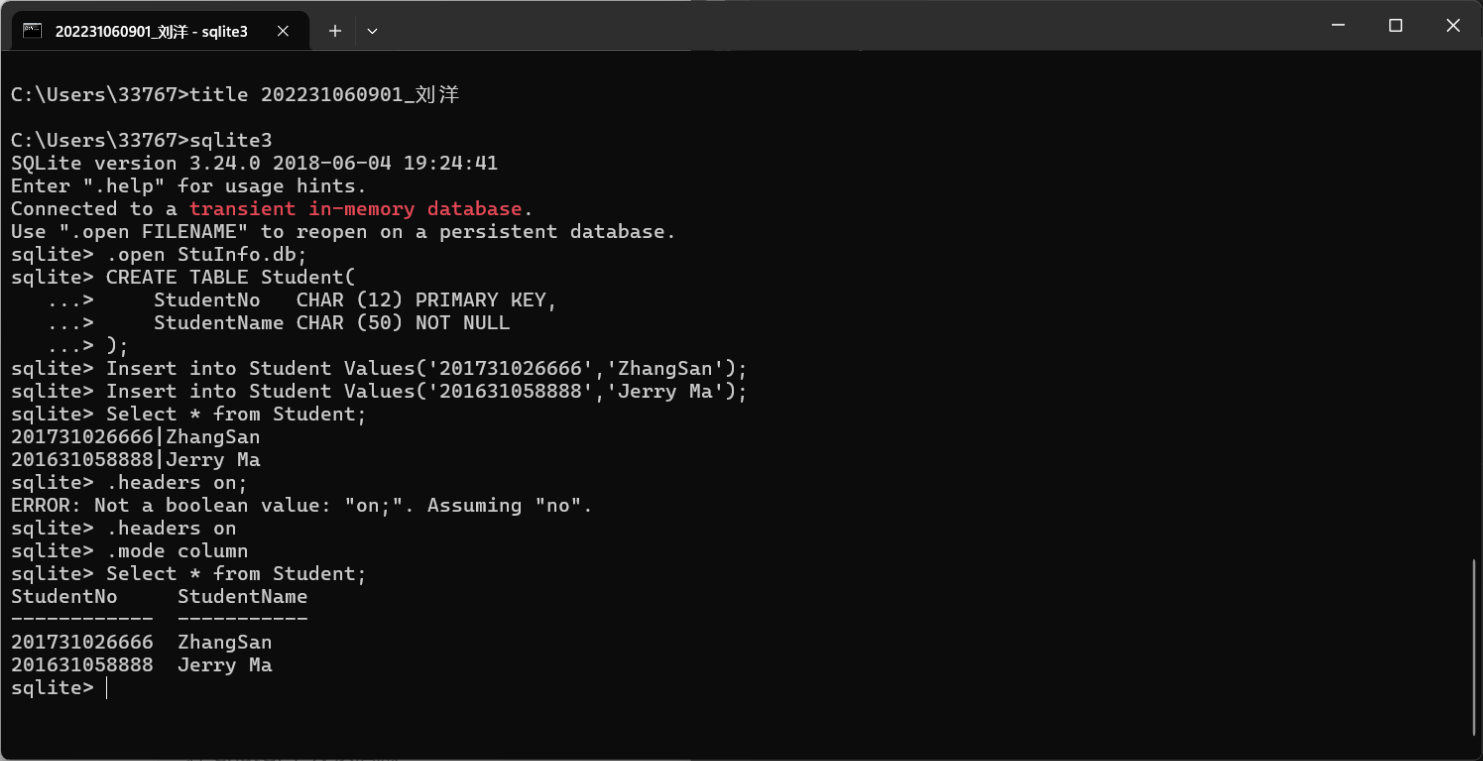
实验报告提交地址：<https://pintia.cn/>

## Sqlite3命令行使用(10分)

1. 你键入的SQL命令：

|  |
| --- |
| SqLite3  .open StuInfo.db  CREATE TABLE Student(  StudentNo CHAR (12) PRIMARY KEY,  StudentName CHAR (50) NOT NULL  );  Insert into Student Values('201731026666','ZhangSan');  Insert into Student Values('201631058888','Jerry Ma');  Select \* from Student;  .headers on  .mode column  Select \* from Student;  .quit |

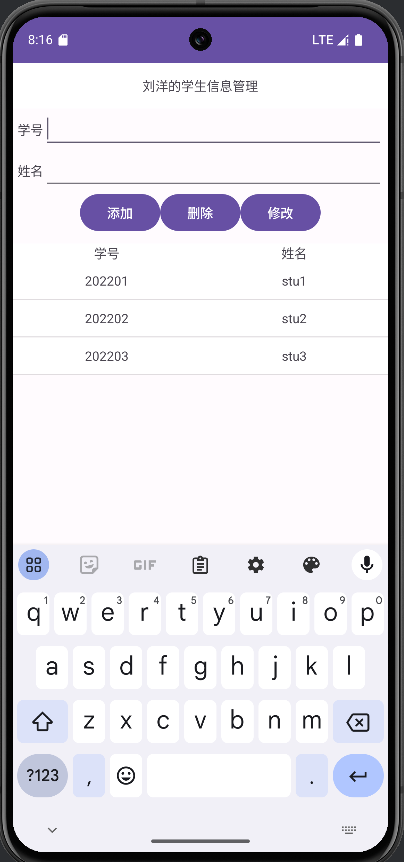
1. 运行截图(命令行的标题栏必须有**你的姓名**)



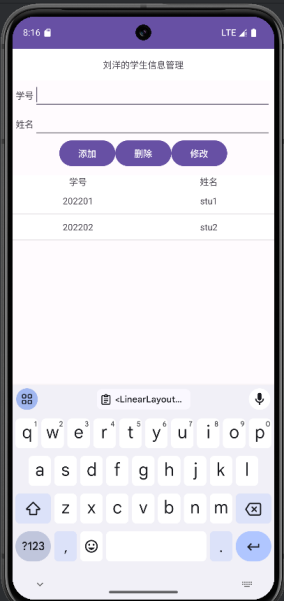
## 根据教师提示实现一个数据存储例子(30分)

提供运行截图之前，必须在界面上出现自**己的姓名**。

1. 成功添加了3条记录的运行截图，10分



1. 删除了一条记录后的运行截图，10分



1. 分析代码的执行，用一百字左右描述点击三个按钮时代码的执行流程，10分。

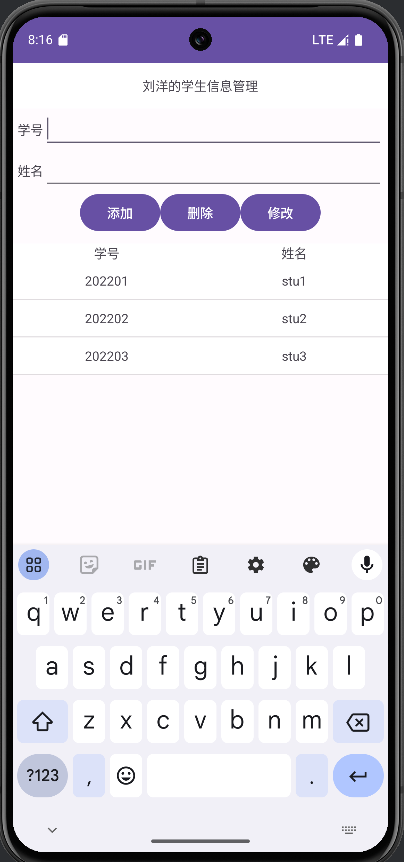
点击 新增按钮 时，代码将检查学号和姓名输入框是否为空，若不为空，将数据插入数据库表中并刷新列表。

点击 删除按钮 时，根据学号构造删除条件，从数据库中删除对应记录，然后刷新列表。

点击 修改按钮 时，代码根据学号更新对应学生的姓名字段，并刷新列表。

## Android APP在真机上的运行(20分)

1. 在模拟器上的运行效果截图，必须**出现你的姓名，10分**。



1. 在真实的Android手机上运行的截图。如果没有安卓手机，请说明。5分。



1. 获取IEMI失败的原因和解决方案，5分。

暂未遇到IEMI失败的问题

## 自行设计一个数据存储的例子(30分)

1. 创建表的SQL代码，5分。

|  |
| --- |
| CREATE TABLE IF NOT EXISTS TodoList (  id INTEGER PRIMARY KEY AUTOINCREMENT,  time TEXT NOT NULL,  content TEXT NOT NULL  ); |

1. 运行截图，组图，10分

添加事项 删除事项 修改事项



1. 布局文件代码，如果有多个，都分别给出，5分

|  |
| --- |
| <?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="40dp"  android:background="@android:color/white">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:orientation="horizontal">  <TextView  android:id="@+id/tv\_Time"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:layout\_weight="1"  android:gravity="center"  android:text="@string/Time" />  <TextView  android:id="@+id/tv\_Backlog"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="@string/Backlog"  android:gravity="center"  android:layout\_gravity="center"/>  </LinearLayout>  </LinearLayout>  Activity\_main.xml  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:fitsSystemWindows="true"  android:orientation="vertical">  <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:background="@android:color/white"  >  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="@string/welcome\_text"  android:layout\_centerInParent="true" />  </RelativeLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  android:layout\_marginLeft="5dp"  android:layout\_marginRight="5dp">  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="@string/Time" />  <EditText  android:id="@+id/et\_Time"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:focusableInTouchMode="true"/>  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  android:layout\_marginLeft="5dp"  android:layout\_marginRight="5dp">  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="@string/Backlog" />  <EditText  android:id="@+id/et\_Backlog"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content" />  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  android:gravity="center">  <Button  android:id="@+id/bt\_insert"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="@string/insert"/>  <Button  android:id="@+id/bt\_delete"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="@string/delete"/>  <Button  android:id="@+id/bt\_update"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="@string/update"/>  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  android:background="@android:color/white"  android:layout\_marginTop="10dp">  <TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="@string/Time"  android:gravity="center"/>  <TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="@string/Backlog"  android:gravity="center"/>  </LinearLayout>  <ListView  android:id="@+id/lv\_todoList"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:listSelector="@android:color/transparent"  android:drawSelectorOnTop="true">  </ListView>  </LinearLayout> |

1. Java文件代码，如果有多个，都分别给出，10分

|  |
| --- |
| DatabaseHelper.java  package edu.swpu.iot2022;  import android.content.Context;  import android.database.sqlite.SQLiteDatabase;  import android.database.sqlite.SQLiteOpenHelper;  public class DatabaseHelper extends SQLiteOpenHelper {  private static final String DB\_NAME = "TodoList.db"; // 数据库名称  private static final int version = 1; // 数据库版本  public static final String TABLE\_NAME = "TodoList";  public static final String FIELD1\_ID = "id"; // 第1个字段，id  public static final String FIELD2\_TIME = "time"; // 第2个字段，时间  public static final String FIELD3\_BACKLOG = "backlog"; // 第3个字段，代办事项内容  public DatabaseHelper(Context context) {  super(context, DB\_NAME, null, version);  }  @Override  public void onCreate(SQLiteDatabase db) {  // 创建表的 SQL 语句  String sql = "CREATE TABLE " + TABLE\_NAME + " (" +  FIELD1\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +  FIELD2\_TIME + " TEXT NOT NULL, " +  FIELD3\_BACKLOG + " TEXT NOT NULL);";  db.execSQL(sql);  }  @Override  public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  // 当数据库版本升级时的处理逻辑  db.execSQL("DROP TABLE IF EXISTS " + TABLE\_NAME);  onCreate(db);  }  }  BacklogAdapter.java  package edu.swpu.iot2022;  import android.content.Context;  import android.view.LayoutInflater;  import android.view.View;  import android.view.ViewGroup;  import android.widget.BaseAdapter;  import android.widget.TextView;  import java.util.ArrayList;  public class BacklogAdapter extends BaseAdapter {  private Context context;  private ArrayList<Backlog> list;  static class ViewHolder{  TextView tvTime;  TextView tvBacklog;  }  public BacklogAdapter(Context context, ArrayList<Backlog> list){  this.context=context;  this.list=list;  }  @Override  public int getCount() {  return list.size();  }  @Override  public Object getItem(int position) {  return list.get(position);  }  @Override  public long getItemId(int position) {  return position;  }  @Override  public View getView(int position, View convertView, ViewGroup parent) {  ViewHolder viewHolder;  // 检查是否存在可复用的视图  if (convertView == null) {  // 如果 convertView 为空，说明这是一个新项，需要创建视图  convertView = LayoutInflater.from(context).inflate(R.layout.layout\_item\_backlog, null);  // 创建一个新的 ViewHolder 对象，用于缓存视图中控件的引用  viewHolder = new ViewHolder();  // 查找布局中的控件并存储到 ViewHolder 中  viewHolder.tvTime = (TextView) convertView.findViewById(R.id.tv\_Time);  viewHolder.tvBacklog = (TextView) convertView.findViewById(R.id.tv\_Backlog);  // 将 ViewHolder 与 convertView 绑定，方便后续复用  convertView.setTag(viewHolder);  } else {  // 如果 convertView 不为空，则直接获取其绑定的 ViewHolder  viewHolder = (ViewHolder) convertView.getTag();  }  // 设置当前项数据到控件中  viewHolder.tvTime.setText(list.get(position).Time); // 设置时间  viewHolder.tvBacklog.setText(list.get(position).Backlog); // 设置待办事项  // 返回填充好数据的视图  return convertView;  }  }  MainActivity.java  package edu.swpu.iot2022;  import android.database.Cursor;  import android.database.sqlite.SQLiteDatabase;  import android.os.Bundle;  import android.view.View;  import android.view.WindowManager;  import android.widget.AdapterView;  import android.widget.EditText;  import android.widget.ListView;  import android.widget.TextView;  import android.widget.Toast;  import androidx.appcompat.app.AppCompatActivity;  import java.util.ArrayList;  public class MainActivity extends AppCompatActivity {  private SQLiteDatabase db;  private EditText et\_Time, et\_Backlog;  private ArrayList<Backlog> alBacklog;  private ListView lvTodoList;  private BacklogAdapter adapter;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  hideSoftKeyboard();  initView();  initData();  }  private void initView() {  DatabaseHelper databaseHelper = new DatabaseHelper(MainActivity.this);  db = databaseHelper.getWritableDatabase();  et\_Time = findViewById(R.id.et\_Time);  et\_Backlog = findViewById(R.id.et\_Backlog);  lvTodoList = findViewById(R.id.lv\_todoList);  alBacklog = new ArrayList<>();  adapter = new BacklogAdapter(this, alBacklog);  lvTodoList.setAdapter(adapter);  lvTodoList.setOnItemClickListener(new AdapterView.OnItemClickListener() {  @Override  public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  et\_Time.setText(alBacklog.get(position).Time);  et\_Backlog.setText(alBacklog.get(position).Backlog);  }  });  // 新增  findViewById(R.id.bt\_insert).setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  if (et\_Time.getText().toString().isEmpty() || et\_Backlog.getText().toString().isEmpty()) {  Toast.makeText(MainActivity.this, getString(R.string.can\_not\_be\_empty), Toast.LENGTH\_SHORT).show();  } else {  String sql = "INSERT INTO " + DatabaseHelper.TABLE\_NAME +  " (" + DatabaseHelper.FIELD2\_TIME + ", " + DatabaseHelper.FIELD3\_BACKLOG + ") " +  "VALUES ('" + et\_Time.getText().toString().trim() + "', '" + et\_Backlog.getText().toString().trim() + "')";  db.execSQL(sql);  initData();  }  }  });  // 删除  findViewById(R.id.bt\_delete).setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  String whereClause = DatabaseHelper.FIELD2\_TIME + "=?";  String[] whereArgs = {et\_Time.getText().toString().trim()};  db.delete(DatabaseHelper.TABLE\_NAME, whereClause, whereArgs);  initData();  }  });  // 修改  findViewById(R.id.bt\_update).setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  String sql = "UPDATE " + DatabaseHelper.TABLE\_NAME +  " SET " + DatabaseHelper.FIELD3\_BACKLOG + "='" + et\_Backlog.getText().toString().trim() + "' " +  "WHERE " + DatabaseHelper.FIELD2\_TIME + "='" + et\_Time.getText().toString().trim() + "'";  db.execSQL(sql);  initData();  }  });  }  // 查询数据  private void initData() {  alBacklog.clear();  Cursor cursor = db.query(DatabaseHelper.TABLE\_NAME, null, null, null, null, null, null);  while (cursor.moveToNext()) {  Backlog backlog = new Backlog();  backlog.id = cursor.getString(cursor.getColumnIndex(DatabaseHelper.FIELD1\_ID));  backlog.Time = cursor.getString(cursor.getColumnIndex(DatabaseHelper.FIELD2\_TIME));  backlog.Backlog = cursor.getString(cursor.getColumnIndex(DatabaseHelper.FIELD3\_BACKLOG));  alBacklog.add(backlog);  }  cursor.close();  adapter.notifyDataSetChanged();  et\_Time.setText("");  et\_Backlog.setText("");  }  // 隐藏软键盘  public void hideSoftKeyboard() {  getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT\_INPUT\_STATE\_ALWAYS\_HIDDEN);  }  } |

## 实验心得体会(10分)

要求不少于100汉字，必须是自己原创，不得雷同。

|  |
| --- |
| 在本次实验中，我深入学习了SQLite数据库的基本操作，包括数据的增删改查功能的实现，并将其与Android应用开发相结合。通过实验，我不仅掌握了如何设计数据库表结构，还了解了如何使用Cursor读取数据并动态更新用户界面。 |