LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG

CHƯƠNG 6:

SQLite

- Giảng viên: Ths. Nguyễn Trung Hiếu
- Email:hieunt.tg@ptithcm.edu.vn
- Mobie: 0983051825

Mục tiêu

- Kết thúc bài học này bạn có khả năng
  - Hiểu rõ về SQLite và các kiểu dữ liệu
  - Sử dụng SQLiteOpenHelper

NỘI DUNG

- 📖 Phần I: SQLite
  - 📖 Giới thiệu về SQLite
  - 📖 Các kiểu dữ liệu trong SQLite
- 📖 Phần II: Sử dụng SQLiteOpenHelper
  - 📖 SQLiteOpenHelper
  - 📖 Tạo Database và Table

1

SQLite

Giới thiệu SQLite

- SQLite là phần mềm quản lý cơ sở dữ liệu SQL nhưng không giống như hầu hết các cơ sở dữ liệu SQL khác, SQLite không có máy chủ riêng biệt để xử lý
- Đặc điểm: gọn nhẹ, đơn giản. Chương trình gồm 1 file duy nhất, không cần cài đặt, không cần cấu hình mà có thể sử dụng ngay
- Dữ liệu database được lưu vào một file duy nhất. Không có khái niệm user, password hay quyền hạn tron database

Tại sao nên dùng SQLite

- SQLite không yêu cầu một tiến trình Server riêng rẽ để hoạt động.
- SQLite không cần cấu hình, nghĩa là không cần thiết phải cài đặt.
- Một SQLite Database đầy đủ được lưu giữ trong một disk file đơn.SQLite là rất nhỏ gọn, nhỏ hơn 400kB đã đươc cấu hình đầy đủ hoặc nhỏ hơn 250kB khi đã bỏ qua các tính năng tùy ý.
- SQLite là tự chứa, nghĩa là không có sự phụ thuộc vào ngoại vi.
- Các Transaction trong SQLite là tuân theo đầy đủ chuẩn ACID, đảm bảo truy cập an toàn từ nhiều tiến trình hoặc thread.
- SQLite hỗ trợ hầu hết các tính năng của một ngôn ngữ truy vấn trong chuẩn SQL92.

Lớp lưu trữ trong SQLite:

Mỗi giá trị được lưu giữ trong một SQLite Database có một trong các lớp lưu trữ (Storage Class)

Lớp lưu trữ	Miêu tả
NULL	Giá trị là một giá trị NULL
INTEGER	Giá trị là một số nguyên có dấu, được lưu giữ trong 1, 2, 3, 4, 6, hoặc 8 byte tùy thuộc vào độ lớn của giá trị
REAL	Giá trị số thực dấu chấm động, được lưu giữ như là một số thực dấu chấm động 8-byte IEEE
TEXT	Giá trị là một text string, được lưu trữ bởi sử dụng Encoding của cơ sở dữ liệu (UTF-8, UTF-16BE hoặc UTF-16LE)
BLOB	Giá trị là một blob của dữ liệu, nhập vào như thế nào thì lưu giữ chính xác như thế

✓**Kiểu dữ liệu Boolean trong SQLite**

SQLite không hỗ trợ lớp lưu trữ Boolean riêng rẽ. Thay vào đó, các giá trị Boolean được lưu trữ dưới dạng các số nguyên: 0 cho false và 1 cho true.

✓**Kiểu dữ liệu Date và Time trong SQLite**

SQLite không có một lớp lưu trữ riêng rẽ để lưu trữ date/time, nhưng SQLite có thể lưu giữ date/time dưới dạng các giá trị TEXT, REAL hoặc INTEGER.

Có thể chọn để lưu giữ date và time trong bất kỳ các kiểu định dạng này và tự do chuyển đổi giữa các định dạng bởi sử dụng các hàm xử lý date và time có sẵn.

Lớp lưu trữ	Định dạng Date
TEXT	Một date trong định dạng "YYYY-MM-DD HH:MM:SS.SSS"
REAL	Số ngày từ Greenwich November 24, 4714 B.C
INTEGER	Số giây từ 1970-01-01 00:00:00 UTC

2

Sử dụng SQLiteOpenHelper

CLASS HỖ TRỢ CỦA SQLITE

Android cung cấp hai Class hỗ trợ cho việc tạo và quản lý Database:

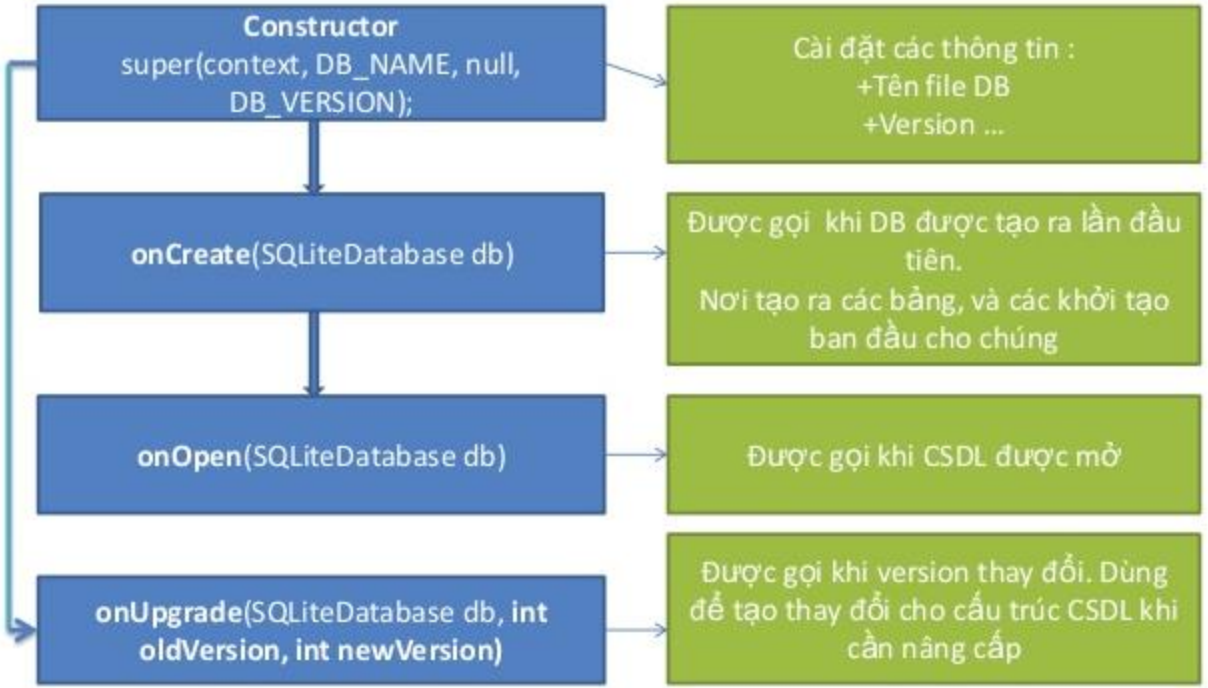
- ✓ SQLiteDatabase: sử dụng đơn giản
- ✓ SQLiteOpenHelper: sử dụng chuyên nghiệp

Class SQLiteOpenHelper hỗ trợ quản lý Database và version SQLite.

3 phương thức của Class:

1. onCreate(SQLiteDatabase)
2. onUpgrade(SQLiteDatabase, int, int)
3. onOpen(SQLiteDatabase)

CLASS SQLITEOPENHELPER



Các bước thực hiện:

1. **Tạo các Class kế thừa SQLiteOpenHelper**

2. **Override phương thức onCreate:**
   - Tạo Table trong phương thức này

3. **Override phương thức onUprade:**
   - Phương thức này sẽ được gọi khi ta nâng version mới, cần xoá các table cũ và gọi lại onCreate

Ví dụ tạo database Demo6 và Table Nhanvien

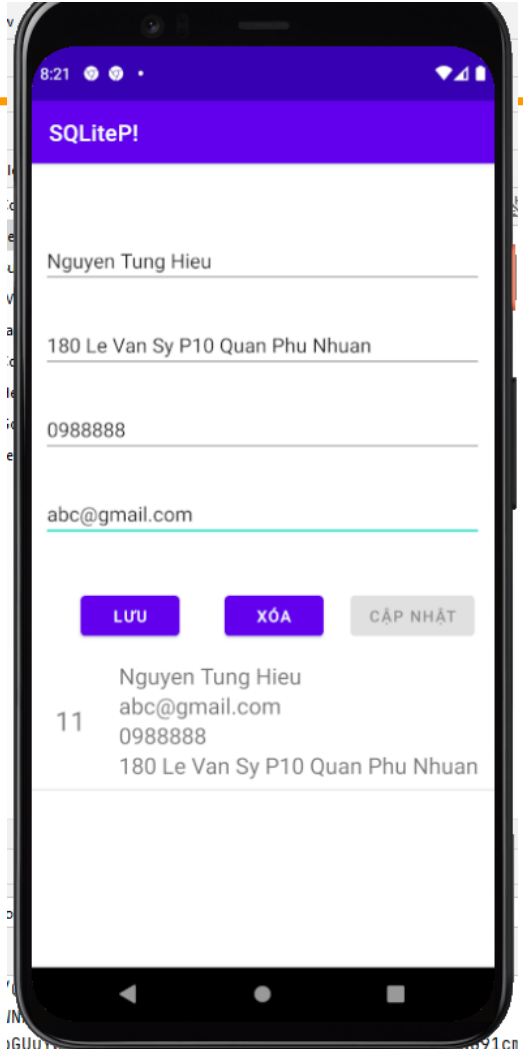
```
package com.kietlpt.quanlynhanvien.SQLite;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DbHelper extends SQLiteOpenHelper {
    public static final String DB_NAME = "Demo6";
    public static final int DB_VERSION = 1;

    public DbHelper(Context context) { super(context, DB_NAME, null, DB_VERSION); }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTableSql =
                "CREATE TABLE nhanvien (" +
                "id TEXT PRIMARY KEY,  " +
                "salary INTEGER NOT NULL, " +
                "name TEXT NOT NULL)";
        db.execSQL(createTableSql);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        String dropTableSql = "DROP TABLE IF EXISTS students";
        db.execSQL(dropTableSql);
        onCreate(db);
    }
}
```

Giải thích code:

✓ Hàm dựng **DbHelper**:
Tạo database Demo6 với version 1

✓ Hàm **onCreate**:
Thực hiện viết code tạo table Nhanvien

✓ Hàm **onUpgrade**:
Thực hiện xoá và gọi lại onCreate nếu có version mới.


8:21
SQLiteP!
Nguyen Tung Hieu
180 Le Van Sy P10 Quan Phu Nhuan
0988888
abc@gmail.com
LƯU
XÓA
CẬP NHẬT
11
Nguyen Tung Hieu
abc@gmail.com
0988888
180 Le Van Sy P10 Quan Phu Nhuan

TẠO CLASS STUDENT


```
private int mID;
private String mName;
private String mAddress;
private String mPhoneNumber;
private String mEmail;
```

```
public Student() {
}
public Student(String mName, String mAddress, String mPhoneNumber,
String mEmail) {
   this.mName = mName;
   this.mAddress = mAddress;
   this.mPhoneNumber = mPhoneNumber;
   this.mEmail = mEmail;
}
public Student(int mID, String mName, String mAddress, String
mPhoneNumber, String mEmail) {
   this.mID = mID;
   this.mName = mName;
   this.mAddress = mAddress;
   this.mPhoneNumber = mPhoneNumber;
   this.mEmail = mEmail;
}
```

```
public class DBManager extends SQLiteOpenHelper

        private final String TAG = "ABC";
        private static final String DATABASE_NAME = "students_manager.sqlite";
        private static final int DATABASE_VERSION = 1;
        private Context context;

        //Tạo Table Sinh_Vien
        private String SinhVienTable = "CREATE TABLE sinhviens (" +
                "ID INTEGER PRIMARY KEY AUTOINCREMENT," +
                "NAME TEXT," +
                "EMAIL TEXT, " +
                "PHONE_NUMBER TEXT, " +
                "ADDRESS TEXT)";
```

Tạo DBManager – Tạo Database và table

```
public class DBManager extends SQLiteOpenHelper

    public DBManager(Context context) {

       super(context, DATABASE_NAME, null, DATABASE_VERSION);
       this.context = context;
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
          db.execSQL(SinhVienTable);
          Log.d(TAG, "On Create");

    }
```

Tạo DBManager – Thêm Sinh Viên

```
public class DBManager extends SQLiteOpenHelper

    public void addStudent(Student student)
    {
       //Khởi tạo đối tượng
       SQLiteDatabase db = this.getWritableDatabase();

       ContentValues values = new ContentValues();
       values.put("NAME",student.getmName());
       values.put("ADDRESS",student.getmAddress());
       values.put("PHONE_NUMBER", student.getmPhoneNumber());
       values.put("EMAIL", student.getmEmail());
       db.insert("sinhviens",null,values);

       db.close();
       Log.d(TAG, "addStudent: Thành công ");
    }
```

Tạo DBManager – Lấy danh sách sinh viên


```
public class DBManager extends SQLiteOpenHelper
    public List<Student> getAllStudent(){
        SQLiteDatabase db = this.getWritableDatabase();
        List<Student> studentList = new ArrayList<>();
        String sql = "SELECT * FROM sinhviens";
        Cursor cursor = db.rawQuery(sql,null);
        if(cursor.moveToFirst()){
            do {
                Student student = new Student();
                student.setmID(cursor.getInt(0));
                student.setmName(cursor.getString(1));
                student.setmAddress(cursor.getString(2));
                student.setmPhoneNumber(cursor.getString(3));
                student.setmEmail(cursor.getString(4));
                studentList.add(student);
            }while (cursor.moveToNext());
        }
        db.close();
        return  studentList;
    }
```

Tạo DBManager – Update thông tin sinh viên

public class DBManager **extends** SQLiteOpenHelper

```
// Kiểu int trả về số reccord đc update
public int UpdateStudent(Student student){
   SQLiteDatabase db = this.getWritableDatabase();
   ContentValues values = new ContentValues();
   values.put("NAME",student.getmName());
   values.put("ADDRESS",student.getmAddress());
   values.put("PHONE_NUMBER", student.getmPhoneNumber());
   values.put("EMAIL", student.getmEmail());

   return db.update("sinhviens",values,"ID =" + student.getmID(),null);
   //db.update("sinhviens",values,"ID =? and NAME=?",new String[]
{String.valueOf(student.getmID()),student.getmName()});
}
```

Tạo DBManager – Xóa sinh viên

public class DBManager **extends** SQLiteOpenHelper

```
// Kiểu int trả về số reccord đc delete
public int DeleteStudent(int ID){
   SQLiteDatabase db = this.getWritableDatabase();
   return db.delete("sinhviens","ID =?",new String[] {String.valueOf(ID)});

}
```

Tạo Student Adapter cho Listview Student

```
public class StudentAdapter extends BaseAdapter {


    private Context context;
    private int layout;
    private List<Student> studentList;

    public StudentAdapter(Context context, int layout, List<Student>
    studentList) {
       this.context = context;
       this.layout = layout;
       this.studentList = studentList;
    }
```

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {

   LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
   convertView = inflater.inflate(layout,null);
   TextView txtId = convertView.findViewById(R.id.txtId);
   TextView txtHoten = convertView.findViewById(R.id.txtHoten);
   TextView txtDiachi = convertView.findViewById(R.id.txtDiachi);
   TextView txtPhonenumber = convertView.findViewById(R.id.txtPhonenumber);
   TextView txtEmail = convertView.findViewById(R.id.txtEmail);

   Student student = studentList.get(position);
   txtId.setText(String.valueOf(student.getmID()) );
   txtHoten.setText(student.getmName());
   txtDiachi.setText(student.getmAddress());
   txtPhonenumber.setText(student.getmPhoneNumber());
   txtEmail.setText(student.getmEmail());

   return convertView;
}
```

Khai báo cho Listview Student

```
List<Student> studentArrayList;
StudentAdapter studentAdapter;

studentArrayList = new ArrayList<>();
studentArrayList =  dbManager.getAllStudent();
studentAdapter = new StudentAdapter(this,R.layout.item_listview_student,studentArrayList);
lvStudent.setAdapter(studentAdapter);
```

```
private Student createStudent() {
    String name = edtName.getText().toString();
    String address = edtAddress.getText().toString();
    String phoneNumber = edtPhonenumber.getText().toString();
    String email = edtEmail.getText().toString();
    Student student= new Student(name,address,phoneNumber,email);
    return student;
}
```

```
btnLuu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Student student = createStudent();
        dbManager.addStudent(student);
        studentArrayList.clear();
        studentArrayList.addAll(dbManager.getAllStudent());
        studentAdapter.notifyDataSetChanged();
        lvStudent.setSelection(studentAdapter.getCount() - 1);
    }
});
```

Sự kiện click button Cập nhật

```
btnCapnhat.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Student student= new Student();
        student.setmID(curPosition) ;
        student.setmName(edtName.getText().toString());
        student.setmAddress(edtAddress.getText().toString());
        student.setmPhoneNumber(edtPhonenumber.getText().toString());
        student.setmEmail(edtEmail.getText().toString());
        int result = dbManager.UpdateStudent(student);
        if (result > 0 )
        {
            btnLuu.setEnabled(true);
            btnCapnhat.setEnabled(false);
            studentArrayList.clear();
            studentArrayList.addAll(dbManager.getAllStudent());
            studentAdapter.notifyDataSetChanged();
            lvStudent.setSelection(curPosition-1);
            Toast.makeText( MainActivity.this,"Update thành công", Toast.LENGTH_SHORT).show();
        }
        else
        {
            Toast.makeText( MainActivity.this,"Không update được", Toast.LENGTH_SHORT).show();
        }
    }
```

Sự kiện click Listview Student

```
lvStudent.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

        Student student = studentArrayList.get(position);
        curPosition = student.getmID();
        edtName.setText(student.getmName());
        edtAddress.setText(student.getmAddress());
        edtPhonenumber.setText(student.getmPhoneNumber());
        edtEmail.setText(student.getmEmail());
        btnCapnhat.setEnabled(true);
        btnLuu.setEnabled(false);


    }
});
```

Sự kiện long click Listview Student – Delete Sinh Viên SQLite

```
lvStudent.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
    @Override
    public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {

        Student student = studentArrayList.get(position);
        int result = dbManager.DeleteStudent(student.getmID());
        if(result>0){
            studentArrayList.clear();
            studentArrayList.addAll(dbManager.getAllStudent());
            studentAdapter.notifyDataSetChanged();
            lvStudent.setSelection(position-1);
        }
        return false;
    }
});
```

TỔNG KẾT NỘI DUNG BÀI HỌC

- 📖 Phần I: SQLite
  - 📖 Giới thiệu về SQLite
  - 📖 Các kiểu dữ liệu trong SQLite
- 📖 Phần II: Sử dụng SQLiteOpenHelper
  - 📖 SQLiteOpenHelper
  - 📖 Tạo Database và Table


PTIT
ptit.edu.vn