

Mỗi ứng dụng đều sử dụng dữ liệu, dữ liệu có thể đơn giản hay đôi khi là cả 1 cấu trúc. Trong Android thì hệ cơ sở dữ liệu được sử dụng là SQLite Database, đây là hệ thống mã nguồn mở được sử dụng rộng rãi trong các ứng dụng.

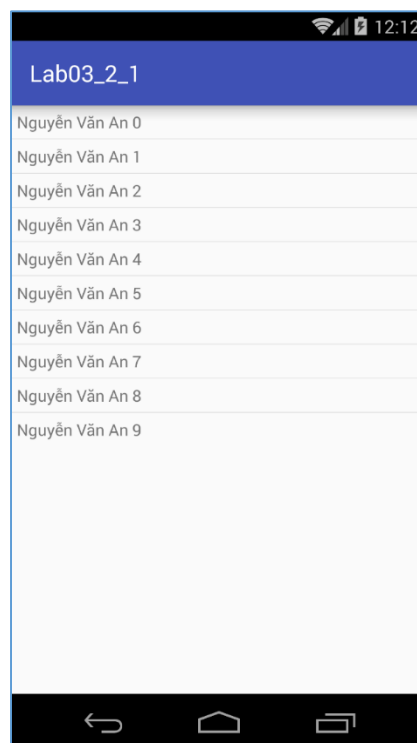
Trong Android , cơ sở dữ liệu mà bạn tạo cho 1 ứng dụng thì chỉ ứng dụng đó có quyền truy cập và sử dụng, các ứng dụng khác thì không. Khi đã được tạo, cơ sở dữ liệu SQLite được chứa trong thư mục **/data/data/<package_name>/databases**

Phần 1: Thực hiện tạo, thêm, cập nhật, xóa, truy vấn CSDL với SQLite

Các bước thực hiện với CSDL:

1. Tạo 1 CSDL (thông thường chỉ cần làm 1 lần)
2. Mở CSDL đó
3. Thêm giá trị vào trong table
4. Truy vấn.
5. Đóng CSDL

1. **[Tham khảo] tạo 1 CSDL SQLite, thêm vào CSDL danh sách các user, sau đó lấy dữ liệu từ CSDL hiện lên listview:**



- Nội dung activity_main.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="vertical">

    <ListView
        android:id="@+id/lv_user"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
```

- Nội dung file item_user.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/tv_user"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="@dimen/margin_base"/>
```

- Tạo 1 class **DatabaseHelper**. Override 2 phương thức onCreate() và onUpgrade() để quản lý việc tạo CSDL và version của CSDL đó.

```
package lab01_1.uit.edu.vn.lab03_2_1.helper;

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

public class DatabaseHelper extends SQLiteOpenHelper {

    private static final String DATABASE_CREATE = "create table users (_id\ninteger primary key autoincrement, "\n\n+ "name text not null);";

    public DatabaseHelper(Context context, String name,\nSQLiteDatabase.CursorFactory factory, int version) {\n        super(context, name, factory, version);\n        // TODO Auto-generated constructor stub\n    }

    @Override\n    public void onCreate(SQLiteDatabase db) {\n        // TODO Auto-generated method stub\n        db.execSQL(DATABASE_CREATE);\n    }

    @Override\n    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)\n    {\n        // TODO Auto-generated method stub\n        db.execSQL("DROP TABLE IF EXISTS users");\n        onCreate(db);\n    }\n}
```

```
}  
}
```

- Đầu tiên các bạn add 1 class **DbAdapter** để xử lý tất cả các thao tác liên quan đến CSDL. Tất cả các chức năng đó đều được cung cấp bởi lớp SQLiteDatabase, các bạn chỉ cần cụ thể hóa bằng các câu truy vấn là được. Trong ví dụ này sẽ có các hàm mở, đóng CSDL; thêm, xóa, get user...

```
package lab01_1.uit.edu.vn.lab03_2_1.adapter;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
  
import lab01_1.uit.edu.vn.lab03_2_1.helper.DatabaseHelper;  
  
public class DbAdapter {  
    public static final String KEY_ID = "_id";  
    public static final String KEY_NAME = "name";  
  
    private DatabaseHelper dbHelper;  
    private SQLiteDatabase sqLiteDatabase;  
    private static final String DATABASE_NAME = "Database_Demo";  
    private static final String DATABASE_TABLE = "users";  
    private static final int DATABASE_VERSION = 2;  
    private final Context context;  
  
    public DbAdapter(Context ctx) {  
        this.context = ctx;  
    }  
  
    public DbAdapter open() {  
        dbHelper = new DatabaseHelper(context, DATABASE_NAME, null,  
DATABASE_VERSION);  
        sqLiteDatabase = dbHelper.getWritableDatabase();  
        return this;  
    }  
  
    public void close() {  
        dbHelper.close();  
    }  
  
    public long createUser(String name) {  
        ContentValues inititalValues = new ContentValues();  
        inititalValues.put(KEY_NAME, name);  
        return sqLiteDatabase.insert(DATABASE_TABLE, null, inititalValues);  
    }  
  
    public boolean deleteUser(long rowId) {  
        return sqLiteDatabase.delete(DATABASE_TABLE, KEY_ID + "=" + rowId,  
null) > 0;  
    }  
}
```

```
    }

    public boolean deleteAllUsers() {
        return SQLiteDatabase.delete(DATABASE_TABLE, null, null) > 0;
    }

    public Cursor getAllUsers() {
        return SQLiteDatabase.query(DATABASE_TABLE, new String[]{KEY_ID,
KEY_NAME}, null, null, null, null, null);
    }
}
```

- Nội dung MainActivity

```
package vn.edu.uit.lab03_2_1;

import android.database.Cursor;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import java.util.ArrayList;
import java.util.List;

import vn.edu.uit.lab03_2_1.adapter.DbAdapter;

public class MainActivity extends ActionBarActivity {

    private DbAdapter dbAdapter;
    private Cursor cursor;
    private List<String> users;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        dbAdapter = new DbAdapter(this);
        dbAdapter.open();
        dbAdapter.deleteAllUsers();

        for (int i = 0; i < 10; i++) {
            dbAdapter.createUser("Nguyễn Văn An " + i);
        }

        users = getData();
        showData();
    }

    private List<String> getData() {
        List<String> users = new ArrayList<>();

        cursor = dbAdapter.getAllUsers();
        while (cursor.moveToNext()) {
            users.add(cursor.getString(cursor.getColumnIndex(DbAdapter.KEY_NAME)));
        }
    }
}
```

```

        return users;
    }

    private void showData() {
        ListView lvUser = (ListView) findViewById(R.id.lv_user);
        ArrayAdapter<String> userAdapter = new
        ArrayAdapter<String>(MainActivity.this, R.layout.item_user, users);
        lvUser.setAdapter(userAdapter);
    }
}

```

2. [Thực hành] Quản lý Contact

2.1 CSDL dữ liệu có cấu trúc như sau:

- **Database Name:** contactsManager
- **Table Name:** contacts

Columns	DataType	Key
id	INTEGER	Primary key
name	TEXT	
phone_number	TEXT	

2.2 Hướng dẫn thực hiện

a) Tạo một lớp Contact:

```

public class Contact {
    private int id;
    private String name;
    private String phoneNumber;

    //...
}

```

b) Tạo 1 lớp DatabaseHandler kế thừa từ SQLiteOpenHelper

- Override hàm onCreate() để tạo table
- Override hàm onUpgrade() hủy table cũ, tạo table mới
- Thêm các phương thức để insert, select, update và delete contact

```
import java.util.ArrayList;
import java.util.List;

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

public class DatabaseHandler extends SQLiteOpenHelper {

    // All Static variables
    // Database Version
    private static final int DATABASE_VERSION = 1;

    // Database Name
    private static final String DATABASE_NAME = "contactsManager";

    // Contacts table name
    private static final String TABLE_CONTACTS = "contacts";

    // Contacts Table Columns names
    private static final String KEY_ID = "id";
    private static final String KEY_NAME = "name";
    private static final String KEY_PH_NO = "phone_number";

    public DatabaseHandler(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    // Creating Tables
    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_CONTACTS_TABLE = "CREATE TABLE " +
TABLE_CONTACTS + "("
            + KEY_ID + " INTEGER PRIMARY KEY," + KEY_NAME + "
TEXT,"
            + KEY_PH_NO + " TEXT" + ")";
        db.execSQL(CREATE_CONTACTS_TABLE);
    }

    // Upgrading database
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
        // Drop older table if existed
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_CONTACTS);

        // Create tables again
        onCreate(db);
    }

    // Các hàm còn lại: để insert, select, update và delete contact sv tự viết
```

```
// Adding new contact
public void addContact(Contact contact) {}

// Getting single contact
public Contact getContact(int id) {}

// Getting All Contacts
public List<Contact> getAllContacts() {}

// Updating single contact
public int updateContact(Contact contact) {}

// Deleting single contact
public void deleteContact(Contact contact) {}
```

c) MainActivity.java

```
public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        DatabaseHandler db = new DatabaseHandler(this);

        /**
         * CRUD Operations
         */
        // Inserting Contacts
        Log.d("Insert: ", "Inserting ..");
        db.addContact(new Contact("Ravi", "9100000000"));
        db.addContact(new Contact("Srinivas", "9199999999"));
        db.addContact(new Contact("Tommy", "9522222222"));
        db.addContact(new Contact("Karthik", "9533333333"));

        // Reading all contacts
        Log.e("Reading: ", "Reading all contacts..");
        List<Contact> contacts = db.getAllContacts();

        for (Contact cn : contacts) {
            String log = "Id: "+cn.getID()+" ,Name: " + cn.getName() + "
,Phone: " + cn.getPhoneNumber();
            // Writing Contacts to log
            Log.e("Name: ", log);
        }
    }
}
```

Kết quả hiển thị hiện tại của ứng dụng (xem tại Log Cat)

Insert:	Inserting ..
Reading:	Reading all contacts..
Name:	Id: 1 ,Name: Ravi ,Phone: 9100000000
Name:	Id: 2 ,Name: Srinivas ,Phone: 9199999999
Name:	Id: 3 ,Name: Tommy ,Phone: 9522222222
Name:	Id: 4 ,Name: Karthik ,Phone: 9533333333
Name:	Id: 5 ,Name: Ravi ,Phone: 9100000000
Name:	Id: 6 ,Name: Srinivas ,Phone: 9199999999
Name:	Id: 7 ,Name: Tommy ,Phone: 9522222222
Name:	Id: 8 ,Name: Karthik ,Phone: 9533333333

Sinh viên tùy chỉnh lại code để hiển thị list Contact ra listview. Khi nhấn giữ lâu vào 1 contact nào đó (long click) thì ứng dụng xóa contact đó khỏi Database.

2.3 Bài thực hành nâng cao:

Quản lý danh sách sinh viên, thêm, xem, xóa, sửa, nhấn vào mỗi 1 sinh viên sẽ rat rang chi tiết 1 sinh viên có sử dụng Recyclerview và SQLite. Thiết kế giao diện theo cảm nhận của mỗi bạn.

Ví dụ:

QUẢN LÝ SINH VIÊN

MSSV

HỌ TÊN

LỚP

INSERT **DELETE**

UPDATE **QUERY**

22020311 - Hoàng An

22020723 - Cao Bình

Phần 2: Thực hiện tạo CSDL SQLITE từ bên ngoài, sao chép và sử dụng CSDL đã có sẵn

Các bước thực hiện với CSDL:

1. Sử dụng phần mềm để tạo 1 CSDL SQLite
2. Copy cơ sở dữ liệu này vào hệ thống

1. Sử dụng một phần mềm hỗ trợ để tạo 1 CSDL SQLite

Có nhiều phần mềm để tạo 1 CSDL SQLite: sử dụng phần mềm DB Browse (SQLite), SQLite Database browser, SQLite Manager (được tích hợp vào firefox).

Trong bài này hướng dẫn sử dụng DB Browse để tạo ra 1 CSDL

Bước 1: Download và cài đặt DB Browse for SQLite

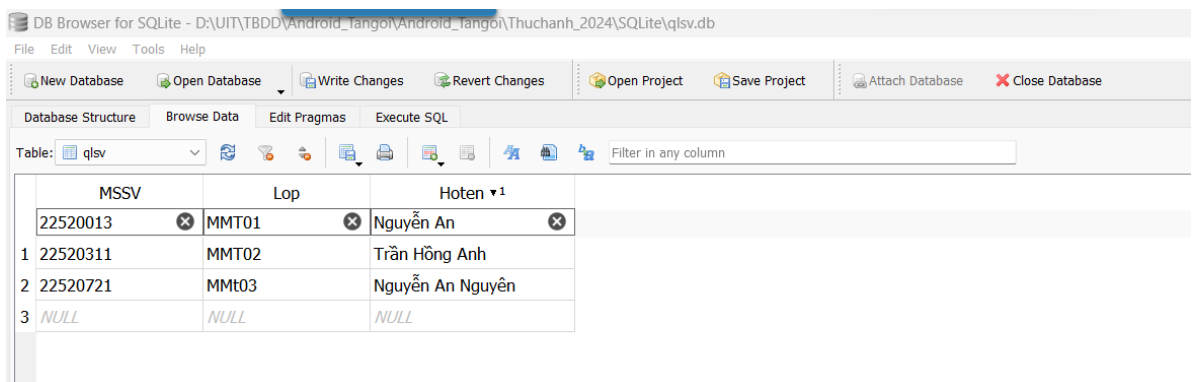
<https://sqlitebrowser.org/dl/>

Windows

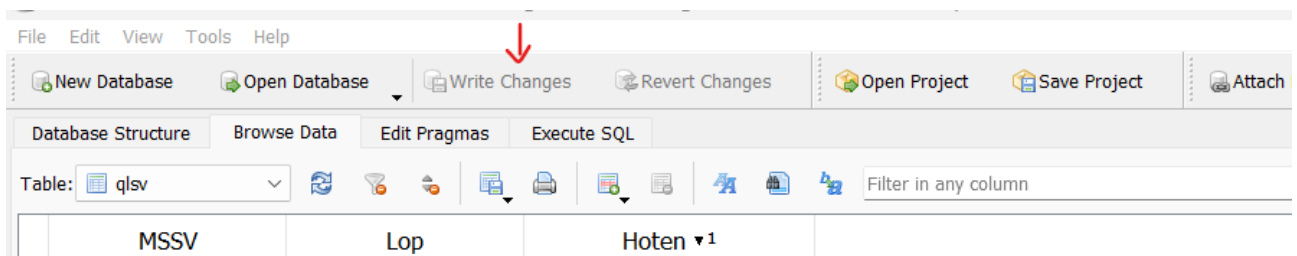
Our latest release (3.12.2) for Windows:

- [DB Browser for SQLite – Standard installer for 32-bit Windows](#)
- [DB Browser for SQLite – .zip \(no installer\) for 32-bit Windows](#)
- [DB Browser for SQLite – Standard installer for 64-bit Windows](#)
- [DB Browser for SQLite – .zip \(no installer\) for 64-bit Windows](#)

Bước 2: Sử dụng DB Browse for SQLite tạo CSDL *qlsv.db*



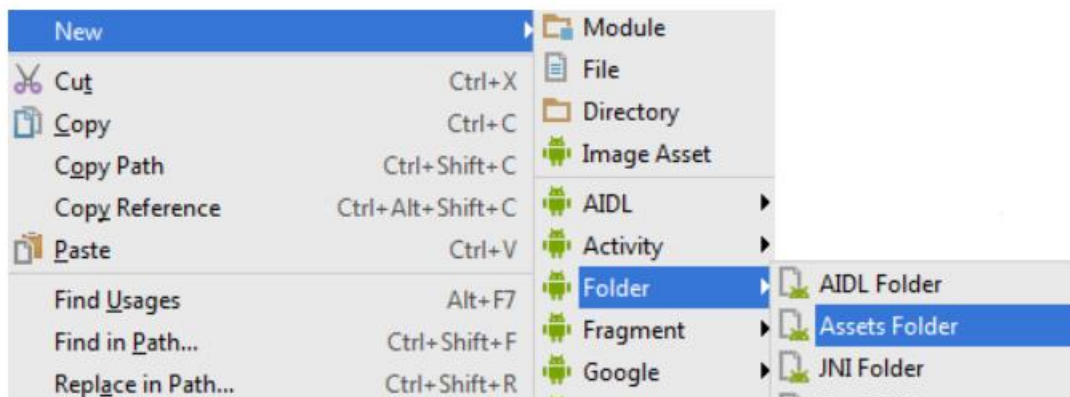
Click vào Write Changes để lưu database lại



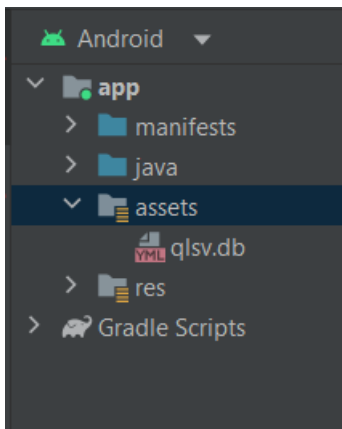
Sau khi tạo xong dữ liệu *qlsv.db*

2. Sử dụng phần mềm hỗ trợ để tạo 1 CSDL SQLite

Copy file *qlsv.db* vào thư mục **assets** trong android project, nếu không tìm thấy folder này thì tạo thư mục như sau



Copy file qlsv.db vào thư mục này



Sau đó thực hiện phần code trong MainActivity như sau:

```
public class MainActivity extends AppCompatActivity {
    String DB_PATH_SUFFIX = "/databases/";
    SQLiteDatabase database=null;
    String DATABASE_NAME="qlsv.db";
    //Khai báo ListView
    ListView lv;
    ArrayList<String> mylist;
    ArrayAdapter<String> myadapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //Ham Copy CSDL từ assets vào thư mục Databases
        processCopy();
        //Mo CSDL trong ung dung len
        database = openOrCreateDatabase("qlsv.db",MODE_PRIVATE, null);
        // Tạo ListView
        lv = findViewById(R.id.lv);
        mylist = new ArrayList<>();
    }
}
```

```
myadapter = new ArrayAdapter<>(MainActivity.this,
    android.R.layout.simple_list_item_1, mylist);
lv.setAdapter(myadapter);
// Truy vấn CSDL và cập nhật hiển thị lên Listview
Cursor c =
database.query("qlsv", null, null, null, null, null, null);
c.moveToFirst();
String data = "";
while (c.isAfterLast() == false)
{
    data = c.getString(0)+"-"+c.getString(1)+"-
"+c.getString(2);
    mylist.add(data);
    c.moveToNext();
}
c.close();
myadapter.notifyDataSetChanged();
}
private void processCopy() {
    //private app
    File dbFile = getDatabasePath(DATABASE_NAME);
    if (!dbFile.exists())
    {
        try {
            CopyDataBaseFromAsset();
            Toast.makeText(this, "Copying sucess from Assets
folder",
                        Toast.LENGTH_LONG).show();
        }
        catch (Exception e){
            Toast.makeText(this, e.toString(),
Toast.LENGTH_LONG).show();
        }
    }
    private String getDatabasePath() {
        return getApplicationInfo().dataDir + DB_PATH_SUFFIX+
DATABASE_NAME;
    }
    // Ham copy file DB tu thu muc Asset vao file DB moi tao ra trong
    ung dung
    public void CopyDataBaseFromAsset() {
        try {
            InputStream myInput;
            myInput = getAssets().open(DATABASE_NAME);
            String outFileName = getDatabasePath();
            // Kiem tra neu duong dan khong co, thi tao moi file
            File f = new File(getApplicationInfo().dataDir +
DB_PATH_SUFFIX);
            if (!f.exists()) f.mkdir();
            // Mo empty db su dung output stream
            OutputStream myOutput = new FileOutputStream(outFileName);
            // Sao chep du lieu bytes tu input toi ouput
            int size = myInput.available();
```

```
byte[] buffer = new byte[size];
myInput.read(buffer);
myOutput.write(buffer);
myOutput.flush();
myOutput.close();
myInput.close();
} catch (IOException e) {
    e.printStackTrace();
}
}
```

Phần tiếp theo các em sẽ lấy dữ liệu được tạo ra để hiển thị danh sách sinh viên lên ListView/RecyclerView nhé!

THAM KHẢO THÊM (Tự xem thêm)

1. Tham khảo thêm cách sử dụng Room

<https://developer.android.com/training/data-storage/room>

2. Cách debug trong Android Studio

2.1 Hướng dẫn: <http://blog.strv.com/debugging-in-android-studio-as/>

2.2 Một số phím thường dùng khi debug:

<http://www.praveenboyalapalli.com/2015/03/how-to-debug-project-android-studio/>

3. GreenDao

3.1 <http://greenrobot.org/greendao/>

3.2 <http://greenrobot.org/greendao/documentation/how-to-get-started/>

3.3 <https://github.com/greenrobot/greenDAO>

4. EventBus; cách dùng interface trong Java

4.1 Interface: http://vietjack.com/java/interface_trong_java.jsp;

<http://www.nguyenvanquan7826.com/2015/04/05/java-su-dung-interface/>

4.2 EventBus: <https://github.com/greenrobot/EventBus>

5. NinePatch

5.1 <https://developer.android.com/studio/write/draw9patch.html>