



**FPT POLYTECHNIC**

LẬP TRÌNH ANDROID CƠ BẢN

## **Bài 7: Content Provider và SQLite**

---

[www.poly.edu.vn](http://www.poly.edu.vn)

---

## Nội dung bài học

- Content Provider
- SQLite



## Content Provider

- Content Provider là một trong 4 thành phần cơ bản của Android
- Content Provider được sử dụng để quản lý và chia sẻ dữ liệu giữa các ứng dụng

```
Browser
CallLog
Contacts
    people
    phones
    photos
    groups
MediaStore
    audio
        albums
        artists
        Geners
        playlists
    images
        thumbnails
    video
Settings
```

Content Provider	Intended Data
Browser	Browser bookmarks, browser history, etc.
CallLog	Missed calls, call details, etc.
Contacts	Contact details
MediaStore	Media files such as audio, video and images
Settings	Device settings and preferences

Một số Content Provider có sẵn hay được sử dụng

## Sử dụng Content Provider

- Truy vấn Content Provider, cung cấp một xâu truy vấn dưới dạng URI, như sau:

**<standard\_prefix>://<authority>/<data\_path>/<id>**

- Ví dụ để lấy thông tin bookmark lưu trữ trong trình duyệt của Android, sử dụng URI sau:

**content://browser/bookmarks**

- Để lấy thông tin tất cả liên hệ của ứng dụng Contacts, sử dụng URI sau:

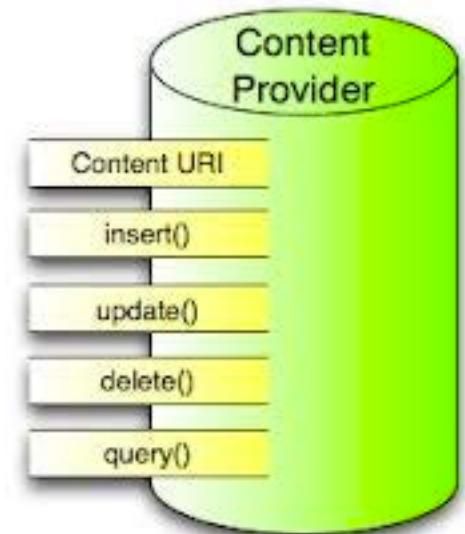
**content://contacts/people**

- Nhận thông tin của một liên hệ với ID

**content://contacts/people/3**

## Sử dụng Content Provider

- Một ứng dụng truy cập dữ liệu từ Content Provider sử dụng đối tượng khách ContentResolver
- Các phương thức ContentResolver cung cấp 3 hàm cơ bản CRUD
- Để truy cập provider, thông thường ứng dụng của bạn phải bổ sung thêm quyền vào android manifest.



## Sử dụng Content Provider

- Nhận con trỏ

```
CursorLoader loader=new
```

```
CursorLoader(context, uri, null, null, null, null);
```

```
Cursor c=loader.loadInBackground();
```

- Hoặc sử dụng

```
Cursor c = getContentResolver()
```

```
.query(uri, null, null, null, null);
```

- `getContentResolver` trả về một đối tượng `ContentResolver` giúp giải quyết Content URI với một Content Provider thích hợp
- Tham số: URI, projection, SQLWHERE, ORDERBY

## Sử dụng Content Provider

- Ví dụ để lấy danh sách từ trong User Dictionary Provider

```
mCursor = getContentResolver().query(  
    UserDictionary.Words.CONTENT_URI, // Content  
    URI    mProjection,                // Số cột trên mỗi  
    dòng    mSelectionClause            // Tiêu chí lựa  
    chọn    mSelectionArgs,            // Tiêu chí lựa  
    chọn    mSortOrder);               // Trật tự sắp xếp của dữ liệu  
 trả về
```

# Sử dụng Content Provider

```
public void showAllContacts()
{
    Uri uri=Uri.parse("content://contacts/people");
    ArrayList<String> list=new ArrayList<String>();
    CursorLoader loader=new
        CursorLoader(this, uri, null, null, null, null);
    Cursor cl=loader.loadInBackground();
    cl.moveToFirst();
    while(cl.isAfterLast()==false) {
        String s="";
        String idColumnName=ContactsContract.Contacts._ID;
        int idIndex=cl.getColumnIndex(idColumnName);
        s=cl.getString(idIndex)+" - ";
        String nameColumnName=ContactsContract.Contacts.DISPLAY_NAME;
        int nameIndex=cl.getColumnIndex(nameColumnName);
        s+=cl.getString(nameIndex);
        cl.moveToNext();
        list.add(s);
    }
    cl.close();
    ListView lv=(ListView) findViewById(R.id.listView1);
    ArrayAdapter<String>adapter=new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, list);
    lv.setAdapter(adapter);
}
```

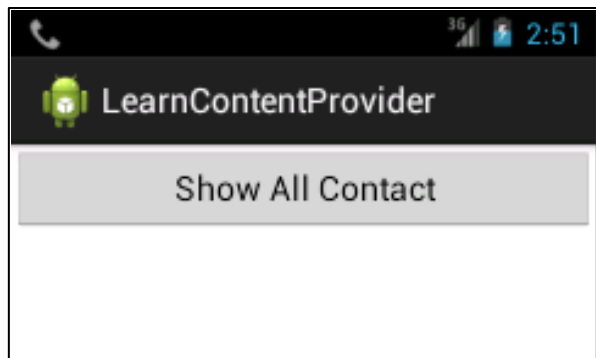


## Sử dụng Content Provider

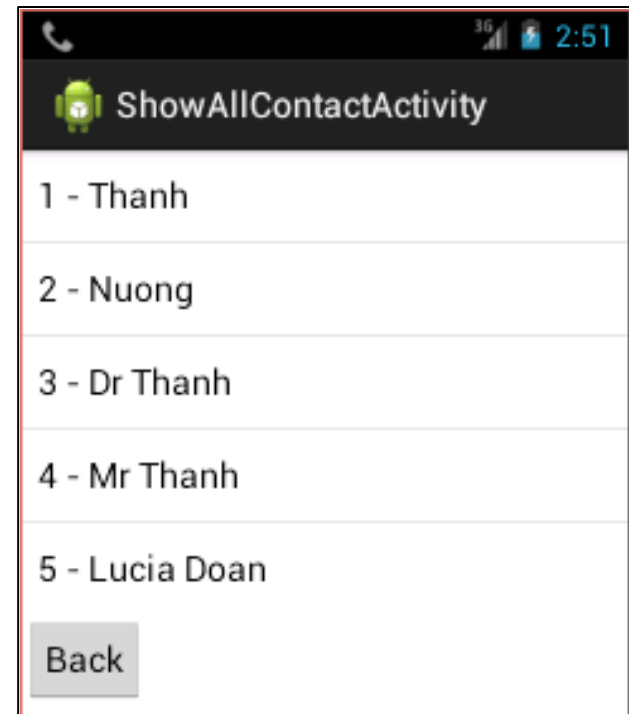
Thông tin liên hệ:

Sử dụng getContentResolver thay vì CursorLoader:

```
Uri uri=Uri.parse("content://contacts/people");
ArrayList<String> list=new ArrayList<String>();
Cursor cl=getContentResolver()
        .query(uri, null, null, null, null);
```



```
<uses-permission
    android:name=
        "android.permission.READ_CONTACTS"
/>
```



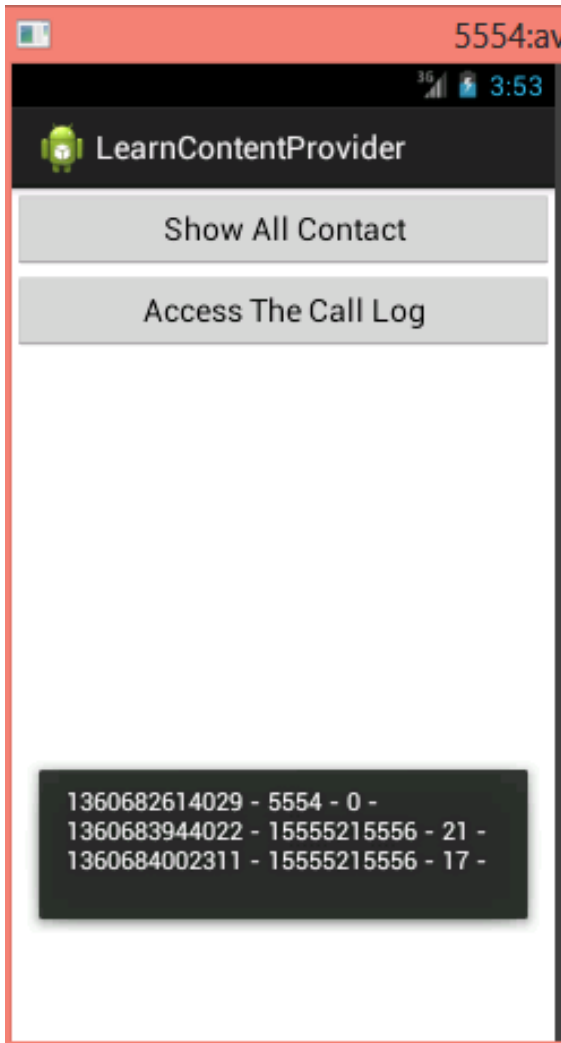
# Sử dụng Content Provider

## Lịch sử cuộc gọi:

```
public void accessTheCallLog()
{
    String [] projection=new String[]{
        Calls.DATE,
        Calls.NUMBER,
        Calls.DURATION
    };
    Cursor c=getContentResolver().query(
        CallLog.Calls.CONTENT_URI,
        projection,
        Calls.DURATION+"<?",new String[]{"30"},
        Calls.DATE +" Asc");
    c.moveToFirst();
    String s="";
    while(c.isAfterLast()==false) {
        for(int i=0;i<c.getColumnCount();i++) {
            s+=c.getString(i)+" - ";
        }
        s+="\n";
        c.moveToNext();
    }
    c.close();
    Toast.makeText(this, s, Toast.LENGTH_LONG).show();
}
```

# Sử dụng Content Provider

Lịch sử cuộc gọi:



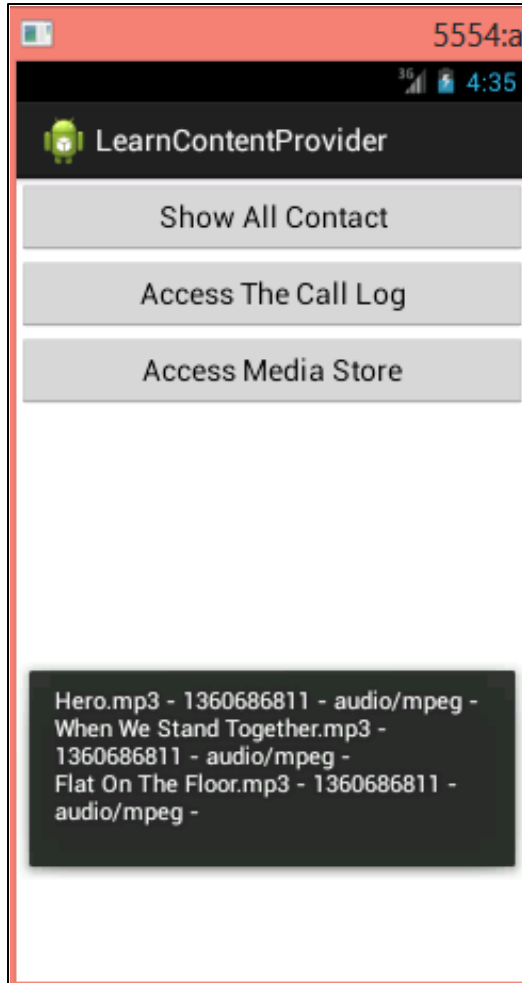
```
<uses-permission  
    android:name=  
        "android.permission.READ_CALL_LOG"  
/>
```

Tương tự như Contact, bạn có thể sử dụng lớp **CursorLoader** để truy cập call log

## Sử dụng Content Provider Access Media Store:

```
public void accessMediaStore()
{
    String []projection={
        MediaStore.MediaColumns.DISPLAY_NAME,
        MediaStore.MediaColumns.DATE_ADDED,
        MediaStore.MediaColumns.MIME_TYPE
    };
    CursorLoader loader=new CursorLoader
        (this, Media.EXTERNAL_CONTENT_URI,
         projection, null, null, null);
    Cursor c=loader.loadInBackground();
    c.moveToFirst();
    String s="";
    while(!c.isAfterLast()){
        for(int i=0;i<c.getColumnCount();i++){
            s+=c.getString(i)+" - ";
        }
        s+="\n";
        c.moveToNext();
    }
    Toast.makeText(this, s, Toast.LENGTH_LONG).show();
    c.close();
}
```

# Sử dụng Content Provider Access Media Store:



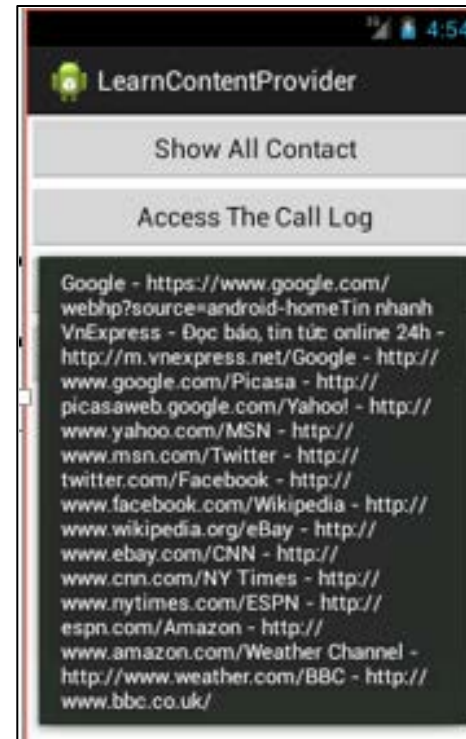
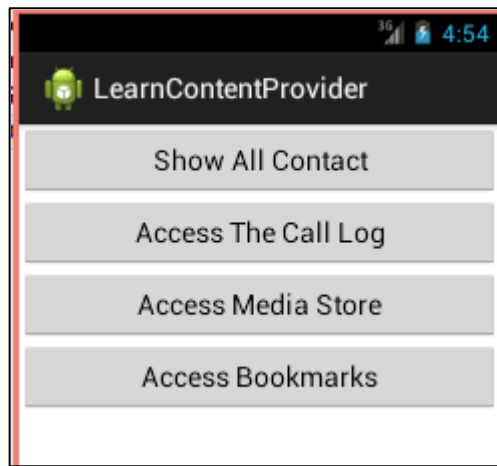
```
<uses-permission  
    android:name=  
        "android.permission.READ_EXTERNAL_STORAGE"  
/>
```

Tương tự như provider khác, bạn có thể sử dụng `getContentResolver` để truy cập media

## Sử dụng Content Provider Truy cập Bookmark:

```
public void accessBookmarks()
{
    String []projection={
        Browser.BookmarkColumns.TITLE,
        Browser.BookmarkColumns.URL,
    };
    Cursor c=getContentResolver()
        .query(Browser.BOOKMARKS_URI, projection,
            null, null, null);
    c.moveToFirst();
    String s="";
    int titleIndex=c.getColumnIndex
        (Browser.BookmarkColumns.TITLE);
    int urlIndex=c.getColumnIndex
        (Browser.BookmarkColumns.URL);
    while(!c.isAfterLast())
    {
        s+=c.getString(titleIndex)+" - "+
            c.getString(urlIndex);
        c.moveToNext();
    }
    c.close();
    Toast.makeText(this, s, Toast.LENGTH_LONG).show();
}
```

# Sử dụng Content Provider Truy cập Bookmark:



```
<uses-permission  
    android:name=  
        "com.android.browser.permission.READ_HISTORY_BOOKMARKS"  
/>
```



**DEMO**

- Đọc dữ liệu từ Contact





# SQLite

- Giới thiệu
- Tạo cơ sở dữ liệu
- Tạo bảng
- Truy vấn: thêm, sửa, xóa
- Truy vấn SQLite
- Demo



## SQLite

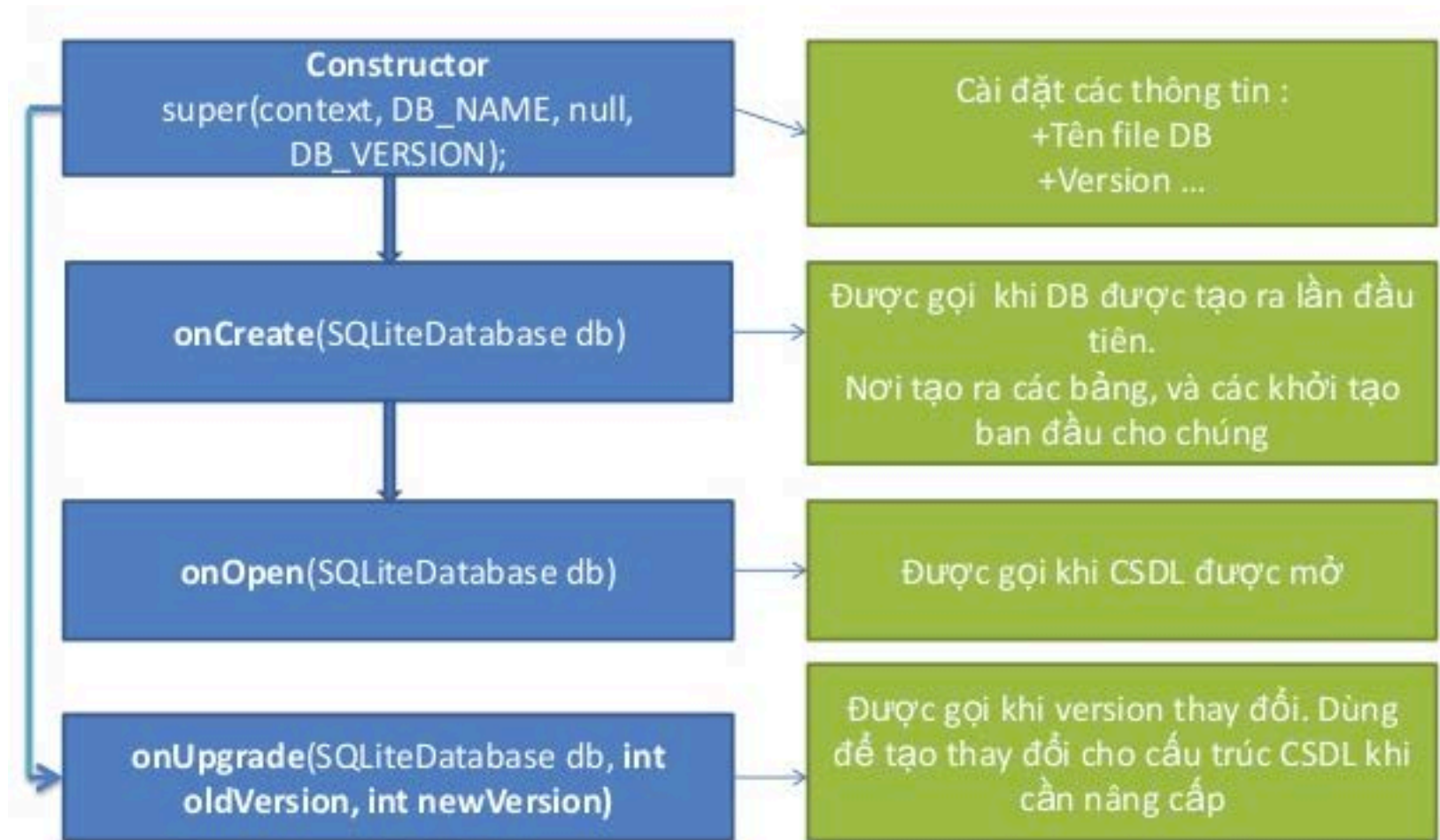
- SQLite là phần mềm quản lý cơ sở dữ liệu SQL nhưng không giống như hầu hết các cơ sở dữ liệu SQL khác, SQLite không có máy chủ riêng biệt để xử lý
- Đặc điểm: gọn nhẹ, đơn giản. Chương trình gồm 1 file duy nhất, không cần cài đặt, không cần cấu hình mà có thể sử dụng ngay
- Dữ liệu database được lưu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

## Cài đặt và sử dụng SQLite

- SQLiteOpenHelper: một lớp trợ giúp hỗ trợ tạo ra CSDL và quản lý phiên bản cho chúng
- SQLiteDatabase: lớp chứa các phương thức dùng để quản lý CSDL như INSERT, UPDATE, DELETE, SELECT,...



# SQLiteOpenHelper



## Tạo database

- Cách đơn giản nhất là tạo thể hiện SQLiteDatabase cho ứng dụng của bạn sử dụng phương thức openOrCreateDatabase() của Context

**import** android.database.sqlite.SQLiteDatabase;

```
public void doCreateDb()  
{  
    SQLiteDatabase database;  
    database=openOrCreateDatabase(  
        "qlsinhvien.db",  
        MODE_PRIVATE,  
        null);  
}
```

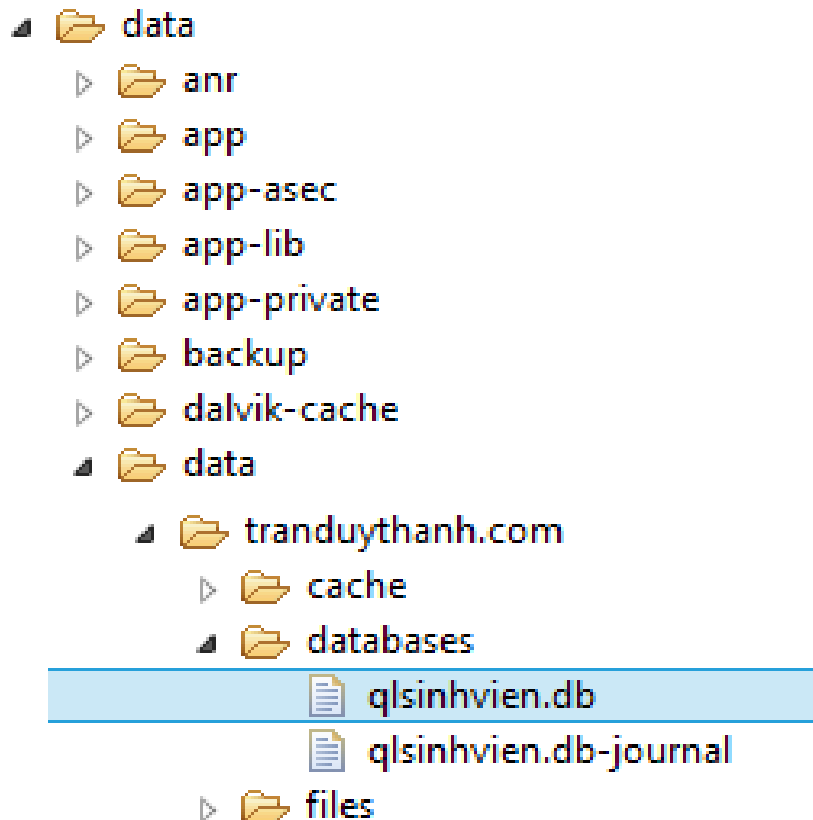
```
database= SQLiteDatabase.openDatabase(  
    "qlsinhvien.db",  
    null,  
    SQLiteDatabase.CREATE_IF_NECESSARY);
```

# Tạo database

Thư mục lưu trữ cơ sở dữ liệu:

/data/data/app/databases/<databasename>

Bạn có thể tìm thấy file cơ sở dữ liệu qua DBMS:



## Tạo database

- Bạn không thể truy cập cơ sở dữ liệu của ứng dụng Android khác (nên sử dụng Content Provider hoặc lưu trữ cơ sở dữ liệu SQLite trên SDCard)
- Nếu lưu trữ SQLite trên SDCard thì phải bổ sung quyền vào Android Manifest như sau:

<uses-permission

android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

```
public void doDeleteDb()
{
    String msg="";
    if(deleteDatabase("qlsinhvien.db")==true)
    {
        msg="Delete database [qlsinhvien.db] is successful";
    }
    else
    {
        msg="Delete database [qlsinhvien.db] is failed";
    }
    Toast.makeText(this, msg, Toast.LENGTH_LONG).show();
}
```

## Tạo bảng

```
public void doCreateLopTable()  
{  
    String sql="CREATE TABLE tbllop ("  
        sql+="malop TEXT primary key,";  
        sql+="tenlop TEXT,";  
        sql+="siso INTEGER)";  
    database.execSQL(sql);  
}
```

```
public void doCreateSinhvienTable()  
{  
    String sql="CREATE TABLE tblsinhvien (" +  
        "masv TEXT PRIMARY KEY ,"+  
        "tensv TEXT,"+  
        "malop TEXT NOT NULL CONSTRAINT malop "+  
        " REFERENCES tbllop(malop) ON DELETE CASCADE)";  
    database.execSQL(sql);  
}
```



## Insert dữ liệu

- Sử dụng phương thức insert() để thêm dữ liệu
- Sử dụng đối tượng ContentValues để nhóm cặp tên trường, giá trị của trường cho bản ghi muốn thêm

```
public void doInsertRecord()
{
    ContentValues values=new ContentValues();
    values.put("malop", "DHTH7C");
    values.put("tenlop", "Đại học 7C");
    values.put("siso", 30);
    String msg="";
    if(database.insert("tbllop", null, values)==-1){
        msg="Failed to insert record";
    }
    else{
        msg="insert record is successful";
    }
    Toast.makeText(this, msg, Toast.LENGTH_LONG).show();
}
```

## Cập nhật dữ liệu

- Bạn có thể cập nhật dữ liệu sử dụng phương thức update() gồm 4 tham số sau:
  - Bảng cần cập nhật
  - Đối tượng ContentValues chứa các giá trị cần cập nhật
  - Mệnh đề WHERE, truyền tham số dưới dạng dấu ?
  - Mảng tham số của mệnh đề WHERE
- Phương thức này trả về số dòng bị ảnh hưởng

**public int update (String table, ContentValues values,  
String whereClause, String[] whereArgs)**

## Cập nhật dữ liệu

```
public void updateLopName(String malop, String new_tenlop)
{
    ContentValues values=new ContentValues();
    values.put("tenlop", new_tenlop);
    int ret=database.update("tbllop", values,
        "malop=?", new String[]{malop});
    if(ret==0) {
        //failed;
    }
    else{
        //ok
    }
}
```

Bởi vì chúng ta không cập nhật các giá trị trường khác nên không cần thiết đưa vào đối tượng **ContentValues**. Chúng ta chỉ thêm vào **tenlop** vì chỉ cần thay đổi thông tin tên lớp

## Xóa dữ liệu

- Bạn có thể xóa các bản ghi từ bảng sử dụng phương thức delete() gồm 3 tham số sau:
  - Tên bảng có bản ghi muốn xóa
  - Mệnh đề WHERE gồm nhiều tham số, mỗi dấu ? thể hiện là một tham số
  - Một mảng giá trị ứng với các tham số ở trên
- Truyền null vào mệnh đề WHERE sẽ xóa toàn bộ bản ghi trong bảng

**public int delete (String table,  
String whereClause,  
String[] whereArgs)**

Phương pháp này trả về số bản ghi bị ảnh hưởng

## Xóa dữ liệu

Xóa toàn bộ bản ghi trong bảng tbllop:

```
database.delete("tbllop", null, null);
```

Xóa bản ghi với malop="dhth7c":

```
String malop="DHTH7C";  
database.delete("tbllop",  
                "malop=?",  
                new String[]{malop});
```

## Truy vấn trong SQLite

Để lấy kết quả trả về của truy vấn chúng ta sử dụng cursor của lớp **android.database.Cursor**. Đối tượng Cursor cũng như con trỏ file cho phép truy cập ngẫu nhiên để truy vấn kết quả

**public Cursor query (String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy)**

## Truy vấn SQLite

- Table
- Columns
- Selections
- SelectionArgs
- groupBy
- Having
- orderBy



## Truy vấn SQLite

```
public void loadalllop()  
{  
    Cursor c=database.query("tbllop",  
        null, null, null, null, null, null);  
    c.moveToFirst();  
    String data="";  
    while(c.isAfterLast()==false)  
    {  
        data+=c.getString(0)+"-"+  
            c.getString(1)+"-"+  
            c.getString(2);  
        data+="\n";  
        c.moveToNext();  
    }  
    Toast.makeText(this, data, Toast.LENGTH_LONG).show();  
    c.close();  
}
```





# DEMO

- Đọc, sửa, xóa dữ liệu từ SQLite



# Tổng kết nội dung bài học

- Content Provider
- SQLite



**Kết thúc!**

