



## Chương 5 :

# CƠ SỞ DỮ LIỆU SQLITE VÀ CONTENT PROVIDER



**TS. Huỳnh Hữu Nghĩa**

luckerhuynhvn@gmail.com

# Contents:

---



- SQLite
- Content Provider



- ❖ SQLite là một cơ sở dữ liệu SQL dùng để lưu trữ dữ liệu.
- ❖ SQLite hỗ trợ tất cả các đặc trưng cơ sở dữ liệu quan hệ.
- ❖ Để truy cập vào cơ sở dữ liệu, không cần thiết lập bất cứ loại kết nối nào đến cơ sở dữ liệu như JDBC, ODBC, ...
- ❖ Chỉ cần thực hiện các câu lệnh SQL, phần còn lại do Android phụ trách.
- ❖ Gói `android.database.sqlite` chứa các class để quản lý các CSDL.

- ❖ Để tạo một CSDL gọi phương thức `openOrCreateDatabase` với tham số tên CSDL và `MODE`, trả về một thể hiện CSDL SQLite theo cú pháp sau:

```
SQLiteDatabase mydatabase =  
openOrCreateDatabase("database name",  
MODE_PRIVATE, null);
```

❖ Một số phương thức khác có sẵn:

```
openDatabase(String path, SQLiteDatabase.CursorFactory factory, int  
flags, DatabaseErrorHandler errorHandler)
```

Chỉ mở CSDL có sẵn với MODE (flags) tương ứng,  
các MODE phổ biến như: **OPEN\_READWRITE** hay  
**OPEN\_READONLY**

```
openDatabase(File file, SQLiteDatabase.CursorFactory factory, int  
flags)
```

Nó giống phương thức trên nhưng không có chỉ  
định xử lý các lỗi CSDL

❖ Một số phương thức khác có sẵn:

```
openOrCreateDatabase(String path, SQLiteDatabase.CursorFactory  
factory)
```

Mở hoặc tạo CSDL (nếu không có sẵn)

```
openOrCreateDatabase(File file, SQLiteDatabase.CursorFactory  
factory)
```

Mở hoặc tạo CSDL (nếu không có sẵn)

- ❖ Có thể tạo bảng hay chèn dữ liệu vào bảng sử dụng phương thức `execSQL` được định nghĩa trong lớp `SQLiteDatabase` với cú pháp:

```
mydatabase.execSQL("CREATE TABLE IF NOT  
    EXISTS User(id VARCHAR, password  
                VARCHAR) ;");
```

```
mydatabase.execSQL("INSERT INTO User  
    VALUES ('123', 'admin') ;");
```





- ❖ Có thể truy hồi dữ liệu từ CSDL sử dụng đối tượng của lớp Cursor thông qua phương thức `rawQuery()` với cú pháp:

```
Cursor resultSet =  
mydatabase.rawQuery("Select * from  
User", null);  
  
resultSet.moveToFirst();  
  
String name = resultSet.getString(0);  
  
String pw = resultSet.getString(1);
```



❖ Một số hàm:

```
getColumnCount();
```

Trả về tổng số cột trong bảng

```
getColumnIndex(String columnName);
```

Trả về chỉ số cột.

```
getColumnName(Int columnIndex);
```

Trả về tên cột được chỉ định

```
getColumnNames();
```

Trả về mảng tên các cột trong bảng



❖ Một số hàm:

```
getCount() ;
```

Trả về tổng số dòng trong cursor

```
getPosition() ;
```

Trả về vị trí hiện tại của cursor trong  
bảng

```
isClosed() ;
```

Trả về true nếu cursor được đóng, false  
ngược lại

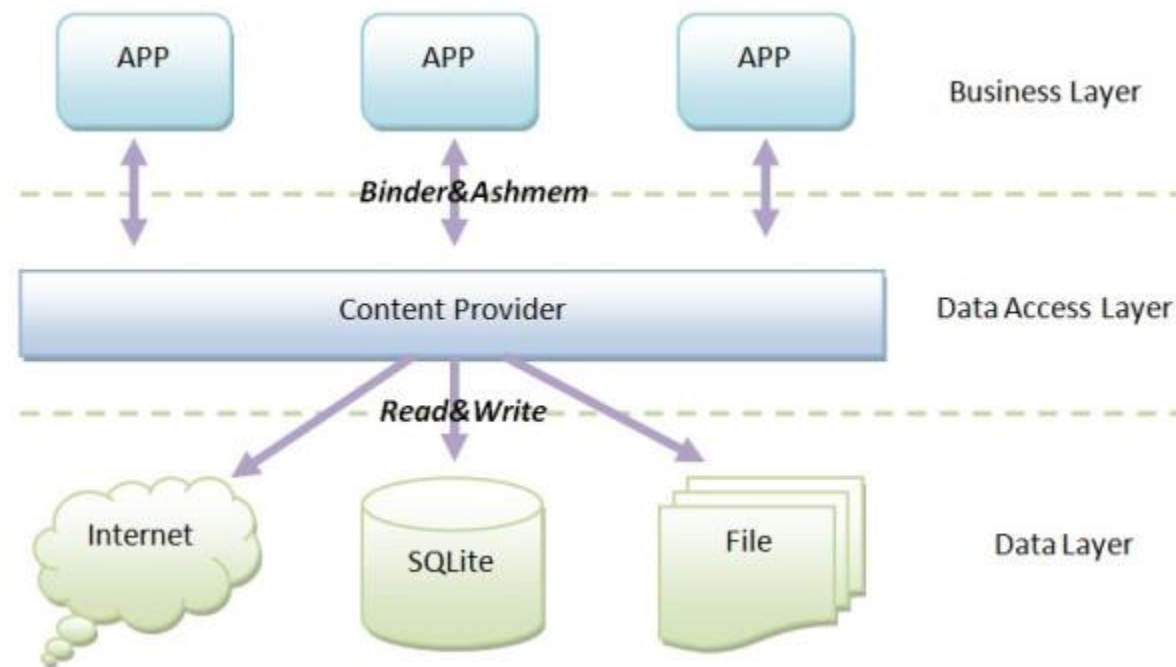
❖ Một số hàm:

❖ Một số hàm:

- ❖ Việc quản lý tất cả các hoạt động liên quan đến CSDL được hỗ trợ bởi lớp SQLiteOpenHelper.
- ❖ Quản lý tự động việc tạo và cập nhật CSDL

```
public class DBHelper extends SQLiteOpenHelper{  
    public DBHelper(){  
        super(context, DATABASE_NAME, null, 1);  
    }  
    public void onCreate(SQLiteDatabase db){}  
    public void onUpgrade(SQLiteDatabase database,  
                           int oldVersion, int newVersion) {}  
}
```

# Content Provider

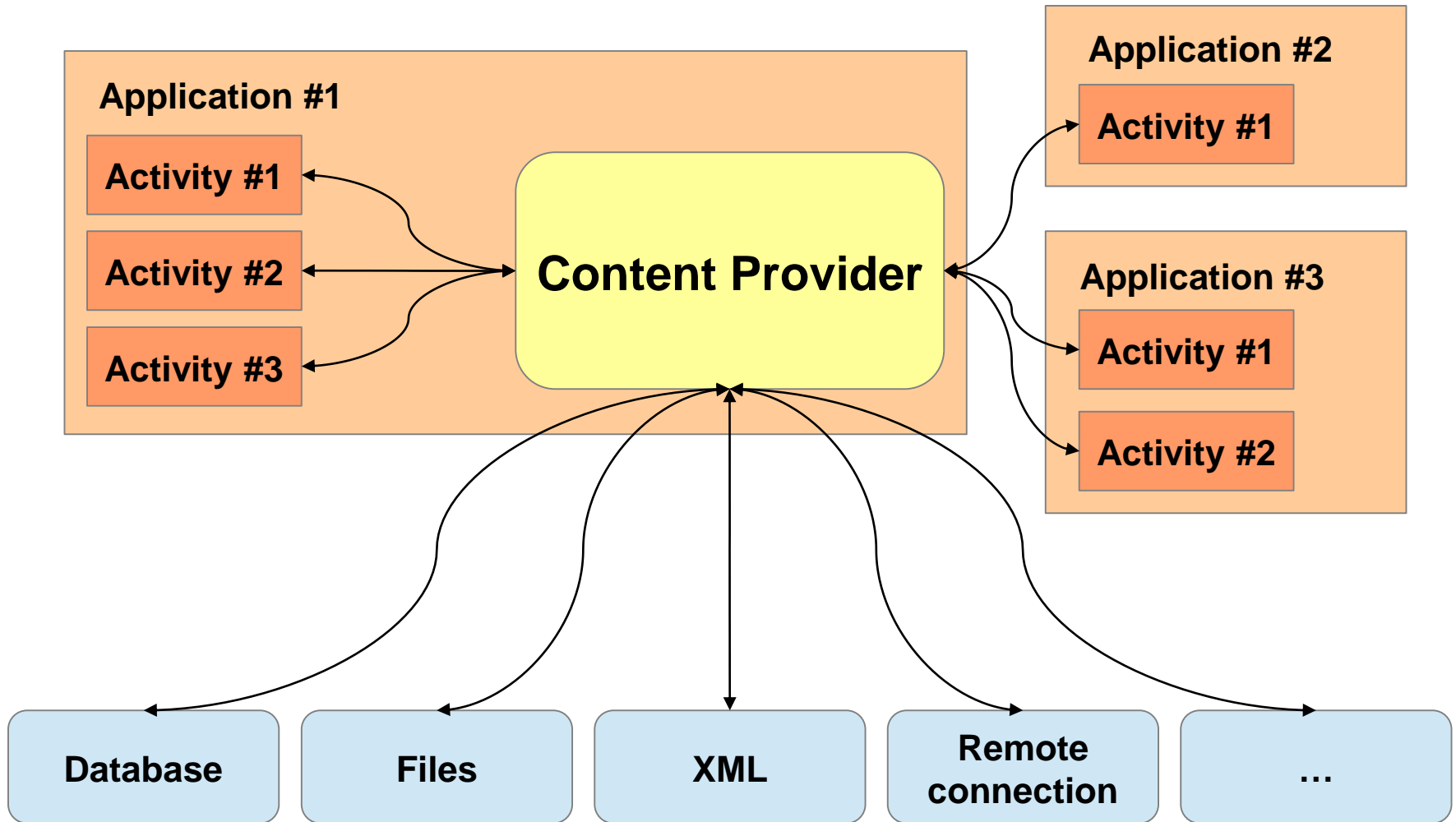


- ❖ Content providers quản lý việc truy cập đến một tập dữ liệu có cấu trúc. Nó đóng gói dữ liệu và cung cấp các cơ chế để xác định bảo mật dữ liệu.
- ❖ Content providers là giao diện chuẩn kết nối dữ liệu trong một tiến trình với mã đang chạy trong tiến trình khác.
- ❖ Được xử lý bởi các phương thức trong lớp ContentResolver



- ❖ Một content provider có thể sử dụng nhiều cách khác nhau để lưu trữ dữ liệu của chính nó và dữ liệu có thể được lưu trữ trong một CSDL, file, hay qua mạng.
- ❖ Mỗi ứng dụng Android chạy trong tiến trình riêng với các quyền riêng giúp ẩn dữ liệu ứng dụng từ ứng dụng khác.
- ❖ Nhưng đôi khi nó được yêu cầu chia sẻ dữ liệu qua các ứng dụng khác. Điều này làm cho content provider trở nên có ích.

- ❖ Content providers cho phép tập trung nội dung ở một nơi và có nhiều ứng dụng khác nhau truy cập khi cần.
- ❖ Một content provider hoạt động giống như một cơ sở dữ liệu khi cần truy vấn, chỉnh sửa nội dung cũng như thêm hoặc xóa nội dung sử dụng các phương thức `insert()`, `update()`, `delete()` và `query()`. Hầu hết các trường hợp này dữ liệu được lưu trữ trên cơ sở dữ liệu SQLite.





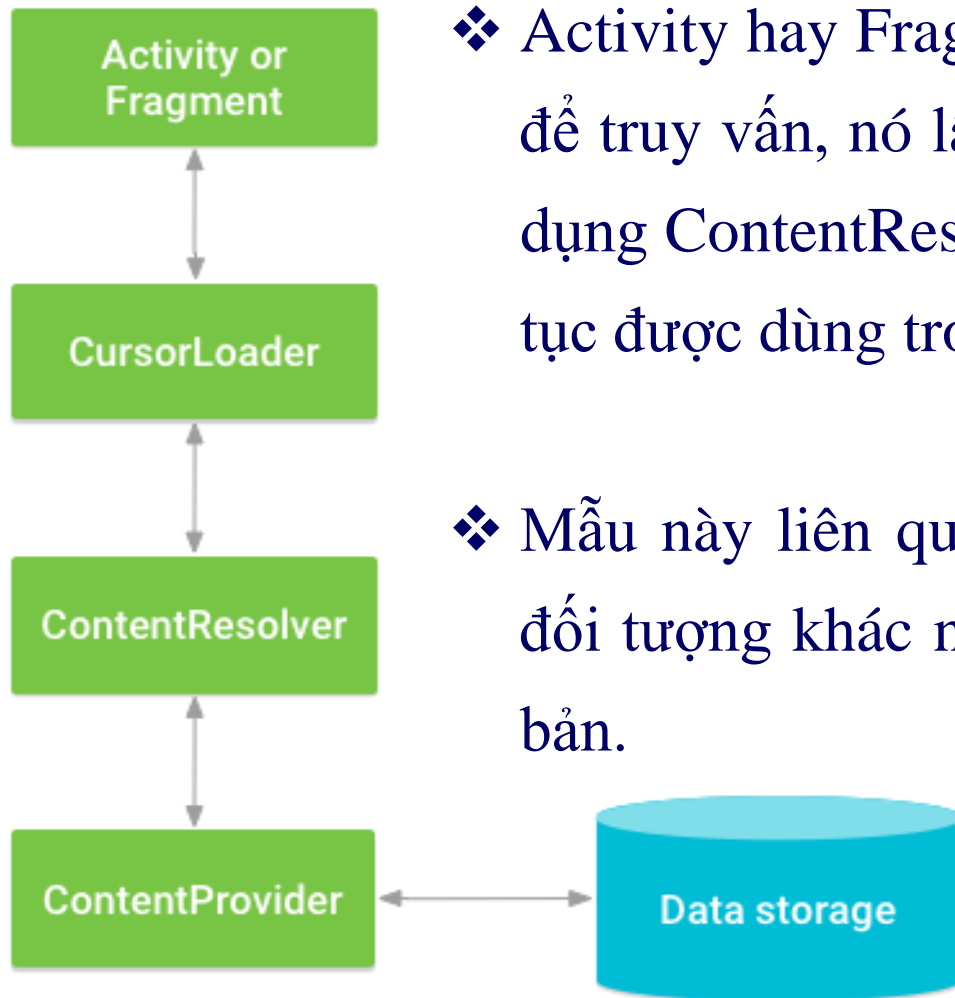
# Truy cập Provider

- ❖ Muốn truy cập dữ liệu từ một content provider, sử dụng đối tượng ContentResolver trong Context của ứng dụng để giao tiếp với provider như một client.
- ❖ Đối tượng ContentResolver giao tiếp với đối tượng provider, một thể của lớp hiện thực ContentProvider.
- ❖ Đối tượng provider nhận dữ liệu yêu cầu từ clients, thực hiện các hoạt động theo yêu cầu, và trả về kết quả.

# Truy cập Provider

- ❖ Đối tượng này có các phương thức gọi các phương thức có tên giống hệt trong đối tượng provider, một thể hiện trong những lớp con của ContentProvider.
- ❖ Các phương thức ContentResolver cung cấp những chức năng thao tác lưu trữ cơ bản “CRUD” (Create, Retrieve, Update, và Delete).
- ❖ Một mẫu phổ biến để truy cập ContentProvider từ UI sử dụng CursorLoader để chạy truy vấn không đồng bộ bên dưới.

# Truy cập Provider



- ❖ Activity hay Fragment trong UI gọi CursorLoader để truy vấn, nó lấy ContentProvider bằng cách sử dụng ContentResolver. Điều này cho phép UI tiếp tục được dùng trong khi truy vấn vẫn chạy.
- ❖ Mẫu này liên quan đến sự tương tác của một số đối tượng khác nhau, cũng như cơ chế lưu trữ cơ bản.



# Truy cập Provider

- ❖ Một trong số các provider được xây trong nền tảng Android là từ điển người dùng. Bảng từ điển người dùng mẫu:

word	app id	frequency	locale	_ID
mapreduce	user1	100	en_US	1
precompiler	user14	200	fr_FR	2
applet	user2	225	fr_CA	3
const	user1	255	pt_BR	4
int	user5	100	en_UK	5



# Truy cập Provider

- ❖ Để lấy words và locales từ User Dictionary Provider, bạn gọi `ContentResolver.query()`. Phương thức `query()` gọi phương thức `ContentProvider.query()` được định nghĩa bởi User Dictionary Provider.

```
// Queries the user dictionary and returns results
cursor = getContentResolver().query(
    UserDictionary.Words.CONTENT_URI,    // The content URI of the words table
    projection,                          // The columns to return for each row
    selectionClause,                      // Selection criteria
    selectionArgs,                       // Selection criteria
    sortOrder);                          // The sort order for the returned rows
```





# Truy cập Provider

- ❖ Các đối số của `query(Uri, projection, selection, selectionArgs, sortOrder)` ứng với SQL Select.

query() argument	SELECT keyword/parameter	Notes
<code>Uri</code>	<code>FROM table_name</code>	<code>Uri</code> maps to the table in the provider named <code>table_name</code> .
<code>projection</code>	<code>col, col, col, ...</code>	<code>projection</code> is an array of columns that should be included for each row retrieved.
<code>selection</code>	<code>WHERE col = value</code>	<code>selection</code> specifies the criteria for selecting rows.
<code>selectionArgs</code>	(No exact equivalent. Selection arguments replace ? placeholders in the selection clause.)	
<code>sortOrder</code>	<code>ORDER BY col, col, ...</code>	<code>sortOrder</code> specifies the order in which rows appear in the returned <b>Cursor</b> .

- ❖ Content URI là một URI xác định dữ liệu trong provider.
- ❖ ContentProvider sử dụng thành phần đường dẫn content URI để chọn bảng truy cập.

# Nhu cầu Content Provider

- ✓ Khi muốn cung cấp dữ liệu hoặc files phức tạp đến các ứng dụng khác.
- ✓ Khi muốn cho phép người dùng copy dữ liệu phức tạp từ ứng dụng của bạn sang những ứng dụng khác.
- ✓ Khi muốn cung cấp những đề xuất tìm kiếm tùy chỉnh sử dụng framework tìm kiếm.
- ✓ Khi muốn hiển thị dữ liệu ứng dụng của bạn đến widgets.
- ✓ Khi muốn hiện thực các lớp `AbstractThreadedSyncAdapter`, `CursorAdapter`, hay `CursorLoader`.



# Tạo một Content Provider

- ❖ **Bước 1:** Thiết kế dữ liệu thô
  - File data (photos, audio, or video)
  - “Structured” data (database, array, or similar structure)
- ❖ **Bước 2:** Triển khai lớp ContentProvider và các phương thức theo yêu cầu.
- ❖ **Bước 3:** Xác định quyền hạn provider, nội dung URI, và các tên cột.
- ❖ **Bước 4:** Một số tùy chọn khác như dữ liệu mẫu hay hiện thực AbstractThreadedSyncAdapter để có thể đồng bộ dữ liệu giữa provider và dữ liệu dựa trên cloud.



# Thiết kế lưu trữ dữ liệu

➤ Demo

