

Lập trình thiết bị di động **SQLite**

ThS. Nguyễn Quang Phúc
phucnq.edu@gmail.com

NỘI DUNG

1. Giới thiệu SQLite
2. Tạo CSDL SQLite
3. Thao tác trên CSDL SQLite
4. Làm việc với dữ liệu hình ảnh

1. Giới thiệu về SQLite

- **SQLite** là cơ sở dữ liệu mở được nhúng trực tiếp vào Android, hỗ trợ các đặc điểm về quan hệ chuẩn của csdl như: cú pháp câu lệnh, transaction, ...
- SQLite có kích thước nhỏ, gọn nhẹ với cấu hình đầy đủ không quá 300kb
- SQLite hỗ trợ các kiểu dữ liệu như: *Text*, *Integer*, *Real*, *Blob*

1. Giới thiệu về SQLite

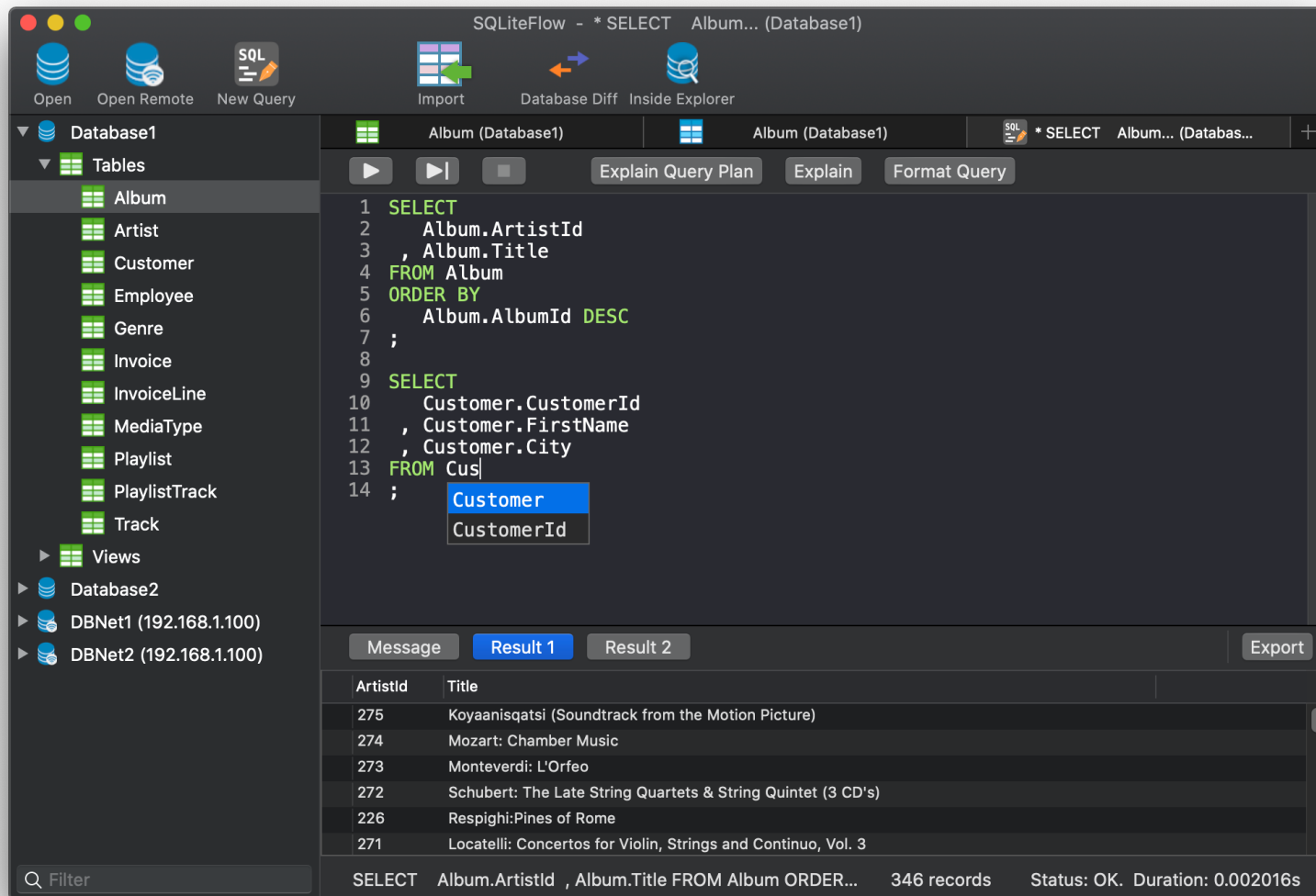
- Đường dẫn của csdl SQLite trên thiết bị:

data/data/APP_NAME/databases/FILE_NAME

➔ *CSDL SQLite được sử dụng cục bộ cho ứng dụng. Trong trường hợp cần cung cấp, chia sẻ dữ liệu với ứng dụng khác nên sử dụng **Content Provider**.*

2. Tạo CSDL SQLite

➤ C1 - Sử dụng Tool: SQLite Studio, SQLiteFlow,



➤ C2 - Coding trực tiếp tạo csdl cho ứng dụng

2. Tạo CSDL SQLite

▼ data	drwxrwx--x	2020-05-18 22:31	4 KB
▶ adb	drwx-----	2020-05-18 22:31	4 KB
▶ anr	drwxrwxr-x	2020-06-06 19:27	4 KB
▶ app	drwxrwx--x	2020-07-16 10:58	4 KB
▶ app-asec	drwx-----	2020-05-18 22:31	4 KB
▼ data	drwxrwx--x	2020-07-16 10:58	8 KB
▶ android	drwx-----	2020-05-18 22:32	4 KB
▶ com.android.backupconfirm	drwx-----	2020-05-18 22:32	4 KB
▼ com.example.sqliteexample1	drwx-----	2020-05-29 13:27	4 KB
▶ cache	drwxrws--x	2020-05-29 13:27	4 KB
▶ code_cache	drwxrws--x	2020-05-29 13:27	4 KB
▼ databases	drwxrwx--x	2020-05-29 13:27	4 KB
note_db.sqlite	-rw-rw----	2020-06-07 07:30	20 KB
note_db.sqlite-journal	-rw-rw----	2020-06-07 07:30	0 B
▼ com.example.sqliteexample2	drwx-----	2020-06-03 23:27	4 KB
▶ cache	drwxrws--x	2020-06-03 23:27	4 KB
▶ code_cache	drwxrws--x	2020-06-03 23:27	4 KB
▼ databases	drwxrwx--x	2020-06-03 23:27	4 KB
songs.db	-rw-rw----	2020-06-07 00:12	16 KB
songs.db-journal	-rw-rw----	2020-06-07 00:12	0 B

2. Tạo CSDL SQLite

➤ Sao chép csdl SQLite vào hệ thống (dùng cho C1)

```
public static final String DATABASE_NAME = "product_db.db";
public static final String DB_PATH_SUFFIX = "/databases/";
public static SQLiteDatabase database = null;

private void copyDataBase(){
    try{
        File dbFile = getDatabasePath(DATABASE_NAME);
        if(!dbFile.exists()){
            if(CopyDBFromAsset()){
                Toast.makeText(MainActivity.this,
                    "Copy database successful!", Toast.LENGTH_LONG).show();
            }else{
                Toast.makeText(MainActivity.this,
                    "Copy database fail!", Toast.LENGTH_LONG).show();
            }
        }
    }
    catch (Exception e){
        Log.e("Error: ", e.toString());
    }
}
```

2. Tạo CSDL SQLite

➤ Sao chép csdl SQLite vào hệ thống (dùng cho C1)

```
private boolean CopyDBFromAsset() {  
    String dbPath = getApplicationInfo().dataDir + DB_PATH_SUFFIX +  
    DATABASE_NAME;  
    try {  
        InputStream inputStream = getAssets().open(DATABASE_NAME);  
        File f = new File(getApplicationInfo().dataDir + DB_PATH_SUFFIX);  
        if(!f.exists()){  
            f.mkdir();  
        }  
        OutputStream outputStream = new FileOutputStream(dbPath);  
        byte[] buffer = new byte[1024]; int length;  
        while((length=inputStream.read(buffer))>0){  
            outputStream.write(buffer,0, length);  
        }  
        outputStream.flush(); outputStream.close(); inputStream.close();  
        return true;  
    } catch (IOException e) {  
        e.printStackTrace();  
        return false;  
    }  
}
```


2. Tạo CSDL SQLite

➤ Coding trực tiếp (C2)

```
public class Databases extends SQLiteOpenHelper {  
  
    public Databases(@Nullable Context context, @Nullable String name,  
                    @Nullable SQLiteDatabase.CursorFactory factory, int version) {  
        super(context, name, factory, version);  
    }  
  
    //Truy vấn không trả kết quả (CREATE, INSERT, UPDATE, DELETE, ...)  
    public void QueryData(String sql){  
        SQLiteDatabase db = getWritableDatabase();  
        db.execSQL(sql);  
    }  
  
    //Truy vấn trả về kết quả (SELECT)  
    public Cursor GetData(String sql){  
        SQLiteDatabase db = getReadableDatabase();  
        return db.rawQuery(sql,null);  
    }  
}
```

2. Tạo CSDL SQLite

➤ Coding trực tiếp (C2)

```
private void PrepareDB() {  
    //Create database  
    databases = new Databases(this, "note_db.sqlite", null, 1);  
    //Create table  
    databases.QueryData(  
        "CREATE TABLE IF NOT EXISTS Works(Id INTEGER PRIMARY  
        KEY AUTOINCREMENT, WorkName VARCHAR(200))"  
    );  
    //Insert data  
    databases.QueryData("INSERT INTO Works VALUES(null, 'Fix bugs')");  
    databases.QueryData("INSERT INTO Works VALUES(null, 'Coding')");  
    .....  
}
```

3. Thao tác trên CSDL SQLite

➤ Mở cơ sở dữ liệu

```
public static final String DATABASE_NAME = "songs.db";  
public static final String TABLE_NAME = "song";  
  
public static SQLiteDatabase database = null;  
database = openOrCreateDatabase(DATABASE_NAME,  
                                MODE_PRIVATE, null);
```

3. Thao tác trên CSDL SQLite

➤ Truy vấn csdl với “rawQuery”

```
Cursor cursor = database.rawQuery("SELECT * FROM " + TABLE_NAME,  
                                     selectionArgs: null);  
  
while(cursor.moveToNext()) {  
    int id = cursor.getInt(0);  
    int songCode = cursor.getInt(1);  
    String songName = cursor.getString(2);  
    String singer = cursor.getString(3);  
    int favourite = cursor.getInt(4);  
    //To do something ....  
}  
cursor.close();
```

3. Thao tác trên CSDL SQLite

➤ Truy vấn csdl với “rawQuery”

```
Cursor cursor = database.rawQuery(  
    "SELECT * FROM " + TABLE_NAME + " WHERE Favourite = ? AND  
    (SongCode LIKE ? OR SongName LIKE ? OR Singer LIKE ?)",  
    new String[]{"1", "%" + s + "%", "%" + s + "%", "%" + s + "%"});  
  
while(cursor.moveToNext()) {  
    int id = cursor.getInt(0);  
    int songCode = cursor.getInt(1);  
    String songName = cursor.getString(2);  
    //To do something ....  
}  
cursor.close();
```

3. Thao tác trên CSDL SQLite

➤ Truy vấn csdl với “query”

```
Cursor cursor = database.query("Product", columns: null,  
                                selection: null, selectionArgs: null,  
                                groupBy: null, having: null, orderBy: null);
```

```
Cursor cursor = database.rawQuery(  
    "SELECT * FROM Product WHERE ProductId = ? OR ProductId = ?",  
    new String[] {"2", "3"});
```



```
Cursor cursor = database.query("Product", columns: null,  
                                selection: "ProductId = ? OR ProductId = ?",  
                                selectionArgs: new String[] {"2", "3"},  
                                groupBy: null, having: null, orderBy: null);
```

3. Thao tác trên CSDL SQLite

➤ Thêm dữ liệu

```
ContentValues values = new ContentValues();
values.put("ProductName", pName);
values.put("ProductPrice", pPrice);
long flag = database.insert("Product", nullColumnHack: null, values);
if(flag > 0)
    Toast.makeText(AddProductActivity.this, "Add product successful!",
        Toast.LENGTH_LONG).show();
else
    Toast.makeText(AddProductActivity.this, "Add product fail!",
        Toast.LENGTH_LONG).show();
```

3. Thao tác trên CSDL SQLite

➤ Thêm dữ liệu

```
//Truy vấn không trả kết quả (CREATE, INSERT, UPDATE, DELETE, ...)  
public void QueryData(String sql){  
    SQLiteDatabase db = getWritableDatabase();  
    db.execSQL(sql);  
}  
  
//Insert data  
databases.QueryData("INSERT INTO Product VALUES(  
    null, 'Laptop Dell XPS', 26500000)");
```


3. Thao tác trên CSDL SQLite

➤ Xóa dữ liệu

```
int flag = database.delete("Product", "ProductId=?", new String[]{"1"});  
if(flag > 0)  
    //Xóa thành công  
else  
    //Xóa thất bại
```

```
//Truy vấn không trả kết quả (CREATE, INSERT, UPDATE, DELETE, ...)  
public void QueryData(String sql){  
    SQLiteDatabase db = getWritableDatabase();  
    db.execSQL(sql);  
}  
  
//Delete data  
databases.QueryData("DELETE FROM Product WHERE ProductId = 1");
```

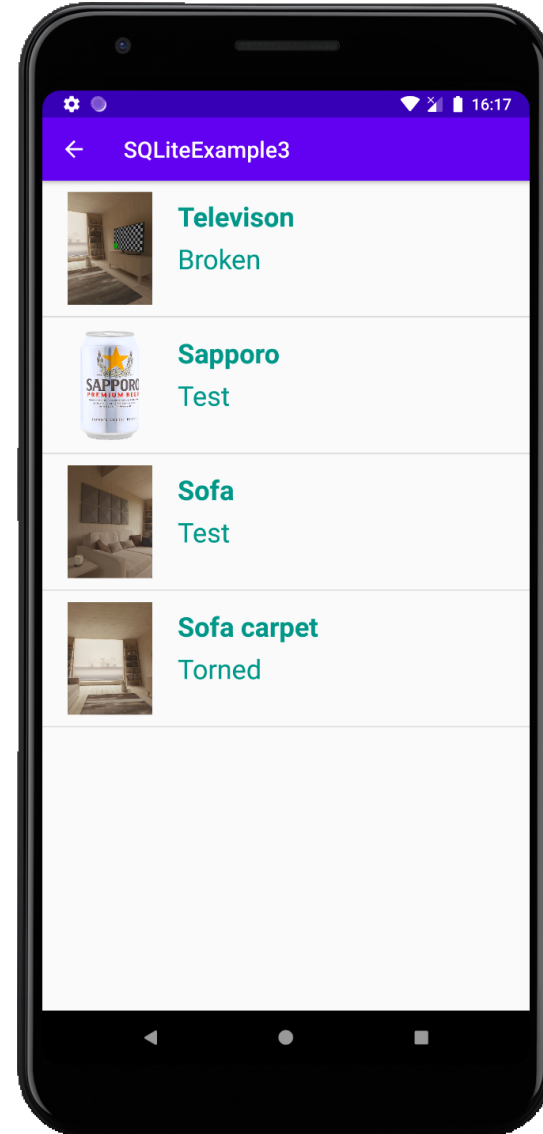
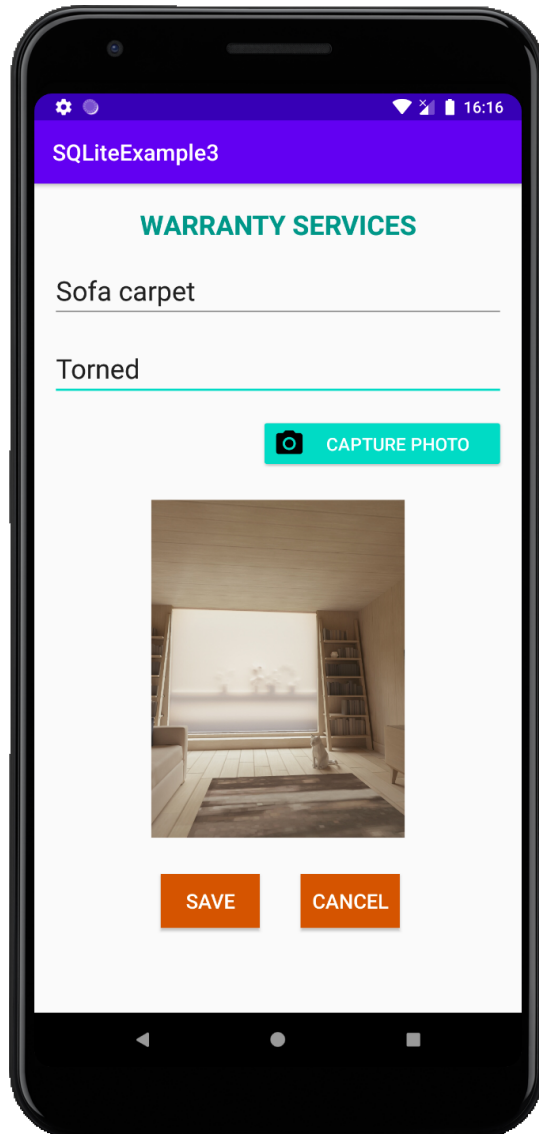
3. Thao tác trên CSDL SQLite

➤ Sửa dữ liệu

```
ContentValues values = new ContentValues();
values.put("WorkName", "Fix bugs");
int flag = database.update("Works", values, "Id=?", new String[]{"1"});
if(flag > 0)
    //Cập nhật thành công
else
    //Cập nhật thất bại
```

```
//Truy vấn không trả kết quả (CREATE, INSERT, UPDATE, DELETE, ...)
public void QueryData(String sql){
    SQLiteDatabase db = getWritableDatabase();
    db.execSQL(sql);
}
//Update data
databases.QueryData("UPDATE Works SET WorkName = 'Fix bugs'
WHERE Id = 1");
```

4. Làm việc với dữ liệu hình ảnh



4. Làm việc với dữ liệu hình ảnh

```
public void insertData(String name, String des, byte[] photo){  
    SQLiteDatabase database = getWritableDatabase();  
    String sql = "INSERT INTO Product VALUES(null, ?, ?, ?)";  
  
    SQLiteStatement statement = database.compileStatement(sql);  
    statement.clearBindings();  
    statement.bindString(1, name);  
    statement.bindString(2, des);  
    statement.bindBlob(3, photo);  
  
    statement.executeInsert();  
}
```

4. Làm việc với dữ liệu hình ảnh

```
private byte[] convertPhoto() {  
    BitmapDrawable drawable = (BitmapDrawable) imgPhoto.getDrawable();  
    Bitmap bitmap = drawable.getBitmap();  
    ByteArrayOutputStream outputStream = new ByteArrayOutputStream();  
    bitmap.compress(Bitmap.CompressFormat.PNG, 100, outputStream);  
    return outputStream.toByteArray();  
}
```

```
database.insertData("Sofa carpet", "Torned", convertPhoto());
```

Q & A

Lập trình thiết bị di động **Content Provider**

ThS. Nguyễn Quang Phúc
phucnq.edu@gmail.com

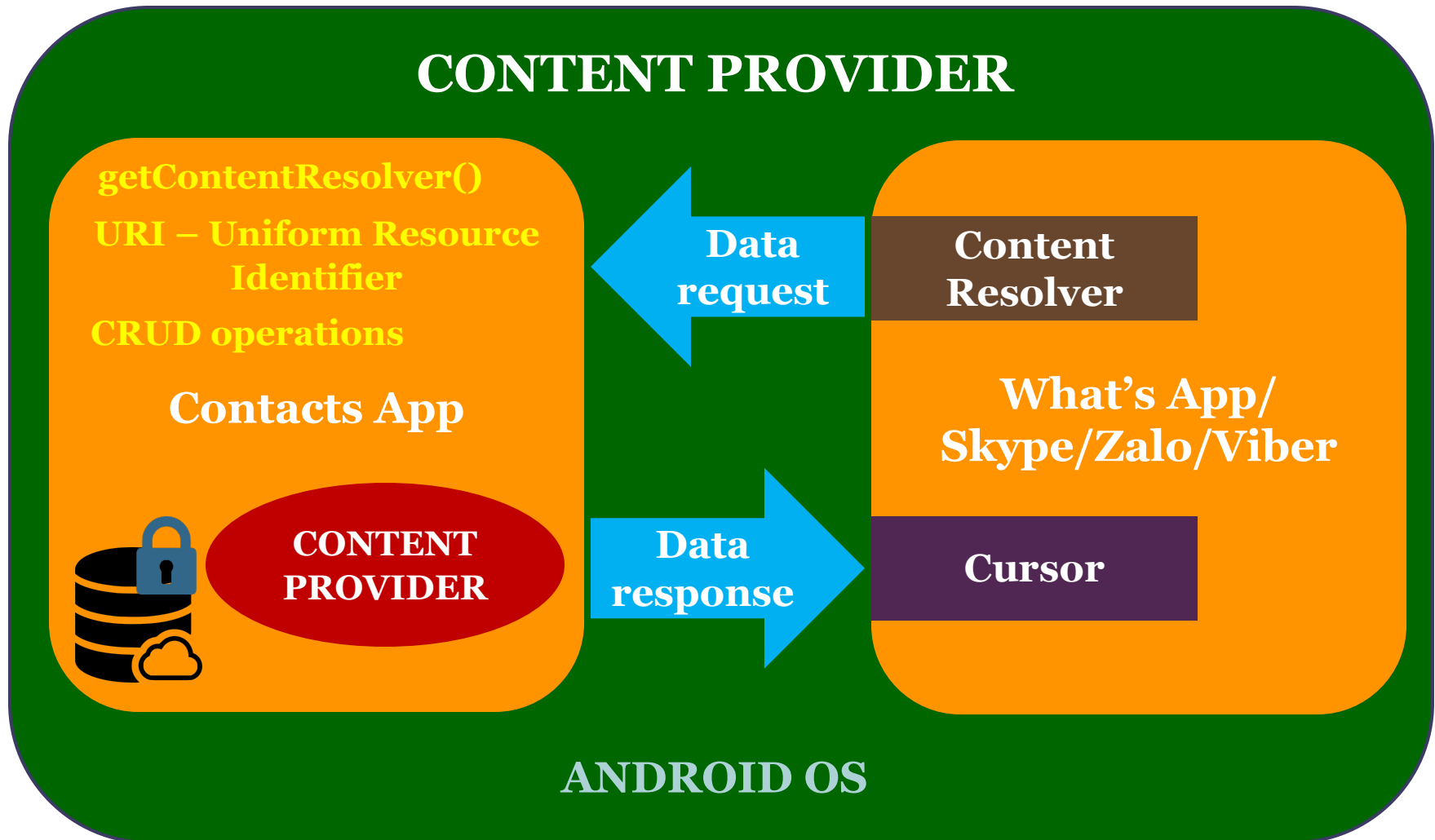
NỘI DUNG

1. Giới thiệu về ContentProvider trong Android
2. Sử dụng ContentProvider
 - *Đọc và lưu danh bạ*
 - *Truy xuất dữ liệu nhật ký cuộc gọi*
 - *Truy xuất dữ liệu sms*
 - *Truy xuất dữ liệu Media*

1. Giới thiệu về Content Provider

- **ContentProvider** là một *thành phần quản lý truy cập dữ liệu*, cung cấp các phương thức khác nhau để các ứng dụng có thể truy cập dữ liệu từ những ứng dụng khác thông qua đối tượng *ContentResolver*.
- Một ứng dụng có thể quản lý quyền truy cập đến dữ liệu được lưu bởi ứng dụng đó thông qua ContentProvider.
- Dữ liệu được quản lý bởi ContentProvider giống như csdl quan hệ → các thao tác *truy vấn, thêm mới, cập nhật, xóa* được thực hiện qua các phương thức: *query(), insert(), update(), delete()*.

1. Giới thiệu về Content Provider



2. Sử dụng ContentProvider

»» Vd: Đọc dữ liệu danh bạ điện thoại

Request
permission

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

```
Uri uri =  
    ContactsContract.CommonDataKinds.Phone.CONTENT_URI;  
Cursor cursor = getContentResolver().query(uri, projection: null,  
    selection: null, selectionArgs: null, sortOrder: null);
```

```
while (cursor.moveToNext()){  
    int nameIndex =  
        cursor.getColumnIndex(ContactsContract.  
                                Contacts.DISPLAY_NAME);  
    String name = cursor.getString(nameIndex); //Get Name  
    int phoneIndex =  
        cursor.getColumnIndex(ContactsContract.CommonDataKinds.  
                                Phone.NUMBER);  
    String phone = cursor.getString(phoneIndex); //Get Phone Number  
    //Todo something ...  
}  
cursor.close();
```

2. Sử dụng ContentProvider

»» Vd: Lưu danh bạ điện thoại

```
Uri addContactsUri = ContactsContract.Data.CONTENT_URI;  
//Add an empty contact and get the generated Id  
long rowContactId = getRawContactId();  
//Add contact name data  
insertContactDisplayName(addContactsUri, rowContactId, "Mr Bean");  
//Add phone number  
insertContactPhoneNumber(addContactsUri, rowContactId,  
                        "0903778899");
```

```
private long getRawContactId() { //Insert an empty contact  
    ContentValues values = new ContentValues();  
    Uri rawContactUri = getContentResolver().  
        insert(ContactsContract.RawContacts.CONTENT_URI, values);  
    return ContentUris.parseId(rawContactUri);  
}
```

2. Sử dụng ContentProvider

»» Vd: Lưu danh bạ điện thoại

```
private void insertContactDisplayName(Uri addContactsUri, long
rawContactId, String displayName)
{
    ContentValues values = new ContentValues();
    values.put(ContactsContract.Data.RAW_CONTACT_ID, rawContactId);
    values.put(ContactsContract.Data.MIMETYPE,
                ContactsContract.CommonDataKinds.
                StructuredName.CONTENT_ITEM_TYPE);
    values.put(ContactsContract.CommonDataKinds.
                StructuredName.GIVEN_NAME, displayName);
    getContentResolver().insert(addContactsUri, values);
}
```

2. Sử dụng ContentProvider

»» Vd: Lưu danh bạ điện thoại

```
private void insertContactPhoneNumber(Uri addContactsUri, long
rawContactId, String phoneNumber)
{
    ContentValues values = new ContentValues();
    values.put(ContactsContract.Data.RAW_CONTACT_ID, rawContactId);
    values.put(ContactsContract.Data.MIMETYPE,
                ContactsContract.CommonDataKinds.
                    Phone.CONTENT_ITEM_TYPE);
    values.put(ContactsContract.CommonDataKinds.Phone.NUMBER,
                phoneNumber);
    getContentResolver().insert(addContactsUri, values);
}
```

2. Sử dụng ContentProvider

»» Vd: Đọc dữ liệu nhật ký cuộc gọi

Request
permission

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

```
String[] projection = new String[]{CallLog.Calls.DATE,  
                                     CallLog.Calls.NUMBER,  
                                     CallLog.Calls.DURATION};  
  
Cursor cursor = getContentResolver().query(  
    CallLog.Calls.CONTENT_URI,  
    projection: projection,  
    selection: null, selectionArgs: null,  
    sortOrder: null);
```

2. Sử dụng ContentProvider

»» Vd: Đọc dữ liệu nhật ký cuộc gọi

```
SimpleDateFormat dateFormat =  
    new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");  
while (cursor.moveToNext()){  
    Date date = new Date(cursor.getLong(0));  
    String sDate = dateFormat.format(date);  
    String number = cursor.getString(1);  
    long duration = cursor.getLong(2);  
    //Todo something ...  
}  
cursor.close();
```


2. Sử dụng ContentProvider

»» Vd: Đọc dữ liệu tin nhắn sms

Request
permission

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

```
Uri uri = Uri.parse("content://sms/inbox");  
Cursor cursor = getContentResolver().query(uri, projection: null,  
    selection: null, selectionArgs: null, sortOrder: null);
```

```
int phoneColumnIdx = cursor.getColumnIndex("address");  
int timeColumnIdx = cursor.getColumnIndex("date");  
int bodyColumnIdx = cursor.getColumnIndex("body");  
while (cursor.moveToNext()){  
    String phoneNumber = cursor.getString(phoneColumnIdx);  
    Date date = new Date(cursor.getLong(timeColumnIdx));  
    String content = cursor.getString(bodyColumnIdx);  
    //Todo something ...  
}  
cursor.close();
```

2. Sử dụng ContentProvider

»» Vd: Đọc dữ liệu Media

Request
permission

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

```
String[] projection = {  
    MediaStore.MediaColumns.DISPLAY_NAME,  
    MediaStore.MediaColumns.DATE_MODIFIED,  
    MediaStore.MediaColumns.MIME_TYPE  
};  
CursorLoader cursorLoader = new CursorLoader(MainActivity.this,  
    MediaStore.Images.Media.EXTERNAL_CONTENT_URI,  
    projection, null, null, null);  
Cursor cursor = cursorLoader.loadInBackground();
```

2. Sử dụng ContentProvider

»» Vd: Đọc dữ liệu Media

```
while (cursor.moveToNext()){  
    String name = cursor.getString(0);  
    Date date = new Date(cursor.getLong(1));  
    String type = cursor.getString(2);  
    //Todo something ...  
}  
cursor.close();
```

Q & A