



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

Lập trình Android

Bài 15: *SQLite*

Phòng LT & Mạng

<http://csc.edu.vn/lap-trinh-va-csdl>





Nội dung

1. SQLite
2. Tạo CSDL
3. Tạo bảng
4. Các thao tác select, insert, delete, update



SQLite

- **SQLite** là một hệ quản trị CSDL (database management system) độc lập được bao gồm trong HĐH Android
- Có thể sử dụng **SQLite** để:
 - Tạo mới CSDL (database)
 - Định nghĩa các bảng (table)
 - Truy vấn nội dung các bảng (select)
 - Thực hiện các thao tác thêm (insert), xóa (delete), cập nhật (update) trên dòng
- **SQLite** sử dụng chuẩn SQL-92
- Minh họa sau tạo và sử dụng một CSDL danh bạ



Tạo CSDL

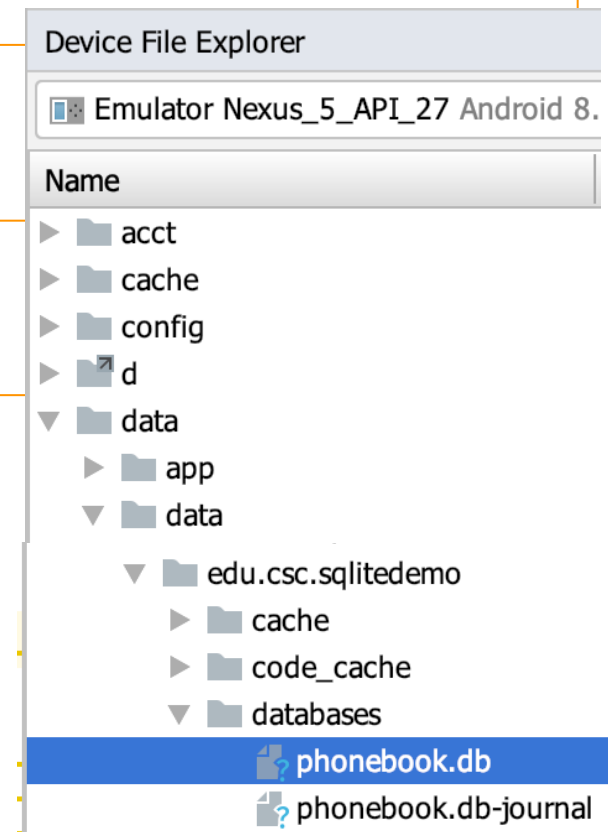
- Mở internal database (hoặc tạo mới và mở nếu chưa tồn tại)

```
private String dbName = "phonebook.db";

private SQLiteDatabase openDB() {
    return openOrCreateDatabase(dbName, MODE_PRIVATE, null);
}
```

- Đóng database đã mở:

```
private void closeDB(SQLiteDatabase db) {
    db.close();
}
```





Tạo CSDL (tiếp)

- Mở/tạo external database:

```
private SQLiteDatabase openExternalDB() {  
    String dbName = "phonebook.edb";  
    String path = Environment.getExternalStorageDirectory().getPath() + "/" + dbName;  
    return SQLiteDatabase.openDatabase(path, null,  
        SQLiteDatabase.CREATE_IF_NECESSARY);  
}
```

- Ngoài biến flag: CREATE_IF_NECESSARY (tạo csdl nếu chưa tồn tại) thì còn các cờ khác: OPEN_READWRITE (mở để đọc và ghi), OPEN_READONLY (mở chỉ đọc)
- Trước khi mở/tạo **external** database, cần phải khai báo **permissions** tương ứng trong **AndroidManifest.xml** và **request runtime permissions** đối với người dùng

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

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



Tạo bảng

- Tạo bảng mới **tblPhonebook**:

```
private void createContactTable() {  
    SQLiteDatabase db = openDB();  
    String sql = "create table if not exists tblPhonebook(id integer PRIMARY KEY  
autoincrement, name text, phone text);";  
    db.execSQL(sql);  
    closeDB(db);  
}
```

tblPhonebook			
<u>id</u>	integer	PRIMARY KEY	auto increment
name	text		
phone	text		



Thêm một dòng mới vào bảng

- Thêm một contact vào bảng **tblPhonebook**:

```
private void insert(Contact c) {  
    SQLiteDatabase db = openDB();  
    ContentValues cv = new ContentValues();  
    cv.put("name", c.name);  
    cv.put("phone", c.phone);  
    db.insert("tblPhonebook", null, cv);  
    closeDB(db);  
}
```



Đọc toàn bộ bảng

- Đọc toàn bộ bảng **tblPhonebook**:

```
private ArrayList<Contact> getContacts() {  
    SQLiteDatabase db = openDB();  
    ArrayList<Contact> arr = new ArrayList<>();  
    String sql = "select * from tblPhonebook";  
    Cursor csr = db.rawQuery(sql, null);  
    if (csr != null) {  
        if (csr.moveToFirst()) {  
            do {  
                String name = csr.getString(1);  
                String phone = csr.getString(2);  
                arr.add(new Contact(name, phone));  
            } while (csr.moveToNext());  
        }  
    }  
    closeDB(db);  
    return arr;  
}
```




Cập nhật một dòng trong bảng

- Cập nhật một dòng trong bảng **tblPhonebook**:

```
private void updateContact(Contact c) {  
    SQLiteDatabase db = openDB();  
    ContentValues cv = new ContentValues();  
    cv.put("name", c.name);  
    cv.put("phone", c.phone);  
    String[] id = {String.valueOf(c.id)};  
    int row = db.update("tblPhonebook", cv, "id = ?", id);  
    closeDB(db);  
}
```



Xoá một dòng trên bảng

- Xoá một dòng của bảng **tblPhonebook**:

```
private void deleteContact(Contact c) {  
    String[] id = {String.valueOf(c.id)};  
    SQLiteDatabase db = openDB();  
    db.delete("tblPhonebook", "id = ?", id);  
    closeDB(db);  
}
```



Đọc 1 dòng

- Đọc một dòng từ bảng tblPhonebook:

```
private Contact getContact(int cld) {  
    String[] fields = {"id", "name", "phone"};  
    String[] ids = {String.valueOf(cld)};  
    SQLiteDatabase db = openDB();  
    Cursor cursor = db.query("tblPhonebook", fields, "id = ?", ids, null, null, null, null);  
    if (cursor != null)  
        cursor.moveToFirst();  
    int id = cursor.getInt(0);  
    String name = cursor.getString(1);  
    String phone = cursor.getString(2);  
    closeDB(db);  
    return new Contact(id, name, phone);  
}
```

