

안드로이드 데이터베이스 시스템 연구

김민수*, 임재걸**, 박병훈^o

**동국대학교 컴퓨터공학부

^o*동국대학교 차세대디지털미디어센터

e-mail: {minsukim, yim, ppark}@dongguk.ac.kr

Review of Android Database Systems

Minsu Kim*, Jaegeol Yim**, Byunghun Park^o

**Dept. of Computer Engineering, Dongguk University

^o*NDMC Dongguk University

● 요약 ●

단순한 데이터는 파일 형태로 관리할 수 있지만 조직화된 대량의 데이터를 효율적으로 다루기 위해서는 데이터베이스가 필요하다. 안드로이드 운영체제는 SQLite 라이브러리를 포함하고 있어 별도의 설정이 없더라도 데이터베이스를 이용할 수 있다. 본 연구는 SQLite helper 메소드들의 사례를 소개한다.

키워드: 데이터베이스(database), 안드로이드(Android), SQLite(SQLite)

I. 서론

안드로이드 내장 데이터베이스인 SQLite를 사용하면 별도의 서버가 필요 없으므로, 연결과 권한 또한 신경 쓸 필요가 없다. 복수 사용자는 지원하지 않지만 용량이 작고 C언어로 작성이 되어있어 속도 또한 빠르다.

방송 콘텐츠 등록 시스템 [2]의 일부를 SQLite에 저장하고 있으면 데이터베이스 접근 시간을 크게 절감할 수 있다. 본 논문은 안드로이드 내장 데이터베이스인 SQLite를 사용하여 데이터베이스를 작성하는 방법에 대하여 알아본다.

```

public class DataManagerDB extends SQLiteOpenHelper {
    SQLiteDatabase db;
    Cursor cursor;
    public DataManagerDB(Context context){
        super(context, "DataManagerDB.db", null, 1);
        db = this.getWritableDatabase();
        db = this.getReadableDatabase();
    }
}

```

그림 1 SQLiteOpenHelper 상속받기
Fig. 1. Extending SQLiteOpenHelper

II. SQLite Helper 메소드

1. 관련SQLiteOpenHelper 클래스 상속

간단한 기능을 구현한다면 SQLiteOpenHelper [1] 클래스를 그대로 사용하여도 되지만 현재 앱에 맞는 기능을 구현하려면, 그림 1에 보이는 바와 같이 SQLiteOpenHelper를 상속받아 자신의 클래스(DataManagerDB)를 생성한다. SQLiteDatabase와 Cursor 객체는 해당 클래스가 아닌 외부에서 사용해도 되지만 본 고에서는 상속받은 클래스에서 데이터베이스에 접근하는 과정을 마치도록 하기 위해 DataManagerDB 클래스에 포함하였다.

생성자 함수를 보면 SQLiteOpenHelper 클래스의 생성자 함수를 호출하는 것을 볼 수 있다. 이때, 주어지는 인자로는 Context, Name, SQLiteDatabase.CursorFactory, Version이 있다. Context는 데이터베이스를 열기위해 사용되며, Name은 데이터베이스 파일을 나타낸다. SQLiteDatabase.CursorFactory는 Cursor 객체를 만드는데 사용되지만 본 고에서는 null로 하여 기본 설정을 사용한다. Version은 데이터베이스의 버전 정보를 나타낸다. 데이터베이스 읽기와 쓰기 설정을 해주기 위해 getWritableDatabase 메소드와 getReadableDatabase 메소드의 결과값을 SQLiteDatabase 객체에 저장한다.

2. 초기화 메소드 구현

SQLiteOpenHelper 클래스를 상속한 DataManagerDB 클래스에 초기화 메소드도 정의해야한다. 그림 2에는 LiveChannel이라는 테이블을 생성하는 초기화 메소드의 예가 보인다.

```
public class DataManagerDB extends SQLiteOpenHelper {
    ...
    @Override
    public void onCreate(SQLiteDatabase db) {
        String sql;
        sql = "CREATE TABLE LiveChannel " +
            ...
            ",channel_webservice_uri TEXT);";
        db.execSQL(sql);
        ...
    }
}
```

그림 2 초기화 함수 정의
Fig. 2. Definition of Initialization method

3. 값 저장 메소드 구현

데이터베이스에 값을 저장하기 위해 그림 3과 같은 메소드를 구현하였다. 메소드의 인자로 테이블에 넣을 값을 입력받아 SQL 구문을 작성하여 해당하는 테이블에 저장하는 형식으로 구현하였다.

```
public void insertLiveChannel(String
    channel_name, ... String channel_webservice_uri) {
    String sql;
    sql = "insert into LiveChannel" +
        "(channel_name" + ...
    ...
}
```

그림 3 테이블에 튜플 삽입 메소드 정의
Fig. 3. A Method of inserting a tuple

4. 테이블 내용 읽기 메소드 구현

테이블에 저장된 내용을 모두 읽어 내는 메소드도 그림 5와 같이 구현한다..rawQuery 메소드의 실행 결과로 Cursor 객체를 반환하

는데 이 객체를 DataManagerDB 클래스의 Cursor 객체에 저장한다. Cursor 클래스의 getCount 메소드는 SQL 구문에 대한 반환 값의 개수를 반환한다. 다음으로 moveToFirst 메소드는 Cursor 객체에서 가장 처음으로 가도록 명시적으로 선언한다.

반복문에서는 튜플 하나를 저장할 Cha() 객체를 생성하여 cursor의 i 번째 튜플로 채우고 channels 리스트에 추가하기를 반복한다.

```
public List<Cha> getLiveChannel( ){
    List<Cha> channels = new ArrayList<Cha>( );
    cursor = ...
    cursor.moveToFirst();
    for(int i=0; i<count ; i++){
        Cha item = new Cha();
        ...
        cursor.moveToNext();
    }
    return channels;
}
```

그림 4 테이블 읽기 메소드
Fig. 4. A Method of reading a table

사사

본 논문은 한국과학재단의 기초과학연구 프로그램(NRF-2011-0006942)과 지정부의 "IPTV 기반 글로벌 문화관광 방송 기반 구축" (10037393)의 지원을 받았습니다.

참고문헌

- [1] <http://developer.android.com/reference/android/>
- [2] Jaegel Yim and Gyeyoung Lee, "The Implementation of a Web System for the Remote Management of IPTV Contents," International Journal of Multimedia and Ubiquitous Engineering Vol.8, No.5 (2013), pp.201-212