



군산대학교
KUNSAN NATIONAL UNIVERSITY

ContentProvider

모바일 응용

남 광 우

ContentProvider 개념

❖ ContentProvider란?

- ✓ Content Provider는 애플리케이션이 보유하고 있는 데이터를 다른 애플리케이션에 공개하기 위한 모듈
 - 애플리케이션 내부 데이터를 외부에 제공-공유하기 위해서 Content Provider는 반드시 구현해야 하는 중요한 작업

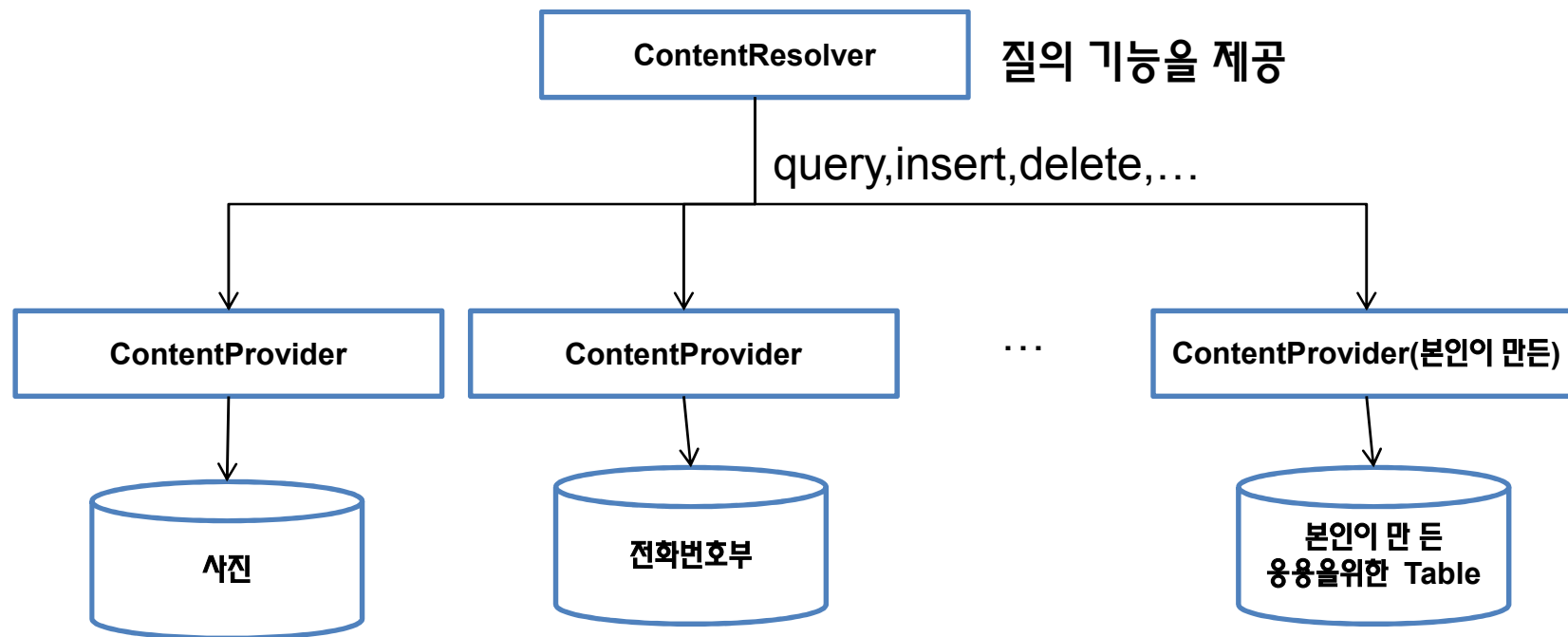
- ✓ 안드로이드 기본제공 ContentProvider
 - 오디오, 비디오, 사진, 전화번호부, 기타 등등 에 대한 몇 개의 콘텐츠 프로바이더 기본 제공

- ✓ 자신의 응용을 위한 ContentProvider
 - 직접 ContentProvider 를 구현하거나,
 - 이미 존재하는 ContentProvider에 추가

ContentProvider 개념

❖ ContentProvider와 ContentResolver

- ✓ ContentResolver의 획득
 - ContentResolver cr = getContentResolver();



ContentProvider 개념

❖ ContentProvider 의 데이터 모델

- ✓ 콘텐츠 프로바이더는 데이터베이스 모델 상의 간단한 테이블처럼 그것의 데이터를 보여줌
 - 각 행은 레코드이며, 각 열은 특정 타입과 의미의 데이터

전화번호에 관한 정보

_ID	NUMBER	NUMBER_KEY	LABEL	NAME	TYPE
13	(425)555 6677	425 555 6677	Kirkland office	Bully Pulpit	TYPE_WORK
44	(212)555-1234	212 555 1234	NY apartment	Alan Vain	TYPE_HOME
45	(212)555-6657	212 555 6657	Downtown office	Alan Vain	TYPE_MOBILE
53	201.555.4433	201 555 4433	Love Nest	Rex Cars	TYPE_HOME

URI로 구분

content://android.provider.Contacts.Phones.CONTENT_URI

content://android.provider.Contacts.Photos.CONTENT_URI

ContentProvider 개념

❖ Provider 클래스의 주요 URI

✓ droid.provider.*

데이터	URI
통화로그	CallLog.Calls.CONTENT_URI
Contract List	Contacts.People.CONTENT_URI
브라우저 내의 북마크	Browser.BOOKMARKS_URI
브라우저 내의 검색 이력	Brouser.SEARCHES_URI
시스템의 설정치	Setting.System.CONTENT_URI
내장 미디어 내의 이미지	MediaStore.Image.Media.INTERNAL_CONTENT_URI
주요한 외부미디어 내의 이미지	MediaStore.Image.Media.EXTERNAL_CONTENT_URI
내장 미디어 내의 영상	MediaStore.Video.Media.INTERNAL_CONTENT_URI
주요한 외부미디어 내의 영상	MediaStore.Video.Media.EXTERNAL_CONTENT_URI
내장 미디어 내의 오디오	MediaStore.Audio.Media.INTERNAL_CONTENT_URI
주요한 외부 미디어 내의 오디오	MediaStore.Audio.Media.EXTERNAL_CONTENT_URI

ContentProvider 사용하기

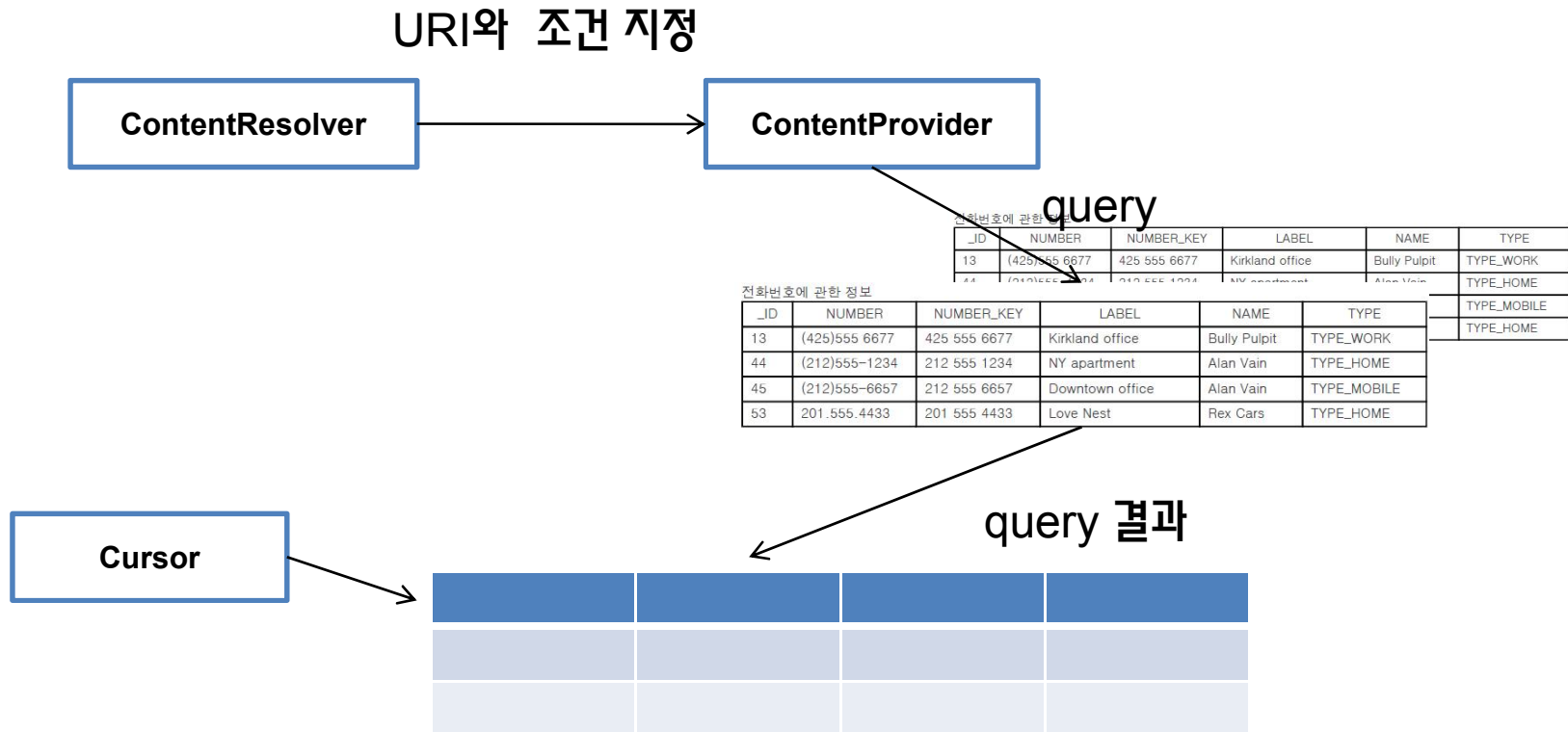
❖ ContentProvider에 Query 하기

- ✓ 콘텐츠 프로바이더에게 쿼리하기 위해서는 세 가지 정보가 필요
 - 프로바이더를 식별하는 URI
 - 받고자 하는 데이터 필드들의 이름들
 - 그 필드들의 데이터 타입들

- ✓ Query하기
 - ContentResolver.query() 메소드
 - 또는 Activity.managedQuery() 메소드를 사용

ContentProvider 사용하기

❖ ContentProvider에 Query Flow 보기



ContentProvider 사용하기

❖ ContentProvider에 Query 하기

✓ URI 만들기

```
Uri myPerson = ContentUris.withAppendedId(People.CONTENT_URI, 23);  
Uri myPerson = Uri.withAppendedPath(People.CONTENT_URI, "23");
```

✓ ContentResolver를 이용하는 방법

```
ContentResolver cr = getContentResolver();  
Cursor allRows = cr.query(myPerson, null, null, null);
```

✓ managedQuery를 이용하는 방법

```
ContentResolver cr = getContentResolver();  
Cursor cur = managedQuery(myPerson, null, null, null, null)
```


ContentProvider 사용하기

❖ ContentProvider에 Query 하기 : managedQuery()

✓ managedQuery()의 특성

- managedQuery의 경우 Activity가 Cursor의 생명 주기를 관리
- 또한, Activity에서 관리 되지 않는 Cursor에 대해 Activity.startManagingCursor 메소드를 사용해 Activity가 Cursor를 관리하도록 할 수 있음

✓ managedQuery(URI, null, null, null, null)

- 리턴받아야 하는 데이터 컬럼의 이름들. 널 값은 모든 모든 컬럼을 리턴한다. 그렇지 않다면, 그 이름으로 리스트된 컬럼만 리턴된다. 그 플랫폼에 속하는 모든 콘텐츠 프로바이더는 그들의 컬럼에 대한 상수를 정의한다. 예를 들어 android.provider.Contacts.Phones 클래스는 앞에서 설명된 phone 테이블에 있는 _ID, NUMBER, NUMBER_KEY, NAME, 그리고 기타 등등의 컬럼 이름에 대한 상수를 정의한다.
- yy SQL WHERE 절 (WHERE 자식은 제약)로 형성되는 것으로, 리턴할 행들을 기술하는 필터. 널(null) 값은 모든 행들을 리턴한다. (단, URI가 하나의 레코드로 쿼리를 제한하지 않는 한) ▪
- yy 선택할 것에 대한 아규먼트들
- yy SQL ORDER BY 절 (ORDER BY 자식은 제약)로 형성되는 것으로, 리턴될 행에 대한 정렬 순서. 널(null) 값은 테이블에 대한 디폴트 순서, 어쩌면 정렬되지 않을 수도 있는 형태로 레코드들을 리턴한다.

ContentProvider 사용하기

❖ ContentProvider에 Query 하기 : managedQuery() 질의 예

```
import android.provider.Contacts.People;
import android.database.Cursor;
// Form an array specifying which columns to return.
String[] projection = new String[] {
    People._ID,
    People._COUNT,
    People.NAME,
    People.NUMBER
};
// Get the base URI for the People table
// in the Contacts content provider.
Uri contacts = People.CONTENT_URI;
// Make the query.
Cursor managedCursor = managedQuery(contacts,
    projection, // Which columns to return
    null, // Which rows to return (all rows)
    null, // Selection arguments (none)
    // Put the results in ascending order by name
    People.NAME + " ASC");
```

ContentProvider 사용하기

❖ ContentProvider에서 Cursor 이용

```
private void getColumnData(Cursor cur){
    if (cur.moveToFirst()) {
        String name;
        String phoneNumber;
        int nameColumn = cur.getColumnIndex(People.NAME);
        int phoneColumn = cur.getColumnIndex(People.NUMBER);
        String imagePath;
        do {
            // Get the field values
            name = cur.getString(nameColumn);
            phoneNumber = cur.getString(phoneColumn);
            // Do something with the values.
            ...
        } while (cur.moveToNext());
    }
}
```

ContentProvider 사용하기

❖ ContentProvider : Query

- ✓ ID를 이용해 특정 Row에 접근하기
 - 특정 Row의 URI 생성 메소드
 - `ContentUris.withAppendedId()`
 - `Uri.withAppendedPath()`

```
import android.provider.Contacts.People;
import android.content.ContentUris;
import android.net.Uri;
import android.database.Cursor;

// Use the ContentUris method to produce the base URI
// for the contact with _ID == 23.
Uri myPerson = ContentUris.withAppendedId(People.CONTENT_URI, 23)
;
// Alternatively, use the Uri method to produce the base URI.
// It takes a string rather than an integer.
Uri myPerson = Uri.withAppendedPath(People.CONTENT_URI, "23");

// Then query for this specific record:
Cursor cur = managedQuery(myPerson, null, null, null, null);
```

ContentProvider 사용하기

❖ ContentProvider에서 데이터 수정하기

- ✓ Content Provider에 의해 유지되는 데이터들은 ContentResolver 메소드를 사용하여 다음과 같은 작업들을 수행할 수 있음
 - 새로운 레코드를 추가하기
 - 기존의 레코드에 새로운 값을 추가하기
 - 기존의 레코드를 배치로 업데이트 하기
 - 레코드를 삭제하기

ContentResolver Class의 주요 메소드

용도	메리트	개요
추가	insert	지정한 데이터를 추가함
갱신	update	지정한 데이터를 갱신함
삭제	delete	지정한 데이터를 삭제함
검색	query	데이터를 검색하고 결과를 나타내는 Cursor를 반환함. 보통 Cursor의 관리를 실행하기 위해 Activity 클래스의 managedQuery 메소드를 사용함
내용의 검색	openInputStream	지정한 URI에서 데이터를 검색함
내용의 입력	openOutputStream	지정한 URI로 데이터를 입력함

ContentProvider 사용하기

❖ 데이터 삽입하기

```
ContentValues values = new ContentValues();
values.put(People.NAME, "Abraham Lincoln");
values.put(People.STARRED, 1);
Uri uri = getContentResolver().insert(People.CONTENT_URI, values);
```

❖ 데이터 갱신하기

```
ContentValues newValues = new ContentValues();
newValues.put(COLUMN_NAME, "값" );
String where = "_id < 5";
getContentResolver().update(MyProvider.CONTENT_URI, newValues, where, null);
```

❖ 데이터 삭제하기

```
// 특정 행 삭제
getContentResolver().delete(myRowUri, null, null);
// 조건을 주고 삭제
String where = "_id < 5";
getContentResolver().delete(MyProvider.CONTENT_URI, where, null);
```

ContentProvider 사용하기

❖ 예 : 전화번호부에 전화번호와 email추가하기

```
Uri phoneUri = null;
Uri emailUri = null;
phoneUri = Uri.withAppendedPath(uri, People.Phones.CONTENT_DIRECTORY);
values.clear();
values.put(People.Phones.TYPE, People.Phones.TYPE_MOBILE);
values.put(People.Phones.NUMBER, "1233214567");
getContentResolver().insert(phoneUri, values);

// Now add an email address in the same way.
emailUri= Uri.withAppendedPath(uri, People.ContactMethods.CONTENT_DIRECTORY);
values.clear();
// ContactMethods.KIND is used to distinguish different kinds of
// contact methods, such as email, IM, etc.
values.put(People.ContactMethods.KIND, Contacts.KIND_EMAIL);
values.put(People.ContactMethods.DATA, "test@example.com");
values.put(
    People.ContactMethods.TYPE, People.ContactMethods.TYPE_HOME);
getContentResolver().insert(emailUri, values);
```

ContentProvider 사용하기

❖ ContentProvider 예 : 바이너리 데이터 보기

```
import android.provider.MediaStore.Images.Media;
import android.content.ContentValues;
import java.io.OutputStream;
// Save the name and description of an image in a ContentValues map.
ContentValues values = new ContentValues(3);
values.put(Media.DISPLAY_NAME, "road_trip_1");
values.put(Media.DESCRPTION, "Day 1, trip to Los Angeles");
values.put(Media.MIME_TYPE, "image/jpeg");
// Add a new record without the bitmap, but with the values just set.
// insert() returns the URI of the new record.
Uri uri = getContentResolver().insert(Media.EXTERNAL_CONTENT_URI, values);
// Now get a handle to the file for that record,
// and save the data into it.
// Here, sourceBitmap is a Bitmap object representing the file
// to save to the database.
try {
    OutputStream outputStream
        = getContentResolver().openOutputStream(uri);
    sourceBitmap.compress(Bitmap.CompressFormat.JPEG, 50, outputStream);
    outputStream.close();
} catch (Exception e) {
    Log.e(TAG, "exception while writing image", e);
}
```


ContentProvider 사용하기

❖ ContentProvider 예 : 통화 로그 보기

```
private String getCallLog(int duration) {
    // 검색 실행
    Cursor c = getContentResolver().query(CallLog.Calls.CONTENT_URI, projection, selection, null,
        order);
    // query(URI, 불러올 컬럼, where 절,
    // 결과가 없으면 null 반환
    if(c == null) return "";
    // 검색 결과를 하나의 문자열로 정리한다.
    StringBuffer buff = new StringBuffer();
    Resources res = getResources();
    if(c.moveToFirst()) {
        do{
            // 착신인지 발신인지를 조사한다.
            if(c.getInt(TYPE_INDEX) == CallLog.Calls.INCOMING_TYPE) {
                buff.append(res.getString(R.string.incomming));
            } else {
                buff.append(res.getString(R.string.outcomming));
            }
            buff.append(" : ");
            // 전화번호
            buff.append(c.getString(NUMBER_INDEX));
            buff.append(" : ");
            // 통화시간
            buff.append(c.getInt(DURATION_INDEX));
            buff.append("\n");
        } while (c.moveToNext());
    }
    c.close();
    return buff.toString();
}
```