

22

CHAPTER

데이터베이스 활용

22.1 데이터베이스 활용 예와 원리

- 22.1.1 데이터베이스 활용의 예
- 22.1.2 데이터베이스(SQLite) 활용 원리
- 22.1.3 동적으로 데이터 출력하기

22.2 Contents Mgt 프로젝트의 DB 연동(갤러리 앱)

- 22.2.1 프로젝트 개요
- 22.2.2 파일 구조와 기능
- 22.2.3 텍스트 리소스와 데이터 매니저 편집
- 22.2.4 갤러리 홈 편집
- 22.2.5 '사진 등록' 액티비티 편집
- 22.2.6 '사진 정보' 액티비티와 '사진 삭제' 편집
- 22.2.7 '사진 목록' 액티비티 편집
- 22.2.8 환경 설정과 실행

22 1

데이터베이스 활용 예와 원리

22.1.1 데이터베이스 활용의 예

스마트폰은 내부적으로 전화번호를 관리하는 경우, **데이터베이스**를 이용하면 정보관리가 편리하다. 또한, 네트워크를 통해 정보를 교환하는 앱의 경우에도 변화가 없는 일부 자료는 앱의 데이터베이스에 보관하고 실시간으로 변하는 정보는 필요한 시점에 다운로드하면 정보관리의 효율성이 좋아지게 된다. 아래 그림의 왼쪽은 **노선버스의 번호 검색**에 관한 것으로, 노선버스 번호는 스마트폰의 데이터베이스에서 관리하고 버스에 관한 실시간 정보는 네트워크를 통해 다운로드 받으면 된다. 오른쪽은 **고속도로의 구간별 교통상황**으로 구간정보는 데이터베이스에서 관리하고 구간의 실시간 정보는 네트워크를 통해 출력하면 효율성이 높아진다.



(a) 노선버스번호 검색



(b) 고속도로 정보

- 데이터베이스 활용의 예

22.1.2 데이터베이스(SQLite) 활용 원리

SQLite(www.sqlite.com)는 가벼운 관계형 데이터베이스관리시스템(DBMS: Database Management System)으로 안드로이드 폰에 내장되어 있다. 21장에서 개발한 사진정보 관리 앱을 SQLite를 사용해서 구현해 보자.

TIP 관계형 데이터베이스 (Relational database)

데이터베이스란 여러 사용자의 정보요구를 충족시키기 위해 서로 관련된 데이터를 체계적으로 구조화하여 저장하는 것이다. 그리고 관계형 데이터베이스란 데이터 개체를 사용자들이 이해하기 쉽게 데이터를 테이블 형태로 표현한 것이다.

사진정보를 구성하는 5개의 속성을 테이블로 만들어보자.

● 사진 테이블의 데이터 예

title	orientation	background	path
풍경1	가로	자연	...
풍경2	세로	자연 건물	...
...

(▶ 배경(background)은 '인물', '자연', '건물' 등의 다중 값이 존재하기 때문에 확장성을 고려하여 사진 테이블과 분리하는 것이 바람직하다. 그러나, 데이터베이스를 처음 접하는 경우 두 개의 테이블을 다루기는 어렵기 때문에, 여기서는 다중 값을 하나의 데이터로 붙여 저장하기로 한다.

● 테이블: photos

속성	데이터유형	설명
title	문자(text)	성명
orientation	문자(text)	성별
background	문자(text)	배경
path	문자(text)	사진 경로

SQLite에서 테이블을 만드는 문법은 다음과 같다.

● 기본 문법

```
01 create table 테이블명(속성명 데이터유형, 속성명 데이터유형, ...);
```

● PHOTOS 테이블의 생성 예

```
01 create table photos(title text, orientation text, background text, path text);
```

photos 테이블이 만들어지면 **SQLiteDatabase** 클래스와 **ContentValues** 클래스를 이용하여 데이터를 추가할 수 있다.

● 데이터 추가 문법

```
01 SQLiteDatabase sqlitedb;  
02 ContentValues values = new ContentValues();  
03 values.put("속성명", 속성값);  
04 ....  
05 long newRowId = sqlitedb.insert("테이블명", null, values);
```

▶ '테이블명'에 **values**를 추가한다. **values**가 공값이면 **null** 값으로 설정한다.

데이터가 정상적으로 추가되지 않으면 newRowId에는 -1이 반환된다

테이블에서 데이터를 추출하고 처리하는 방법은 **SQLiteDatabase** 와 **Cursor** 클래스를 이용한다.

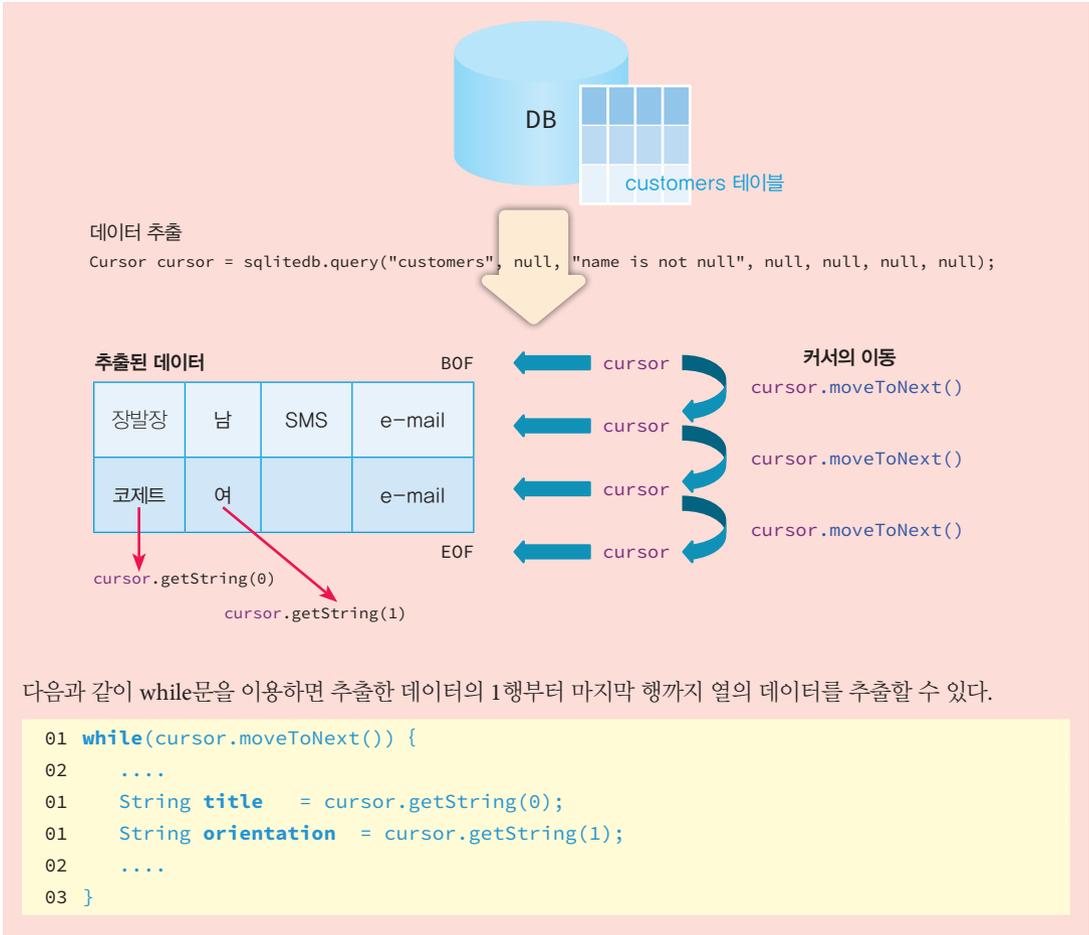
● 데이터 추출

photos 테이블에서 **title** 속성의 데이터가 null 이 아닌 조건을 만족하는 모든 속성들의 값을 추출한다.

```
01 SQLiteDatabase sqlitedb;  
02 Cursor cursor = sqlitedb.query("photos", null, "title is not null", null, null,  
    null, null);
```

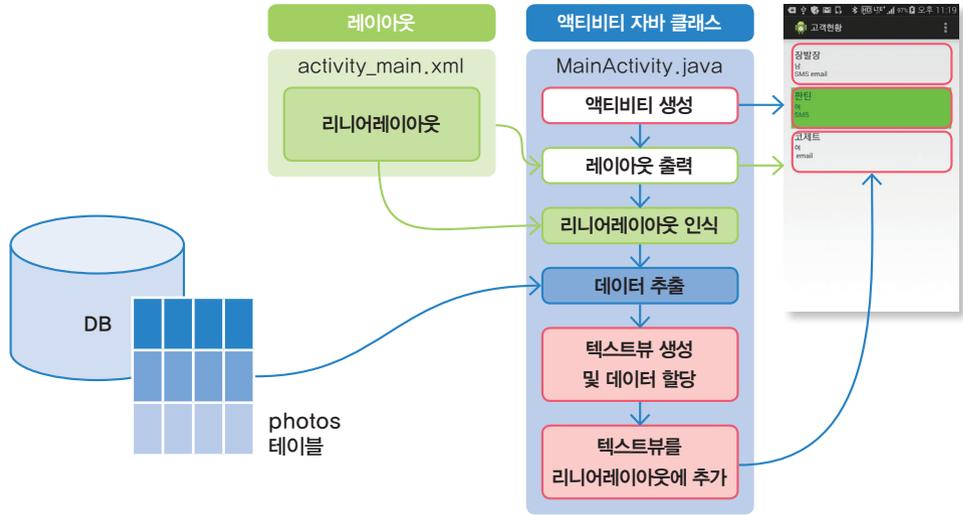
● 추출된 데이터의 처리

데이터는 일반적으로 여러 행에 걸쳐 나타나는데, 반복문을 이용하여 **cursor**를 다음 행으로 이동하면서 데이터를 처리한다. 데이터를 추출하면, 데이터가 존재할 때 커서(Cursor)는 첫 번째 데이터의 이전(BOF: Begin of File)을 가리키고 있다. **Cursor** 클래스의 **moveToNext()** 메소드를 이용하면 다음 행으로 이동이 가능하다. 만일 마지막 행의 다음(EOFL End of File)을 가리키고 있다면 **moveToNext()** 메소드는 **null**을 반환한다. **Cursor** 클래스의 객체 **cursor**가 임의의 행을 가리키고 있을 때, 첫 번째 열과 두 번째 열의 데이터 추출은 **getString(0)**, **getString(1)**과 같이 하면 된다. 첫 번째 속성은 **title**이기 때문에 색인 대신에 속성명을 사용하면 **getString(0)** 대신 **getString(cursor.getColumnIndex("title"))**를 사용하면 된다.



22.1.3 동적으로 데이터 출력하기

실시간으로 데이터베이스에서 데이터를 추출하거나 공공 DB 사이트에서 데이터를 수신하는 경우, 출력해야 할 데이터가 사전에 정해져 있지 않기 때문에 데이터 발생 시점에 화면으로 출력하게 된다. 예를 들면, 레이아웃에는 리니어레이아웃을 배치하고, 액티비티 자바 클래스에서 뷰에 데이터를 할당 후, 앞에서 배치한 리니어레이아웃 내에 반복해서 추가하면 된다.



22 2

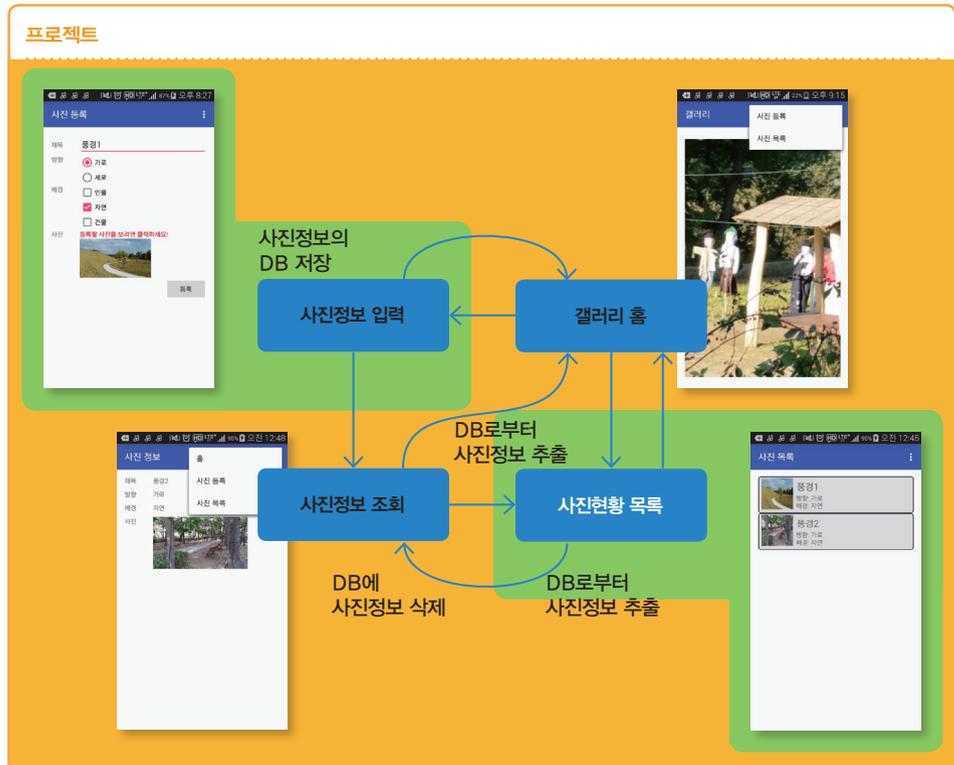
Contents Mgt 프로젝트의 DB 연동(갤러리 앱)

22.2.1 프로젝트 개요



21장의 프로젝트에서 입력하는 사진정보를 SQLite 데이터베이스에 저장하고, 저장된 정보들을 목록으로 출력하는 앱으로 확장해서 만들어 보자.

갤러리 홈은 메뉴 아이템을 추가할 수 있다. **사진정보 입력**에는 등록 버튼을 클릭하면, 사진정보 조회 화면을 호출하기 전에 사진정보를 데이터베이스에 저장하는 기능을 추가한다. **사진정보 조회**는 저장한 정보를 삭제할 수 있는 기능을 추가한다. 한편, **사진현황 목록**은 데이터베이스에 저장된 모든 사진정보를 추출하여 목록으로 출력하고, 목록의 아이템을 클릭하면 사진정보를 조회하는 화면으로 이동하는 액티비티로 새로 만들 수 있다.



■ 수정 ■ 추가

22.2.2 파일 구조와 기능

21장의 프로젝트에서 수정되고 추가되는 부분은 다음과 같다.

모듈	폴더	소스 파일	편집 내용	관련 액티비티	
manifests		AndroidManifest.xml	<ul style="list-style-type: none"> • ContentsReg 클래스 등록 • ContentsItem 클래스 등록 • ContentsList 클래스 등록 	갤러리홈, 사진정보 입력, 사진정보 조회, 사진정보 목록	
java	com.example.ychang.contentsmgt	MainActivity.java	<ul style="list-style-type: none"> • 이미지 출력 • 메뉴 아이템 구성 	갤러리홈	
		ContentsReg.java	<ul style="list-style-type: none"> • 사진정보 입력화면 • 사진정보 DB 등록 • 사진정보 조회 액티비티 호출 	사진정보 입력	
		ContentsList.java	<ul style="list-style-type: none"> • DB에서 사진정보 추출 • 사진정보 목록 출력 	사진정보 목록	
		ContentsItem.java	<ul style="list-style-type: none"> • DB에서 사진정보 추출 • 사진정보 출력 	사진정보 조회	
res	drawable	home.png		갤러리홈	
	layout	activity_main.xml	<ul style="list-style-type: none"> • 메인 화면 구성 틀 	갤러리홈	
		content_main.xml	<ul style="list-style-type: none"> • 메인 화면의 배경 이미지 배치 	갤러리홈	
		activity_reg.xml	<ul style="list-style-type: none"> • 사진정보 입력화면 구성 틀 	사진정보 입력	
		contents_reg.xml	<ul style="list-style-type: none"> • 사진정보 입력 컨트롤의 배치 	사진정보 입력	
		activity_item.xml	<ul style="list-style-type: none"> • 사진정보 출력화면 구성 틀 	사진정보 조회	
		content_item.xml	<ul style="list-style-type: none"> • 사진정보 출력화면 배치 	사진정보 조회	
		activity_list.xml	<ul style="list-style-type: none"> • 사진정보 목록 화면 구성 틀 	사진정보 목록	
		content_list.xml	<ul style="list-style-type: none"> • 사진정보 목록 화면 배치 	사진정보 목록	
	menu	menu_main.xml	<ul style="list-style-type: none"> • 메인 화면 액션 바의 메뉴 아이템 	갤러리홈	
		menu_reg.xml	<ul style="list-style-type: none"> • 사진정보 입력화면 액션 바의 메뉴 아이템 	사진정보 입력	
		menu_list.xml	<ul style="list-style-type: none"> • 사진정보 목록화면 액션 바의 메뉴 아이템 	사진정보 목록	
		menu_item.xml	<ul style="list-style-type: none"> • 사진정보 출력화면 액션 바의 메뉴 아이템 	사진정보 조회	
	values	colors.xml			
		dimens.xml			
strings.xml		<ul style="list-style-type: none"> • 어플리케이션 라벨 수정 • 화면 구성을 위한 문자 자원 정의 • 메뉴 아이템을 위한 문자 자원 정의 	갤러리홈, 사진정보 입력, 사진정보 조회, 사진정보 목록		
styles.xml					

■ 수정 ■ 추가

22.2.3 텍스트 리소스와 데이터 매니저 편집

● 편집

1 텍스트 자원

액티비티 라벨, 액션 바 메뉴 아이템, 화면 구성 문자 자원들을 추가한다.

소스 | strings.xml

```
01 <resources>
02   <string name="app_name">갤러리</string>
03   <string name="act_reg">사진 등록</string>
04   <string name="act_list">사진 목록</string>
05   <string name="act_item">사진 정보</string>
06
07   <string name="action_home">홈</string>
08   <string name="action_reg">사진 등록</string>
09   <string name="action_list">사진 목록</string>
10   <string name="action_delete">사진 삭제</string>
11
12   <string name="title">제목</string>
13   <string name="orientation">방향</string>
14   <string name="background">배경</string>
15   <string name="photo">사진</string>
16 </resources>
```

액티비티 라벨

메뉴 아이템

화면 구성 아이템

2 데이터 매니저 생성

처음에 데이터베이스와 테이블을 만들기 위한 **DBManager** 클래스를 만들어 사용한다. **DBManager** 클래스는 **SQLiteOpenHelper** 클래스로부터 데이터베이스와 테이블을 관리하는 메소드를 상속받으며, Context 클래스와 SQLite 데이터베이스를 관리하기 위해 SQL 명령어를 실행하는 기능이 있는 **SQLiteDatabase** 클래스를 사용하고 있다. DBManager 클래스 객체가 생성될 때 'myDB' 데이터베이스가 존재하지 않으면 새로 생성한다. DBManager 액티비티가 생성될 때는 'myDB' 내에 'customers' 테이블이 존재하지 않으면 새로 테이블을 만들게 된다.

소스 | DBManager.java → H

```
01 package com.example.yschang.contentsmgt;
02
03 import android.content.Context;
```

```

04 import android.database.sqlite.SQLiteDatabase;
05 import android.database.sqlite.SQLiteOpenHelper;
06
07 public class DBManager extends SQLiteOpenHelper {
08
09     public DBManager(Context context) {
10         super(context, "myDB", null, 1);
11     }
12
13     @Override
14     public void onCreate(SQLiteDatabase db) {
15         db.execSQL("create table photos(title text, orientation text,
16                    background text, path text);");
17     }
18
19     @Override
20     public void onUpgrade(SQLiteDatabase db, int oldVersion,
21                          int newVersion) {
22     }
23 }

```

DBManager 클래스 생성자

"myDB" 데이터베이스 생성

데이터베이스 생성 시 호출

photos 테이블 생성

데이터베이스 업그레이드 시 호출

클래스와 속성/메소드

클래스

클래스/인터페이스	설명
SQLiteOpenHelper	7행. 데이터베이스 생성과 버전을 관리하는 클래스
SQLiteDatabase	14행. SQLite 데이터베이스를 관리하는 메소드를 제공함

메소드

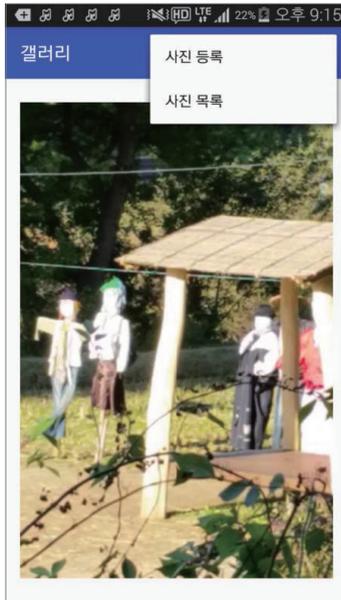
클래스	메소드	설명
SQLiteOpenHelper	SQLiteOpenHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version)	7행. 데이터베이스를 생성하며, 오픈하고, 관리하는 Object를 만들어줌
	abstract void onCreate(SQLiteDatabase db)	14행. 데이터베이스가 처음 생성될 때 불러짐
	abstract void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)	19행. 데이터베이스가 업그레이드될 필요가 있을 때 불러짐
SQLiteDatabase	void execSQL(String sql)	15행. 문자열(String)로 작성된 SQL문(조회문 제외)을 실행함

22.2.4 갤러리 홈 편집



메뉴에 사진 목록을 추가한다.

프로젝트



메인 액티비티의 메뉴는 '사진등록'과 '사진 목록'으로 구성

● 파일 간의 연관관계

strings.xml에는 초기치로 설정되어 있는 어플리케이션 라벨을 '갤러리'로 수정한다.

drawable 폴더에 초기 화면에 사용할 이미지(home.png)를 저장한다.

content_main.xml에는 출력할 이미지(home.png)를 배치한다.

이미지 추가



home.png (drawable)

메뉴 아이템 리소스

```

menu
  item
    id @+id/action_settings
    title @string/action_settings
    
```

메뉴 아이템

menu_main.xml (menu)

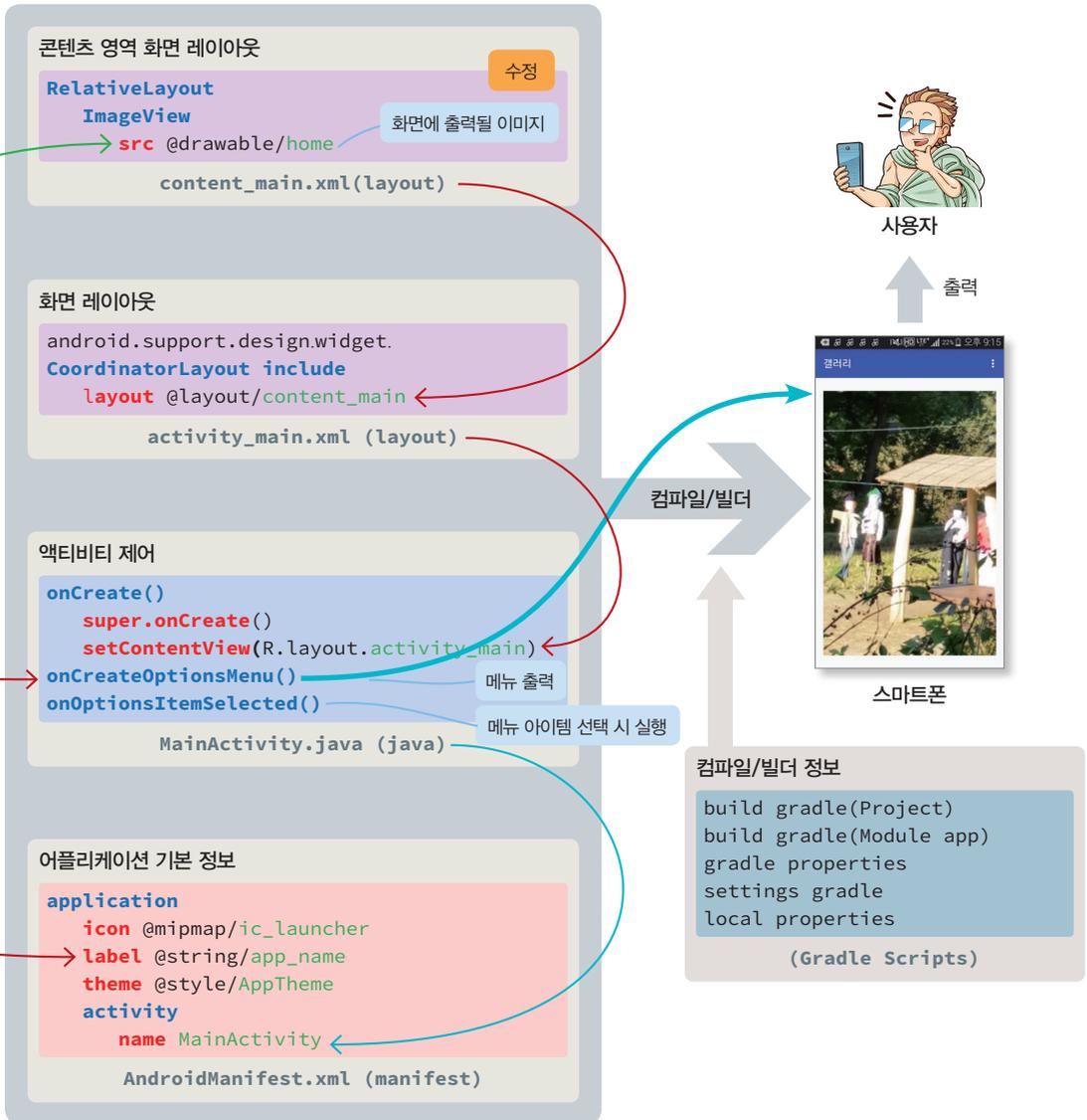
텍스트 리소스 수정

```

string
  app_name 갤러리
  action_settings Settings
    
```

앱 라벨

strings.xml (values)



● 편집

1 이미지 파일

21장의 **drawable** 폴더에 있는 **home.png**를 그대로 사용한다.

모듈	폴더	소스 파일	이미지
res	drawable	home.png	메인 화면 이미지

2 메뉴 구성

menu_main.xml에 '사진 등록'과 '사진 목록' 메뉴 아이템을 구성한다.

```
소스 | menu_main.xml → I
01 <menu xmlns:android="http://schemas.android.com/apk/res/android"
02     xmlns:app="http://schemas.android.com/apk/res-auto"
03     xmlns:tools="http://schemas.android.com/tools"
04     tools:context="com.example.contentsmgmt.MainActivity">
05     <item
06         android:id="@+id/action_reg" J
07         android:orderInCategory="100"
08         android:title="@string/action_reg"
09         app:showAsAction="never" /> A
10     <item
11         android:id="@+id/action_list" K
12         android:orderInCategory="100"
13         android:title="@string/action_list"
14         app:showAsAction="never" /> B
15 </menu>
```

“사진 등록” 메뉴 아이템

“사진 목록” 메뉴 아이템

3 화면 배치(res/drawable 폴더)

21장의 **layout** 폴더에 있는 **activity_main.xml**과 **contents_main.xml**을 그대로 사용한다.

모듈	폴더	소스 파일	내용
res	layout	activity_main.xml	21장과 동일 → L
res	layout	contents_main.xml	21장과 동일

4 액티비티 제어

사진 등록과 사진 목록 메뉴 아이템을 클릭하면 이동하는 기능을 추가한다.

소스 | MainActivity.java

```
01 package com.example.yschang.contentsmgt;
02
03 import android.content.Intent;
04 import android.os.Bundle;
05 import android.support.v7.app.AppCompatActivity;
06 import android.support.v7.widget.Toolbar;
07 import android.view.Menu;
08 import android.view.MenuItem;
09
10 public class MainActivity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
17         setSupportActionBar(toolbar);
18     }
19
20     @Override
21     public boolean onCreateOptionsMenu(Menu menu) {
22         getMenuInflater().inflate(R.menu.menu_main, menu);
23         return true;
24     }
25
26     @Override
27     public boolean onOptionsItemSelected(MenuItem item) {
28         int id = item.getItemId();
29
30         if(id == R.id.action_reg) {
31             Intent it = new Intent(this, ContentsReg.class);
32             startActivity(it);
33             finish();
34             return true;
35         }
36
37         if(id == R.id.action_list) {
38             Intent it = new Intent(this, ContentsList.class);
39             startActivity(it);
40             finish();
41             return true;
42         }
43
44         return super.onOptionsItemSelected(item);
45     }
46 }
```

선택된 메뉴 아이템의 id 인식

선택된 메뉴 아이템의 id가 action_reg일 때 ContentsReg 클래스 호출

선택된 메뉴 아이템의 id가 action_list일 때 ContentsList 클래스 호출

22.2.5 '사진 등록' 액티비티 편집

'사진 등록' 액티비티의 '등록' 버튼을 클릭하면 '사진 정보' 액티비티로 정보가 전달되게 된다. 이제는 '등록' 버튼을 클릭하면 사진을 데이터베이스에 저장하고 제목 데이터만 전달하고, '사진 정보' 액티비티는 데이터베이스에서 제목에 해당하는 사진 정보를 추출하여 출력한다.

프로젝트



사진정보의 입력 및 DB 저장

STEP 2 ▶ 파일 편집

● 편집

① 화면 설계

activity_reg.xml은 21장의 activity_reg_form.xml과 같으며, 콘텐츠는 contents_reg로 설정한다.

소스 | activity_reg.xml → M

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```

03     xmlns:app="http://schemas.android.com/apk/res-auto"
04     xmlns:tools="http://schemas.android.com/tools"
05     android:layout_width="match_parent"
06     android:layout_height="match_parent"
07     android:fitsSystemWindows="true"
08     tools:context="com.example.contentsmgmt.MainActivity">
09
10     <android.support.design.widget.AppBarLayout
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:theme="@style/AppTheme.AppBarOverlay">
14
15         <android.support.v7.widget.Toolbar
16             android:id="@+id/toolbar"
17             android:layout_width="match_parent"
18             android:layout_height="?attr/actionBarSize"
19             android:background="?attr/colorPrimary"
20             app:popupTheme="@style/AppTheme.PopupOverlay" />
21
22     </android.support.design.widget.AppBarLayout>
23
24     <include layout="@layout/contents_reg" />
25
26 </android.support.design.widget.CoordinatorLayout>

```

Annotations in the image:
 - A red circle with 'N' points to the `android.support.v7.widget.Toolbar` tag.
 - A blue callout box with the text "사진등록 화면 배치" (Photo registration screen layout) points to the `<include layout="@layout/contents_reg" />` tag.
 - A red circle with 'O' points to the `<include layout="@layout/contents_reg" />` tag.

contents_reg.xml은 21장의 contents_reg.form.xml과 같다.

소스 | contents_reg.xml → O

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
03     android:layout_width="match_parent"
04     android:layout_height="match_parent"
05     android:orientation="vertical"
06     android:paddingBottom="@dimen/activity_vertical_margin"
07     android:paddingLeft="@dimen/activity_horizontal_margin"
08     android:paddingRight="@dimen/activity_horizontal_margin"
09     android:paddingTop="70dp" >
10
11     <LinearLayout
12         android:orientation="horizontal"
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content" >
15         <TextView
16             android:layout_width="60sp"

```

```

17         android:layout_height="wrap_content"
18         android:text="@string/title" />
19     <EditText
20         android:id="@+id/title"
21         android:layout_width="match_parent"
22         android:layout_height="wrap_content" />
23 </LinearLayout>
24
25 <LinearLayout
26     android:orientation="horizontal"
27     android:layout_width="match_parent"
28     android:layout_height="wrap_content" >
29     <TextView
30         android:layout_width="60sp"
31         android:layout_height="wrap_content"
32         android:text="@string/orientation" />
33     <RadioGroup
34         android:id="@+id/orientation"
35         android:layout_width="match_parent"
36         android:layout_height="wrap_content"
37         android:orientation="vertical" >
38         <RadioButton
39             android:id="@+id/landscape"
40             android:layout_width="wrap_content"
41             android:layout_height="wrap_content"
42             android:text="가로" />
43         <RadioButton
44             android:id="@+id/portrait"
45             android:layout_width="wrap_content"
46             android:layout_height="wrap_content"
47             android:text="세로" />
48     </RadioGroup>
49 </LinearLayout>
50
51 <LinearLayout
52     android:orientation="horizontal"
53     android:layout_width="match_parent"
54     android:layout_height="wrap_content" >
55     <TextView
56         android:layout_width="60sp"
57         android:layout_height="wrap_content"
58         android:text="@string/background" />
59
60     <LinearLayout
61         android:orientation="vertical"

```

```

62         android:layout_width="match_parent"
63         android:layout_height="wrap_content" >
64     <CheckBox
65         android:id="@+id/background1"
66         android:layout_width="wrap_content"
67         android:layout_height="wrap_content"
68         android:text="인물" />
69     <CheckBox
70         android:id="@+id/background2"
71         android:layout_width="wrap_content"
72         android:layout_height="wrap_content"
73         android:text="자연" />
74     <CheckBox
75         android:id="@+id/background3"
76         android:layout_width="wrap_content"
77         android:layout_height="wrap_content"
78         android:text="건물" />
79     </LinearLayout>
80 </LinearLayout>
81
82 <LinearLayout
83     android:orientation="horizontal"
84     android:layout_width="match_parent"
85     android:layout_height="wrap_content" >
86     <TextView
87         android:layout_width="60sp"
88         android:layout_height="wrap_content"
89         android:text="@string/photo" />
90     <LinearLayout
91         android:orientation="vertical"
92         android:layout_width="match_parent"
93         android:layout_height="wrap_content" >
94         <TextView
95             android:id="@+id/tv_photo"
96             android:layout_width="wrap_content"
97             android:layout_height="wrap_content"
98             android:textColor="#ff0000"
99             android:text="등록할 사진을 보려면 클릭하세요!"
100            android:clickable="true"
101            android:onClick="selectPhoto" />
102         <ImageView
103             android:id="@+id/photo"
104             android:layout_width="150dp"
105             android:layout_height="wrap_content"
106             android:adjustViewBounds="true" />

```

```

107     </LinearLayout>
108 </LinearLayout>
109
110 <LinearLayout
111     android:orientation="horizontal"
112     android:gravity="right"
113     android:layout_width="match_parent"
114     android:layout_height="wrap_content" >
115     <Button
116         android:layout_width="wrap_content"
117         android:layout_height="wrap_content"
118         android:text="등록"
119         android:onClick="register" />
120 </LinearLayout>
121 </LinearLayout>

```

2 메뉴 구성

menu_reg.xml은 메뉴 아이템을 갤러리 홈과 사진 목록으로 구성한다.

소스 | menu_reg.xml → P

```

01 <menu xmlns:android="http://schemas.android.com/apk/res/android"
02     xmlns:app="http://schemas.android.com/apk/res-auto"
03     xmlns:tools="http://schemas.android.com/tools"
04     tools:context="com.example.contentsmgmt.MainActivity">
05     <item
06         android:id="@+id/action_home"
07         android:orderInCategory="100"
08         android:title="@string/action_home"
09         app:showAsAction="never" />
10     <item
11         android:id="@+id/action_list"
12         android:orderInCategory="100"
13         android:title="@string/action_list"
14         app:showAsAction="never" />
15 </menu>

```

Diagram annotations: A red box highlights the two <item> elements. Arrows point from labels Q, R, D, and F to specific attributes: Q points to android:id="@+id/action_home", R points to android:id="@+id/action_list", D points to app:showAsAction="never" in the first item, and F points to app:showAsAction="never" in the second item.

3 액티비티 제어

ContentsReg.java는 1장의 ContentsRegForm.java에서 입력 정보를 ContentsDetail.java로 전송하는 대신, 데이터베이스에 저장하고 사진 제목만 ContentsItem.java로 전송하는 것으로 수정한다.

소스 | ContentsReg.java 

```
01 package com.example.yschang.contentsmgt;
02
03 import android.content.ContentValues;
04 import android.content.Context;
05 import android.content.Intent;
06 import android.database.Cursor;
07 import android.database.sqlite.SQLiteDatabase;
08 import android.database.sqlite.SQLiteException;
09 import android.graphics.Bitmap;
10 import android.net.Uri;
11 import android.os.Bundle;
12 import android.provider.MediaStore;
13 import android.support.v7.app.AppCompatActivity;
14 import android.support.v7.widget.Toolbar;
15 import android.view.Menu;
16 import android.view.MenuItem;
17 import android.view.View;
18 import android.widget.CheckBox;
19 import android.widget.EditText;
20 import android.widget.ImageView;
21 import android.widget.RadioButton;
22 import android.widget.RadioGroup;
23 import android.widget.TextView;
24 import android.widget.Toast;
25
26 import java.io.FileNotFoundException;
27 import java.io.IOException;
28
29 public class ContentsReg extends AppCompatActivity {
30
31
32     public final static int PICK_PHOTO_CODE = 1046;
33     ImageView iv_photo;
34     TextView tv_photo;
35     String str_path;
36
37     SQLiteDatabase sqldb;
38     DBManager dbmanager;
39
40     @Override
41     protected void onCreate(Bundle savedInstanceState) {
42         super.onCreate(savedInstanceState);
43         setContentView(R.layout.activity_reg);
```



```

44     Toolbar toolbar =(Toolbar) findViewById(R.id.toolbar);
45     setSupportActionBar(toolbar);
46
47     iv_photo =(ImageView) findViewById(R.id.photo);
48     tv_photo =(TextView) findViewById(R.id.tv_photo);
49 }
50
51 public void selectPhoto(View v) {
52     Intent intent = new Intent(Intent.ACTION_PICK,
53         MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
54     startActivityForResult(intent, PICK_PHOTO_CODE);
55 }
56
57 @Override
58 public void onActivityResult(int requestCode, int resultCode,
59                                 Intent data) {
60
61     super.onActivityResult(requestCode, resultCode, data);
62
63     Uri photoUri = data.getData();
64     str_path = getRealPathFromURI(this, photoUri);
65
66     try {
67         Bitmap bm = MediaStore.Images.Media.getBitmap(
68             this.getContentResolver(), photoUri);
69
70         int WIDTH = 150;
71         int width = WIDTH;
72         float ratio =(float) bm.getHeight() / bm.getWidth();
73         int height =(int)(WIDTH * ratio);
74
75         Bitmap thumb = Bitmap.createScaledBitmap(
76             bm, width, height, false);
77
78         iv_photo.setImageBitmap(thumb);
79     } catch(FileNotFoundException e) {
80         Toast.makeText(this, e.getMessage(), Toast.LENGTH_SHORT).show();
81     } catch(IOException e) {
82         Toast.makeText(this, e.getMessage(), Toast.LENGTH_SHORT).show();
83     }
84 }
85
86 public String getRealPathFromURI(Context context, Uri contentUri) {
87     Cursor cursor = null;
88     try {
89         String[] proj = { MediaStore.Images.Media.DATA };

```

```

86         cursor = context.getContentResolver().query(
87             contentUri, proj, null, null, null);
88         int column_index = cursor.getColumnIndexOrThrow(
89             MediaStore.Images.Media.DATA);
90         cursor.moveToFirst();
91         return cursor.getString(column_index);
92     } finally {
93         if(cursor != null) {
94             cursor.close();
95         }
96     }
97     public void register(View v) {
98         EditText et_title = (EditText)findViewById(R.id.title);
99         String str_title = et_title.getText().toString();
100
101         RadioGroup rg_orientation = (RadioGroup)findViewById(
102             R.id.orientation);
103
104         RadioButton rb_orientation;
105         String str_orientation = "";
106         if(rg_orientation.getCheckedRadioButtonId() == R.id.landscape) {
107             rb_orientation = (RadioButton)findViewById(R.id.landscape);
108             str_orientation = rb_orientation.getText().toString();
109         }
110         if(rg_orientation.getCheckedRadioButtonId() == R.id.portrait) {
111             rb_orientation = (RadioButton)findViewById(R.id.portrait);
112             str_orientation = rb_orientation.getText().toString();
113         }
114
115         CheckBox chk_background;
116         String str_background1 = "";
117         String str_background2 = "";
118         String str_background3 = "";
119
120         chk_background = (CheckBox)findViewById(R.id.background1);
121         if(chk_background.isChecked()) {
122             str_background1 = (String)chk_background.getText();
123         }
124         chk_background = (CheckBox)findViewById(R.id.background2);
125         if(chk_background.isChecked()) {
126             str_background2 = (String)chk_background.getText();
127         }
128         chk_background = (CheckBox)findViewById(R.id.background3);
129         if(chk_background.isChecked()) {

```

```

128     str_background3 =(String)chk_background.getText();
129 }
130
131 String str_background = str_background1 + " " + str_background2 +
132                             " " + str_background3;
133 str_background = str_background.trim();
134
135 try {
136     dbmanager = new DBManager(this);
137     sqllitedb = dbmanager.getWritableDatabase();
138     ContentValues values = new ContentValues();
139     values.put("title", str_title);
140     values.put("orientation", str_orientation);
141     values.put("background", str_background);
142     values.put("path", str_path);
143     long newRowId = sqllitedb.insert("photos", null, values);
144     sqllitedb.close();
145     dbmanager.close();
146
147     Intent it = new Intent(this, ContentsItem.class);
148     it.putExtra("it_title", str_title);
149     startActivity(it);
150     finish();
151 } catch(SQLiteException e) {
152     Toast.makeText(this, e.getMessage(), Toast.LENGTH_LONG).show();
153 }
154
155 @Override
156 public boolean onCreateOptionsMenu(Menu menu) {
157     // Inflate the menu; this adds items to the action bar
158     // if it is present.
159     getMenuInflater().inflate(R.menu.menu_reg, menu);
160     return true;
161 }
162
163 @Override
164 public boolean onOptionsItemSelected(MenuItem item) {
165     int id = item.getItemId();
166     if(id == R.id.action_home) {
167         Intent it = new Intent(this, MainActivity.class);
168         startActivity(it);
169         finish();
170         return true;

```

입력한 사진정보를 데이터베이스에 저장

데이터베이스에 데이터를 저장하기 위한 SQLiteDatabase 객체 생성

테이블의 title 속성에 값 할당

photos 테이블에 할당된 데이터 추가

SQLiteDatabase 객체 자원 해제

ContentsItem 클래스 호출

P

D

```

171     }
172
173     return super.onOptionsItemSelected(item);
174 }
175 }

```

클래스와 속성/메소드

● 클래스

클래스/인터페이스	설명
ContentValues	137행. ContentResolver 클래스가 처리할 수 있는 값들의 집합을 저장할 때 사용함
SQLiteException	150행. SQL 파싱이나 실행 시 에러를 명시

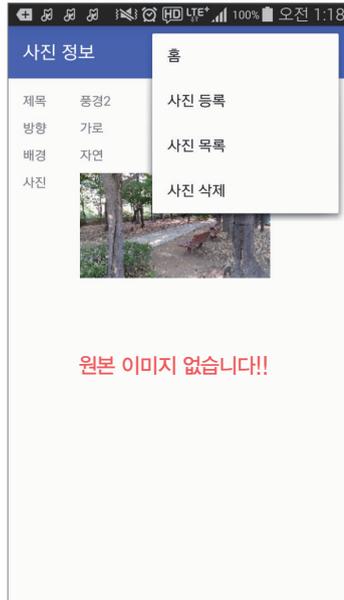
● 메소드

클래스	메소드	설명
ContentValues	void put (String key, String value)	138, 139, 140, 141행. 집합에 key와 value를 추가함
SQLiteOpenHelper	SQLiteDatabase getWritableDatabase ()	136행. 기록 또는 조회에 유용한 데이터베이스를 생성하거나 오픈함(데이터베이스 객체인 SQLiteDatabase 객체를 반환함)
SQLiteClosable	void close ()	143, 144행. 자원을 해제함. SQLiteDatabase는 SQLiteClosable의 서브 클래스임
SQLiteDatabase	long insert (String table, String nullColumnHack, ContentValues values)	142행. 데이터베이스에 한 행을 추가함. 추가 시에 에러가 발생하면 반환 값이 -1이 됨
String	String trim ()	132행. 문자열의 시작과 끝 부분에 있는 공백 문자 삭제한 문자열 반환

22.2.6 '사진 정보' 액티비티와 '사진 삭제' 편집

'사진 등록' 액티비티의 '등록' 버튼을 클릭하면 '사진 정보' 액티비티로 정보가 전달되게 된다. 이제는 '등록' 버튼을 클릭하면 사진을 데이터베이스에 저장하고 제목 데이터만 전달하고, '사진 정보' 액티비티는 데이터베이스에서 제목에 해당하는 사진 정보를 추출하여 출력한다.

프로젝트



메뉴 아이템은 홈, 사진 등록, 사진 목록, 사진 삭제로 구성함

STEP 2 > 파일 편집

● 편집

① 화면 설계

activity_item.xml은 21장의 activity_detail.xml과 같으며, 콘텐츠는 contents_item.xml으로 설정한다.

소스 | activity_item.xml —> T

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```

03  xmlns:app="http://schemas.android.com/apk/res-auto"
04  xmlns:tools="http://schemas.android.com/tools"
05  android:layout_width="match_parent"
06  android:layout_height="match_parent"
07  android:fitsSystemWindows="true"
08  tools:context="com.example.contentsmgmt.MainActivity">
09
10  <android.support.design.widget.AppBarLayout
11      android:layout_width="match_parent"
12      android:layout_height="wrap_content"
13      android:theme="@style/AppTheme.AppBarOverlay">
14
15      <android.support.v7.widget.Toolbar
16          android:id="@+id/toolbar"
17          android:layout_width="match_parent"
18          android:layout_height="?attr/actionBarSize"
19          android:background="?attr/colorPrimary"
20          app:popupTheme="@style/AppTheme.PopupOverlay" />
21
22  </android.support.design.widget.AppBarLayout>
23
24  <include layout="@layout/contents_item" />
25
26  </android.support.design.widget.CoordinatorLayout>

```

사진정보 출력하면 배치

contents_item.xml은 21장의 contents_detail.xml과 같다.

소스 | contents_item.xml

```

01  <?xml version="1.0" encoding="utf-8"?>
02  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
03      android:layout_width="match_parent"
04      android:layout_height="match_parent"
05      android:orientation="vertical"
06      android:paddingBottom="@dimen/activity_vertical_margin"
07      android:paddingLeft="@dimen/activity_horizontal_margin"
08      android:paddingRight="@dimen/activity_horizontal_margin"
09      android:paddingTop="70dp" >
10
11  <LinearLayout
12      android:orientation="horizontal"
13      android:layout_width="match_parent"
14      android:layout_height="wrap_content" >
15  <TextView
16      android:layout_width="60sp"

```

```

17         android:layout_height="wrap_content"
18         android:text="@string/title" />
19     <TextView
20         android:id="@+id/title"
21         android:layout_width="match_parent"
22         android:layout_height="wrap_content" />
23 </LinearLayout>
24
25 <LinearLayout
26     android:orientation="horizontal"
27     android:layout_width="match_parent"
28     android:layout_height="wrap_content"
29     android:paddingTop="10dp" >
30     <TextView
31         android:layout_width="60sp"
32         android:layout_height="wrap_content"
33         android:text="@string/orientation" />
34     <TextView
35         android:id="@+id/orientation"
36         android:layout_width="match_parent"
37         android:layout_height="wrap_content" />
38 </LinearLayout>
39
40 <LinearLayout
41     android:orientation="horizontal"
42     android:layout_width="match_parent"
43     android:layout_height="wrap_content"
44     android:paddingTop="10dp" >
45     <TextView
46         android:layout_width="60sp"
47         android:layout_height="wrap_content"
48         android:text="@string/background" />
49     <TextView
50         android:id="@+id/background"
51         android:layout_width="wrap_content"
52         android:layout_height="wrap_content" />
53 </LinearLayout>
54
55 <LinearLayout
56     android:orientation="horizontal"
57     android:layout_width="match_parent"
58     android:layout_height="wrap_content"
59     android:paddingTop="10dp" >
60     <TextView
61         android:layout_width="60sp"

```

```

62         android:layout_height="wrap_content"
63         android:text="@string/photo" />
64     <ImageView
65         android:id="@+id/photo"
66         android:layout_width="200dp"
67         android:layout_height="wrap_content"
68         android:adjustViewBounds="true" />
69 </LinearLayout>
70
71 </LinearLayout>

```

2 메뉴 구성

menu_item.xml은 메뉴 아이템을 갤러리 홈, 사진 목록, 사진 등록, 사진 삭제로 구성한다.

소스 | menu_item.xml W

```

01 <menu xmlns:android="http://schemas.android.com/apk/res/android"
02     xmlns:app="http://schemas.android.com/apk/res-auto"
03     xmlns:tools="http://schemas.android.com/tools"
04     tools:context="com.example.contentsmgmt.MainActivity">
05     <item
06         android:id="@+id/action_home" X
07         android:orderInCategory="100"
08         android:title="@string/action_home"
09         app:showAsAction="never" /> D
10     <item
11         android:id="@+id/action_list" Y
12         android:orderInCategory="100"
13         android:title="@string/action_list"
14         app:showAsAction="never" /> F
15     <item
16         android:id="@+id/action_reg" Z
17         android:orderInCategory="100"
18         android:title="@string/action_reg"
19         app:showAsAction="never" /> E
20     <item
21         android:id="@+id/action_delete" AA
22         android:orderInCategory="100"
23         android:title="@string/action_delete"
24         app:showAsAction="never" /> G
25 </menu>

```

“홈” 메뉴 아이템

“사진 목록” 메뉴 아이템

“사진 등록” 메뉴 아이템

“사진 삭제” 메뉴 아이템

④ 액티비티 제어

ContentsItem.java는 21장의 ContentsDetail.java에서 사진 정보를 보여주기 전에, 데이터베이스에서 해당 사진 데이터를 추출하여 출력하고 삭제 메뉴를 클릭하면 데이터베이스에서 정보가 삭제되는 기능을 추가하면 된다.

소스 | ContentsItem.java → AB

```
01 package com.example.yschang.contentsmgmt;
02
03 import android.content.Intent;
04 import android.database.Cursor;
05 import android.database.sqlite.SQLiteDatabase;
06 import android.database.sqlite.SQLiteException;
07 import android.graphics.Bitmap;
08 import android.graphics.BitmapFactory;
09 import android.os.Bundle;
10 import android.support.v7.app.AppCompatActivity;
11 import android.support.v7.widget.Toolbar;
12 import android.view.Menu;
13 import android.view.MenuItem;
14 import android.widget.ImageView;
15 import android.widget.TextView;
16 import android.widget.Toast;
17
18 public class ContentsItem extends AppCompatActivity {
19
20     SQLiteDatabase sqldb;
21     DBManager dbmanager;
22
23     String str_title;
24
25     @Override
26     protected void onCreate(Bundle savedInstanceState) {
27         super.onCreate(savedInstanceState);
28         setContentView(R.layout.activity_item);
29         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
30         setSupportActionBar(toolbar);
31
32         TextView tv_title = (TextView) findViewById(R.id.title);
33         TextView tv_orientation = (TextView) findViewById(R.id.orientation);
34         TextView tv_background = (TextView) findViewById(R.id.background);
35         ImageView iv_photo = (ImageView) findViewById(R.id.photo);
36     }
}
```

사진정보 출력을 위한 텍스트뷰와 이미지뷰 인식

```

37 Intent it = getIntent();
38 str_title = it.getStringExtra("it_title");
39 String str_orientation = "";
40 String str_background = "";
41 String str_path = "";
42
43 try {
44     dbmanager = new DBManager(this);
45     sqlitedb = dbmanager.getReadableDatabase();
46     Cursor cursor = sqlitedb.query("photos", null, "title = ?",
47         new String[]{str_title}, null, null, null);
48
49     int i = 0;
50     if(cursor.moveToNext()) {
51         str_orientation = cursor.getString(
52             cursor.getColumnIndex("orientation"));
53         str_background = cursor.getString(
54             cursor.getColumnIndex("background"));
55         str_path = cursor.getString(cursor.getColumnIndex("path"));
56     }
57
58     sqlitedb.close();
59     dbmanager.close();
60 } catch(SQLiteException e) {
61     Toast.makeText(this, e.getMessage(), Toast.LENGTH_LONG).show();
62 }
63
64 tv_title.setText(str_title);
65 tv_orientation.setText(str_orientation);
66 tv_background.setText(str_background);
67
68 BitmapFactory.Options bmOptions = new BitmapFactory.Options();
69 Bitmap bm = BitmapFactory.decodeFile(str_path, bmOptions);
70
71 int WIDTH = 200;
72 int width = WIDTH;
73 float ratio = (float) bm.getHeight() / bm.getWidth();
74 int height = (int)(WIDTH * ratio);
75
76 Bitmap thumb = Bitmap.createScaledBitmap(bm, width, height, true);
77 iv_photo.setImageBitmap(thumb);
78
79 }
80
81 @Override
82 public boolean onCreateOptionsMenu(Menu menu) {

```

현재 액티비티를 호출한 인텐트로부터 사진 제목 추출

photos 테이블에서 사진 정보 추출

데이터베이스에 데이터를 조회하기 위한 SQLiteDatabase 객체 생성

photos 테이블에서 사진 제목에 해당하는 사진 정보 추출

조회결과와 확인을 위한 커서 이동

사진의 방향, 배경, 이미지 경로 추출

추출한 사진정보 출력

추출한 사진의 경로에 해당하는 이미지 출력

```

79     getMenuInflater().inflate(R.menu.menu_item, menu);
80     return true;
81 }
82
83 @Override
84 public boolean onOptionsItemSelected(MenuItem item) {
85     int id = item.getItemId();
86
87     if(id == R.id.action_home) {
88         Intent it = new Intent(this, MainActivity.class);
89         startActivity(it);
90         finish();
91         return true;
92     }
93
94     if(id == R.id.action_list) {
95         Intent it = new Intent(this, ContentsList.class);
96         startActivity(it);
97         finish();
98         return true;
99     }
100
101     if(id == R.id.action_reg) {
102         Intent it = new Intent(this, ContentsReg.class);
103         startActivity(it);
104         finish();
105         return true;
106     }
107
108     if(id == R.id.action_delete) {
109
110         int del_count = deleteItem(str_title);
111         if(del_count > 0) {
112             Intent it = new Intent(this, ContentsList.class);
113             startActivity(it);
114             finish();
115         } else {
116             Toast.makeText(this, "사진을 삭제하지 못했습니다!",
117                             Toast.LENGTH_LONG).show();
118         }
119         return true;
120     }
121
122     return super.onOptionsItemSelected(item);
123 }

```

W

선택된 메뉴 아이템의 id 인식

초기화면으로 이동

사진목록 화면으로 이동

사진등록 화면으로 이동

AC

사진 제목에 해당하는 사진이 삭제되면
사진목록 화면으로 이동

```

123
124 private int deleteItem(String str_title) {
125     int row = 0;
126     try {
127         dbmanager = new DBManager(this);
128         sqllitedb = dbmanager.getReadableDatabase();
129         row = sqllitedb.delete("photos", "title=" + "=?",
130                               new String[]{str_title});
131         sqllitedb.close();
132         dbmanager.close();
133     } catch(SQLiteException e) {
134         Toast.makeText(this, e.getMessage(), Toast.LENGTH_LONG).show();
135     }
136     return row;
137 }

```

AC: photos 테이블에서 사진정보 삭제 후, 삭제된 행의 수 반환

H: photos 테이블에서 사진제목에 해당하는 사진정보 삭제

클래스와 속성/메소드

● 메소드

클래스	메소드	설명
Cursor	abstract int getColumnIndex (String columnName)	50, 51, 52행. 주어진 columnName에 대한 색인을 반환함
	abstract boolean moveToNext ()	49행. 커서(cursor)를 다음 행으로 이동함
SQLiteOpenHelper	SQLiteDatabase getReadableDatabase ()	45, 128행. 데이터베이스를 생성하거나 오픈함
SQLiteDatabase	int delete (String table, String whereClause, String[] whereArgs)	129행. 데이터베이스 행을 지움
	Cursor query (String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy)	46행. 주어진 table에 대한 질의문을 실행하여 실행한 결과로 생긴 데이터셋에 대한 커서(Cursor)를 반환함

22.2.7 '사진 목록' 액티비티 편집

등록된 사진정보를 추출하며 목록으로 출력한다.

프로젝트



데이터베이스에 저장된 사진정보의 목록

STEP 2 파일 편집

● 편집

① 화면 설계

목록을 출력한 틀은 contents_list.xml에서 설정한다.

소스 | activity_list.xml → AD

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
03     xmlns:app="http://schemas.android.com/apk/res-auto"
04     xmlns:tools="http://schemas.android.com/tools"
05     android:layout_width="match_parent"
06     android:layout_height="match_parent"
```

```

07     android:fitsSystemWindows="true"
08     tools:context="com.example.contentsmgmt.MainActivity">
09
10     <android.support.design.widget.AppBarLayout
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:theme="@style/AppTheme.AppBarOverlay">
14
15         <android.support.v7.widget.Toolbar AE
16             android:id="@+id/toolbar"
17             android:layout_width="match_parent"
18             android:layout_height="?attr/actionBarSize"
19             android:background="?attr/colorPrimary"
20             app:popupTheme="@style/AppTheme.PopupOverlay" />
21
22     </android.support.design.widget.AppBarLayout>
23
24     <include layout="@layout/contents_list" /> 사진목록 출력하면 배치
25
26 </android.support.design.widget.CoordinatorLayout>

```

사진 정보들이 목록으로 출력되도록 리니어 레이아웃으로 화면을 배치한다. 각 사진 아이템들은 액티비티 제어 부분에서 데이터베이스에 저장된 정보들을 추출하여 동적으로 리니어 레이아웃에 추가한다.

소스 | contents_list.xml → AF

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
03     android:layout_height="match_parent"
04     android:layout_width="match_parent" >
05     <LinearLayout
06         android:layout_width="match_parent"
07         android:layout_height="match_parent"
08         android:orientation="vertical"
09         android:paddingBottom="@dimen/activity_vertical_margin"
10         android:paddingLeft="@dimen/activity_horizontal_margin"
11         android:paddingRight="@dimen/activity_horizontal_margin"
12         android:paddingTop="70dp"
13         android:id="@+id/photos" >
14
15     </LinearLayout> AG
16 </ScrollView>

```

photos 테이블에 있는 사진 아이템을 추가하기 위한 사진목록의 리니어 레이아웃

2 메뉴 구성

메뉴 아이템은 갤러리 홈과 사진 등록으로 구성한다.

```
소스 | menu_list.xml → AH
01 <menu xmlns:android="http://schemas.android.com/apk/res/android"
02     xmlns:app="http://schemas.android.com/apk/res-auto"
03     xmlns:tools="http://schemas.android.com/tools"
04     tools:context="com.example.contentsmgmt.MainActivity">
05     <item
06         android:id="@+id/action_home" AI
07         android:orderInCategory="100"
08         android:title="@string/action_home"
09         app:showAsAction="never" /> D
10     <item
11         android:id="@+id/action_reg" AJ
12         android:orderInCategory="100"
13         android:title="@string/action_reg"
14         app:showAsAction="never" /> E
15 </menu>
```

“홈” 메뉴 아이템

“사진 등록” 메뉴 아이템

3 액티비티 제어

데이터베이스에 등록되어 있는 사진 정보들을 모두 추출하여 각 사진 아이템별로 리니어 레이아웃에 동적으로 추가한다. 각 아이템을 클릭하면 사진정보를 조회하는 액티비티(ContentItem)로 이동한다.

```
소스 | ContentsList.java → AK
01 package com.example.yschang.contentsmgmt;
02
03 import android.content.Intent;
04 import android.database.Cursor;
05 import android.database.sqlite.SQLiteDatabase;
06 import android.database.sqlite.SQLiteException;
07 import android.graphics.Bitmap;
08 import android.graphics.BitmapFactory;
09 import android.graphics.Color;
10 import android.graphics.drawable.GradientDrawable;
11 import android.net.Uri;
12 import android.os.Bundle;
13 import android.support.v7.app.AppCompatActivity;
14 import android.support.v7.widget.Toolbar;
```

```

15 import android.view.Menu;
16 import android.view.MenuItem;
17 import android.view.View;
18 import android.view.View.OnClickListener;
19 import android.widget.ImageView;
20 import android.widget.LinearLayout;
21 import android.widget.TextView;
22 import android.widget.Toast;
23
24 import java.io.File;
25 import java.io.FileNotFoundException;
26
27 public class ContentsList extends AppCompatActivity implements
                                OnClickListener {
28
29     SQLiteDatabase sqlitedb;
30     DBManager dbmanager;
31
32     @Override
33     protected void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
35         setContentView(R.layout.activity_list);
36         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
37         setSupportActionBar(toolbar);
38
39         LinearLayout layout = (LinearLayout) findViewById(R.id.photos);
40
41         try {
42             dbmanager = new DBManager(this);
43             sqlitedb = dbmanager.getReadableDatabase();
44             Cursor cursor = sqlitedb.query(
45                 "photos", null, null, null, null, null, "title");
46
47             int i = 0;
48             while(cursor.moveToNext()) {
49                 String str_title = cursor.getString(
50                     cursor.getColumnIndex("title"));
51                 String str_orientation = cursor.getString(
52                     cursor.getColumnIndex("orientation"));
53                 String str_background = cursor.getString(
54                     cursor.getColumnIndex("background"));
55                 String str_path = cursor.getString(
56                     cursor.getColumnIndex("path"));
57
58                 LinearLayout layout_list = new LinearLayout(this);

```

id가 photos인 뷰를 인식하여 리니어 레이아웃 객체 생성(layout)

AD

AE

H

데이터베이스에 데이터를 조회하기 위한 SQLiteDatabase 객체 생성

photos 테이블에서 모든 사진 정보 추출

사진 제목, 방향, 배경, 경로 추출

추출한 모든 사진정보를 하나씩 사진목록에 추가

사진 이미지와 사진정보를 담은 리니어 레이아웃(layout_list) 생성

```

54 layout_list.setOrientation(LinearLayout.HORIZONTAL);
55 layout_list.setPadding(20, 10, 20, 10);
56 layout_list.setId(i);
57 layout_list.setTag(str_title);
58
59 ImageView iv_photo = new ImageView(this);
60 Uri uriFromPath = Uri.fromFile(new File(str_path));
61 Bitmap bitmap = null;
62 try {
63     bitmap = BitmapFactory.decodeStream(
64         getContentResolver().openInputStream(uriFromPath));
65     Bitmap thumb = Bitmap.createScaledBitmap(
66         bitmap, 200, 200, true);
67     iv_photo.setImageBitmap(thumb);
68     layout_list.addView(iv_photo);
69 } catch (FileNotFoundException e) {
70     Toast.makeText(this, "에러: " + e.getMessage(),
71         Toast.LENGTH_LONG).show();
72 }
73
74 int strokeWidth = 5;
75 int roundRadius = 15;
76 int strokeColor = Color.parseColor("#2E3135");
77 int fillColor = Color.parseColor("#DFDFE0");
78 GradientDrawable gd = new GradientDrawable();
79 gd.setColor(fillColor);
80 gd.setCornerRadius(roundRadius);
81 gd.setStroke(strokeWidth, strokeColor);
82 layout_list.setBackground(gd);
83
84 LinearLayout layout_item = new LinearLayout(this);
85 layout_item.setOrientation(LinearLayout.VERTICAL);
86 layout_item.setPadding(20, 10, 20, 10);
87
88 TextView tv_list = new TextView(this);
89 tv_list.setText(str_title);
90 tv_list.setTextSize(20);
91 tv_list.setPadding(5, 5, 5, 5);
92 layout_item.addView(tv_list);
93
94 TextView tv_list2 = new TextView(this);
95 tv_list2.setText("방향: " + str_orientation + "\n");
96 tv_list2.append("배경: " + str_background);
97 layout_item.addView(tv_list2);

```

layout_list에 사진 이미지 추가

이미지뷰에 썸네일 이미지 출력

이미지뷰를 layout_list에 추가

layout_list의 출력모양 설정

사진정보 배치 (layout_item)

사진제목 추가

사진방향과 배경 추가

```

96
97     layout_list.addView(layout_item);
98
99     layout.addView(layout_list);
100     layout_list.setOnClickListener(this);
101
102     i++;
103 }
104
105 cursor.close();
106 sqllitedb.close();
107 dbmanager.close();
108
109 } catch(SQLiteException e) {
110     Toast.makeText(this, e.getMessage(), Toast.LENGTH_LONG).show();
111 }
112 }
113
114 @Override
115 public void onClick(View view) {
116     int id = view.getId();
117
118     LinearLayout layout =(LinearLayout)findViewById(id);
119     String str_title =(String)layout.getTag();
120
121     Intent it = new Intent(getApplicationContext(),
122                                     ContentsItem.class);
123     it.putExtra("it_title", str_title);
124     startActivity(it);
125     finish();
126 }
127
128 @Override
129 public boolean onCreateOptionsMenu(Menu menu) {
130     getMenuInflater().inflate(R.menu.menu_list, menu);
131     return true;
132 }
133
134 @Override
135 public boolean onOptionsItemSelected(MenuItem item) {
136     int id = item.getItemId();
137
138     if(id == R.id.action_home) {
139         Intent it = new Intent(this, MainActivity.class);
140         startActivity(it);
141     }
142 }

```

사진정보(layout_item)를 layout_list에 추가

layout_list를 사진목록(layout)에 추가

layout_list에 클릭 리스너 설정

클릭 시 onClick() 콜백 메소드 호출

클릭한 뷰에서 사진제목 추출하여 ContentsItem 클래스 호출

AH

선택된 메뉴 아이템의 id 인식

초기화면으로 이동

```

140     finish();
141     return true;
142 }
143
144     if(id == R.id.action_reg) {
145         Intent it = new Intent(this, ContentsReg.class);
146         startActivity(it);
147         finish();
148         return true;
149     }
150
151     return super.onOptionsItemSelected(item);
152 }
153 }

```

“사진정보 등록” 화면으로 이동

동적 출력에 의한 리스트 아이템의 모양 출력에 대응하는 XML 설정

71~80행에 설정된 layout_list의 출력모양에 대응하는 XML 내용은 다음과 같다.

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <shape xmlns:android="http://schemas.android.com/apk/res/android"
03     android:shape="rectangle" >
04     <solid android:color="#DFDFE0" />
05     <corners
06         android:bottomLeftRadius="15dp"
07         android:bottomRightRadius="15dp"
08         android:topLeftRadius="15dp"
09         android:topRightRadius="15dp" />
10     <stroke
11         android:width="5dp"
12         android:color="#2E3135" />
13 </shape>

```

클래스와 속성/메소드

● 클래스

클래스/인터페이스	설명
Color	73,74행. 데이터베이스 조회에 의한 데이터셋을 읽고 쓰는 인터페이스를 제공함
File	60행. 경로에 있는 파일 시스템 개체의 추상화 표현
GradientDrawable	76행. 버튼, 배경 등에 대한 색의 그라데이션을 가진 드로워블

● XML 속성

클래스	메소드	설명
LinearLayout	int HORIZONTAL	54행. 리니어 레이아웃의 수평 방향을 의미하는 상수
	int VERTICAL	83행. 리니어 레이아웃의 수직 방향을 의미하는 상수

● 메소드

클래스	메소드	설명
Color	static int parseColor (String colorString)	73, 74행. 색의 이름에 해당하는 정수로 변환함
Context	abstract Context getApplicationContext ()	121행. 현재 프로세스의 어플리케이션 객체의 콘텍스트를 반환
GradientDrawable	Void setColor (int argb)	77행. 그라데이션 대신에 단일 색을 사용하기 위한 드로워블로 변경함
	Void setCornerRadius (float radius)	78행. 그라데이션의 코너를 위한 반경을 설정함
	Void setStroke (int width, int color)	79행. 선의 굵기와 색을 설정함
LinearLayout	void setOrientation (int orientation)	54, 83행. 레이아웃의 방향을 수평 또는 수직으로 설정함
TextView	Void setTextSize (float size)	88행. 문자의 디폴트 크기를 주어진 size 크기로 변경함
	void setPadding (int left, int top, int right, int bottom)	89행. 패딩을 설정함
Uri	Static Uri fromFile (File file)	60행. 파일에 해당하는 Uri를 만들
View	Void setId (int id)	56행. 뷰에 대한 ID를 설정함
	Void setBackground (Drawable background)	80행. 주어진 background로 배경을 설정함
	Void setOnClickListener (View.OnClickListener l)	100행. 뷰가 클릭될 때 불러질 수 있도록 콜백을 등록함
	Void setTag (Object tag)	57행. 뷰와 관련있는 태그를 설정함
ViewGroup	Void addView (View child)	66, 90, 95, 97, 99행. 차일드 뷰를 추가함

22.2.8 환경 설정과 실행

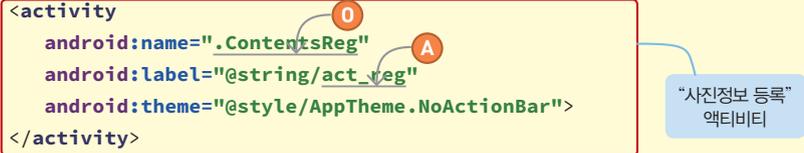
STEP 2 파일 편집

● 편집

환경설정에는 사진 등록과 카메라 사진 선택에 대한 허용을 설정하고, 새롭게 추가되는 액티비티클래스를 등록한다.

소스 | AndroidManifest.xml

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
03     package="com.example.ysechang.contentsmgt">
04
05     <uses-permission android:name="android.permission.READ_EXTERNAL_
                                STORAGE" />
06     <uses-permission android:name="android.permission.WRITE_EXTERNAL_
                                STORAGE" />
07     <uses-permission android:name="android.permission.CAMERA" />
08     <permission android:name="android.permission.FLASHLIGHT" />
09     <uses-feature android:name="android.hardware.camera"/>
10
11     <application
12         android:allowBackup="true"
13         android:icon="@mipmap/ic_launcher"
14         android:label="@string/app_name"
15         android:supportRtl="true"
16         android:theme="@style/AppTheme">
17         <activity
18             android:name=".MainActivity"
19             android:label="@string/app_name"
20             android:theme="@style/AppTheme.NoActionBar">
21             <intent-filter>
22                 <action android:name="android.intent.action.MAIN" />
23
24                 <category android:name="android.intent.category.LAUNCHER" />
25             </intent-filter>
26         </activity>
27         <activity
28             android:name=".ContentsReg"
29             android:label="@string/act_yeg"
30             android:theme="@style/AppTheme.NoActionBar">
31     </activity>
```



```

32 <activity
33     android:name=".ContentsList"
34     android:label="@string/act_list"
35     android:theme="@style/AppTheme.NoActionBar">
36 </activity>
37
38 <activity
39     android:name=".ContentsItem"
40     android:label="@string/act_item"
41     android:theme="@style/AppTheme.NoActionBar">
42 </activity>
43
44 </application>
45 </manifest>

```

AK → android:name=".ContentsList"
B → android:label="@string/act_list"
AB → android:name=".ContentsItem"
C → android:label="@string/act_item"

"사진목록" 액티비티
"사진정보" 액티비티

STEP 3 프로젝트 실행

다음 2단계 절차에 따라 프로젝트를 실행하고 그 결과를 살펴보자.

절차	내용
실행메뉴 선택	'Run' 메뉴에서 'Run app' 클릭(또는 'Run app' 아이콘 클릭)
디바이스 선택	스마트폰 디바이스를 선택하고 'OK' 버튼 클릭



21, 22장을 응용하면 다양한 정보를 체계적으로 잘 관리할 수 있다. 각자 이를 응용하여 개발해 보자.