



5장 : 안드로이드 실습교재 SQLite데이터 처리

옥 상 훈

Blog: <http://okgosu.tistory.com>

Web Site: <http://okgosu.net>

twitter @okgosu

SQLite를 활용한 상품조회



My Prod DB Manager (상품 관리자)

❑ 목표

- ❑ SQLite를 이용한 데이터 저장 및 조회

❑ 작성할 파일

- ❑ DB 스키마 상수: MyProdDBCons.java
- ❑ DB 연결관리자 : MyProdDBHelper.java
- ❑ DB 데이터 관리자 : MyProdDBManager.java
 - ❑ 레이아웃 : my_prod_db_manager.xml
- ❑ DB 검색 : MyProdDBList.java
 - ❑ 리스트뷰 컬럼 정의 : my_prod_col.xml



1단계 – 작업 파일 : MyProdDBCons.java

❑ 역할

- ❑ DB 스키마(DB, Table, Column 등)를 상수로 담아서 참조하기 위한 클래스

❑ 클래스 생성

- ❑ **import** android.provider.BaseColumns;
- ❑ MyProdDBCons implements BaseColumns



2단계 – 작업 파일 : MyProdDBCons.java

☐ 멤버 변수 추가

☐ DB 정보

- ☐ `public static final String DB_NAME = "my_prod_sqlite.db";`
- ☐ `public static final int DB_VERSION = 1;`

☐ TABLE 정보

- ☐ `public static final String TABLE_NAME = "prod";`

☐ COLUMN 정보

- ☐ `public static final String PROD_CODE = "prod_code";`
- ☐ `public static final String PROD_NAME = "prod_name";`



3단계 – 작업파일 : MyProdDBHelper.java

- ❑ 클래스 생성
 - ❑ MyProdDBHelper extends SQLiteOpenHelper
- ❑ 생성자 함수 정의
 - ❑ MyProdDBCons의 DB명, 버전을 생성자에서 호출

```
public MyProdDBHelper(Context context, String name, CursorFactory factory, int
version) {
super(context, MyProdDBCons.DB_NAME, factory, MyProdDBCons.DB_VERSION);
}
```



4단계 – 작업파일 : MyProdDBHelper.java

❑ SQLiteOpenHelper의 주요 함수 구현

❑ onCreate

@Override

```
public void onCreate(SQLiteDatabase arg0) {  
    arg0.execSQL("CREATE TABLE " + MyProdDBCons.TABLE_NAME + " (" +  
        + "_id INTEGER PRIMARY KEY AUTOINCREMENT, " +  
        + MyProdDBCons.PROD_CODE + " text, " +  
        + MyProdDBCons.PROD_NAME + " text not null);");  
}
```

❑ onUpgrade

@Override

```
public void onUpgrade(SQLiteDatabase arg0, int arg1, int arg2) {  
    // 이전 버전 DB 삭제하고 다시 생성  
    arg0.execSQL("DROP TABLE " + MyProdDBCons.TABLE_NAME);  
    onCreate(arg0);  
}
```



5단계 : 작업파일 – my_prod_db_manager.xml

□ xml 레이아웃 생성

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout android:orientation="horizontal"
    android:layout_width="fill_parent" android:layout_height="wrap_content">
        <EditText android:id="@+id/et_search" android:hint="검색어를 입력하세요"
            android:layout_height="wrap_content" android:layout_width="wrap_content"
            android:singleLine="true" android:layout_weight="1"/>
        <Button android:id="@+id/btn_query" android:text="DB 조회"
            android:layout_height="wrap_content" android:layout_width="wrap_content" />
    </LinearLayout>
    <Button android:id="@+id/btn_insert" android:text="데이터 입력"
    android:layout_height="wrap_content" android:layout_width="fill_parent" />
</LinearLayout>
```



6단계 – 작업파일 : MyProdDBManager.java

- ❑ 클래스 생성
 - ❑ MyProdDBManager **extends** Activity

- ❑ 멤버변수 추가
 - ❑ private MyProdDBHelper openHelper;
 - ❑ private SQLiteDatabase sdb;



7단계 – 작업파일 : MyProdDBManager.java

- ❑ onCreate 함수 구현
 - ❑ **super.onCreate**(savedInstanceState);
 - ❑ setContentView(R.layout.*my_prod_db_manager*);
 - ❑ **if** (checkDBCount()==0) initMyProdDB();



8단계 – 작업파일 : MyProdDBManager.java

❑ checkDBCount() 함수 구현

```
private int checkDBCount() {  
    // db 연결 준비  
    openHelper = new MyProdDBHelper(this, null, null, 0);  
    sdb = openHelper.getReadableDatabase();  
    // DB 조회  
    Cursor c = sdb.query(MyProdDBCons.TABLE_NAME, null, null, null, null, null, null);  
    startManagingCursor(c);  
    int count = c.getCount();  
    c.close();  
    sdb.close();  
    return count;  
}
```



9단계 – 작업파일 : MyProdDBManager.java

□ initMyProdDB() 구현

```
private void initMyProdDB() {  
    sdb = openHelper.getWritableDatabase();  
    // 컬럼별로 값 설정하는 map 객체  
    ContentValues values = new ContentValues();  
    String[] items = {"새우깡", "맛동산", "카라멜콘", "오징어땅콩", "꿀짱구" };  
    for (int i=0; i < items.length; i++) {  
        values.put(MyProdDBCons.PROD_CODE, "SNK" + i);  
        values.put(MyProdDBCons.PROD_NAME, items[i]);  
        sdb.insert(MyProdDBCons.TABLE_NAME, null, values);  
    }  
    sdb.close();  
}
```



10단계 – 작업파일 : MyProdDBManager.java

□ onCreate 함수에 검색 버튼 처리

```
// 검색버튼 & 검색어 이벤트 처리
final EditText searchEt =
    (EditText)findViewById(R.id.et_search);
    Button findBtn =
    (Button)findViewById(R.id.btn_query);
    findBtn.setOnClickListener(new OnClickListener() {
        public void onClick(View arg0) {
            Intent intent = new
            Intent("okgosu.net.MyProdDBList");
            intent.putExtra("searchCode",
            searchEt.getText().toString());
            startActivity(intent);
        }
    });
```



11단계 : 작업파일 : my_prod_db_col.xml

□ 리스트뷰의 컬럼 정의

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/my_prod_code"
        android:layout_height="wrap_content"
        android:layout_width="50sp" />
    <TextView
        android:id="@+id/my_prod_name"
        android:padding="5px"
        android:layout_height="wrap_content"
        android:layout_width="fill_parent" />
</LinearLayout>
```



12단계 : 작업파일 – MyProdDBList.java

- ❑ MyProdDBList 클래스
 - ❑ **ListActivity** 확장
 - ❑ 멤버 변수
 - ❑ private MyProdDBHelper openHelper;
 - ❑ private SQLiteDatabase sdb;
 - ❑ private SimpleCursorAdapter myAdapter;
 - ❑ private String searchCode = null;



13단계: 작업파일 – MyProdDBList.java

- ❑ onCreate 오버라이딩
 - ❑ super.onCreate(savedInstanceState);
 - ❑ // Intent parameter 가져오기
 - ❑ Intent i = getIntent();
 - ❑ searchCode = i.getStringExtra("searchCode");
 - ❑ **if**(searchCode==**null**) searchCode = "";



14단계: 작업파일 – MyProdDBList.java

- ❑ onResume 오버라이딩
 - ❑ // DB 연결 준비
 - ❑ openHelper = **new** MyProdDBHelper(**this**,
MyProdDBCons.*DB_NAME*, **null**,
MyProdDBCons.*DB_VERSION*);
 - ❑ sdb = openHelper.getReadableDatabase();

- ❑ // 반환할 컬럼 항목, _id 컬럼 정의
- ❑ String columns[] = {MyProdDBCons.*_ID*,
MyProdDBCons.*PROD_CODE*, MyProdDBCons.*PROD_NAME*};
- ❑ String order_by = MyProdDBCons.*PROD_NAME*;
- ❑ String qWhere = MyProdDBCons.*PROD_NAME* + " like ?";
- ❑ String qLike = "%" + searchCode + "%";



15단계: 작업파일 – MyProdDBList.java

- ❑ onResume 오버라이딩 (이어서)
 - ❑ // 커서 생성
 - ❑ Cursor c = sdb.query(MyProdDBCons.*TABLE_NAME*, columns, qWhere, **new** String[] {qLike}, **null**, **null**, order_by, **null**);
 - ❑ startManagingCursor(c);

- ❑ // SimpleCursorAdapter 연결
- ❑ String[] FROM = **new** String[]{MyProdDBCons.*PROD_CODE*, MyProdDBCons.*PROD_NAME*};
- ❑ **int**[] TO = **new int**[] {R.id.*my_prod_code*, R.id.*my_prod_name*};
- ❑ myAdapter = **new** SimpleCursorAdapter(**this**, R.layout.*my_prod_db_col*, c, FROM, TO);
- ❑ setListAdapter(myAdapter);



16단계: 작업파일 – AndroidManifest.xml

□ MyProdDBManager, MyProdDBList 액티비티 추가

```
<activity android:name=".MyProdDBManager" android:label="MyProdDBManager">
    <intent-filter> <action android:name="okgosu.net.MyProdDBManager" /> <category
android:name="android.intent.category.DEFAULT" /> </intent-filter>
</activity>
<activity android:name=".MyProdDBList" android:label="MyProdDBList">
    <intent-filter> <action android:name="okgosu.net.MyProdDBList" /> <category
android:name="android.intent.category.DEFAULT" /> </intent-filter>
</activity>
```



SQLite를 활용한 상품입력



1단계: 작업파일 – MyProdDBManager.java

□ onCreate 함수에 btn_insert 버튼 클릭리스너 처리

```
Button insertBtn = (Button)findViewById(R.id.btn_insert);
insertBtn.setOnClickListener(new OnClickListener() {
    public void onClick(View arg0) {
        Intent intent = new Intent("okgosu.net.MyProdDBInsert");
        startActivity(intent);
    }
});
```



2단계: 작업파일 – my_prod_db_insert.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:stretchColumns="*"
    android:padding="10dp"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TableRow android:layout_height="fill_parent">
        <TextView android:text="상품코드값" android:layout_height="wrap_content"
            android:layout_width="wrap_content" />
        <EditText android:id="@+id/et_prod_code" android:hint="상품코드값을 입력하세요"
            android:layout_height="wrap_content" android:layout_width="fill_parent"
            android:singleLine="true" android:layout_span="2" android:layout_weight="1"/>
    </TableRow>
    <TableRow android:layout_height="fill_parent">
        <TextView android:text="상품명" android:layout_height="wrap_content"
            android:layout_width="wrap_content" />
        <EditText android:id="@+id/et_prod_name" android:hint="상품명을 입력하세요"
            android:layout_height="wrap_content" android:layout_width="fill_parent"
            android:singleLine="true" android:layout_span="2" android:layout_weight="1"/>
    </TableRow>
    <TableRow android:layout_weight="1" android:layout_height="fill_parent">
        <Button android:id="@+id/btn_del" android:text="삭제" android:layout_height="wrap_content"
            android:visibility="invisible"
            android:layout_column="1" android:layout_width="fill_parent" />
        <Button android:id="@+id/btn_ok" android:text="확인" android:layout_height="wrap_content"
            android:layout_column="2" android:layout_width="fill_parent" />
    </TableRow>
</TableLayout>
```



3단계: 작업파일 – MyProdDBInsert.java

- ❑ 클래스 생성
 - ❑ MyProdDBInsert **extends** Activity
- ❑ 콘텐츠프로바이더 1단계 수행
 - ❑ MyProdDBCons.CONTENT_TYPE 설정
- ❑ onCreate 함수에 btn_ok 버튼 클릭리스너 처리

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.my_prod_db_insert);  
    final EditText prod_code = (EditText)findViewById(R.id.et_prod_code);  
    final EditText prod_name = (EditText)findViewById(R.id.et_prod_name);  
    Button btn_ok = (Button)findViewById(R.id.btn_ok);  
    btn_ok.setOnClickListener(new OnClickListener() {  
        public void onClick(View arg0) {  
            Intent intent = new Intent(Intent.ACTION_INSERT);  
            intent.putExtra(MyProdDBCons.PROD_CODE, prod_code.getText().toString());  
            intent.putExtra(MyProdDBCons.PROD_NAME,  
                prod_name.getText().toString());  
            intent.setType(MyProdDBCons.CONTENT_TYPE);  
            startActivity(intent);  
        }  
    });  
}
```



4단계: 작업파일 – MyProdDBManager.java

□ onCreate() 아래에 추가

```
// Intent parameter 가져오기
Intent intent = getIntent();
final String action = intent.getAction();
String prod_code = intent.getStringExtra(MyProdDBCons.PROD_CODE);
String prod_name = intent.getStringExtra(MyProdDBCons.PROD_NAME);
ContentValues values = new ContentValues();
values.put(MyProdDBCons.PROD_CODE, prod_code);
values.put(MyProdDBCons.PROD_NAME, prod_name);
if (Intent.ACTION_INSERT.equals(action)) {
    sdb = openHelper.getWritableDatabase();
    // 컬럼별로 값 설정하는 map 객체
    sdb.insert(MyProdDBCons.TABLE_NAME, null, values);
    sdb.close();
    Toast.makeText(getApplicationContext(), "입력되었습니다",
    Toast.LENGTH_SHORT).show();
}
```



5단계: 작업파일 – AndroidManifest.xml

□ MyProdDBInsert 액티비티 추가

```
<activity android:name=".MyProdDBInsert" android:label="MyProdDBInsert">  
  <intent-filter> <action android:name="okgosu.net.MyProdDBInsert" /> <category  
    android:name="android.intent.category.DEFAULT" /> </intent-filter>  
</activity>
```



6단계: 작업파일 – AndroidManifest.xml

❑ MyProdDBManager 액티비티에 대한 인텐트 필터 추가

```
<intent-filter>  
    <action android:name="android.intent.action.INSERT" />  
    <category android:name="android.intent.category.DEFAULT" />  
    <data android:mimeType="vnd.android.cursor.dir/vnd.okgosu.net.MyProd" />  
</intent-filter>
```



SQLite를 활용한 상품수정, 삭제



1단계: 작업파일 – MyProdList.java

□ 리스트 선택 이벤트 리스너 추가

@Override

```
protected void onListItemClick(ListView l, View v, int position, long id) {  
    TextView my_prod_code = (TextView) v.findViewById(R.id.my_prod_code);  
    TextView my_prod_name = (TextView) v.findViewById(R.id.my_prod_name);
```

```
    Intent intent = new Intent(Intent.ACTION_PICK);  
    intent.putExtra(MyProdDBCons.PROD_CODE, my_prod_code.getText());  
    intent.putExtra(MyProdDBCons.PROD_NAME, my_prod_name.getText());  
    intent.setType(MyProdDBCons.CONTENT_TYPE);  
    startActivity(intent);  
}
```



2단계: 작업파일 – MyProdDBItem.java

- ❑ 클래스 생성
 - ❑ MyProdDBItem **extends** Activity
- ❑ onCreate 함수 오버라이딩

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.my_prod_db_insert);  
    final EditText prod_code = (EditText)findViewById(R.id.et_prod_code);  
    final EditText prod_name = (EditText)findViewById(R.id.et_prod_name);  
  
    Intent intent = getIntent();  
    final String action = intent.getAction();  
    if (Intent.ACTION_PICK.equals(action)) {  
        String prod_code_t = intent.getStringExtra(MyProdDBCons.PROD_CODE);  
        String prod_name_t = intent.getStringExtra(MyProdDBCons.PROD_NAME);  
        prod_code.setText(prod_code_t);  
        prod_name.setText(prod_name_t);  
    }  
}
```



3단계: 작업파일 – AndroidManifest.xml

□ MyProdDBItem 액티비티 추가

```
<activity android:name= ".MyProdDBItem" android:label= "MyProdDBItem">  
    <intent-filter> <action android:name= "okgosu.net.MyProdDBItem" /> <category  
android:name= "android.intent.category.DEFAULT" /> </intent-filter>  
    <intent-filter>  
        <action android:name= "android.intent.action.PICK" />  
        <category android:name= "android.intent.category.DEFAULT" />  
        <data android:mimeType= "vnd.android.cursor.dir/vnd.okgosu.net.MyProd" />  
    </intent-filter>  
</activity>
```



4단계: 작업파일 – MyProdDBItem.java

□ onCreate에 삭제, 수정(확인) 버튼 이벤트 리스너 처리

// 삭제

```
Button btn_del = (Button)findViewById(R.id.btn_del);
btn_del.setVisibility(View.VISIBLE);
btn_del.setOnClickListener(new OnClickListener() {
    public void onClick(View arg0) {
        Intent intent = new Intent(Intent.ACTION_DELETE);
        intent.putExtra(MyProdDBCons.PROD_CODE, prod_code.getText().toString());
        intent.putExtra(MyProdDBCons.PROD_NAME, prod_name.getText().toString());
        intent.setType(MyProdDBCons.CONTENT_TYPE);
        startActivity(intent);
    }
});
```

// 수정

```
Button btn_ok = (Button)findViewById(R.id.btn_ok);
btn_ok.setOnClickListener(new OnClickListener() {
    public void onClick(View arg0) {
        Intent intent = new Intent(Intent.ACTION_EDIT);
        intent.putExtra(MyProdDBCons.PROD_CODE, prod_code.getText().toString());
        intent.putExtra(MyProdDBCons.PROD_NAME, prod_name.getText().toString());
        intent.setType(MyProdDBCons.CONTENT_TYPE);
        startActivity(intent);
    }
});
```



5단계: 작업파일 – MyProdDBManager.java

□ onCreate() 아래에 다음 내용 추가

```
String whereClause = MyProdDBCons.PROD_CODE + "= ? ";
String[] whereArgs = {prod_code};
if (Intent.ACTION_EDIT.equals(action)) {
    sdb = openHelper.getWritableDatabase();
    sdb.update(MyProdDBCons.TABLE_NAME, values, whereClause, whereArgs);
    Toast.makeText(getApplicationContext(), "수정되었습니다", Toast.LENGTH_SHORT).show();
}
if (Intent.ACTION_DELETE.equals(action)) {
    sdb = openHelper.getWritableDatabase();
    sdb.delete(MyProdDBCons.TABLE_NAME, whereClause, whereArgs);
    Toast.makeText(getApplicationContext(), "삭제되었습니다", Toast.LENGTH_SHORT).show();
}
```



6단계: 작업파일 – AndroidManifest.xml

❑ MyProdDBManager 액티비티에 대한 인텐트 필터 추가

변경전

```
<intent-filter>
    <action android:name="android.intent.action.INSERT" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:mimeType="vnd.android.cursor.dir/vnd.okgosu.net.MyProd" />
</intent-filter>
```

변경후

```
<intent-filter>
    <action android:name="android.intent.action.INSERT" />
    <action android:name="android.intent.action.EDIT" />
    <action android:name="android.intent.action.DELETE" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:mimeType="vnd.android.cursor.dir/vnd.okgosu.net.MyProd" />
</intent-filter>
```



컨텐츠 프로바이더



1단계 : 작업파일 – MyProdDBCons.java

□ 콘텐츠프로바이더를 위한 상수 추가

```
public static final String AUTHORITY = "okgosu.net.provider.MyProd";  
public static final Uri CONTENT_URI = Uri.parse("content://" + AUTHORITY + "/myprods");  
public static final String CONTENT_TYPE = "vnd.android.cursor.dir/vnd.okgosu.net.MyProd";  
public static final String CONTENT_ITEM_TYPE  
= "vnd.android.cursor.item/vnd.okgosu.net.MyProd";
```



2단계 : 작업파일 – MyProdContentProvider.java

- ❑ MyProdContentProvider.java 정의
 - ❑ MyProdContentProvider **extends** ContentProvider

- ❑ ContentProvider 함수 확인
 - ❑ delete, getType, insert, onCreate, query, update

- ❑ 멤버변수 선언
 - ❑ **private** MyProdDBHelper mOpenHelper;
 - ❑ **private static** HashMap<String, String> *sNotesProjectionMap*;
 - ❑ **private static final** UriMatcher *sUriMatcher*;
 - ❑ **private static final int** *MYPRODS* = 1;
 - ❑ **private static final int** *MYPRODS_ID* = 2;



3단계 : 작업파일 – MyProdContentProvider.java

```
static {  
    sUriMatcher = new UriMatcher(UriMatcher.NO_MATCH);  
    sUriMatcher.addURI(MyProdDBCons.AUTHORITY, "myprods", MYPRODS);  
    sUriMatcher.addURI(MyProdDBCons.AUTHORITY, "myprods/#", MYPRODS_ID);  
  
    sNotesProjectionMap = new HashMap<String, String>();  
    sNotesProjectionMap.put(MyProdDBCons._ID, MyProdDBCons._ID);  
    sNotesProjectionMap.put(MyProdDBCons.PROD_CODE, MyProdDBCons.PROD_CODE);  
    sNotesProjectionMap.put(MyProdDBCons.PROD_NAME, MyProdDBCons.PROD_NAME);  
}
```



4단계 : 작업파일 – MyProdContentProvider.java

❑ onCreate 함수 구현

```
@Override
public boolean onCreate() {
    mOpenHelper = new MyProdDBHelper(getContext(), null, null, 0);
    return false;
}
```

❑ getType 함수 구현

```
@Override
public String getType(Uri uri) {
    switch (sUriMatcher.match(uri)) {
        case MYPRODS:
            return MyProdDBCons.CONTENT_TYPE;
        case MYPRODS_ID:
            return MyProdDBCons.CONTENT_ITEM_TYPE;
        default:
            throw new IllegalArgumentException("Unknown URI " + uri);
    }
}
```



5단계 : 작업파일 – MyProdContentProvider.java

□ query 함수 구현

```
@Override
public Cursor query(Uri uri, String[] projection, String selection,
String[] selectionArgs, String sortOrder) {
    SQLiteQueryBuilder qb = new SQLiteQueryBuilder();
    qb.setTables(MyProdDBCons.TABLE_NAME);
    switch (sUriMatcher.match(uri)) {
        case MYPRODS:
            qb.setProjectionMap(sNotesProjectionMap);
            break;

        case MYPRODS_ID:
            qb.setProjectionMap(sNotesProjectionMap);
            qb.appendWhere(MyProdDBCons.ID + "=" + uri.getPathSegments().get(1));
            break;
        default:
            throw new IllegalArgumentException("Unknown URI " + uri);
    }
    String order_by = MyProdDBCons.PROD_NAME;
    SQLiteDatabase db = mOpenHelper.getReadableDatabase();
    Cursor c = qb.query(db, projection, selection, selectionArgs, null, null, order_by);
    c.setNotificationUri(getContext().getContentResolver(), uri);
    return c;
}
```



6단계 : 작업파일 – MyProdContentProvider.java

insert

```
@Override
public Uri insert(Uri uri, ContentValues values_in) {
    if (sUriMatcher.match(uri) != MYPRODS) {
        throw new IllegalArgumentException("Unknown URI " + uri);
    }
    ContentValues values;
    if (values_in != null) {
        values = new ContentValues(values_in);
    } else {
        values = new ContentValues();
    }
    if (values.containsKey(MyProdDBCons.PROD_CODE) == false) {
        values.put(MyProdDBCons.PROD_CODE, "DEFAULT");
    }
    if (values.containsKey(MyProdDBCons.PROD_NAME) == false) {
        values.put(MyProdDBCons.PROD_NAME, "");
    }
    SQLiteDatabase db = mOpenHelper.getWritableDatabase();
    long rowId = db.insert(MyProdDBCons.TABLE_NAME, null, values);
    if (rowId > 0) {
        Uri myProdUri = ContentUris.withAppendedId(MyProdDBCons.CONTENT_URI, rowId);
        getContext().getContentResolver().notifyChange(myProdUri, null);
        return myProdUri;
    }
    throw new SQLException("Failed to insert row into " + uri);
}
```



7단계 : 작업파일 – MyProdContentProvider.java

❑ delete

```
@Override
public int delete(Uri uri, String where, String[] whereArgs) {
    SQLiteDatabase db = mOpenHelper.getWritableDatabase();
    int count;
    switch (sUriMatcher.match(uri)) {
        case MYPRODS:
            count = db.delete(MyProdDBCons.TABLE_NAME, where, whereArgs);
            break;
        case MYPRODS_ID:
            String prodid = uri.getPathSegments().get(1);
            count = db.delete(MyProdDBCons.TABLE_NAME, MyProdDBCons.ID + "=" + prodid
                + (!TextUtils.isEmpty(where) ? " AND (" + where + ')' : ""), whereArgs);
            break;
        default:
            throw new IllegalArgumentException("Unknown URI " + uri);
    }
    getContext().getContentResolver().notifyChange(uri, null);
    return count;
}
```



8단계 : 작업파일 – MyProdContentProvider.java

□ update

```
@Override
public int update(Uri uri, ContentValues values, String
where, String[] whereArgs) {
    SQLiteDatabase db = mOpenHelper.getWritableDatabase();
    int count;
    switch (sUriMatcher.match(uri)) {
        case MYPRODS:
            count = db.update(MyProdDBCons.TABLE_NAME,
values, where, whereArgs);
            break;
        case MYPRODS_ID:
            String prodid = uri.getPathSegments().get(1);
            count = db.update(MyProdDBCons.TABLE_NAME,
values, MyProdDBCons._ID + "=" + prodid
                + (!TextUtils.isEmpty(where) ? " AND (" +
where + ")'" : ""), whereArgs);
            break;
        default:
            throw new IllegalArgumentException("Unknown
URI " + uri);
    }
    getContext().getContentResolver().notifyChange(uri, null);
    return count;
}
```



9단계 : 작업파일 - AndroidManifest.xml

□ contentprovider로 등록 : application노드 밑에

```
<provider android:name= ".MyProdContentProvider"  
          android:authorities= "okgosu.net.provider.MyProd" />
```



10단계 : 작업파일 – my_cp_demo.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="4dip"
    android:gravity="center_horizontal"
    android:layout_width="fill_parent" android:layout_height="fill_parent">
    <Button android:id="@+id/call"
        android:layout_width="fill_parent"
android:layout_height="wrap_content"
        android:text="Content Provider 호출">
    </Button>
    <TextView android:id="@+id/callback"
        android:layout_width="fill_parent"
android:layout_height="wrap_content"
        android:layout_weight="0"
        android:gravity="center_horizontal" android:paddingTop="4dip"/>
    <ListView
android:id="@android:id/list" android:drawSelectorOnTop="false"
android:layout_width="fill_parent" android:layout_height="fill_parent"/>
</LinearLayout>
```



11단계 : 작업파일 – MyCPDemo.java

- ❑ 클래스 생성
 - ❑ MyCPDemo **extends** ListActivity

- ❑ 멤버변수
 - ❑ TextView mCallbackText;
 - ❑ Button call_btn;
 - ❑ **private static** String[] *MY_PROJECTION* = **new** String[] {
MyProdDBCons._ID, MyProdDBCons.PROD_CODE,
MyProdDBCons.PROD_NAME};



12단계 : 작업파일 – MyCPDemo.java

- ❑ onCreate 오버라이딩
 - ❑ **super.onCreate(savedInstanceState);**
 - ❑ **setContentView(R.layout.my_cp_demo);**
 - ❑ **mCallbackText = (TextView)findViewById(R.id.callback);**
 - ❑ **mCallbackText.setText("ContentProvider 예제");**
 - ❑ **call_btn = (Button)findViewById(R.id.call);**
 - ❑ **call_btn.setOnClickListener(mClickListener);**



13단계 : 작업파일 – MyCPDemo.java

❑ mCaIIIListener 구현

```
private OnClickListener mCaIIIListener = new OnClickListener() {  
    public void onClick(View v) {  
        Cursor c =  
getContentResolver().query(MyProdDBCons.CONTENT_URI  
        , MY_PROJECTION, null, null, null);  
        startManagingCursor(c);  
        ListAdapter adapter = new SimpleCursorAdapter(MyCPDemo.this,  
            android.R.layout.simple_list_item_2,  
            c,  
            new String[] {  
MyProdDBCons.PROD_CODE, MyProdDBCons.PROD_NAME},  
            new int[] {android.R.id.text1, android.R.id.text2});  
        setListAdapter(adapter);  
    }  
};
```



14단계:

❑ 매니페스트에 *MyProdDBList* 에 추가

```
<intent-filter>  
    <action android:name="android.intent.action.VIEW" />  
    <category android:name="android.intent.category.DEFAULT" />  
    <data android:mimeType="vnd.android.cursor.dir/vnd.okgosu.net.MyProd" />  
</intent-filter>
```

❑ *MyCPDemo.java*에서 인텐트 호출

```
Intent intent = new Intent(Intent.ACTION_VIEW,  
    Uri.parse("content://okgosu.net.provider.MyProd/myprods"));  
startActivity(intent);
```



수고하셨습니다 ^^/

