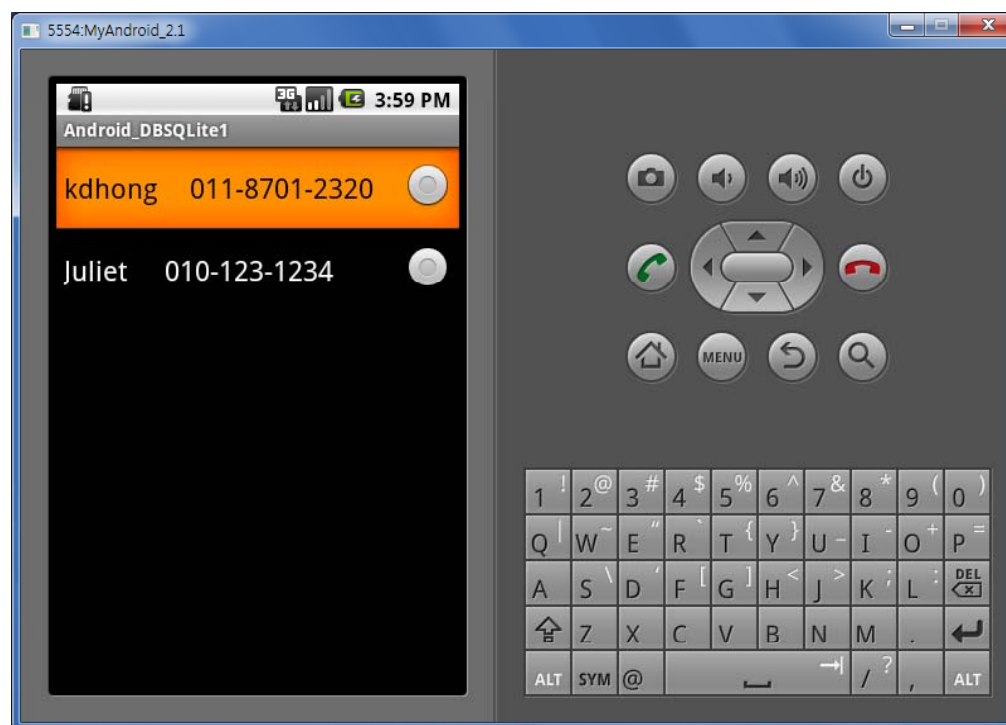




# DB 연동 방법





# 학습 목표

## 교육 목표

- ❖ Android DB 연동 방법
- ❖ Android DB 연동 구현
  - ◆ ListView
- ❖ Android DB 확인
- ❖ 실습 : Android DB 연동 구현 I
  - ◆ 데이터 추가
- ❖ 실습 : Android DB 연동 구현 II
  - ◆ TableRow



Navigation voice



Traffic view



Street View





# Android DataBase 연동 방법 (1)

## Database 객체

### ❖ Cursors

◆ 특정 record 지정

### ❖ ContentValues

◆ record 와 동일

```
1 package com.inha.android.GoogleMapView;
2
3 import android.os.Bundle;
4 import com.google.android.maps.MapActivity;
5
6 public class Android_GoogleMapView extends MapActivity {
7
8     /** Called when the activity is first created. */
9     @Override
10    public void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.main);
13    }
14
15    @Override
16    protected boolean isRouteDisplayed() {
17        return false;
18    }
19 }
20 }
```

Code

ContentValues

Cursors

No	Type	Displacement	Performance	Price
1	AvanteXD	1600	17	1500
2	Benz	3500	8	12000
3	GrandeurXG	2700	10	3500
4	SMS	2000	12	2500
5	Matiz	800	20	1000

Database





# Android DataBase 연동 방법 (2)

## ■ ContentValues 객체

❖ 기능

◆ 데이터 저장

index 0	index 1	index 2
_id	Name	Phone_No
1	kdhong	0101231234
2	Romio	0115670918
3	Juliet	0102340918

```
ContentValues newCValues = new ContentValues(); // 데이터 입력
```

```
newCValues.put("Name", "kdhong");
```

```
newCValues.put("Phone_No", "0101231234");
```

```
myDB.insert("members", null, newCValues); // 레코드 추가
```





# Android DataBase 연동 방법 (3)

## ■ Cursor 객체

### ❖ 기능

#### ◆ 데이터 반환

### ❖ Format

```
public Cursor query (String table, String[] columns, String selection,
String[] selectionArgs, String groupBy,
String having, String orderBy, String limit)
```

Argument	설명
<b>String</b> table	테이블 이름
<b>String[]</b> columns	반환 받을 필드 자료 (null : 모든 필드 반환)
<b>String</b> selection	SQL의 "where" 구문에 해당되는 조건 입력 (조건이 많을 경우, ?로 대체)
<b>String[]</b> selectionArgs	selection을 ?로 지정 : 조건 입력
<b>String</b> groupBy	SQL의 "group by" 구문
<b>String</b> having	groupBy 지정 : 조건 입력
<b>String</b> orderBy	결과값 정렬 방식 지정(null : 기본 정렬)
<b>String</b> limit	결과값의 수 제한





# Android DataBase 연동 방법 (4)

## ❖ Cursor ^사용 예

// 모든 레코드 반환

```
Cursor allRCD = myDB.query("members", null, null, null, null, null, null, null);
```

// Name = google 레코드 반환

```
Cursor selectRCD = myDB.query("members", "name = google", null, null, null, null, null, null);
```





# Android DataBase 연동 방법 (5)

## ❖ Cursor interface의 Method

Method	설 명
moveToFirst()	첫번째 레코드 지정
moveToNext()	다음 레코드로 커서 이동
moveToPrevious()	이전 레코드로 커서 이동
getCount()	질의 결과값(레코드)의 수 반환
getColumnIndexOrThrow()	특정 필드의 인덱스값을 반환하며, 필드가 존재하지 않을 경우 예외 발생
getColumnName()	특정 인덱스값에 해당하는 필드 명 반환
getColumnNames()	필드 명을 String 배열 형태로 반환
moveToPosition()	커서를 특정 레코드로 이동
getPosition()	커서가 현재 가리키고 있는 위치 반환





# Android DataBase 연동 방법 (6)

## ■ Cursor 적용 예

### ❖ Data 가져오기

◆ `get<datatype><index>`

index 0	index 1	index 2
_id	Name	Phone_No
1	kdhong	0101231234
2	Romio	0115670918
3	Juliet	0102340918

```
String strName = allRCD.getString(1); // Name 반환.
```

```
long lgPhoneNo = allRCD.getLong(2); // Phone_No 반환.
```



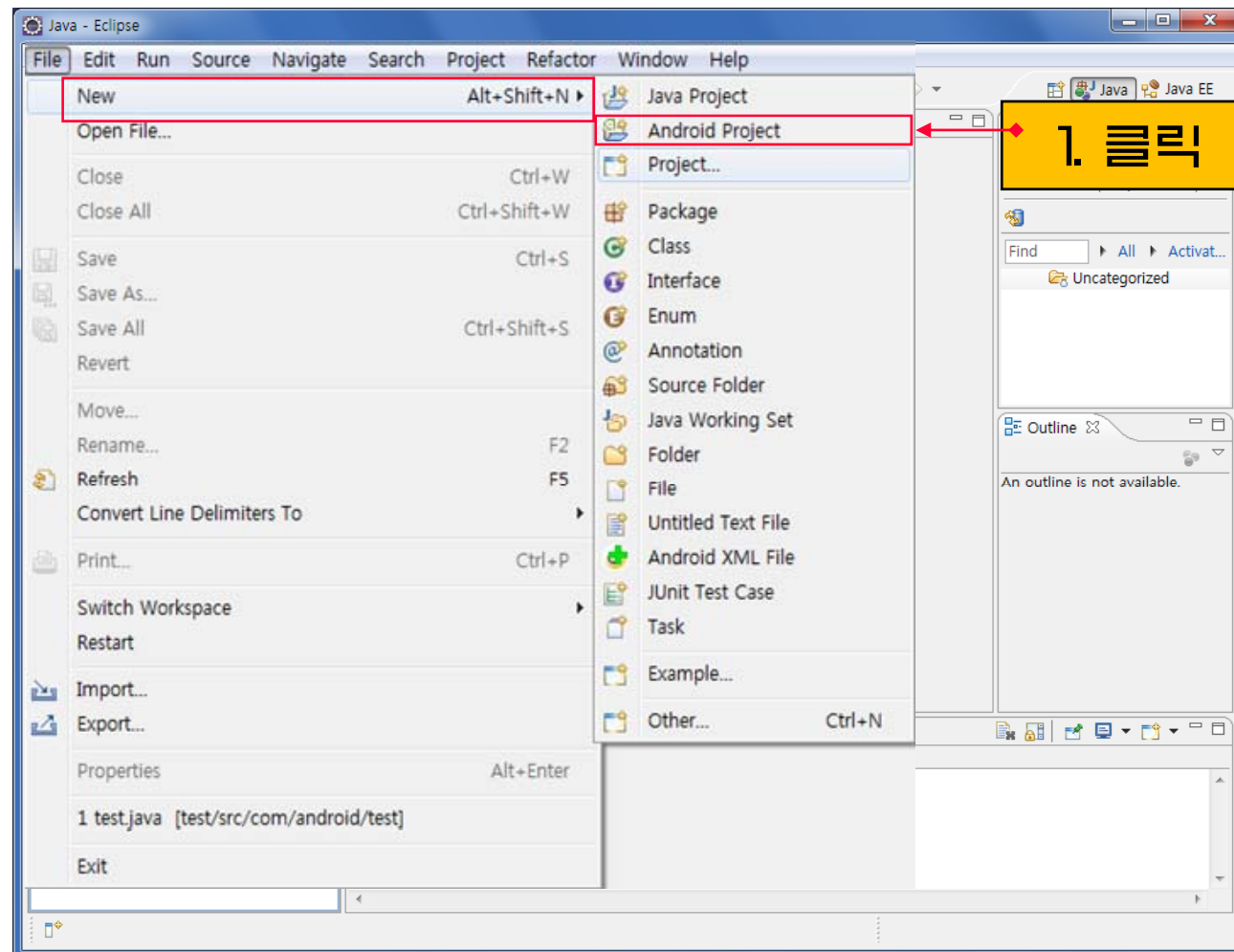




# Android DataBase 연동 구현 (1)

## ■ Android 프로젝트 생성

❖ 프로젝트 명 : Android\_DBSQLite1





# Android DataBase 연동 구현 (2)

New Android Project

Creates a new Android Project resource.

Project name:

Contents

☒ Create new project in workspace  
☐ Create project from existing source  
☒ Use default location

Location:

☐ Create project from existing sample

Samples:

Build Target

Target Name	Vendor	Platform	API ...
<input type="checkbox"/> Android 1.1	Android Open Source Project	1.1	2
<input type="checkbox"/> Android 2.0.1	Android Open Source Project	2.0.1	6
<input checked="" type="checkbox"/> Android 2.1	Android Open Source Project	2.1	7
<input type="checkbox"/> Google APIs	Google Inc.	2.0.1	6
<input type="checkbox"/> Google APIs	Google Inc.	2.1	7

Standard Android platform 2.1

Properties

Application name:

Package name:

☒ Create Activity:

Min SDK Version:

2. Android\_DBSQLite1 입력

3. 클릭

4. Android\_DBSQLite1 입력

5. com.inhatc.android\_DBSQLite1 입력

6. Android\_DBSQLite1 입력

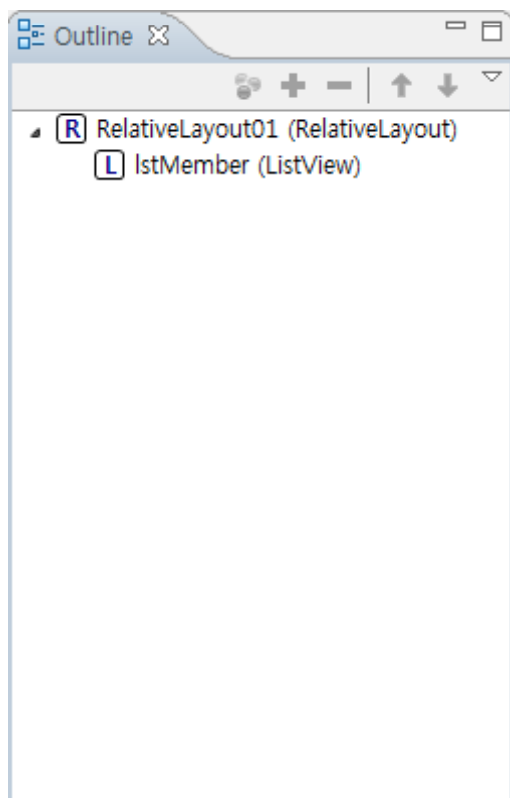
7. 클릭



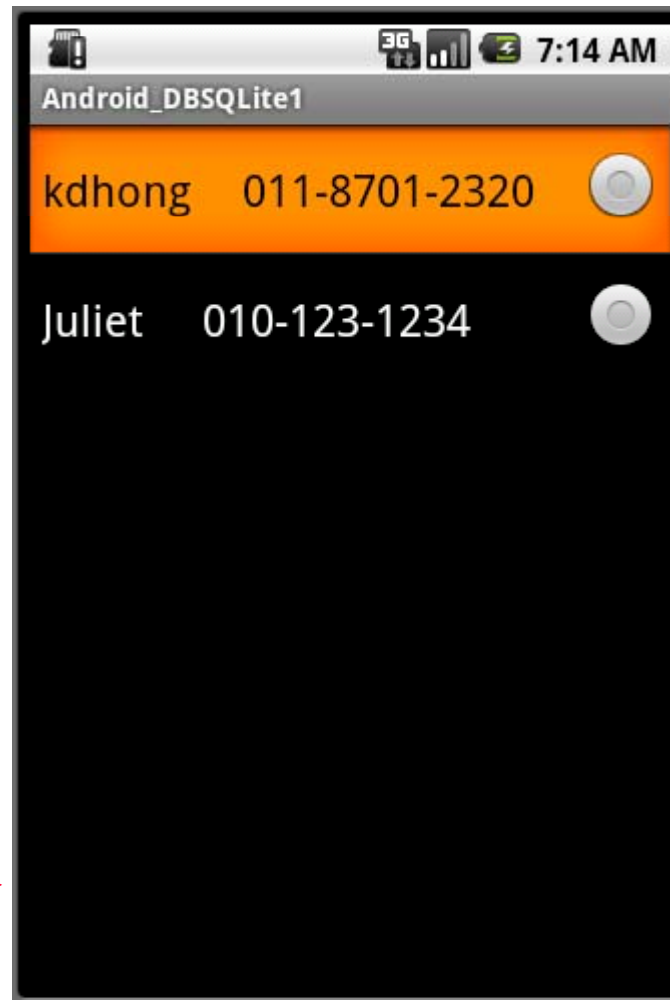


# Android DataBase 연동 구현 (3)

## ■ UI 설계



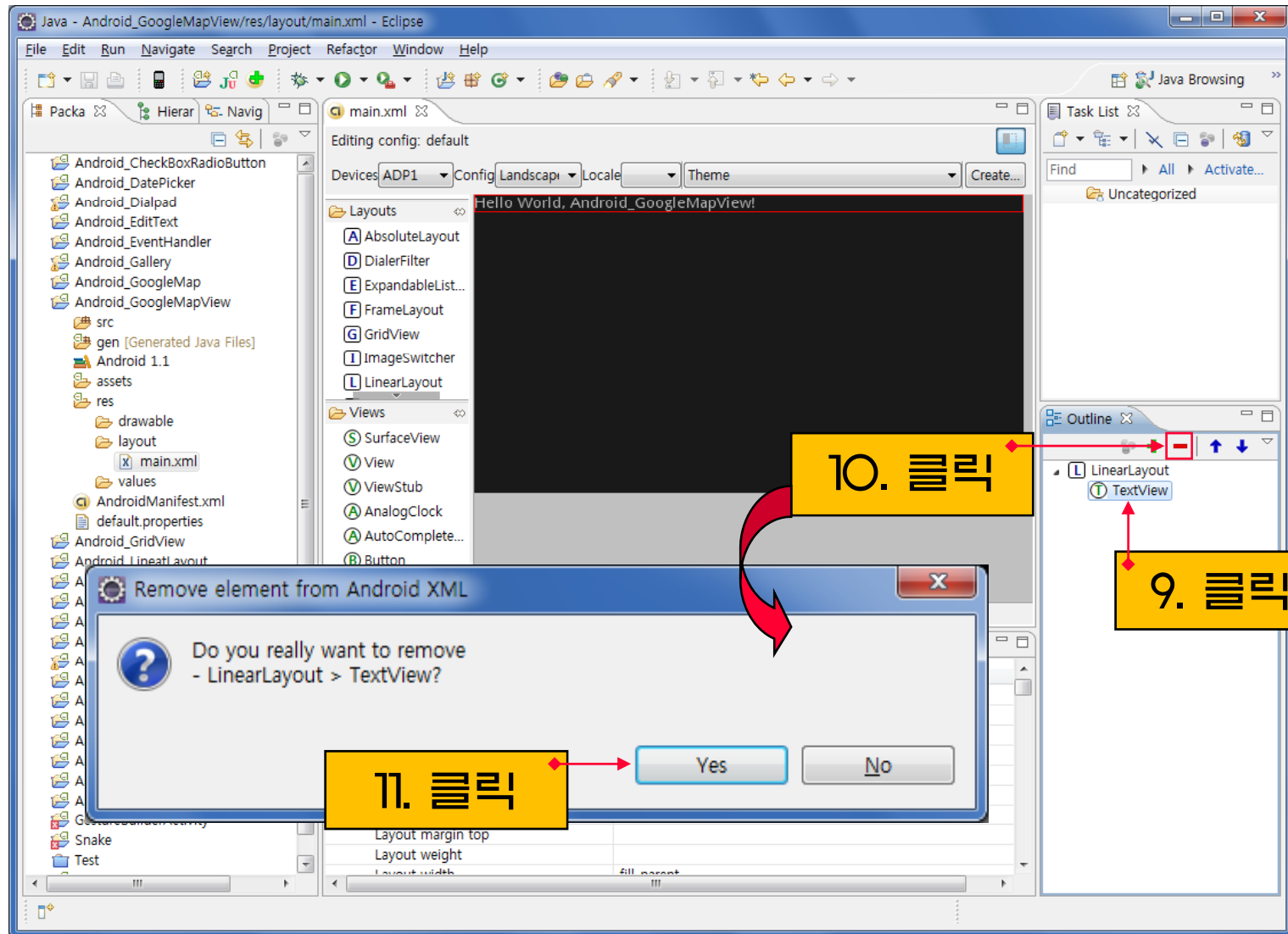
8. UI 설계 및 속성 지정





# Android DataBase 연동 구현 (4)

## ❖ LinearLayout / Text View 삭제





# Android DataBase 연동 구현 (5)

## ■ Main.xml 추가

❖ RelativeLayout 추가

❖ Listview 추가

```
*main.xml X
1<?xml version="1.0" encoding="utf-8"?>
2<RelativeLayout
3    android:id="@+id/RelativeLayout01"
4    android:layout_width="fill_parent"
5    android:layout_height="fill_parent"
6    xmlns:android="http://schemas.android.com/apk/res/android">
7    <ListView
8        android:layout_height="wrap_content"
9        android:id="@+id/lstMember"
10       android:layout_width="fill_parent">
11    </ListView>
12</RelativeLayout>
```

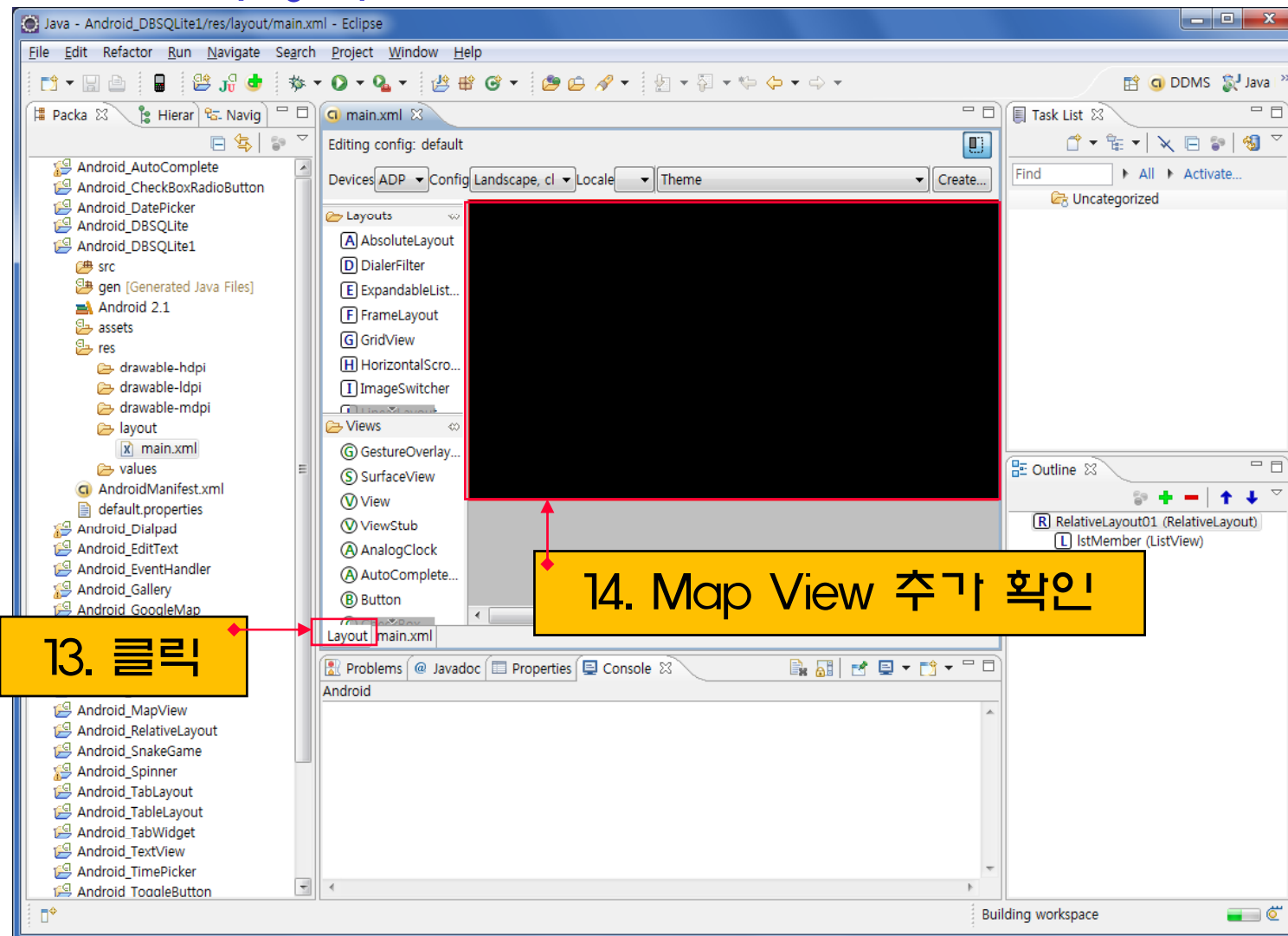
12. XML code 수정 확인





# Android DataBase 연동 구현 (6)

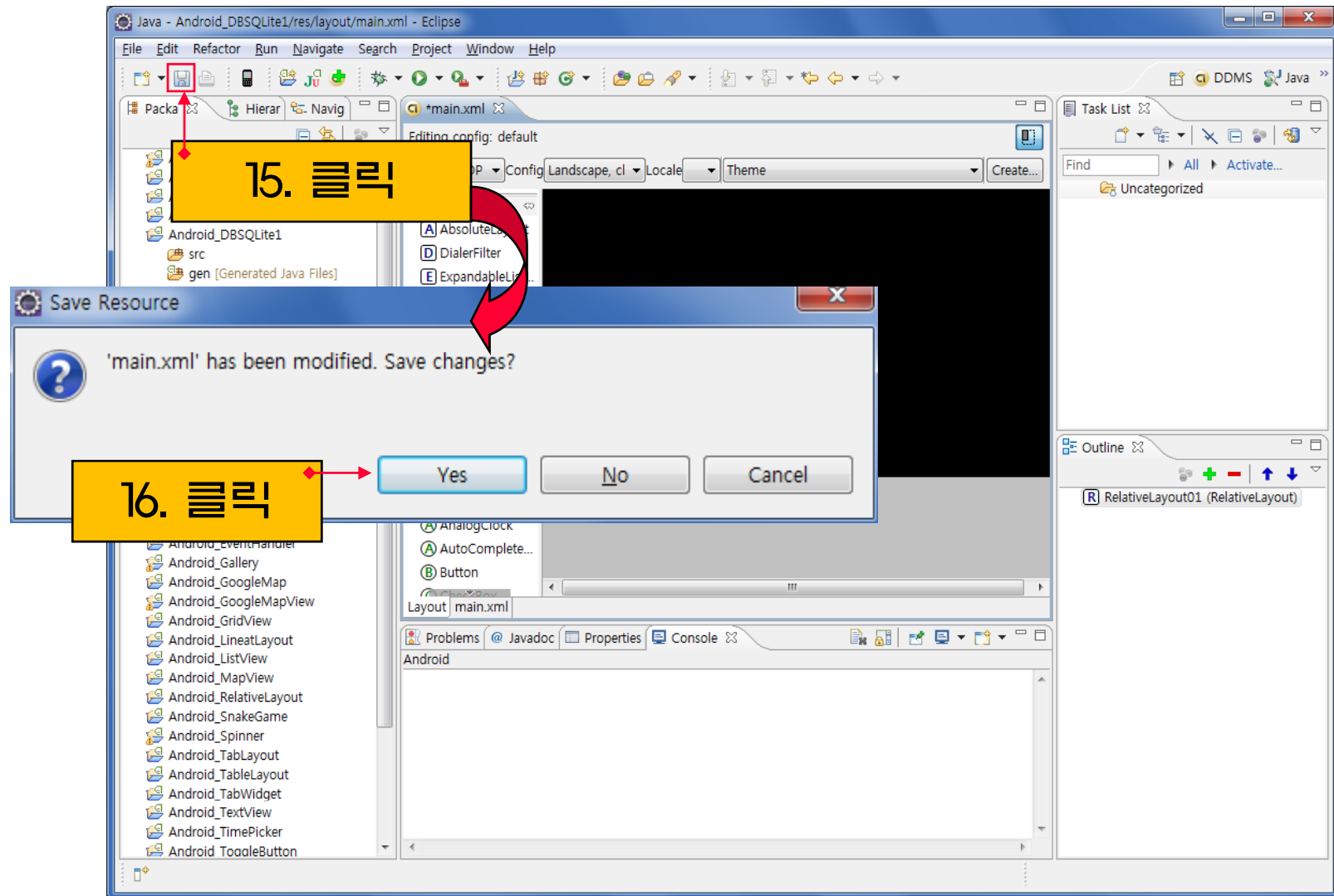
## ❖ Main.xml 수정 확인





# Android DataBase 연동 구현 (7)

## ❖ Main.xml 저장





# Android DataBase 연동 구현 (8)

## ❖ R.java

```
R.java X
1+/* AUTO-GENERATED FILE. DO NOT MODIFY.
7
8 package com.inhatc.Android_DBSQLite1;
9
10 public final class R {
11     public static final class attr {
12     }
13     public static final class drawable {
14         public static final int icon=0x7f020000;
15     }
16     public static final class id {
17         public static final int RelativeLayout01=0x7f050000;
18         public static final int 1stMember=0x7f050001;
19     }
20     public static final class layout {
21         public static final int main=0x7f030000;
22     }
23     public static final class string {
24         public static final int app_name=0x7f040001;
25         public static final int hello=0x7f040000;
26     }
27 }
```

17. id 클래스 변수 추가 확인







# Android DataBase 연동 구현 (9)

## ■ Android\_DBSQLite1.java

```
Android_DBSQLite1.java X
1 package com.inhatc.Android_DBSQLite1;
2
3 import java.util.ArrayList;
4 import android.app.Activity;
5 import android.content.ContentValues;
6 import android.database.Cursor;
7 import android.database.sqlite.SQLiteDatabase;
8 import android.os.Bundle;
9 import android.widget.AdapterView;
10 import android.widget.ListView;
11 import android.widget.SimpleAdapter;
12
13 public class Android_DBSQLite1 extends Activity {
14
15     SQLiteDatabase myDB;           //Database 객체 선언
16     SimpleAdapter myADT;           //Adapter 객체 선언
17     ArrayList<String> aryMBRList;  //ArrayList 객체 선언
18     ArrayAdapter<String> adtMembers; //ArrayAdapter 객체선언
19     ListView lstView;             //ListView 객체 선언
20
21     /** Called when the activity is first created. */
22     @Override
23     public void onCreate(Bundle savedInstanceState) {
24         String strRecord = null;   //Record data
25
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.main);
```



# Android DataBase 연동 구현 (10)

## ❖ DB 연동 Code

```
Android_DBSQLite1.java 38
28
29 //DB 생성 (DB 명 : PhoneBook)
30 myDB = this.openOrCreateDatabase("PhoneBook", MODE_PRIVATE, null);
31 myDB.execSQL("Drop table if exists members");
32
33 //Table 생성 (Table 명 : members)
34 myDB.execSQL("Create table members (" +
35     " _id integer primary key autoincrement, " +
36     " Name text not null, " + "Phone_No text not null);" );
37
38 //Data 저장 ("kdhong", "011-8701-2320")
39 myDB.execSQL("Insert into members " +
40     " (Name, Phone_No) values ('kdhong', '011-8701-2320');" );
41
42 //members 테이블에서 Data 저장
43 ContentValues insertValue = new ContentValues();
44 insertValue.put("Name", "Juliet");
45 insertValue.put("Phone_No", "010-123-1234");
46 myDB.insert("members", null, insertValue);
47
48 //members 테이블에서 모든 Record Data 가져오기
49 Cursor allRCD = myDB.query("members", null,
50     null, null, null, null, null);
51
```



# Android DataBase 연동 구현 (11)

## ❖ DB data → ListView

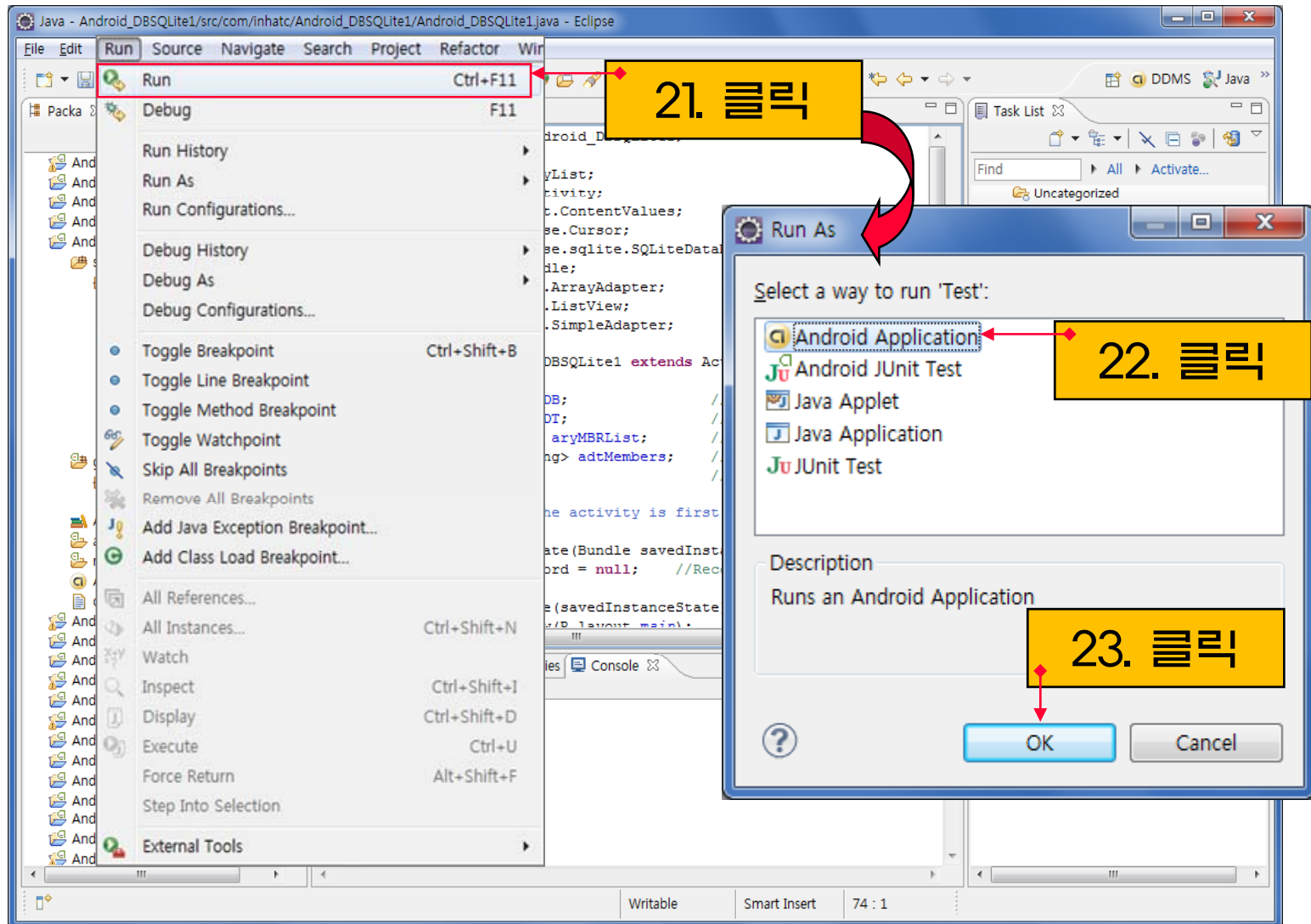
```
*Android_DBSQLite1.java X
51
52 //ArrayList 생성
53 aryMBRList = new ArrayList<String>();
54 if (allRCD != null){
55     if (allRCD.moveToFirst()) {
56         do{
57             strRecord = allRCD.getString(1)+"\t\t"+allRCD.getString(2);
58             aryMBRList.add(strRecord);
59         }while(allRCD.moveToNext());
60     }
61 }
62 adtMembers = new ArrayAdapter<String>(this,
63     android.R.layout.simple_list_item_single_choice, aryMBRList);
64
65 //ListView 생성
66 lstView = (ListView) findViewById(R.id.lstMember);
67 lstView.setAdapter(adtMembers);
68 lstView.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
69
70 if(myDB != null) myDB.close(); //DB 연결 해제
71 }
72 }
```

20. Coding



# Android DataBase 연동 구현 (12)

## ❖ Android 프로젝트 실행





# Android DataBase 연동 구현 (13)

## ❖ 실행 결과



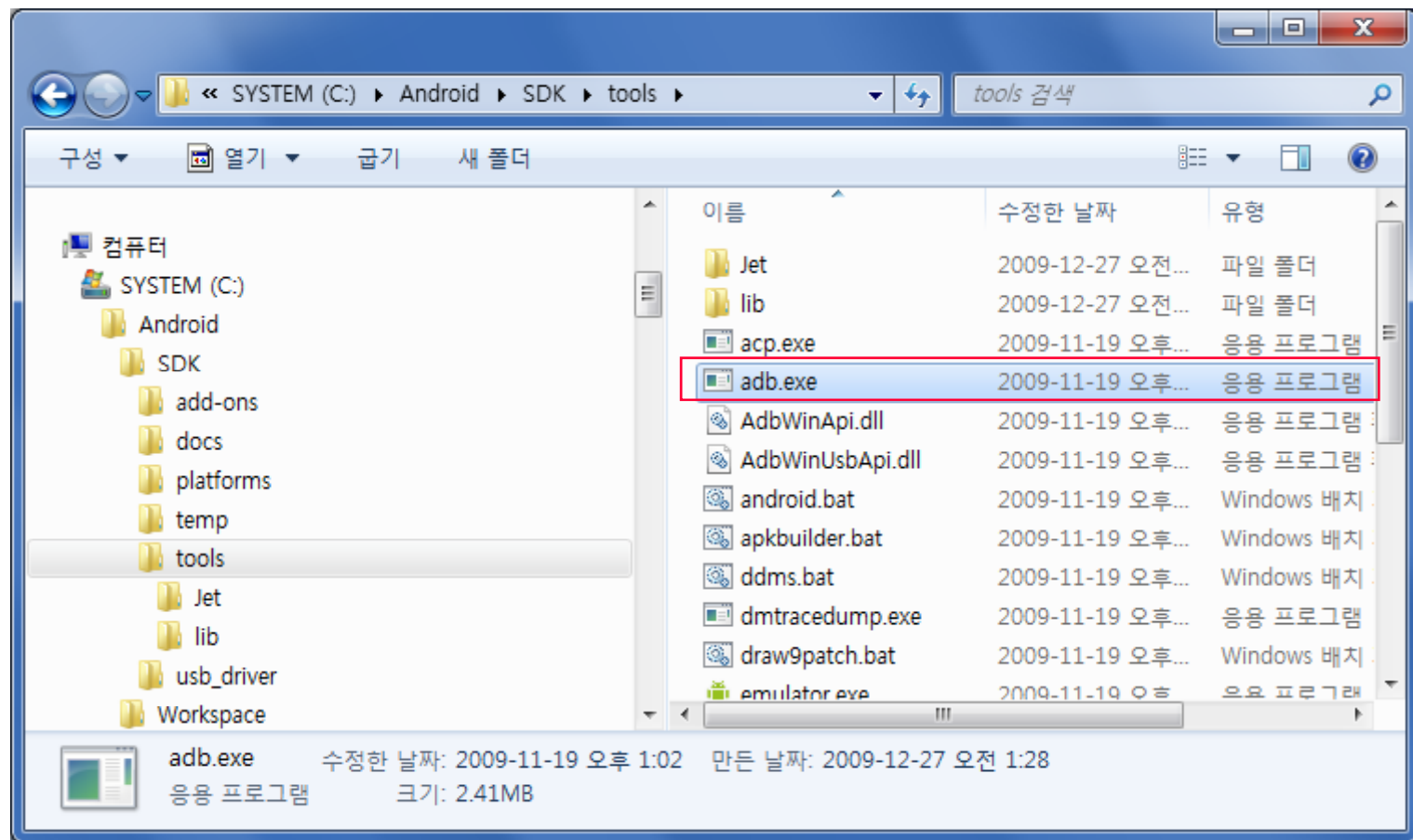


# Android DataBase 확인 (1)

## ■ DB 확인

❖ Android > SDK > tools

◆ adb.exe





# Android DataBase 확인 (2)

## ■ DB 확인 과정

- ① 명령 프롬프트(cmd) 실행
- ② cd c:\android\sdk\tools 폴더 이동
- ③ adb shell 명령어 실행
- ④ cd /data/data/com.inhtc.Android\_DBSQLite1 폴더 이동
- ⑤ cd ./databases 폴더 이동 (PhoneBook DB 파일 존재)
- ⑥ sqlite3 PhoneBook 실행
- ⑦ .table (members table 존재)
- ⑧ Select \* from members; (Query문 실행)
- ⑨ members table에 저장된 data 확인

```
select * from members;  
1:kdhong!011-8701-2320  
2:Juliet!010-123-1234
```







# Android DataBase 확인 (3)

❖ abd.exe 실행

```
명령 프롬프트 - adb shell
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Lee Won-Joo>cd c:\Android\SDK\tools
c:\Android\SDK\tools>adb shell
# ls
ls
sqlite_stmt_journals
config
cache
sdcard
d
etc
system
sys
sbin
proc
init.rc
init.goldfish.rc
init
default.prop
data
root
dev
#
```

1. 디렉토리(C:\Android\SDK\tools) 이동

2. adb shell 실행

3. ls 실행







The first screenshot shows the initial steps of navigating to the database directory:

```
# cd /data/data
cd /data/data
# ls
ls
```

The second screenshot continues from the previous state, listing the contents of the current directory and further navigating into the database folder:

```
com.android.providers.media
com.android.email
com.android.launcher
com.android.browser
my.HelloTabWidget
com.inhatc.android.TabWidget
com.inhatc.android.AutoComplete
com.inhatc.android.DBSQLite
com.inhatc.Android.DBSQLite1
com.inhatc.android.WebView
# cd ./com.inhatc.Android.DBSQLite1
cd ./com.inhatc.Android.DBSQLite1
/data/data/com.inhatc.Android.DBSQLite1
# ls
ls
lib
databases
# cd ./databases
cd ./databases
/data/data/com.inhatc.Android.DBSQLite1/databases
# ls
PhoneBook
```



# Android DataBase 확인 (5)

## ❖ DB data 확인

```
명령 프롬프트 - adb shell
# ls
ls
PhoneBook
# sqlite3 PhoneBook
sqlite3 PhoneBook
SQLite version 3.5.9
Enter ".help" for instructions
sqlite> .tables
.tables
android_metadata  members
sqlite> select * from members;
select * from members;
1!kdhong!011-8701-2320
2!Juliet!010-123-1234
sqlite>
```

10. sqlite3 PhoneBook 명령어 실행

11. .tables 명령어 실행 (table 보기)

12. select \* from members; (Query문 실행)

13. data 확인



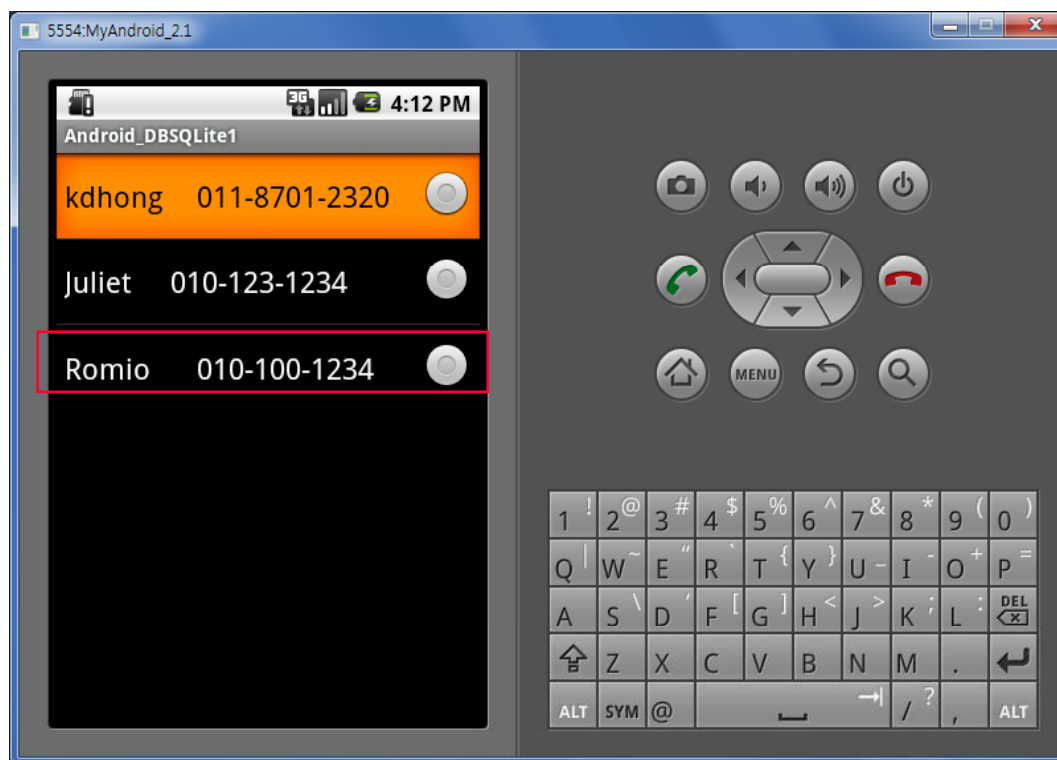


# 실습 I : DataBase 연동 구현

## ■ Android\_DBSQLite1 (실습 시간 : 20분)

❖ 아래 그림과 같이 member 테이블에 새로운 데이터를 저장하여 ListView에 출력되도록 구현하시오.

◆ Romio, 010-100-1234





# 실습 I : DataBase 연동 구현 (1)

## ❖ DB 연동 Code 추가

```
Android_DBSQLite1.java
28
29 //DB 생성 (DB 명 : PhoneBook)
30 myDB = this.openOrCreateDatabase("PhoneBook", MODE_PRIVATE, null);
31 myDB.execSQL("Drop table if exists members");
32
33 //Table 생성 (Table 명 : members)
34 myDB.execSQL("Create table members (" +
35     " _id integer primary key autoincrement, " +
36     "Name text not null, " + "Phone_No text not null);" );
37
38 //Data 저장 ("kdhong", "011-8701-2320")
39 myDB.execSQL("Insert into members " +
40     " (Name, Phone_No) values ('kdhong', '011-8701-2320');" );
41
42 //members 테이블에서 Data 저장
43 ContentValues insertValue = new ContentValues();
44 insertValue.put("Name", "Juliet");
45 insertValue.put("Phone_No", "010-123-1234");
46 myDB.insert("members", null, insertValue);
47
48 insertValue.put("Name", "Romio");
49 insertValue.put("Phone_No", "010-100-1234");
50 myDB.insert("members", null, insertValue);
51
52 //members 테이블에서 모든 Record Data 가져오기
53 Cursor allRCD = myDB.query("members", null,
54     null, null, null, null, null, null);
55
```

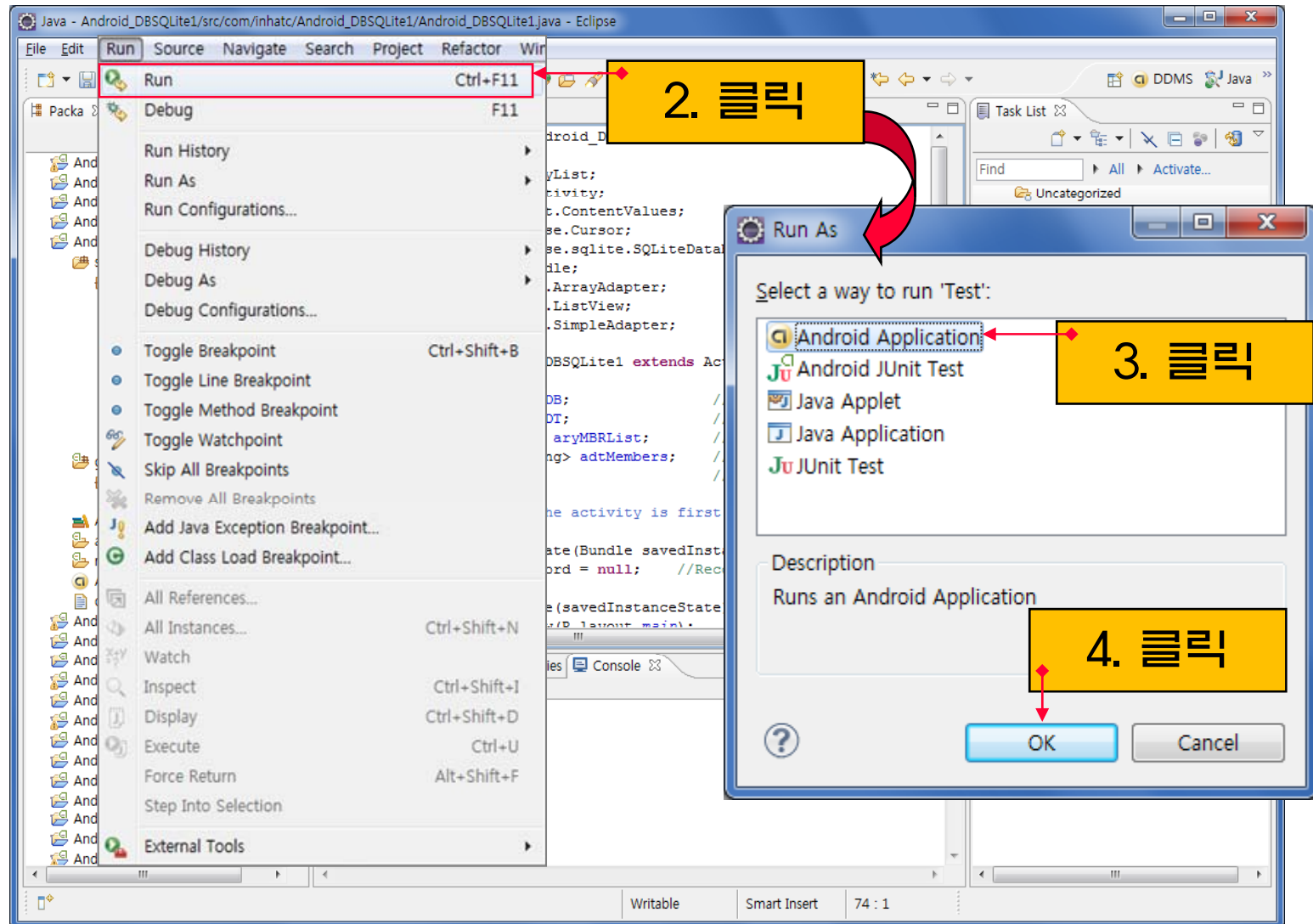
1. Coding





# 실습 I : DataBase 연동 구현 (2)

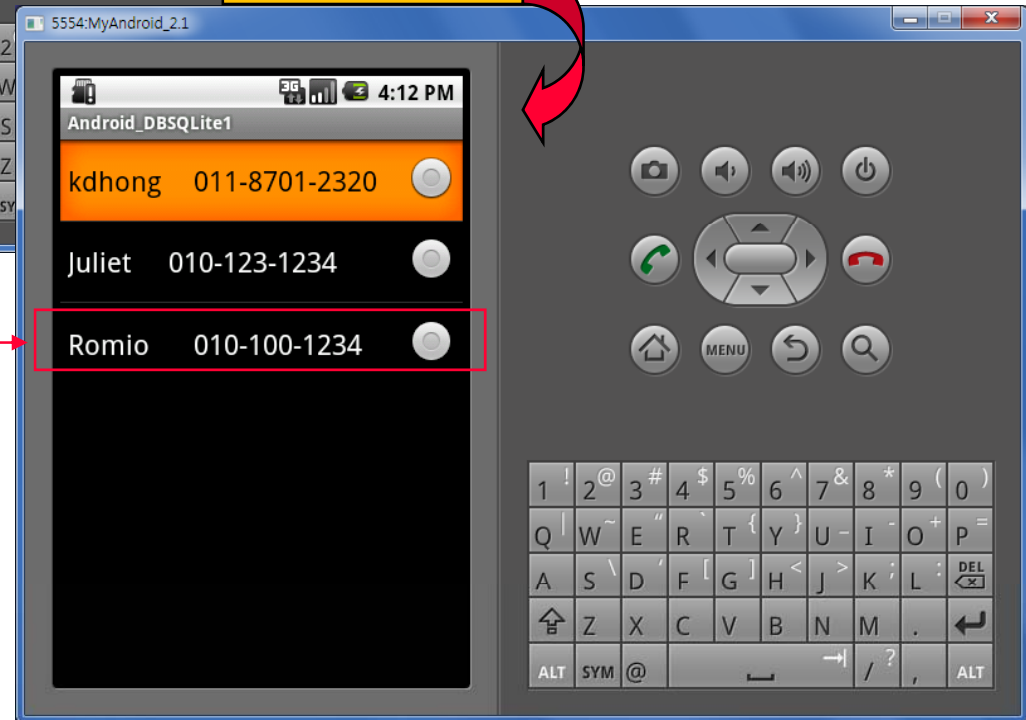
## ❖ Android 프로젝트 실행





# 실습 I : DataBase 연동 구현 (3)

## ❖ 실행 결과





## 실습 II : DataBase 연동 구현

### ■ Android\_DBSQLite2 (실습 시간 : 20분)

- ❖ 아래 그림과 같이 member 테이블의 데이터를 TableRow에 출력되도록 구현하시오.



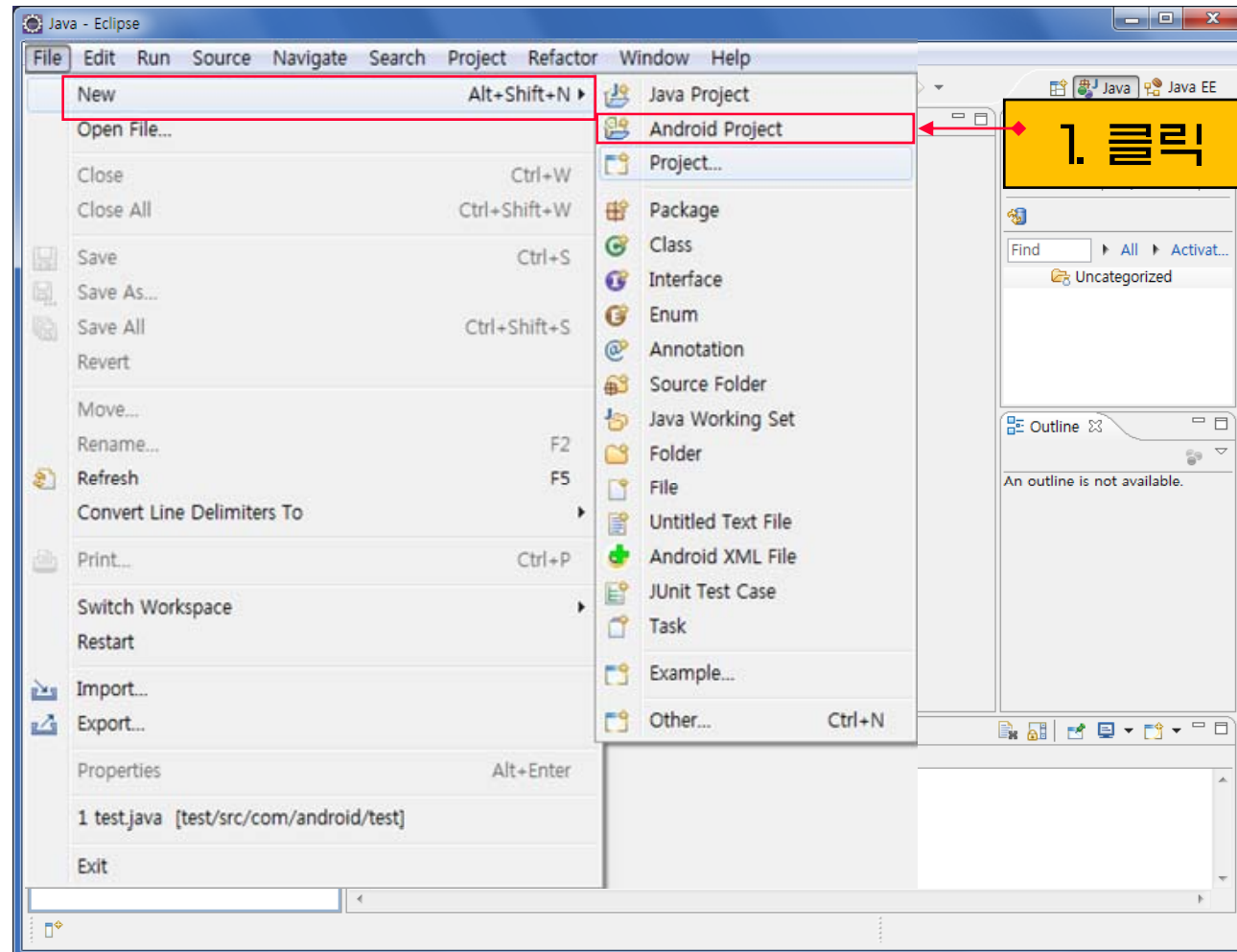




# 실습 II : DataBase 연동 구현 (1)

## ■ Android 프로젝트 생성

❖ 프로젝트 명 : Android\_DBSQLite2







## 실습 II : DataBase 연동 구현 (2)

New Android Project

Creates a new Android Project resource.

Project name:

Contents

☒ Create new project in workspace  
☐ Create project from existing source  
☒ Use default location

Location:

☐ Create project from existing sample

Samples:

Build Target

Target Name	Vendor	Platform	API ...
<input type="checkbox"/> Android 1.1	Android Open Source Project	1.1	2
<input type="checkbox"/> Android 2.0.1	Android Open Source Project	2.0.1	6
<input checked="" type="checkbox"/> Android 2.1	Android Open Source Project	2.1	7
<input type="checkbox"/> Google APIs	Google Inc.	2.0.1	6
<input type="checkbox"/> Google APIs	Google Inc.	2.1	7

Standard Android platform 2.1

Properties

Application name:

Package name:

☒ Create Activity:

Min SDK Version:

2. Android\_DBSQLite2 입력

3. 클릭

4. Android\_DBSQLite2 입력

5. com.inhatec.android\_DBSQLite2 입력

6. Android\_DBSQLite2 입력

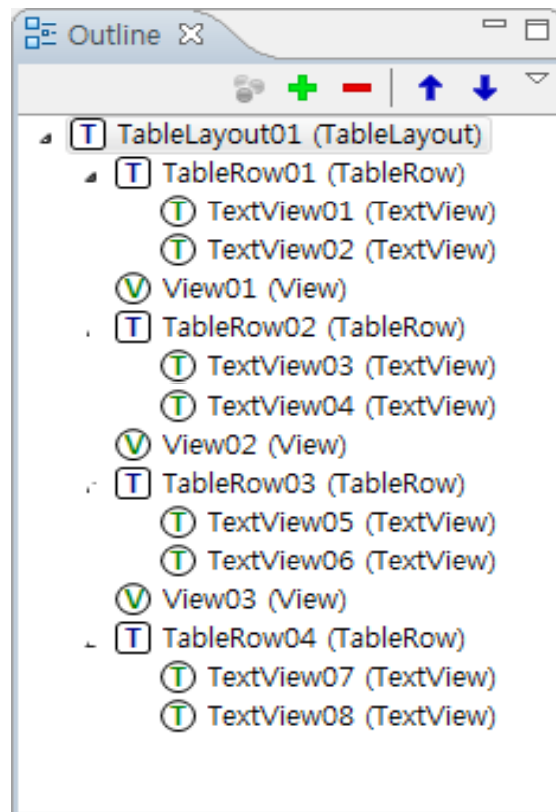
7. 클릭



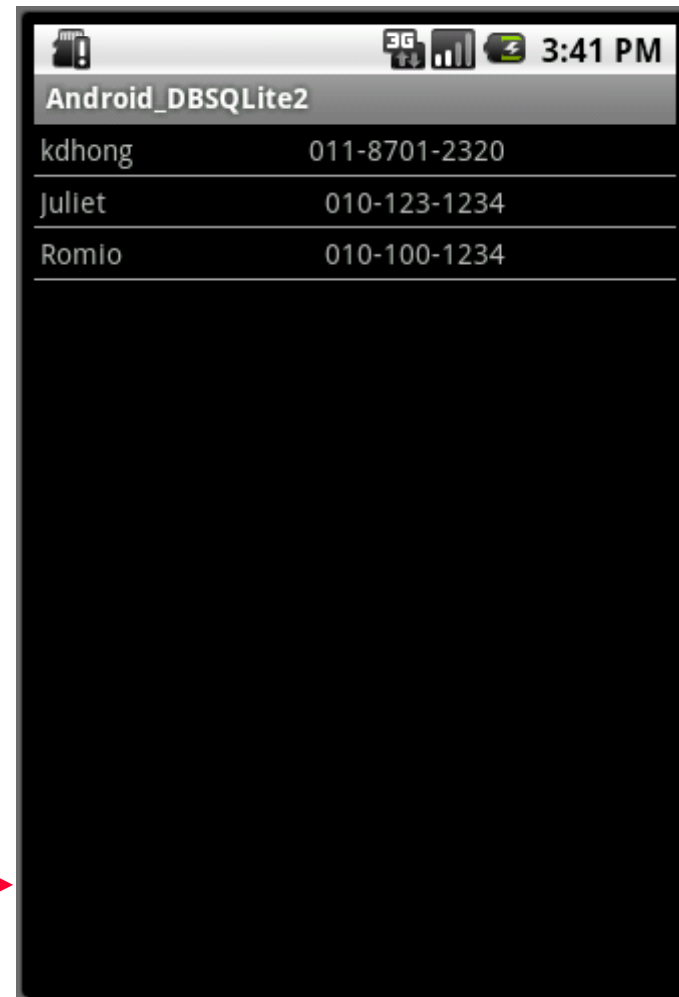


# 실습 II : DataBase 연동 구현 (3)

## UI 설계



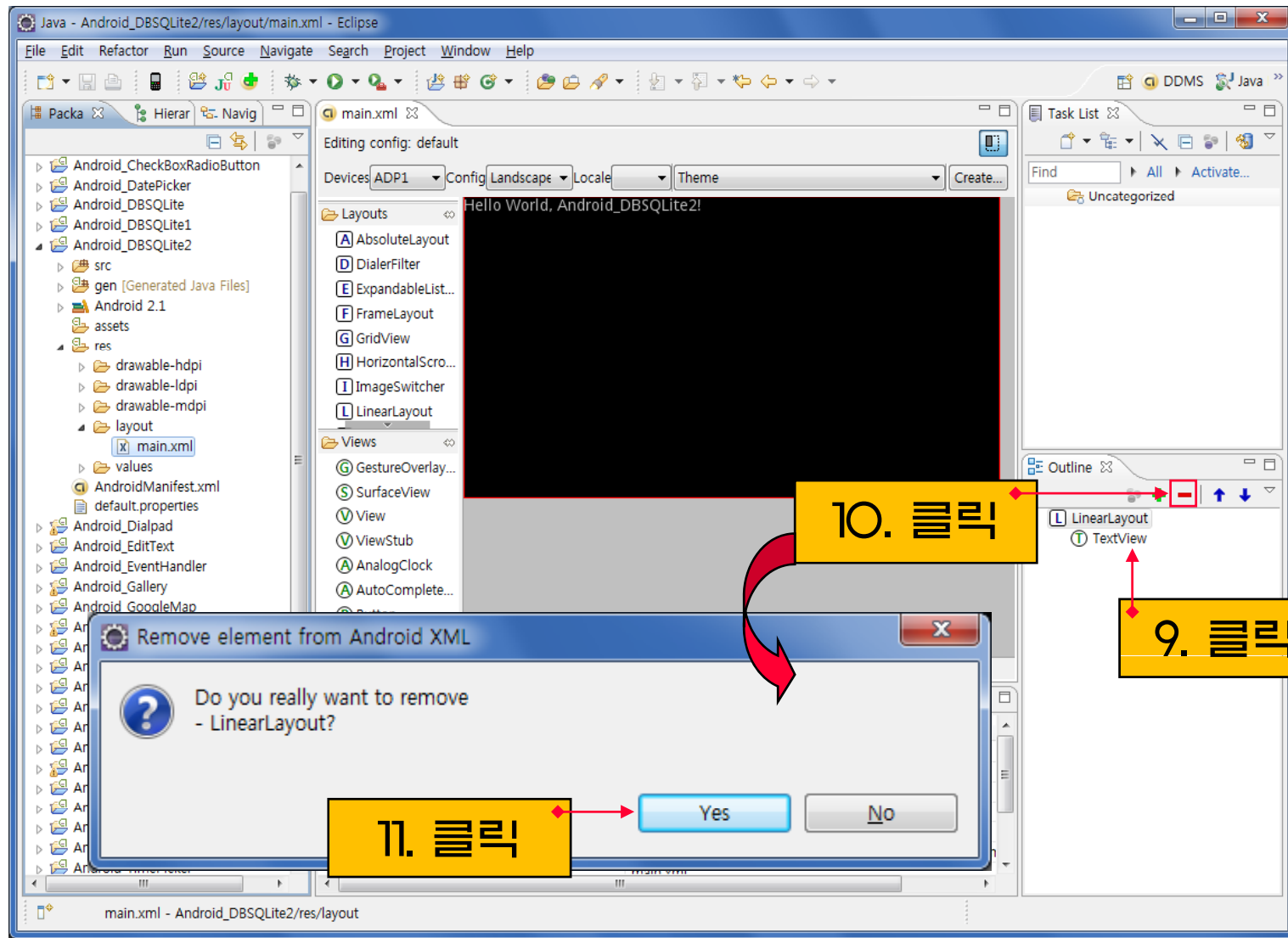
8. UI 설계 및 속성 지정





# 실습 II : DataBase 연동 구현 (4)

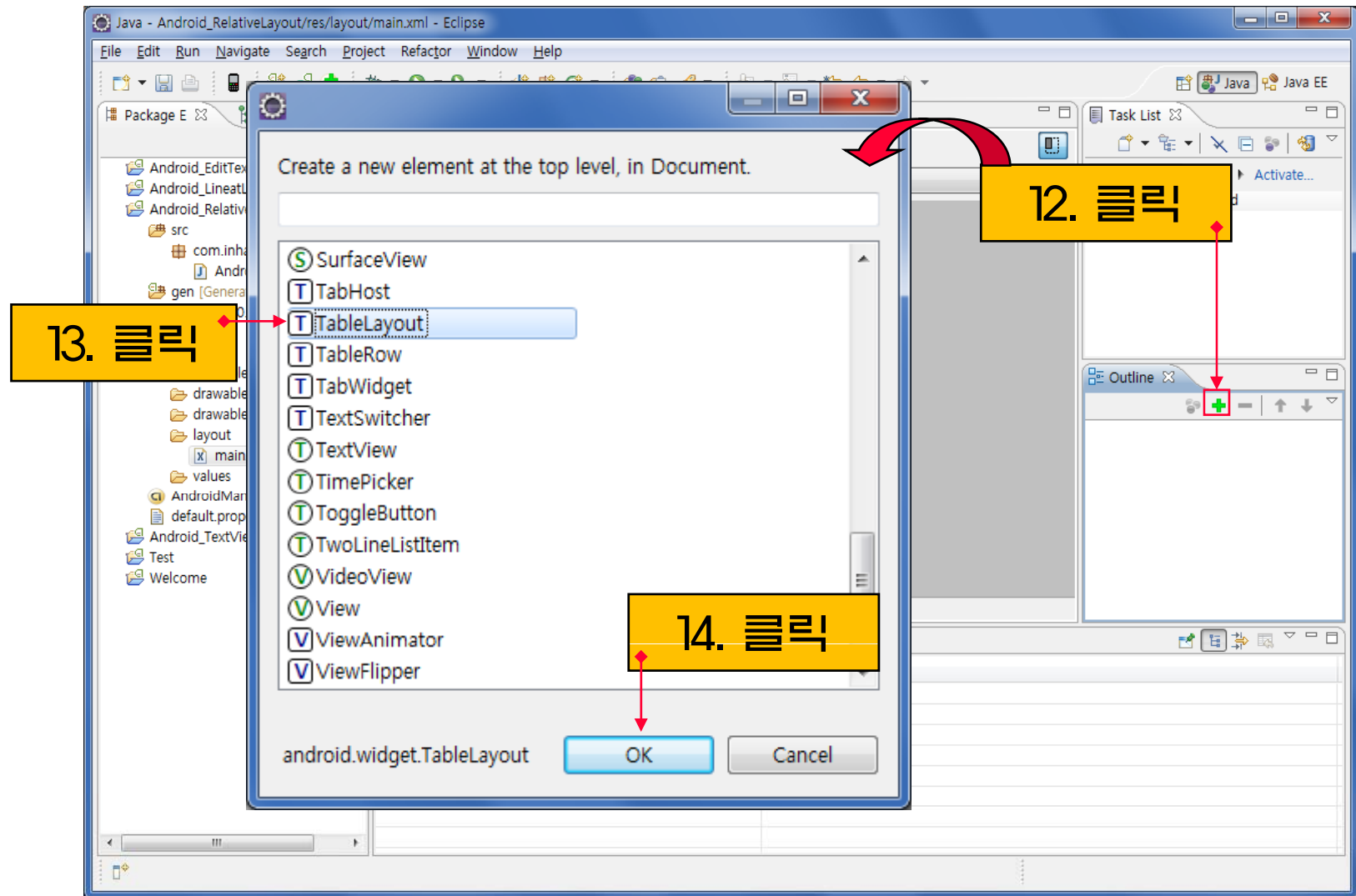
## ❖ LinearLayout / Text View 삭제





## 실습 II : DataBase 연동 구현 (5)

### ❖ TableLayout 추가





## 실습 II : DataBase 연동 구현 (6)

### ❖ TableLayout 01 속성 지정

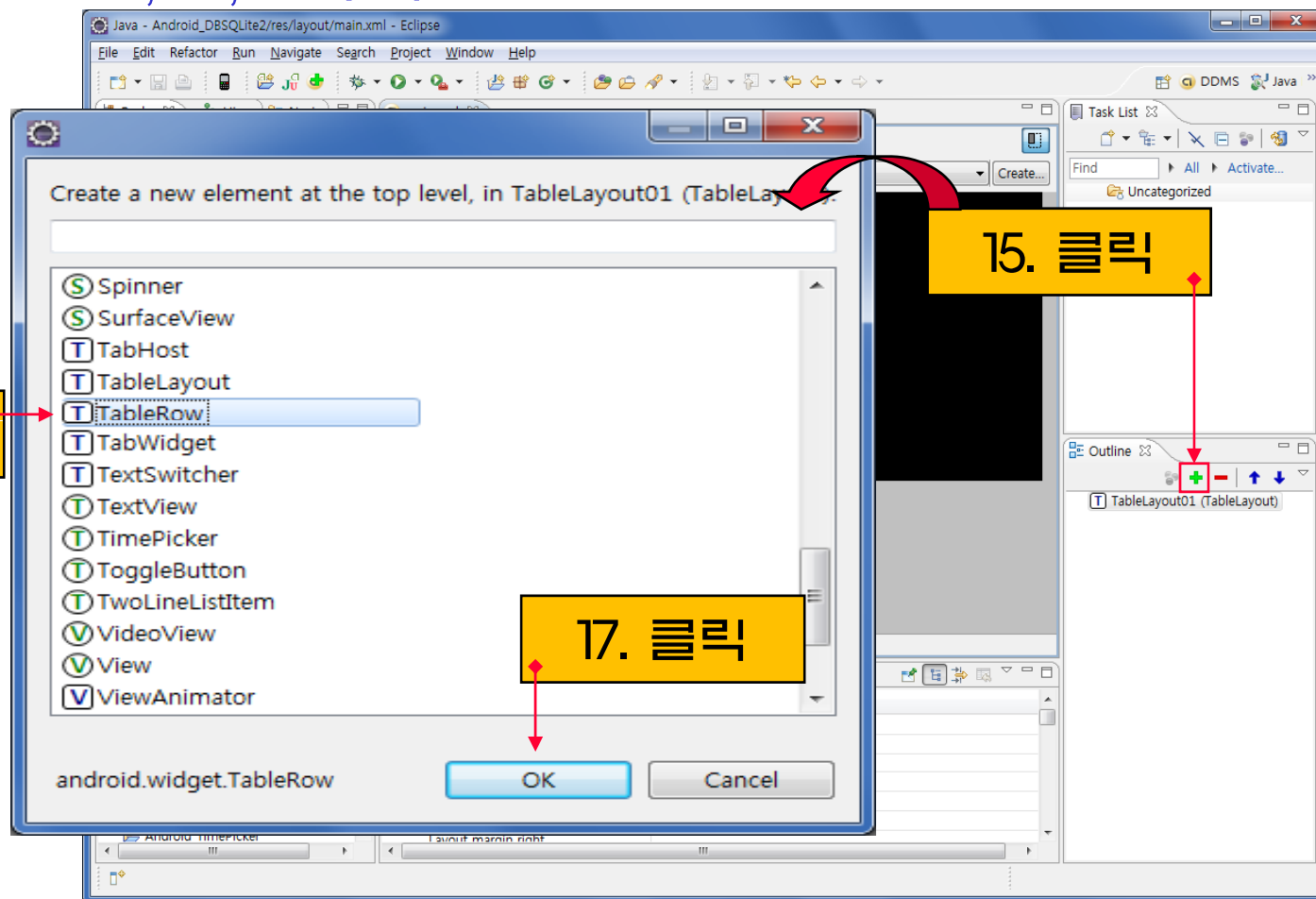
컨트롤	속성 지정
TableLayout01	<ul style="list-style-type: none"><li>• id="@+id/TableLayout01"</li><li>• layout_width="fill_parent"</li><li>• layout_height="fill_parent"</li><li>• android:stretchColumns="1"</li></ul>





## 실습 II : DataBase 연동 구현 (7)

- ❖ TableRow 01~04 추가
- ❖ TextView 01~08 추가
- ❖ View 01, 02, 03 추가



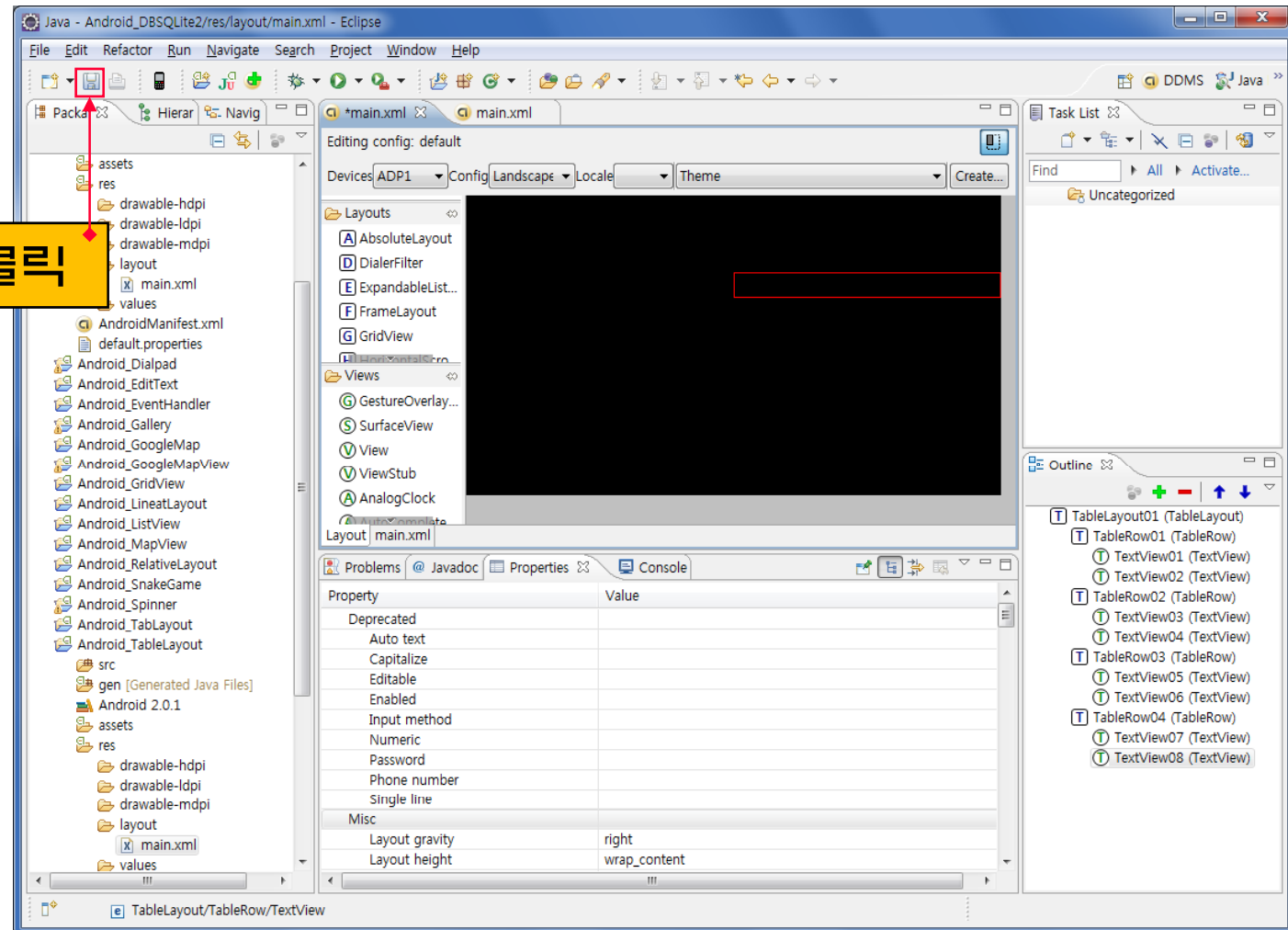


# 실습 II : DataBase 연동 구현 (8)

❖ Main.xml 저장

◆ TableRow, TextView, View 재배열

18. 클릭





## 실습 II : DataBase 연동 구현 (9)

### ❖ TableRow01, TextView 01, 02 속성 지정

컨트롤	속성 지정
TableRow01	<ul style="list-style-type: none"><li>• android:id="@+id/TableRow01"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li></ul>
TextView01	<ul style="list-style-type: none"><li>• android:id="@+id/TextView01"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:padding="3dip"</li><li>• android:layout_weight="1"</li></ul>
TextView02	<ul style="list-style-type: none"><li>• android:id="@+id/TextView02"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:padding="3dip"</li><li>• android:gravity="right"</li><li>• android:layout_weight="1"</li></ul>







## 실습 II : DataBase 연동 구현 (10)

❖ TableRow02, TextView 03, 04 속성 지정

컨트롤	속성 지정
TableRow02	<ul style="list-style-type: none"><li>• android:id="@+id/TableRow02"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li></ul>
TextView03	<ul style="list-style-type: none"><li>• android:id="@+id/TextView03"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:padding="3dip"</li><li>• android:layout_weight="1"</li></ul>
TextView04	<ul style="list-style-type: none"><li>• android:id="@+id/TextView04"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:padding="3dip"</li><li>• android:gravity="right"</li><li>• android:layout_weight="1"</li></ul>





## 실습 II : DataBase 연동 구현 (11)

❖ TableRow03, TextView 05, 06 속성 지정

컨트롤	속성 지정
TableRow03	<ul style="list-style-type: none"><li>• android:id="@+id/TableRow03"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li></ul>
TextView05	<ul style="list-style-type: none"><li>• android:id="@+id/TextView05"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:layout_weight="1"</li><li>• android:padding="3dip"</li></ul>
TextView06	<ul style="list-style-type: none"><li>• android:id="@+id/TextView06"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:padding="3dip"</li><li>• android:layout_weight="1"</li><li>• android:gravity="right"</li></ul>





## 실습 II : DataBase 연동 구현 (12)

❖ TableRow04, TextView 07, 08 속성 지정

컨트롤	속성 지정
TableRow04	<ul style="list-style-type: none"><li>• android:id="@+id/TableRow04"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li></ul>
TextView07	<ul style="list-style-type: none"><li>• android:id="@+id/TextView07"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:padding="3dip"</li><li>• android:layout_weight="1"</li></ul>
TextView08	<ul style="list-style-type: none"><li>• android:id="@+id/TextView08"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="wrap_content"</li><li>• android:padding="3dip"</li><li>• android:layout_weight="1"</li><li>• android:gravity="right"</li></ul>





## 실습 II : DataBase 연동 구현 (13)

### ❖ View01, 02, 03 속성 지정

컨트롤	속성 지정
View01	<ul style="list-style-type: none"><li>• android:id="@+id/View01"</li><li>• android:layout_width="wrap_content"</li><li>• android:background="#AAAAAA"</li><li>• android:layout_height="1dip"</li></ul>
View02	<ul style="list-style-type: none"><li>• android:id="@+id/View02"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="1dip"</li><li>• android:background="#AAAAAA"&gt;</li></ul>
View03	<ul style="list-style-type: none"><li>• android:id="@+id/View03"</li><li>• android:layout_width="wrap_content"</li><li>• android:layout_height="1dip"</li><li>• android:background="#AAAAAA"&gt;</li></ul>





# 실습 II : DataBase 연동 구현 (14)

## ❖ TableRow, TextView, View 속성 지정 결과

19. 속성 지정 결과 확인



# 실습 II : DataBase 연동 구현 (15)

## ❖ Main.xml 저장

20. 클릭

21. 클릭

Save Resource

'main.xml' has been modified. Save changes?

Yes No Cancel

TableLayout01 (TableLayout)

Property	Value
Misc	
Layout gravity	
Layout height	fill_parent
Layout margin	
Layout margin bottom	
Layout margin left	
Layout margin right	
Layout margin top	
Layout width	fill_parent
TableLayout	
Add states from children	
Always drawn with cache	
Animation cache	
Background	

TableLayout/TableRow/TextView/android:id

Launching Android\_DBSQLite2



## 실습 II : DataBase 연동 구현 (16)

❖ Main.xml

```
*main.xml
53     android:padding="3dip">
54     </TextView>
55 </TableRow>
56 <View android:id="@+id/View03"
57     android:layout_width="wrap_content"
58     android:layout_height="1dip"
59     android:background="#AAAAAA">
60 </View>
61 <TableRow
62     android:id="@+id/TableRow04"
63     android:layout_width="wrap_content"
64     android:layout_height="wrap_content">
65     <TextView
66         android:id="@+id/TextView07"
67         android:layout_width="wrap_content"
68         android:layout_height="wrap_content"
69         android:layout_weight="1"
70         android:padding="3dip">
71     </TextView>
72     <TextView
73         android:id="@+id/TextView08"
74         android:layout_width="wrap_content"
75         android:layout_height="wrap_content"
76         android:layout_weight="1"
77         android:layout_gravity="right"
78         android:padding="3dip">
79     </TextView>
80 </TableRow>
81</TableLayout>
```

22. XML code 추가 확인



## 실습 II : DataBase 연동 구현 (17)

❖ R.java

```
*R.java X
1 10/* AUTO-GENERATED FILE. DO NOT MODIFY.
7 package com.inhatc.Android_DBSQLite2;
8
9 public final class R {
10     public static final class attr {
11     }
12     public static final class drawable {
13         public static final int icon=0x7f020000;
14     }
15     public static final class id {
16         public static final int TableLayout01=0x7f050000;
17         public static final int TableRow01=0x7f050001;
18         public static final int TableRow02=0x7f050005;
19         public static final int TableRow03=0x7f050009;
20         public static final int TableRow04=0x7f05000d;
21         public static final int TextView01=0x7f050002;
22         public static final int TextView02=0x7f050003;
23         public static final int TextView03=0x7f050006;
24         public static final int TextView04=0x7f050007;
25         public static final int TextView05=0x7f05000a;
26         public static final int TextView06=0x7f05000b;
27         public static final int TextView07=0x7f05000e;
28         public static final int TextView08=0x7f05000f;
29         public static final int View01=0x7f050004;
30         public static final int View02=0x7f050008;
31         public static final int View03=0x7f05000c;
32     }
33     public static final class layout {
34         public static final int main=0x7f030000;
35     }
36     public static final class string {
37         public static final int app_name=0x7f040001;
38         public static final int hello=0x7f040000;
39     }
40 }
```

23. id 클래스 변수 추가 확인







# 실습 II : DataBase 연동 구현 (18)

## ■ Android\_DBSQLite2.java

```
Android_DBSQLite2.java x
1 package com.inhatc.Android_DBSQLite2;
2
3 import java.util.ArrayList;
4 import android.app.Activity;
5 import android.content.ContentValues;
6 import android.database.Cursor;
7 import android.database.sqlite.SQLiteDatabase;
8 import android.os.Bundle;
9 import android.widget.ArrayAdapter;
10 import android.widget.SimpleAdapter;
11 import android.widget.TextView;
12
13 public class Android_DBSQLite2 extends Activity {
14
15     SQLiteDatabase myDB;           //Database 객체 선언
16     SimpleAdapter myADT;           //Adapter 객체 선언
17     ArrayList<String> aryMBRList;  //ArrayList 객체 선언
18     ArrayAdapter<String> adtMembers; //ArrayAdapter 객체선언
19     TextView[] objTV;             //TextView 객체선언
20
21     /** Called when the activity is first created. */
22     @Override
```

24. Coding



# 실습 II : DataBase 연동 구현 (19)

## ❖ DB 연동

## 25. Coding

```
Android_DBSQLite2.java X
23 public void onCreate(Bundle savedInstanceState) {
24
25     super.onCreate(savedInstanceState);
26     setContentView(R.layout.main);
27
28     //DB 생성 (DB 명 : PhoneBook)
29     myDB = this.openOrCreateDatabase("PhoneBook", MODE_PRIVATE, null);
30     myDB.execSQL("Drop table if exists members");
31
32     //Table 생성 (Table 명 : members)
33     myDB.execSQL("Create table members (" +
34         " _id integer primary key autoincrement, " +
35         "Name text not null, " + "Phone_No text not null);" );
36
37     //Data 저장 ("kdhong", "011-8701-2320")
38     myDB.execSQL("Insert into members " +
39         " (Name, Phone_No) values ('kdhong', '011-8701-2320');" );
40
41     //members 테이블에서 Data 저장
42     ContentValues insertValue = new ContentValues();
43     insertValue.put("Name", "Juliet");
44     insertValue.put("Phone_No", "010-123-1234");
45     myDB.insert("members", null, insertValue);
46
47     insertValue.put("Name", "Romio");
48     insertValue.put("Phone_No", "010-100-1234");
49     myDB.insert("members", null, insertValue);
50
51     //members 테이블에서 모든 Record Data 가져오기
52     Cursor allRCD = myDB.query("members", null,
53         null, null, null, null, null, null);
```



## 실습 II : DataBase 연동 구현 (20)

❖ DB data → TableRow(Textview)

```
Android_DBSQLite2.java
54
55 //ArrayList 생성
56 aryMBRList = new ArrayList<String>();
57 if (allRCD != null){
58     if (allRCD.moveToFirst()) {
59         do{
60             aryMBRList.add(allRCD.getString(1));
61             aryMBRList.add(allRCD.getString(2));
62         }while(allRCD.moveToNext());
63     }
64 }
65 objTV = new TextView[8]; //ArrayList 객체 instance 생성
66 objTV[0] = (TextView) findViewById(R.id.TextView01);
67 objTV[1] = (TextView) findViewById(R.id.TextView02);
68 objTV[2] = (TextView) findViewById(R.id.TextView03);
69 objTV[3] = (TextView) findViewById(R.id.TextView04);
70 objTV[4] = (TextView) findViewById(R.id.TextView05);
71 objTV[5] = (TextView) findViewById(R.id.TextView06);
72 objTV[6] = (TextView) findViewById(R.id.TextView07);
73 objTV[7] = (TextView) findViewById(R.id.TextView08);
74
75 //ArrayList -> TextView
76 for(int i=0; i < aryMBRList.size(); i++){
77     objTV[i].setText(aryMBRList.get(i).toString());
78 }
79
80 if(myDB != null) myDB.close(); //DB 연결 해제
81 }
82 }
```

26. Coding



# 실습 II : DataBase 연동 구현 (21)

## ❖ Android 프로젝트 실행

The screenshot shows the Eclipse IDE interface. The 'Run' menu is open, and the 'Run As' dialog box is displayed. The following steps are annotated:

- 27. 클릭**: Click on the 'Run' button in the Eclipse IDE (indicated by a red arrow pointing to the 'Run' button in the top toolbar).
- 28. 클릭**: Click on 'Android Application' in the 'Run As' dialog box (indicated by a red arrow pointing to the 'Android Application' option).
- 29. 클릭**: Click on the 'OK' button in the 'Run As' dialog box (indicated by a red arrow pointing to the 'OK' button).

The 'Run As' dialog box shows the following options:

- Android Application
- Android JUnit Test
- Java Applet
- Java Application
- JUnit Test

The description for 'Android Application' is: 'Runs an Android Application'.



## 실습 II : DataBase 연동 구현 (22)

### ❖ 실행 결과



30. 클릭

31. 결과 확인



Android_DBSQLite2	
kdhong	011-8701-2320
Juliet	010-123-1234
Romio	010-100-1234





# 학습 요약

- Android DB 연동 방법
- Android DB 연동 구현
  - ❖ ListView
- Android DB 확인
- 실습 : Android DB 연동 구현 I
  - ❖ 데이터 추가
- 실습 : Android DB 연동 구현 II
  - ❖ TableRow



open handset alliance

