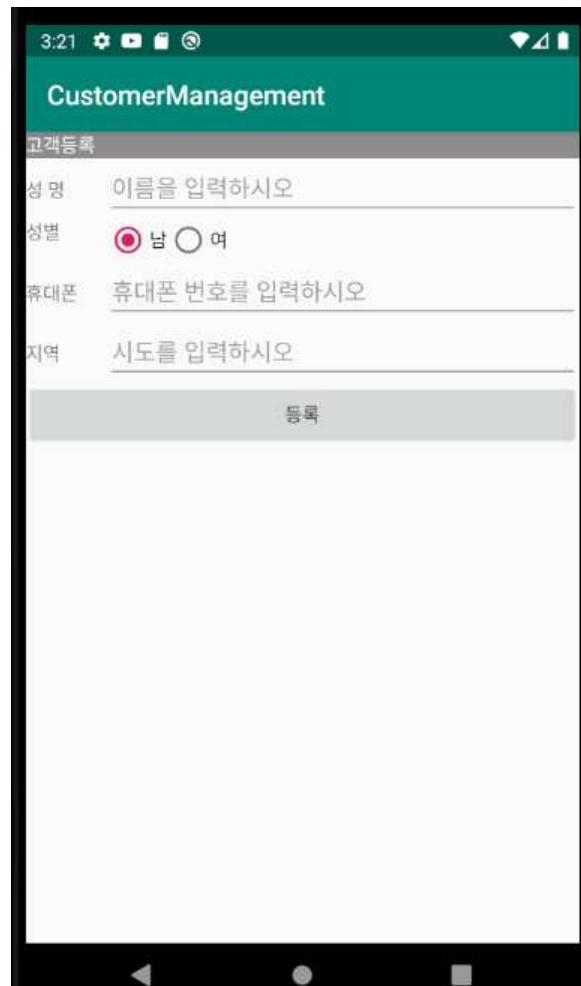


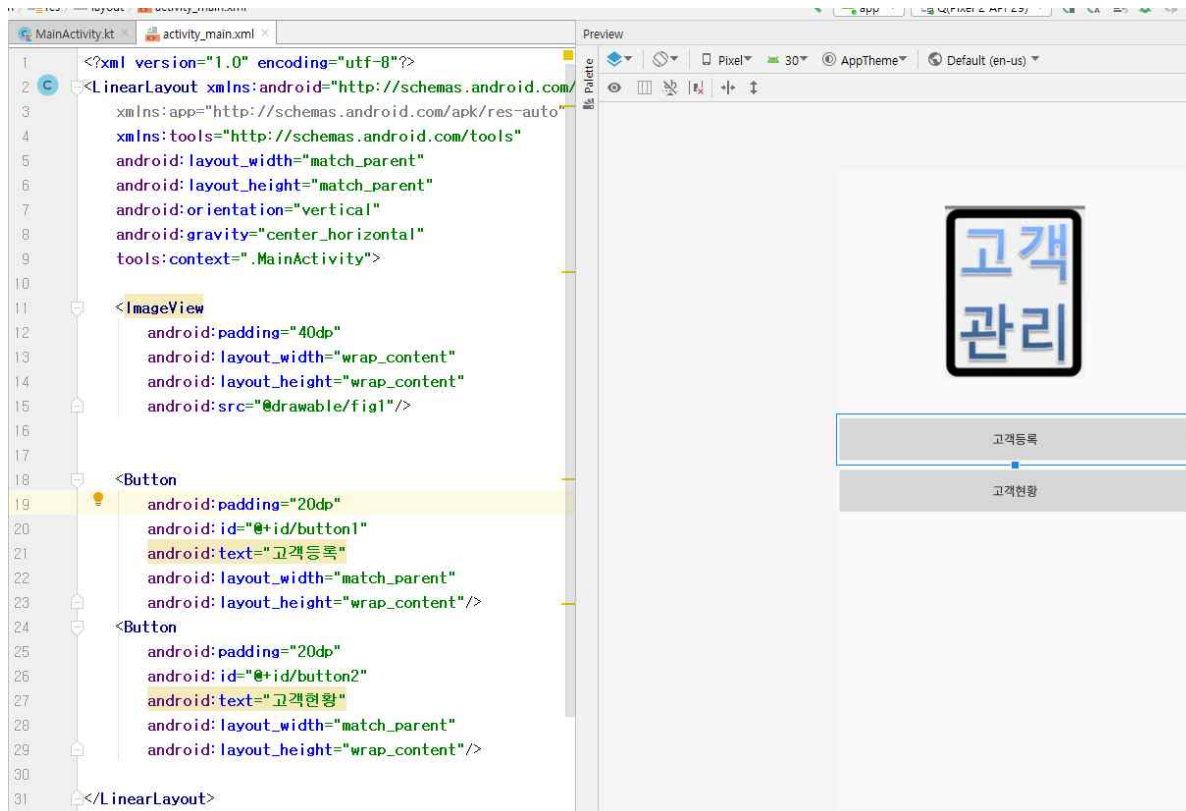
# 모바일 프로그래밍

# 데이터베이스 생성 및 데이터 삽입



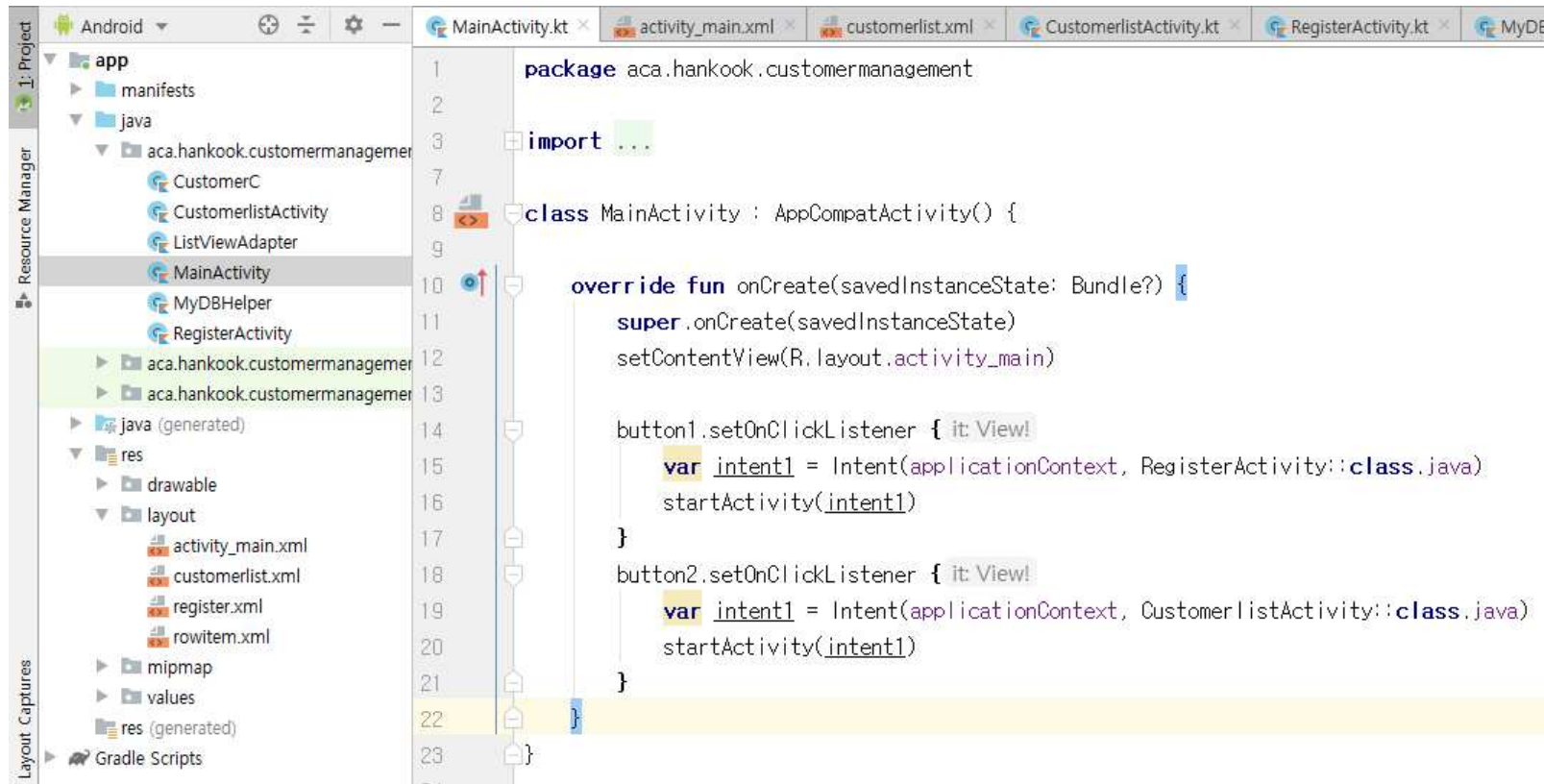
# 데이터베이스 생성 및 데이터 삽입

## Activity\_main.xml



# 데이터베이스 생성 및 데이터 삽입

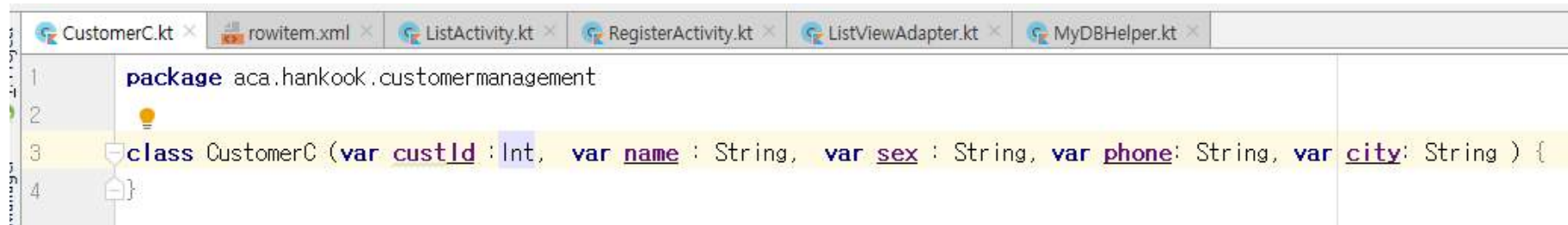
## MainActivity Class



# 고객관리 고객클래스

---

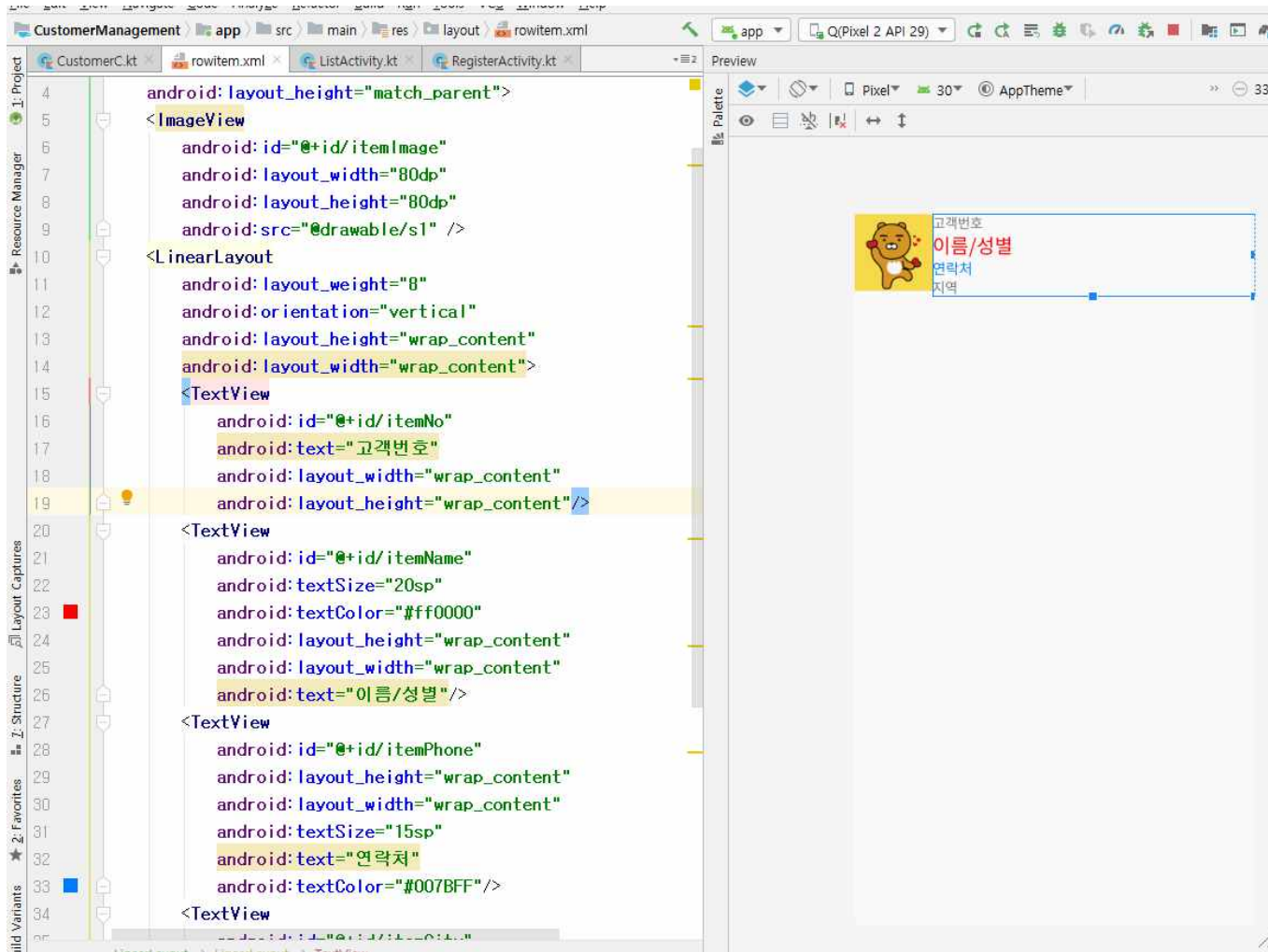
## CustomerC



```
1 package aca.hankook.customermanagement
2
3 class CustomerC (var custId : Int, var name : String, var sex : String, var phone: String, var city: String ) {
4 }
```

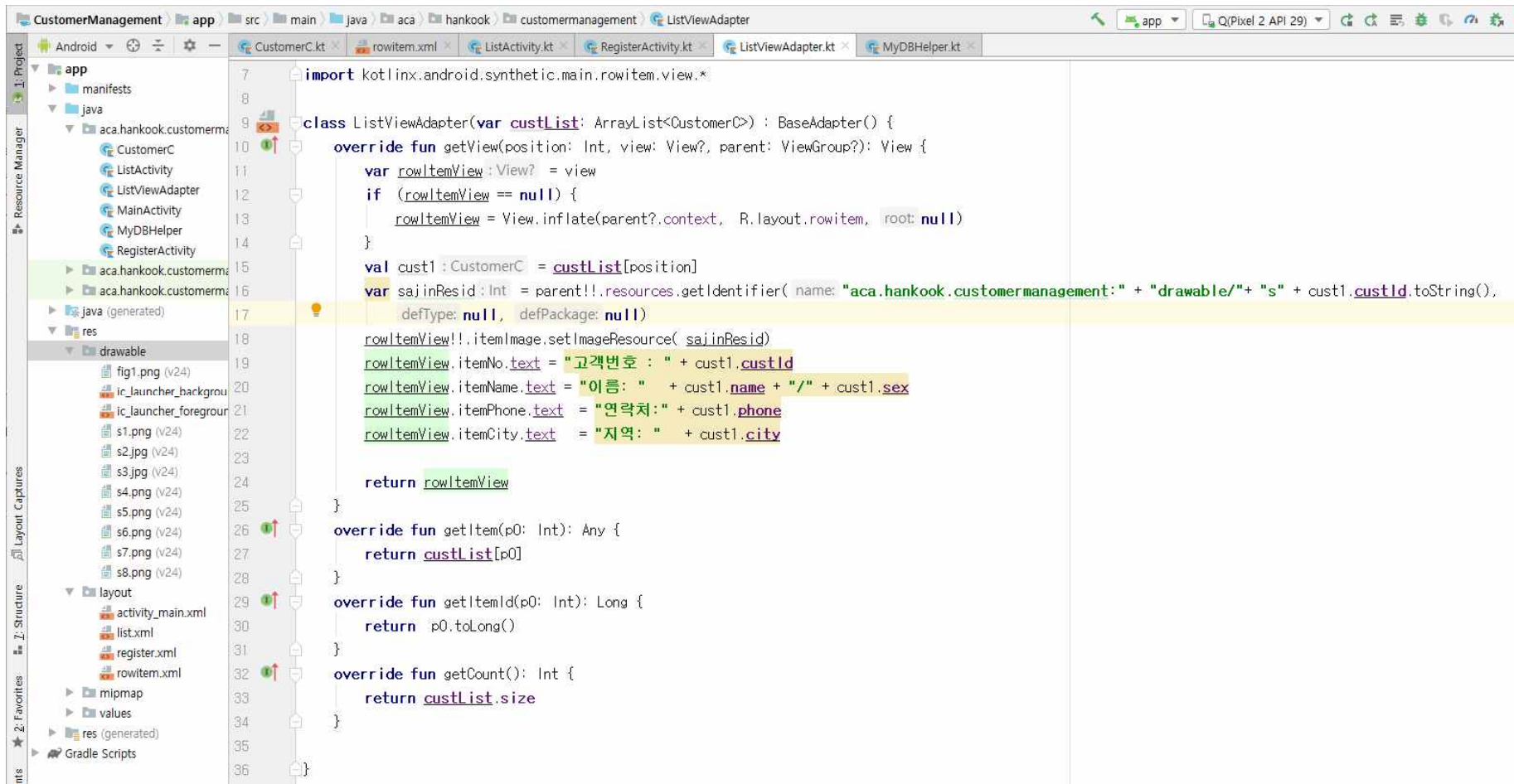
# 데이터베이스 생성 및 데이터 삽입

## Rowitem.xml



# 데이터베이스 생성 및 데이터 삽입

## ListViewAdapter



```
7 import kotlinx.android.synthetic.main.rowitem.view.*
8
9 class ListViewAdapter(var custList: ArrayList<CustomerC>) : BaseAdapter() {
10     override fun getView(position: Int, view: View?, parent: ViewGroup?): View {
11         var rowItemView: View? = view
12         if (rowItemView == null) {
13             rowItemView = View.inflate(parent?.context, R.layout.rowitem, root: null)
14         }
15         val cust1: CustomerC = custList[position]
16         var sajinResid: Int = parent!!.resources.getIdentifier( name: "aca.hankook.customermanagement:" + "drawable/" + "s" + cust1.custId.toString(),
17             defType: null, defPackage: null)
18         rowItemView!!.itemImage.setImageResource( sajinResid)
19         rowItemView.itemNo.text = "고객번호 : " + cust1.custId
20         rowItemView.itemName.text = "이름: " + cust1.name + "/" + cust1.sex
21         rowItemView.itemPhone.text = "연락처: " + cust1.phone
22         rowItemView.itemCity.text = "지역: " + cust1.city
23
24         return rowItemView
25     }
26     override fun getItem(p0: Int): Any {
27         return custList[p0]
28     }
29     override fun getItemId(p0: Int): Long {
30         return p0.toLong()
31     }
32     override fun getCount(): Int {
33         return custList.size
34     }
35 }
36 }
```



# 데이터베이스 생성 및 데이터 삽입

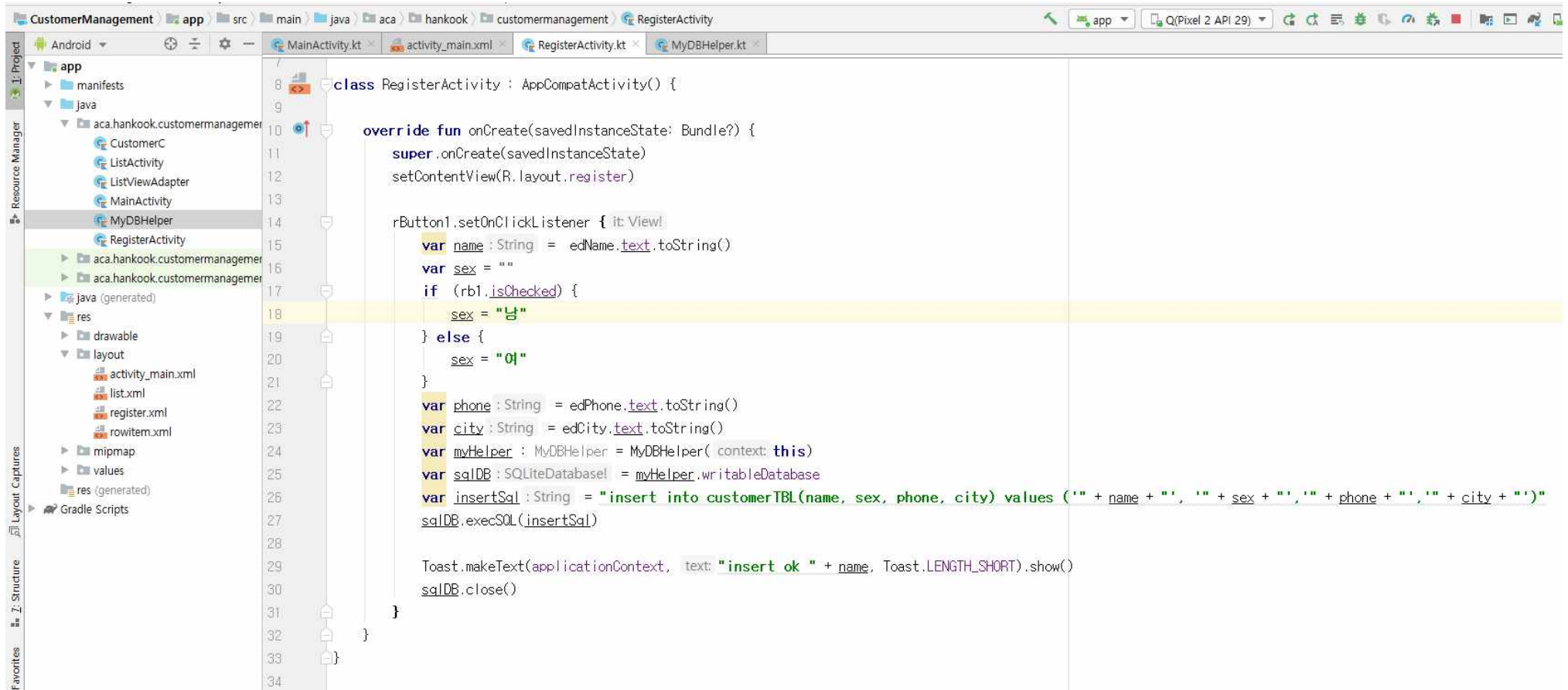
## 데이터베이스 생성 및 테이블 생성, 데이터 입력(insert)

```
CustomerC.kt x rowitem.xml x ListActivity.kt x RegisterActivity.kt x ListViewAdapter.kt x MyDBHelper.kt x
1 package aca.hankook.customermanagement
2
3 import android.content.Context
4 import android.database.sqlite.SQLiteDatabase
5 import android.database.sqlite.SQLiteOpenHelper
6
7 class MyDBHelper(var context: Context) : SQLiteOpenHelper(context, name: "customerDB", factory: null, version: 1) {
8     override fun onCreate(p0: SQLiteDatabase?) {
9         var tableCreSql = "create table customerTBL ( custId integer primary key autoincrement, name varchar(20), sex varchar(2), phone varchar(20), city varchar(20) );"
10        p0!!.execSQL(tableCreSql) //id(PK), name, dept, sajin, hdong1, hdong2
11        p0.execSQL( sql: "insert into customerTBL(name, sex, phone, city) values('홍길동', '남', '010-1111-1111', '서울');" )
12        p0.execSQL( sql: "insert into customerTBL(name, sex, phone, city) values('한지민', '여', '010-2222-2222', '경기');" )
13        p0.execSQL( sql: "insert into customerTBL(name, sex, phone, city) values('남주혁', '남', '010-3333-3333', '인천');" )
14        p0.execSQL( sql: "insert into customerTBL(name, sex, phone, city) values('조정석', '남', '010-4444-4444', '부산');" )
15        p0.execSQL( sql: "insert into customerTBL(name, sex, phone, city) values('유연석', '남', '010-5555-5555', '대구');" )
16        p0.execSQL( sql: "insert into customerTBL(name, sex, phone, city) values('정경호', '남', '010-6666-6666', '대전');" )
17        p0.execSQL( sql: "insert into customerTBL(name, sex, phone, city) values('전미도', '여', '010-7777-7777', '강원');" )
18    }
19
20    override fun onUpgrade(p0: SQLiteDatabase?, p1: Int, p2: Int) {
21        p0!!.execSQL( sql: "drop table if exists customerTBL " )
22        onCreate(p0)
23    }
24 }
```



# 고객등록

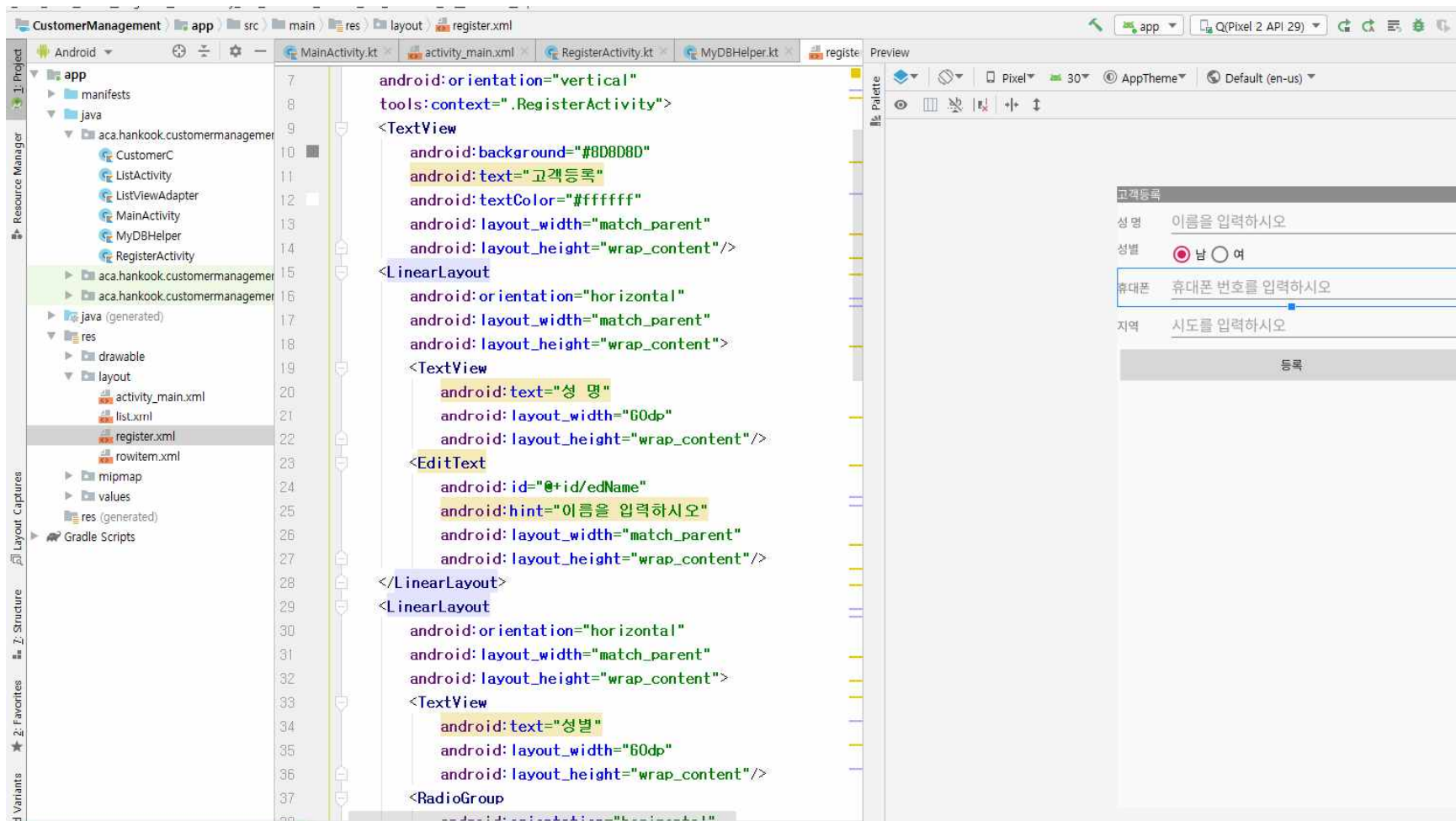
## RegisterActivity



```
7
8 class RegisterActivity : AppCompatActivity() {
9
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.register)
13
14         rButton1.setOnClickListener { it: View!
15             var name: String = edName.text.toString()
16             var sex = ""
17             if (rb1.isChecked) {
18                 sex = "남"
19             } else {
20                 sex = "여"
21             }
22             var phone: String = edPhone.text.toString()
23             var city: String = edCity.text.toString()
24             var myHelper: MyDBHelper = MyDBHelper(context: this)
25             var sqlDB: SQLiteDatabase = myHelper.writableDatabase
26             var insertSql: String = "insert into customerTbl(name, sex, phone, city) values ('" + name + "', '" + sex + "', '" + phone + "', '" + city + "')"
27             sqlDB.execSQL(insertSql)
28
29             Toast.makeText(applicationContext, text: "insert ok " + name, Toast.LENGTH_SHORT).show()
30             sqlDB.close()
31         }
32     }
33 }
34
```

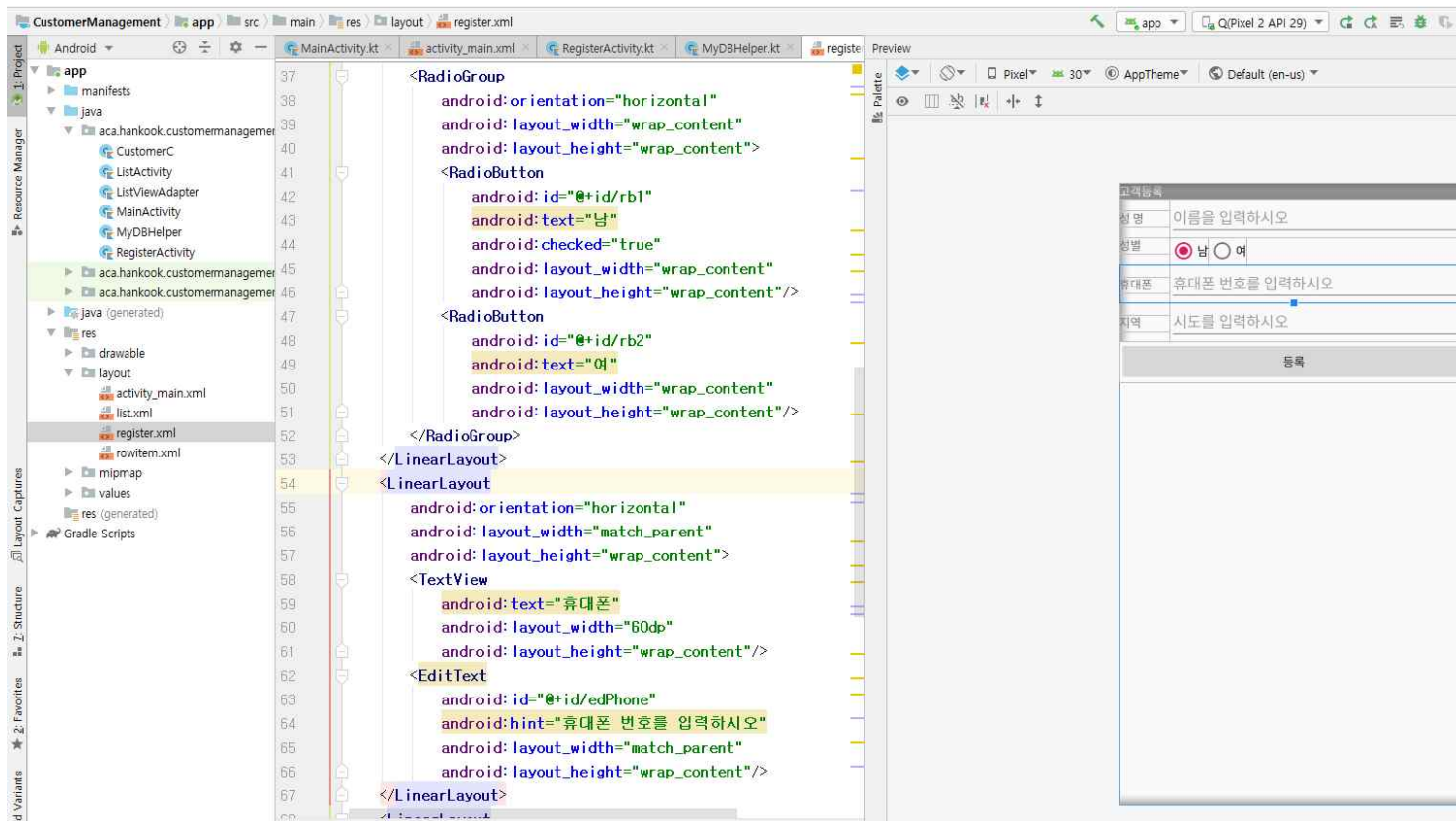
# 데이터베이스 생성 및 데이터 삽입

## register.xml



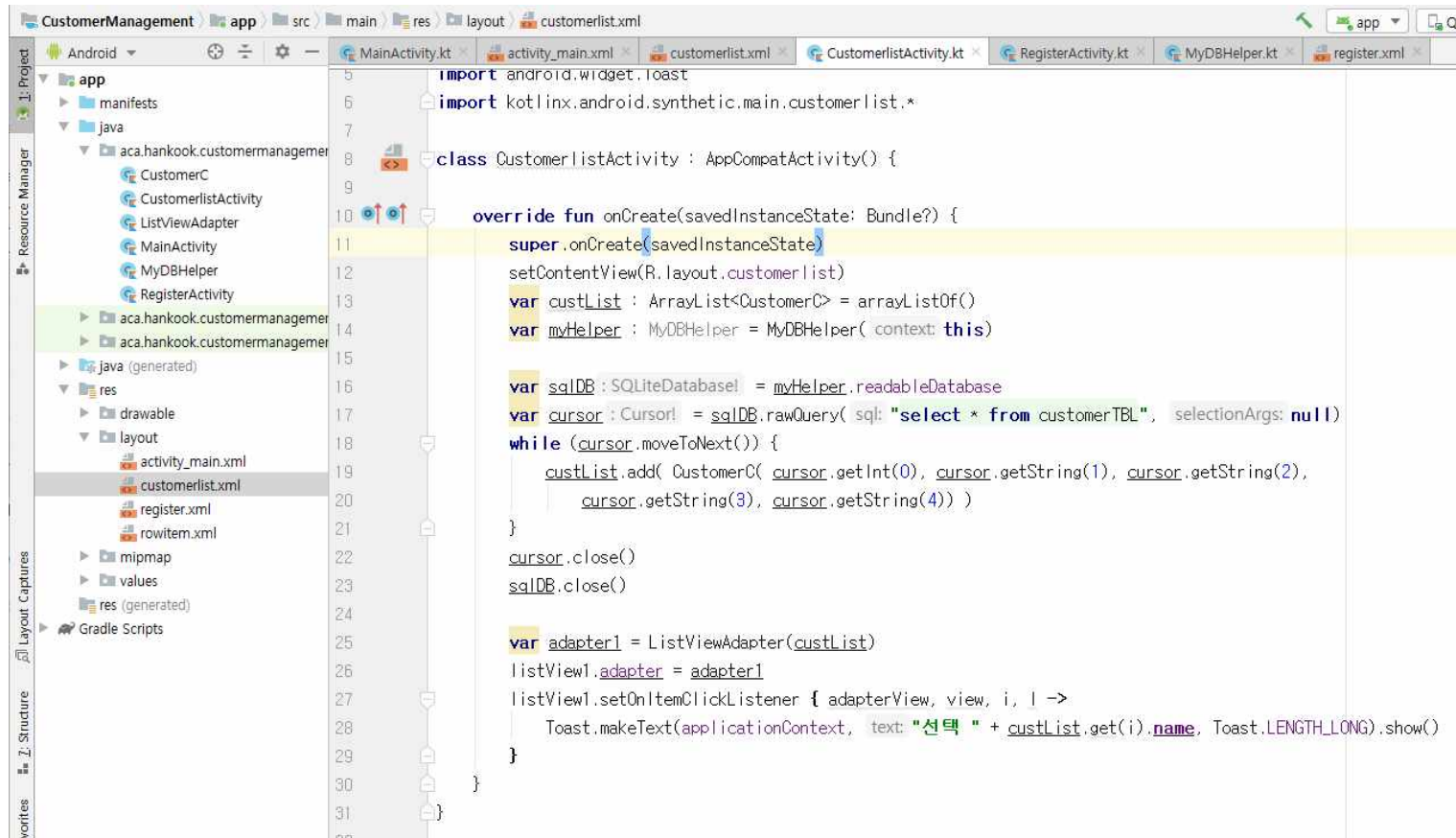
# 데이터베이스 생성 및 데이터 삽입

## register.xml



# 데이터베이스 생성 및 데이터 삽입

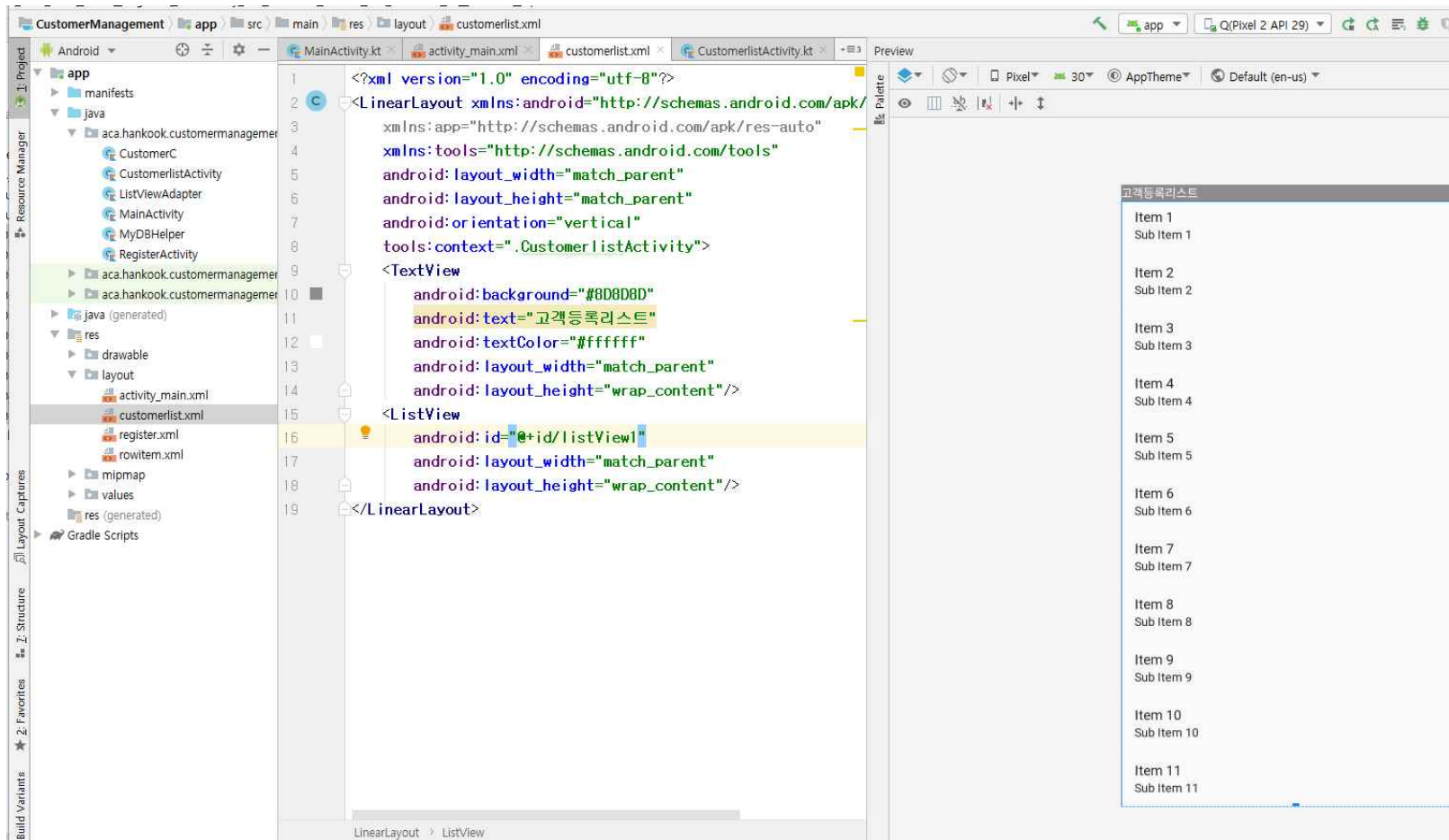
## CustomerlistActivity Class



```
5 import android.widget.Toast
6 import kotlinx.android.synthetic.main.customerlist.*
7
8 class CustomerlistActivity : AppCompatActivity() {
9
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.customerlist)
13         var custList : ArrayList<CustomerC> = arrayListOf()
14         var myHelper : MyDBHelper = MyDBHelper( context: this)
15
16         var sqldb : SQLiteDatabase! = myHelper.readableDatabase
17         var cursor : Cursor! = sqldb.rawQuery( sql: "select * from customerTBL", selectionArgs: null)
18         while (cursor.moveToNext()) {
19             custList.add( CustomerC( cursor.getInt(0), cursor.getString(1), cursor.getString(2),
20                                     cursor.getString(3), cursor.getString(4)) )
21         }
22         cursor.close()
23         sqldb.close()
24
25         var adapter1 = ListViewAdapter(custList)
26         listView1.adapter = adapter1
27         listView1.setOnItemClickListener { adapterView, view, i, l ->
28             Toast.makeText(applicationContext, text: "선택 " + custList.get(i).name, Toast.LENGTH_LONG).show()
29         }
30     }
31 }
```

# 데이터베이스 생성 및 데이터 삽입

## customerlist.xml



## 과제 및 기말고사 대비

---

- 이메일, 고객등급항목 추가 등록
- 리스트뷰 고객목록에서 해당 고객을 선택하면 고객세부정보화면 보이기



The screenshot displays a mobile application interface for 'CustomerManagement'. At the top, there is a teal header bar with the title 'CustomerManagement'. Below the header, a grey bar contains the text '고객등록' (Customer Registration). The form consists of several input fields with labels on the left and placeholder text on the right: '성명' (Name) with '이름을 입력하십시오' (Enter name), '성별' (Gender) with radio buttons for '남' (Male) and '여' (Female), '휴대폰' (Mobile Phone) with '휴대폰 번호를 입력하십시오' (Enter mobile phone number), '이메일' (Email) with 'e-mail을 입력하십시오' (Enter e-mail), '지역' (Region) with '시도를 입력하십시오' (Enter region), and '고객등급' (Customer Grade) with a dropdown menu currently showing 'VIP고객'. A grey button labeled '등록' (Register) is positioned below the form fields. The bottom portion of the screen is a large, empty white area.