Lab 6 – SQlite

2019/7/30

What is SQL?

- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

What Can SQL do?

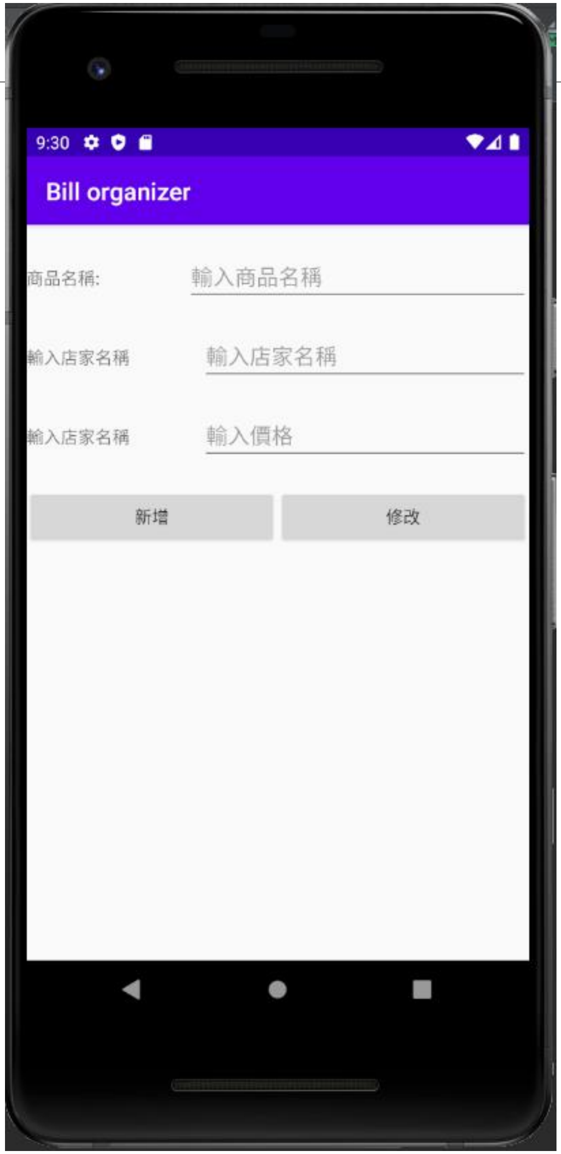
- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

The Basic Advantages of SQLite

- It’s a light weight database
- Requires very little memory
- An Automatically managed database

Today's Lab

- Bill organizer

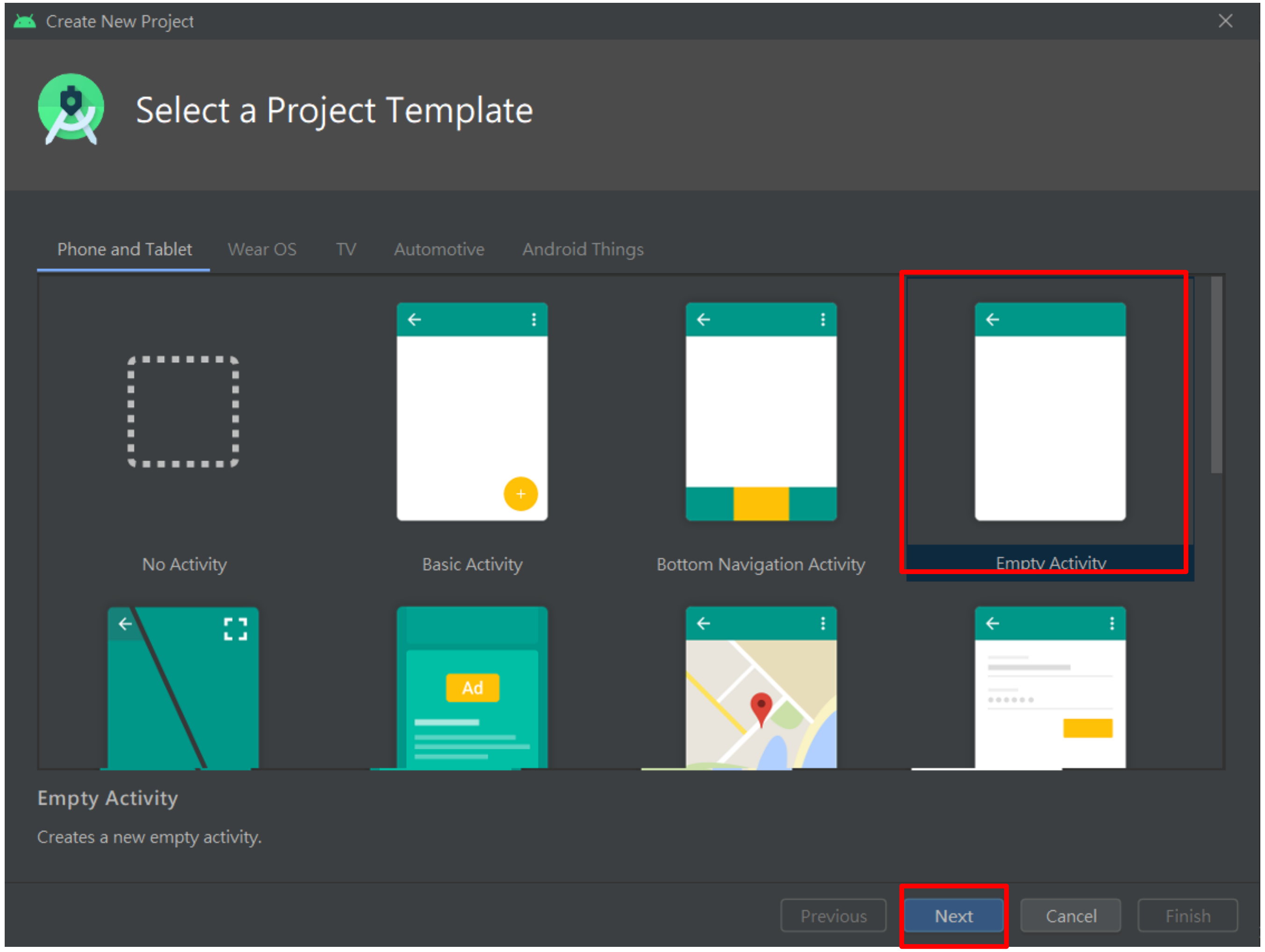


商品名稱: 輸入商品名稱

輸入店家名稱 輸入店家名稱

輸入店家名稱 輸入價格

Create Project



Create New Project

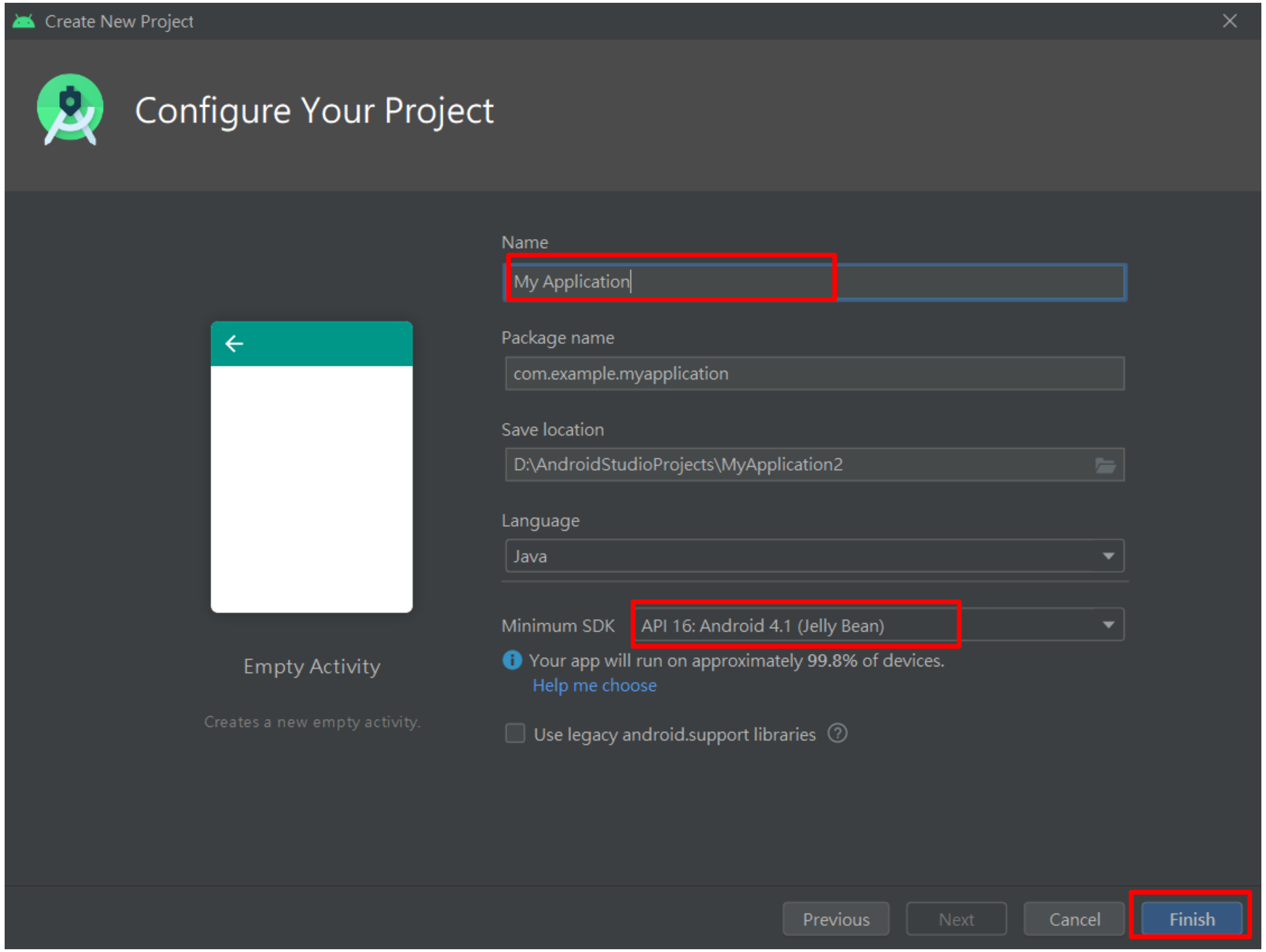
Select a Project Template

Empty Activity

Creates a new empty activity.

Create Project

1. Change your project name.
2. Check your SDK version.
3. Finish.



Configure Your Project

Name: My Application

Package name: com.example.myapplication

Save location: D:\AndroidStudioProjects\MyApplication2

Language: Java

Minimum SDK: API 16: Android 4.1 (Jelly Bean)

Your app will run on approximately 99.8% of devices.

Help me choose

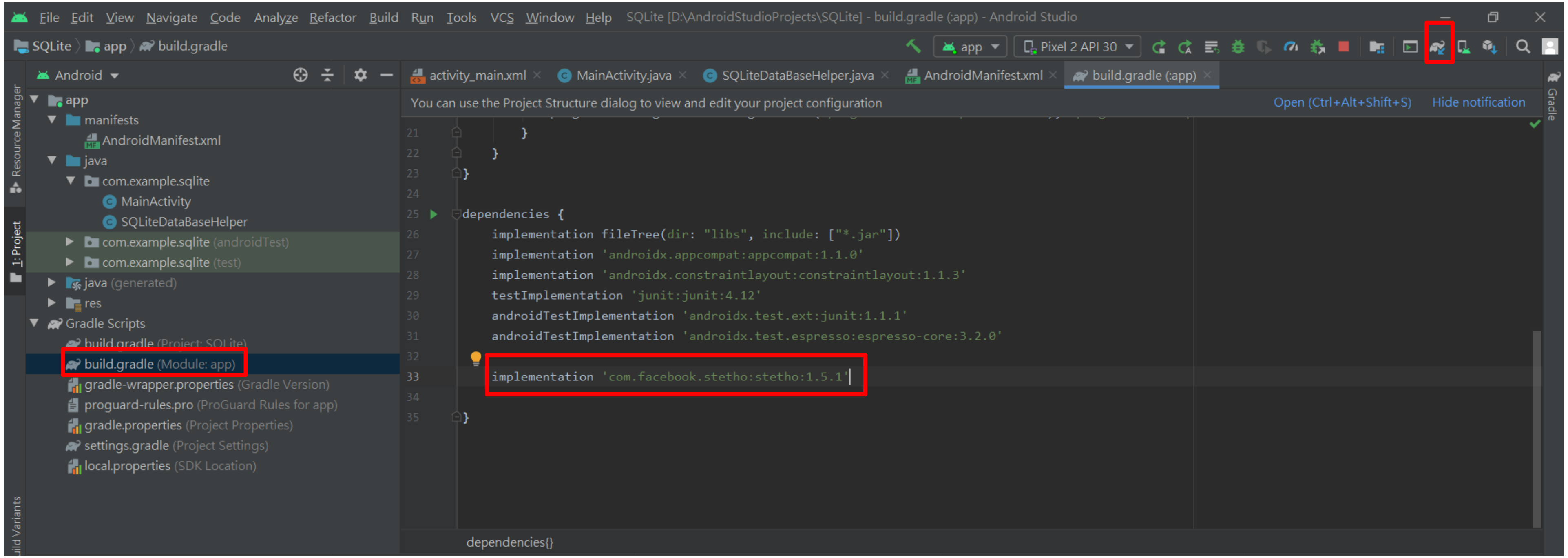
□ Use legacy android.support libraries

Empty Activity

Creates a new empty activity.

Add dependencies

1. Click build.gradle
2. Add “ implementation 'com.facebook.stetho:stetho:1.5.1’ “
3. Sync it.



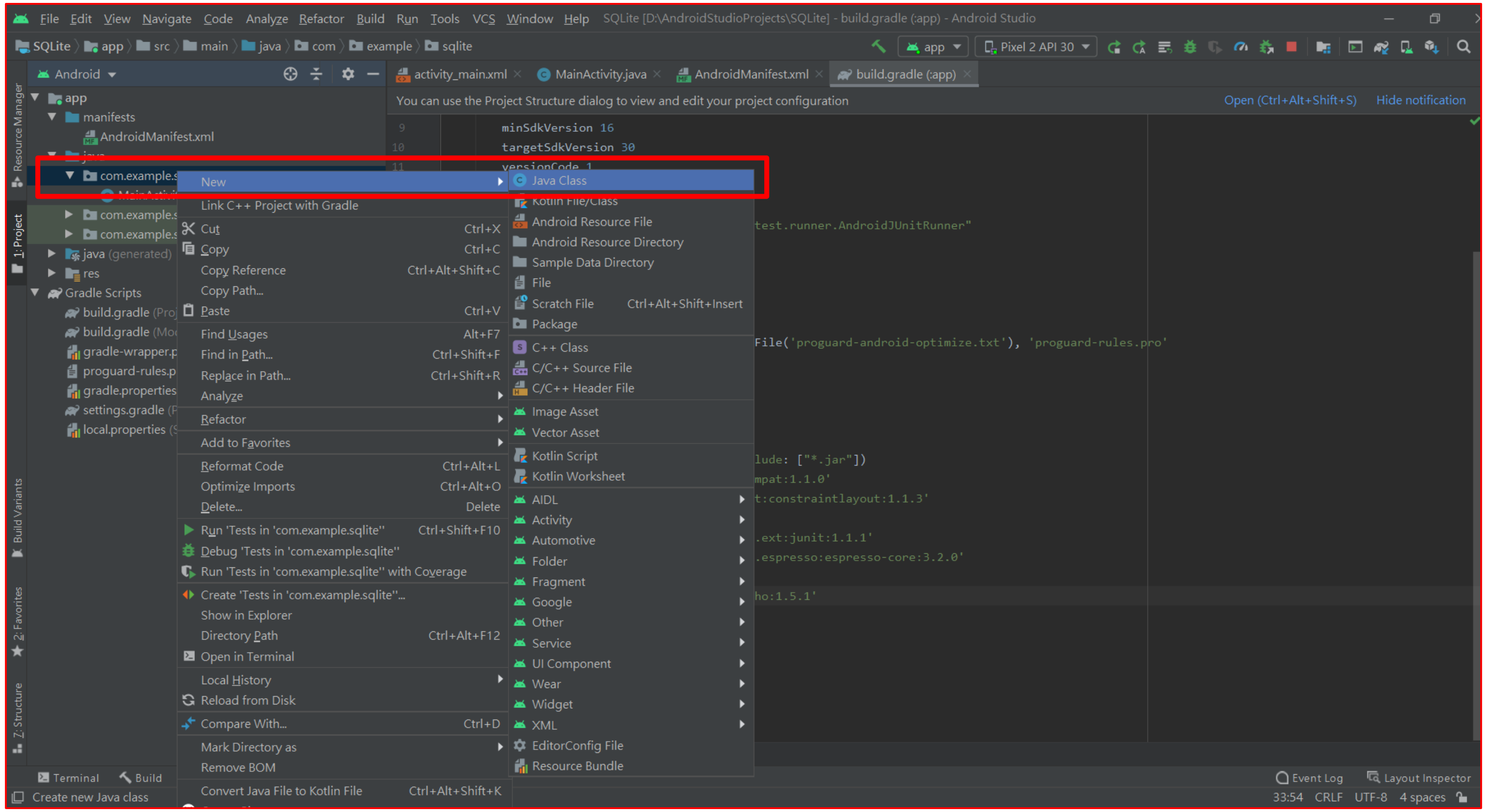
File
Edit
View
Navigate
Code
Analyze
Refactor
Build
Run
Tools
VCS
Window
Help
SQLite [D:\AndroidStudioProjects\SQLite] - build.gradle (:app) - Android Studio
SQLite
app
build.gradle
app
Pixel 2 API 30
Android
app
manifests
AndroidManifest.xml
java
com.example.sqlite
MainActivity
SQLiteDataBaseHelper
com.example.sqlite (androidTest)
com.example.sqlite (test)
java (generated)
res
Gradle Scripts
build.gradle (Project: SQLite)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
Resource Manager
1: Project
Build Variants
Gradle
activity_main.xml
MainActivity.java
SQLiteDataBaseHelper.java
AndroidManifest.xml
build.gradle (:app)
Open (Ctrl+Alt+Shift+S)
Hide notification
dependencies{}

You can use the Project Structure dialog to view and edit your project configuration

```
        }
    }
}

dependencies {
    implementation fileTree(dir: "libs", include: ["*.jar"])
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'

    implementation 'com.facebook.stetho:stetho:1.5.1'

}
```

Create New Java Class

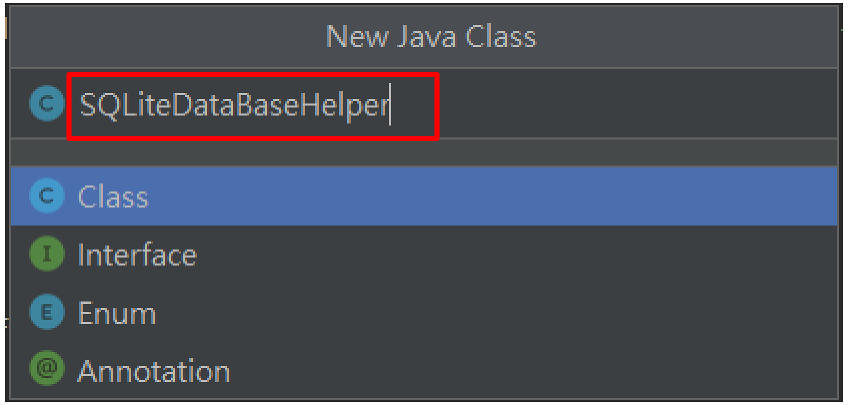
1. Right click the folder.
2. Create a new Java Class.



```
        minSdkVersion 16
        targetSdkVersion 30
        versionCode 1
test.runner.AndroidJUnitRunner"
File('proguard-android-optimize.txt'), 'proguard-rules.pro'
lude: ["*.jar"])
mpat:1.1.0'
t:constraintlayout:1.1.3'
.ext:junit:1.1.1'
.espresso:espresso-core:3.2.0'
ho:1.5.1'
```

Create New Java Class

1. Name for the Java class.
2. Double click Class.

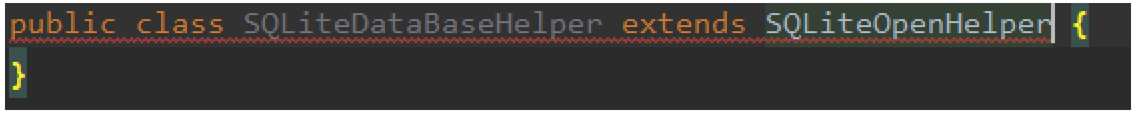


New Java Class

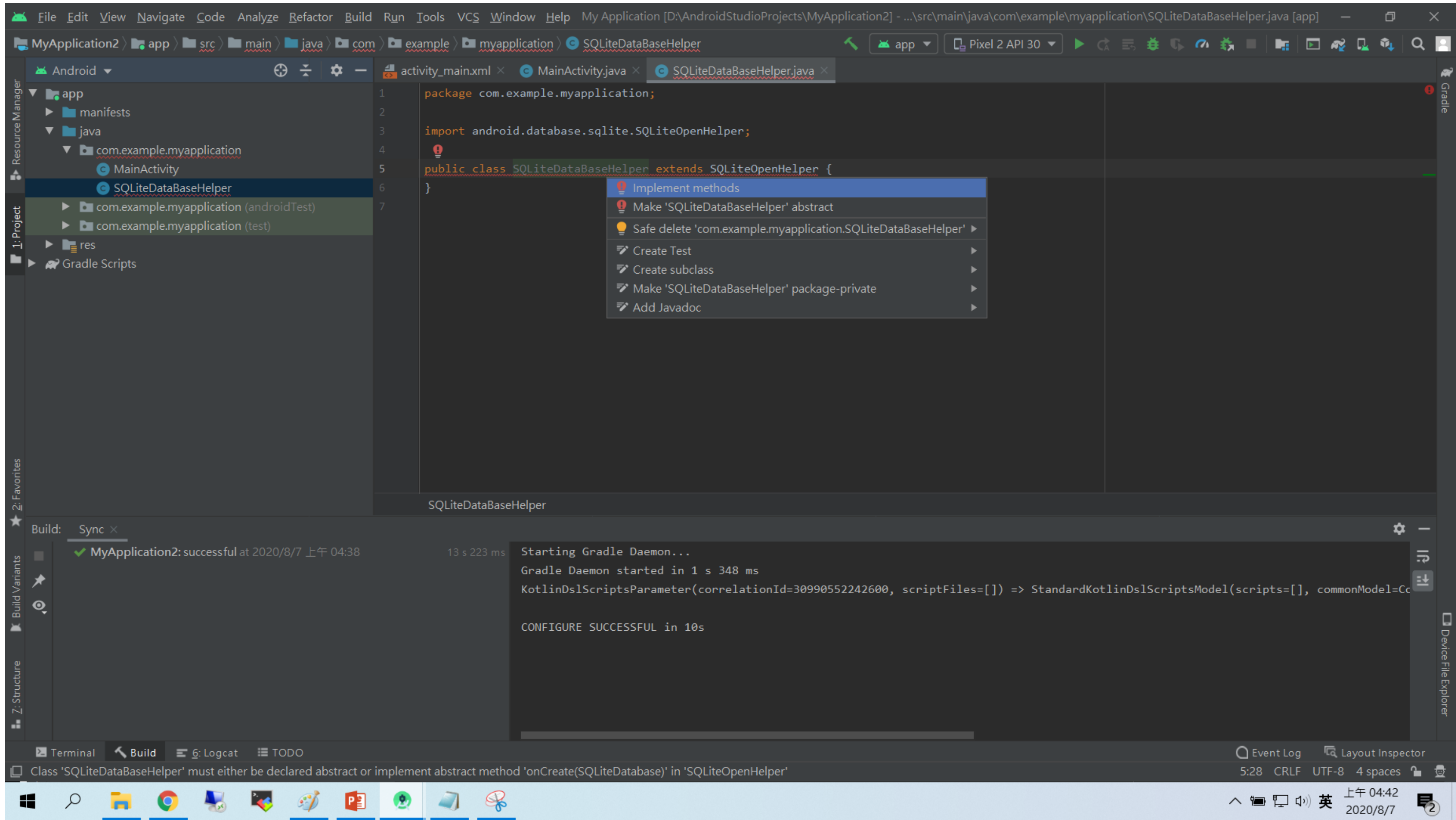
SQLiteDataBaseHelper

SQLiteDataBaseHelper

1. Add “extends SQLiteOPenHelper”
2. ALT + ENTER
3. Implement methods



```java
public class SQLiteDataBaseHelper extends SQLiteOpenHelper {
}
```



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
My Application [D:\AndroidStudioProjects\MyApplication2] - ...\src\main\java\com\example\myapplication\SQLiteDataBaseHelper.java [app]
MyApplication2 app src main java com example myapplication SQLiteDataBaseHelper
app
Pixel 2 API 30
Android
app
manifests
java
com.example.myapplication
MainActivity
SQLiteDataBaseHelper
com.example.myapplication (androidTest)
com.example.myapplication (test)
res
Gradle Scripts
activity_main.xml
MainActivity.java
SQLiteDataBaseHelper.java
Implement methods
Make 'SQLiteDataBaseHelper' abstract
Safe delete 'com.example.myapplication.SQLiteDataBaseHelper'
Create Test
Create subclass
Make 'SQLiteDataBaseHelper' package-private
Add Javadoc
Build: Sync
Terminal
Build
6: Logcat
TODO
Event Log
Layout Inspector
5:28 CRLF UTF-8 4 spaces

```java
package com.example.myapplication;

import android.database.sqlite.SQLiteOpenHelper;

public class SQLiteDataBaseHelper extends SQLiteOpenHelper {
}
```

SQLiteDataBaseHelper

MyApplication2: successful at 2020/8/7 上午 04:38 13 s 223 ms

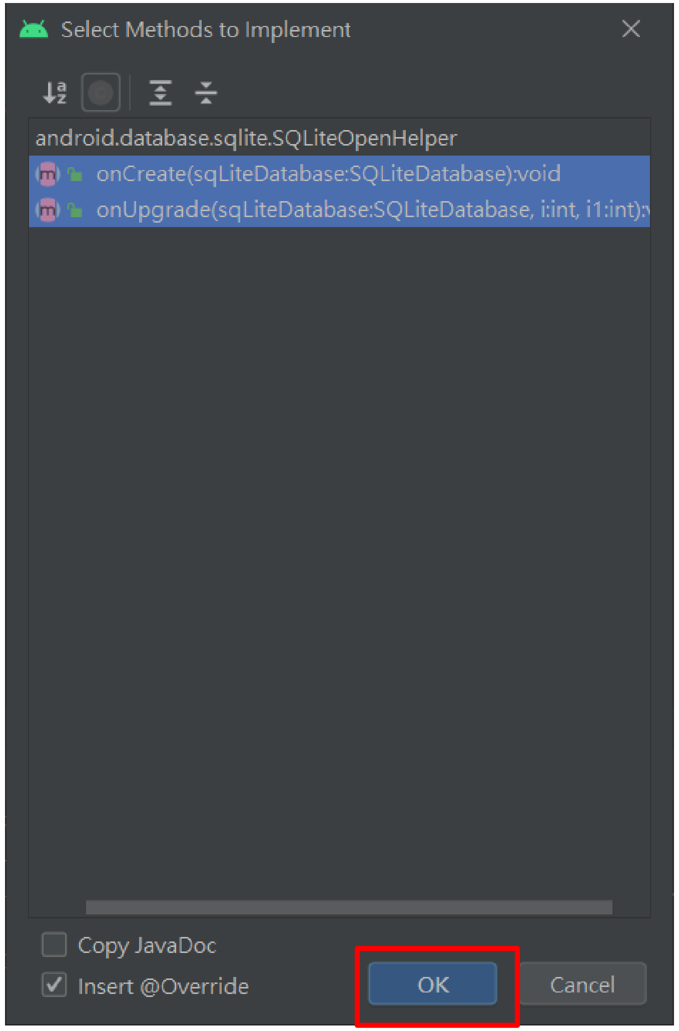
```
Starting Gradle Daemon...
Gradle Daemon started in 1 s 348 ms
KotlinDslScriptsParameter(correlationId=30990552242600, scriptFiles=[]) => StandardKotlinDslScriptsModel(scripts=[], commonModel=Cc

CONFIGURE SUCCESSFUL in 10s
```

Class 'SQLiteDataBaseHelper' must either be declared abstract or implement abstract method 'onCreate(SQLiteDatabase)' in 'SQLiteOpenHelper'

上午 04:42
2020/8/7

SQLiteDataBaseHelper

1. Click “OK”



Select Methods to Implement

android.database.sqlite.SQLiteOpenHelper

onCreate(sqLiteDatabase:SQLiteDatabase):void

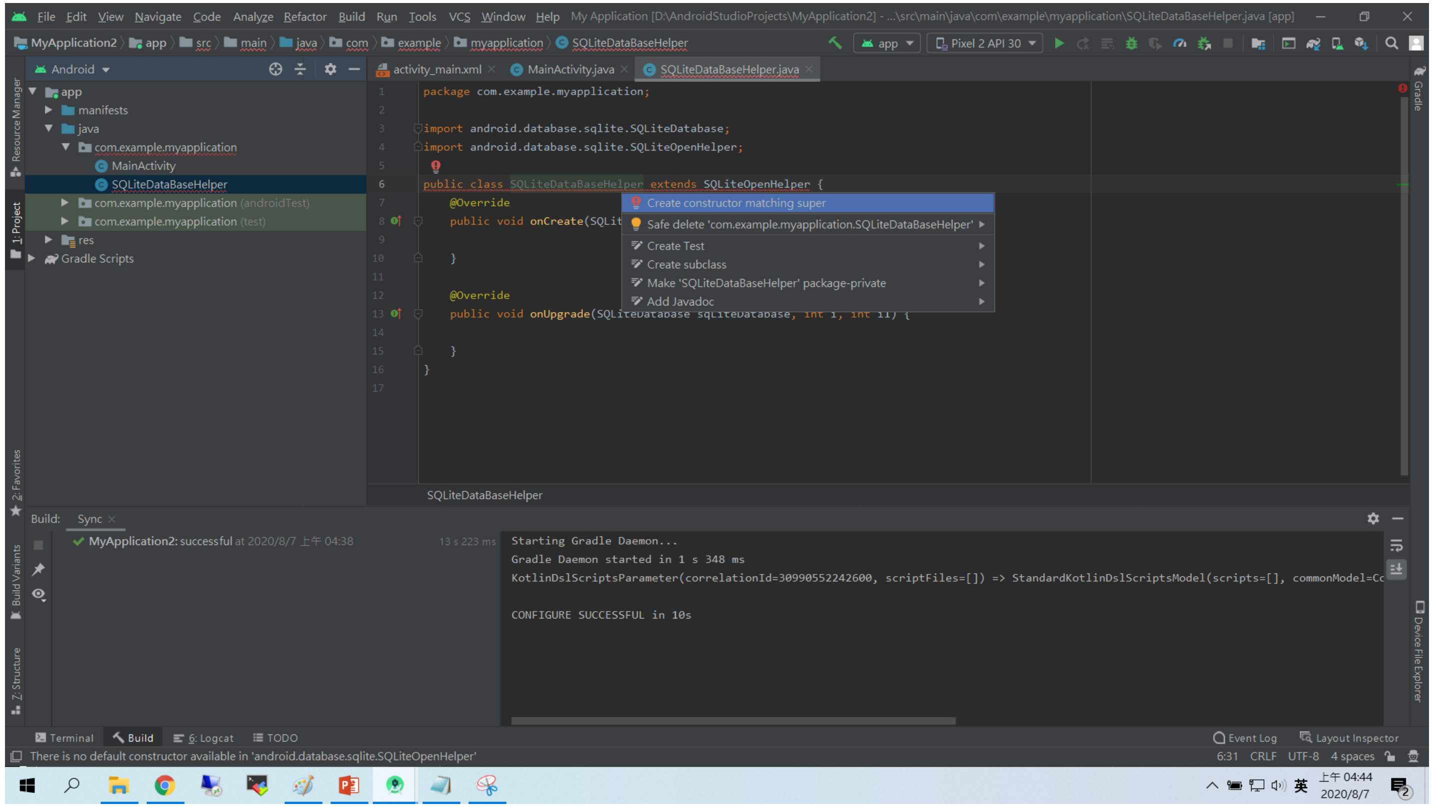
onUpgrade(sqLiteDatabase:SQLiteDatabase, i:int, i1:int):

☐ Copy JavaDoc

☑ Insert @Override

SQLiteDataBaseHelper

1. ALT + ENTER
2. Create constructor maching super



```java
package com.example.myapplication;

import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class SQLiteDataBaseHelper extends SQLiteOpenHelper {
    @Override
    public void onCreate(SQLit

    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

    }
}
```

Create constructor matching super
Safe delete 'com.example.myapplication.SQLiteDataBaseHelper'
Create Test
Create subclass
Make 'SQLiteDataBaseHelper' package-private
Add Javadoc
MyApplication2 app src main java com example myapplication SQLiteDataBaseHelper
activity_main.xml
MainActivity.java
SQLiteDataBaseHelper.java
app
Pixel 2 API 30
Android
app
manifests
java
com.example.myapplication
MainActivity
SQLiteDataBaseHelper
com.example.myapplication (androidTest)
com.example.myapplication (test)
res
Gradle Scripts
Terminal
Build
6: Logcat
TODO
Event Log
Layout Inspector

Build: Sync

MyApplication2: successful at 2020/8/7 上午 04:38 13 s 223 ms

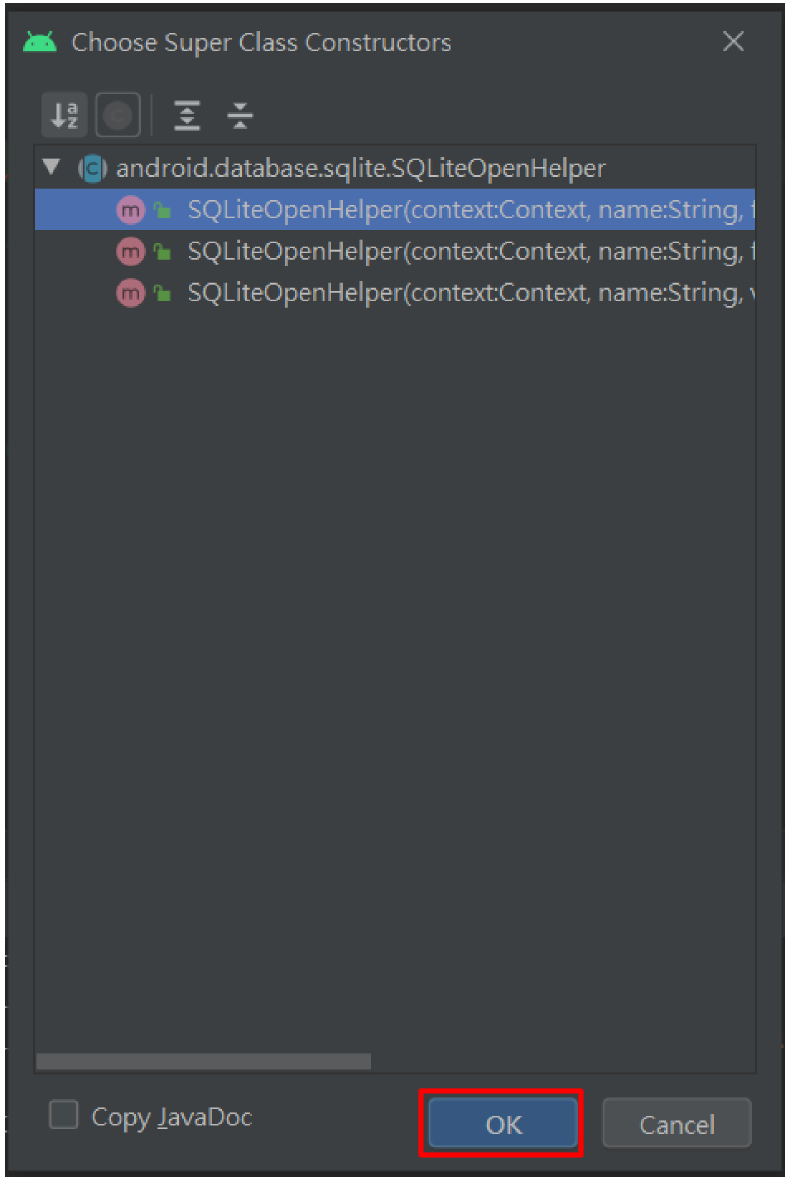
```
Starting Gradle Daemon...
Gradle Daemon started in 1 s 348 ms
KotlinDslScriptsParameter(correlationId=30990552242600, scriptFiles=[]) => StandardKotlinDslScriptsModel(scripts=[], commonModel=Cc

CONFIGURE SUCCESSFUL in 10s
```

There is no default constructor available in 'android.database.sqlite.SQLiteOpenHelper'

6:31 CRLF UTF-8 4 spaces

SQLiteDataBaseHelper

1. Click “OK”



Choose Super Class Constructors

android.database.sqlite.SQLiteOpenHelper

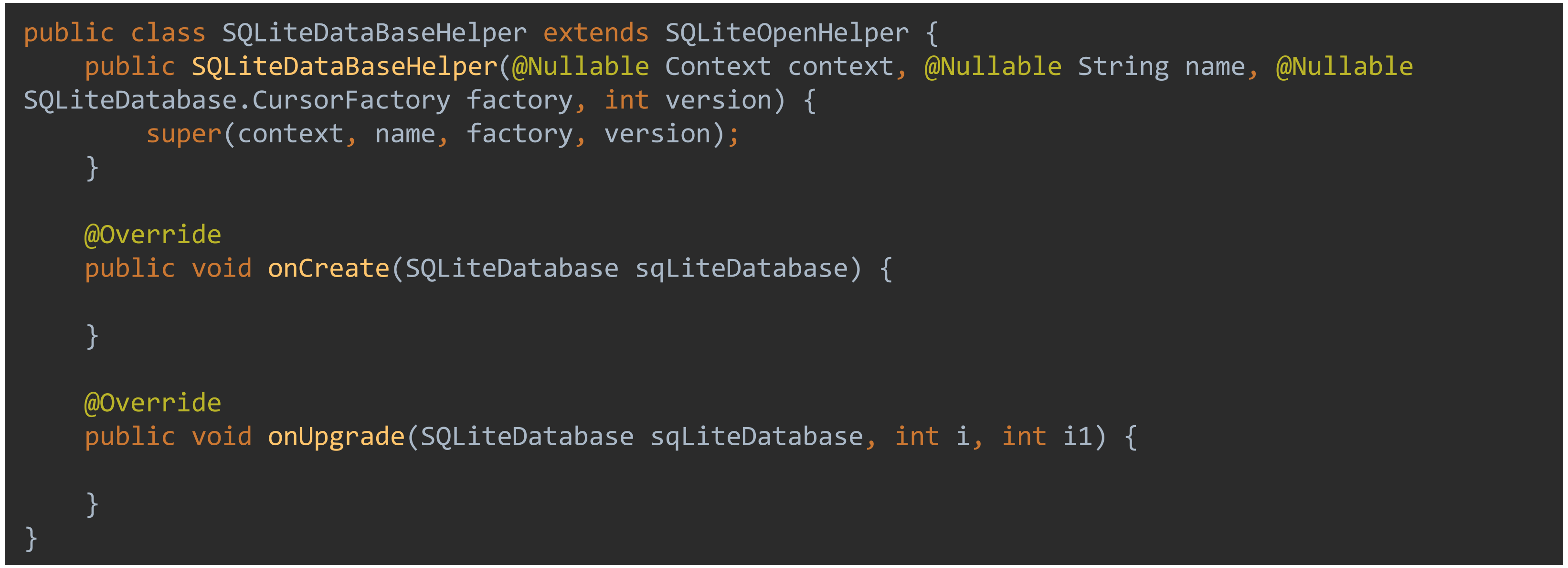
SQLiteOpenHelper(context:Context, name:String, f

SQLiteOpenHelper(context:Context, name:String, f

SQLiteOpenHelper(context:Context, name:String, v

□ Copy JavaDoc

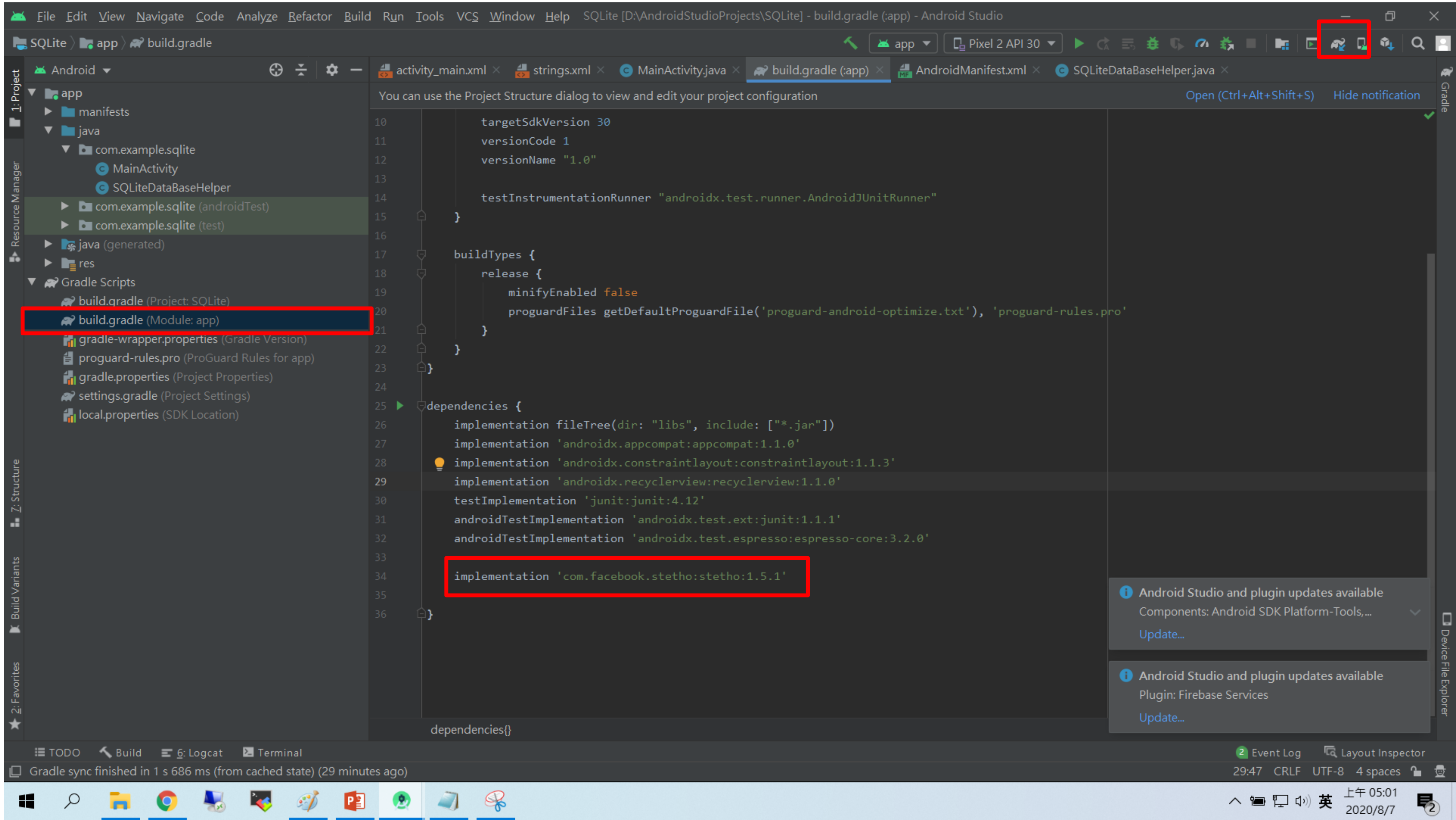
SQLiteDataBaseHelper



```java
public class SQLiteDataBaseHelper extends SQLiteOpenHelper {
    public SQLiteDataBaseHelper(@Nullable Context context, @Nullable String name, @Nullable
SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {

    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

    }
}
```

Create Table

1. Add dependencies
2. Click build.gradle
3. Add implementation 'com.facebook.stetho:stetho:1.5.1’
4. Sync now


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SQLite [D:\AndroidStudioProjects\SQLite] - build.gradle (:app) - Android Studio
SQLite 〉 app 〉 build.gradle
app
Pixel 2 API 30
Android
app
manifests
java
com.example.sqlite
MainActivity
SQLiteDataBaseHelper
com.example.sqlite (androidTest)
com.example.sqlite (test)
java (generated)
res
Gradle Scripts
build.gradle (Project: SQLite)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
activity_main.xml
strings.xml
MainActivity.java
build.gradle (:app)
AndroidManifest.xml
SQLiteDataBaseHelper.java
1: Project
Resource Manager
Z: Structure
Build Variants
2: Favorites
Gradle
Device File Explorer
TODO
Build
6: Logcat
Terminal
2 Event Log
Layout Inspector
29:47 CRLF UTF-8 4 spaces
上午 05:01
2020/8/7
英

You can use the Project Structure dialog to view and edit your project configuration Open (Ctrl+Alt+Shift+S) Hide notification

```
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: "libs", include: ["*.jar"])
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'androidx.recyclerview:recyclerview:1.1.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'

    implementation 'com.facebook.stetho:stetho:1.5.1'

}
```

dependencies{}

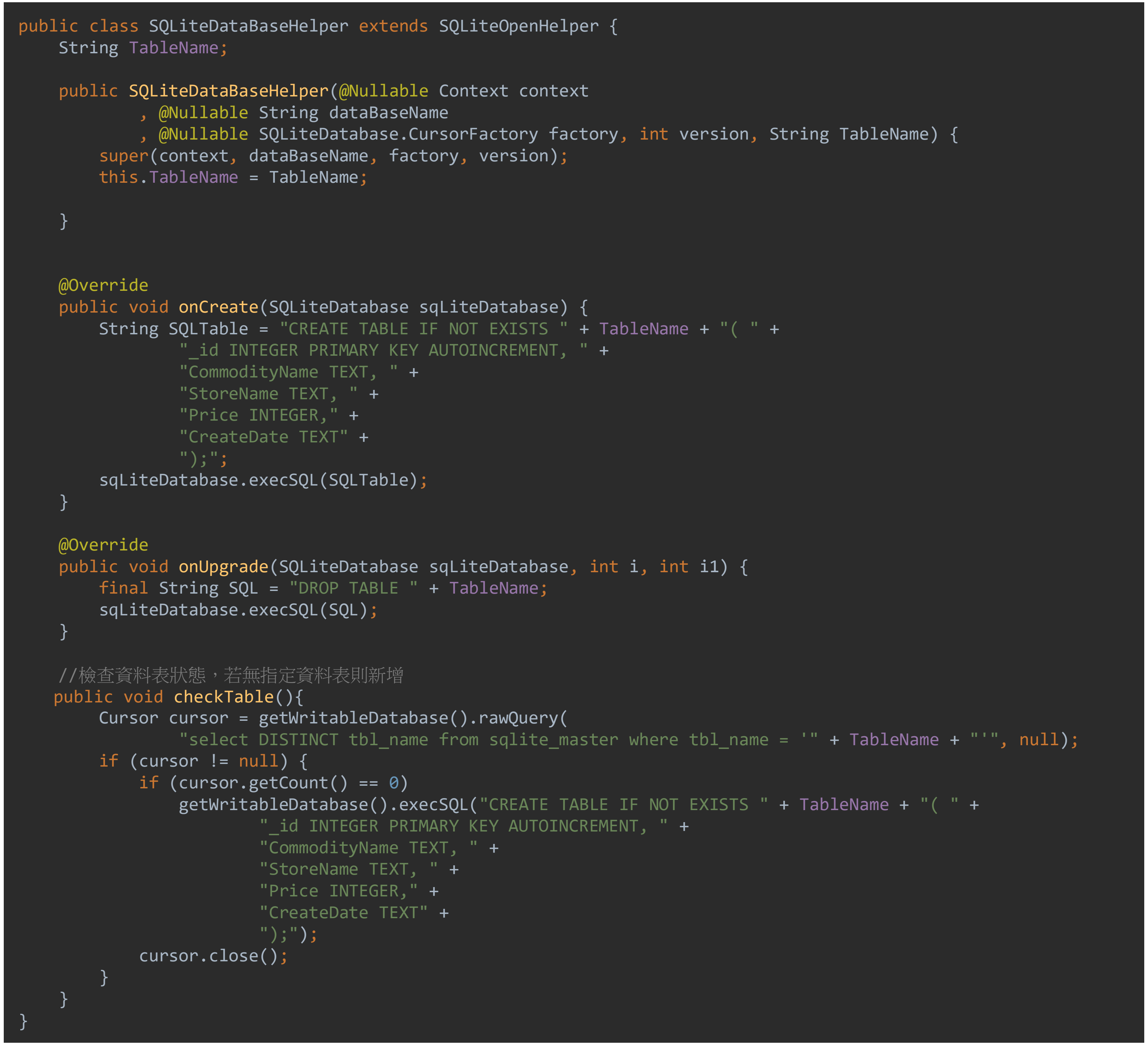
Android Studio and plugin updates available
Components: Android SDK Platform-Tools,...
Update...

Android Studio and plugin updates available
Plugin: Firebase Services
Update...

Gradle sync finished in 1 s 686 ms (from cached state) (29 minutes ago)

Create Table

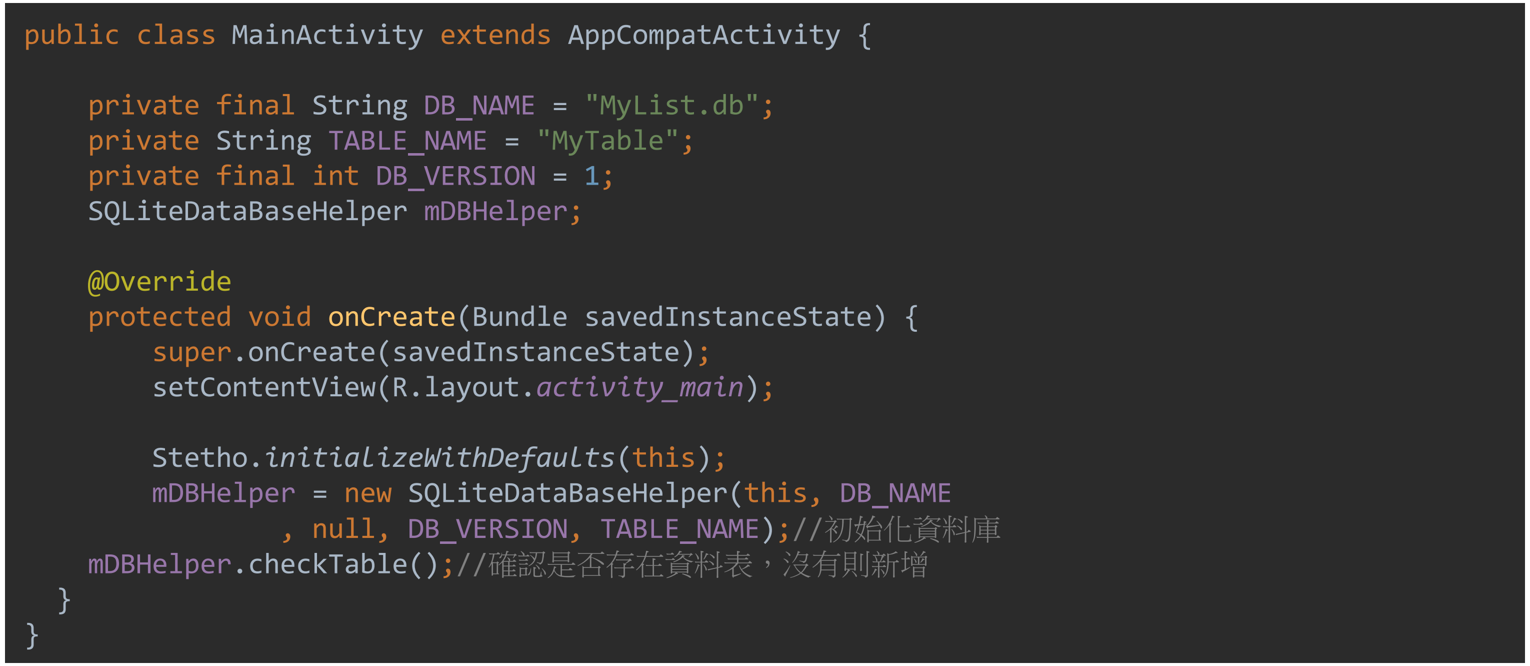
SQLiteDataBaseHelper.class



```
public class SQLiteDataBaseHelper extends SQLiteOpenHelper {
    String TableName;

    public SQLiteDataBaseHelper(@Nullable Context context
            , @Nullable String dataBaseName
            , @Nullable SQLiteDatabase.CursorFactory factory, int version, String TableName) {
        super(context, dataBaseName, factory, version);
        this.TableName = TableName;

    }


    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        String SQLTable = "CREATE TABLE IF NOT EXISTS " + TableName + "( " +
                "_id INTEGER PRIMARY KEY AUTOINCREMENT, " +
                "CommodityName TEXT, " +
                "StoreName TEXT, " +
                "Price INTEGER," +
                "CreateDate TEXT" +
                ");";
        sqLiteDatabase.execSQL(SQLTable);
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
        final String SQL = "DROP TABLE " + TableName;
        sqLiteDatabase.execSQL(SQL);
    }

    //檢查資料表狀態，若無指定資料表則新增
    public void checkTable(){
        Cursor cursor = getWritableDatabase().rawQuery(
                "select DISTINCT tbl_name from sqlite_master where tbl_name = '" + TableName + "'", null);
        if (cursor != null) {
            if (cursor.getCount() == 0)
                getWritableDatabase().execSQL("CREATE TABLE IF NOT EXISTS " + TableName + "( " +
                        "_id INTEGER PRIMARY KEY AUTOINCREMENT, " +
                        "CommodityName TEXT, " +
                        "StoreName TEXT, " +
                        "Price INTEGER," +
                        "CreateDate TEXT" +
                        ");");
            cursor.close();
        }
    }
}
```

Create Table

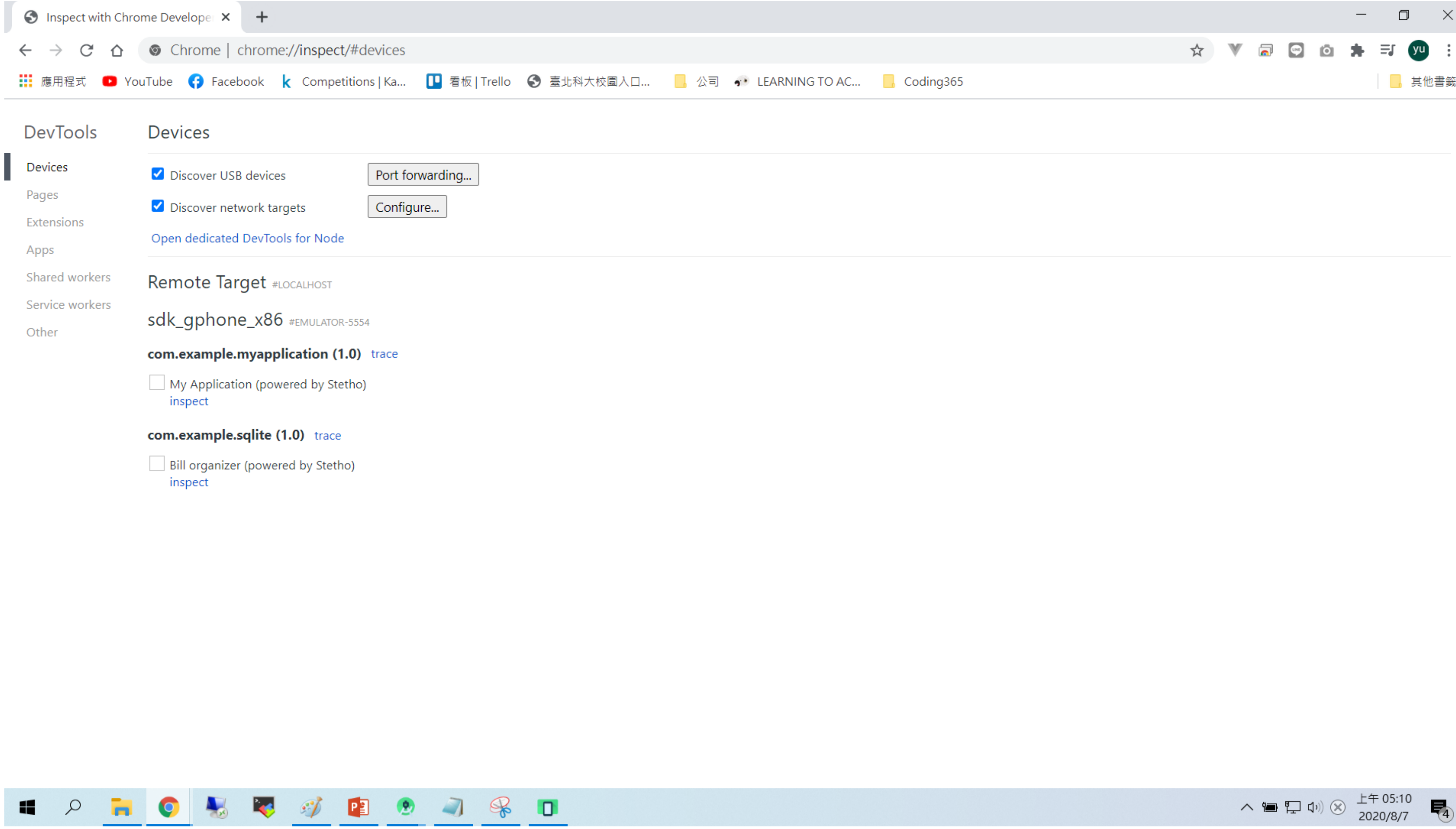
MainActivity.java



```java
public class MainActivity extends AppCompatActivity {

    private final String DB_NAME = "MyList.db";
    private String TABLE_NAME = "MyTable";
    private final int DB_VERSION = 1;
    SQLiteDataBaseHelper mDBHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Stetho.initializeWithDefaults(this);
        mDBHelper = new SQLiteDataBaseHelper(this, DB_NAME
                , null, DB_VERSION, TABLE_NAME);//初始化資料庫
    mDBHelper.checkTable();//確認是否存在資料表，沒有則新增
  }
}
```

Create Table

1. Compile the sample code
2. Open browser and open the page "chrome://inspect/#devices"



Devices

☑ Discover USB devices

☑ Discover network targets

Open dedicated DevTools for Node

Remote Target #LOCALHOST

sdk_gphone_x86 #EMULATOR-5554

com.example.myapplication (1.0) trace

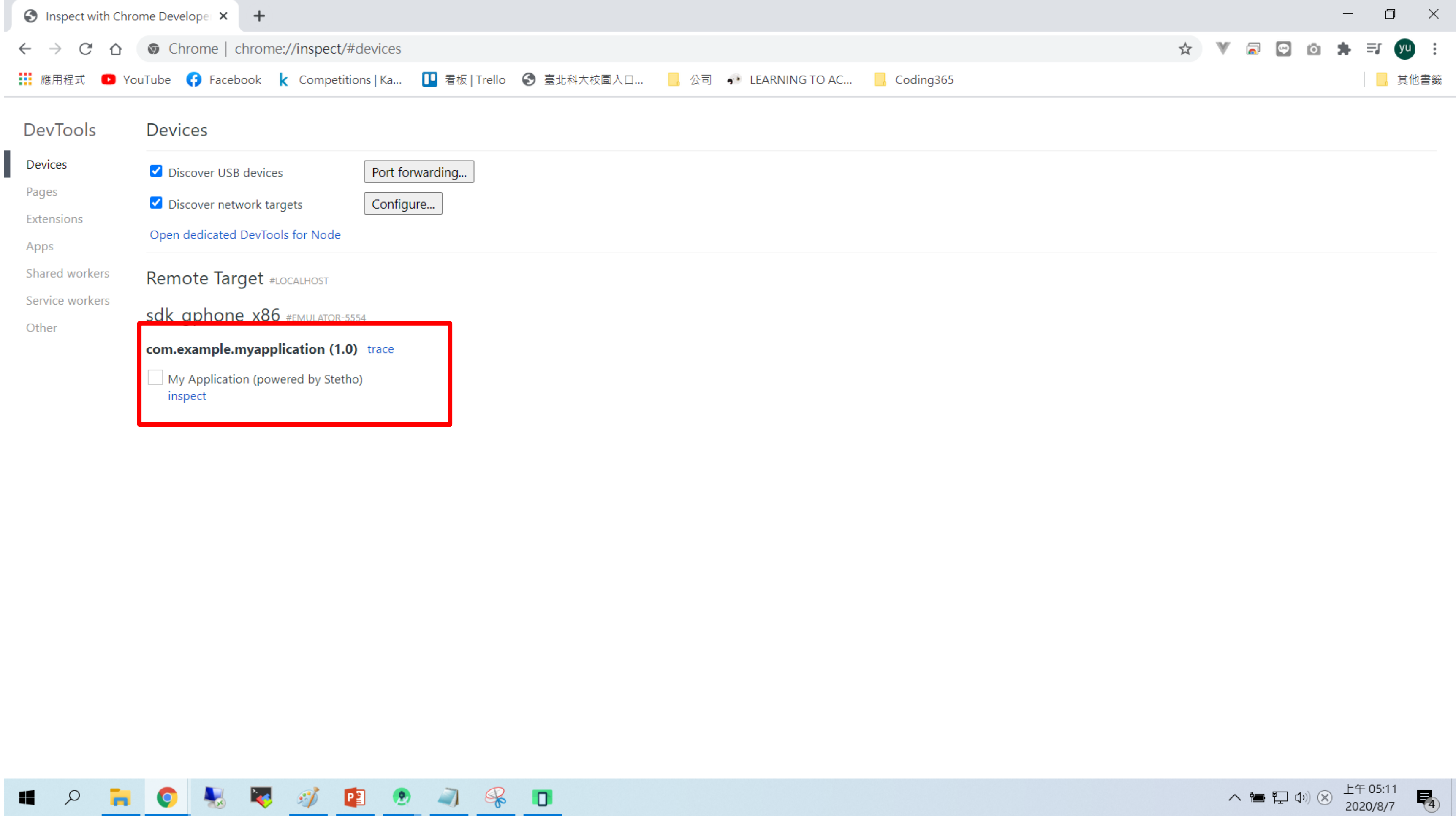
□ My Application (powered by Stetho)
inspect

com.example.sqlite (1.0) trace

□ Bill organizer (powered by Stetho)
inspect

Create Table

1. Choose your device
2. Click inspect



Devices

☑ Discover USB devices [Port forwarding...]

☑ Discover network targets [Configure...]

Open dedicated DevTools for Node

Remote Target #LOCALHOST

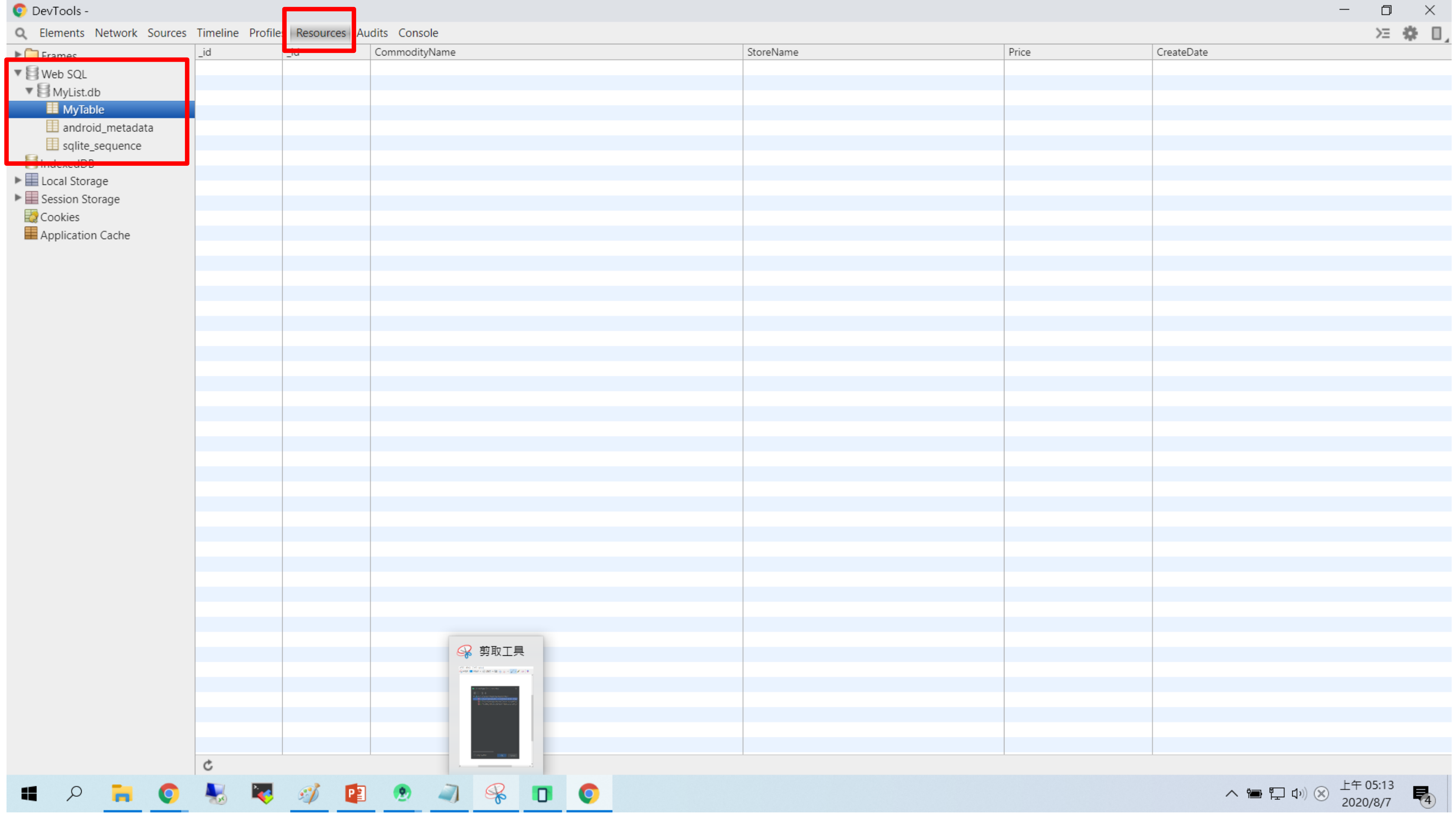
sdk_gphone_x86 #EMULATOR-5554

com.example.myapplication (1.0) trace

□ My Application (powered by Stetho)
inspect

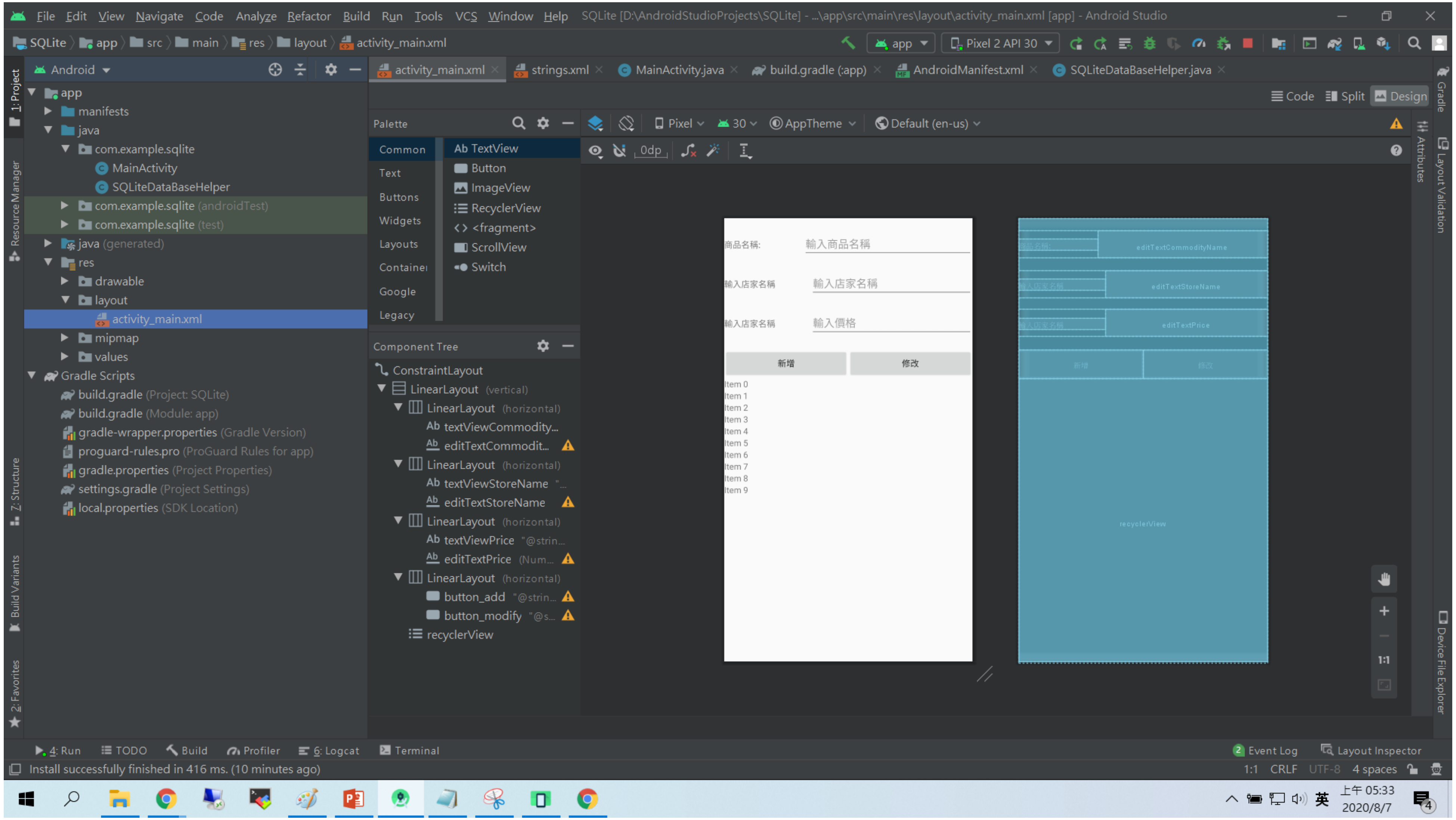
Create Table

1. Click Resource
2. Click Web SQL
3. Select your Table



Complete Code

activity_main.xml



SQLite [D:\AndroidStudioProjects\SQLite] - ...\app\src\main\res\layout\activity_main.xml [app] - Android Studio

Install successfully finished in 416 ms. (10 minutes ago)

Complete Code

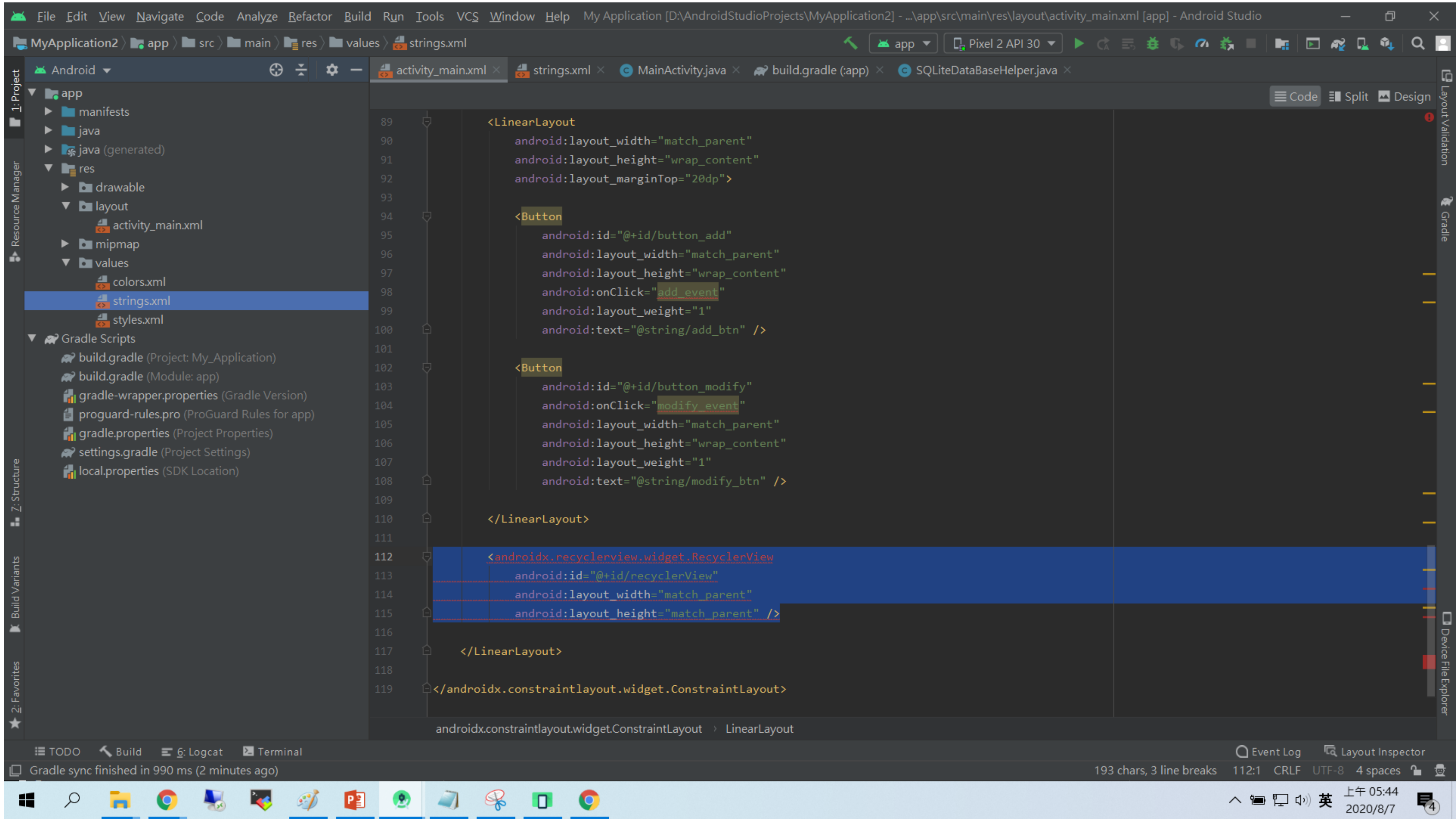
activity_main.xml & string.xml

Download for google drive
Paste it and cover your old layout file.

Complete Code

activity_main.xml

You will see a error part.
Don't worry, you just add
Dependencies about it.



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
My Application [D:\AndroidStudioProjects\MyApplication2] - ...\app\src\main\res\layout\activity_main.xml [app] - Android Studio
MyApplication2 › app › src › main › res › values › strings.xml
app
Pixel 2 API 30
Android
activity_main.xml
strings.xml
MainActivity.java
build.gradle (:app)
SQLiteDataBaseHelper.java
Code
Split
Design
app
manifests
java
java (generated)
res
drawable
layout
activity_main.xml
mipmap
values
colors.xml
strings.xml
styles.xml
Gradle Scripts
build.gradle (Project: My_Application)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
1: Project
Resource Manager
Z: Structure
Build Variants
2: Favorites
Layout Validation
Gradle
Device File Explorer
androidx.constraintlayout.widget.ConstraintLayout › LinearLayout
TODO
Build
6: Logcat
Terminal
Event Log
Layout Inspector
193 chars, 3 line breaks
112:1
CRLF
UTF-8
4 spaces
上午 05:44
2020/8/7

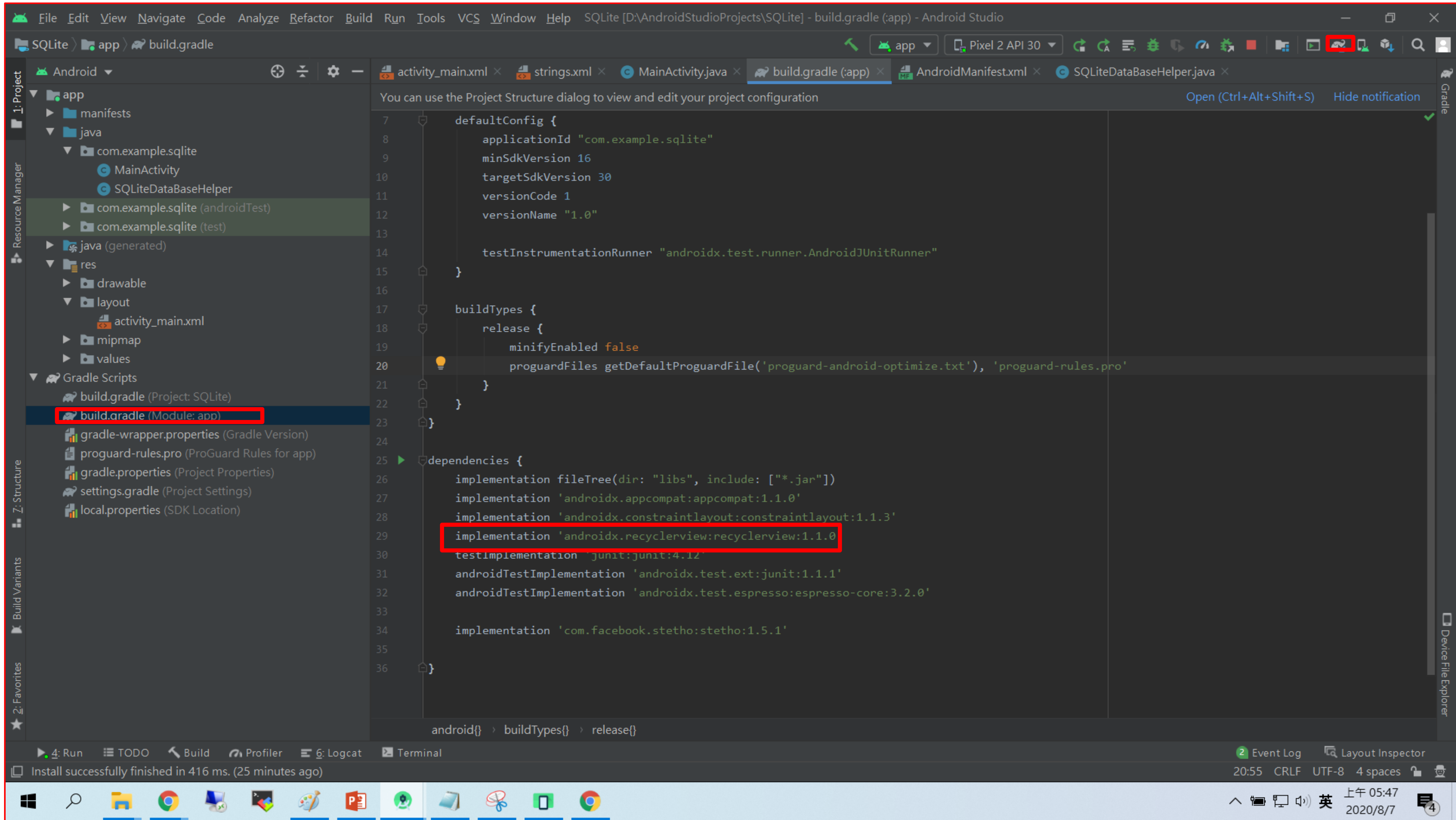
```
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp">

            <Button
                android:id="@+id/button_add"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:onClick="add_event"
                android:layout_weight="1"
                android:text="@string/add_btn" />

            <Button
                android:id="@+id/button_modify"
                android:onClick="modify_event"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="@string/modify_btn" />

        </LinearLayout>

        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/recyclerView"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />

    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Gradle sync finished in 990 ms (2 minutes ago)

Complete Code

activity_main.xml

1. Open build.gradle
2. Add implementation 'androidx.recyclerview:recyclerview:1.1.0’
3. Sync now



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SQLite [D:\AndroidStudioProjects\SQLite] - build.gradle (:app) - Android Studio
SQLite › app › build.gradle
app
Pixel 2 API 30
Android
app
manifests
java
com.example.sqlite
MainActivity
SQLiteDataBaseHelper
com.example.sqlite (androidTest)
com.example.sqlite (test)
java (generated)
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
build.gradle (Project: SQLite)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
activity_main.xml
strings.xml
MainActivity.java
build.gradle (:app)
AndroidManifest.xml
SQLiteDataBaseHelper.java
Open (Ctrl+Alt+Shift+S)
Hide notification
android{} › buildTypes{} › release{}
4: Run
TODO
Build
Profiler
6: Logcat
Terminal
Event Log
Layout Inspector
20:55 CRLF UTF-8 4 spaces
1: Project
Resource Manager
Z: Structure
Build Variants
2: Favorites
Gradle
Device File Explorer
上午 05:47
2020/8/7

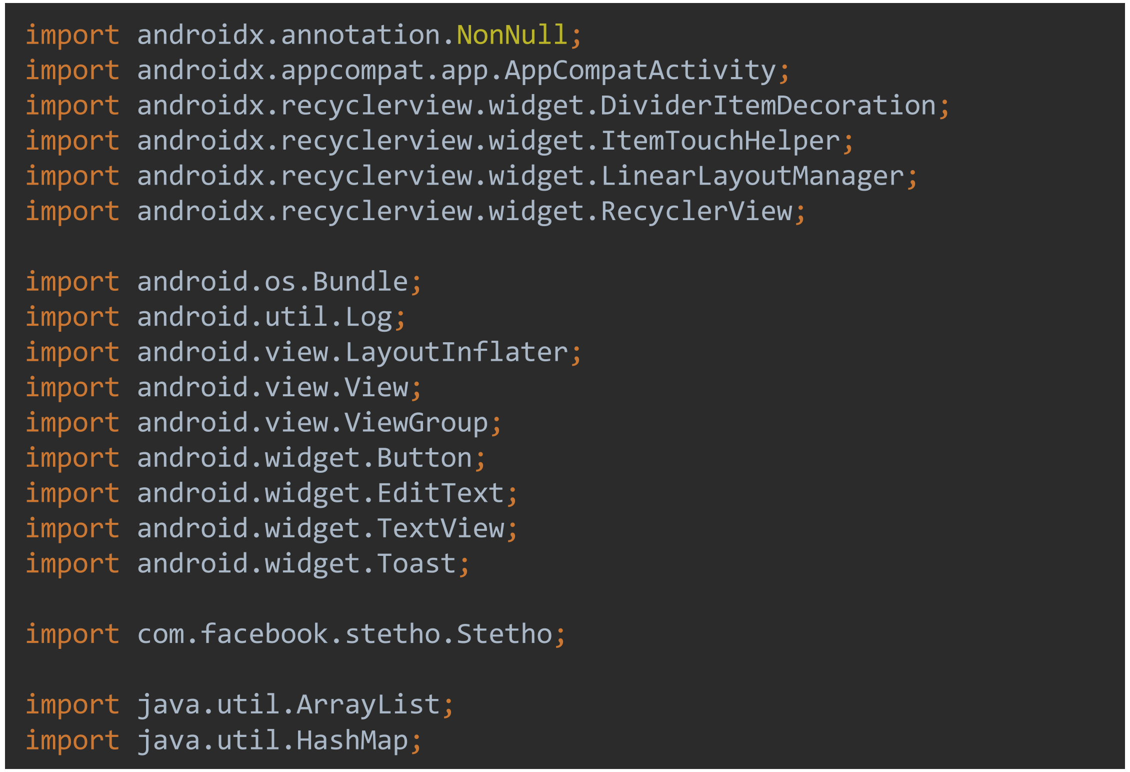
You can use the Project Structure dialog to view and edit your project configuration

```
    defaultConfig {
        applicationId "com.example.sqlite"
        minSdkVersion 16
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: "libs", include: ["*.jar"])
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'androidx.recyclerview:recyclerview:1.1.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'

    implementation 'com.facebook.stetho:stetho:1.5.1'

}
```

Install successfully finished in 416 ms. (25 minutes ago)

Complete Code

1. MainActivity.java



```java
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.DividerItemDecoration;
import androidx.recyclerview.widget.ItemTouchHelper;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.facebook.stetho.Stetho;

import java.util.ArrayList;
import java.util.HashMap;
```

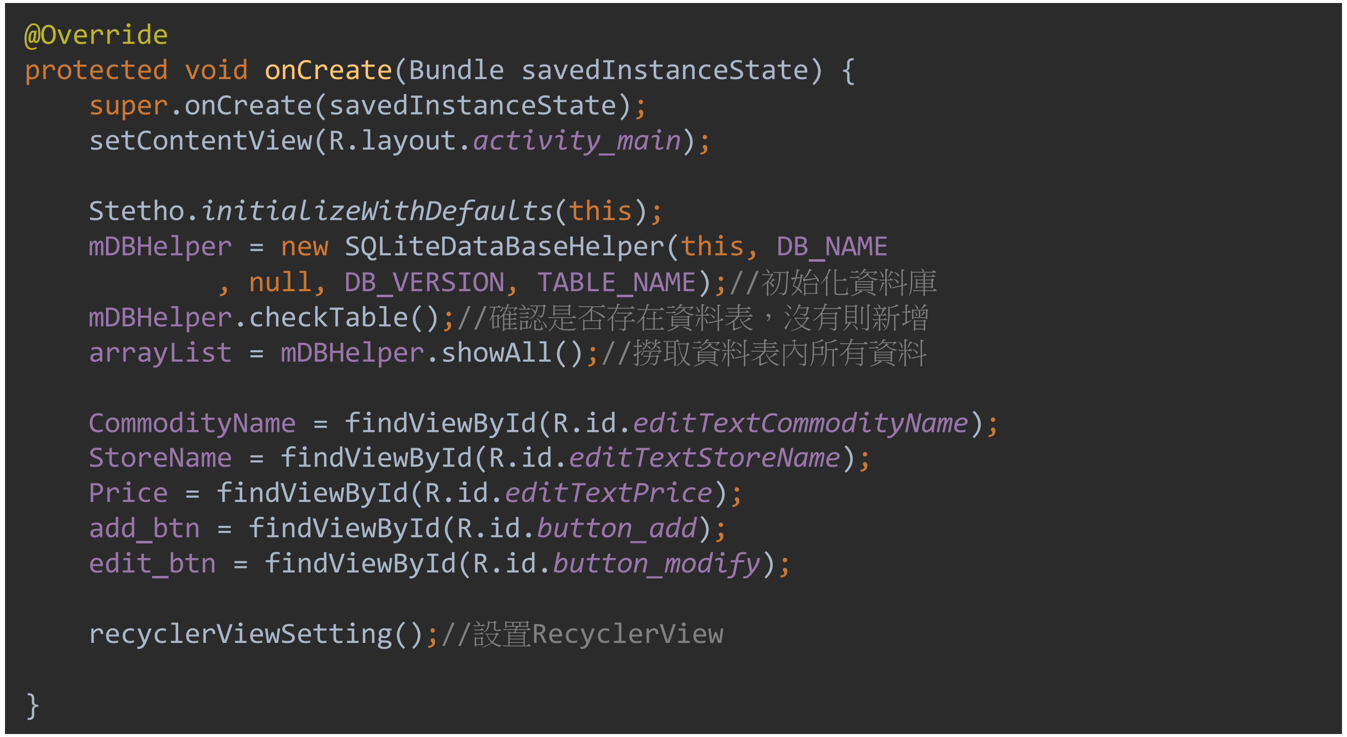
Complete Code

1. MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    String TAG = MainActivity.class.getSimpleName() + "My";

    private final String DB_NAME = "MyList.db";
    private String TABLE_NAME = "MyTable";
    private final int DB_VERSION = 1;
    SQLiteDataBaseHelper mDBHelper;

    ArrayList<HashMap<String, String>> arrayList = new ArrayList<>();//取得所有資料
    ArrayList<HashMap<String, String>> getNowArray = new ArrayList<>();//取得被選中的項目資料

    EditText CommodityName, StoreName, Price;
    Button add_btn, edit_btn;
    MyAdapter myAdapter;
            …
}
```

Complete Code

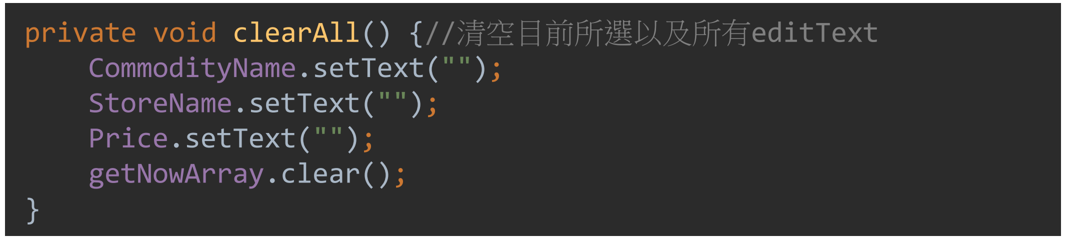
1. MainActivity.java



```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    Stetho.initializeWithDefaults(this);
    mDBHelper = new SQLiteDataBaseHelper(this, DB_NAME
            , null, DB_VERSION, TABLE_NAME);//初始化資料庫
    mDBHelper.checkTable();//確認是否存在資料表，沒有則新增
    arrayList = mDBHelper.showAll();//撈取資料表內所有資料

    CommodityName = findViewById(R.id.editTextCommodityName);
    StoreName = findViewById(R.id.editTextStoreName);
    Price = findViewById(R.id.editTextPrice);
    add_btn = findViewById(R.id.button_add);
    edit_btn = findViewById(R.id.button_modify);

    recyclerViewSetting();//設置RecyclerView

}
```

Complete Code

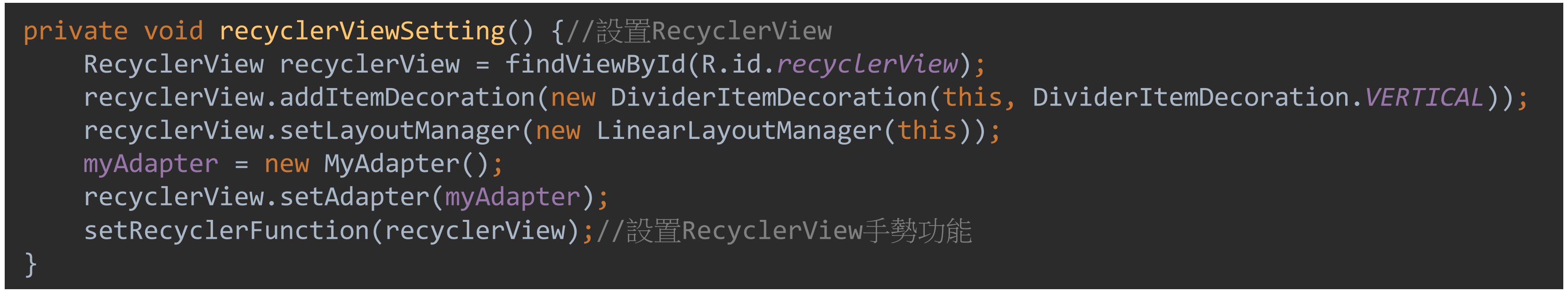
1. MainActivity.java



```
private void clearAll() {//清空目前所選以及所有editText
    CommodityName.setText("");
    StoreName.setText("");
    Price.setText("");
    getNowArray.clear();
}
```

Complete Code

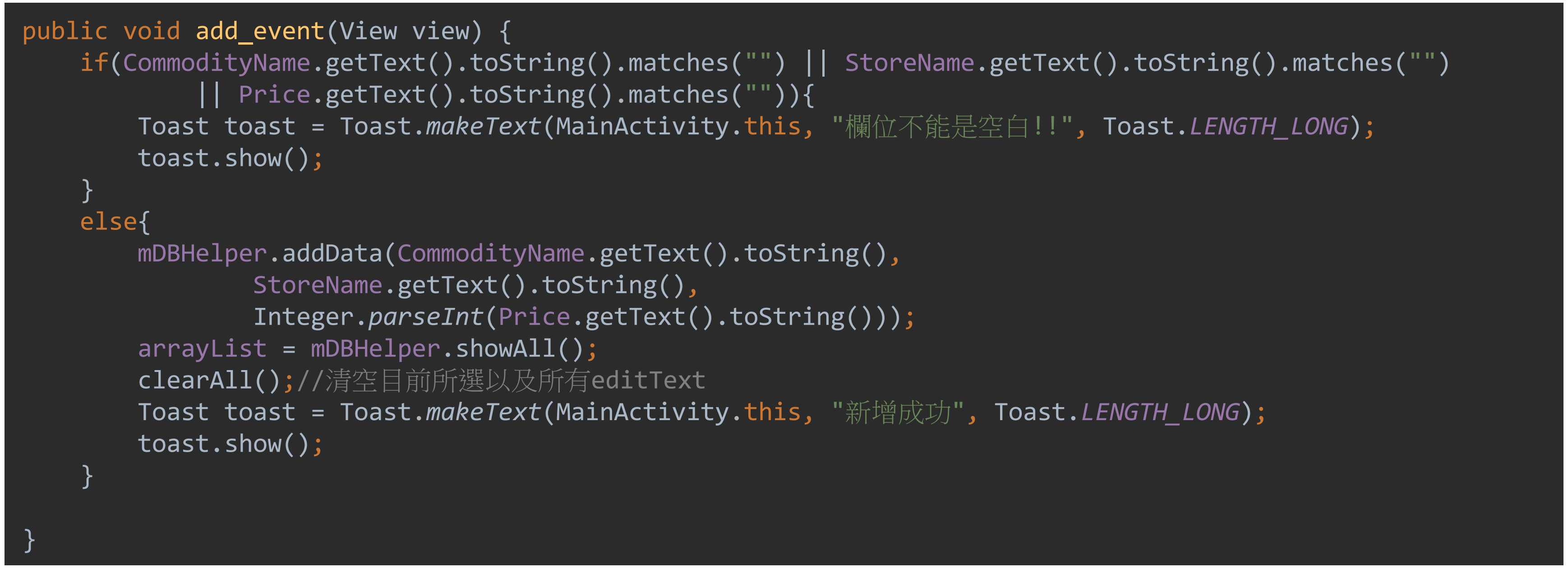
1. MainActivity.java



```java
private void recyclerViewSetting() {//設置RecyclerView
    RecyclerView recyclerView = findViewById(R.id.recyclerView);
    recyclerView.addItemDecoration(new DividerItemDecoration(this, DividerItemDecoration.VERTICAL));
    recyclerView.setLayoutManager(new LinearLayoutManager(this));
    myAdapter = new MyAdapter();
    recyclerView.setAdapter(myAdapter);
    setRecyclerFunction(recyclerView);//設置RecyclerView手勢功能
}
```

Complete Code

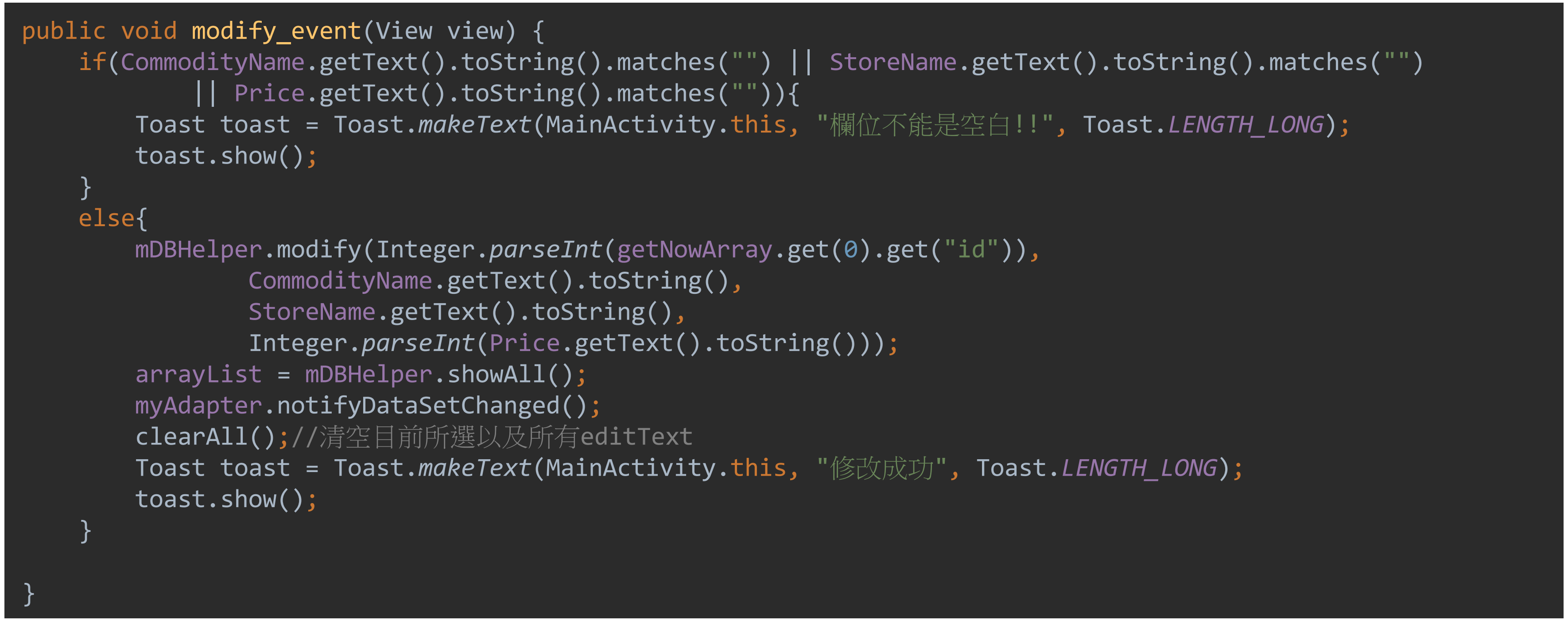
1. MainActivity.java



```
public void add_event(View view) {
    if(CommodityName.getText().toString().matches("") || StoreName.getText().toString().matches("")
            || Price.getText().toString().matches("")){
        Toast toast = Toast.makeText(MainActivity.this, "欄位不能是空白！！", Toast.LENGTH_LONG);
        toast.show();
    }
    else{
        mDBHelper.addData(CommodityName.getText().toString(),
                StoreName.getText().toString(),
                Integer.parseInt(Price.getText().toString()));
        arrayList = mDBHelper.showAll();
        clearAll();//清空目前所選以及所有editText
        Toast toast = Toast.makeText(MainActivity.this, "新增成功", Toast.LENGTH_LONG);
        toast.show();
    }

}
```

Complete Code

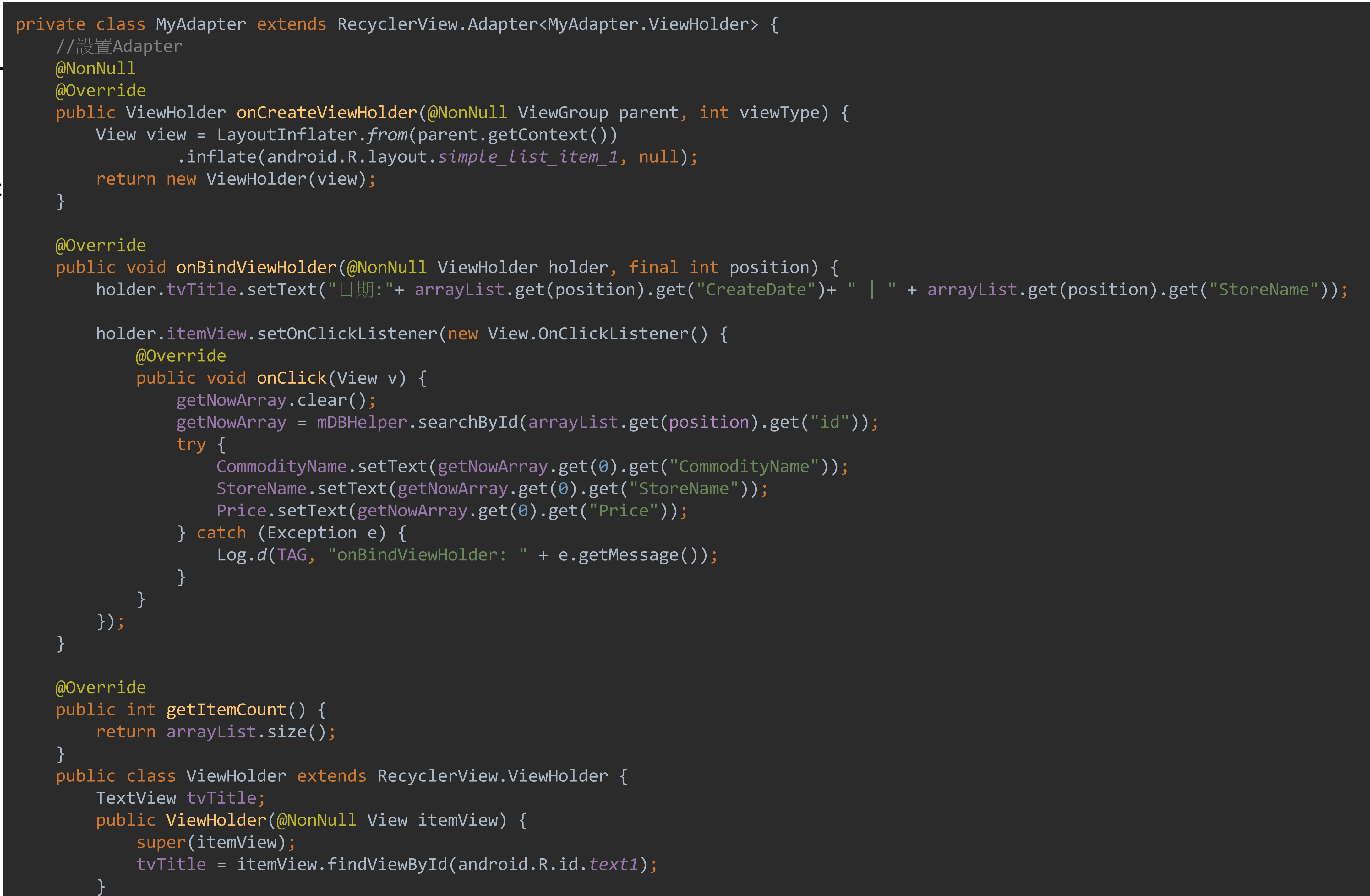
1. MainActivity.java



```java
public void modify_event(View view) {
    if(CommodityName.getText().toString().matches("") || StoreName.getText().toString().matches("")
            || Price.getText().toString().matches("")){
        Toast toast = Toast.makeText(MainActivity.this, "欄位不能是空白!!", Toast.LENGTH_LONG);
        toast.show();
    }
    else{
        mDBHelper.modify(Integer.parseInt(getNowArray.get(0).get("id")),
                CommodityName.getText().toString(),
                StoreName.getText().toString(),
                Integer.parseInt(Price.getText().toString()));
        arrayList = mDBHelper.showAll();
        myAdapter.notifyDataSetChanged();
        clearAll();//清空目前所選以及所有editText
        Toast toast = Toast.makeText(MainActivity.this, "修改成功", Toast.LENGTH_LONG);
        toast.show();
    }

}
```

Complet

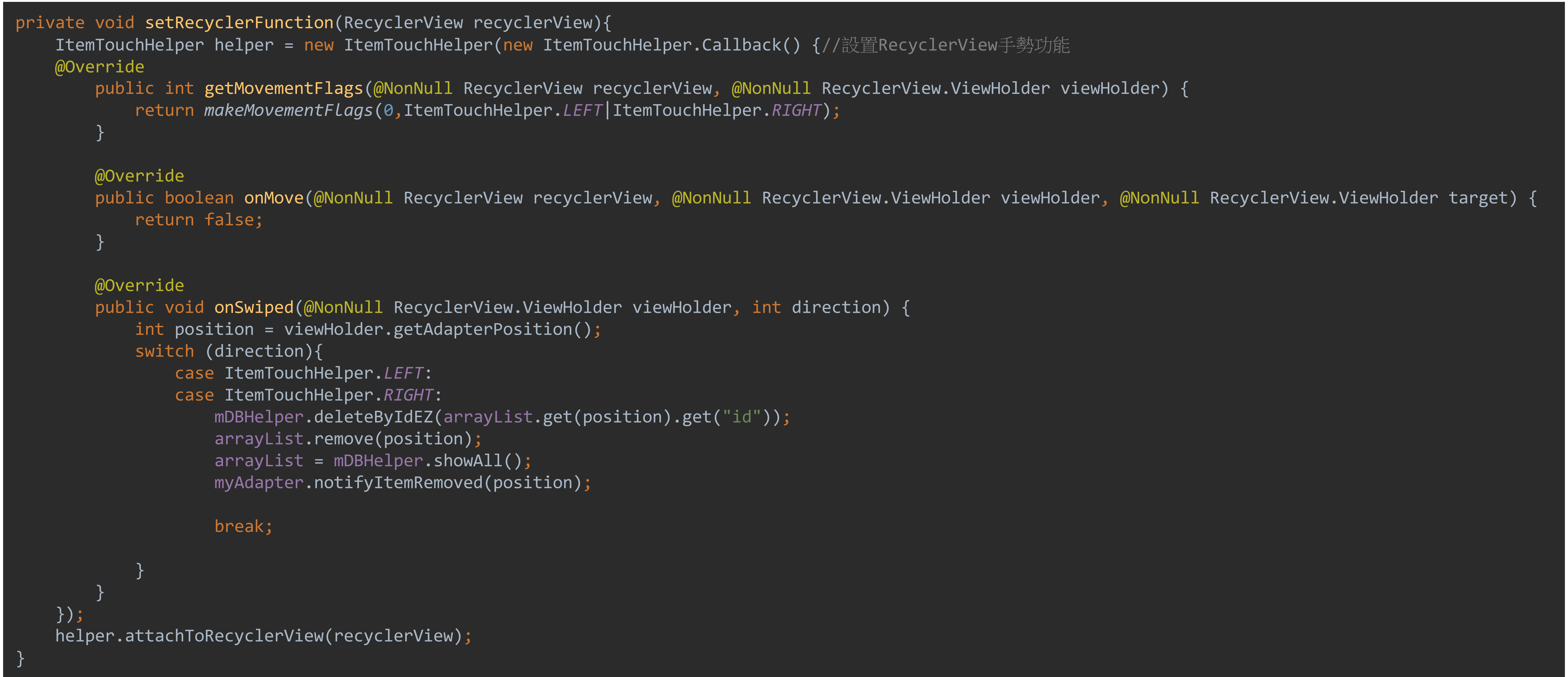
1. MainAc



```
private class MyAdapter extends RecyclerView.Adapter<MyAdapter.ViewHolder> {
    //設置Adapter
    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext())
                .inflate(android.R.layout.simple_list_item_1, null);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, final int position) {
        holder.tvTitle.setText("日期:"+ arrayList.get(position).get("CreateDate")+ " | " + arrayList.get(position).get("StoreName"));

        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                getNowArray.clear();
                getNowArray = mDBHelper.searchById(arrayList.get(position).get("id"));
                try {
                    CommodityName.setText(getNowArray.get(0).get("CommodityName"));
                    StoreName.setText(getNowArray.get(0).get("StoreName"));
                    Price.setText(getNowArray.get(0).get("Price"));
                } catch (Exception e) {
                    Log.d(TAG, "onBindViewHolder: " + e.getMessage());
                }
            }
        });
    }

    @Override
    public int getItemCount() {
        return arrayList.size();
    }
    public class ViewHolder extends RecyclerView.ViewHolder {
        TextView tvTitle;
        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            tvTitle = itemView.findViewById(android.R.id.text1);
        }
```

Complete Code

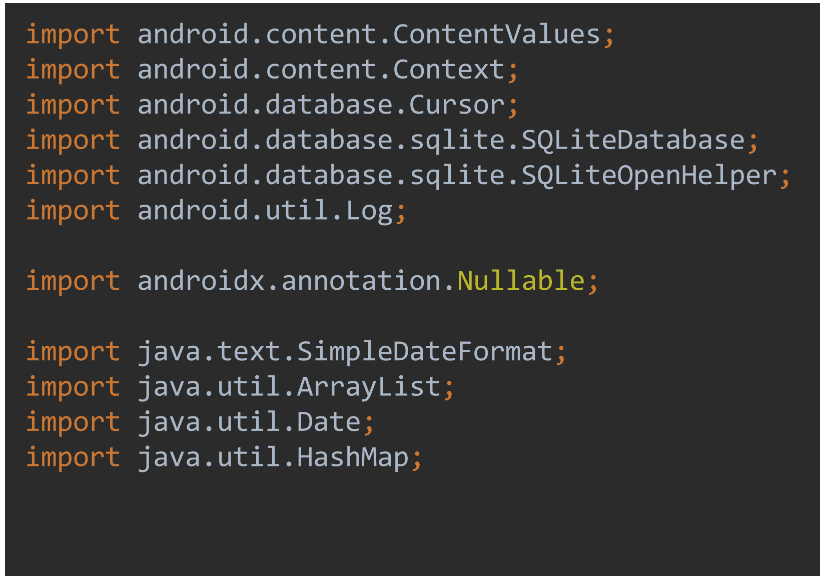
1. MainActivity.java



```java
private void setRecyclerFunction(RecyclerView recyclerView){
    ItemTouchHelper helper = new ItemTouchHelper(new ItemTouchHelper.Callback() {//設置RecyclerView手勢功能
    @Override
        public int getMovementFlags(@NonNull RecyclerView recyclerView, @NonNull RecyclerView.ViewHolder viewHolder) {
            return makeMovementFlags(0,ItemTouchHelper.LEFT|ItemTouchHelper.RIGHT);
        }

        @Override
        public boolean onMove(@NonNull RecyclerView recyclerView, @NonNull RecyclerView.ViewHolder viewHolder, @NonNull RecyclerView.ViewHolder target) {
            return false;
        }

        @Override
        public void onSwiped(@NonNull RecyclerView.ViewHolder viewHolder, int direction) {
            int position = viewHolder.getAdapterPosition();
            switch (direction){
                case ItemTouchHelper.LEFT:
                case ItemTouchHelper.RIGHT:
                    mDBHelper.deleteByIdEZ(arrayList.get(position).get("id"));
                    arrayList.remove(position);
                    arrayList = mDBHelper.showAll();
                    myAdapter.notifyItemRemoved(position);

                    break;

            }
        }
    });
    helper.attachToRecyclerView(recyclerView);
}
```

Complete Code

2. SQLiteDataBaseHelper.class



```java
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

import androidx.annotation.Nullable;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
```

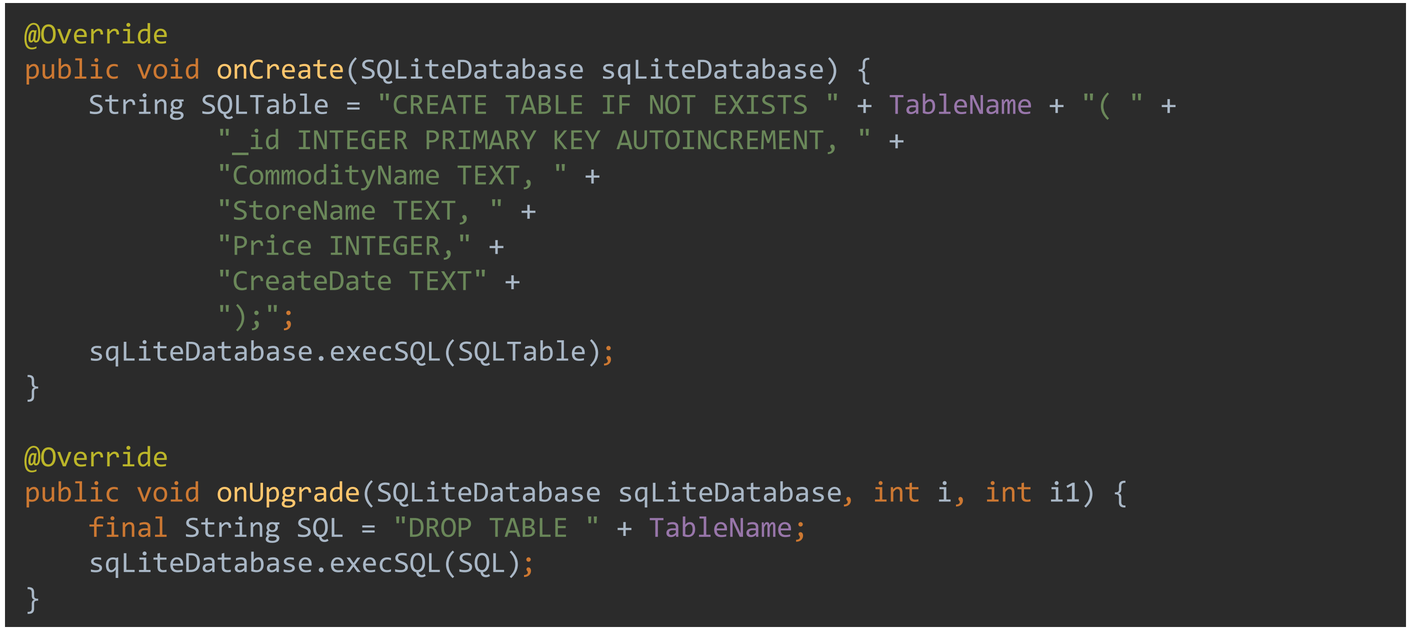
Complete Code

2. SQLiteDataBaseHelper.class

```
public class SQLiteDataBaseHelper extends SQLiteOpenHelper {
    String TableName;

    public SQLiteDataBaseHelper(@Nullable Context context
            , @Nullable String dataBaseName
            , @Nullable SQLiteDatabase.CursorFactory factory, int version, String TableName) {
        super(context, dataBaseName, factory, version);
        this.TableName = TableName;

    }
            …
}
```

Complete Code

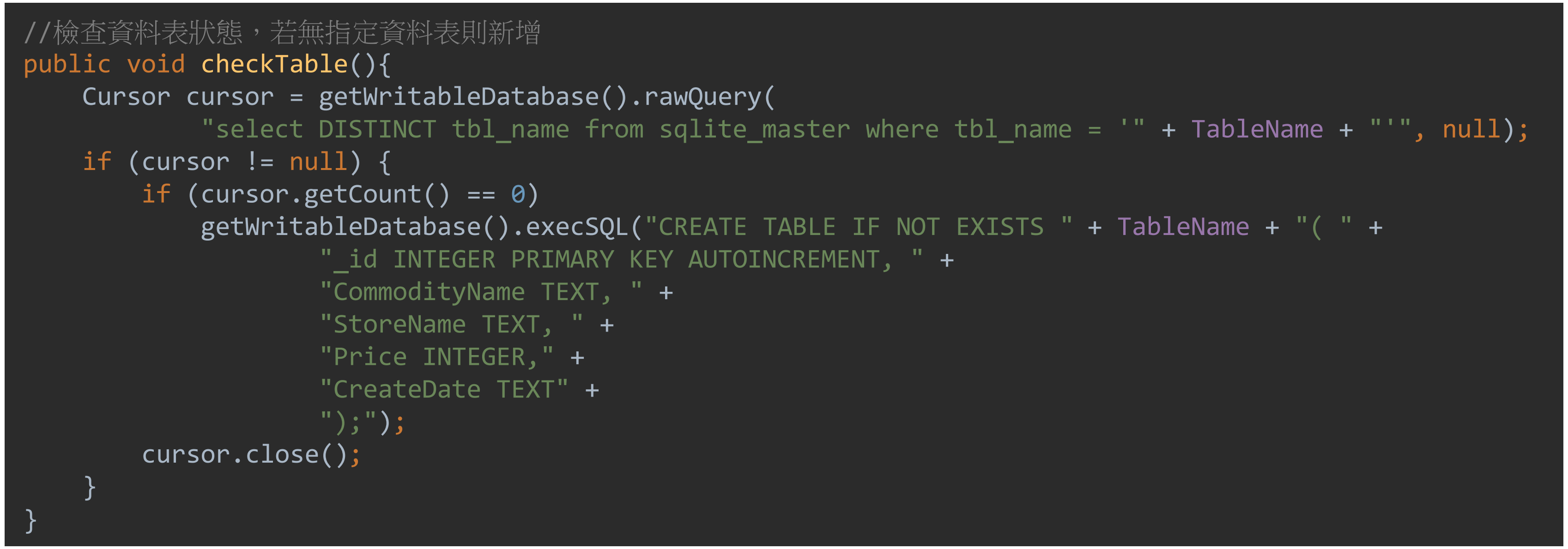
2. SQLiteDataBaseHelper.class



```
@Override
public void onCreate(SQLiteDatabase sqLiteDatabase) {
    String SQLTable = "CREATE TABLE IF NOT EXISTS " + TableName + "( " +
            "_id INTEGER PRIMARY KEY AUTOINCREMENT, " +
            "CommodityName TEXT, " +
            "StoreName TEXT, " +
            "Price INTEGER," +
            "CreateDate TEXT" +
            ");";
    sqLiteDatabase.execSQL(SQLTable);
}

@Override
public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
    final String SQL = "DROP TABLE " + TableName;
    sqLiteDatabase.execSQL(SQL);
}
```

Complete Code

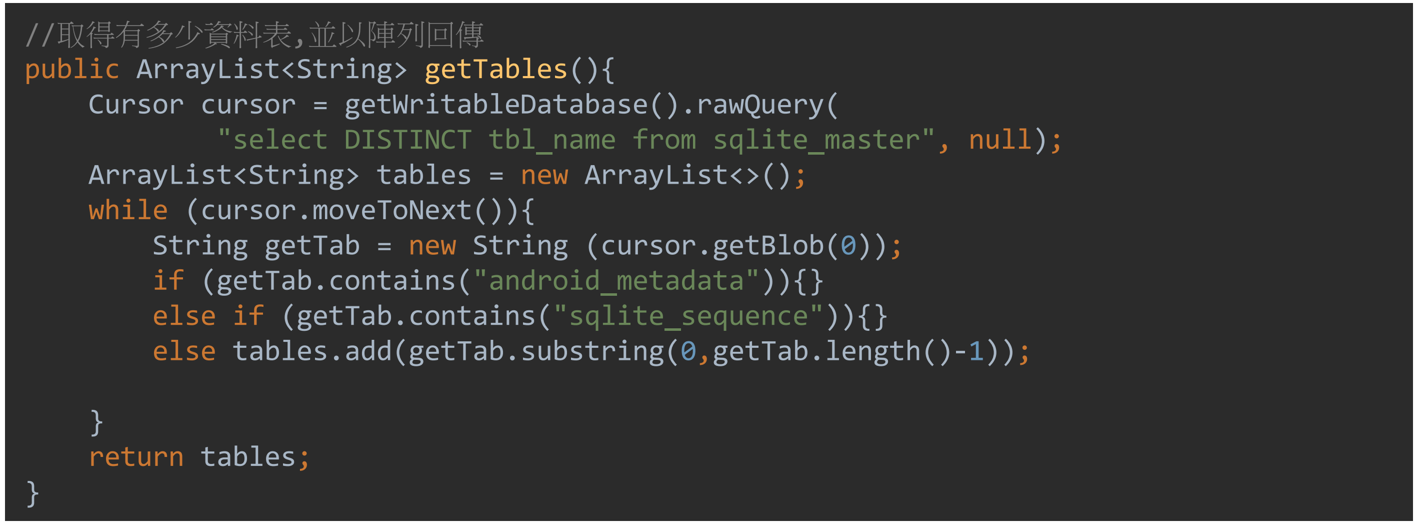
2. SQLiteDataBaseHelper.class



```
//檢查資料表狀態，若無指定資料表則新增
public void checkTable(){
    Cursor cursor = getWritableDatabase().rawQuery(
            "select DISTINCT tbl_name from sqlite_master where tbl_name = '" + TableName + "'", null);
    if (cursor != null) {
        if (cursor.getCount() == 0)
            getWritableDatabase().execSQL("CREATE TABLE IF NOT EXISTS " + TableName + "( " +
                    "_id INTEGER PRIMARY KEY AUTOINCREMENT, " +
                    "CommodityName TEXT, " +
                    "StoreName TEXT, " +
                    "Price INTEGER," +
                    "CreateDate TEXT" +
                    ");");
        cursor.close();
    }
}
```

Complete Code

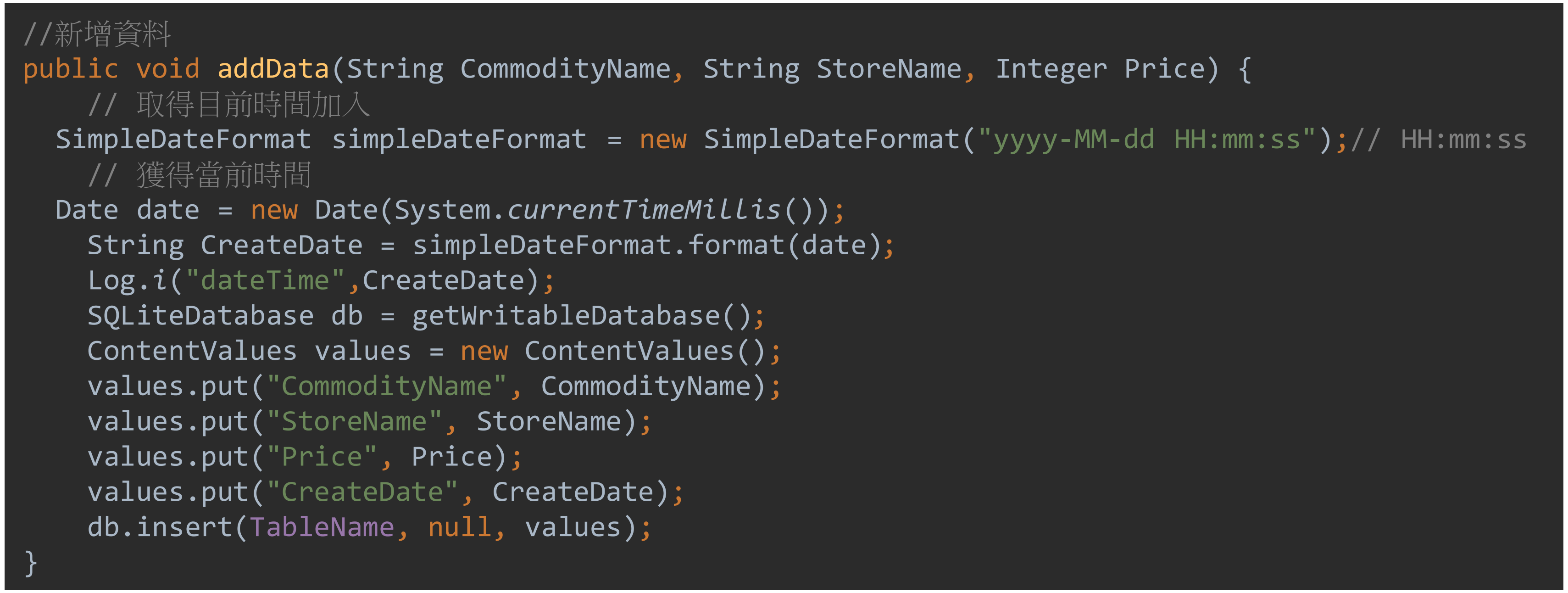
2. SQLiteDataBaseHelper.class



```java
//取得有多少資料表,並以陣列回傳
public ArrayList<String> getTables(){
    Cursor cursor = getWritableDatabase().rawQuery(
            "select DISTINCT tbl_name from sqlite_master", null);
    ArrayList<String> tables = new ArrayList<>();
    while (cursor.moveToNext()){
        String getTab = new String (cursor.getBlob(0));
        if (getTab.contains("android_metadata")){}
        else if (getTab.contains("sqlite_sequence")){}
        else tables.add(getTab.substring(0,getTab.length()-1));

    }
    return tables;
}
```

Complete Code

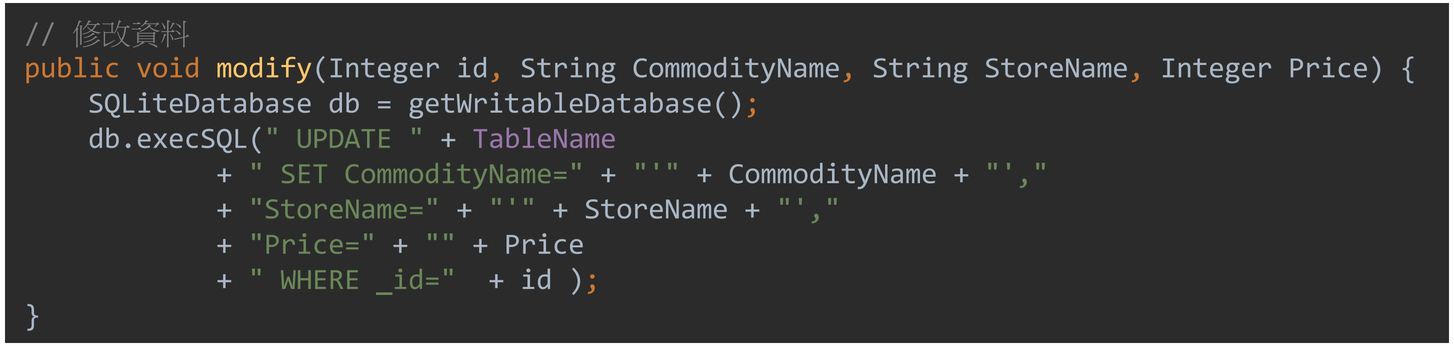
2. SQLiteDataBaseHelper.class



```java
//新增資料
public void addData(String CommodityName, String StoreName, Integer Price) {
    // 取得目前時間加入
  SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");// HH:mm:ss
    // 獲得當前時間
  Date date = new Date(System.currentTimeMillis());
    String CreateDate = simpleDateFormat.format(date);
    Log.i("dateTime",CreateDate);
    SQLiteDatabase db = getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("CommodityName", CommodityName);
    values.put("StoreName", StoreName);
    values.put("Price", Price);
    values.put("CreateDate", CreateDate);
    db.insert(TableName, null, values);
}
```

Complete Code

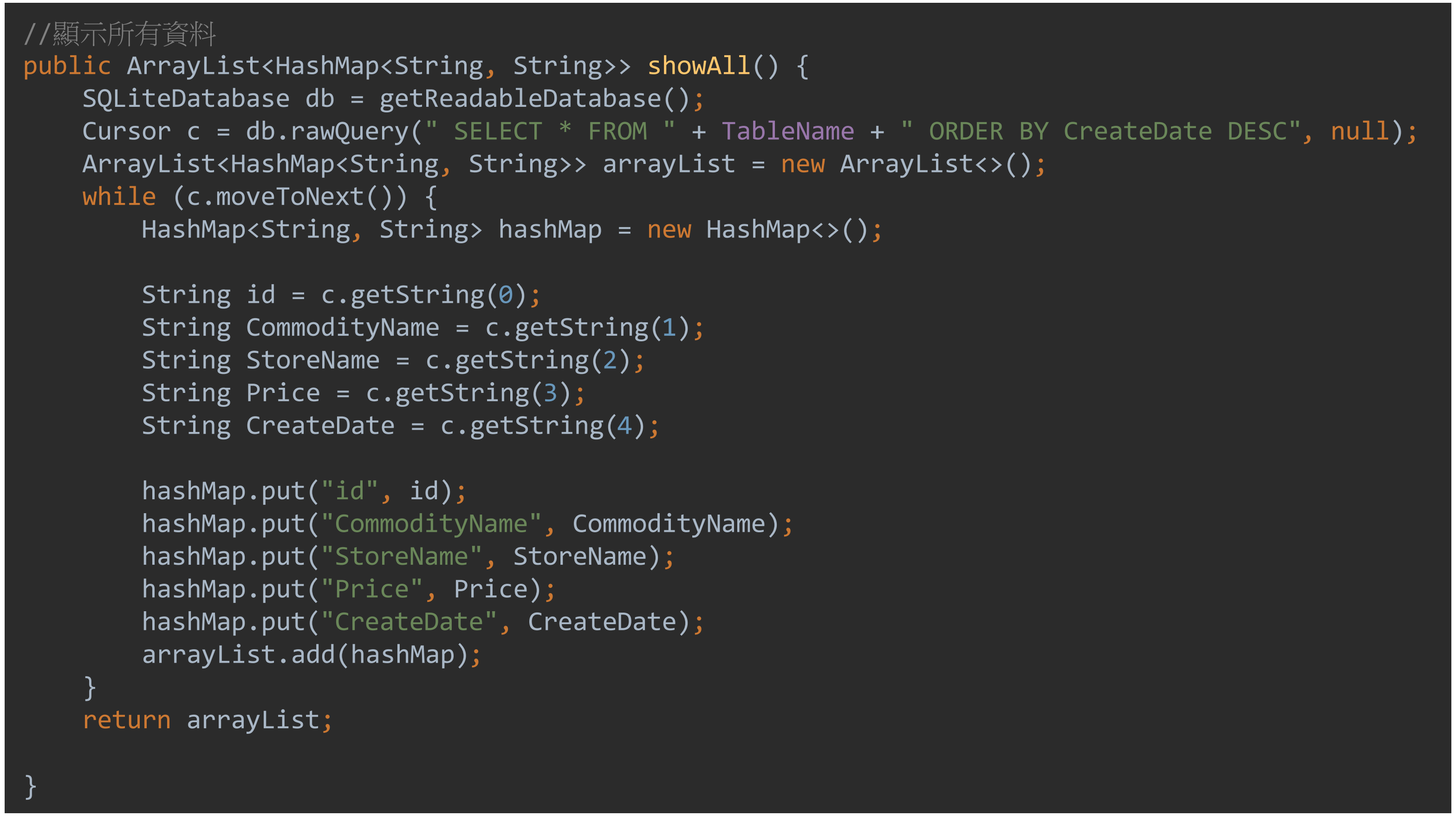
2. SQLiteDataBaseHelper.class



```
// 修改資料
public void modify(Integer id, String CommodityName, String StoreName, Integer Price) {
    SQLiteDatabase db = getWritableDatabase();
    db.execSQL(" UPDATE " + TableName
            + " SET CommodityName=" + "'" + CommodityName + "',"
            + "StoreName=" + "'" + StoreName + "',"
            + "Price=" + "" + Price
            + " WHERE _id="  + id );
}
```

Complete Code

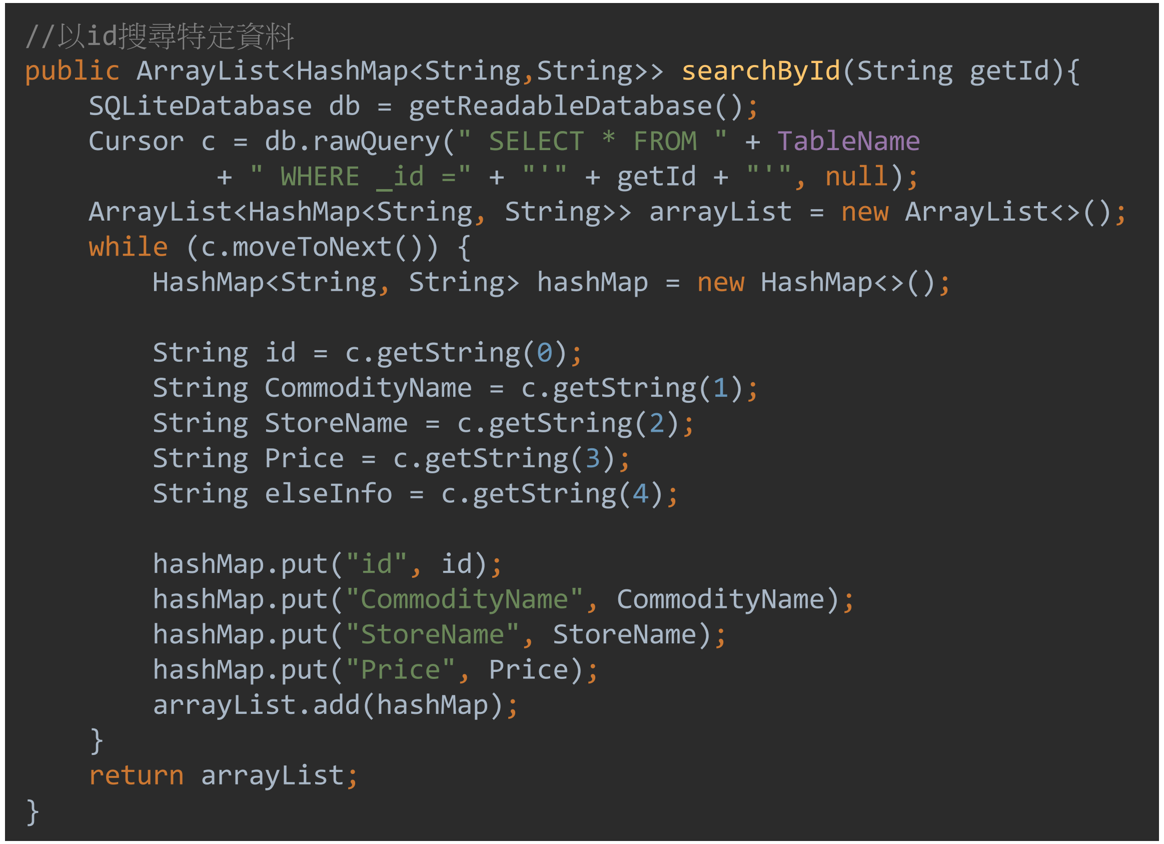
2. SQLiteDataBaseHelper.class



```java
//顯示所有資料
public ArrayList<HashMap<String, String>> showAll() {
    SQLiteDatabase db = getReadableDatabase();
    Cursor c = db.rawQuery(" SELECT * FROM " + TableName + " ORDER BY CreateDate DESC", null);
    ArrayList<HashMap<String, String>> arrayList = new ArrayList<>();
    while (c.moveToNext()) {
        HashMap<String, String> hashMap = new HashMap<>();

        String id = c.getString(0);
        String CommodityName = c.getString(1);
        String StoreName = c.getString(2);
        String Price = c.getString(3);
        String CreateDate = c.getString(4);

        hashMap.put("id", id);
        hashMap.put("CommodityName", CommodityName);
        hashMap.put("StoreName", StoreName);
        hashMap.put("Price", Price);
        hashMap.put("CreateDate", CreateDate);
        arrayList.add(hashMap);
    }
    return arrayList;

}
```

Complete Code

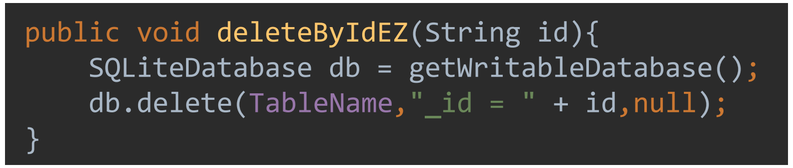
2. SQLiteDataBaseHelper.class



```
//以id搜尋特定資料
public ArrayList<HashMap<String,String>> searchById(String getId){
    SQLiteDatabase db = getReadableDatabase();
    Cursor c = db.rawQuery(" SELECT * FROM " + TableName
            + " WHERE _id =" + "'" + getId + "'", null);
    ArrayList<HashMap<String, String>> arrayList = new ArrayList<>();
    while (c.moveToNext()) {
        HashMap<String, String> hashMap = new HashMap<>();

        String id = c.getString(0);
        String CommodityName = c.getString(1);
        String StoreName = c.getString(2);
        String Price = c.getString(3);
        String elseInfo = c.getString(4);

        hashMap.put("id", id);
        hashMap.put("CommodityName", CommodityName);
        hashMap.put("StoreName", StoreName);
        hashMap.put("Price", Price);
        arrayList.add(hashMap);
    }
    return arrayList;
}
```

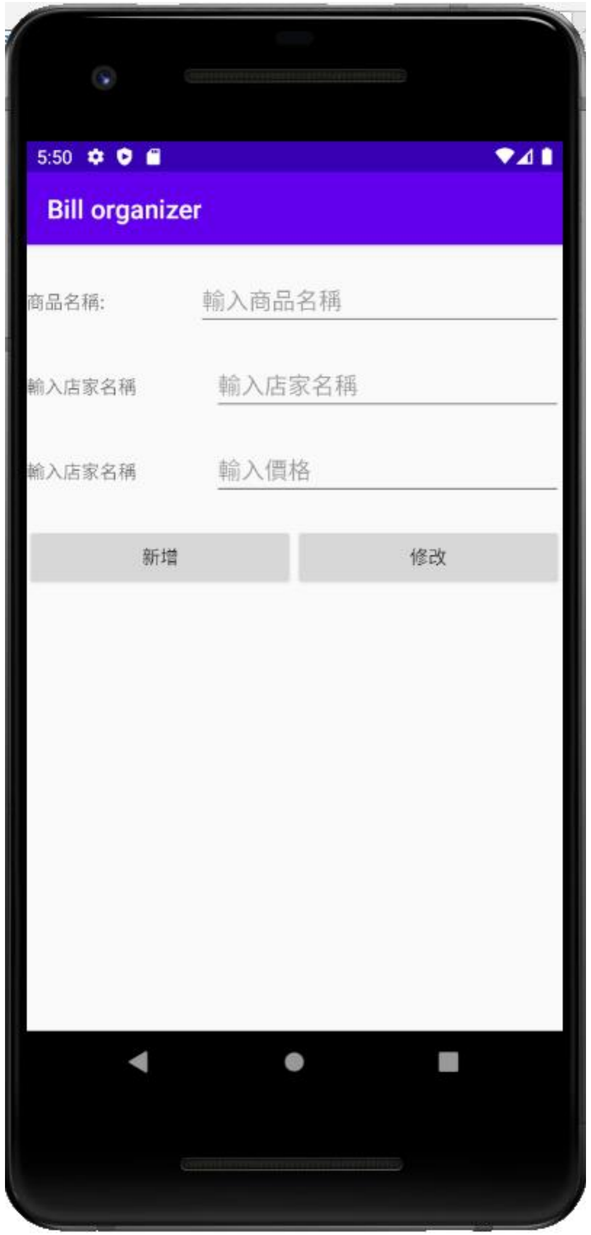
Complete Code

2. SQLiteDataBaseHelper.class



```java
public void deleteByIdEZ(String id){
    SQLiteDatabase db = getWritableDatabase();
    db.delete(TableName,"_id = " + id,null);
}
```

Demo


5:50
Bill organizer
商品名稱:
輸入商品名稱
輸入店家名稱
輸入店家名稱
輸入店家名稱
輸入價格
新增
修改