SQLite

We will improve on the *MyPlanet* app that we built three weeks ago. It obtained mock data from a *Country* class. Here’s a reminder of the three activities:

List of continents

List of countries in a continent

Details about a country

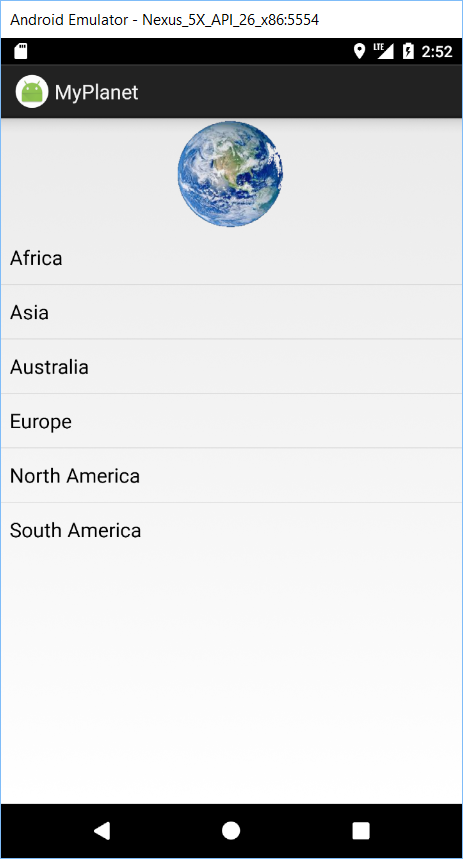
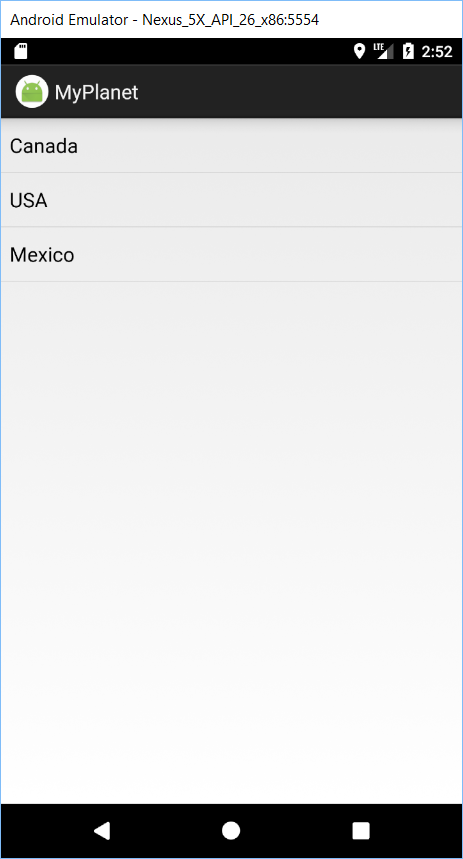
The new database driven app will have the same structure as before except that it will read data from a SQLite database instead of static data from *Country.java.*  The SQLite helper will maintain the *MyPlanet.db* database, and provide access to it for the other activities.

Download and extract the file named MyPlanet\_START.zip file and extract it. Open the *MyPlanet* application in Android Studio as your starting point. Before you run it do the following in Android Studio:

*Build* >> *Clean Project*

*Build* >> *Build APK*

Run the app and see what it does.

Add a class file named *MyPlanetDbHelper.java* to your project. Replace the contents with the code below:

public class MyPlanetDbHelper extends SQLiteOpenHelper{

01. MyPlanetDbHelper.java.txt

private static final String DB\_NAME = "MyPlanet.db";

private static final int DB\_VERSION = 1;

public MyPlanetDbHelper(Context context) {

// The 3'rd parameter (null) is an advanced feature relating to cursors

super(context, DB\_NAME, null, DB\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase sqLiteDatabase) {

}

@Override

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

}

}

You specify the database name and version by passing them to the constructor of the *SQLiteOpenHelper* superclass.

The next step is to tell it what tables to create. Our table will look like this:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| \_id | CONTINENT | COUNTRY | DESCRIPTION | IMAGE\_RESOURCE\_ID |
| 1 | North America | Canada | Canada is a country in … | R.drawable.canada |
| 2 | North America | USA | The United States of America (USA) … | R.drawable.usa |
| 3 | North America | Mexico | Mexico , officially the United Mexican … | R.drawable.mexico |
| 4 | Africa | Djibouti | Djibouti, officially the Republic… | R.drawable.djibouti |
| 5 | Africa | Botswana | Botswana, is a landlocked country … | R.drawable.botswana |

It is recommended that your table has a single column *\_id* to hold the *primary key* that contains integer values. Also, Android code is hardwired to expect a numeric *\_id* column.

Here are the main data types you can use in SQLite, and what they can store:

|  |  |
| --- | --- |
| INTEGER | Any integer type |
| TEXT | Any character type |
| REAL | Any floating-point number |
| NUMERIC | Booleans, dates and date-times |
| BLOB | Binary large object |

We will use the *onCreate()* method to create the *COUNTRY* table and populate it with sample data.

Back in the *Country.java* class file, change the access level of the constructor to *public* from *private* and add the following setter methods:

02. Country setters.txt

public void set\_continent(String \_continent) {

this.\_continent = \_continent;

}

public void set\_description(String \_description) {

this.\_description = \_description;

}

public void set\_name(String \_name) {

this.\_name = \_name;

}

public void set\_imageResourceId(int \_imageResourceId) {

this.\_imageResourceId = \_imageResourceId;

}

Add the following helper method to *MyPlanetDbHelper.java*:

03. getCreateCountryTableSql.txt

private String getCreateCountryTableSql() {

String sql = "";

sql += "CREATE TABLE COUNTRY (";

sql += "\_id INTEGER PRIMARY KEY AUTOINCREMENT, ";

sql += "CONTINENT TEXT, ";

sql += "COUNTRY TEXT, ";

sql += "DESCRIPTION TEXT, ";

sql += "IMAGE\_RESOURCE\_ID INTEGER);";

return sql;

}

The above method returns the SQL create table command, which you will find very familiar.

04. insertCountry.txt

Add the following *insertCountry()* method to the *MyPlanetDbHelper* class.

private void insertCountry(SQLiteDatabase db, Country country) {

ContentValues values = new ContentValues();

values.put("CONTINENT", country.getContinent());

values.put("COUNTRY", country.getName());

values.put("DESCRIPTION", country.getDescription());

values.put("IMAGE\_RESOURCE\_ID", country.getImageResourceId());

db.insert("COUNTRY", null, values);

}

Find the image file pertaining to the African country named Botswana (*botswana.png*) and add it to the *drawable* folder.

Add the following *northAmericaCountries* static array, which will provide us with seed data that can be added to our database:

05. seed data.txt

private static final Country[] northAmericaCountries = {

new Country("North America", "Canada", "Canada is a country in the northern part of North America. Its ten provinces and three territories extend from the Atlantic to the Pacific and northward into the Arctic Ocean, covering 9.98 million square kilometres (3.85 million square miles), making it the world's second-largest country by total area and the fourth-largest country by land area. Canada's southern border with the United States is the world's longest bi-national land border. The majority of the country has a cold or severely cold winter climate, but southerly areas are warm in summer. Canada is sparsely populated, the majority of its land territory being dominated by forest and tundra and the Rocky Mountains. It is highly urbanized with 82 per cent of the 35.15 million people concentrated in large and medium-sized cities, many near the southern border. Its capital is Ottawa, and its largest metropolitan areas are Toronto, Montreal and Vancouver.", R.drawable.canada),

new Country("North America", "USA", "The United States of America (USA), commonly known as the United States (U.S.) or America is a federal republic composed of 50 states, a federal district, five major self-governing territories, and various possessions. Forty-eight states and the federal district are contiguous and located in North America between Canada and Mexico. The state of Alaska is in the northwest corner of North America, bordered by Canada to the east and across the Bering Strait from Russia to the west. The state of Hawaii is an archipelago in the mid-Pacific Ocean. The U.S. territories are scattered about the Pacific Ocean and the Caribbean Sea, stretching across nine official time zones. The extremely diverse geography, climate and wildlife of the United States make it one of the world's 17 megadiverse countries.", R.drawable.usa),

new Country("North America", "Mexico", "Mexico , officially the United Mexican States (Spanish: Estados Unidos Mexicanos, listen (help·info)), is a federal republic in the southern portion of North America. It is bordered to the north by the United States; to the south and west by the Pacific Ocean; to the southeast by Guatemala, Belize, and the Caribbean Sea; and to the east by the Gulf of Mexico. Covering almost two million square kilometers (over 760,000 sq mi),Mexico is the sixth largest country in the Americas by total area and the 13th largest independent nation in the world.\n" +

"With an estimated population of over 120 million, Mexico is the eleventh most populous country and the most populous Spanish-speaking country in the world while being the second most populous country in Latin America. Mexico is a federation comprising 31 states and a special federal entity that is also its capital and most populous city. Other metropolises include Guadalajara, León, Monterrey, Puebla, Toluca, and Tijuana.", R.drawable.mexico),

new Country("Africa", "Djibouti", "Djibouti, officially the Republic of Djibouti, is a country located in the Horn of Africa. It is bordered by Eritrea in the north, Ethiopia in the west and south, and Somalia in the southeast. The remainder of the border is formed by the Red Sea and the Gulf of Aden at the east. Djibouti occupies a total area of just 23,200 km2 (8,958 sq mi).", R.drawable.djibouti),

new Country("Africa", "Botswana", "Botswana, is a landlocked country located in Southern Africa. The citizens refer to themselves as Batswana (singular: Motswana).Formerly the British protectorate of Bechuanaland, Botswana adopted its new name after becoming independent within the Commonwealth on 30 September 1966. Since then, it has maintained a strong tradition of stable representative democracy, with a consistent record of uninterrupted democratic elections and the best perceived corruption ranking in Africa for the last four years.", R.drawable.botswana),

};

The *onUpgrade()* method is used whenever you want to make a change to your database. This could result from a need to alter the data or alter the schema. Let us assume we want to add a new NUMERIC column named POPULATION to the database. This would be the version 2 of our application.

Change the version number of *DB\_VERSION* from 1 to 2

private static final int DB\_VERSION = 2;

Add the following instance variable to *MyPlanetDbHelper.java*:

private Context context;

Add the following code to the *MyPlanetDbHelper* constructor:

this.context = context;

We will need to put logic somewhere so that we can inspect the version number to determine what changes need to be done. Add a new method named *updateMyDatabase()* to *MyPlanetDbHelper* as follows:

private void updateMyDatabase(SQLiteDatabase db, int oldVersion, int newVersion) {

if (oldVersion < 1) {

06. updateMyDatabase.txt

db.execSQL(getCreateCountryTableSql());

for (Country c : northAmericaCountries) {

insertCountry(db, c);

}

}

if (oldVersion<2) {

db.execSQL("ALTER TABLE COUNTRY ADD COLUMN POPULATION NUMERIC;");

}

}

The above code changes the schema of the *COUNTRY* table in the database, in version 2 of the app, by adding an additional *POPULATION* column.

Add this code inside the *onCreate()* method, which calls *updateMyDatabase()* as shown below:

@Override

public void onCreate(SQLiteDatabase sqLiteDatabase) {

updateMyDatabase(sqLiteDatabase, 0, DB\_VERSION);

}

Similarly, add a call to *updateMyDatabase()* in the *onUpgrade()* method. Your *onUpdrade()* method will look like this:

@Override

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

updateMyDatabase(sqLiteDatabase, i, i1);

}

Even though the application runs without any errors at this stage, it does not use SQLite yet. The reason is because we have not yet wired up the code that uses a database rather than the mock data from the *Country* class.

There are a number of steps we need to go through to change *CountryActivity* so that it uses the *MyPlanet* SQLite database.

1. Get a reference to the *MyPlanet* database

SQLiteOpenHelper helper = new MyPlanetDbHelper(this);

SQLiteDatabase db = helper.getWriteableDatabase()  
OR  
SQLiteDatabase db = helper.getReadableDatabase()

1. Create a cursor to read country data from the database
2. Navigate to the country record
3. Display details about a country in *CountryDetailsActivity*

### ActivityMain.java

Our first task is to read the continents in *MainActivity.java* from the database rather than using the *string-array* in *strings.xml*.

Delete the following attribute in activity\_main.xml, which loads the content of the ListView from strings.xml:

~~android:entries="@array/continents"~~

And add these instance variables to activities *MainActivity*, *CountryActivity* and *CountryDetailsActivity*:

private SQLiteDatabase db;

private Cursor cursor;

Similarly, also in activities *MainActivity*, *CountryActivity* and *CountryDetailsActivity*, override the *onDestroy()* method so that these variables are garbage collected when they are no more used:

@Override

protected void onDestroy() {

07. onDestroy.txt

super.onDestroy();

if (cursor != null)

cursor.close();

if (db != null)

db.close();

}

Add the following *getContinents()* method to *MainActivity.java*:

private String[] getContinents() {

SQLiteOpenHelper helper = new MyPlanetDbHelper(this);

String[] continents = null;

try {

db = helper.getReadableDatabase();

Cursor cursor= db.rawQuery("select DISTINCT CONTINENT from COUNTRY", null);

int count = cursor.getCount();

continents = new String[count];

if (cursor.moveToFirst()) {

int ndx=0;

do {

continents[ndx++] = cursor.getString(0);

} while (cursor.moveToNext());

}

} catch (SQLiteException sqlex) {

String msg = "[MainActivity / getContinents] DB unavailable";

msg += "\n\n" + sqlex.toString();

Toast t = Toast.makeText(this, msg, Toast.LENGTH\_LONG);

t.show();

}

return continents;

}

Replace the onCreate() method in MainActivity with the following:

09. MainActivity onCreate.txt

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

ListView list\_continents = (ListView) findViewById(R.id.list\_continents);

String[] continents = getContinents();

ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(

this, android.R.layout.simple\_list\_item\_1, continents

);

list\_continents.setAdapter(arrayAdapter);

list\_continents.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

TextView tv = (TextView) view;

String continent = tv.getText().toString();

Intent intent = new Intent(MainActivity.this, CountryActivity.class);

intent.putExtra("continent", continent);

startActivity(intent);

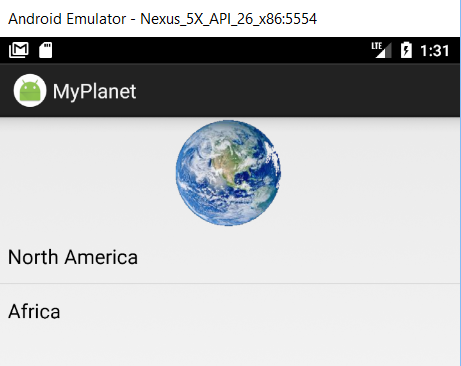
}

});

}

Notice that we are passing on the selected continent to the next activity (CountryActivity.java) using the *intent.putExtra()* method.

Run the application. You will notice continents North America & Africa are being pulled from the database. At the moment, there are only two continents in the database.



### CountryActivity.java

Comment out or delete the following statement in the *onCreate()* method in *CountryActivity*:

~~ArrayAdapter<Country> arrayAdapter = new ArrayAdapter<Country>(~~

~~this, android.R.layout.simple\_list\_item\_1, Country.northAmericaCountries~~

~~);~~

~~ListView listCountries = getListView();~~

~~listCountries.setAdapter(arrayAdapter);~~

10. CountryActivity onCreate.txt

Replace the above code with the following:

// get the continent from the intent

String continent = getIntent().getExtras().get("continent").toString();

SQLiteOpenHelper helper = new MyPlanetDbHelper(this);

try {

db = helper.getReadableDatabase();

cursor = db.query("COUNTRY",

new String[] {"\_id", "COUNTRY"},

"CONTINENT = ?",

new String[] {continent},

null, null, null);

SimpleCursorAdapter adapter = new SimpleCursorAdapter(this,

android.R.layout.simple\_list\_item\_1,

cursor,

new String[] {"COUNTRY"},

new int[] {android.R.id.text1});

ListView listCountries = getListView();

listCountries.setAdapter(adapter);

} catch (SQLiteException sqlex) {

String msg = "[CountryActivity / onCreate] DB unavailable";

msg += "\n\n" + sqlex.toString();

Toast t = Toast.makeText(this, msg, Toast.LENGTH\_LONG);

t.show();

}

Replace contents of the onListItemClick() method in CountryActivity with this code:

TextView tv = (TextView) v;

11. CountryActivity onListItemClick.txt

String country = tv.getText().toString();

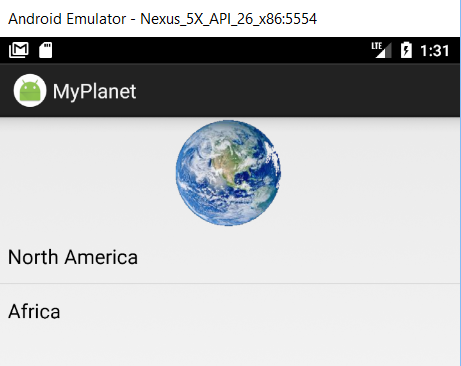
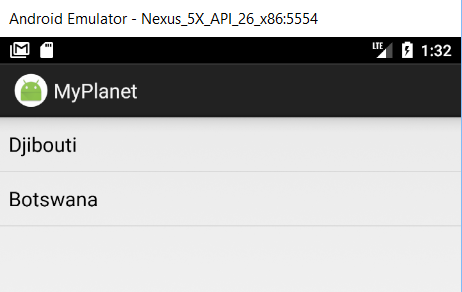
Intent i = new Intent(this, CountryDetailsActivity.class);

i.putExtra("country", country);

startActivity(i);

We are passing the selected country to *CountryDetailsActivity* using the *intent.putExtra()* method.

Run the app. You will be able to click or tap on a continent and see the countries in that continent.

### CountryDetailsActivity

Add the following getCountry() method to CountryDetailsActivity.java. This method searches the database for a given country.

12. getCountry.txt

private Country getCountry(String cntry) {

Country country = null;

SQLiteOpenHelper helper = new MyPlanetDbHelper(this);

try {

SQLiteDatabase db = helper.getReadableDatabase();

Cursor cursor = db.query("COUNTRY",

new String[] {"CONTINENT", "COUNTRY", "DESCRIPTION", "IMAGE\_RESOURCE\_ID"},

"COUNTRY = ?",

new String[] {cntry},

null, null, null);

// move to the first record

if (cursor.moveToFirst()) {

// get the country details from the cursor

country = new Country(

cursor.getString(0),

cursor.getString(1),

cursor.getString(2),

cursor.getInt(3) );

}

} catch (SQLiteException sqlex) {

String msg = "[CountryDetailsActivity/getCountry] DB unavailable";

msg += "\n\n" + sqlex.toString();

Toast t = Toast.makeText(this, msg, Toast.LENGTH\_LONG);

t.show();

}

return country;

}

Comment out or delete the following statement in the *onCreate()* method in *CountryDetailsActivity*:

~~int countryIndex = (Integer) getIntent().getExtras().get("index");~~

~~Country country = Country.northAmericaCountries[countryIndex];~~

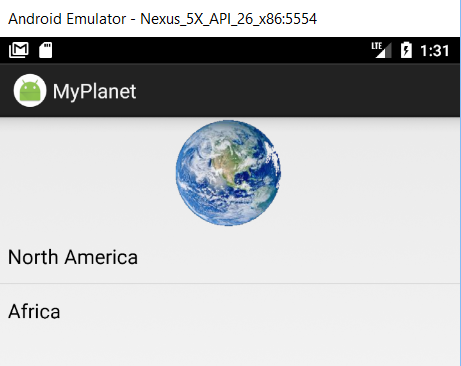
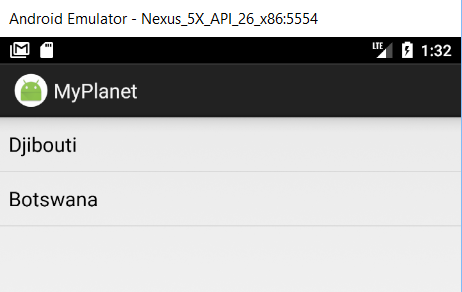
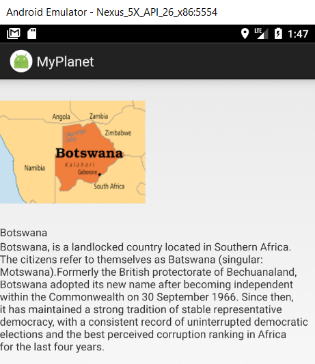
Replace the above code with the following:

13. CountryDetailActivity onCreate.txt

String cntry = getIntent().getExtras().get("country").toString();

Country country = getCountry(cntry);

Run your application:

Make the details page nicer. Replacie the TextView controls in *activity\_country\_detail.xml* with:

14. activity\_country\_details.java.txt

<TextView

android:id="@+id/name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceLarge"

android:layout\_margin="5dp"/>

<TextView

android:id="@+id/description"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="5dp"/>