

Chapitre 6 : SQLite

Mousaab Bada m.bada@univ-batna2.dz
http://staff.univ-batna2.dz/bada_mosaab

24/01/2018 (rev. 1)

Objectifs :

- L'objectif est d'écrire une application Android qui permet de manipuler une base de données SQLite

1. Introduction:

De nombreuses applications manipulent des données et il est souvent nécessaire des les stocker dans des base de données.

Ces bases de données peuvent être exploités par des requêtes SQL.

L'application à développer devra utilisée la technique dite *embedded SQL* : les instructions en langage SQL seront incorporées dans le code source d'un programme écrit dans un autre langage (ici le Java sous Android).

2. SQLite

SQLite est une bibliothèque écrite en C qui propose un moteur de base de données relationnelle accessible par le langage SQL.

Q1 : Est ce que la base de donnée est stockée dans un serveur distant (Modèle client/serveur) ?

R1 : **Non**, L'intégralité de la base de données (déclarations, tables, index et données) est **stockée dans un fichier indépendant** de la plateforme.

Q2 : ou est il stocké le fichier de la BDD ?

R2 : Si l'application crée une base de données, celle-ci est par défaut enregistrée dans le répertoire:

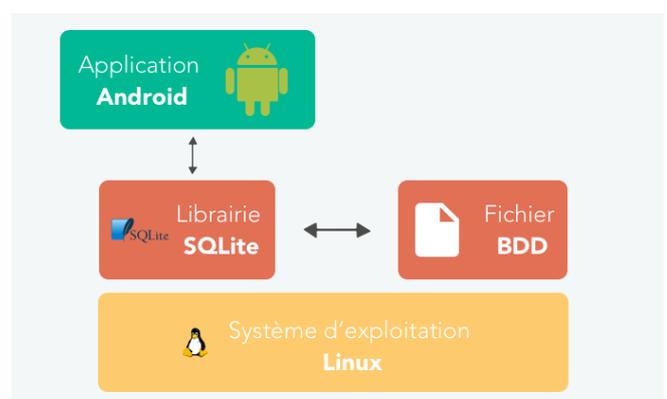
`/data/APP_NAME/databases/ DATABASE_NAME.`

Q3 : pourquoi le mot Lite ?

R3 : Lite = léger

Sa taille est extrêmement **légère** (moins de 2 Mo), il est également très populaire sur :

- les systèmes embarqués
- la plupart des smartphones modernes : l'iPhone ainsi que les systèmes d'exploitation mobiles Symbian et Android l'utilisent comme base de données embarquée.



3. Etape de création d'une base de donnée SQLite

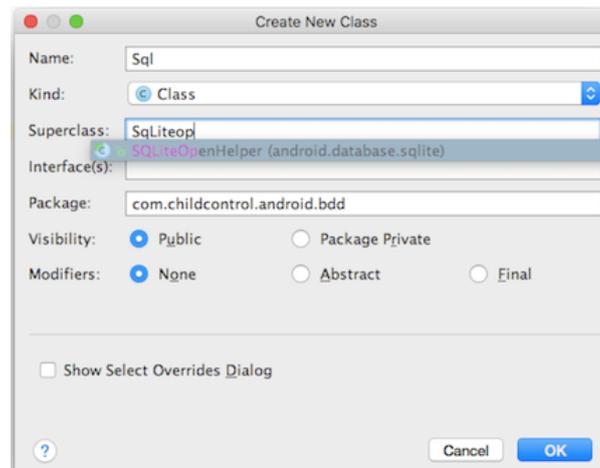
Nous allons apprendre la manipulation d'une base de donnée SQLite à travers un exemple complet qui permet de :

- Créer une base de donnée.
- Créer la table étudiant (id, nom, prénom)
- insérer des étudiants.
- Récupérer des étudiants selon un ou plusieurs critères
- Modifier des champs
- supprimer un etudiant.

3.1. Création de la class « Helper »

1. Pour créer et mettre à jour une base de données dans une application Android, on doit créer une classe qui hérite de la class SQLiteOpenHelper.

Dans cette exemple, nous allons donner le nom « Sql » à cette class.



2. Ensuite, il faut créer le **constructeur** qui prend comme paramètre le contexte de l'activité appelante.

```
public Sql (Context context) {  
}
```

3. Maintenant, il faut ajouter l'instruction qui permet de **créer notre Base De Donnée "LMD3SI.db"**.

Ceci n'est rien qu'un appel au constructeur de la classe mère en précisant le nom de la base de données et sa version actuelle.

```
public Sql (Context context) {
    super(context, "LMD3SI.db", null, 1);
}
```

4. Dans cette classe, on doit redéfinir les méthodes suivantes pour créer et mettre à jour votre base de données :
- onCreate() : pour accéder à une base de données qui n'est pas encore créée.
 - onUpgrade() : si la version de la base de données évolue, cette méthode permettra de mettre à jour le schéma de base de données existant ou de supprimer la base de données existante et la recréer par la méthode onCreate().
 - Les deux méthodes reçoivent en paramètre un objet SQLiteDatabase qui est la représentation Java de la base de données.

```
a. public void onCreate(SQLiteDatabase db)
b. public void onUpgrade(SQLiteDatabase db, int
    oldVersion, int newVersion)
```

5. Une fois la base de données a été créée, il nous reste que **la création de la table** : « **étudiant** » en appelant la méthode **execSQL** qui permet d'exécuter des requêtes SQL qui n'ont pas des donnée de retour (ex : insertion, suppression, modification).

```
public void onCreate(SQLiteDatabase db) {
    db.execSQL( " create table etudiant (ID INTEGER PRIMARY KEY AUTOINCREMENT,
                NOM TEXT,
                PRENOM TEXT) ");
}
```

6. Enfin, dans la class MainActivity on rajoute l'instruction suivant qui fait l'appel au constructeur de la class précédente en passant le contexte de l'activité comme paramètre.

```
Sql s = new Sql(this);
```

3.2. SQLite : insertion d'un étudiant (execSQL) :

1. Dans la classe **Sql**, il faut créer la méthode : **insertion** qui prend en paramètre une requête sous forme d'une chaîne de caractère.

```
public void insertion (String Requete){

    //récupérer la base de donnée en mode écriture
    SQLiteDatabase db = this.getWritableDatabase();
    db.execSQL ( Requete);

}
```

2. Dans la class MainActivity on rajoute l'instruction qui fait l'appel à la méthode insertion.

```
Sql s = new Sql(this);
String Req ="insert into etudiant (NOM,PRENOM) values (' +nom+"',"+prenom+"') " ;
s.insertion(Req);
```

3.3. SQLite : Selectionner tout les étudiants (rawQuery) :

1. Dans la classe **Sql**, il faut créer la méthode : **listAll** qui prend en paramètre une requête sous forme d'une chaîne de caractère.

Contrairement à l'instruction **execSQL**, la requête **rawQuery** c'est une instruction de sélection qui retourne des données.

Le résultat est un objet de type Cursor (table).

```
public Cursor listAll (){
    SQLiteDatabase db = this.getWritableDatabase();

    Cursor c = db.rawQuery("select * from etudiant",null);

    return c;
}
```

2. Dans la class MainActivity on rajoute les instructions qui permettent d'appeler la méthode **listAll** et de parcourir l'objet Cursor.

```
Cursor c= s.listAll();
if (c.getCount()==0){
    Log.i("msg", "la table est vide");

} else{

    while (c.moveToNext()){
        int id = c.getInt(0) ;
        String nom= c.getString(1) ;
        String prenom = c.getString(2) ;

    }
}
```

Code source :

MainActivity

```

import android.database.Cursor;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    DataBaseHelper DataBase;

    EditText nom;
    EditText prenom;
    Button valider;
    Button afficherTout;
    Button Recherche;
    EditText RechNom;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Log.i("msg", " main activity");

        DataBase = new DataBaseHelper(this);
        nom = (EditText) findViewById(R.id.editText4);
        prenom = (EditText) findViewById(R.id.editText5);
        valider = (Button) findViewById(R.id.button2);
        afficherTout = (Button) findViewById(R.id.button);
        Recherche = (Button) findViewById(R.id.button3);
        RechNom = (EditText) findViewById(R.id.editText);

        /***** l'insertion *****/

        valider.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String Req = "insert into etudiant (NOM,PRENOM) values (" + nom.getText().toString() + ", " + prenom.getText().toString() + ")";
                DataBase.insertion(Req);

                Toast.makeText(MainActivity.this, "insertion avec success", Toast.LENGTH_SHORT).show();
                nom.setText("");
                prenom.setText("");

            }

        });

        /***** afficher tout les contacts *****/

        afficherTout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Cursor c= DataBase.listAll();
                if (c.getCount()==0){
                    Log.i("msg", "la table est vide");
                    Afficheur("MSG: ", "Pas de contact");
                }
            }
        });
    }
}

```

```

    }else{
        StringBuffer buffer = new StringBuffer();
        while (c.moveToNext()){
            buffer.append("ID: "+c.getString(0)+ " Nom= "+c.getString(1)+" Prenom= "+c.getString(2) +"\n\n");
        }
        // afficheur c'est methode que vous le trouverez a la fin: elle permet d'afficher le msg dans une boite de dialogue
        Afficheur ("liste des contacts",buffer.toString());
    }
}
});

```

/****** la recherche par nom *****/

```

Recherche.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        Cursor c= DataBase.selectionParNom(RechNom.getText().toString());

        if (c.getCount()==0){
            Log.i("msg", "la table est vide");
            Afficheur("MSG: ", "Pas de contact");
        }else{

            StringBuffer buffer = new StringBuffer();
            while (c.moveToNext()){
                buffer.append("ID: "+c.getString(0)+ " Nom= "+c.getString(1)+" Prenom= "+c.getString(2) +"\n\n");
            }
            // afficheur c'est methode que vous le trouverez a la fin: elle permet d'afficher le msg dans une boite de dialogue
            Afficheur ("liste des contacts",buffer.toString());
        }
    }
});
}

```

/******
 /****** fin on create *****/
 /******

```

public void Afficheur (String titre, String msg ){
    AlertDialog.Builder builder= new AlertDialog.Builder(this);
    builder.setCancelable(true);
    builder.setTitle(titre);
    builder.setMessage(msg);
    builder.show();
}

}

```

La classe : DataBaseName

```

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 android.util.StringBuilderPrinter;

/**
 * Created by bada on 14/04/2018.
 */

public class DataBaseHelper extends SQLiteOpenHelper {

    static String DataBaseName ="LMD3.db";

    public DataBaseHelper(Context context) {
        super(context, DataBaseName , null, 1);

        Log.i("msg", "constructeur de Data base helper ");
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        Log.i("msg", "appel OnCreat () "+db.toString());
        db.execSQL("create table etudiant (ID INTEGER PRIMARY KEY AUTOINCREMENT, NOM TEXT, PRENOM TEXT)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        Log.i("msg", "appel OnUpgrade () "+db.toString());

        db.execSQL("DROP TABLE IF EXISTS etudiant");
        onCreate(db);
    }

    // ici la requête a été préparé au niveau de l' activity
    public void insertion (String Req){

        SQLiteDatabase db = this.getWritableDatabase();
        db.execSQL(Req);
    }

    public Cursor listAll (){
        SQLiteDatabase db = this.getWritableDatabase();

        Cursor c = db.rawQuery("select * from etudiant",null);

        return c;
    }

    // ici on a preparer la requete dans la classe base de donnee.
    public Cursor selectionParNom (String nom){
        SQLiteDatabase db = this.getWritableDatabase();
        Log.i("msg ", "select * from etudiant where nom like '"+nom+"%'");
        Cursor c = db.rawQuery("select * from etudiant where nom like '"+nom+"%' ",null);
        return c;
    }
}

```

l'interface :



```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context="com.childcontrol.android.sql.MainActivity">
```

Il faut utiliser le nom de votre context ici

```
<EditText
  android:id="@+id/editText4"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_x="123dp"
  android:layout_y="13dp"
  android:ems="10"
  android:inputType="textPersonName"
  android:text="" />
```

```
<EditText
  android:id="@+id/editText5"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_x="120dp"
  android:layout_y="65dp"
  android:ems="10"
  android:inputType="textPersonName"
  android:text="" />
```

```
<TextView
  android:id="@+id/textView4"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="13dp"
    android:layout_y="26dp"
    android:text="nom" />
```

```
<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="18dp"
    android:layout_y="79dp"
    android:text="prenom" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="34dp"
    android:layout_y="206dp"
    android:text="Valider" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="252dp"
    android:layout_y="213dp"
    android:text="Afficher" />
```

```
<Button
    android:id="@+id/find"
    android:layout_width="wrap_content"
    android:layout_height="69dp"
    android:layout_x="260dp"
    android:layout_y="332dp"
    android:text="Button" />
```

```
<EditText
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="26dp"
    android:layout_y="341dp"
    android:ems="10"
    android:inputType="textPersonName" />
```

```
</AbsoluteLayout>
```