

Des bases de données SQLite sous Android

Jérôme Tanghe

Mercredi 16 novembre



Les bases de données SQLite sous Android

- Les bases de données sont stockées dans le répertoire `/data/data/com.votre.application/databases`
 - Uniquement accessible par votre application¹
 - Pas besoin d'autorisation supplémentaire
- La base de données a sa propre version
- Cycle de vie :
 - 1 onCreate
 - 2 onUpgrade
 - 3 onDowngrade (facultatif)

1. Sauf si votre téléphone est *rooté* !

Créer une base de données

```
1 public class MyDatabaseHelper extends SQLiteOpenHelper
2 {
3     public static final int DATABASE_VERSION = 1;
4     public static final String DATABASE_NAME = "MyDatabase";
5
6     public MyDatabase(Context context)
7     {
8         super(context, DATABASE_NAME, null,
9             DATABASE_VERSION);
10    }
11
12    @Override
13    public void onCreate(SQLiteDatabase db)
14    {
15        db.execSQL("Requêtes CREATE TABLE");
16    }
17 }
```

Mettre à jour la structure de la base de données

```
1  @Override
2  public void onUpgrade(SQLiteDatabase db, int oldVersion,
3  int newVersion)
4  {
5      db.execSQL("Requêtes de mise à jour de la base de données");
6  }
```

Insérer des données

```
1  MyDatabaseHelper dbHelper = new MyDatabaseHelper(getContext());
2  SQLiteDatabase db = dbHelper.getWritableDatabase();
3
4  ContentValues values = new ContentValues();
5  values.put("champ1", "valeur1");
6  values.put("champ2", "valeur2");
7  // etc...
8
9  long id = db.insert("nomTable", null, values);
```

Supprimer des données

```
1  MyDatabaseHelper dbHelper = new MyDatabaseHelper(getContext());
2  SQLiteDatabase db = dbHelper.getWritableDatabase();
3
4  String where = "champ1 = ? AND ...";
5  String[] whereArgs = { "valeur1", ... };
6
7  int nb = db.delete("nomTable", where, whereArgs);
```

Modifier des données

```
1  MyDatabaseHelper dbHelper = new MyDatabaseHelper(getContext());
2  SQLiteDatabase db = dbHelper.getReadableDatabase();
3
4  ContentValues values = new ContentValues();
5  values.put("champ1", "valeur1");
6  values.put("champ2", "valeur2");
7  // etc...
8
9  String where = "champ1 = ? AND ...";
10 String[] whereArgs = { "valeur1", ... };
11
12 int nb = db.update(
13     "nomTable",
14     values,
15     where,
16     whereArgs
17 );
```

Attention

Contrairement à la création et la suppression de données, ici, on utilise la base de données **en mode lecture seule** !

Lire des données

```
1  MyDatabaseHelper dbHelper = new MyDatabaseHelper(getContext());
2  SQLiteDatabase db = dbHelper.getReadableDatabase();
3
4  String[] champsDesires = {
5      "champ1",
6      "champ2",
7      ...
8  };
9
10 String where = "champ1 = ? AND ...";
11 String[] whereArgs = { "valeur1", ... };
12
13 Cursor cursor = db.query(
14     "nomTable",
15     champsDesires,
16     where,
17     whereArgs,
18     groupBy,
19     groupFilter,
20     having,
21     orderBy
22 );
```


Les curseurs

```
1  cursor.moveToFirst();
2  while(!cursor.isAfterLast())
3  {
4      String champ1 = cursor.getString(0);
5      int champ2 = cursor.getInt(1);
6      // etc.
7
8      cursor.moveToNext();
9  }
```

Quelques conseils

- N'oubliez pas de créer un champ `_ID` avec une contrainte de clé primaire !
- Le SDK Android propose une interface `BaseColumns`, qui peut être utilisée pour stocker les noms des champs d'une table :

```
1  static class Entries implements BaseColumns {  
2      public static final String CHAMP_1 = "champ1";  
3      ...  
4  }
```

`BaseColumns` implémente déjà le champ obligatoire `_ID` ! 😊

- Créez des méthodes dans votre classe `MyDatabaseHelper` pour faire vos requêtes pour mieux séparer le code de votre application du code de gestion de la base de données

C'est bien compliqué, tout ça ☹

**DON'T
PANIC**

ORMLite

Une bibliothèque Java permettant de générer une base de données à partir de classes.

- Utilise les annotations Java
- Support de nombreux moteurs de bases de données, dont SQLite
- Support d'Android

ormlite.com

GreenDAO

Une bibliothèque similaire à ORMLite, mais dédiée à Android.

- Performant ("probably the fastest ORM for Android")
- Faible utilisation de la mémoire
- Chiffrement des bases de données

greenrobot.org/greendao



Wikipédia

SQLite

<https://fr.wikipedia.org/wiki/SQLite>



Documentation d'Android

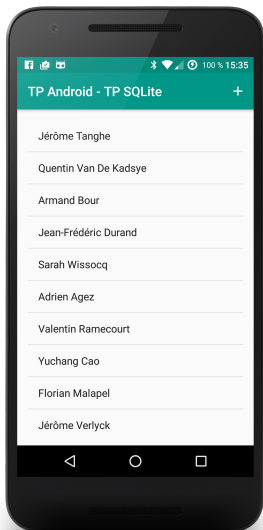
Saving Data in SQL Databases

<https://developer.android.com/training/basics/data-storage/databases.html>

Téléchargez les sources de base
et importez-les sur Android Studio :

<https://www.github.com/Deuchnord/android-sqlite>

La seule classe à laquelle vous devrez toucher est StudentDb.



Exercice

Complétez la classe `StudentDb` de façon à permettre de :

- lister les étudiants (`getStudents()`)
- ajouter des étudiants (`putStudent()`)

Exercice bonus (pour ceux qui vont vite :P)

Ajoutez les deux fonctionnalités suivantes à l'application :

- supprimer des étudiants
- modifier le nom d'un étudiant

Il vous faudra ajouter ces fonctionnalités dans l'application, celles-ci n'ayant pas été implémentées.