Programmation Android TP4 –SQLite

Android dispose d'un moteur de base de données SQLite, assez proche de SQL (avec quelques simplifications et fonctionnalités en moins).

L'objectif de ce TP est de découvrir l'utilisation de SQLite par le biais d'une petite application de gestion des produits.

Etape 1 : Création de la classe chargée de créer la base de données

Créer une classe « **DBConnect** » qui hérite de **SQLiteOpenHelper**: Cette classe est responsable de la création de la base de données.

```
ublic class DBConnect extends SQLiteOpenHelper {
        //Constructeur
        public DBConnect(@Nullable Context context) {
            super(context, "GestionProduit", null, 1);
        }
        @Override
        public void onCreate(SQLiteDatabase db) {
            //création de la table
            db.execSQL("create table produit( id integer primary
key,description varchar(25),prix varchar(25))");
        }
        Override
        public void onUpgrade(SQLiteDatabase db, int i, int i1) {
            db.execSQL("drop table produit");
            onCreate(db);
        }
    }
```

Etape 2 : Définition de la table

Créer une classe « produit » pour la gestion de la table produit (insert, delete, update, select...).

Dans cette classe, on doit faire appel à la première classe **DBConnect** qui permet de créer une BD. Le code va ressembler à ça :

```
public class produit {
DBConnect cn;
//constructeur
public produit(Context context) {
    cn=new DBConnect(context);
}
//insertion
public void insertdt(int code, String descp, String prix){
    SQLiteDatabase db=cn.getWritableDatabase();
    ContentValues values=new ContentValues();
    values.put("_id",code);
    values.put("description",descp);
    values.put("prix",prix);
    db.insert("produit",null,values);
```

```
public void delete(Integer code) {
         SQLiteDatabase db=cn.getWritableDatabase();
         String[] s=new String[]{String.valueOf(code)};
         db.delete("produit"," id=?",s);
      }
      public void update(Integer code,String descp,String prix) {
         SQLiteDatabase db=cn.getWritableDatabase();
         ContentValues contentValue=new ContentValues();
         //stocker la valeur dans contentValue
         contentValue.put("description", descp);
         contentValue.put("prix", prix);
         String[] rsl=new String[]{String.valueOf(code)};
db.update("produit", contentValue, "_id=?", rsl);
      public Cursor data() {
         SQLiteDatabase db=cn.getReadableDatabase();
         return db.rawQuery("select * from produit", null);
      }
```

Etape 3 : Activité principale

Créer le fichier xml (activity_main.xml) dans lequel vous déclarez la liste. L'interface ressemble à celle-ci:

Produits		
Code: Description: Prix		_
SUPPRIMER	ENREGISTRER	MODIFIER
SUPPRIMER 110 PC ACER	ENREGISTRER	MODIFIER
SUPPRIMER 110 PC ACER 123 PC LENEOVO	ENREGISTRER	MODIFIER

Etape 4 :

Dans le MainActivity.java , vous créez le code java suivant :

```
public class MainActivity extends AppCompatActivity {
   EditText d1,d2,d3;
   ListView lsv;
   Button b1,b2,b4;
   produit p=new produit(this );
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout. activity_main);
      lsv = findViewById(R.id.lsv);
      // EditText description
```

```
d1 = findViewById(R.id.edi);
       // EditText prix
       d2 = findViewById(R.id.edi2);
       //EditText code
       d3 = findViewById(R.id.editTextTextPersonName);
       lsv = findViewById(R.id.lsv);
       produit p=new produit(this );
       //boutons
       b1 = findViewById(R.id.bt1);
       b2 = findViewById(R.id.btt2);
       b4 = findViewById(R.id.bt3);
       //affichage des données de la base de données dans listeview
       actualiser();
       //Associer l'écouteur au bouton Ajouter
       b1.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View view) {
p.insertdt(Integer.parseInt(d3.getText().toString()),d1.getText().toString(
),d2.getText().toString());
               actualiser();
               d1.setText(null);
               d2.setText(null);
               d3.setText(null);
           }
       });
        //bouton update
       b2.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View view) {
p.update(Integer.parseInt(d3.getText().toString()),d1.getText().toString(),
d2.getText().toString());
               actualiser();
               d1.setText(null);
               d2.setText(null);
               d3.setText(null);
           }
       });
        //bouton Suppression
       b4.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View view) {
               p.delete(Integer.parseInt(d3.getText().toString()));
               actualiser();
               d3.setText(null);
           }
       });
    }
   public void actualiser() {
       SimpleCursorAdapter adap=new
SimpleCursorAdapter(getApplicationContext(),
               android.R.layout.simple list item 2,
               p.data(),new String[]{" id", "description"},
               new int[]{android.R.id.text1, android.R.id.text2});
       lsv.setAdapter(adap);
    };
}
```