MAMAN Alexandre
BTS SIO2

Compte rendu TP6 : SQLite pour Android

Sommaire :

1) Contexte

On veut, dans ce TP, construire une application Android qui permet de gérer des contacts. Ces contacts seront mis dans une base de données Android gérée par SQLite.

2) Construction de la base de données

1°) Définir une classe Contact contenant les informations :

```java
package com.example.td_6;

public class Contact {
    private int _id;
    private String nom;
    private String numTelephone;


    Contact(int _id, String nom, String numTelephone)
    {
        this._id=_id;
        this.nom = nom;
        this.numTelephone=numTelephone;
    }
```

Compléter cette classe avec des accesseurs et des constructeurs appropriés

```java
    public int getId()
    {
        return _id;
    }

    public void setId(int _id)
    {
        this._id = _id;
    }

    public String getNom()
    {
        return nom;
    }

    public void setNom(String nom)
    {
        this.nom = nom;
    }

    public String getNumTelephone()
    {
        return numTelephone;
    }

    public void setNumTelephone(String numTelephone)
    {
        this.numTelephone = numTelephone;
    }
}
```

2°) Construire un "Database Helper" permettant de gérer une base de données

```java
package com.example.td_6;

import ...

public class LeDatabaseHandler extends SQLiteOpenHelper {

    private static SQLiteDatabase db;
    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME = "contactsManager";
    private static final String TABLE_CONTACTS = "contacts";
    private static final int KEY_ID = 0;
    private static final String KEY_NAME = "name";
    private static final String KEY_PH_NO = "phone_number";


    public LeDatabaseHandler(Context context){
        super(context, DATABASE_NAME, factory: null,DATABASE_VERSION);
        db = this.getWritableDatabase();
    }


    private static final String TABLE_CONTATCS = "contacts";
    private static final String COLONNE_ID= "_id";
    private static final String COLONNE_NOM = "nom";
    private static final String COLONNE_TEL = "numTelephone";


    private static final String REQUETE_CREATION_TABLE = "create table "
            + TABLE_CONTATCS + " (" + COLONNE_ID
            + " integer primary key autoincrement, " + COLONNE_NOM
            + " varchar(30) not null, " + COLONNE_TEL + " varchar(30) not null);";
```

Compléter cette classe de sorte à :

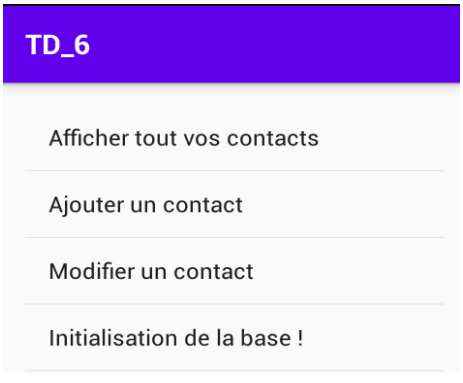
a) Créer la base de données
b) Pouvoir insérer des Contacts dans cette base
c) Récupérer tous les contacts de la base à l'aide de la méthode :

```java
public long insertContact(Contact contact) {
    ContentValues valeurs = new ContentValues();
    valeurs.put(COLONNE_NOM, contact.getNom());
    valeurs.put(COLONNE_TEL, contact.getNumTelephone());
    return db.insert(TABLE_CONTATCS,  nullColumnHack: null, valeurs);
}


public List<Contact> getAllContact()
{
    List<Contact> contacts = new ArrayList<Contact>();
    String selectQuery = "SELECT  * FROM " + TABLE_CONTACTS;

    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(selectQuery,  selectionArgs: null);
    if (cursor.moveToFirst()) {
        do {
            Contact contact = new Contact( _id: 0, nom: "", numTelephone: "");
            contact.setId(Integer.parseInt(cursor.getString( i: 0)));
            //contact.setId(cursor.getString(0));
            contact.setNom(cursor.getString( i: 1));
            contact.setNumTelephone(cursor.getString( i: 2));

            contacts.add(contact);
        } while (cursor.moveToNext());
    }

    return contacts;
}
```

```java
@Override
public void onCreate(SQLiteDatabase db) { db.execSQL(REQUETE_CREATION_TABLE); }

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    db.execSQL("create table " + TABLE_CONTATCS + ";");
    onCreate(db);
}

public void clearDatabase(List<Contact> TABLE_NAME) {
    String clearDBQuery = "DELETE FROM "+TABLE_CONTATCS;
    db.execSQL(clearDBQuery);
}

public boolean updateContact(String s, String s1, int s2) {
    SQLiteDatabase db = this.getReadableDatabase();
    db.execSQL("UPDATE "+TABLE_CONTACTS+" SET nom = "+"'"+s+"' "+ "," +
            " numTelephone="+"'"+s1+"' "+ "WHERE _id = "+"'"+s2+"'");
    return true;
}

}
```

3°) Définir une activity qui fait afficher :


```xml
<TextView
    android:id="@+id/textViewNom"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="15dp"
    android:layout_marginBottom="1dp"
    app:layout_constraintBottom_toTopOf="@+id/_dynamic"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:ignore="MissingConstraints,UnknownId" />

<TextView
    android:id="@+id/textViewTel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="15dp"
    android:layout_marginBottom="1dp"
    app:layout_constraintBottom_toTopOf="@+id/_dynamic"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:ignore="MissingConstraints,UnknownId" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="15dp"
    android:layout_marginBottom="1dp"
    android:text="Liste des contacts"
    app:layout_constraintBottom_toTopOf="@+id/_dynamic"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintTop_toBottomOf="@id/textViewNom"
    tools:ignore="MissingConstraints,UnknownId" />
```



4°) Ajouter le code de sorte que lorsque l'utilisateur clique sur l'item "Initialisation de la base !", la base (table) est recréée avec 4 Contacts :

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

    final LeDatabaseHandler db = new LeDatabaseHandler( context: this);

    mListView = (ListView) findViewById(R.id.listView);
    oListView = (ListView) findViewById(R.id.listViewIni);

    final ArrayAdapter<String> adapter = new ArrayAdapter<String>( context: MainActivity.this,
            android.R.layout.simple_list_item_1, choix);
    mListView.setAdapter(adapter);

    mListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        public void onItemClick(AdapterView<?> parent, View view,
                                int position, long id) {
            if (position == 0) {
                setContentView(R.layout.activity_liste);
                List<Contact> contacts = db.getAllContact();
                TableLayout tableLayout = (TableLayout)findViewById(R.id.tableLayoutListe);
                for (Contact cn : contacts){
                    TableRow row = new TableRow( context: MainActivity.this);
                    TextView tNom = new TextView( context: MainActivity.this);
                    TextView tNum = new TextView( context: MainActivity.this);
                    tNom.setText(cn.getNom());
                    tNum.setText(cn.getNumTelephone());
                    row.addView(tNom);
                    row.addView(tNum);
                    tableLayout.addView(row,new TableLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
                            ViewGroup.LayoutParams.WRAP_CONTENT));
                }
```

5°) Ecrire une activity qui affiche :

```
<EditText
    android:id="@+id/inputModNom"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textShortMessage"
    tools:ignore="MissingConstraints"
    tools:text="Nom">
    <requestFocus />
</EditText>

<EditText
    android:id="@+id/inputModTel"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textShortMessage"
    app:layout_constraintTop_toBottomOf="@id/inputModNom"
    tools:text="Telephone">
    <requestFocus />
</EditText>

<EditText
    android:id="@+id/ViewId"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textShortMessage"
    app:layout_constraintTop_toBottomOf="@id/inputModTel"
    tools:text="Id">
    <requestFocus />
</EditText>

<Button
    android:id="@+id/buttonMod"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Mettre à jour"
    app:layout_constraintTop_toBottomOf="@id/ViewId"
    tools:ignore="MissingConstraints" />
```

3) Conclusion

Ce TD m'a permis d'utiliser les bases de données SQLite et les listes en JAVA mais m'a posé de nombreuses difficultés.