

```
package dz.univannaba;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Hashtable;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.DatabaseUtils;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteDatabase;

// TP3 - Développement mobile
// sabri.ghazi@univ-annaba.dz
//Cette classe vous permet d'effectuer tous les traitement base de données.
public class DBHelper extends SQLiteOpenHelper {

    // le nom de la table de base de données.
    public static final String BOOKS_TABLE_NAME = "Books";

    private HashMap hp;

    public DBHelper(Context context)
    {
        super(context, "bibliotheque" , null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // TODO Auto-generated method stub
        db.execSQL(
            "create table Books " +
            "(id integer primary key, titre text, auteur text, motCles text)"
        );
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // TODO Auto-generated method stub
        db.execSQL("DROP TABLE IF EXISTS Books");
        onCreate(db);
    }

    //Insérer un nouveau rendez-vous
    public boolean insertBooks (String ptitre, String pauteur, String pmotCles)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put("titre", ptitre);
        contentValues.put("auteur", pauteur);
        contentValues.put("motCles", pmotCles);
        db.insert("Books", null, contentValues);
        return true;
    }
}
```

```
public Books RechercherBooksByTitre(String ptitre){

    SQLiteDatabase db = this.getReadableDatabase();

    Cursor res = db.rawQuery( "select * from Books where titre='"+ptitre+"'", null );

    res.moveToFirst();
    Books b;
    // on parcourt le résultat et on crée pour chaque ligne un objet Rdv
    if(res.isAfterLast() == false){
        r= new Books() // on crée un nouveau objet Books
        b.setId(res.getInt(0)); // on mis son ID
        b.setTitre(res.getString(1)); // on mis son Titre
        b.setAuteur(res.getString(2)); // on mis son Auteur
        b.setMotCles(res.getString(3)); // on mis ça MotCles

        res.moveToNext();
    }

    return b;
}

//nombre de lignes se trouvant dans la table.
public int numberOfRows(){
    SQLiteDatabase db = this.getReadableDatabase();
    int numRows = (int) DatabaseUtils.queryNumEntries(db, BOOKS_TABLE_NAME);
    return numRows;
}

//mettre à jour un Books.
public boolean updateBooks (Integer id, String titre, String auteur, String motCles)
{
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put("titre", nom);
    contentValues.put("auteur", tel);
    contentValues.put("motCles", dt);

    db.update("Books", contentValues, "id = ? ", new String[] { Integer.toString(id) } );
    return true;
}

// supprimer un Books
public Integer deleteContact (Integer id)
{
    SQLiteDatabase db = this.getWritableDatabase();
    return db.delete("Books",
    "id = ? ",
    new String[] { Integer.toString(id) });
}

// Lister tous les Books
public ArrayList<Books> ListerTousRDV()
{
    //on crée un liste vide
    ArrayList<Books> array_list = new ArrayList<Books>();
}
```

```
SQLiteDatabase db = this.getReadableDatabase();

// on lance la requête
Cursor res = db.rawQuery( "select * from Books", null );
res.moveToFirst();
Books b;
// on parcourt le résultat et on crée pour chaque ligne un objet Books
while(res.isAfterLast() == false){
    b= new Books() // on crée un nouveau objet Books
    b.setId(res.getInt(0)); // on mis son ID
    b.setTitre(res.getString(1)); // on mis son Titre
    b.setAuteur(res.getString(2)); // on mis son Auteur
    b.setMotCles(res.getString(3)); // on mis ça MoteCles
    array_list.add(b);
    res.moveToNext();
}
// on renvoi le résultat.
return array_list;
}
}
```