



## **SAMPLE PROJECT**

### **BÀI 3: THỰC HIỆN DỰ ÁN 1**

- ⊙ Kết thúc bài học này bạn có khả năng
  - ⊙ Thiết kế CSDL
  - ⊙ Nắm vững hơn kiến thức lập trình với SQLite
  - ⊙ Lập trình CSDL với mô hình DAO
    - **Cách tạo / xóa một cơ sở dữ liệu SQLite trong Android**
    - **Cách tạo / xóa bảng trong SQLite**
    - **Cách thêm/ sửa/ xóa dữ liệu trong bảng**
    - **Cách truy vấn dữ liệu trong bảng**

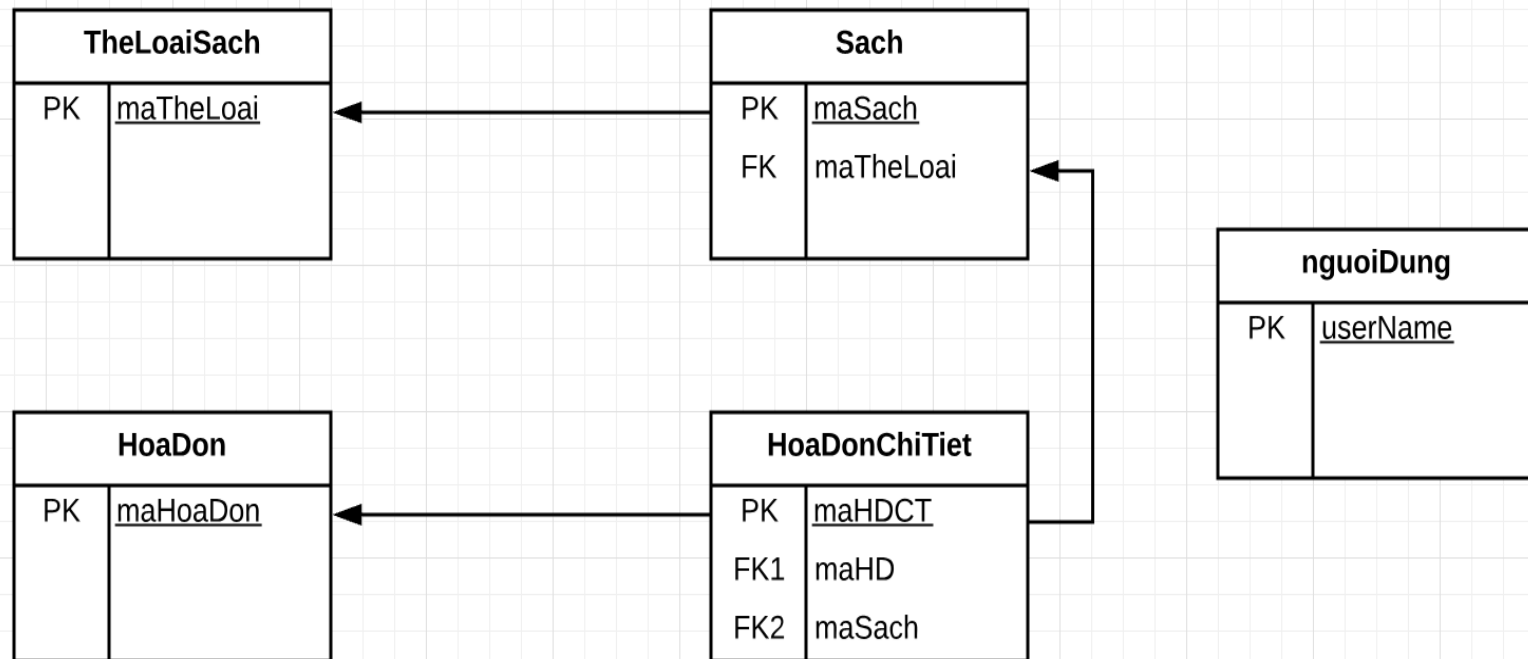


- ⊙ Kết thúc phần này sinh viên có khả năng
  - ⊙ Xây dựng được sơ đồ quan hệ thực thể ERD
  - ⊙ Thiết kế chi tiết cho mỗi thực thể với các ràng buộc hợp lý
  - ⊙ Cài đặt được CSDL trên SQLite
  - ⊙ Viết mã tạo các bảng
  - ⊙ Thiết lập mối quan hệ giữa các thực thể

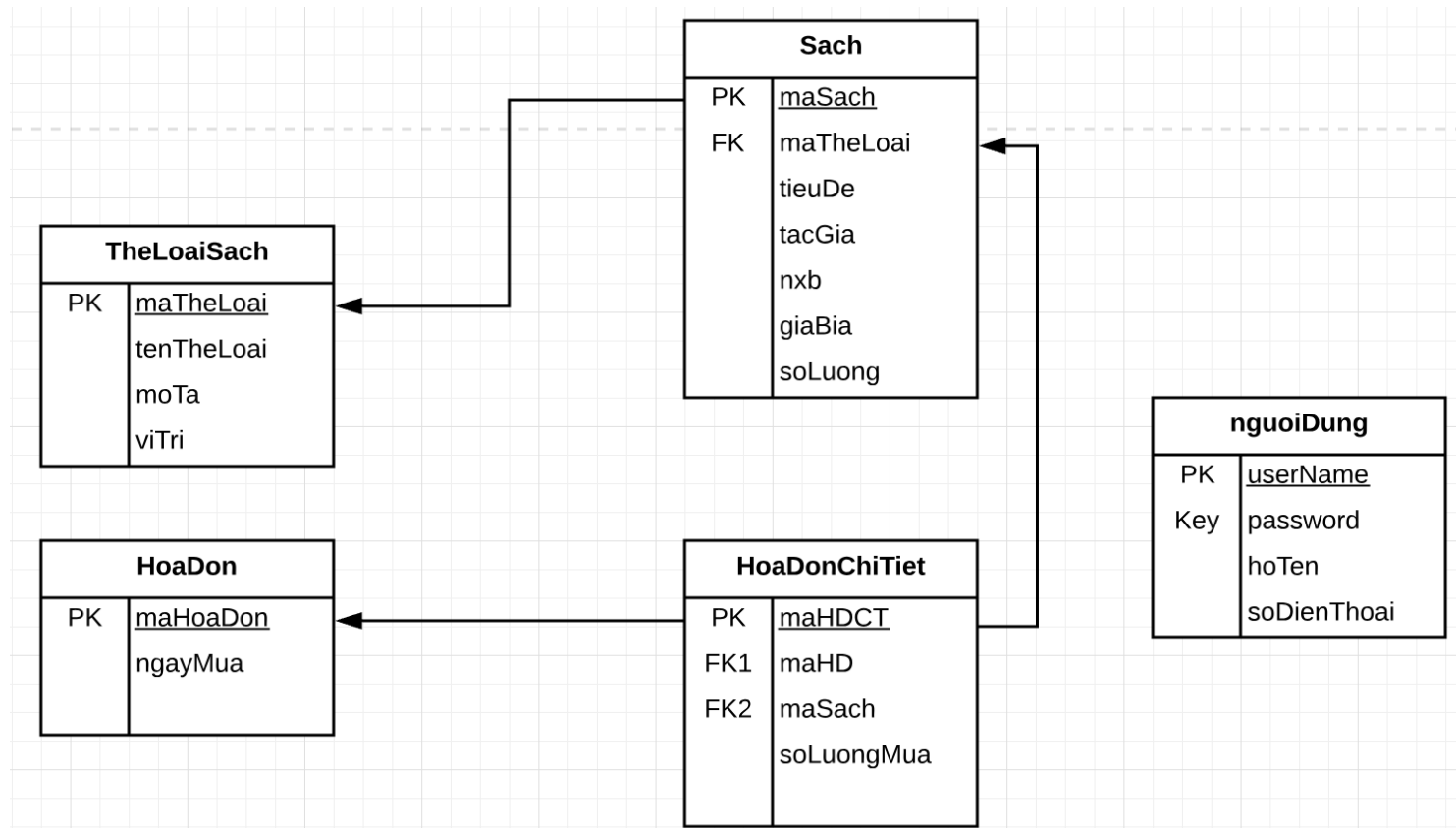


- ❑ Thiết kế ứng dụng là dựa vào các chức năng đã phân tích ở giai đoạn trước để đưa ra mô hình dữ liệu và các giao diện phù hợp.
- ❑ Phần này bạn phải thực hiện 2 công việc
  - ❖ Thiết kế giao diện (đã học bài 2)
  - ❖ Thiết kế dữ liệu

- ❑ Phác thảo sơ đồ ERD level 1, chưa có thông tin chi tiết của các thực thể mà chỉ thể hiện mối quan hệ giữa các thực thể



- ❑ Sơ đồ ERD Level 2, chi tiết hơn để từ đó tạo cơ sở dữ liệu trong SQLite.



Tên cột	Kiểu	Ràng buộc	Ghi chú
MaTheLoai	CHAR(5)	PK, NOT NULL	Mã thẻ thoại
Tên thẻ loại	NVARCHAR(50)	NOT NULL	Tên thẻ loại
Mô tả	NVARCHAR(255)		Mô tả chi tiết
Vị trí	INT		Vị trí của loại sách đó trong cửa hàng, các kệ được đánh số kệ 1, kệ 2...

Tên cột	Kiểu	Ràng buộc	Ghi chú
MaSach	NCHAR(5)	PK, NOT NULL	Mã Sách
MaTheLoai	NCHAR(50)	FK, NOT NULL	Mã thể loại mà sách thuộc về
TacGia	NVARCHAR(50)		Tác giả sách
NXB	NVARCHAR(50)		Nhà xuất bản sách
giaBia	FLOAT	NOT NULL	Giá bán sách
soLuong	INT	NOT NULL	Số lượng tồn kho



Tên cột	Kiểu	Ràng buộc	Ghi chú
<b>MaHoaDon</b>	NCHAR(7)	PK, NOT NULL	Mã hoá đơn
<b>NgayMua</b>	DATE	NOT NULL	Ngày mua hàng

Tên cột	Kiểu	Ràng buộc	Ghi chú
MaHDCT	INT	PK, Tự tăng	Mã hoá đơn chi tiết
MaHoaDon	NCHAR(7)	FK, NOT NULL	Mã hoá đơn
MaSach	NCHAR(5)	FK, NOT NULL	Mã sách được mua
SoLuongMua	INT	NOT NULL	Số lượng mua

Tên cột	Kiểu	Ràng buộc	Ghi chú
<b>userName</b>	NVARCHAR(50)	PK	Tên đăng nhập
<b>Password</b>	NVARCHAR(50)	NOT NULL	Mật khẩu đăng nhập
<b>Phone</b>	NCHAR(10)	NOT NULL	Số điện thoại
<b>hoTen</b>	NVARCHAR(50)		Họ tên người dùng

❑ Sinh viên chọn một trong hai cách tạo Cơ sở dữ liệu:

❖ Cách 1: Tạo bằng code Java kết hợp với SQLite. Các phương thức chứa trong lớp SQLiteOpenHelper.java

- Tạo class **MyDatabaseHelper** kế thừa từ lớp **SQLiteOpenHelper**.
- Ghi đè lên hai phương thức **onCreate()** và **onUpgrade()**
  - ✓ **onCreate()** - Nơi mà chúng ta cần phải viết để tạo bảng. Nó được gọi (called) khi cơ sở dữ liệu được tạo ra.
  - ✓ **onUpgrade()** - Phương thức này được gọi khi cơ sở dữ liệu được nâng cấp như thay đổi cấu trúc bảng, thêm ràng buộc cho cơ sở dữ liệu, v.v.

```

public class MyDatabaseHelper extends SQLiteOpenHelper {
    public static final String DATABASE_NAME = "bookstore.db";
    public static final String TABLE_NAME1 = "theLoaiSach";

    public MyDatabaseHelper(Context context) {
        super(context, DATABASE_NAME, factory: null, version: 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table " + TABLE_NAME1 + " (maLoai INTEGER PRIMARY KEY AUTOINCREMENT, " +
            "tenTheLoai TEXT,moTa TEXT,viTri INTEGER)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME1);
        onCreate(db);
    }
}

```

- ❑ Mỗi bảng cần viết các câu lệnh SQL cần thiết để sử dụng trong phần lập trình sau này
- ❑ Ví dụ:
  - ❖ Thêm thể loại sách

```
public boolean insertTheLoai(String ten,String mota,int vitri) {  
    SQLiteDatabase db = this.getWritableDatabase();  
    ContentValues contentValues = new ContentValues();  
    contentValues.put("tenTheLoai",ten);  
    contentValues.put("moTa",mota);  
    contentValues.put("viTri",vitri);  
    long result = db.insert(TABLE_NAME1, nullColumnHack: null ,contentValues);  
    if(result == -1)  
        return false;  
    else  
        return true;  
}
```

## ❑ Cập nhật thể loại sách

```
public boolean updateTheLoai(String id,String ten,String mota,int vitri) {  
    SQLiteDatabase db = this.getWritableDatabase();  
    ContentValues contentValues = new ContentValues();  
    contentValues.put("maLoai",id);  
    contentValues.put("tenTheLoai",ten);  
    contentValues.put("moTa",mota);  
    contentValues.put("viTri",vitri);  
    db.update(TABLE_NAME1, contentValues, whereClause: "maLoai = ?",new String[] { id });  
    return true;  
}
```

## ❑ Xóa 1 loại sách

```
public Integer deleteTheLoai (String id) {  
    SQLiteDatabase db = this.getWritableDatabase();  
    return db.delete(TABLE_NAME1, whereClause: "ID = ?",new String[] {id});  
}
```

## ❑ Truy vấn toàn bảng

```
public Cursor getAllTheLoai() {  
    SQLiteDatabase db = this.getWritableDatabase();  
    Cursor res = db.rawQuery( sql: "select * from "+TABLE_NAME1, selectionArgs: null);  
    return res;  
}
```

## ❑ Cách 2: Tạo bằng phần mềm SQLite Administrator, sau đó đưa file CSDL vào máy ảo bằng cách sử dụng DDMS

Table Name:

Columns: (Fields with a \* are required)

	Name*	Type*	Length or Precision	Scale	Not Null	Primary Key	Unique	Default Value
1.	<input type="text" value="maTheLoai"/>	<input type="text" value="INTEGER"/>	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
2.	<input type="text" value="tenTheLoai"/>	<input type="text" value="TEXT"/>	<input type="text" value="25"/>	<input type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
3.	<input type="text" value="moTa"/>	<input type="text" value="TEXT"/>	<input type="text" value="25"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
4.	<input type="text" value="viTri"/>	<input type="text" value="INTEGER"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>

```
CREATE TABLE theLoaiSach
(maTheLoai INTEGER NOT NULL,
tenTheLoai TEXT(25) NOT NULL,
moTa TEXT(25),
viTri INTEGER,
PRIMARY KEY (maTheLoai))
```



- ❑ **SQLite** là một cơ sở dữ liệu quan hệ, mã nguồn mở, nó được tích hợp sẵn trên hệ điều hành **Android**, vì vậy bạn có thể sử dụng nó bất cứ lúc nào, và không cần phải cấu hình gì thêm.
- ❑ Thay vì lưu trữ bằng text file, XML hay SharedPreferences thì có thể lưu trữ thông tin bằng SQLite.
- ❑ SQLite đã được tích hợp sẵn trong Android SDK



## **SAMPLE PROJECT**

### **BÀI 3: THỰC HIỆN DỰ ÁN 1**

# Lập trình CSDL theo mô hình DAO

- ❑ UI: giao diện (chứa các màn hình Activity)
- ❑ DB: cơ sở dữ liệu
- ❑ Model: các lớp mô tả dữ liệu theo cấu trúc các bảng
- ❑ DAO (Data Access Object): là các lớp thao tác và truy vấn dữ liệu. Nó có nhiệm vụ chuyển đổi Model sang SQL và ngược lại.

12:51

← THÊM NGƯỜI DÙNG

**FPT** Education  
**FPT POLYTECHNIC**

tielongnu

.....

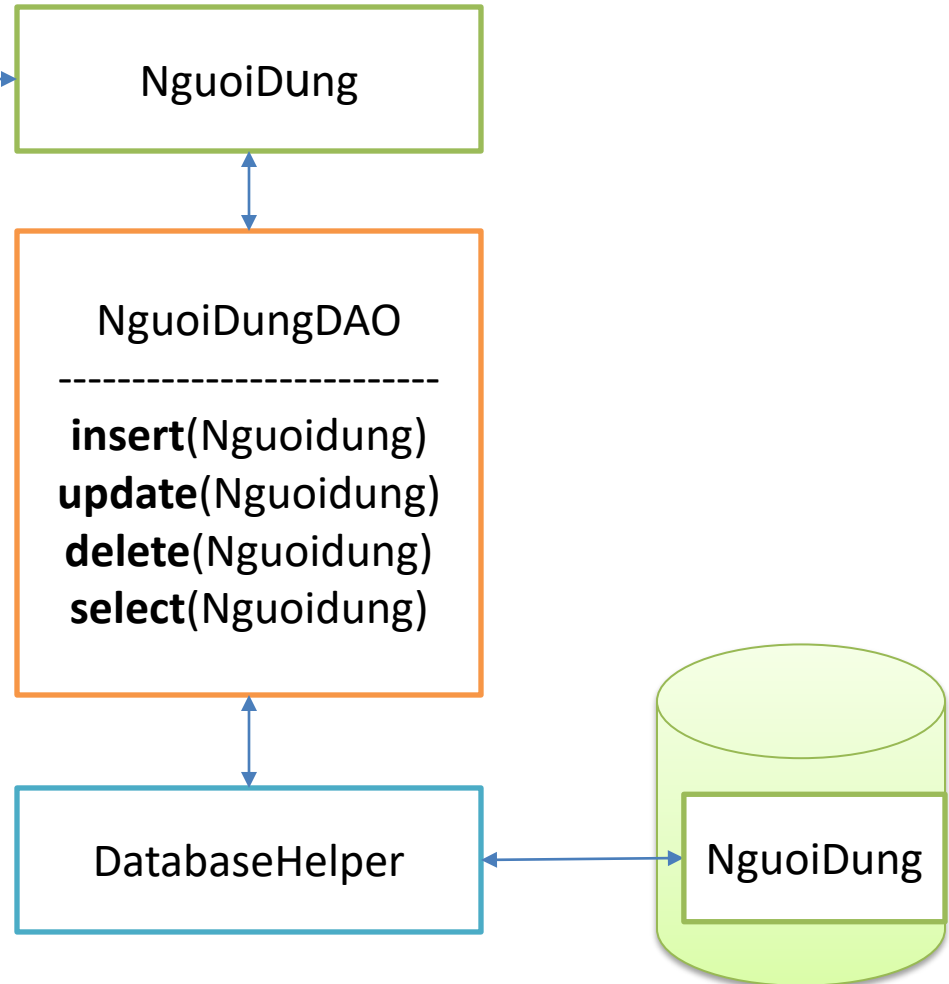
.....









0905999888

Tieu Long Nu

LƯU HỦY

DANH SÁCH NGƯỜI DÙNG



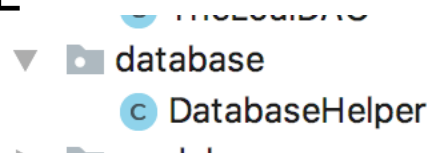
- ▼  **app**
  - ▶  manifests
  - ▼  java
    - ▼  com.edu.fpoly.bookmanager
      - ▶  adapter
      - ▶  dao
      - ▶  database
      - ▶  model
      - ChangePasswordActivity
      - DatePickerFragment
      - HoaDonActivity
      - ListBookActivity
      - ListNguoiDungActivity
      - ListTheLoaiActivity
      - LoginActivity
      - MainActivity
      - NguoiDungActivity
      - NguoiDungDetailActivity
      - SplashScreenActivity
      - TheLoaiActivity
      - ThemSachActivity

❑ Ứng dụng được chia làm 3 tầng:

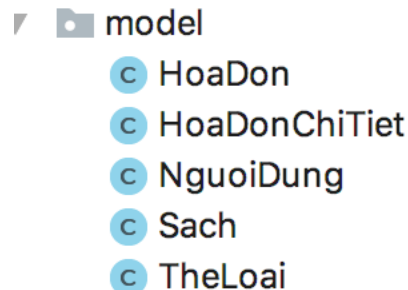
❖ DAO



❖ DATABASE



❖ MODEL



- ❑ **Model** là các lớp chứa thông tin đối tượng (Dữ liệu), tương tác với Database.
- ❑ Chịu trách nhiệm chính trong mô hình hóa đối tượng
- ❑ Ví dụ:

```
public class NguoIDung {  
    private String userName;  
    private String password;  
    private String phone;  
    private String hoTen;  
  
    public NguoIDung() {  
    }  
  
    public NguoIDung(String userName, String password, String phone, String hoTen) {  
        this.userName = userName;  
        this.password = password;  
        this.phone = phone;  
        this.hoTen = hoTen;  
    }  
  
    public String getUserName() { return userName; }  
  
    public void setUserName(String userName) { this.userName = userName; }  
  
    public String getPassword() { return password; }  
  
    public void setPassword(String password) { this.password = password; }  
  
    public String getPhone() { return phone; }  
  
    public void setPhone(String phone) { this.phone = phone; }  
  
    public String getHoTen() { return hoTen; }  
  
    public void setHoTen(String hoTen) { this.hoTen = hoTen; }  
}
```



- ❑ Chứa lớp DatabaseHelper thừa kế từ lớp SQLiteOpenHelper, để tạo CSDL

```
public class DatabaseHelper extends SQLiteOpenHelper {  
  
    public static final String DATABASE_NAME = "dbBookManager";  
    public static final int VERSION = 1;  
    public DatabaseHelper(Context context) {  
        super(context, DATABASE_NAME, null, VERSION);  
    }  
  
    @Override  
    public void onCreate(SQLiteDatabase db) {  
        db.execSQL(NguoiDungDAO.SQL_NGUOI_DUNG);  
        db.execSQL(TheLoaiDAO.SQL_THE_LOAI);  
        db.execSQL(SachDAO.SQL_SACH);  
        db.execSQL(HoaDonDAO.SQL_HOADON);  
        db.execSQL(HoaDonChiTietDAO.SQL_HOADONCHITTIET);  
    }  
  
    @Override  
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
        db.execSQL("Drop table if exists "+NguoiDungDAO.TABLE_NAME);  
        db.execSQL("Drop table if exists "+TheLoaiDAO.TABLE_NAME);  
        db.execSQL("Drop table if exists "+SachDAO.TABLE_NAME);  
        db.execSQL("Drop table if exists "+HoaDonDAO.TABLE_NAME);  
        db.execSQL("Drop table if exists "+HoaDonChiTietDAO.TABLE_NAME);  
        onCreate(db);  
    }  
}
```

- ❑ Tạo lớp DatabaseHelper để làm việc với cơ sở dữ liệu **SQLite**, lớp này kế thừa từ lớp **SQLiteOpenHelper**.
- ❑ Có 2 phương thức quan trọng cần phải ghi đè (override) là **onCreate()** và **onUpgrade()**.
  - **onCreate()** - Nơi mà chúng ta cần phải viết để tạo bảng. Nó được gọi (called) khi cơ sở dữ liệu được tạo ra.
  - **onUpgrade()** - Phương thức này được gọi khi cơ sở dữ liệu được nâng cấp như thay đổi cấu trúc bảng, thêm ràng buộc cho cơ sở dữ liệu, v.v.

- ❑ Chứa các thao tác tương tác với các bảng trong CSDL
- ❑ Bao gồm các phương thức:
  - ❖ Insert()
  - ❖ Update()
  - ❖ Delete()
  - ❖ getListAll ()
  - ❖ ...

```

public class NguoiDungDAO {
    private SQLiteDatabase db;
    private DatabaseHelper dbHelper;

    public static final String TABLE_NAME = "NguoiDung";
    public static final String SQL_NGUOI_DUNG = "CREATE TABLE NguoiDung (username text primary key, password text, phone text, hoten text);";
    public static final String TAG = "NguoiDungDAO";
    public NguoiDungDAO(Context context) {
        dbHelper = new DatabaseHelper(context);
        db = dbHelper.getWritableDatabase();
    }

    //insert
    public int insertNguoiDung(NguoiDung nd){...}
    //getAll
    public List<NguoiDung> getAllNguoiDung(){...}
    //update
    public int updateNguoiDung(NguoiDung nd){...}
    public int changePasswordNguoiDung(NguoiDung nd){...}
    public int updateInfoNguoiDung(String username,String phone, String name){...}

    //delete
    public int deleteNguoiDungByID(String username){...}
    //check login
    public int checkLogin(String username, String password){...}
}

```

```
//insert
public int insertNguoiDung(NguoiDung nd){
    ContentValues values = new ContentValues();
    values.put("username",nd.getUserName());
    values.put("password",nd.getPassword());
    values.put("phone",nd.getPhone());
    values.put("hoten",nd.getHoTen());
    try {
        if(db.insert(TABLE_NAME, nullColumnHack: null,values)== -1){
            return -1;
        }
    } catch (Exception ex){
        Log.e(TAG,ex.toString());
    }
    return 1;
}
```

```
//update
public int updateNguoiDung(NguoiDung nd){
    ContentValues values = new ContentValues();
    values.put("username",nd.getUserName());
    values.put("password",nd.getPassword());
    values.put("phone",nd.getPhone());
    values.put("hoten",nd.getHoTen());
    int result = db.update(TABLE_NAME,values, whereClause: "username=?", new String[]{nd.get
    if (result == 0){
        return -1;
    }
    return 1;
}
```

```
//delete
public int deleteNguoiDungByID(String username){
    int result = db.delete(TABLE_NAME, whereClause: "username=?", new String[]{username});
    if (result == 0)
        return -1;
    return 1;
}
```

```

//getAll
public List<NguoiDung> getAllNguoiDung(){
    List<NguoiDung> dsNguoiDung = new ArrayList<>();
    Cursor c = db.query(TABLE_NAME, columns: null, selection: null, selectionArgs: null, groupBy: null, having: null, orderBy: null);
    c.moveToFirst();
    while (c.isAfterLast()==false){
        | NguoiDung ee = new NguoiDung();
        ee.setUserName(c.getString( i: 0));
        ee.setPassword(c.getString( i: 1));
        ee.setPhone(c.getString( i: 2));
        ee.setHoTen(c.getString( i: 3));
        dsNguoiDung.add(ee);
        Log.d( tag: "//====", ee.toString());
        c.moveToNext();
    }
    c.close();
    return dsNguoiDung;
}

```



- ⊙ Thiết kế CSDL
- ⊙ Ôn lại kiến thức lập trình với SQLite
- ⊙ Lập trình CSDL với mô hình DAO
  - **Cách tạo / xóa một cơ sở dữ liệu SQLite trong Android**
  - **Cách tạo / xóa bảng trong SQLite**
  - **Cách thêm/ sửa/ xóa dữ liệu trong bảng**
  - **Cách truy vấn dữ liệu trong bảng**



**Cảm ơn**