

# Lập trình với SQLite

---

GV: TRƯƠNG BÁ THÁI

Email: [truongbathai@tdc.edu.vn](mailto:truongbathai@tdc.edu.vn)

ĐT: 0932.577.765

# | Mục tiêu:

- Trình bày được các khái niệm về SQLite
- Trình bày được các bước thực hiện database trong SQLite
- Tạo được database trong SQLite
- Thực hiện được các thao tác dữ liệu trong Sqlite



# Giới thiệu về SQLite

# | Giới thiệu về SQLite

- SQLite là cơ sở dữ liệu mở được viết dưới dạng thư viện tích hợp **nhúng** vào Android, hỗ trợ các đặc điểm về quan hệ chuẩn của cơ sở dữ liệu như cú pháp, transaction, các câu lệnh. SQLite được sử dụng rộng rãi trong các ứng dụng di động trên Android, iOS.

- SQLite hỗ trợ các kiểu dữ liệu : ***TEXT, INTEGER, REAL.***
- Đường dẫn của cơ sở dữ liệu: ***DATA/data//databases/FILENAME***





# Sử dụng SQLite trong Android

# | Tạo và xóa cơ sở dữ liệu

- Tạo cơ sở dữ liệu thông qua lớp **SQLiteOpenHelper**.
- **SQLiteOpenHelper** là một lớp ảo, SQLiteOpenHelper giúp tạo các cơ sở dữ liệu dùng SQLite.
- SQLiteOpenHelper thực hiện các phương thức cần thiết cho phép khởi tạo, nâng cấp cơ sở dữ liệu. Tạo đối tượng để truy cập cơ sở dữ liệu (Read và Write).



- **SQLiteDatabase** cung cấp phương thức **insert()**, **update()**, **delete()**, hoặc **execSQL()** cho phép thực hiện truy xuất dữ liệu và Override phương thức **onCreate()** để tạo cơ sở dữ liệu. Override phương thức **onUpgrade()** để nâng cấp cơ sở dữ liệu.

- Override phương thức **onCreate()** để tạo cơ sở dữ liệu.
- Override phương thức **onUpgrade()** để nâng cấp cơ sở dữ liệu.
- Lớp SQLiteOpenHelper cung cấp 2 phương thức **getReadableDatabase()** và **getWritableDatabase()** để trả về đối tượng SQLiteDatabase

# Ví dụ: Tạo một CSDL có tên: COUNTRY\_DB

## Trong đó:

- Tạo một Table có tên: COUNTRY.
- Table COUNTRY có 3 cột là: \_id, enName và viName với \_id là khóa chính và có giá trị tự động tăng.

```
private static final String DB_NAME= "Country_db";  
public static final String TABLE_NAME = "Country";  
public static final String COL_ID = "_id";  
public static final String COL_EN_NAME = "enName";  
public static final String COL_VI_NAME = "viName";  
private static final String CREATE_TABLE = ""  
    + "create table " + TABLE_NAME + " ( "  
    + COL_ID + " integer primary key autoincrement ,"  
    + COL_EN_NAME + " text not null, "  
    + COL_VI_NAME + " text not null );"  
  
public MySqliteHelper(Context context) {  
    super(context, TABLE_NAME, null, 1);  
}
```

# | Hàm khởi tạo có các thông số sau

- **Context**: Biến ngữ cảnh.
- **DB\_NAME**: Tên của cơ sở dữ liệu.
- **CursorFactory**: Đôi lúc chúng ta có thể extend lớp cursor để kế thừa một số phương thức và truy vấn. Trong trường hợp đó, ta dùng một instance của CursorFactory để tham chiếu đến lớp chúng ta tạo thay cho mặc định. Khi dùng mặc định thì ta để nó null.
- **Version**: Version của cơ sở dữ liệu.

# Tiến hành tạo CSDL trong hàm onCreate():

```
public void onCreate(SQLiteDatabase db) { db.execSQL(CREATE_TABLE);  
}
```

*@Override*

```
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
  
}
```





# Mở Cơ sở dữ liệu



# | Mở Cơ sở dữ liệu

- Dùng cú pháp **getReadableDatabase()** và **getWritableDatabase()** để trả về đối tượng SQLiteDatabase cần sử dụng.
- SQLiteDatabase là đối tượng để truy cập cơ sở dữ liệu (Read và Write).
- SQLiteDatabase cung cấp phương thức **insert()**, **update()**, **delete()**, hoặc **execSQL()** cho phép thực hiện truy xuất dữ liệu.

```
public class MyCounTryDB {  
    SQLiteDatabase database;  
    MySqliteHelper myHelper;  
    public MyCounTryDB(Context context) {  
        myHelper = new MySqliteHelper(context);  
        try {  
            database = myHelper.getWritableDatabase();  
        } catch (SQLException ex) {  
            database = myHelper.getReadableDatabase();  
        }  
    }  
}
```

# | Đóng Cơ sở dữ liệu

```
public void close()  
{  
  
    myHelper.close();  
  
}
```

# | Truy vấn dữ liệu trong các bảng

- Truy vấn (Query): Sử dụng phương thức ***rawQuery()*** của lớp SQLiteDatabase. Phương thức ***rawQuery()*** nhận vào một giá trị chuỗi là câu lệnh SQL dùng để truy vấn dữ liệu và trả ra một đối tượng Cursor.

```
public Cursor Select_ALL() {  
    String sql="Select * From CounTry where _id = ? and enName =?";  
    Cursor cursor = database.rawQuery(sql,new String[]{"1","VietNam"});  
    return cursor;  
}
```

# | Truy vấn với cú pháp hàm

- Sử dụng phương thức `query()` của lớp `SQLiteDatabase`.
- Phương thức ***query()*** cũng trả ra một đối tượng **Cursor**. Để lấy dữ liệu từ `Cursor`, trước hết cần chuyển đến vị trí xác định với các phương thức:  
***moveToFirst()***, ***moveToNext()***,  
***moveToPosition(int position)***

```
Cursor cursor = database.query(String table,  
                                String[] columns,  
                                String selection,  
                                String[] selectionArgs,  
                                String groupBy,  
                                String having,  
                                String orderBy);
```

- String table: tên của bảng cần truy vấn.
- String[] columns: danh sách các cột sẽ trả về dữ liệu.
- String selection: chứa các điều kiện truy vấn.
- String[] selectionArgs: danh sách các tham số phụ cho câu điều kiện.



- `String[] groupBy`: gom nhóm các cột kết quả.
- `String[] having`: bộ lọc theo điều kiện.
- `String[] orderBy`: sắp xếp theo mảng cột được chỉ định.

```

public ArrayList<Country> Select_query() {
    ArrayList<Country> data = new ArrayList<>();
    String[] select_Column = {"id", "enName", "viName"};
    Cursor cursor = database.query("Country", select_Column, null, null, null,
    null, null);
    if (cursor.moveToFirst()) {
        data.clear();
        while (!cursor.isAfterLast()) {
            int id =
Integer.parseInt(cursor.getString(cursor.getColumnIndex(select_Column[0])));
            String enName =
cursor.getString(cursor.getColumnIndex(select_Column[1]));
            String viName =
cursor.getString(cursor.getColumnIndex(select_Column[2]));
            Country counTry = new Country(id, enName, viName);
            data.add(counTry);
        }
    }
    return data;
}

```

- Lớp SQLiteDatabase cung cấp các phương thức như: insert(), update(), delete() hoặc execSQL() cho phép người dùng có thể quản lý và truy xuất dữ liệu.

# | Insert

```
public long Insert(CounTry counTry) {  
    ContentValues values = new ContentValues();  
    values.put("enName", counTry.getEnName());  
    values.put("viName", counTry.getViName());  
    return database.insert("CounTry", null, values);  
}
```

# | Update

```
public long update(CounTry counTry) {  
    ContentValues values = new ContentValues();  
    values.put("enName", counTry.getEnName());  
    values.put("viName", counTry.getViName());  
    return database.update("CounTry", values, "_id = " + counTry.getId(), null);  
}
```

# Delete

```
public long Delete(CounTry counTry) {  
    return database.delete("Country", "id = " + counTry.getId(), null);  
}
```



Sắp xếp dữ liệu

# | Tăng dần (ASCENDING)

- Truy vấn với câu lệnh SQL:

```
SELECT * FROM COUNTRY ORDER BY enName ASC
```

- Truy vấn với cú pháp hàm:

```
String orderBy = myHelper.COL_EN_NAME + " ASC";  
database.query(TABLE_NAME, null, null, null, null, orderBy);
```



# | Giảm dần (DESCENDING)

`ORDER BY columnName DESC`

`ORDER BY columnName1 + columnName2 DESC`

`database.query(TABLE_NAME, null, null, null, null, orderBy);`

# Xây dựng chương trình quản lý môn học



## Quản Lí Môn Học

Mã MH:

Tên MH:

Số Tiết:

## Danh Sách Môn Học

	Mã MH: 1 Tên MH: Lap <u>Tr</u> inh <u>Đi</u> Dong 1 Số Tiết: 60	
---	---	---

# | Bước 1:

- Tạo một project tên là QLMonhoc: **File → New → Android Application Project** điền các thông tin → **Next → Finish.**

## Bước 2:

- Vào thư mục **res/values** bổ sung **string.xml**

- ```
<resources>
    <string name="app_name">Test</string>
    <string name="qlmh">Quản Lí Môn Học</string>
    <string name="dsmh">Danh Sách Môn Học</string>
    <string name="mamonhoc">Mã MH:</string>
    <string name="tenmonhoc">Tên MH:</string>
    <string name="sotiet">Số Tiết:</string>
    <string name="nhapmonhoc">Insert</string>
    <string name="loadmonhoc">Load</string>
    <string name="update">Update</string>
    <string name="delete">Delete</string>
</resources>
```

## | Bước 3:

- Mở **res** → **layout**

# Tạo file listview\_item.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginRight="30dp"
        android:layout_weight="1"
        android:orientation="vertical">
        <TextView
            android:id="@+id/tvTenSv"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            />
    </LinearLayout>
</LinearLayout>
```

*<TextView*

*android:id="@+id/tvLop"*

*android:layout\_width="match\_parent"*

*android:layout\_height="wrap\_content"*

*/>*

*</LinearLayout>*

*<ImageView*

*android:id="@+id/imDelete"*

*android:layout\_width="wrap\_content"*

*android:layout\_height="wrap\_content"*

*android:layout\_gravity="center"*

*android:src="@drawable/ic\_delete\_forever\_black\_24dp"*

*/>*

*</LinearLayout>*

# Tạo file main\_activity.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MonHoc.Activity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="2"
        android:orientation="vertical">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="#787878"
            android:gravity="center"
            android:text="@string/qlmh"
            android:textColor="#ffffff"
            android:textSize="20sp"
            android:textStyle="bold" />
```



*<LinearLayout*

*android:layout\_width="match\_parent"*

*android:layout\_height="wrap\_content"*

*android:orientation="horizontal">*

*<TextView*

*android:layout\_width="wrap\_content"*

*android:layout\_height="wrap\_content"*

*android:text="@string/mamonhoc" />*

*<EditText*

*android:id="@+id/mamonhoc"*

*android:layout\_width="match\_parent"*

*android:layout\_height="wrap\_content" />*

*</LinearLayout>*

*<LinearLayout*

*android:layout\_width="match\_parent"*

*android:layout\_height="wrap\_content"*

*android:orientation="horizontal">*

*<TextView*

*android:layout\_width="wrap\_content"*

*android:layout\_height="wrap\_content"*

*android:text="@string/tenmonhoc" />*

*<EditText*

*android:id="@+id/tenmonhoc"*

*android:layout\_width="match\_parent"*

*android:layout\_height="wrap\_content" />*

*</LinearLayout>*

*<LinearLayout*

*android:layout\_width="match\_parent"*

*android:layout\_height="wrap\_content"*

*android:orientation="horizontal">*

*<TextView*

*android:layout\_width="wrap\_content"*

*android:layout\_height="wrap\_content"*

*android:text="@string/sotiet" />*

*<EditText*

*android:id="@+id/sotiethoc"*

*android:layout\_width="match\_parent"*

*android:layout\_height="wrap\_content" />*

*</LinearLayout>*

*</LinearLayout>*

*<LinearLayout*

*android:layout\_width="match\_parent"*

*android:layout\_height="wrap\_content"*

*android:orientation="horizontal">*

*<Button*

*android:id="@+id/btnNhapMH"*

*android:layout\_width="wrap\_content"*

*android:layout\_height="wrap\_content"*

*android:text="@string/nhapmonhoc" />*

*<Button*

*android:id="@+id/btnLoad"*

*android:layout\_width="wrap\_content"*

*android:layout\_height="wrap\_content"*

*android:text="@string/loadmonhoc" />*

*<Button*

```
android:id="@+id/btnUpdate"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="@string/update" />
```

*<Button*

```
android:id="@+id/btnDelete"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="@string/delete" />
```

*</LinearLayout>*

<LinearLayout

*android:layout\_width="match\_parent"*

*android:layout\_height="0dp"*

*android:layout\_weight="3"*

*android:orientation="vertical">*

<TextView

*android:layout\_width="match\_parent"*

*android:layout\_height="wrap\_content"*

*android:background="#787878"*

*android:text="@string/dsmh"*

*android:textColor="#ffffff"*

*android:textSize="20sp"*

*android:textStyle="bold" />*

<ListView

*android:id="@+id/listView1"*

*android:layout\_width="match\_parent"*

*android:layout\_height="match\_parent">*

</ListView>

</LinearLayout>|

</LinearLayout>

# | Bước 4

- Mở **app** → **src**

# | Tạo file MonHoc.java

```
public class MonHoc {  
    private int img;  
    private String ten, ma, sotiet;  
  
    public MonHoc() {  
    }  
  
    public MonHoc(int img, String ten, String ma, String sotiet) {  
        this.img = img;  
        this.ten = ten;  
        this.ma = ma;  
        this.sotiet = sotiet;  
    }
```



# Tạo file MonHocAdapter.java

```
public class MonHocAdapter extends ArrayAdapter<MonHoc> {  
    Context context;  
    int layoutResourceId;  
    ArrayList<MonHoc> data = null;  
  
    public MonHocAdapter(Context context, int layoutResourceId,  
        ArrayList<MonHoc> data) {  
        super(context, layoutResourceId, data);  
        this.context = context;  
        this.layoutResourceId = layoutResourceId;  
        this.data = data;  
    }  
  
    static class MonHocHolder {  
        ImageView img,imgDelete;  
        TextView txtMaMh, txtTenMH, txtSoTiet;  
    }  
}
```

```

public View getView(final int position, View convertView, ViewGroup parent) {
    View row = convertView;
    MonHocHolder holder = null;
    if(row != null)
    {
        holder = (MonHocHolder) row.getTag();
    }
    else
    {
        holder = new MonHocHolder();
        LayoutInflater inflater = ((Activity)context).getLayoutInflater();
        row = inflater.inflate(R.layout.list_item_row, parent, false);

        holder.img = (ImageView) row.findViewById(R.id.tdc);
        holder.txtMaMh = (TextView) row.findViewById(R.id.mamh);
        holder.txtTenMH = (TextView) row.findViewById(R.id.tenmh);
        holder.txtSoTiet = (TextView) row.findViewById(R.id.sotiet);
        holder.imgDelete = (ImageView) row.findViewById(R.id.imgDelete);
        row.setTag(holder);
    }
}

```

```

final MonHoc mh = data.get(position);
holder.img.setImageResource(R.drawable.tdc);
holder.txtMaMh.setText("Mã MH: " + mh.getMa());
holder.txtTenMH.setText("Tên MH: " + mh.getTen());
holder.txtSoTiet.setText("Số Tiết: " + mh.getSotiet());
holder.imgDelete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        MonHocDatabase db = new MonHocDatabase(context);
        db.deleteMH(mh.getMa());
        Toast.makeText(context, "Đã xóa mã"+ mh.getMa(),
        Toast.LENGTH_SHORT).show();
    }
});

return row;
}
}

```

# Tạo file MonHocDataBase.java

```
public class MonHocDatabase extends SQLiteOpenHelper {  
    private static String DB_NAME = "dbMonHoc.db";  
    private static int DB_VERSION = 1;  
    //Define table Monhoc  
    private static final String TB_MONHOCS = "tbMonHoc";  
    private static final String COL_MONHOC_ID = "monhoc_id";  
    private static final String COL_MONHOC_MA = "monhoc_ma";  
    private static final String COL_MONHOC_TEN = "monhoc_ten";  
    private static final String COL_MONHOC_SOTIET = "monhoc_sotiet";  
  
    public MonHocDatabase(Context context)  
    {  
        super(context, DB_NAME, null, DB_VERSION);  
    }  
}
```

### *@Override*

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

### *@Override*

```
public void onCreate(SQLiteDatabase db) {  
    String scriptTBMonHocs = "CREATE TABLE " + TB_MONHOCS + "(" +  
        COL_MONHOC_ID + " INTEGER PRIMARY KEY  
AUTOINCREMENT NOT NULL, " +  
        COL_MONHOC_MA + " TEXT, " +  
        COL_MONHOC_TEN + " TEXT, " +  
        COL_MONHOC_SOTIET + " TEXT) ";  
    //Execute script  
    db.execSQL(scriptTBMonHocs);  
}
```

```

public void getMonHoc(ArrayList<MonHoc> monHocs)
{
    SQLiteDatabase db = getWritableDatabase();
    Cursor cursor = db.query(TB_MONHOCS, new
String[]{COL_MONHOC_MA, COL_MONHOC_TEN,
COL_MONHOC_SOTIET}, null,null,null, null, null);
    if(cursor.moveToFirst()) {
        do {
            MonHoc mh = new MonHoc();

            mh.setMa(cursor.getString(cursor.getColumnIndex(COL_MONHOC_MA)));

            mh.setTen(cursor.getString(cursor.getColumnIndex(COL_MONHOC_TEN)));

            mh.setSotiet(cursor.getString(cursor.getColumnIndex(COL_MONHOC_SOTIET))
);

            monHocs.add(mh);
        } while (cursor.moveToNext());
    }
}

```

```
public void getAllData(ArrayList<MonHoc> monHocs) {  
    SQLiteDatabase db = this.getWritableDatabase();  
    String query = "select * from "+TB_MONHOCS;  
    Cursor cursor = db.rawQuery(query,null);  
    if(cursor.moveToFirst()) {  
        do {  
            MonHoc mh = new MonHoc();  
  
            mh.setMa(cursor.getString(cursor.getColumnIndex(COL_MONHOC_MA)));  
  
            mh.setTen(cursor.getString(cursor.getColumnIndex(COL_MONHOC_TEN)));  
  
            mh.setSotiet(cursor.getString(cursor.getColumnIndex(COL_MONHOC_SOTIET)  
));  
            monHocs.add(mh);  
        } while (cursor.moveToNext());  
    }  
}
```

```
public void saveMH(MonHoc mh)  
{  
    SQLiteDatabase db = getWritableDatabase();  
    ContentValues values = new ContentValues();  
    values.put(COL_MONHOC_MA, mh.getMa());  
    values.put(COL_MONHOC_TEN, mh.getTen());  
    values.put(COL_MONHOC_SOTIET, mh.getSotiet());  
    db.insert(TB_MONHOCS, null, values);  
    db.close();  
}
```



```
public void deleteMH(String maHH)
{
    SQLiteDatabase db = this.getWritableDatabase();
    String query = "DELETE FROM tbMonHoc WHERE monhoc_ma=" +
maHH;
    db.execSQL(query);
}
```

```
public void updateMH(MonHoc mh)
{
    SQLiteDatabase db = this.getWritableDatabase();
    String sql = "Update "+ TB_MONHOCS +" set ";
    sql += COL_MONHOC_TEN +" = '"+ mh.getTen()+"', ";
    sql += COL_MONHOC_SOTIET +" = '"+ mh.getSotiet()+"' ";
    sql += " WHERE "+ COL_MONHOC_MA +" = '"+ mh.getMa();

    db.execSQL(sql);
}
}
```





**CẢM ƠN TẤT CẢ  
ĐÃ LẮNG NGHE**