

Lập trình thiết bị di động

Ts. Lâm Chí Nguyễn

Chương 4 - Database

- Lưu dữ liệu bằng cơ sở dữ liệu cục bộ SQLite
 - 2000, D.Richard Hipp viết mã nguồn bằng ngôn ngữ C
 - Tạo ra CSDL không cần phần mềm quản trị
- Ưu điểm của SQLite
 - Nhu cầu không gian lưu trữ nhỏ
 - Dễ dàng sử dụng
 - Phần mềm mã nguồn mở
 - Sử dụng chuẩn câu lệnh SQL92,
- Android team : đang cung cấp một thư viện ROOM là một phiên bản mở rộng từ SQLite

Chương 4 - Database

- Đặc điểm của SQLite
 - Hoạt động trên cơ chế không có phần server (stand alone)
 - Cơ sở dữ liệu được lưu trữ trong một tập tin
 - Số lượng kiểu dữ liệu ít
 - Kiểu dữ liệu mỗi trường không bị ràng buộc bởi thông tin lúc tạo ra
 - Không giới hạn kích thước cột
 - Hỗ trợ tìm kiếm bằng văn bản
- Nơi lưu trữ trên hệ điều hành
 - `/data/data/package-name/databases`
 - Ví dụ :
 - `/data/data/com.grokkingandroid.android/databases/sample.db.`

Chương 4 - Database

- Lưu dữ liệu bằng cơ sở dữ liệu cục bộ SQLite
 - Gói thư viện android.database.*
 - Gói thư viện android.content.*
- Các lớp đối tượng cần thiết :
 - SQLiteOpenHelper
 - SQLiteDatabase
 - Cursor
 - DatabaseUtils
 - ContentValues

Chương 4 - Database

- SQLiteOpenHelper : Class hỗ trợ truy xuất đến cơ sở dữ liệu
- LTV cần phát triển lớp thừa kế từ SQLiteOpenHelper.

```
public class DataHandle extends SQLiteOpenHelper {
    String CREATE_TABLE="";
    String DROP_TABLE = "";
    Context context;
    public DataHandle(Context context) {
        super(context, dbName, null, version);
        context=this.context;
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        try {
            db.execSQL(CREATE_TABLE);
        } catch (Exception e) {

        }
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int i, int i1) {
        db.execSQL(DROP_TABLE);
        onCreate(db);
    }
}
```

Chương 4 - Database

- Sử dụng SQLiteDatabase trong lớp thừa kế SQLiteOpenHelper.
- Thêm một mẫu tin

```
public boolean InsertRecord(Entity obj)
{
    SQLiteDatabase db=this.getWritableDatabase();
    ContentValues cv = new ContentValues();
    cv.put("Col name 1",obj.getValue1());
    cv.put("Col name 2",obj.getValue2());
    cv.put("Col name 3",obj.getValue3());

    long result = db.insert("TABLE_NAME",null,cv);
    if(result == -1)
        return false; // thất bại
    else
        return true;  // thành công
}
```

Chương 4 - Database

- Sử dụng SQLiteDatabase trong lớp thừa kế SQLiteOpenHelper.
- thay đổi nội dung mẫu tin

```
public boolean updateRecord (Entity obj) {  
    SQLiteDatabase db = this.getWritableDatabase();  
    ContentValues contentValues = new ContentValues();  
    contentValues.put("Col name 1",obj.getValue1());  
    contentValues.put("Col name 2",obj.getValue1());  
    contentValues.put("Col name 3",obj.getValue1());  
    db.update("table name", contentValues, "Điều kiện", "tham số điều kiện");  
    return true;  
}
```

Điều kiện : "id = ? And name= ?"

Tham số điều kiện : ["id123", "joe"]

Chương 4 - Database

- Sử dụng SQLiteDatabase trong lớp thừa kế SQLiteOpenHelper.
- xóa mẫu tin

```
public Integer deleteRecord ("value 1", "value 2",...) {  
    SQLiteDatabase db = this.getWritableDatabase();  
    return db.delete("Table name","điều kiện",[tham số điều kiện]);  
}
```

Điều kiện : "id = ? And name= ?"

Tham số điều kiện : ["id123", "joe"]

Chương 4 - Database

- Sử dụng SQLiteDatabase trong lớp thừa kế SQLiteOpenHelper.
- Tìm mẫu tin

```
public Cursor findRecord(int id,String name) {
    SQLiteDatabase db = this.getReadableDatabase();
    String SQL = "select * from contacts where id="+id+" and name = '"+name+"'";
    Cursor res = db.rawQuery(SQL, null );
    return res;
}
int number_of_record = res.getCount();
res.moveToFirst();
while(res.isAfterLast() == false){
    int id = res.getColumnIndex(COLUMN_NAME);
    fieldname_1 = res.getString(id1);
    fieldname_2 = res.getInt(id2);
    //res.getFloat(id2);
    //res.getLong(id2);
    //res.getDouble(id2);
    res.moveToNext();
}
```

Chương 4 - Database

- Sử dụng SQLiteDatabase trong lớp thừa kế SQLiteOpenHelper.
- tìm mẫu tin

```
public Cursor findRecord(int id,String name) {
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor res = db.query (tablename, ["col1", "col2", "col2"],
                           "id = ? And name = ?",
                           ["id123", "steven chau"])

    return res;
}
int number_of_record = res.getCount();
res.moveToFirst();
while(res.isAfterLast() == false){
    int id = res.getColumnIndex(COLUMN_NAME);
    fieldname_1 = res.getString(id1);
    fieldname_2 = res.getInt(id2);
    //res.getFloat(id2); res.getLong(id2); res.getDouble(id2);

    res.moveToNext();
}
```

Chương 4 - Database

- Sử dụng SQLiteDatabase trong lớp thừa kế SQLiteOpenHelper.
- Câu lệnh xử lý SQL chuẩn :
 - INSERT, UPDATE, DELETE → db.execSQL (String sql);
 - SELECT → db.rawQuery(String sql, null);

