



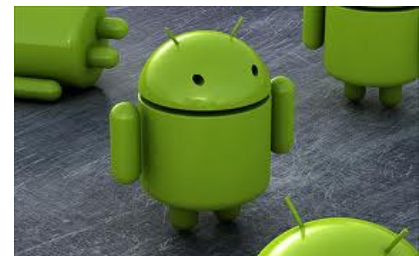
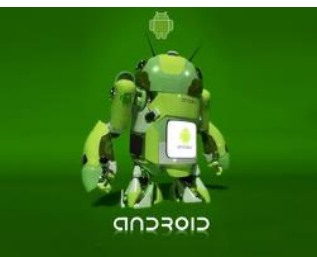
Trường Đại học Bách Khoa Hà Nội
Hanoi University of Science and Technology

Chapter 4. Graphical User Interfaces

Hanoi University of Science and Technology

Nguyen Hong Quang
19/8/2012

© HUST 2012





Nội dung

- Khai báo CSDL
- Tạo bảng CSDL
- Thực hiện truy vấn SQL
- Duyệt các hàng của kết quả truy vấn
- Kết nối kết quả truy vấn với ListView



SQLiteOpenHelper

Thao tác với CSDL bởi lớp SQLiteOpenHelper

```
class RestaurantHelper extends SQLiteOpenHelper {  
    private static final String DATABASE_NAME="lunchlist.db";  
    private static final int SCHEMA_VERSION=1;  
    public RestaurantHelper(Context context) {  
        super(context, DATABASE_NAME, null, SCHEMA_VERSION);  
    }  
    @Override  
    public void onCreate(SQLiteDatabase db) {  
    }  
    @Override  
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
    }  
}
```



Tạo bảng dữ liệu

@Override

```
public void onCreate(SQLiteDatabase db) {  
    db.execSQL("CREATE TABLE restaurants (_id  
        INTEGER PRIMARY KEY AUTOINCREMENT,  
        name TEXT, address TEXT, type TEXT,  
        notes TEXT);"  
    );  
}
```



Mở/đóng CSDL

```
RestaurantHelper helper = new  
    RestaurantHelper(this);
```

```
...
```

```
helper.close();
```



Insert dữ liệu vào bảng

```
SQLiteDatabase db = helper.getWritableDatabase();  
public void insert(String name, String address,  
String type, String notes) {  
    ContentValues cv=new ContentValues();  
    cv.put("name", name);  
    cv.put("address", address);  
    cv.put("type", type);  
    cv.put("notes", notes);  
    db.insert("restaurants", "name", cv);  
}
```



Xóa dữ liệu trong bảng

```
SQLiteDatabase db = helper.getWritableDatabase();
```

```
private void processDelete(long rowId) {  
    String[] args={String.valueOf(rowId)};  
    db.delete("constants", "_ID=?", args);  
}
```



Thực hiện truy vấn dữ liệu với phương thức rawquery()

```
SQLiteDatabase db = helper.getWritableDatabase();  
public Cursor getAll() {  
    return(  
        db.rawQuery("SELECT _id, name, address, type,  
            notes FROM restaurants ORDER BY name", null)  
    );  
}
```




Thực hiện truy vấn dữ liệu với phương thức query()

- The name of the table to query against
- The list of columns to retrieve
- The WHERE clause, optionally including positional parameters
- The list of values to substitute in for those positional parameters
- The GROUP BY clause, if any
- The HAVING clause, if any
- The ORDER BY clause, if any



Thực hiện truy vấn dữ liệu với phương thức query()

```
String[] columns={"ID", "inventory"};  
String[] parms={"snicklefritz"};  
Cursor result=db.query("widgets", columns,  
    "name=?", parms, null, null, null);
```

Duyệt kết quả truy vấn

Cursor

- Khai báo Cursor:
 - `Cursor cursor;`
- Các phương thức:
 - `getCount()`,
 - `moveToFirst()`, `moveToNext()`, `moveToPosition(position)`
 - `isAfterLast()`
 - `getColumnNames()`, `getColumnIndex()`
 - `getString()`, `getInt()`
 - `requery()`
 - `close()`

Lấy dữ liệu từ một hàng trong kết quả truy vấn

```
public Cursor getAll() {  
    return( getReadableDatabase().rawQuery("SELECT _id, name, address, " +  
        "type, notes FROM restaurants ORDER BY name", null));  
}  
  
public String getName(Cursor c) {  
    return(c.getString(1));  
}  
  
public String getAddress(Cursor c) {  
    return(c.getString(2));  
}  
  
public String getType(Cursor c) {  
    return(c.getString(3));  
}  
  
public String getNotes(Cursor c) {  
    return(c.getString(4));  
}
```



Ví dụ

```
Cursor result=
db.rawQuery("SELECT ID, name, inventory FROM widgets",
    null);
while (!result.moveToNext()) {
    int id = result.getInt(0);
    String name = result.getString(1);
    int inventory = result.getInt(2);
    // do something useful with these
}
result.close();
```

Kết nối kết quả truy vấn với

ListView

```
SimpleCursorAdapter adapter;  
adapter=new SimpleCursorAdapter(this,  
    R.layout.row, constantsCursor,  
    new String[] {DatabaseHelper.TITLE,  
        DatabaseHelper.VALUE},  
    new int[] {R.id.title, R.id.value}  
);
```

```
list.setAdapter(adapter);
```



Tùy biến ListView

- CursorAdapter and its subclasses have a default implementation of getView().
- What getView() does is inspect the supplied View to recycle and, if it is null, calls newView() then bindView(), or just calls bindView() if it is not null.



Ví dụ tùy biến ListView

```
public View newView(Context context, Cursor cursor,
ViewGroup parent) {
    LayoutInflater inflater=getLayoutInflater();
    View row=inflater.inflate(R.layout.row, null);
    ViewWrapper wrapper=new ViewWrapper(row);
    row.setTag(wrapper);
    return(row);
}

public void bindView(View row, Context context, Cursor cursor) {
    ViewWrapper wrapper=(ViewWrapper)row.getTag();
    // actual logic to populate row from Cursor goes here
}
```




Trường Đại học Bách Khoa Hà Nội

Hanoi University of Science and Technology

End of Lecture



Q&A

