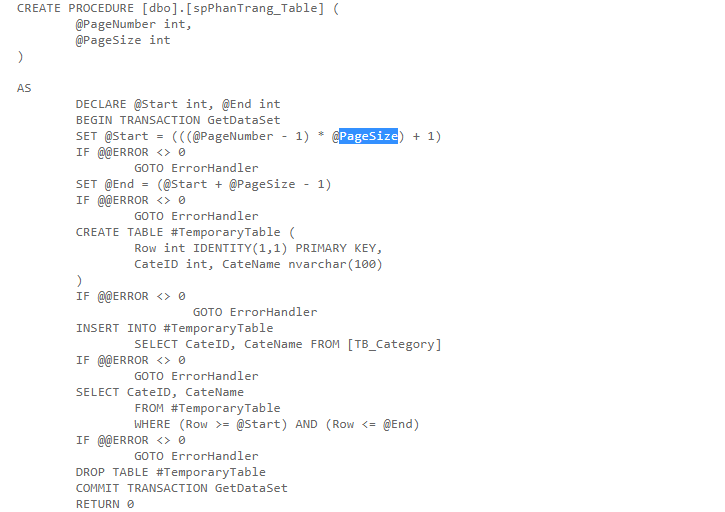
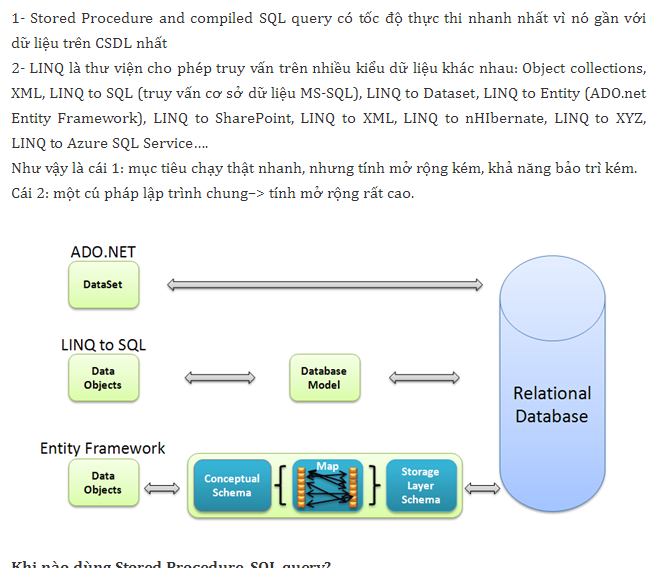
* List and IEnumerable
  + List: là một tập hợp chung. Cho phép truy cập ngẫu nhiên thông qua index.
  + IEnumerable: là một interface. Không cho phép truy cập ngẫu nhiên.
  + Hiệu xuất thì việc lặp ra các IEnumerable nhanh hơn so với một List.
* Get: không cho gửi nhiều parameter (maximum URL length is 2048 characters).
* Cache
* Authentication and Authorization:
* Phân trang trong cơ sở dữ liệu SQL SERVER



* Set id của user đó cho cookie
* Repository Pattern là gì?
* Repository Pattern là lớp trung gian giữa tầng Business Logic và Data Access, giúp cho việc truy cập dữ liệu chặt chẽ và bảo mật hơn.  
  Repository đóng vai trò là một lớp kết nối giữa tầng Business và Model của ứng dụng.  
  Hiểu đơn giản thì khi t muốn truy xuất dữ liệu từ database, thay vì viết code xử lý trong controller thì ta tạo ra 1 thư mục là Repository rồi viết code xử lý vào đây. Sau đó chúng ta chỉ việc inject vào thông qua \_\_construct.  
  Những lý do ta nên sử dụng mẫu Repository Pattern:
  + Code dễ dàng maintain.
  + Tăng tính bảo mật và rõ ràng cho code.
  + Lỗi ít hơn.
  + tránh việc lặp code.
* Trigger:  là một thủ tục SQL được thực thi ở phía server khi có một sự kiện như Insert, Delete, hay Update. Trigger thường được sử dụng để kiểm tra ràng buộc (check constraints) trên nhiều quan hệ (nhiều bảng/table) hoặc trên nhiều dòng (nhiều record) của bảng.Ngoài ra việc sử dụng Trigger để chương trình có những hàm chạy ngầm nhằm phục vụ nhưng trường hợp hữu hạn và thường không sử dụng cho mục đích kinh doanh hoặc giao dịch.
* Function:
* Stored:
* Cách xử lý dữ liệu lớn -> phân trang -> client,server -> sqlserver->store

Load 10 record -> 11->20

* 

# [Difference between virtual and abstract methods [duplicate]](https://stackoverflow.com/questions/14728761/difference-between-virtual-and-abstract-methods)

Virtual methods have an implementation and provide the derived classes with the option of overriding it. (có thân hàm, nội dung hàm cho các lớp kế thừa nó)

Abstract methods do not provide an implementation and force the derived classes to override the method. (không có thân hàm, chỉ có tên hàm, và bắt buộc phải dc ghi đè.)

Cookei – Session :

Unit test:

Read only, static

Role – Permission:

Nếu user đó là **admin(Role)** thì có quyền **xóa(permisstion)**

[username] -> database -> Cookei

* -- **preventDefault**: Huỷ bỏ event nếu nó có thể huỷ mà không dừng sự lan rộng(propagation) của event tới phần khác.

(ví dụ khi click thẻ a, nó ko chuyển trang nhưng vẫn thực hiện trong hàm)

* **stopPropagation** Ngăn chặn sự lan rộng của sự kiện hiện tại tới thằng khác. (không lan truyền si75 kiện qua element parent)

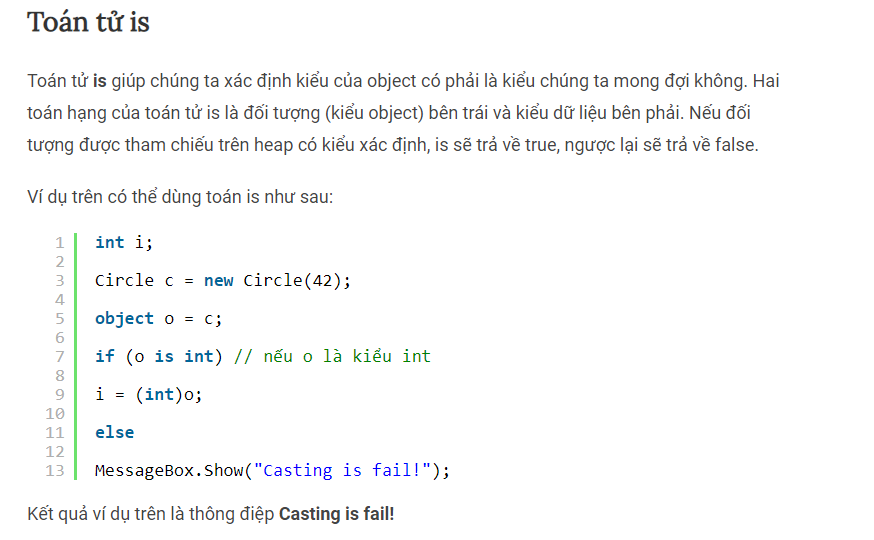


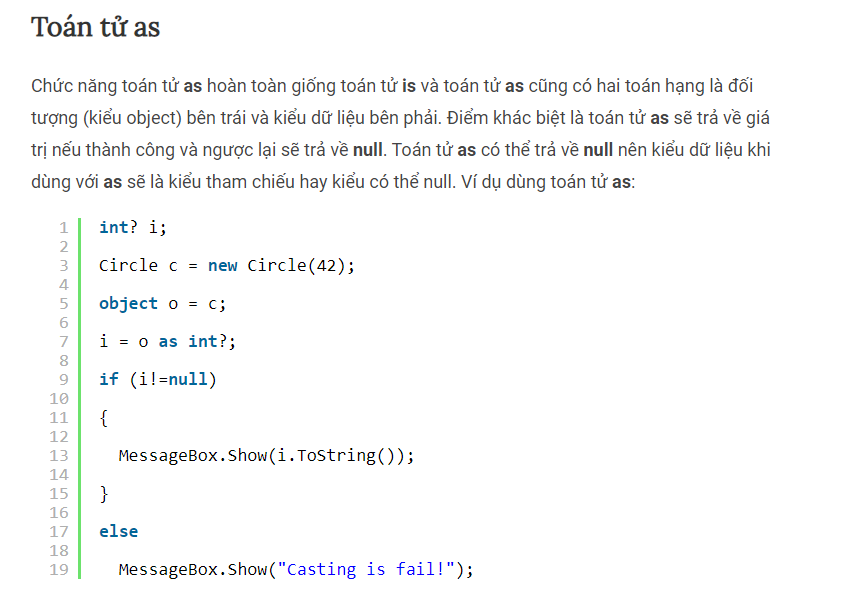


**- Database first**: là phương pháp chỉ nên dùng khi ta đã có sẵn CSDL (không phải tạo), EF Wizard sẽ tạo Model và Code cho ta.

- **Models first**: nên dùng khi ta bắt đầu thiết kế CSDL từ đầu (từ chưa có gì). Ta sẽ thiết kế mô hình CSDL (Model) EF sẽ tự tạo code cho ta, sau đó nhờ EF Wizard tạo CSDL.

- **Code first**: nên dùng khi đã có mô hình CSDL, ta sẽ chỉ viết code từ đó tạo Database.





Ref: bắt buộc phải khởi tạo giá trị trc khi vào hàm

Out: bắt buộc phải gán giá trị trc khi ra khỏi hàm

# localStorage

Data is shared between all tabs and windows from the same origin.

The data will not expire.

It will remain even after browser restart and survive OS reboot too.

# sessionStorage

The sessionStorage exists only within the current browser tab. Another tab with the same page will have a different session storage.

However it is shared between iframes in the same tab (assuming they come from the same origin).

The data survives page refresh, but not closing/opening the tab.

**dependency injection**