I. Entity Framework

\*Definition: Là một DB ORM (Database Object-Relational Mapper) trên nền tảng .NET, giúp người phát triển không phải tốn công sức xây dựng lớp truy xuất dữ liệu (data-access code).

\*Version: 6.1.1

\*Features:

-ORM: là một khái niệm về công nghệ cho phép làm việc trên đối tượng trên các CSDL quan hệ.

-Lazy loading: Trì hoãn load dữ liệu khi chưa cần thiết, dữ liệu chỉ được load khi có lời gọi tới lần đầu tiên (Giảm thời gian nạp dữ liệu ban đầu).

-Code first: là một cách tiếp cận mới trong các ứng dụng hướng CSDL, thay vì thiết kế cấu trúc CSDL trước thì Code first cung cấp các đặc tả để người lập trình có thể định nghĩa cấu trúc CSDL bằng các class và các logic, ràng buộc trên class, tận dụng được các tính năng kế thừa trên class, sau đó Entity Framework sẽ tự động ánh xạ xuống cấu trúc CSDL.

+Code first to new Database

+Migration: dịch chuyển phiên bản trên CSDL

+Datasbe Version (Upgrade/Downgrade/Rebase)

+Data Seeding

+Model cheking (Structure changed): Kiểm tra cấu trúc CSDL có bị thay đổi bởi các tác nhân ngoài hệ thống, đảm bảo hệ thống làm việc ổn định và dúng đắn.

+Custom Database Initalier: Chỉ định phương thức khởi tạo CSDL (tự động tạo CSDL nếu chưa có, tự động tạo bảng nếu chưa có, tự động tạo dữ liệu mẫu ban đầu,…)

-BootStrapper (override some method for specific business process) vs Event Hooker: Cho phép đăng ký các sự kiệ xảy ra trên Entity khi Entity Framework làm việc, sử dụng để thực hiện một vài hành động logic như: sau khi cập nhật thành công thì trường date\_modified sẽ có giá trị giờ hiện tại.

-Class proxy (override defined class)

-Transaction Manager (Rollback/commit): Đảm bảo tính đơn nguyên của giao dịch, hoặc là tất cả thao tác dữ liệu đều được thực thi hoặc là không.

-Entity State tracking and “Filter Update”

-Entity framework kết hợp mô hình 3 lớp

-Manual media cached (class HinhAnh)

-Entity vs DataFilter

-Database Context vs Singleton Database Instance Provider

-TPC (Table per Concreted class) pattern: tận dụng code logic

-OOP

\*References source:

<http://www.entityframeworktutorial.net/>

<http://msdn.microsoft.com/en-us/data/aa937723>

<https://github.com/tidyui/EFBootstrapper>

II. Sync Framework

\*Definition: Đồng bộ giữa các CSDL

\*Version: 2.1

\*Features:

-Đồng bộ dữ liệu giữa các CSDL

-Sử dụng kỹ thuật trigger trên từng Table, lưu thông tin về sự thay đổi trên CSDL trong các Table tracking mà Sync Framework tạo ra khi được cài đặt Scope

-Chỉ đồng bộ những dữ liệu mới dựa trên các tracking trên Scope đã có sẵn (tiết kiệm được thời gian và lượng dữ liệu phải xử lý)

-Sync Scope

+Define: là một định nghĩa về phiên đồng bộ trên một CSDL cụ thể, chứa tập hợp các thông tin về Table cần đồng bộ. Một CSDL có thể có nhiều Sync Scope.

+Set up scope to existed database: Chỉ định dãy tuyến tính các Table cần đồng bộ, sau đó gọi phương thức khởi tạo Scope, Sync Framework sẽ thực hiện nhiệm vụ còn lại.

+Remove scope from existed database: Chỉ định tên Scope muốn gõ bỏ, sau đó gọi phương thức gõ bỏ Scope, Sync Framework sẽ thực hiện nhiệm vụ còn lại.

+Fetching scope among databases: Đăng ký một Scope cho CSDL này (Scope ) từ CSDL khác (Scope provider)

+Sync Direction (Up/Down/Bidirectional link): chọn hướng đồng bộ, đáp ứng một số yêu cầu về bảo mật trên CSDL.

\*Challenge:

-Đụng độ dữ liệu:

+Đụng độ khóa chính: Xét 2 CSDL độc lập có cùng cấu trúc Table và dữ liệu, nếu khóa chính được thiết lập dạng Auto\_Increasement thì khi gọi Insert trên 2 CSDL, hệ quản trị CSDL địa phương ở cả 2 CSDL trên sẽ có khả năng tạo ra khóa chính trùng nhau (khả năng rất cao). Nên khi đồng bộ dữ liệu sẽ bị đụng độ khóa chính, 1 trong 2 dữ liệu mới Insert đó sẽ bị loại bỏ.

=> Sử dụng khóa chính Guid: Khóa chính Guid là khóa chính dạng “tự nhiên” (Natural), được tạo ra dựa trên các giá trị định danh (địa chỉ MAC của card mạng) và ngẫu nhiên (thời gian hiện tại trên Hệ quản trị CSDL), Hệ quản trị sẽ đảm bảo mỗi Guid được cấp phát sẽ là duy nhất trên toàn cầu (mặc dù khả năng trùng là có thể xảy ra trên lý thuyết nhưng có thể chấp nhận được).

- Database Schema “dbo” cho User (SQL Server): nếu sai khác giữa các Table sẽ gây lỗi do thiếu quyền chạy trigger trên các Table tracking

=>Trước khi sử dụng User đó để sử dụng cho các lệnh thiết lập đồng bộ, nhất thiết phải chỉ định Default Database Schema là “dbo”.

-Ràng buộc thứ tự tiến trình trên Table: thứ tự các Table khi đồng bộ là rất quan trọng, vì nếu Table B có chứa khóa ngoại tham chiếu đến Table A thì Table A phải được đồng bộ trước Table B

=> Xây dựng thứ tự tiến trình trong đó luôn đảm bảo Table dạng B luôn nằm sau Table dạng A

\*References source:

<http://msdn.microsoft.com/en-us/library/bb902854(v=sql.110).aspx>

III. DevExpress

\*Definition:

\*Version:

\*Features: