**Unit OF Work**

**I. Định nghĩa**

Unit Of Work là một khái niệm liên quan đến hiệu quả của việc triển khai Repository Pattern.

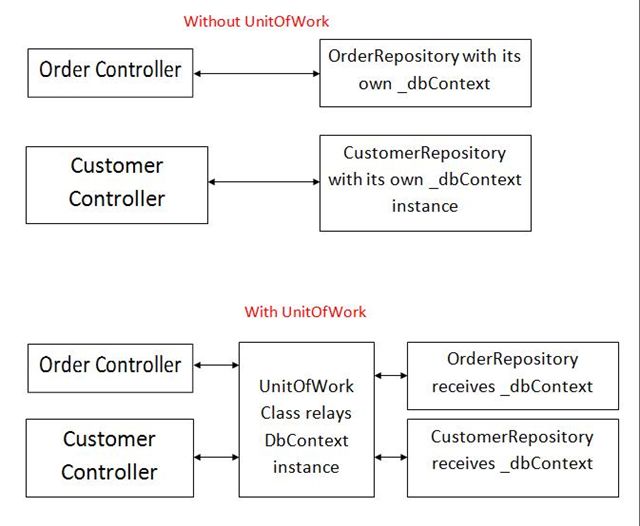
**II. Lý do sử dụng**

Unit Of Work được sử dụng để đảm bảo nhiều hành động như insert, update, delete...được thực thi trong cùng một transaction thống nhất. Nói đơn giản hơn, nghĩa là khi một hành động của người dùng tác động vào hệ thống, tất cả các hành động như insert, update, delete...phải thực hiện xong thì mới gọi là một transaction thành công. Gói tất cả các hành động đơn lẻ vào một transaction để đảm bảo tính toàn vẹn dữ liệu.



Đoạn code trên nhìn có vẻ đúng. Nhưng có một vấn đề ở đây khi chúng ta thêm một repository cho thực thể khác ví dụ như Order. Trong trường hợp này cả hai repository sẽ cùng phải khởi tạo, và sử dụng trên đối tượng DbContext của riêng nó. Điều này sẽ có rủi ro trong tương lai khi một trong 2 hàm SaveChange() của một trong 2 repository bị lỗi và cái kia thành công nên dữ liệu trong cơ sở dữ liệu sẽ bị sai. Đây là lúc mà Unit Of Work cần dùng đến.

Để ngăn chặn điều này, chúng ta sẽ thêm một tầng mới hoặc một tầng trung gian giữa Controller và Customer Repository. Lớp này sẽ tập trung hóa việc lưu trữ cho tất cả các Repository để nhận đối tượng thể hiện của DbContext. Điều này đảm bảo rằng mỗi một transaction sẽ dùng chung 1 thể hiện của DbContext cho tất cả các Repository liên quan. Hoặc là thành công tất cả hoặc thất bại cũng thất bại tất cả. Trong ví dụ trên khi thêm dữ liệu cho Order và Customer trong một transaction duy nhất, cả hai sẽ sử dụng cùng 1 thể hiện của DbContext. Trường hợp nếu không dùng và có dùng Unit Of Work có thể được trình bày như sau:



Trong trình bày hình ảnh trên, trong khi một hành động xảy ra, cả Customer và Order đều sử dụng chung một đối tượng của lớp DbContext. Điều này sẽ đảm bảo ngay cả khi một trong hai bị lỗi, cái kia sẽ không được lưu lại, vì thế database sẽ  toàn vẹn. Khi SaveChange() được thực thi, nó sẽ thành công khi cả 2 hành động trên 2 Repository được thực hiện xong.

Chúng ta hãy cùng triển khai khái niệm này cho ví dụ trên. Chúng ta sẽ thêm mới 1 class tên là UnitOfWork và class này sẽ nhận thể hiện của lớp DbContext. Class này cũng tạo ra thể hiện cho các Repository cần thiết hay nói cách khác thì các thể hiện của Repository Order và Customer đều được sử dụng chung 1 đối tượng DbContext:



