Hibernate Life Cycle

* Transient : bước đầu tiên trong hành trình vòng đời hibernate. Khi lần đầu tạo 1 java object( thông qua new keyword ) , obj này được coi là trong trạng thái transient.

Trong trạng thái này Hibernate không thể lần theo transient object và lưu trữ nó vào trong db (hoặc update dựa trên giá trị của nó)

* Persistent : Khi 1 obj ở trong persistent state , Hibernate hoàn toàn nhận biết được nó và có thể giữ db đồng bộ với giá trị của nó

Làm sao để đi từ transient đến persistent : Có 2 cách chính để nhảy từ transient obj sang persistent obj :

1. Load obj từ db thông qua Hibernate APIs
2. Session.get()
3. Session.createCriteria()
4. Save obj từ db thông qua Hibernate APIs
5. Session.save();
6. Session.saveOrUpdate();

Khi đưa 1 entity object vào session.save, sau đó mọi thay đôir tới obj đó sẽ vào lệnh update trong db cho tới khi gọi commit và close() nó và chỉ thực hiện update thay đổi cuối cùng

Get : hàng au

Load : hàng fake

* Detached : đối tượng đã từng trong persistent state nhưng giờ không còn giữ quan hệ với session

//transient to persistent

* Save : trả về object id sau khi gọi , nó thực hiện lệnh insert ngay lập tức không cần biết có trong transaction hay không
* Persist : kiểu trả về là void , cũng thực hiện lệnh insert nhưng nó đảm bảo rằng lệnh insert sẽ không được chạy nếu gọi persist ở ngoài transaction.
* SaveOrUpdate : có thể hoặc insert hoặc update dựa trên sự tồn tại của bản ghi. Nó linh hoạt hơn save() nhưng nó gồm thêm 1 bước xử lý để tìm ra bản ghi có tồn tại hay không. Nếu bản ghi tồn tại sẽ thực hiện UPDATE , nếu không sẽ INSERT.
* Update : chuyển obj từ detached state về persistent state, dùng để update db row đã tồn tại. Nếu không tồn tại sẽ trả về exception. Update không thể sử dụng với obj giống nhau tồn tại trong cùng session.
* Merge : cũng giống update có thể chuyển từ detached về persistent, dùng để update db row đã tồn tại, nếu không tồn tại sẽ cố gắng tạo 1 db row mới . Merge có thể sử dụng với obj giống nhau trong cùng session. Merge có thể lưu thay đổi ở bất cứ lúc nào mà không quan tâm đến trạng thái của session.

//persistent to detached

* Evict() : xóa từng object một ra khỏi session.
* Clear() : xóa tất cả object trong session, gọi trong transaction . Gọi clear() tức là xóa tất cả obj liên kết tới session.
* Close() : đóng session, ngắt liên kết tới obj liên quan, kết thúc và thoát khỏi JDBC connection và dọn dẹp.
* Session.delete() : xóa 1 detached obj , xóa khỏi db . Nếu đối tượng không tồn tại trong db sẽ trả về exception