**FetcherType**

Hibernate cung cấp các Fetcher hỗ trợ việc lấy thông tin từ cơ sở dữ liệu. Cài đặt fetcher bằng các đánh dấu java annotation hoặc bằng file cấu hình xml.

Quan hệ cha con trong cơ sở dữ liệu có thể hiểu là quan hệ 1 – n (một cha có nhiều con).

Ta có 2 tuỳ chọn FetcherType ở đây:  
*FercherType.LAZY:*  Chế độ LAZY được đặt mặc định khi cài đặt. Khi một session gọi đến một đối tượng thì chỉ đối tượng đó được gọi ra trong bộ nhớ, khi Session kết thúc, không có đối tượng liên quan nào đc gọi thêm.

*FetcherType.EAGER:* Chế độ EAGER được cài đặt khi ta cần gọi lên đối tượng nào đó, hệ thống sẽ gọi luôn các quan hệ và các đối tượng liên quan. Hay nói các khác, khi Session đóng lại ta vẫn có thể thực hiện các công việc với các đối tượng được gọi lên mà không cần mở thêm Session hay để một Session mở quá lâu.

**CascadeType**

Tương tự như FetchType, Cascade cũng hỗ trợ việc truy từ cơ sở dữ liệu. Ta có thể cài đặt Fetcher bằng đánh dấu java annotation hoặc bằng file cấu hình xml.

Cascade được đặt mặc định là NONE. Tức là các thay đổi ở đối tượng cha sẽ tự không ảnh hưởng đối với các đối tượng con.

*CascadeType.ALL*: Các thay đổi ở đối tượng cha sẽ tự động được áp dụng đối với đối tượng con, ví dụ: thêm một bản ghi vào cha, các bản ghi ở con cũng sẽ được thêm vào. Ngoài ra còn có các tuỳ chọn riêng rẽ DELETE, SAVE, PERSIS,… cũng tương tự như ALL

*CascadeType.DETACH*: Ngắt quan hệ giữa 2 đối tượng cha-con trong cơ sở dữ liệu, hay nói cách khác, nó sẽ vô hiệu hoá khoá ngoài của 2 bảng. Như vậy sau khi thực hiện một thay đổi liên quan đến quan hệ cha-con mà ta không cần thiết giữ quan hệ đó cho các thao tác sau, ta sẽ dùng DETACH.

**CACHE**

First Level Cache

* + First level Cache chỉ liên kết với 1 Session Object duy nhất, các Session Object khác không thấy, không tác động vào được.
  + Phạm vi của 1 First Level Cache là Session, khi Session đóng thì cũng đóng theo.
  + First Level Cache là mặc định ta không thể hủy nó đi.
  + Khi ta thực hiện query dữ liệu lần đầu, dữ liệu được lấy ra từ DB và lưu vào First Level Cache được liên kết với Session.
  + Nếu query cùng 1 đối tượng và 1 Session như trên thì dữ liệu được trả về sẽ được lấy từ First Level Cache được liên kết với Session này.
  + Xóa dữ liệu trong Firsrt Level Cache của 1 Session sử dụng method evict().
  + Xóa dữ liệu trọng tất cả các Cache của các Session sử dụng method clear().

Second Level Cache

* + Bất cứ khi nào Session cố gắng load 1 entity thì nơi đầu tiên mà nó tìm kiếm chính là First Level Cache.
  + Nếu Entity có trong First Level Cache thì Entity sẽ được trả về.
  + Nếu trong First Level Cache không có thì Second Level Cache sẽ được tìm kiếm.
  + Nếu Entity có trong Second Level Cache thì entity sẽ được trả về. Nhưng trước khi được trả về Entity sẽ được đặt trong First Level Cache để tiện cho những lần query tiếp theo không phải tìm trong Second Level Cache.
  + Nếu Entity đều không có trong First và Second Level Cache thì Entity sẽ được lấy ra từ DB và được ghi vào First Second Level Cache trước khi được trả về.

**HQL – Native SQL – Criteria**

*HQL*

* Là một ngôn ngữ truy vấn hướng đối tượng, tương tự như SQL
* HQL sẽ tương tác với các persistent objects và các properties chúng thay vì tương tác với các tables và columns giống như SQL
* HQL sẽ được chuyển đổi thành SQL để tương tác với database bằng Hibernate.

*Native SQL*

* Native SQL thao tác với CSDL quan hệ, k có tính hướng đối tượng.
* SQL điều khiển dữ liệu được lưu trữ trong bảng và sử đổi các hàng các cột của nó

trong khi HQL quan tâm đến các đối tượng và thuộc tính của nó.

* SQl quan tâm đến quan hệ tồn tại giữa 2 bảng trong khi HQL quan tâm đến mối quan hệ giữa các đối tượng.

*Criteria*

* Đối tượng thao tác: Persistence, tính hướng đối tượng tương tự HQL.
* Sử dụng các API được cung cấp sẵn.
* Tốc độ nhanh đối với các truy vấn đơn giản.
* Hạn chế: các câu lệnh điều kiện phức tạp khó viết chuẩn hơn HQL.