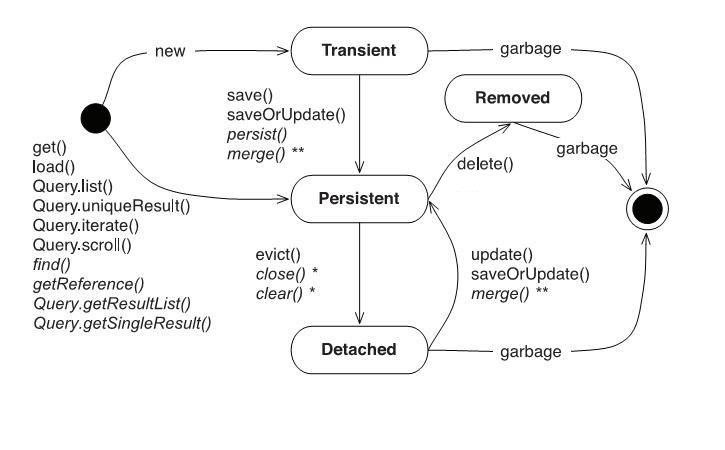
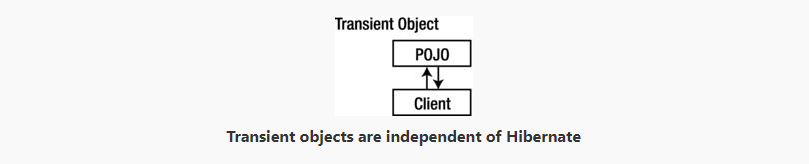
**HIBERNATE LIFECYCLE**

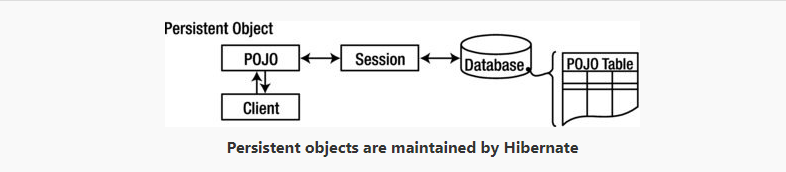


1. **Transient**: khi một đối tượng mới được tạo, không có một mối quan hệ nào với database, có nghĩa là không có quan hệ với một session object, khi khởi tạo một entity thì nó sẽ vào trạng thái transient



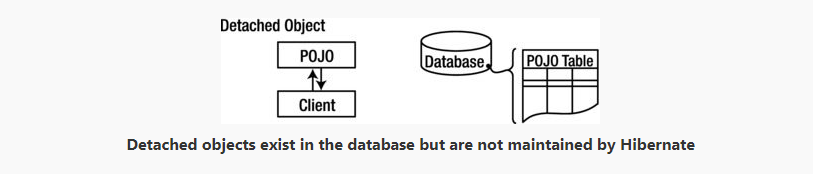
1. **Persisstent**: có quan hệ với db, kết nối với một session object duy nhất. khi kết nối với db thì dữ liệu được đồng bộ với db,

Trở thành một persistent thì có thể dùng hai cách là load object từ db thông qua hibernate api hoặc save object vào db thông qua API



1. **Detached**:

Khi đóng session, chuyển từ persistent state sang detached, nó không đính kèm với một session nào nhưng vẫn có thể sửa đổi và gắn lại với một session mới. Detached object có các mục tương ứng trong database, đó là persistent và nó không kết nối với Session object. Các object đó có đồng bộ dữ liệu với database khi session đóng. Sau khi đóng thì nếu thay đổi dữ liệu thì nó sẽ không tự động update về db, khi kết nối lại thì dữ liệu sẽ được update. Detached object có thể trở về persistation mà không cần dùng data tranfer object. Detached object có thể sửa đổi ngoài và sau đó có thể gắn lại vào một transaction mới. Có thể dùng session.merge(), session.update, session.saveOrUpdate().



**4 Remove**

Khi Obj bị gọi trong session.delete(Obj o) thì obj o sẽ bị xóa khỏi Db, biết mất khỏi DB, đối tượng java có thể tồn tại nhưng sẽ không có trong hibernate

(\*\*) chuyển từ detached sang persistent sẽ nhanh hơn chuyển từ transient sang persistent vì

* Ở dạng detached thì entity đã có một kết nối với DB nhưng bị ngắt và chỉ cần kết nối lại còn từ transient phải tạo một kết nối mới
* Ở dạng detached thì deatched Obj đã có các thuộc tính tương ứng với các cột trong DB, sẽ không cần phải kết nối mới

(\*\*\*) Thường thì các Entity sẽ ở trong trạng thái persistent để thực hiện các hành động với DB

(\*\*\*\*) Khác nhau giữa getCurrentSession() và openSession

* getCurrentSession():dùng session cũ nếu nó đang tồn tại và tạo cái mới nếu nó không tồn tại. Hibernate tự động đẩy dữ liệu session.flush() và tự động đóng session.close() khi thực hiện xong
* openSection():luôn tạo mới session

**Chi tiết:**

1. Transient 🡪 Persistent

* Session.save() : Trả về Entity Id sau khi gọi, không liên quan tới khóa trong bảng của DB, nó thực hiện lệnh insert ngay lập tức không cần biết nó có trong transaction hay không. Nếu Obj có id trùng với một Obj trong DB thì báo exeption, nếu không thì thêm vào bảng
* SaveOrUpdate(): Kiểu void, nó thực hiện save khi Obj chưa tồn tại, thực hiện update khi đã có Obj trùng khóa với Obj trong DB
* Session.persist(): kiểu trả về là void, cũng thực hiện lệnh insert nhưng nó đảm bảo lệnh insert sẽ không được chạy nếu goi persist **ở ngoài transaction**
* Session.merge(): trả về một Obj, Không đưa đối tượng vào hibernate quản lý mà tạo ra một bản sao và quản lý đối tượng đó, Nó thực hiện save khi Obj chưa tồn tại hoặc thực hiện update khi đã có Obj trùng khóa với Obj trong DB

1. Persistent 🡪 Detached

* Session.evict(Obj o): kiểu void, ngắn kết nối một persistent o với DB
* Session.clear():kiểu void, ngắt tất cả kết nối tất cả của các Entity với Db
* Session.close(): đóng session, tất cả các persistent obj sẽ bị ngắt kết nối với DB

1. Detached 🡪 Persistent

* Session.update(): kiểu void, thực hiện update một Obj đã tồn tại, trả về exception khi không tồn tại
* Session.saveOrUpdate(): kiểu void
* Session.merge(): trả về một Obj

1. Persistent 🡪 Remove

* Session.delete(Obj o): kiểu void, Xóa Obj o khỏi DB và session

1. New 🡪Persistent

* Session.get(String s, “<key>”): Trả về một Obj thực có kiểu s có khóa là <key>, kết nối với DB, trở thành một persistent Obj. Nếu không tồn tại Obj cần tìm thì return null, làm việc trực tiếp với DB, sử dụng query để tìm Obj cần tìm
* Session.load(String s, “<key>”) : Trả về một proxy (Obj fake tham chiếu tới Obj cần tìm), có khóa là <key>, cũng tạo kết nối với DB, trở thành một persistent Obj. Nếu Obj o không tồn tại thì return exception, không làm việc trực tiếp với DB. Khi session.load() được gọi thì hibernate tạo một Obj fake giống Obj có id = key

(\*)Dùng get() khi cần load một Obj, dùng load() khi tham chiếu đến đối tượng mà không cần query obj đó trong db