|  |  |
| --- | --- |
| ORM | * Object relational mapping là kỹ thuật để liên kết giữa đối tượng trong java và đối tượng trong csdl * Cho phép truy xuất dễ dàng đến dữ liệu thông qua các đối tượng trong java * Khi dùng ORM ltv sẽ k cần quá quan tâm đến DB đang sử dụng * Ưu điểm:   + Quản lý dữ liệu tập trung vào code  + Các thao tác với dữ liệu sẽ được thực hiện tự động  + Tránh được các lỗi cú pháp SQL  + Hỗ trợ transaction  + Có thể cache dữ liệu để tăng tốc dữ liệu   * Nhược điểm:   + Đối vs dự án lớn thì hiệu năng truy xuất dữ liệu thường hạn chế  + Ltv sẽ dễ bị rơi vào bẫy truy xuất dữ liệu  + Đối với các thao tác phức tạp (nhiều bảng) thì cần phải viết câu lệnh SQL thuần để xử lý |
| JPA | * Java Persistence API: cung cấp interface để duy trì, đọc, quản lý dữ liệu từ đối tượng Java đến các quan hệ trong DB * Cung cấp 1 mô hình POJO (Plain Old Java Object) cho phép ánh xạ các table/relationship giữa các table trong database sang các class/mqh giữa các class |
| Entity | * Là đối tượng đại diện cho dữ liệu trong ứng dụng * Thường là POJO (Plain Old Java Object) * Sẽ đc ánh xạ vs 1 bảng trong database |
| Persisence Context | * Là tập các thể hiện của entity được quản lý, tồn tại trong DB |
| Entity Manager | * Là 1 interface khai báo các phương thức để tương tác với persistence context * Tạo/xóa các thể hiện của entity * Thực hiện các câu lệnh truy vấn lên entity * Tìm kiếm entity dựa trên primary key |
| Query động và tĩnh | * Query động:   Khai báo lúc gọi method createQuery() của entityManager  Không có lưu ý gì   * Query tĩnh:   Khai báo ở class entity bằng @NameQuery  Cần truyền tên câu query tĩnh đã khai báo ở trên vào method createNamedQuery để thực thi   * Khai báo và truyền tham số   Dùng param để định nghĩa tham số và truyền vào bằng method setParameter(“param”, value)  Sử dụng ?x trong đó x là vị trí tham số cần truyền vào và truyền bằng cách dùng setParameter(x, value) |