## Chương 1: Lý thuyết csdlpt

# Tổng quát:

* Giới thiệu về cơ sở dữ liệu phân tán: Giới thiệu các khái niệm về cơ sở dữ liệu phân tán, so sánh cơ sở dữ liệu phân tán với cơ sở dữ liệu tập trung (tính trùng lắp, tính sẵn sàng,…) từ đó rút ra những lý do để phát triển một hệ thống dựa trên cơ sở dữ liệu phân tán, cuối cùng trình bày khái quát về hệ quản trị cơ sở dữ liệu phân tán.
* Kiến trúc về cơ sở dữ liệu phân tán
* Kiến trúc của hệ quản trị cơ sở dữ liệu phân tán

# Giới thiệu về csdlpt:

Chúng ta có thể định nghĩa sơ nét về csdlpt như sau: 1 csdlpt là 1 tập hợp dữ liệu của 1 hệ thống nhưng được phân bố trên nhiều site của 1 mạng máy tính. Định nghĩa này nhấn mạnh 2 khía cạnh quan trọng của csdlpt là:

1. **Sự phân tán:** dữ liệu không lưu trữ trên cùng 1 địa điểm vì thế chúng ta có thể phân biệt nó với cơ sở dữ liệu tập trung
2. **Mối tương quan luận lý:** Các dữ liệu có 1 số thuộc tính ràng buộc với nhau từ các cơ sở dữ liệu cục bộ mà được lưu trữ tại các địa điểm khác nhau trên mạng.

Có 2 loại ràng buộc:

* + Ràng buộc cứng: định nghĩa sẵn trên db
    - PK
    - UK
    - FK
    - Range
    - NOT NULL
  + Ràng buộc mềm: Lập trình trên hệ thống (Không thể đưa vào trực tiếp trên db) thông thường sẽ bắt trên phần mềm (chương trình)
    - Vẫn có thể viết ràng buộc mềm trên csdl -> trigger
      * -> làm chậm hệ thống khi chạy
      * -> không cản được lỗi cơ bản ở mức client -> chỉ phát huy tác dụng khi đưa vào db
* Khi lập trình ngoài ràng buộc cứng cần phải bắt trigger trên db và bắt cả lỗi trên lập trình.

?: Tại sao phải bắt lỗi trên hệ thống thay vì csdl? Vì khi bắt lỗi trên database người dùng nhập dữ liệu vào thì dữ liệu phải đưa xuống database để kiểm tra sau đó trả về kết quả → Tốn tg.

* Các ràng buộc cứng + mềm -> Ràng buộc toàn vẹn của db tập trung đều phải đưa được vào hệ thống phân tán (tránh sự mất mát dữ liệu)

?: Mỗi chi nhánh chỉ lưu trữ csdl cục bộ có đủ chưa? Cơ sở dữ liệu phân tán có phải là 1 tập các csdl cục bộ?

* Về mặt kỹ thuật, chúng ta thấy cần có các ứng dụng mà truy xuất dữ liệu đang đặt ở nhiều nhánh. Các ứng dụng này được gọi là ứng dụng toàn cục hay ứng dụng phân tán.
* Một ứng dụng toàn cục thông thường trong ví dụ ngân hàng là việc chuyển tiền từ 1 tk này đến tk khác. Ứng dụng này yêu cầu cập nhật csdl ở cả 2 nhánh.
* Hơn nữa ứng dụng toàn cục giúp cho người dùng không phân biệt được dữ liệu đó là cục bộ hay toàn cục. Đó là tính trong suốt dữ liệu trong csdlpt. Và đương nhiên khi ứng dụng toàn cục truy cập dữ liệu cục bộ sẽ nhanh hơn ứng dụng từ xa điều này nói lên sự nhân bản dữ liệu ở các nơi cũng làm tăng tốc độ xử lý chương trình.

?: Trên 1 quan hệ nhân bản giả sử có Mã vật tư và muốn tìm số lượng tồn của mã vật tư đó thì tìm trên site hiện tại nếu không có thì tìm ở các site tiếp theo nếu không có thì báo lỗi còn nếu có thì in ra số lượng tồn là đúng hay sai?

**SAI**. Khi đã nhân bản thì ở mọi nơi đều có như nhau nên nếu site hiện tại đã không có thì các site khác cũng không có.

# ĐỊnh nghĩa về csdlpt