UIT VNUHCM University of Information Technology

CHƯƠNG 5:
MỘT SỐ MÔ HÌNH CSDL TIÊN TIẾN:
CSDL PHÂN TÁN

Khoa Khoa học và kỹ thuật thông tin
Bộ môn Thiết bị di động và Công nghệ Web

Nội dung

1. Khái niệm về CSDL phân tán.
2. Các đặc điểm của Cơ sở dữ liệu phân tán.
3. Các kỹ thuật phân mảnh.
4. Thiết kế CSDL phân tán.

Khái niệm

Khái niệm

- **Khái niệm 1:**

  CSDL phân tán là tập dữ liệu mà về mặt logic chúng thuộc cùng 1 hệ thống nhưng về mặt vật lý được trải ra nhiều nơi trong 1 mạng máy tính.

- **Khái niệm 2:**

  CSDL phân tán là tập CSDL phân bố trên các máy tính khác nhau cùng một mạng. Mỗi máy có khả năng xử lý tự trị, có các ứng dụng local, tham gia vào ứng dụng global bằng hệ thống mạng.

Ví dụ minh hoạ

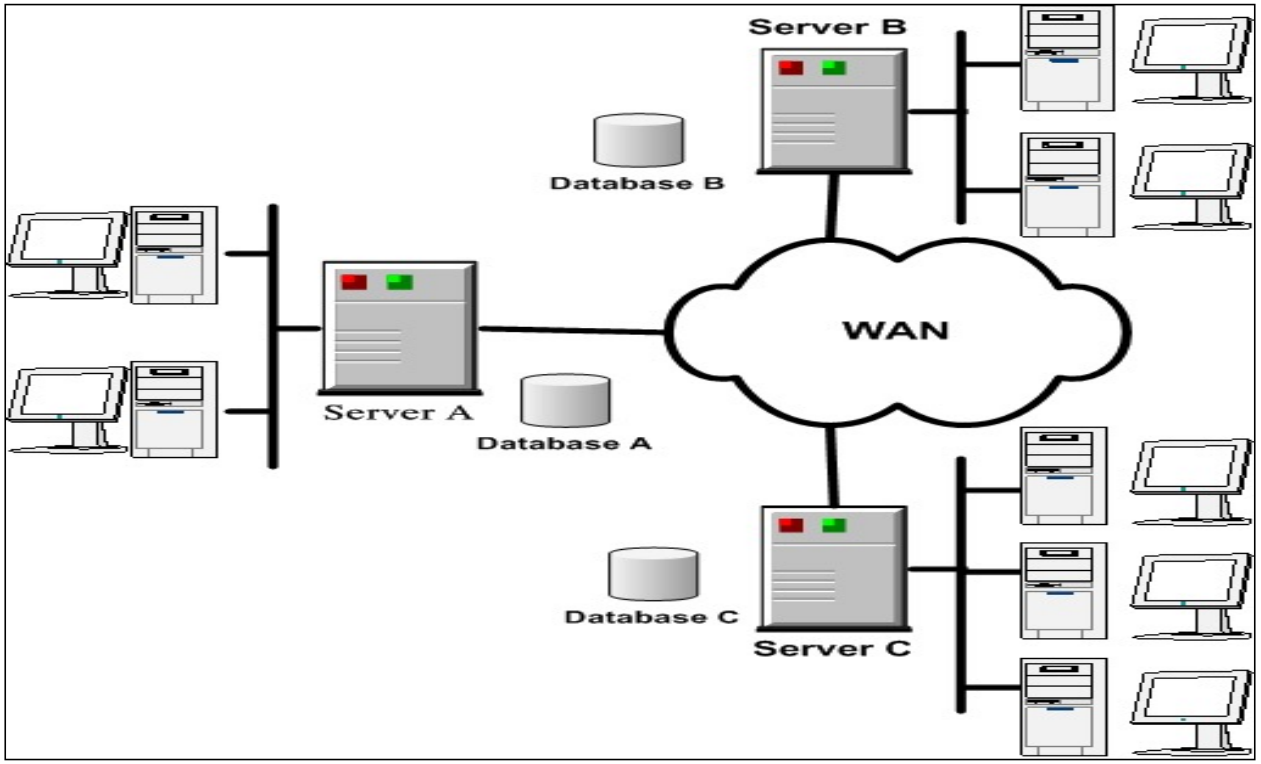
Ví dụ: mạng máy tính của ngân hàng ACB có 3 chi nhánh ở Hà Nội, Đà Nẵng, Sài Gòn. Mỗi chi nhánh chứa các tài khoản người dùng.

KHACHHANG(mskh, tenkh)

GIAODICH(msgd, mskh, sotien, guirut)

➔Khách hàng Peter thực hiện giao dịch: gởi tiền vào tài khoản ở chi nhánh C, rút tiền ở C và gởi tiền vào tài khoản ở chi nhánh A. Đây là hệ CSDL phân tán vì dữ liệu nằm ở hai nơi và có quan hệ mật thiết - một khách hàng mở tài khoản ở hai chi nhánh phải có cùng MSKH.

Minh hoạ


Server B
Database B
WAN
Server A
Database A
Database C
Server C

Đặc điểm CSDL phân tán

Đặc điểm

- Độc lập dữ liệu, tự trị.
- Dư thừa dữ liệu.
- Cấu trúc vật lý phức tạp.
- Tính toàn vẹn, toàn cục.
- Điều khiển đồng thời.
- Tính bảo mật.

CSDL tập trung vs. CSDL phân tán

- Không độc lập dữ liệu cao.
- Tự trị duy nhất.
- Rủi ro cao.

- Độc lập dữ liệu cao.
- Tính tự trị cao.
- Cấu trúc vật lý, quản trị phức tạp.
- Chi phí lớn.

Hệ quản trị CSDL phân tán

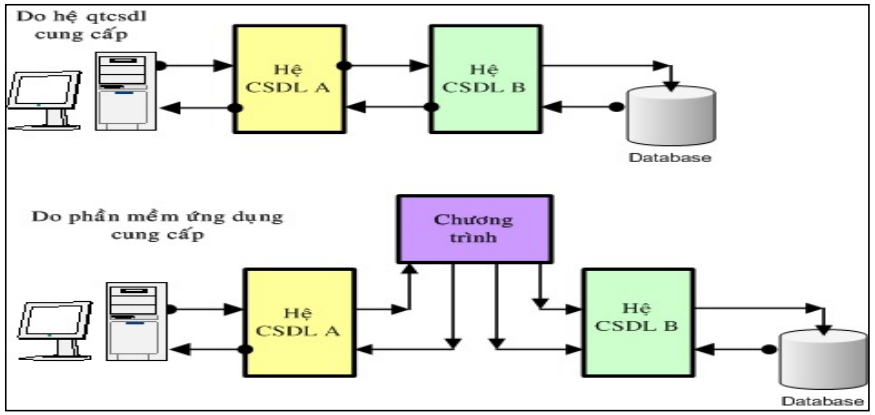
- Truy xuất dữ liệu từ xa (remote access)
- Hỗ trợ mức trong suốt (transparency) cho csdl phân tán.
- Hỗ trợ quản trị, giám sát csdl.
- Hỗ trợ phục hồi dữ liệu.
- Hỗ trợ môi trường không đồng nhất.

VD: Hệ quản trị CSDL phân tán

– Truy suất dữ liệu từ xa (remote access)

+ Do hệ quản trị CSDL cung cấp: không phong phú, chưa đáp ứng được nhu cầu đa dạng.

+ Do phần mềm ứng dụng cung cấp.

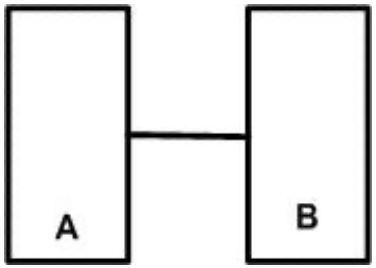


VD: Hệ quản trị CSDL phân tán

– **Hỗ trợ mức trong suốt cho csdl phân tán**

+ *HOADON(mshd, tt)*
+ *CTHD(mshd,msmh, sl)*
+ *MATHANG(msmh, ten, dongia)*

Tại A không có dữ liệu Mathang. Sự hỗ trợ trong suốt làm cho A có cảm giác Mathang vẫn có tại A.



VD: Hệ quản trị CSDL phân tán

- **Hỗ trợ quản trị , giám sát (audit, monitor) csdl**
  - + Đứng tại A hay B đều có thể thêm, xóa, sửa, xem trên các dữ liệu còn lại.
- **Hỗ trợ phục hồi (recover) dữ liệu**
  - + Khi có giao tác phân tán không hoàn thành, hệ quản trị phải hỗ trợ phục hồi csdl.

VD: Hệ quản trị CSDL phân tán

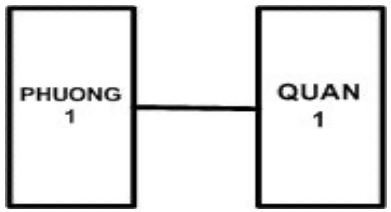
- NHANKHAU(msnk, tennk)

01	A
02	B

- NHANKHAU(msnk, tennk, phuong)

01	A	1
02	B	1
03	C	2

Khi B cập nhật (02,'B',2) và truyền cập nhật tới Quận , nếu có sự cố xảy ra thi phải khôi phục đồng thời tại Phường và Quận, nghĩa là tại Phường 1 có giá trị (02,'B').



VD: Hệ quản trị CSDL phân tán

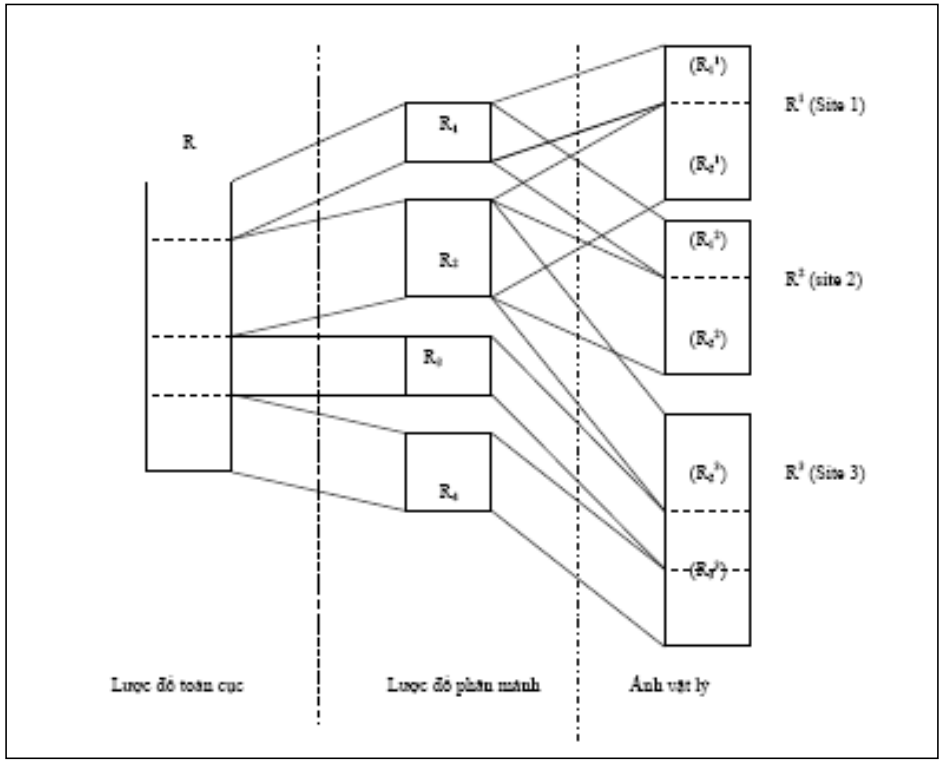
– Hỗ trợ môi trường không đồng nhất (in-homogeneous)

+ Các server có thể khác biệt phần cứng, HDH, hệ quản trị csdl. Tuy nhiên khác biệt về hệ quản trị csdl (khác về xử lý, lưu trữ, dữ liệu) là khó khăn lớn.
+ Một hệ phân tán hình thành từ các hệ đã tồn tại trước khó đồng nhất.
+ Một hệ phân tán hình thành từ khảo sát, phân tích, thiết kế từ đầu dễ đồng nhất.

Kiến trúc CSDL phân tán

- Mỗi quan hệ toàn cục có thể được chia thành các thành phần không trùng nhau được gọi là các phân mảnh.
- Có nhiều cách để phân mảnh mà chúng ta sẽ bàn đến sau.
- Ánh xạ từ các quan hệ toàn cục đến các phân mảnh được định nghĩa trong lược đồ phân mảnh.
- Phép ánh xạ này là một-nhiều nghĩa là có một số phân mảnh tương ứng với một quan hệ toàn cục nhưng chỉ có một quan hệ toàn cục ứng với một phân mảnh.
- Các phân mảnh được chỉ định bởi tên quan hệ toàn cục với một chỉ mục (chỉ mục phân mảnh) ví dụ Ri chỉ phân mảnh thứ i của quan hệ toàn cục R.

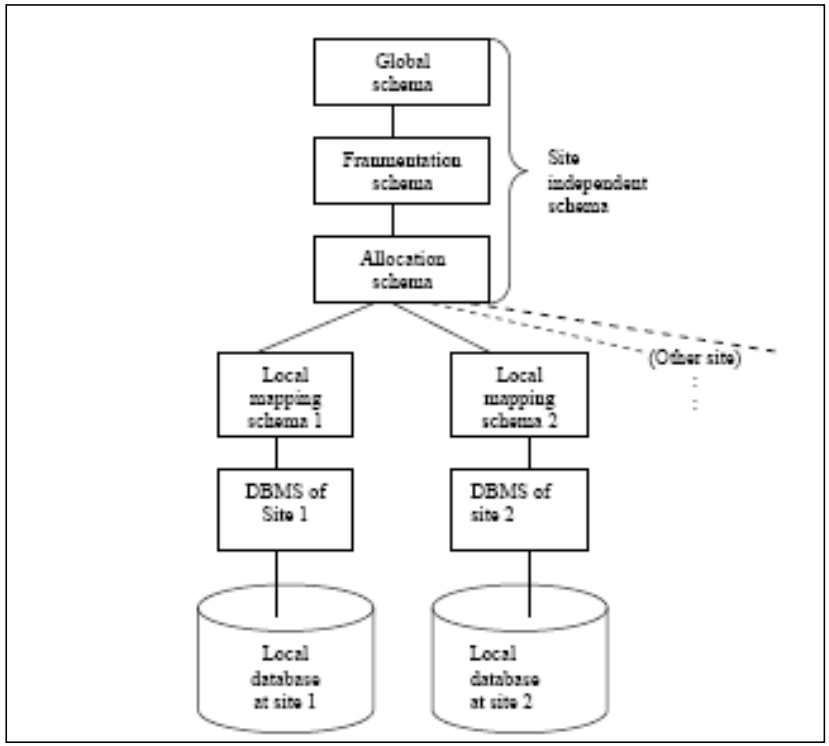
Kiến trúc phân tán



Kiến trúc phân tán

- Global schema: Là các lược đồ quan hệ toàn cục.
- Fragmentation: là các lược đồ quan hệ đã phân mảnh.
- Allocation schema: gồm các lược đồ quan hệ đã phân mảnh gắn liền với vị trí vật lý tương ứng.

- Sau công đoạn thiết kế sẽ là phần cài đặt trên các hệ QTSDL tại các vị trí vật lý cụ thể.



CÁC KỸ THUẬT PHÂN MẢNH

NHÂN BẢN (REPLICATION)

- Một quan hệ toàn cục R(A_1, A_2, . . ., A_n), các quan hệ Ri được phân bố giống hoàn toàn về cấu trúc cũng như tất cả dữ liệu so với R tạo ra hiện tượng nhân bản.
- Cho quan hệ toàn cục SV(*MSSV, TENSV, NAMS, NOIS, GT, DIACHI, EMAIL, MSK*) Nếu ta có 2 quan hệ SV1, SV2 có cùng cấu trúc và số dòng như SV nhưng khác site, thì SV1, SV2 được gọi là nhân bản của SV.

Phân mảnh ngang (Horizontal fragmentation)

- Một quan hệ toàn cục $R(A_1, A_2, \ldots, A_n)$ chia thành tập các quan hệ con Ri dựa vào các thuộc tính trên R. Các R_i được gọi là phân mảnh ngang của R.
- Kí hiệu: $R_i = \partial$ dk (R).
- Ví dụ: cho quan hệ toàn cục.

  SV(MSSV, TENSV, NAMS, NOIS, GT, DIACHI, EMAIL, MSK)
- Ta tạo ra 02 quan hệ SV3, SV4 thỏa điều kiện sau:

  + *SV3 = ∂ MSK='CNTT" (SV)*

  + *SV4 = ∂ MSK='DTVT" (SV)*
- Người ta phân loại phân mảnh ngang thành: PMN nguyên thủy và PMN dẫn xuất.

Phân mảnh ngang (Horizontal fragmentation)

Phân mảnh ngang nguyên thủy

- PMN nguyên thủy là PMN chỉ dựa trên một quan hệ.
- Ví dụ: cho quan hệ toàn cục *SV(MSSV, TENSV, NAMS, NOIS, GT, DIACHI, EMAIL, MSK)*
- Ta tạo ra 02 quan hệ SV3' SV4'thỏa điều kiện sau, chúng tạo thành 2 phân mảnh ngang của SV.

  SV3'= ∂ GT='Nu' (SV)

  SV4'= ∂ GT="Nam' (SV)

Phân mảnh ngang dẫn xuất

- PMN dẫn xuất là PMN một quan hệ nhưng cần dựa vào quan hệ khác để phân mảnh.
- Ví dụ

SV5 = ∏ [∂ tenkhoa='Cong Nghe Thong Tin " (SV►◄KHOA) + Đk phép kết]

SV6 = ∏ [∂ tenkhoa='Dien Tu Vien Thong" (SV►◄KHOA) + Đk phép kết]

Một cách thể hiện khác của SV5, SV6

SV5 = ∏ [(SV3►◄KHOA)]

SV6 = ∏ [(SV4►◄KHOA)]

Phân mảnh dọc (Vertical fragmentation)

Phân mảnh dọc không dư thừa (non-redundant fragmentation)

- Phân mảnh dọc không dư thừa là các phân mảnh dọc không chứa thuộc tính chung không khóa nào cả.

- Ví dụ: *GV(MSGV, TENGV, NAMS, NOIS, GT, DC, SDT, NGAYVD)*
- Chia SV thành 2 phân mảnh dọc
  + GV1(MSGV, TENGV, NAMS, NOIS, GT, DC, SDT)
  + GV2(MSGV, NGAYVD)

Phân mảnh dọc dư thừa (redundant fragmentation)

- Phân mảnh dọc dư thừa là các phân mảnh dọc chứa một hoặc nhiều thuộc tính chung không khóa.

- Ví dụ: *GV(MSGV, TENGV, NAMS, NOIS, GT, DC, SDT, NGAYVD)*
- Chia GV thành 2 phân mảnh dọc
  + GV3(MSGV, TENGV, NAMS, NOIS, GT, DC, SDT)
  + GV4(MSGV, TENGV, NGAYVD)

Phân mảnh hỗn hợp

- Một quan hệ toàn cục $R(A_1, A_2, \ldots, A_n)$ được chia thành các quan hệ con Ri kết hợp cả phân mảnh ngang lẫn phân mảnh dọc.
- Ví dụ: cho quan hệ toàn cục

  SV(MSSV, TENSV, NAMS, NOIS, GT, DIACHI, EMAIL, MSK)
- 02 quan hệ SV3, SV4 thỏa điều kiện sau, chúng tạo thành 2 phân mảnh ngang của SV.
  + SV3 = [∂ MSK='CNTT" (SV)]
  + SV4 = ∂ MSK='DTVT" (SV)
- Tạo ra SV9, SV10 theo công thức:
  + SV9=∏ MSSV, NAMS, NOIS, GT, DIACHI, EMAIL, MSK [∂ MSK='CNTT" (SV)]
  + SV10= ∏ MSSV, TENSV [∂ MSK='CNTT" (SV)]
- Ta nói SV9, SV10 là phân mảnh hỗn hợp của SV.

Ưu khuyết của việc phân mảnh

Trùng lắp dữ liệu và cấu trúc (nhân bản)

– Ưu điểm:

+ Tốc độ truy xuất dữ liệu nhanh chóng
+ Do khoảng cách vật lý nhỏ.
+ Số xử lý trong cùng một thời điểm là nhỏ.
+ Sức đề kháng cao: xác suất toàn bộ mạng sụp đổ cùng lúc là nhỏ.

– Khuyết điểm:

+ Cơ sở vật chất tốn kém(CPU, harddisk)
+ Xử lý vấn đề thêm xóa sửa phức tạp do dữ liệu cần đồng nhất.

Ưu khuyết của việc phân mảnh

Trùng về cấu trúc khác dữ liệu (cắt ngang)

– Ưu điểm:

+ Tự nhiên.
+ Hiệu quả tìm kiếm.
+ An toàn dữ liệu, dễ tìm lỗi.

– Khuyết điểm:

+ Tốc độ không tốt khi truy xuất từ xa.
+ Gặp sự cố trên node chứa dữ liệu độc quyền dẫn đến mất mát dữ liệu.
+ Rất phổ biến.

Ưu khuyết của việc phân mảnh

Khác nhau về cấu trúc và dữ liệu (cắt dọc):

- Ưu điểm:
  + Tiết kiệm không gian lưu trữ.
- Khuyết điểm:
  + Mất thời gian lấy dữ liệu từ xa.

Ưu khuyết của việc phân mảnh

Mô hình hỗn hợp

+ Là sự trộn lẫn giữ 3 mô hình 1, 2, 3.

– Ưu điểm:

+ Lấy ưu của 3 mô hình.
+ Độ linh động cao.

– Khuyết:

+ Lấy khuyết của 3 mô hình.
+ Quản lý rất phức tạp.
+ Mang tính tự nhiên cao nhất

Ví dụ

UIT VNUHCM University of Information Technology

Cho CSDL Quản lý sinh viên như sau:

SINHVIEN(MSSV, HoTen, NgVDoan, NgVDang, NgSinh, QueQuan, MaKhoa)

MSSV	HoTen	NgVDoan	NgVDang	NgSinh	QueQuan	MaKhoa
123	Ng. V. A	8/2/2012	8/8/2015	11/2/2000	TPHCM	KHKTTT
124	Ng. V. B	8/1/2017	8/3/2019	9/7/2000	Vinh Long	KHMT

MONHOC(MSMH, TenMH, SoTC, MaKhoaQL)

MSMH	TenMH	SoTC	MaKhoaQL
IE103	QLTT	4	KHKTTT
IT003	CTDL&GT	4	KHMT

DIEMSO(MSSV, MSMH, Diem)

MSSV	MSMH	Diem
123	IE103	8
123	IT003	8
124	IE103	9

KHOA(MaKhoa, TenKhoa, NGTL)

MaKhoa	TenKhoa	NGTL
KHKTTT	Khoa hoc va Ky Thuat Thong tin	09/11/2018
KHMT	Khoa hoc may tinh	08/06/2006

Xây dựng CSDL phân bố cho các phòng ban sau

- Phòng đào tạo.
- Phòng CTSV.
- Văn phòng Khoa (cụ thể là Khoa Khoa học và Kỹ thuật thông tin).

Phòng đào tạo

SINHVIEN(MSSV,	HoTen,	NgSinh,	QueQuan,	MaKhoa)	Phân mảnh dọc
123	*Ng. V. A*	*11/2/2000*	*TPHCM*	*KHKTTT*	
124	*Ng. V. B*	*9/7/2000*	*Vinh Long*	*KHMT*	
MONHOC(MSMH,	TenMH,	SoTC,	MaKhoaQL)		Nhân bản
IE103	*QLTT*	*4*	*KHKTTT*		
IT003	*CTDL&GT*	*4*	*KHMT*		
DIEMSO(MSSV,	MSMH,	Diem)			Nhân bản
123	*IE103*	*8*			
123	*IT003*	*8*			
124	*IE103*	*9*			
KHOA(MaKhoa,	TenKhoa,			NGTL)	Nhân bản
KHKTTT	*Khoa hoc va Ky Thuat Thong tin*			*09/11/2018*	
KHMT	*Khoa hoc may tinh*			*08/06/2006*	

Phòng CTSV

SINHVIEN(MSSV,	HoTen,	NgVDoan,	NgVDang,	NgSinh,	QueQuan,	MaKhoa)	Nhân bản
123	*Ng. V. A*	*8/2/2012*	*8/8/2015*	*11/2/2000*	*TPHCM*	*KHKTTT*	
124	*Ng. V. B*	*8/1/2017*	*8/3/2019*	*9/7/2000*	*Vinh Long*	*KHMT*	

KHOA(MaKhoa,	TenKhoa,	NGTL)	Nhân bản
KHKTTT	*Khoa hoc va Ky Thuat Thong tin*	*09/11/2018*	
KHMT	*Khoa hoc may tinh*	*08/06/2006*	

Văn phòng Khoa

SINHVIEN(MSSV,	HoTen,	NgVDoan,	NgVDang,	NgSinh,	QueQuan)
123	*Ng. V. A*	*8/2/2012*	*8/8/2015*	*11/2/2000*	*TPHCM*

Phân mảnh dọc
Phân mảnh ngang

MONHOC(MSMH,	TenMH,	SoTC)
IE103	*QLTT*	*4*

Phân mảnh ngang, pm dọc

DIEMSO(MSSV,	MSMH,	Diem)
123	*IE103*	*8*
124	*IE103*	*9*

Phân mảnh ngang

THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN

CÁC MỤC TIÊU THIẾT KẾ

1. Sự truy xuất cục bộ.
2. Tính sẵn sàng của các dữ liệu phân tán.
3. Sự phân bố tải.
4. Chi phí lưu trữ.

1. Sự truy xuất cục bộ

- Mục tiêu của sự phân tán dữ liệu là để các ứng dụng truy xuất dữ liệu cục bộ càng nhiều càng tốt, giảm bớt các truy xuất dữ liệu từ xa.
- Việc thiết kế sự phân tán dữ liệu để tối đa hoá truy xuất cục bộ có thể được thực hiện bằng cách thêm số lượng các CSDL cục bộ tahy thế cho các tham khảo CSDL từ xa tương ứng.

2. Tính sẵn sàng của các dữ liệu phân tán.

- Mức độ sẵn sàng cao đối với các ứng dụng chỉ đọc được thực hiện bằng cách lưu trữ nhiều bản sao của cùng một thông tin; hệ thống phải có khả năng chuyển đến bản sao được chọn thích hợp khi một bản sao không được truy xuất bình thường.
- Độ khả tín cũng được thực hiện bằng cách lưu trữ nhiều bản sao, khi đó nó có khả năng phục hồi khi có sự phá huỷ một số bản sao.

3. Sự phân bố tải.

- Sự phân tán bố trên các sites là một tính chất quan trọng của các hệ thống máy tính phân tán.
- Sự phân bố tải để tận dụng sức mạnh của việc sử dụng các máy tính, và cực đại hoá mức độ xử lý song song các lệnh thực thi của các ứng dụng. Vì sự phân bố tải có thể ảnh hưởng xấu đến sự truy xuất cục bộ nên cần xem xét để cân bằng hai mục tiêu này.

4. Chi phí lưu trữ

- Sự phân tán cơ sở dữ liệu phản ánh chi phí của sự lưu trữ tại các sites khác nhau.
- Tuy nhiên chi phí lưu trữ dữ liệu không đáng kể so với chi phí xuất nhập, chi phí truyền thông của các ứng dụng.
- Những giới hạn của bộ lưu trữ phải được xem xét kỹ.

CÁC CHIẾN LƯỢC THIẾT KẾ CSDL PHÂN TÁN

– Có hai cách tiếp cận cho thiết kế cơ sở dữ liệu:

+ Tiếp cận từ trên-xuống (top-down).

+ Tiếp cận từ dưới-lên (bottom-up).

TOP-DOWN

- Đặc điểm của tiếp cận Top-down:
  - + Thiết kế lược đồ phổ quát
  - + Thiết kế sự phân mảnh cơ sở dữ liệu
  - + Cấp phát các mảnh đến các sites, tạo các ảnh vật lý của chúng.
- Cách tiếp cận này thích hợp đối với các hệ thống được phát triển từ đầu và nó cho phép thiết kế một cách hợp lý.
- Khi cơ sở dữ liệu phân tán được phát triển như là sự tổ hợp các cơ sở dữ liệu sẵn có thì nó lại không dễ dàng đối với phương pháp tiếp cận này. Trong trường hợp này lược đồ phổ quát thường được tạo ra từ sự thoả hiệp giữa các mô tả dữ liệu sẵn có. Từ đó cách tiếp cận từ dưới-lên có thể được sử dụng để thiết kế sự phân tán dữ liệu.

BOTTOM UP

— Cách thiết kế Bottom-up:

+ Chọn một mô hình cơ sở dữ liệu chung để mô tả lược đồ phổ quát của cơ sở dữ liệu.
+ Chuyển dịch mỗi lược đồ cục bộ vào trong mô hình dữ liệu chung.
+ Tổ hợp lại lược đồ cục bộ vào trong lược đồ phổ quát chung.
+ Ba vấn đề này không riêng biệt gì đối với cơ sở dữ liệu phân tán mà nó hiện diện ngay trong các hệ thống tập trung.

TÀI LIỆU THAM KHẢO

1. Nguyễn Gia Tuấn Anh, Trương Châu Long, *Bài tập và bài giải SQL Server*, NXB Thanh niên (2005).
2. Đỗ Phúc, Nguyễn Đăng Tỵ, *Cơ sở dữ liệu*, NXB Đại học quốc gia TPHCM (2010).
3. *Nguyễn Gia Tuấn Anh, Mai Văn Cường, Bùi Danh Hường, Cơ sở dữ liệu nâng cao*, NXB Đại học quốc gia TPHCM (2019).
4. Itzik Ben-Gan, *Microsoft SQL Server 2012- TSQL Fundamentals*.


Q
A

Bài tập

1. Cho CSDL toàn cục sau, hãy đề xuất 1 mô hình phân tán, và lí giải cách chọn lựa; biết nhà trường hiện có 5 cơ sở tại các tỉnh A, B, C, D, E; trường gồm 8 khoa: k1, k2, k3,... k8.

SV(#mssv, tensv, noisinh, namsinh, msk)

KHOA(#msk, tenkhoa)

MON(#msm, tenm, STC)

SV-MON(#mssv, #msm, diem)

COSO (#mscs, tencs, diachi, sdt)

Bài tập

2. Tìm các ứng dụng phổ biến, đang sử dụng mô hình phân tán? Tại sao đó là lựa chọn tốt nhất?

3. Tiến hành cài đặt bài 1 bằng 1 hệ QTCSDL? Viết các thực nghiệm so sánh nó với CSDLTT.

4. So sánh các tính năng phân tán được hỗ trợ trên 2 hệ QTCSDL Oracle và MySQL?