Cơ sở dữ liệu phân tán

Phạm Minh Khan
pmkhan@hcmunre.edu.vn

Chương 4: Tối ưu hóa truy vấn trong CSDL phân tán

1. Truy vấn, biểu thức chuẩn tắc của truy vấn
2. Tối ưu hóa truy vấn trong cơ sở dữ liệu tập trung
3. Tối ưu hóa trong cơ sở dữ liệu phân tán

Mở đầu

Tối ưu hóa nhằm để làm gì ?

- Giảm thời gian thực hiện truy vấn
- Giảm vùng nhớ trung gian
- Chi phí truy vấn thông trong quá trình thực hiện truy vấn

Ví dụ

Ta sử dụng một cơ sở dữ liệu gồm các quan hệ sau đây để minh họa cho các nội dung được trình bày trong chương:

Sinhvien *(masv, hoten, ngaysinh, malop)*
Lop *(malop, tenlop, malt, tenkhoa)*
Monhoc*(mamh, tenmh)*
Hoc *(masv, mamh, Diem)*

Trong đó :

Sinhvien : chứa thông tin về sinh viên gồm: mã sinh viên (masv), họ tên (hoten), Ngày sinh (ngaysinh), thuộc lớp (malop). Khóa là masv.

Lop : chứa thông tin về lớp học gồm: mã lớp (malop), tên lớp (tenlop), mã lớp Trưởng (malt), thuộc khoa (tenkhoa). Khóa là malop.

Monhoc : chứa thông tin về môn học gồm: mã môn học (mamh), tên môn học (tenmh).

Hoc : chứa thông tin về sinh viên (masv) học môn học (mamh) có điểm thi cuối Kỳ (diem). Khóa là masv và mamh.

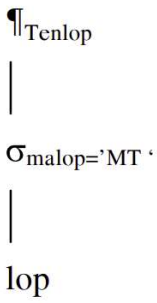
Truy vấn, biểu thức chuẩn tắc của truy vấn

Truy vấn (query) là một biểu thức được biểu diễn bằng một ngôn ngữ thích hợp và dùng để xác định một phần dữ liệu được chứa trong cơ sở dữ liệu.

Ví dụ: Xét truy vấn **cho biết tên lớp** của lớp có **mã lớp là 'MT'** . Truy vấn này có thể được biểu diễn bởi một biểu thức đại số quan hệ như sau :

$$¶_{Tenlop}(\sigma_{malop='MT'}(lop))$$

Truy vấn trên có thể được biểu diễn bằng một cây toán tử như sau:



Một truy vấn có thể được biểu diễn bởi một cây toán tử. Một cây toán tử (operator tree) của một truy vấn, còn được gọi là cây truy vấn (query tree) hoặc cây đại số quan hệ (relational algebra tree).

Truy vấn, biểu thức chuẩn tắc của truy vấn

Biểu thức chuẩn tắc của một biểu thức đại số quan hệ trên lược đồ toàn cục là một biểu thức có được bằng cách thay thế mỗi tên quan hệ toàn cục xuất hiện trong biểu thức bởi biểu thức tái lập của quan hệ toàn cục này.

Ví dụ : Giả sử chúng ta có hai khoa tên là 'CNTT' và 'QTKD'. Quan hệ lop được phân mảnh ngang dựa vào tenkhoa thành hai mảnh lop1 và lop2

$Lop1 = \sigma_{tenkhoa='CNTT'}(lop)$

$Lop2 = \sigma_{tenkhoa='QTKD'}(lop)$

Biểu thức tái lập của quan hệ toàn cục lop là :

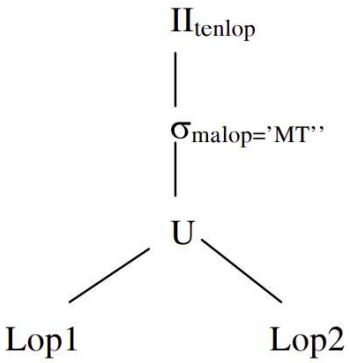
$Lop = lop1 \cup lop2$

Biểu thức chuẩn tắc của biểu thức truy vấn là :

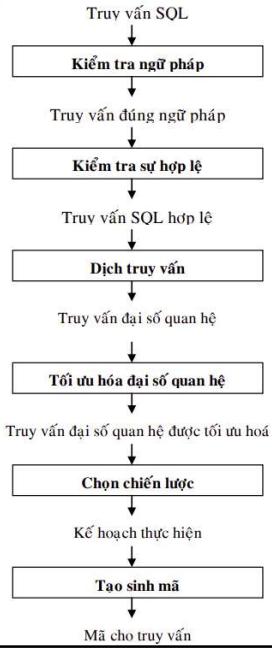
$\Pi_{tenlop}(\sigma_{malop='MT'}(lop1 \cup lop2))$

Truy vấn, biểu thức chuẩn tắc của truy vấn

Thay thế quan hệ toàn cục lop trong cây toán tử bởi biểu thức tái lập ở trên, chúng ta được cây toán tử như sau:



Tối ưu hóa truy vấn trong cở sở dữ liệu tập trung



- **Bước 1: Kiểm tra ngữ pháp (Syntax Checking)**

Ví dụ: Xét truy vấn Q1

Q1: ***SELECT*** *masv,hoten* ***FORM*** *sinhvien;*

FROM

- **Bước 2: Kiểm tra sự hợp lệ (Validation)**

- Kiểm tra sự tồn tại của các đối tượng dữ liệu (các cột, các biến, các bảng,…) của truy vấn trong CSDL.
- Kiểm tra sự hợp lệ về kiểu dữ liệu của các đối tượng dữ liệu (các cột, các biến, …) trong truy vấn.

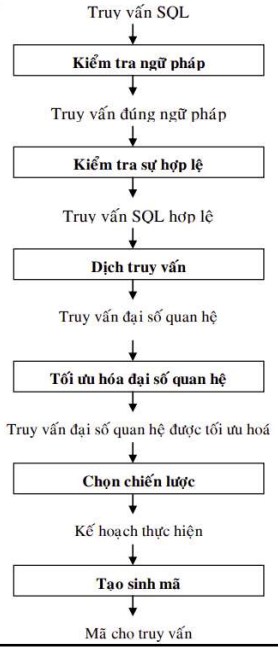
Ví dụ : Xét truy vấn Q2

Q2: SELECT masv, hoten FROM sinh_vien ;

Ví dụ: Xét truy vấn Q3

Q3: SELECT masv, hoten FROM sinhvien
WHERE masv='123';

Tối ưu hóa truy vấn trong cở sở dữ liệu tập trung



- **Bước 3: Dịch truy vấn (Translation) :** DBMS sẽ biến đổi truy vấn hợp lệ này thành một dạng biểu diễn bên trong hệ thống ở mức thấp hơn mà DBMS có thể sử dụng được.

Ví dụ: Xét truy vấn Q4 sau đây cho biết các mã môn học mà các sinh viên thuộc lớp có mã 'MT' học.

Q4 : ***SELECT*** *DISTINCT mamh*

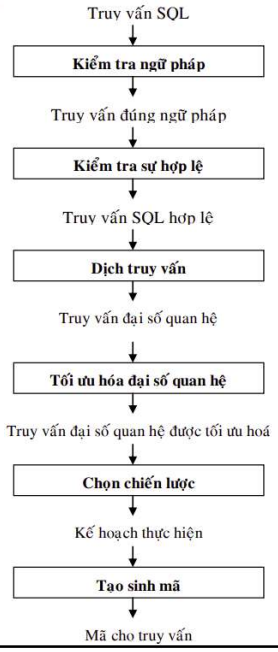
FROM *sinhvien, hoc*

WHERE *sinhvien.masv=hoc.masv AND malop='MT'*

Truy vấn này sẽ được biến đổi thành biểu thức đại số quan hệ như sau:

$$\Pi_{mamh}(\sigma_{malop='MT'}(sinhvien \blacktriangleright\blacktriangleleft_{masv=masv} hoc))$$

Tối ưu hóa truy vấn trong cở sở dữ liệu tập trung



- **Bước 4: Tối ưu hóa biểu thức đại số quan hệ (relational Algebra Optimization)**

DBMS sử dụng các phép biến đổi tương đương của đại số quan hệ để biến đổi biểu thức đại số quan hệ có được ở bước 3 thành một biểu thức đại số quan hệ tương đương nhưng biểu thức sau sẽ hiệu quả hơn: **loại bỏ các phép toán không cần thiết và giảm vùng nhớ trung gian**.

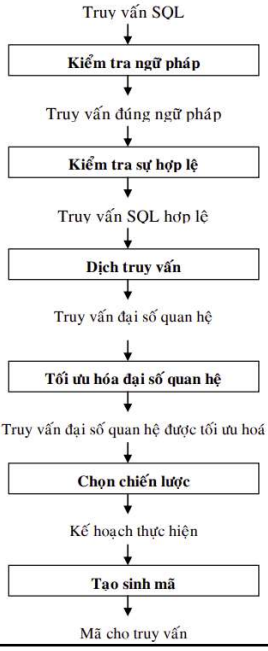
Ví dụ*: Biểu thức quan hệ của truy vấn Q4 ở cuối bước 3 có thể' được biến đổi thành biểu thức đại số quan hệ tương đương tốt hơn như sau:*

$$\Pi_{mamh}\ (\Pi_{masv}(\sigma_{malop='MT'}(sinhvien)) \blacktriangleright\blacktriangleleft_{masv=masv} \Pi_{masv,mamh}(hoc))$$

Truy vấn SQL:

```
SELECT MaMH
FROM (SELECT MaSV FROM SinhVien WHERE MaLop = @malop) SV,
     (SELECT MaSV, MaMH FROM Hoc) Hoc
WHERE SV.MaSV = Hoc.MaSV
```

Tối ưu hóa truy vấn trong cở sở dữ liệu tập trung



- **Bước 5: Chọn lựa chiến lược truy xuất (strategy selection)**

DBMS sử dụng các thông số về ***kích thước của các bảng, các chỉ mục… để xác định cách xử lý truy vấn***.

DBMS sẽ ***đánh giá chi phí*** của các kế hoạch thực hiện khác nhau có thể có để từ đó chọn ra một kế hoạch thực hiện (execution plan) cụ thể ***sao cho tốn ít chi phí nhất (thời gian xử lý và vùng nhớ trung gian)***.

Các thông số dùng để đánh giá chi gồm: ***số lần và loại truy xuất đĩa, kích thước của vùng nhớ chính và vùng nhớ ngoài, và thời gian thực hiện của các tác vụ để tạo ra kết quả của truy vấn.***

- **Bước 6: Tạo sinh mã (code generation)**

Kế hoạch thực hiện của truy vấn có được ở cuối bước 5 sẽ được mã hoá và được thực hiện.

Các phép biến đổi tương đương

Ký hiệu ≡ là sự tương đương

(1) $P_1 \wedge P_2 \equiv P_2 \wedge P_1$
(2) $P_1 \vee P_2 \equiv P_2 \vee P_1$
(3) $P_1 \wedge (P_2 \wedge P_3) \equiv (P_1 \wedge P_2) \wedge P_3$
(4) $P_1 \vee (P_2 \vee P_3) \equiv (P_1 \vee P_2) \vee P_3$
(5) $P_1 \wedge (P_2 \vee P_3) \equiv (P_1 \wedge P_2) \vee (P_1 \wedge P_3)$
(6) $P_1 \vee (P_2 \wedge P_3) \equiv (P_1 \vee P_2) \wedge (P_1 \vee P_3)$
(7) $\neg(P_1 \wedge P_2) \equiv \neg P_1 \vee \neg P_2$
(8) $\neg(P_1 \vee P_2) \equiv \neg P_1 \wedge \neg P_2$
(9) $\neg(\neg P) \equiv P$
(10) $P \wedge P \equiv P$
(11) $P \vee P \equiv P$
(12) $P \wedge true \equiv P$
(13) $P \vee false \equiv P$
(14) $P \wedge false \equiv false$
(15) $P \vee true \equiv true$
(16) $P \wedge \neg P \equiv false$
(17) $P \vee \neg P \equiv true$
(18) $P_1 \wedge (P_1 \vee P_2) \equiv P_1$
(19) $P_1 \vee (P_1 \wedge P_2) \equiv P_1$

Các phép biến đổi tương đương

Ví dụ: Xét truy vấn

SELECT malop
FROM sinhvien
WHERE (NOT (malop='MT1')
AND (malop='MT1' OR malop='MT2')
AND NOT (malop='MT2')) OR hoten='Nam'

Điều kiện của mệnh đề WHERE là:

(NOT (malop='MT1') AND (malop='MT1' OR malop='MT2')
AND NOT (malop='MT2')) OR hoten='Nam'

Ký hiệu:
P1 là malop='MT1'
P2 là malop='MT2'
P3 là hoten='Nam'
Điều kiện q sẽ là: (¬P1 ^ (P1 v P2) ^ ¬ P2) v P3

Các phép biến đổi tương đương

Bằng cách áp dụng các phép biến đổi (3), (5) để đưa điều kiện q về dạng chuẩn hợp:

(((¬P1 ^ P1) v (¬P1 ^ P2)) ^ ¬ P2) v P3

(¬P1 ^ P1 ^ ¬P2) v (¬P1 ^ P2 ^ ¬ P2) v P3

Bằng cách áp dụng phép biến đổi (16), chúng ta được:

(false ^ ¬P2) v (¬P1 ^ false) v P3

Áp dụng phép biến đổi (14), chúng ta được:

False v False v P3

Áp dụng phép biến đổi (15), chúng ta được điều kiện q cuối cùng là P3, tức là hoten='Nam'.

➔ **SELECT malop FROM sinhvien WHERE hoten='Nam';**

Các phép biến đổi tương đương trên phép kết

Ký hiệu ≡ là sự tương đương

(1) $R \bowtie R \equiv R$

(2) $R \cup R \equiv R$

(3) $R - R \equiv \varnothing$

(4) $R \bowtie \sigma_F(R) \equiv \sigma_F R$

(5) $R \cup \sigma_F(R) \equiv R$

(6) $R - \sigma_F(R) \equiv \sigma_{\neg F}(R)$

(7) $\sigma_{F1}(R) \bowtie \sigma_{F2}(R) \equiv \sigma_{F1 \wedge F2}(R)$

(8) $\sigma_{F1}(R) \cup \sigma_{F2}(R) \equiv \sigma_{F1 \vee F2}(R)$

(9) $\sigma_{F1}(R) - \sigma_{F2}(R) \equiv \sigma_{F1 \wedge \neg F2}(R)$

(10) $R \cap R \equiv R$

(11) $R \cap \sigma_F(R) \equiv \sigma_F R$

(12) $\sigma_{F1}(R) \cap \sigma_{F2}(R) \equiv \sigma_{F1 \wedge F2}(R)$

(13) $\sigma_F(R) - R \equiv \varnothing$

Tối ưu hóa trong cơ sở dữ liệu phân tán

Tối ưu hoá truy vấn trong cơ sở dữ liệu phân tán bao gồm một số bước đầu của tối ưu hóa truy vấn trong cơ sở dữ liệu tập trung và một số bước tối ưu hóa có liên quan đến sự phân tán dữ liệu.

Bước 1 Phân rã truy vấn (Query Decomposition)

Bước 2 Định vị dữ liệu

Bước 3 Tối ưu hoá truy vấn toàn cục

Bước 4 Tối ưu hoá truy vấn cục bộ

Tối ưu hóa trong cơ sở dữ liệu phân tán

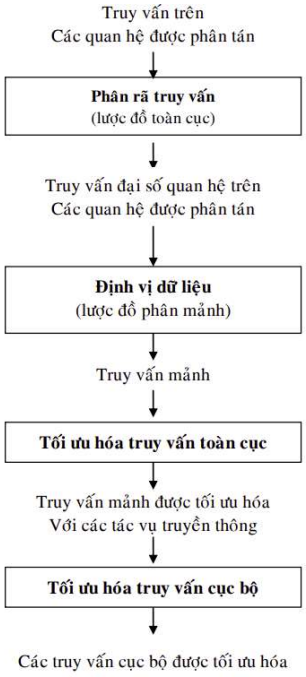
Bước 1 Phân rã truy vấn (Query Decomposition)

Bước 2 Định vị dữ liệu

Bước 3 Tối ưu hoá truy vấn toàn cục

Bước 4 Tối ưu hoá truy vấn cục bộ

Sơ đồ tối ưu hóa truy vấn trong cơ sở dữ liệu phân tán bao gồm các bước sau:



Tối ưu hóa trong cơ sở dữ liệu phân tán

Bước 1 Phân rã truy vấn (Query Decomposition)

Bước 1.1- Phân tích truy vấn

DBMS kiểm tra ngữ pháp của truy vấn, kiểm tra sự tồn tại của các đối tượng dữ liệu (tên cột, tên bảng,…) của truy vấn trong cơ sở dữ liệu, phát hiện các phép toán trong truy vấn bị sai về kiểu dữ liệu, điều kiện của mệnh đề WHERE có thể bị sai về ngữ nghĩa.

Phân tích điều kiện của mệnh đề WHERE để phát hiện truy vấn bị sai. Có hai loại sai:

- Sai về kiểu dữ liệu (type incorrect)
- Sai về ngữ nghĩa (semantically incorrect)

Tối ưu hóa trong cơ sở dữ liệu phân tán

Bước 1 Phân rã truy vấn (Query Decomposition)

Bước 1.1- Phân tích truy vấn

Truy vấn bị sai về kiểu dữ liệu

Ví dụ: *Xét truy vấn:*

SELECT mssv, hoten FROM sinhvien
WHERE masv='123';

Truy vấn này có hai lỗi sai:

(1) mssv không tồn tại trong quan hệ sinhvien

(2) masv thuộc kiểu number không thể so sánh với hằng chuỗi '123'.

Tối ưu hóa trong cơ sở dữ liệu phân tán

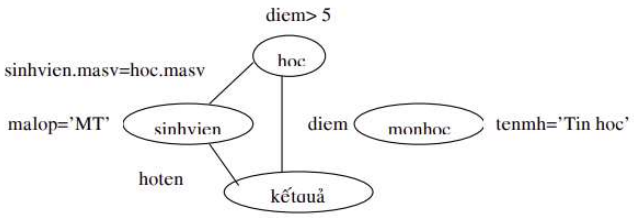
Bước 1 Phân rã truy vấn (Query Decomposition)

Bước 1.1- Phân tích truy vấn

Truy vấn bị sai về ngữ nghĩa

Ví dụ: *Xét truy vấn:*

SELECT hoten, diem
FROM sinhvien, hoc, ***monhoc***
WHERE sinhvien.masv=hoc.masv
AND malop='MT' AND diem > 5
AND tenmh = 'Tin hoc';



SELECT hoten, diem
FROM sinhvien, hoc
WHERE sinhvien.masv =hoc.masv AND malop ='MT'
AND diem > 5;

Tối ưu hóa trong cơ sở dữ liệu phân tán

Bước 1 Phân rã truy vấn (Query Decomposition)

Bước 1.2- Chuẩn hóa điều kiện của mệnh đề WHERE

Điều kiện ghi trong mệnh đề WHERE là một biểu thức luận lý có thể bao gồm các phép toán luận lý (not, and, or) được viết dưới một dạng bất kỳ. Ký hiệu các phép toán luận lý: not (-), and (^), or (v).

Bước 1.3- Đơn giản hoá điều kiện của mệnh đề WHERE

Bước này sử dụng các phép biến đổi tương đương của các phép toán luận lý (not, and, or) để rút gọn điều kiện của mệnh đề WHERE.

Bước 1.4- Biến đổi truy vấn thành một biểu thức đại số quan hệ hiệu quả

- Biến đổi truy vấn thành một biểu thức đại số quan hệ, biểu diễn biểu thức đại số quan hệ này bằng một cây toán tử.
- Đơn giản hóa cây toán tử để có được một biểu thức đại số quan hệ hiệu quả.

Tối ưu hóa trong cơ sở dữ liệu phân tán

Bước 1.4- Biến đổi truy vấn thành một biểu thức đại số quan hệ hiệu quả

- **Biểu diễn truy vấn bằng cây toán tử**

Quá trình biến đổi một truy vấn được viết bằng lệnh SELECT thành một cây toán tử bao gồm các bước sau:

(1) Các nút lá được tạo lập từ các quan hệ ghi trong mệnh đề From

(2) Nút gốc được tạo lập bằng phép chiếu trên các thuộc tính ghi trong mệnh đề SELECT.

(3) Điều kiện ghi trong mệnh đề WHERE được biến đổi thành một chuỗi thích hợp các phép toán đại số quan hệ (phép chọn, phép kết, phép hợp…) đi từ các nút lá đến nút gốc. Chuỗi các phép toán này có thể được cho trực tiếp bởi thứ tự của các vị từ đơn giản và các phép toán luận lý.

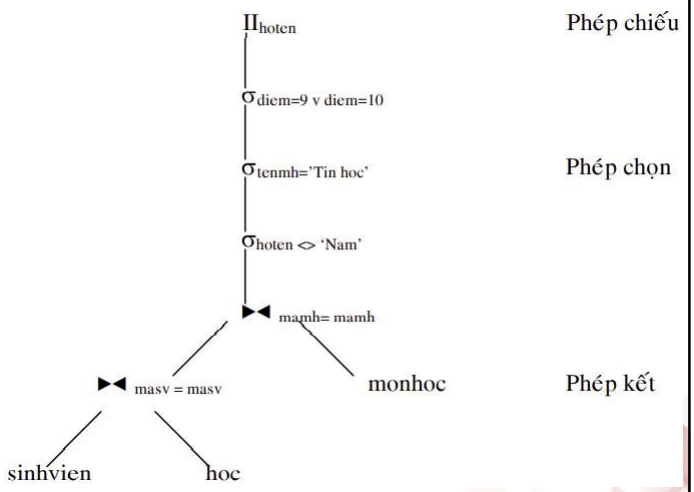
Bước 1.4- Biến đổi truy vấn thành một biểu thức đại số quan hệ hiệu quả

- **Biểu diễn truy vấn bằng cây toán tử**

Ví dụ: Xét truy vấn cho biết họ tên của các sinh viên không phải là 'Nam' học môn học 'Tin hoc' đạt điểm 9 hoặc 10.

```
SELECT hoten
FROM sinhvien, hoc, monhoc
WHERE sinhvien.masv= hoc.masv
 AND hoc.mamh= monhoc.mamh
 AND hoten<> 'Nam'
 AND tenmh= 'Tin hoc'
 AND (diem= 9 OR diem = 10);
```


Ví dụ: Xét truy vấn cho biết họ tên của các sinh viên không phải là 'Nam' học môn học 'Tin hoc' đạt điểm 9 hoặc 10.



Tối ưu hóa trong cơ sở dữ liệu phân tán

- **Đơn giản hóa cây toán tử**

Đơn giản hoá cây toán tử nhằm mục đích để đạt hiệu quả (loại bỏ các phép toán dư thừa trên các quan hệ, giảm vùng nhớ trung gian, giảm thời gian xử lý truy vấn) bằng cách sử dụng các phép biến đổi tương đương của các phép toán đại số quan hệ.

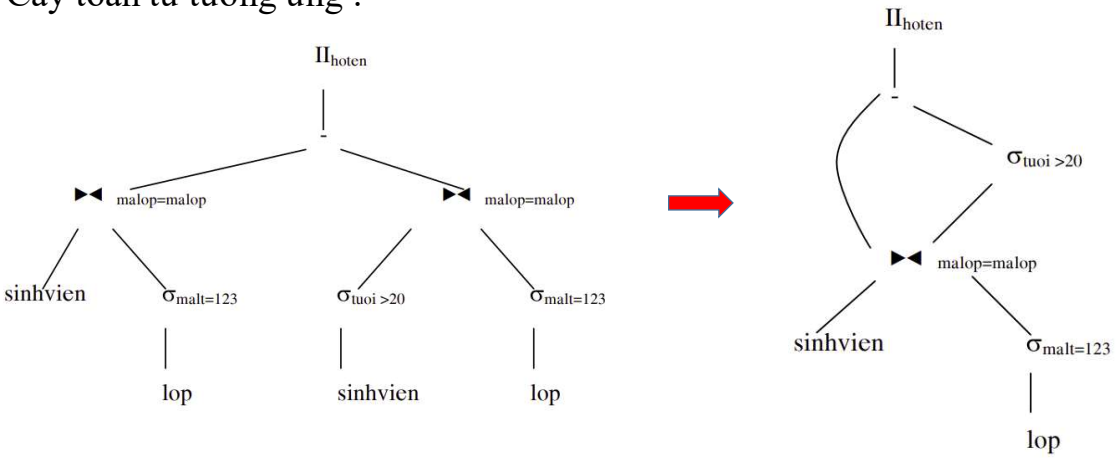
Ví dụ: Xét truy vấn cho biết các họ tên của các sinh viên thuộc lớp có mã lớp trưởng là 123 và các sinh viên này có tuổi không lớn hơn 20 tuổi. Một biểu thức cho truy vấn này là:

$$\Pi_{hoten}((sinhvien \bowtie_{malop=malop} \sigma_{malt=123}(lop)) - (\sigma_{tuoi>20}(sinhvien) \bowtie_{malop=malop} \sigma_{malt=123}(lop))$$

Tối ưu hóa trong cơ sở dữ liệu phân tán

- **Đơn giản hóa cây toán tử**

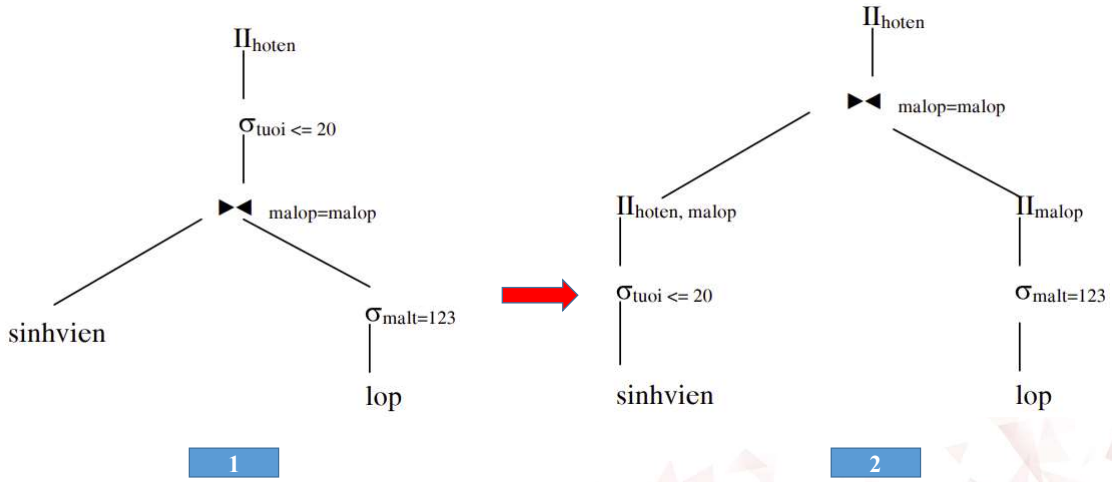
Cây toán tử tương ứng :



Tối ưu hóa trong cơ sở dữ liệu phân tán

- **Đơn giản hóa cây toán tử**

Cây toán tử tương ứng :



Tối ưu hóa trong cơ sở dữ liệu phân tán

- **Đơn giản hóa cây toán tử**

Cây toán tử tương ứng :

Biểu thức đại số quan hệ sau khi đã đơn giản hoá là :

$$\Pi_{hoten}\ ((sinhvien \bowtie_{malop=malop} \sigma_{malt=123}(lop)) - (\sigma_{tuoi>20}(sinhvien) \bowtie_{malop=malop} \sigma_{malt=123}(lop))$$

$$\Pi_{hoten}(\Pi_{hoten,malop}(\sigma_{tuoi<=20}(sinhvien)) \bowtie_{malop=malop} \Pi_{malop}(\sigma_{malt=123}(lop)))$$

Tối ưu hóa trong cơ sở dữ liệu phân tán

Bước 2 – Định vị dữ liệu

Bước định vị dữ liệu (Data Localization) còn được gọi là bước tối ưu hóa truy vấn trên lược đồ phân mảnh . Bước này biến đổi truy vấn toàn cục (kết quả của Bước 1) thành các truy vấn mảnh hiệu quả: *loại bỏ các phép toán đại số quan hệ không cần thiết trên các mảnh và giảm vùng nhớ trung gian.*

Tối ưu hóa truy vấn trên lược đồ phân mảnh bao gồm 2 bước sau:

Bước 1: Biến đổi biểu thức đại số quan hệ trên lược đồ toàn cục

Bước 2: Đơn giản hoá biểu thức đại số quan hệ trên lược đồ phân mảnh

Tối ưu hóa trong cơ sở dữ liệu phân tán

Bước 2 – Định vị dữ liệu

Bước 2.1: Biến đổi biểu thức đại số quan hệ trên lược đồ toàn cục

Xét lược đồ quan hệ sinhvien và lop sau đây:

Sinhvien (masv, hoten, tuoi, malop)

Lop (malop, tenlop, malt, tenkhoa)

Có hai khoa tên là 'CNTT' và 'DIEN'. Quan hệ lop được phân mảnh ngang dựa vào tenkhoa thành hai mảnh lop1 và lop2. Quan hệ sinhvien được phân mảnh ngang suy dẫn theo lop dựa vào malop thành hai mảnh sinhvien1 và sinhvien2. Lược đồ phân mảnh như sau:

Lop1 (malop, tenlop, malt, tenkhoa)

Lop2 (malop, tenlop, malt, tenkhoa)

Sinhvien1 (masv, hoten, tuoi, malop)

Sinhvien2 (masv, hoten, tuoi, malop)

Các biểu thức tái lập của quan hệ lop và sinhvien là:

Lop = Lop1 U Lop2

Sinhvien= sinhvien1 U sinhvien2

Bước 2 – Định vị dữ liệu

Bước 2.1: Biến đổi biểu thức đại số quan hệ trên lược đồ toàn cục

Xét lược đồ quan hệ sinhvien và lop sau đây:

Sinhvien (masv, hoten, tuoi, malop)

Lop (malop, tenlop, malt, tenkhoa)

Trong đó:

$Lop1 = \sigma_{tenkhoa = 'CNTT'}(lop)$

$Lop2 = \sigma_{tenkhoa = 'DIEN'}(lop)$

$Sinhvien1 = sinhvien \triangleright\!< (Lop1)$

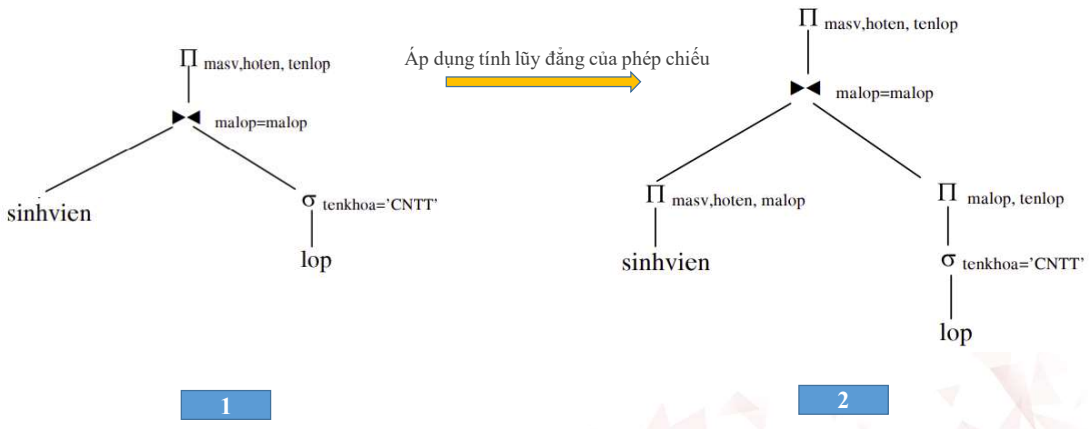
$Sinhvien2 = sinhvien \triangleright\!< (Lop2)$

Tối ưu hóa trong cơ sở dữ liệu phân tán

Bước 2 – Định vị dữ liệu

Bước 2.1: Biến đổi biểu thức đại số quan hệ trên lược đồ toàn cục

Ví dụ: Xét cây toán tử



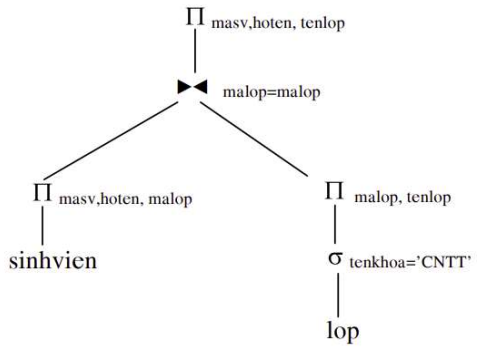
Tối ưu hóa trong cơ sở dữ liệu phân tán

Ví dụ: Xét cây toán tử, thay thế sinhvien và lop bởi biểu thức tái lập:

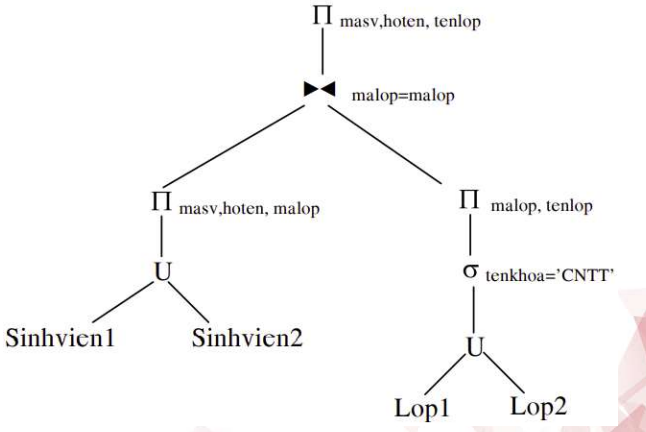
Sinhvien = sinhvien1 U sinhvien2

Lop = lop1 U lop2

Ta được cây toán tử sau:



2



3

Tối ưu hóa trong cơ sở dữ liệu phân tán

Bước 2.2- Đơn giản hoá biểu thức đại số quan hệ trên lược đồ phân mảnh

Đơn giản hoá biểu thức đại số quan hệ trên lược đồ phân mảnh để có được một biểu thức hiệu quả (loại bỏ các phép toán không can thiết, giảm vùng nhớ trung gian)

Các phép biến đổi tương đương:

(1) $\sigma_F(\varnothing) \equiv \varnothing$
(2) $\Pi_X(\varnothing) \equiv \varnothing$
(3) $R \times \varnothing \equiv \varnothing$
(4) $R \cup \varnothing \equiv R$
(5) $R \cap \varnothing \equiv \varnothing$
(6) $R - \varnothing \equiv R$
(7) $\varnothing - R \equiv \varnothing$
(8) $R \blacktriangleright\!\blacktriangleleft \varnothing \equiv \varnothing$
(9) $R \blacktriangleright\!< \varnothing \equiv \varnothing$
(10) $\varnothing \blacktriangleright\!< R \equiv \varnothing$

Tối ưu hóa trong cơ sở dữ liệu phân tán

Bước 2.2- Đơn giản hoá biểu thức đại số quan hệ trên lược đồ phân mảnh

Ví dụ : Xét cây toán tử trên lược đồ phân mảnh trên

Đẩy phép chọn và phép chiếu xuống khỏi phép hợp ta được:

$$\Pi_{malop,tenlop}(\sigma_{tenkhoa='CNTT'}(lop1 \cup lop2))$$
$$= \Pi_{malop,tenlop}(\sigma_{tenkhoa='CNTT'}(lop1)) \cup \Pi_{malop,tenlop}(\sigma_{tenkhoa='CNTT'}(lop2))$$

Ta nhận thấy kết quả của phép chọn $\sigma_{tenkhoa='CNTT'}(lop2)$ là rỗng và phép chọn $\sigma_{tenkhoa='CNTT'}(lop1)$ là không cần thiết vì điều kiện chọn của lop1 là tenkhoa= 'CNTT'.

Do đó:

$$\Pi_{malop,tenlop}(\sigma_{tenkhoa='CNTT'}(lop1 \cup lop2)) = \Pi_{malop,tenlop}(lop1)$$

Đẩy phép chiếu xuống khỏi phép hợp trong biểu thức:

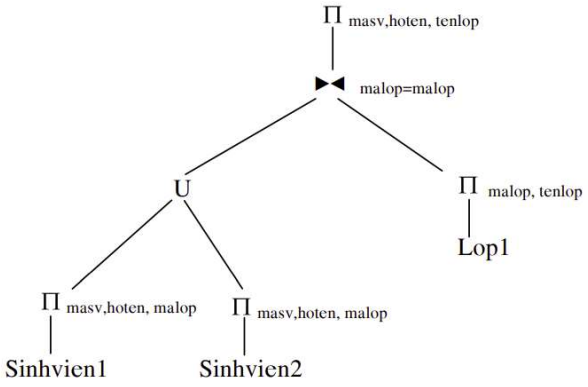
$$\Pi_{masv,hoten,\ malop}(sinhvien1 \cup sinhvien2) =$$
$$\Pi_{masv,hoten,\ malop}(sinhvien1) \cup \Pi_{masv,hoten,\ malop}(sinhvien2)$$

Tối ưu hóa trong cơ sở dữ liệu phân tán

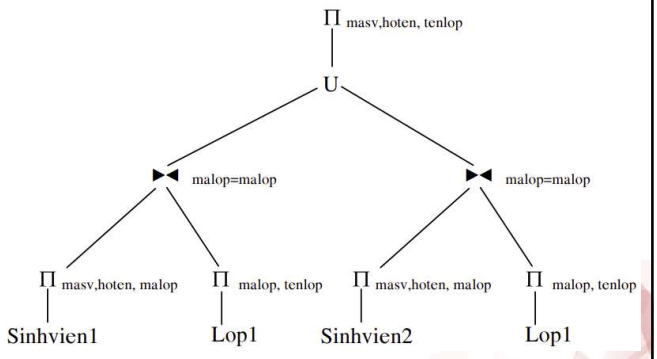
Bước 2.2- Đơn giản hoá biểu thức đại số quan hệ trên lược đồ phân mảnh

Ví dụ : Xét cây toán tử trên lược đồ phân mảnh trên

Ta có cây toán tử:



Sau đó phân phối phép kết với phép hợp ta được:

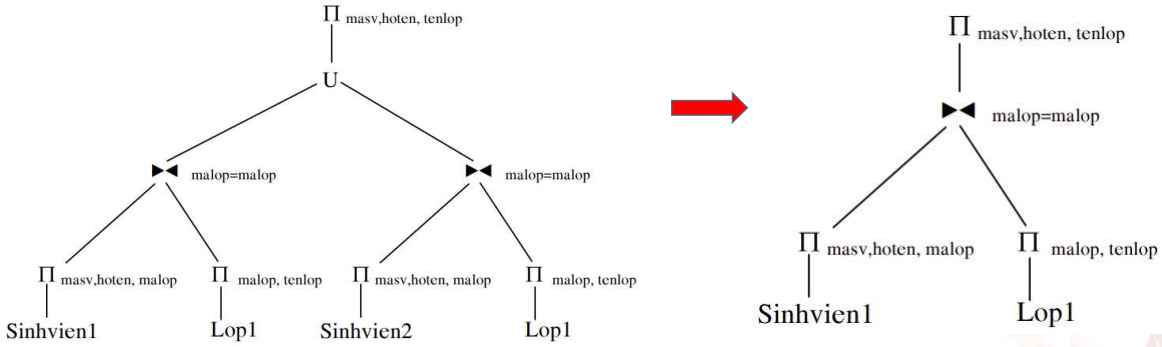


Tối ưu hóa trong cơ sở dữ liệu phân tán

Bước 2.2- Đơn giản hoá biểu thức đại số quan hệ trên lược đồ phân mảnh

Ví dụ : Xét cây toán tử trên lược đồ phân mảnh trên

Cuối cùng ta có cây toán tử trên lược đồ phân mảnh như sau:



Tối ưu hóa trong cơ sở dữ liệu phân tán

Bước 3 Tối ưu hoá truy vấn toàn cục

Bước tối ưu hoá truy vấn toàn cục nhằm để tìm ra một chiến lược thực hiện truy vấn sao cho chiến lược này gần tối ưu =>(*theo nghĩa giảm thời gian thực hiện truy vấn trên dữ liệu được phân tán, giảm vùng nhớ trung gian*).

Tối ưu hóa truy vấn toàn cục là tìm ra một thứ tự thực hiện các phép toán trong biểu thức truy vấn sao cho ít tốn thời gian nhất.

Bước 4 Tối ưu hoá truy vấn cục bộ

Tối ưu hoá truy vấn cục bộ nhằm để thực hiện các truy vấn con được phân tán tại mỗi vị trí, gọi là **truy vấn cục bộ có chứa các mảnh**, sau đó được **tối ưu hoá trên lược đồ cục bộ tại mỗi vị trí**. Tối ưu hoá truy vấn cục bộ sử dụng các thuật toán tối ưu hoá truy vấn của cơ sở dữ liệu tập trung

Tài liệu tham khảo

- Tài liệu giảng dạy cơ sở dữ liệu phân tán của PIIT
- Cơ sở dữ liệu phân tán, PGS.TS Nguyễn Mậu Hân

Thank you for listening