Chương 4 Tối ưu hóa truy vấn trong cơ sở dữ liệu phân tán

Mục tiêu

Chương này đề cập đến vấn đề tối ưu hoá trong cơ sở dữ liệu phân tán nghĩa là *giảm chi phí bộ nhớ trung gian, giảm thời gian truy vấn cũng như giảm thời gian truyền dữ liệu* trong các truy vấn phân tán.

Các vấn đề được đề cập trong chương này như sau:

4.1. Truy vấn. Biểu thức chuẩn tắc của truy vấn:

Phần này nêu lên khái niệm về truy vấn và thế nào là biểu thức chuẩn tắc của một câu truy vấn. Biểu thức chuẩn tắc là một biểu thức được sử dụng nhiều trong việc truy vấn cơ sở dữ liệu phân tán.

4.2. Tối ưu hóa truy vấn trong cơ sở dữ liệu tập trung:

Phần này nhắc lại quá trình tối ưu hoá một câu truy vấn cục bộ, nó gồm các bước sau:

4.2.1. Bước 1- Kiểm tra ngữ pháp
4.2.2. Bước 2- Kiểm tra sự hợp lệ
4.2.3. Bước 3- Dịch truy vấn
4.2.4. Bước 4- Tối ưu hóa biểu thức đại số quan hệ
4.2.5. Bước 5- Chọn lựa chiến lược truy xuất
4.2.6. Bước 6-Tạo sinh mã

4.3. Tối ưu hóa trong cơ sở dữ liệu phân tán:

Phần này trình bày quá trình tối ưu hoá một câu truy vấn phân tán, nó bao gồm các bước sau:

4.3.1. Bước 1 – Phân rã truy vấn
4.3.1.1. Bước 1.1- Phân tích truy vấn
4.3.1.2. Bước 1.2- Chuẩn hóa điều kiện của mệnh đề WHERE
4.3.1.3. Bước 1.3- Đơn giản hóa điều kiện của mệnh đề WHERE
4.3.1.4. Bước 1.4- Biến đổi truy vấn thành biểu thức đại số quan hệ hiệu quả
4.3.1.5. Một giải thuật tối ưu hóa một biểu thức đại số quan hệ trên lược đồ toàn cục
4.3.2. Bước 2- Định vị dữ liệu
4.3.2.1. Bước 2.1. Biến đổi biểu thức đại số quan hệ trên lược đồ toàn cục
4.3.2.2. Bước 2.2. Đơn giản hóa biểu thức đại số quan hệ trên lược đồ phân mảnh
4.3.2.3. Một giải thuật tối ưu hóa một biểu thức đại số quan hệ trên lược đồ phân mảnh
4.3.3. Bước 3- Tối ưu hóa truy vấn toàn cục
4.3.4. Bước 4- Tối ưu hóa truy vấn cục bộ

Mở đầu

Chương này trình bày về các bước thực hiện trong việc tối ưu hóa truy vấn trong cơ sở dữ liệu tập trung và trong cơ sở dữ liệu phân tán, các tiêu chuẩn tối ưu hóa nhằm để làm giảm thời gian thực hiện truy vấn, giảm vùng nhớ trung gian và chi phí truy vấn thông trong qúa trình thực hiện truy vấn, bộ suy diễn dùng trong việc đơn giản hóa biểu thức đại số quan hệ của truy vấn.

Chương này sử dụng một cơ sở dữ liệu sau đây để minh họa cho các nội dung được trình bày trong chương:

Sinhvien (masv, hoten, tuoi, 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), Tuổi (tuoi), 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.

4.1. Truy vấn. Biểu thức chuẩn tắc của truy vấn

4.1.1. 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.

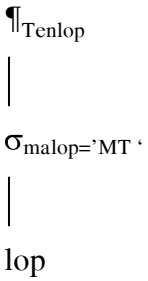
Một truy vấn có thể được dùng để xác định ngữ nghĩa của một ứng dụng, hoặc nó có thể được dùng để xác định công việc cần được thực hiện bởi một ứng dụng nhằm để truy xuất cơ sở dữ liệu.

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 :

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

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)*, là một cây mà một nút lá là một quan hệ trong cơ sở dữ liệu, và một nút khác lá (nút trung gian hoặc nút gốc) là một quan hệ trung gian được tạo ra bởi một phép toán đại số quan hệ. Chuỗi các phép toán đại số quan hệ được thực hiện từ các nút lá đến nút gốc để tạo ra kết quả truy vấn.

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



4.1.3. 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.

Tương tự, chúng ta có thể biến đổi một cây toán tử trên lược đồ toàn cục thành một cây toán tử trên lược đồ phân mảnh bằng cách thay thế các nút lá của cây đầu tiên bằng các biểu thức chuẩn tắc của chúng. Một điều quan trọng là bây giờ các nút lá của cây toán tử của biểu thức chuẩn tắc là các mảnh thay vì là các quan hệ toàn cục.

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

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

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

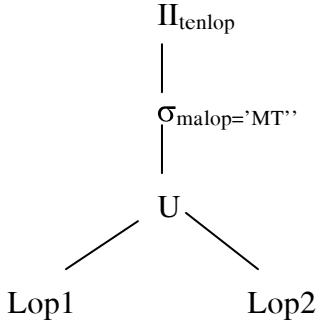
Biểu thức tái lập của quan hệ toàn cục lop là :

$Lop = lop1 \cup lop2$

Biểu thức chuẩn tắc của biểu thức truy vấn là :

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

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 :



4.2. Tối ưu hóa truy vấn trong cơ sở dữ liệu tập trung

Khi một hệ quản trị dữ liệu (DBMS) nhận một truy vấn viết bằng ngôn ngữ cao cấp, chẳng hạn SQL, DBMS thực hiện các bước sau đây:

4.2.1. Bước 1- Kiểm tra ngữ pháp (Syntax Checking)

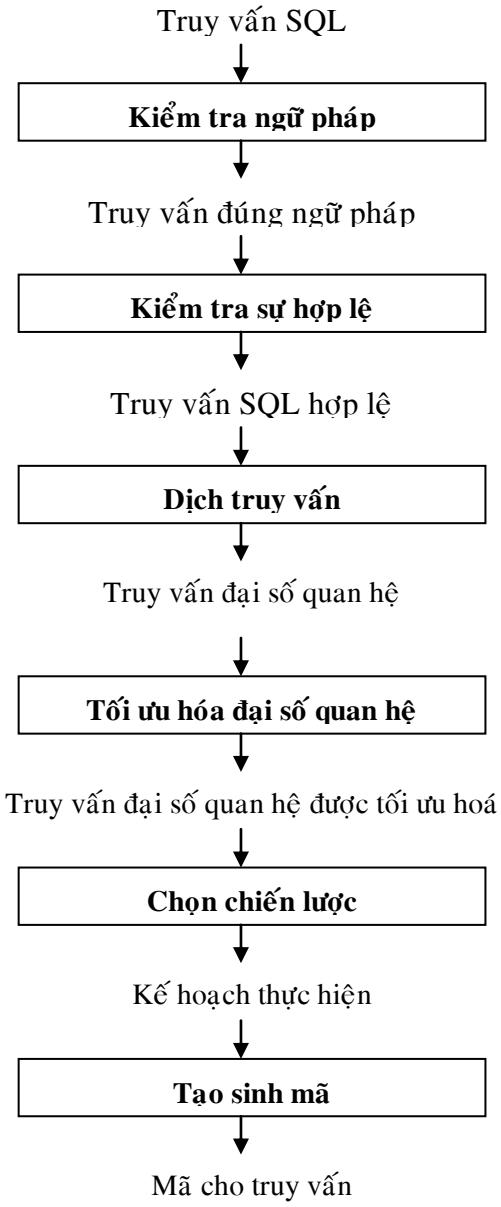
Trong bước này, DBMS sẽ kiểm tra ngữ pháp của truy vấn ban đầu (SQL query). Nếu truy vấn bị sai ngữ pháp thì DBMS sẽ thông báo truy vấn bị sai ngữ pháp và truy vấn này sẽ không được thực hiện. Nếu truy vấn đúng ngữ pháp (syntactically correct SQL query) thì DBMS sẽ tiếp tục thực hiện bước 2.

Ví dụ: Xét truy vấn Q1

Q1: SELECT masv,hoten FORM sinhvien;

Truy vấn này bị sai ngữ pháp (viết sai từ khóa FROM)

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



4.2.2. Bước 2- Kiểm tra sự hợp lệ (Validation)

Trong bước này, DBMS sẽ thực hiện các công việc:

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

Ví dụ : Xét truy vấn Q2

Q2: SELECT masv, hoten FROM sinh_vien ;

Truy vấn này có bảng sinh_vien không tồn tại trong cơ sở dữ liệu.

Ví dụ: Xét truy vấn Q3

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

Truy vấn này không hợp lệ vì có cột masv (thuộc kiểu dữ liệu number) so sánh với một hằng chuỗi '123'trong mệnh đề WHERE.

Nếu truy vấn chứa các đối tượng dữ liệu không tồn tại hoặc truy vấn của các đối tượng dữ liệu không phù hợp kiểu dữ liệu với nhau thì DBMS sẽ thông báo các đối tượng dữ liệu nào không tồn tại hoặc các đối tượng dữ liệu nào không phù hợp kiểu dữ liệu và truy vấn này sẽ không được thực hiện. Nếu các đối tượng dữ liệu này đều tồn tại trong cơ sở dữ liệu (truy vấn hợp lệ – valid SQL query) thì DBMS sẽ tiếp tục thực hiện bước 3.

4.2.3. Bước 3 – Dịch truy vấn (Translation)

Trong bước này, 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. Một trong các dạng biểu diễn bên trong này là việc sử dụng đại số quan hệ bởi vì các phép toán đại số quan hệ được biến đổi dễ dàng thành các tác vụ của hệ thống : truy vấn ban đầu được biến đổi thành một biểu thức đại số quan hệ hay còn gọi là truy vấn đại số quan hệ (relational algebra query)

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 :

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

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

Trong bước này 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 (theo nghĩa chúng có cùng một kết quả) nhưng biểu thức sau sẽ hiệu quả hơn: loại bỏ các phép toán không cần thiết và giảm vùng nhớ trung gian. Cuối bước này, DBMS tạo ra một truy vấn đại số quan hệ đã được tối ưu hoá (optimized relational algebra query).

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:

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

4.2.5. Bước 5- Chọn lựa chiến lược truy xuất (strategy selection)

Trong bước này, DBMS sử dụng các thông số về kích thước của các bảng, các chỉ mục vv... 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 phí của kế hoạch thực hiện 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. Cuối bước này, DBMS tạo ra một kế hoạch thực hiện cho truy vấn.

4.2.6. Bước 6- Tạo sinh mã (code generation)

Trong bước này, kế hoạch thực hiện của truy vấn có được ở cuối bước 5 sẽ được mã hoá và được thực hiện.

4.3. Tối ưu hóa truy vấn 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.

4.3.1. Bước 1- Phân rã truy vấn (Query Decomposition)

Bước này còn được gọi là bước **Tối ưu hóa truy vấn trên lược đồ toàn cục.** Bước này giống với các bước 1, 2, 3 và 4 của tối ưu hóa truy vấn trong cơ sở dữ liệu tập trung, nhằm để biến đổi một truy vấn viết bằng ngôn ngữ cấp cao, chẳng hạn SQL, thành một biểu thức đại số quan hệ tương đương (theo nghĩa chúng cho ra cùng một kết quả) và hiệu quả (theo nghĩa loại bỏ các phép toán đại số quan hệ

không cần thiết, giảm vùng nhớ trung gian). Bước này chưa đề cập đến sự phân tán dữ liệu.

Tối ưu hóa truy vấn trên lược đồ toàn cục bao gồm 4 bước sau:

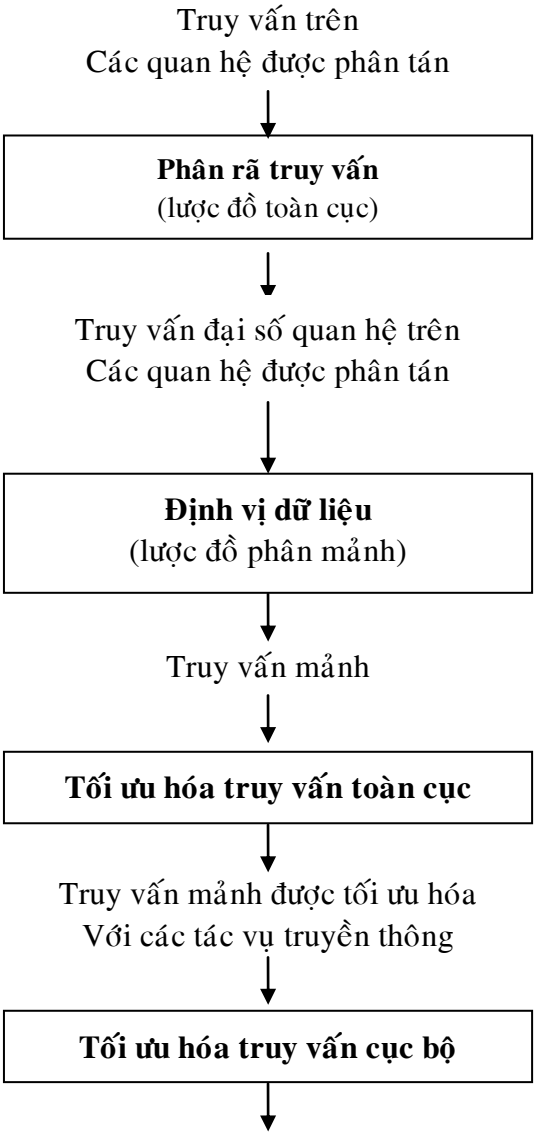
4.3.1.1. Bước 1.1- Phân tích truy vấn

Trong bước này, DBMS kiểm tra ngữ pháp của truy vấn, kiểm tra sự tồn tại của các đối tượng dữ liệu (tên cột, tên bảng, vv...) 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)

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



Truy vấn bị sai về kiểu dữ liệu

Một truy vấn bị sai về kiểu dữ liệu nếu các thuộc tính của nó hoặc các tên quan hệ không được định nghĩa trong lược đồ qoàn cục, hoặc nếu các phép toán được áp dụng cho các thộc tính bị sai về kiểu dữ liệu.

Để giải quyết cho vấn đề này, trong lược đồ tòan cục chúng ta phải mô tả kiểu dữ liệu của các thuộc tính của các quan hệ.

Ví dụ: Xét truy vấn Q5:

```
Q5: SELECT mssv, hoten     FROM sinhvien
     WHERE masv='123';
```

Truy vấn này có hai lỗi sai:

(1) mssv không tồn tại trong quan hệ sinhvien, và

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

Truy vấn bị sai về ngữ nghĩa

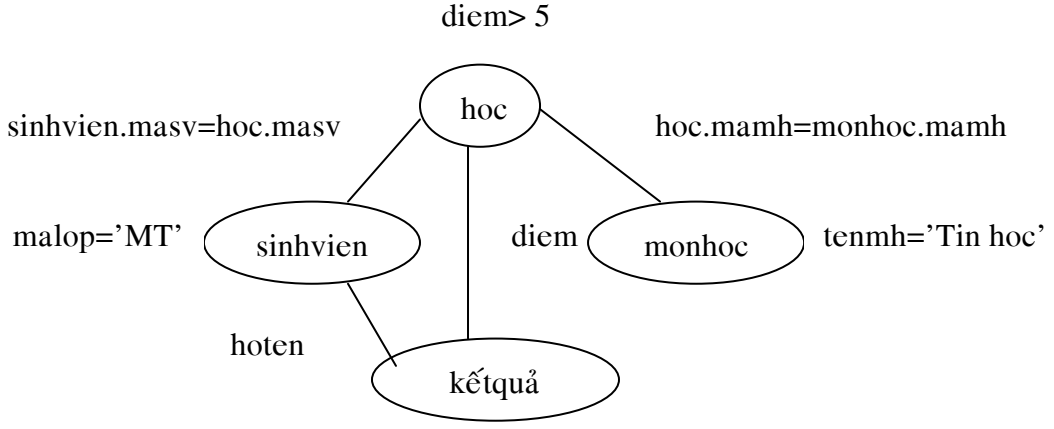
Một truy vấn bị sai về ngữ nghĩa nếu nó có chứa các thành phần không tham gia vào quá trình tạo ra kết quả của truy vấn.

Để phát hiện một truy vấn bị sai về ngữ nghĩa, chúng ta dùng một đồ thị truy vấn (query graph) hoặc đồ thị kết nối quan hệ (relation connection graph) cho các truy vấn có chứa các phép chọn, phép chiếu và phép kết. Trong một đồ thị truy vấn, một nút biểu diễn cho một quan hệ kết quả (result relation) và các nút khác biểu diễn cho các quan hệ toán hạng (operand relation). Một cạnh giữa hai nút quan hệ toán hạng biểu diễn cho một phép kết, một cạnh giữa một nút quan hệ toán hạng với một nút quan hệ kết quả biểu diễn cho một phép chiếu. Một nút quan hệ toán hạng có thể chứa một điều kiện chọn. Một đồ thị con quan trọng của đồ thị này là đồ thị kết quả (join graph) được dùng trong bước tối ưu hóa truy vấn.

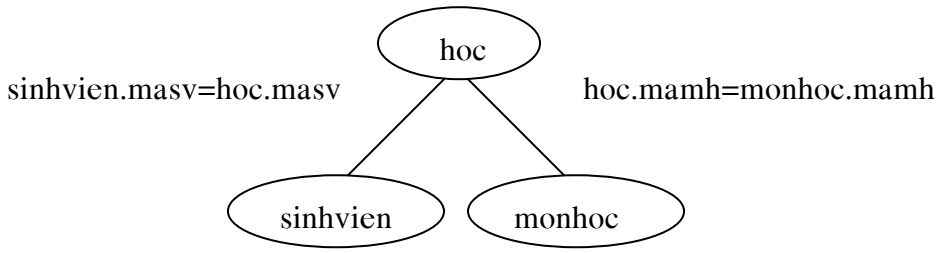
Ví dụ: Xét truy vấn Q6 liệt kê họ tên sinh viên và điểm của môn học 'Tin học' của lớp mã 'MT' với điều kiện đạt điểm trên 5.

```
Q6:  SELECT hoten, diem
     FROM sinhvien, hoc, monhoc
     WHERE sinhvien.masv=hoc.masv
          AND hoc.mamh=monhoc.mamh
          AND malop='MT'   AND diem > 5   AND tenmh = 'Tin hoc';
```

Đồ thị truy vấn của truy vấn này như sau:



Và đồ thị kết nối tương ứng là:

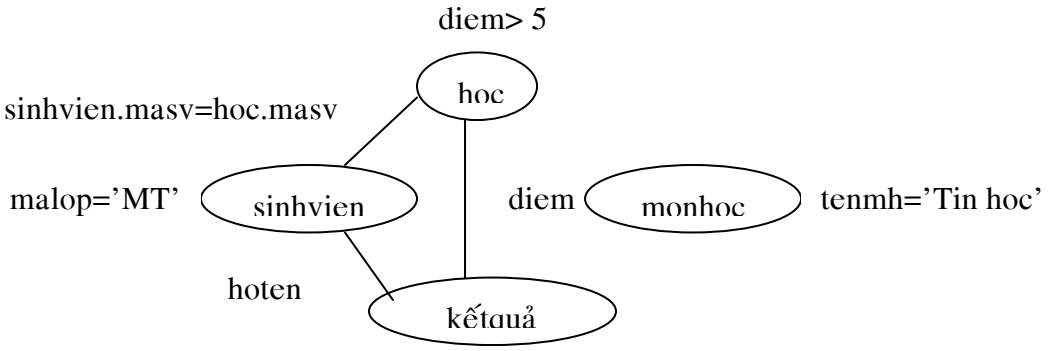


Một truy vấn bị sai về ngữ nghĩa nếu đồ thị truy vấn của nó là không liên thông. Đồ thị không liên thông là một đồ thị bao gồm nhiều thành phần liên thông, mỗi thành phần liên thông là một đồ thị con riêng biệt, hai thành phần liên thông không dược nối với nhau thông qua các cạnh. Trong trường hợp này, một truy vấn được xem là đúng đắn bằng cách chỉ giữ lại thành phần có liên quan đến quan hệ kết quả và loại bỏ các thành phần còn lại.

Ví dụ: Xét truy vấn Q7

```
Q7:  SELECT hoten, diem
     FROM sinhvien, hoc, monhoc
     WHERE sinhvien.masv=hoc.masv
          AND malop='MT'    AND diem > 5    AND tenmh = 'Tin hoc';
```

Đồ thị truy vấn của truy vấn này như sau:



Đồ thị truy vấn của truy vấn này là không liên thông, nên truy vấn bị sai về ngữ nghĩa. Có ba giải pháp cho vấn đề này là:

(1) Hủy bỏ truy vấn này.
(2) Hủy bỏ các bảng không cần thiết trong mệnh đề From và các điều kiện có liên quan đến các bảng này trong mệnh đề WHERE.

Giả sử truy xuất đến monhoc là không cần thiết, ta hủy bỏ bảng monhoc trong mệnh đề From và điều kiện tenmh ='Tin hoc' trong mệnh đề WHERE. Ta có truy vấn Q8 như sau:

Q8: SELECT hoten, diem
FROM sinhvien, hoc
WHERE sinhvien.masv =hoc.masv AND malop ='MT' AND diem > 5;

(3) Bổ sung điều kiện kết sao cho đồ thị truy vấn được liên thông. Một đồ thị truy vấn có thể không bị sai ngữ nghĩa nếu đồ thị này là một dồ thị đơn (có nhiều nhất một cạnh nối giữa hai đỉnh), liên thông và số cạnh bằng số đỉnh trừ 1.

Bổ sung điều kiện kết hoc.mamh = monhoc.mamh vào trong mệnh đề WHERE.
Ta có truy vấn Q9:

Q9: SELECT hoten, diem
FROM sinhvien, hoc
WHERE sinhvien.masv = hoc.masv
AND hoc.mamh =monhoc.mamh AND malop ='MT' AND diem > 5
AND tenmh ='Tin hoc';

4.3.1.2. 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 này nhằm mục đích chuẩn hóa điều kiện của mệnh đề Where về một trong hai dạng chuẩn:

- Dạng chuẩn giao (conjunctive normal form)
  $(P_{11} \text{ v } P_{12} \text{ v...v } P_{1n}) \text{ ^...^ } (P_{m1} \text{ v } P_{m2} \text{ v...v } P_{mn})$

- Dạng chuẩn hợp (disjunctive normal form)
  $(P_{11} \text{ ^ } P_{12} \text{ ^...^ } P_{1n}) \text{ v...v } (P_{m1} \text{ ^ } P_{m2} \text{ ^...^ } P_{mn})$

trong đó P_{ij} là một biến luận lý (có giá trị là true hoặc false) hoặc là một vị từ đơn giản (simple predicate) có dạng:

a **R** b

với a,b là các biểu thức số học và **R** là một trong những phép toán so sánh:

=	bằng
< > hoặc !=	không bằng
<	nhỏ hơn
<=	nhỏ hơn hoặc bằng
>	lớn hơn
>=	lớn hơn hoặc bằng

Để biến đổi điều kiện của mệnh đề WHERE về một trong hai dạng chuẩn trên, chúng ta sử dụng các phép biến đổi tương đương của các phép toán luận lý.
Ký hiệu ≡ là sự tương đương.

Các phép biến đổi 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) $\neg(\neg P) \equiv P$

Ví dụ: Xét truy vấn Q10

```
Q10:  SELECT malop
      FROM sinhvien
      WHERE (  NOT (malop='MT1')
                  AND (malop='MT1' OR malop='MT2')
                 AND NOT (malop='MT2')  )     OR hoten='Nam';
```

Điều kiện q của mệnh đề WHERE là:

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

Ký hiệu:

P_1 là malop='MT1'
P_2 là malop='MT2'
P_3 là hoten='Nam'

Điều kiện q sẽ là:

$(\neg P_1 \wedge (P_1 \vee P_2) \wedge \neg P_2) \vee P_3$

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

$(((\neg P_1 \wedge P_1) \vee (\neg P_1 \wedge P_2)) \wedge \neg P_2) \vee P_3$
$(\neg P_1 \wedge P_2 \wedge \neg P_2) \vee P_3 = P_3$

4.3.1.3. 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.

Các phép biến đổi tương đương gồm có :

(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) $P_1 \vee (P_1 \wedge P_2) \equiv P_1$

Vídụ: Xét truy vấn Q10 ở trên, điều kiện q ở dạng chuẩn hợp là:

$(\neg P_1 \wedge P_1 \wedge \neg P_2) \vee (\neg P_1 \wedge P_2 \wedge \neg P_2) \vee P_3$

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

$(false \wedge \neg P_2) \vee (\neg P_1 \wedge false) \vee P_3$

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

$False \vee False \vee P_3$

Áp dụng phép biến đổi (15), chúng ta được diều kiện q cuối cùng là P_3, tức là hoten='Nam'. Vậy truy vấn Q10 trở thành truy vấn Q11 như sau:

```
Q11:  SELECT malop
      FROM sinhvien
      WHERE hoten='Nam';
```

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

Bước này sử dụng các phép biến đổi tương đương của các phép toán đại số quan hệ nhằm để loại bỏ các phép toán đại số quan hệ không cần thiết và giảm vùng nhớ trung gian được sử dụng trong quá trình thực hiện các phép toán đại số quan hệ cần thiết cho truy vấn.

Bước này bao gồm hai bước sau đây:

Bước 1.4.1 – 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.

Bước 1.4.2 – Đơn giản hóa cây toán tử để có được một biểu thức đại số quan hệ hiệu quả.

Bước 1.4.1. 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.

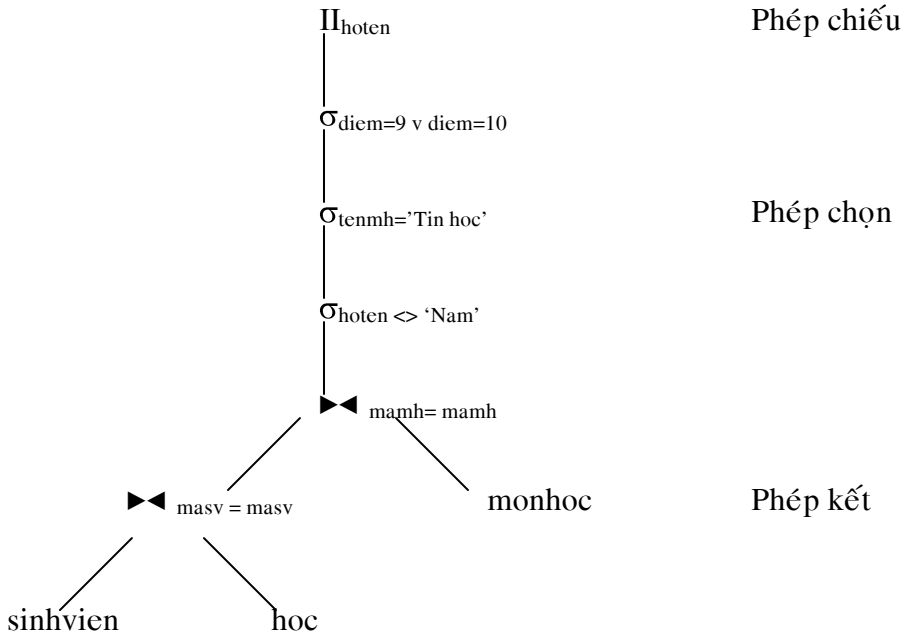
(3) Điều kiện ghi trong mệnh đề WHERE được biến đổi thành một chuỗi thích hợp các phép toán đại số quan hệ (phép chọn, phép kết, phép hợp...) đi từ các nút lá đến nút gốc. Chuỗi các phép toán này có thể được cho trực tiếp bởi thứ tự của các vị từ đơn giản và các phép toán luận lý.

Một cây toán tử tương ứng với một biểu thức đại số quan hệ.

Ví dụ: Xét truy vấn Q12 cho biết họ tên của các sinh viên không phải là 'Nam' học môn học 'Tin hoc' đạt điểm 9 hoặc 10.

```
Q12:  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);
```

Truy vấn này có thể được biểu diễn thành một cây toán tử, các vị từ đơn giản được biến đổi theo thứ tự xuất hiện tương ứng với các phép kết rồi đến các phép chọn.



Biểu thức đại số quan hệ tương ứng là:

Π_{hoten} (σ (diem=9 v diem=10) ^ tenmh= 'Tin hoc' ^ hoten <> 'Nam'
((sinhvien ▶◀ $_{masv=masv}$ hoc) ▶◀ $_{mamh=mamh}$ monhoc))

Bước 1.4.2. 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.

Trong bước đơn giản hoá cây toán tử, *một điều quan trọng trong việc áp dụng các phép biến đổi tương đương cho một biểu thức truy vấn là việc phát hiện các biểu thức con chung (common subexpression) có trong biểu thức truy vấn*, nghĩa là các biểu thức con xuất hiện nhiều lần trong biểu thức truy vấn. Điều này có ý nghĩa là tiết kiệm thời gian thực hiện truy vấn vì các biểu thức con này chỉ được định trị duy nhất một lần. Một phương pháp để nhận biết chúng là ở chỗ việc biến đổi cây toán tử tương ứng thành một đồ thị toán tử bằng cách trước tiên gộp các nút lá giống nhau của cây (nghĩa là các quan hệ giống nhau), và sau đó gộp các nút trung gian khác của cây tương ứng với cùng các phép toán và có cùng các toán hạng.

Khi các biểu thức con đã được xác định, chúng ta có sử dụng các phép biến đổi tương đương sau đây để đơn giản hóa một cây toán tử:

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

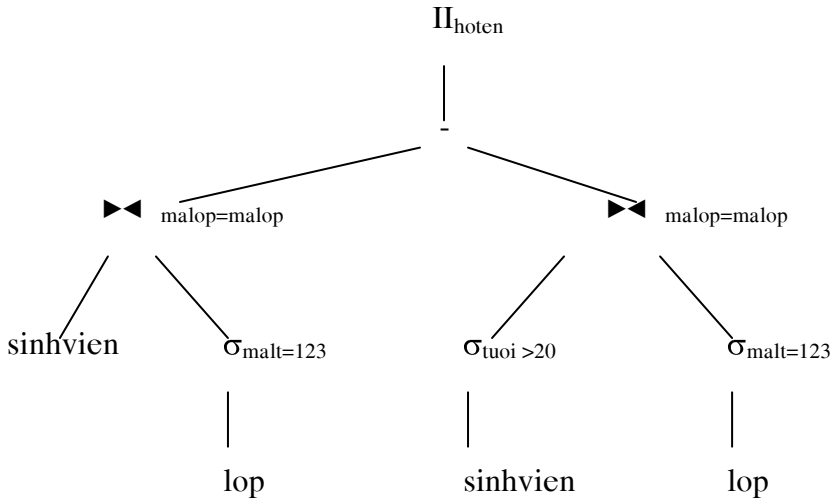
Ý nghĩa của các phép biến đổi này là loại bỏ các phép toán dư thừa.

Ví dụ: Xét truy vấn Q13 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:

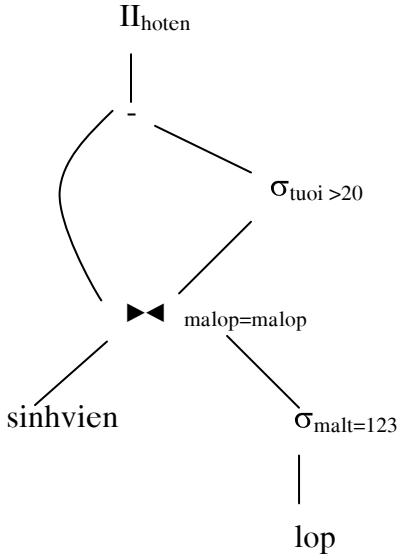
Π_{hoten} ((sinhvien ▶◀ $_{malop=malop}$ $\sigma_{malt=123}$ (lop)) –

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

Cây toán tử tương ứng :



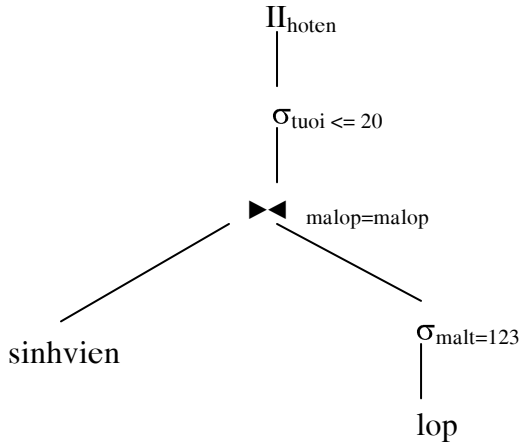
Để phát hiện ra biểu thức con chung, chúng ta bắt đầu bằng cách gộp các nút lá tương ứng với các quan hệ sinhvien và lop. Sau đó chúng ta đặt thừa số là phép chọ trên tuoi đối với phép kết (trong cách làm này, chúng ta di chuyển phép chọn lên phía trên). Bây giờ chúng ta có thể trộn các nút tương ứng với phép chọn trên malt và cuối cùng các nút tương ứng với phép kết, chúng ta được cây toán tử sau:



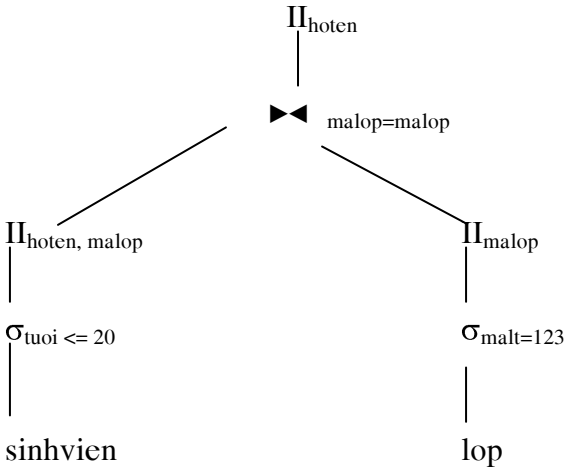
Áp dụng phép biến đổi tương đương (6) với R là biểu thức :

$sinhvien \bowtie_{malop=malop} \sigma_{malt=123} lop$

chúng ta được cây toán tử sau:



Sau đó áp dụng tính phân phối của phép chiếu và phép chọn đối với phép kết, ta được cây toán tử :



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

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

Đơn giản hoá một biểu thức đại số quan hệ được thực hiện dựa trên các tiêu chuẩn sau đây :

Tiêu chuẩn 1. Dùng tính idempotence (tương đương) của phép chọn và phép chiếu để tạo ra các phép chọn và phép chiếu thích hợp cho mỗi quan hệ toán hạng.

Tiêu chuẩn 2. Thực hiện các phép chọn và các phép chiếu càng sớm càng tốt, tức là đẩy các phép chọn và các phép chiếu xuống phía dưới cây càng xa càng tốt.

Tiêu chuẩn 3. Khi các phép chọn được thực hiện sau một phép tích thì kết hợp các phép toán này để tạo thành một phép kết.

Tiêu chuẩn 4. Kết hợp chuỗi các phép toán một ngôi liên tiếp nhau áp dụng cho một quan hệ toán hạng. Một chuỗi các phép chọn liên tiếp nhau (hoặc một chuỗi các phép liên kết liên tiếp nhau) có thể được kết hợp thành một phép chọn (hoặc một phép kết).

Tiêu chuẩn 5. Khi phát hiện các biểu thức con chung trong biểu thức truy vấn, áp dụng các phép biến đổi tương đương để đơn giản hoá biểu thức truy vấn.

4.3.1.5. Một giải thuật tối ưu hóa một biểu thức đại số quan hệ trên lược đồ toàn cục

Vào: Một biểu thức đại số quan hệ trên lược đồ toàn cục
Ra: Một biểu thức đại số quan hệ đã được tối ưu hóa

Giải thuật tối ưu hoá một biểu thức đại số quan hệ trên lược đồ toàn cục bao gồm các bước sau đây:

Bước 1. Phát hiện các biểu thức con chung có trong cây toán tử, biến đổi cây toán tử dựa trên biểu thức con chung

Bước 2. Thực hiện phép chọn càng sớm càng tốt. Sử dụng tính idempotence của phép chọn, tính giao hoán của phép chọn với phép chiếu, và tính phân phối của phép chọn đối với phép hợp, phép giao, phép hiệu, phép kết và phép tích để di chuyển phép chọn càng xuống phía dưới cây càng tốt.

Sử dụng các phép biến đổi tương đương:

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

$\Pi_X(\sigma_F(R)) \leftrightarrow \sigma_F(\Pi_X(R))$
$(\rightarrow$ nếu $Attr(F) \subseteq X)$
$\Pi_X(\sigma_F(R)) \equiv \Pi_X(\sigma_F(\Pi_{X \cup Attr(F)}(R)))$
$\sigma_F(R \cup S) \equiv \sigma_F(R) \cup \sigma_F(S)$
$\sigma_F(R \cap S) \equiv \sigma_F(R) \cap \sigma_F(S)$
$\sigma_{F1 \wedge F2}(R \cap S) \leftrightarrow \sigma_{F1}(R) \cap \sigma_{F2}(S)$
$(\rightarrow$ nếu $Attr(F1) \subseteq Attr(R)$ và $Attr(F2) \subseteq Attr(S))$

$\sigma_F (R - S) \equiv \sigma_F (R) - \sigma_F (S)$

$\sigma_F (R \blacktriangleright\blacktriangleleft_{F1} S) \leftrightarrow \sigma_F (R) \blacktriangleright\blacktriangleleft_{F1} S$
($\rightarrow$ nếu $Attr(F) \subseteq Attr(R)$)

$\sigma_{F1 \wedge F2} (R \blacktriangleright\blacktriangleleft_{F3} S) \leftrightarrow \sigma_{F1} (R) \blacktriangleright\blacktriangleleft_{F3} \sigma_{F2} (S)$
($\rightarrow$ nếu $Attr(F1) \subseteq Attr(R)$ và $Attr(F2) \subseteq Attr(S)$)

$\sigma_F (R \blacktriangleright\blacktriangleleft_{F3} S) \leftrightarrow \sigma_{F2} (\sigma_{F1} (R) \blacktriangleright\blacktriangleleft_{F3} S)$
($\rightarrow$ nếu $F=F1 \wedge F2$ và $Attr(F1) \subseteq Attr(R)$ và $Attr(F2) \subseteq Attr(R) \cup Attr(S)$)

$\sigma_F (R \times S) \leftrightarrow \sigma_F (R) \times S$
($\rightarrow$ nếu $Attr(F) \subseteq Attr(R)$)

$\sigma_{F1 \wedge F2} (R \times S) \leftrightarrow \sigma_{F1} (R) \times \sigma_{F2} (S)$
($\rightarrow$ nếu $Attr(F1) \subseteq Attr(R)$ và $Attr(F2) \subseteq Attr(S)$)

$\sigma_F (R \times S) \leftrightarrow \sigma_{F2} (\sigma_{F1} (R) \times S)$
($\rightarrow$ nếu $F=F1 \wedge F2$ và $Attr(F1) \subseteq Attr(R)$ và $Attr(F2) \subseteq Attr(R) \cup Attr(S)$)

Bước 3. Thực hiện phép chiếu càng sớm càng tốt. Sử dụng tính idempotence của phép chiếu, tính phân phối của phép chiếu đối với phép hợp, phép kết và phép tích để di chuyển phép chiếu càng xuống phía dưới cây càng tốt. Kiểm tra tất cả các phép chiếu là cần thiết, loại bỏ phép chiếu không cần thiết nếu phép này chiếu trên tất cả các thuộc tính của quan hệ toán hạng.

Sử dụng phép biến đổi:

$\Pi_{X1} (\Pi_{X2} (R)) \equiv \Pi_{X1} (R)$ với $X1 \subseteq X2$

$\Pi_X (R \cup S) \equiv \Pi_X (R) \cup \Pi_X (S)$

$\Pi_X (R \blacktriangleright\blacktriangleleft_F S) \leftrightarrow \Pi_X (R) \blacktriangleright\blacktriangleleft_F (S)$
($\rightarrow$ nếu $Attr(F_R) \subseteq X$ và $X \subseteq Attr(R)$)

$\Pi_{X1 \cup X2} (R \blacktriangleright\blacktriangleleft_F S) \leftrightarrow \Pi_{X1} (R) \blacktriangleright\blacktriangleleft_F \Pi_{X2} (S)$
($\rightarrow$ nếu $Attr(F) \subseteq X1 \cup X2$ và $X1 \subseteq Attr(R)$ và $X2 \subseteq Attr(S)$)

$\Pi_{X1 \cup X2} (R \times S) \leftrightarrow \Pi_{X1} (R) \times \Pi_{X2} (S)$
($\rightarrow$ nếu $X1 \subseteq Attr(R)$ và $X2 \subseteq Attr(S)$)

Bước 4. Nếu một phép chọn được thực hiện ngay sau một phép tích, mà phép chọn bao gồm các thuộc tính của các quan hệ trong phép tích, thì biến đổi phép tích thành phép kết. Nếu phép chọn chỉ bao gồm các thuộc tính của một quan hệ trong phép tích, thì thực hiện phép chọn cho quan hệ này trước khi thực hiện phép tích.

Sử dụng các phép biến đổi:

$\sigma_F (R \times S) \leftrightarrow \sigma_F (R) \times S$
($\rightarrow$ nếu $Attr(F) \subseteq Attr(R)$)

$\sigma_{F1 \wedge F2}(R \times S) \leftrightarrow \sigma_{F1}(R) \times \sigma_{F2}(S)$
($\rightarrow$ nếu $Attr(F1) \subseteq Attr(R)$ và $Attr(F2) \subseteq Attr(S)$)

$\sigma_{F}(R \times S) \leftrightarrow \sigma_{F2}(\sigma_{F1}(R) \times S)$
($\rightarrow$ nếu $F=F1 \wedge F2$ và $Attr(F1) \subseteq Attr(R)$ và $Attr(F2) \subseteq Attr(R) \cup Attr(S)$)

Bước 5. Nếu có một chuỗi các phép chọn và/ hoặc các phép chiếu, sử dụng tính giao hoán hoặc tính idempotence để kết hợp chúng thành một phép chọn, một phép chiếu hoặc một phép chọn đi trước một phép chiếu và áp dụng chúng cho mỗi bộ của quan hệ toán hạng. Nếu một phép kết hoặc phép tích đi trước một chuỗi các phép chọn hoặc các phép chiếu, thì áp dụng chúng cho mỗi bộ của phép kết hoặc phép chiếu ngay khi tạo ra kết quả.

Bước 6. Sử dụng tính kết hợp của phép giao, phép tích và phép kết để sắp xếp lại các quan hệ trong cây toán tử, sao cho phép toán nào mà nó tạo ra kết quả ít nhất sẽ được thực hiện trước tiên.

Sử dụng các phép biến đổi:

$(R \cap S) \cap T \equiv (R \cap T) \cap S$
$(R \times S) \times T \equiv (R \times T) \times S$
$(R \bowtie_{F1} S) \bowtie_{F2} T \leftrightarrow (R \bowtie_{F2} T) \bowtie_{F1} S$
($\rightarrow$ nếu $Attr(F2) \subseteq Attr(R) \cup Attr(T)$)
($\leftarrow$ nếu $Attr(F1) \subseteq Attr(R) \cup Attr(S)$)

4.3.2. 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 2.1. Biến đổi biểu thức đại số quan hệ trên lược đồ toàn cục (chứa các quan hệ toàn cục) thành biểu thức đại số quan hệ trên lược đồ phân mảnh (chứa các mảnh của quan hệ toàn cục) bằng cách thay thế các quan hệ toàn cục bởi biểu thức tái lập của chúng.

Bước 2.2. Đơn giản hoá biểu thức đại số quan hệ trên lược đồ phân mảnh để có được một biểu thức hiệu qủa (loại bỏ các phép toán không cần thiết giảm vùng nhớ trung gian) bằng cách sử dụng các phép biến đổi tương đương của đại số quan hệ và các đại số quan hệ được tuyển chọn.

4.3.2.1. Bước 2.1 – Biến đổi biểu thức đại số quan hệ trên lược đồ toàn cục

Bước này sẽ biến đổi biểu thức đại số quan hệ trên lược đồ toàn cục (chứa các quan hệ toàn cuc) thành biểu thức đại số quan hệ trên lược đồ phân mảnh (chứa các mảnh của quan hệ toàn cục) bằng cach thay thế mỗi quan hệ toàn cục trong cây toán tử bởi biểu thức tái lập của nó. Biểu thức tái lập của một quan hệ toàn cục là một biểu thức đại số quan hệ bao gồm các mảnh của quan hệ này mà biểu thức này cho phép tạo lại quan hệ toàn cục này. Biểu thức tái lập cũng được biểu diễn bằng một cây toán tử.

Xét lược đồ quan hệ sinhvien và lop sau đây:
Sinhvien (masv, hoten, tuoi, malop)
Lop (malop, tenlop, malt, tenkhoa)

Giả sử chúng ta 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)

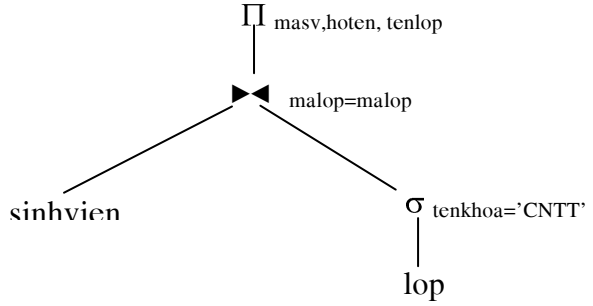
Các biểu thức tái lập của quan hệ *lop* và *sinhvien* là:

Lop = Lop1 U Lop2
Sinhvien= sinhvien1 U sinhvien2

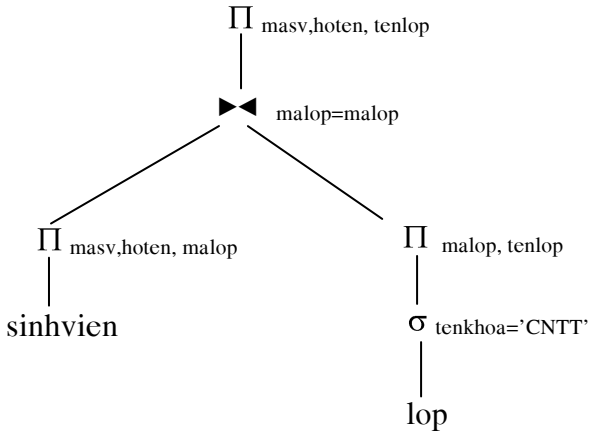
Trong đó:

Lop1 = $\sigma_{tenkhoa = 'CNTT'}$ (lop)
Lop2 = $\sigma_{tenkhoa = 'DIEN'}$ (lop)
Sinhvien1 = sinhvien ►< (Lop1)
Sinhvien2 = sinhvien ►< (Lop2)

Ví dụ: Xét cây toán tử



Áp dụng tính idempotence của phép chiếu, chúng ta được:

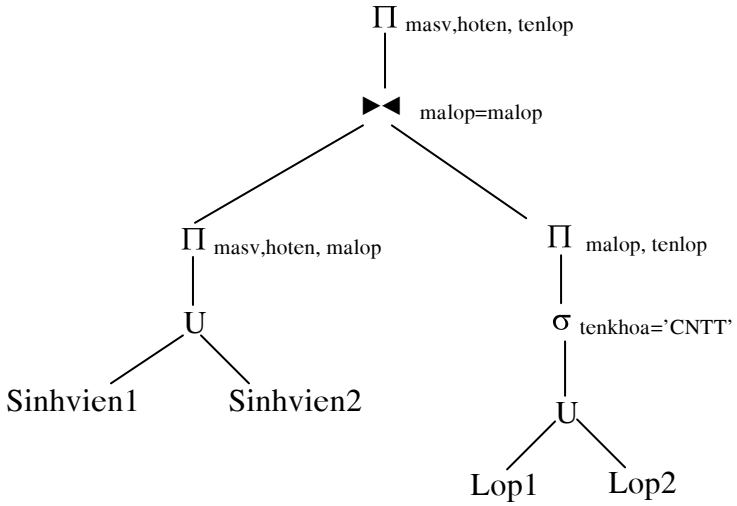


Thay thế *sinhvien* và *lop* bởi biểu thức tái lập:

Sinhvien = sinhvien1 U sinhvien2

Lop = lop1 U lop2

Ta được cây toán tử sau:



4.3.2.2 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)* bằng cách sử dụng các phép biến đổi tương đương của đại số quan hệ và của đại số quan hệ được tuyển chọn.

Các phép biến đổi tương đương (áp dụng cho các quan hệ và các quan hệ được tuyển chọn) gồm có:

(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) $\varnothing \blacktriangleright< R \equiv \varnothing$

Đơn giản hoá một biểu thức đại số quan hệ trên lược đồ phân mảnh được thực hiện dựa trên các tiêu chuẩn sau:

Tiêu chuẩn 6: Di chuyển các phép chọn xuống các nút lá của cây, và sau đó áp dụng chúng bằng cách dùng đại số quan hệ được tuyển chọn; thay thế các kết quả chọn lựa bởi quan hệ rỗng nếu điều kiện chọn của kết quả bị mâu thuẫn.

Tiêu chuẩn 7: Để phân phối các phép kết xuất hiện trong một truy vấn toàn cục, các phép hợp (biểu diễn tập hợp của các phân mảnh) phải được di chuyển lên phía trên các phép kết mà chúng ta muốn phân phối để loại bỏ các phép kết không cần thiết.

Tiêu chuẩn 8: Dùng đại số quan hệ được tuyển chọn để định trị điều kiện chọn của các toán hạng của các phép kết; thay thế cây con, bao gồm phép kết và các toán hạng của nó, bằng quan hệ rỗng nếu điều kiện chọn của kết quả của phép kết bị mâu thuẫn.

Ví dụ : Xét cây toán tử trên lược đồ phân mảnh trên
Đẩy phép chọn và phép chiếu xuống khỏi phép hợp ta được:

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

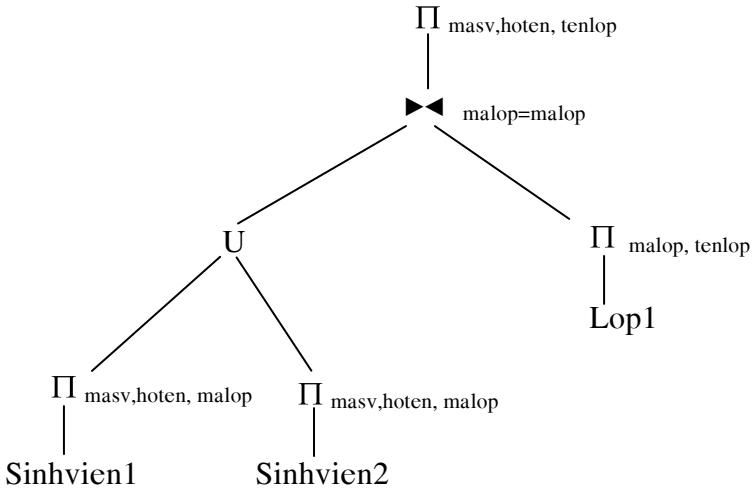
Ta nhận thấy kết quả của phép chọn $\sigma_{tenkhoa='CNTT'}(lop2)$ là rỗng và phép chọn $\sigma_{tenkhoa='CNTT'}(lop1)$ là không cần thiết vì điều kiện chọn của lop1 là tenkhoa= 'CNTT'.
Do đó:

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

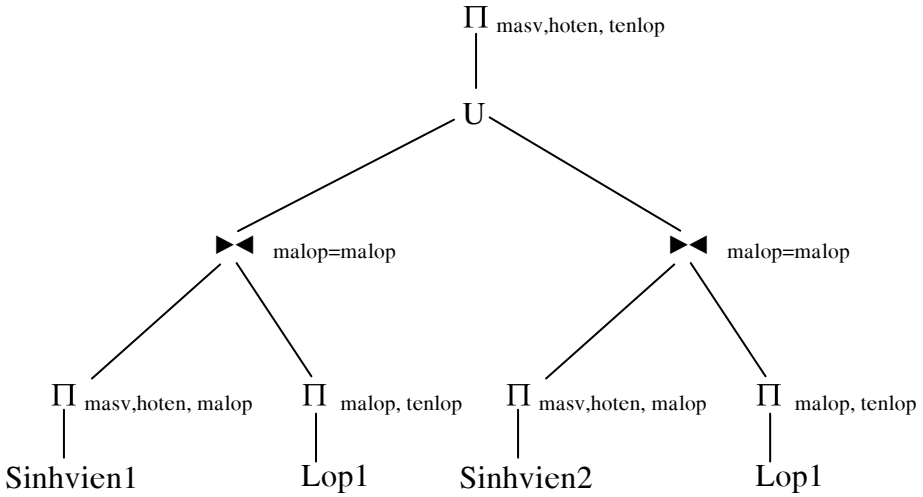
Đẩy phép chiếu xuống khỏi phép hợp trong biểu thức:

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

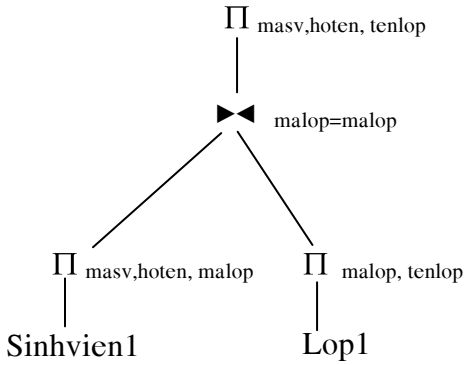
Ta có cây toán tử:



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



Tuy nhiên phếp kết giữa sinhviên2 và lop1 là rỗng do điều kiện chọn của phân mảnh lop1 và sinhvien2 mâu thuẫn nhau.
Cuối cùng ta có cây toán tử trên lược đồ phân mảnh như sau:



Đơn giản hoá biểu thức đại số quan hệ trong lược đồ phân mảnh còn dựa vào một hệ suy diễn được gọi là Bộ chứng minh định lý (Theorem Prover).

Ví dụ: Giả sử chúng ta chỉ có hai khoa là 'CNTT','DIEN' và có tối đa 20 lớp, các lớp có mã lớp từ 1 đến 10 thuộc khoa 'CNTT' và các lớp có mã từ 11 đến 20 thuộc khoa 'DIEN'. Từ đó, chúng ta có các luật suy diễn sau:

Malop > 10 → tenkhoa = 'DIEN'
Malop <= 10 → NOT (Malop >10)
Malop > 10 → NOT (Malop <=10)
tenkhoa = 'CNTT' → Malop <= 10
tenkhoa = 'DIEN' → Malop > 10
tenkhoa = 'CNTT' → not(tenkhoa = 'DIEN')
tenkhoa = 'DIEN' → not(tenkhoa = 'CNTT')

Xét truy vấn Q14 cho biết tên lớp của lớp có mã lớp bằng 1:

```
Q14: Select tenlop
     From lop
     Where malop = 1
```

Trước khi thực hiện truy vấn này, chúng ta có các suy diễn sau đây:

Malop = 1 → malop <=10
Malop <= 10 → tenkhoa= 'CNTT'

Do đó truy vấn này chỉ liên quan đến lop1 vì điều kiện chọn của lop1 là tenkhoa = 'CNTT'. Vì thế biểu thức đại số quan hệ của truy vấn này là:

$\Pi_{tenlop}(\sigma_{malop=1}(lop1))$

4.3.3 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).

Một chiến lược được đặc trưng bởi *thứ tự thực hiện các phép toán đại số quan hệ và các tác vụ truyền thông cơ bản (gởi/nhận)* dùng để truyền dữ liệu giữa các vị trí. Bằng các hoán đổi thứ tự của các phép toán trong biểu thức truy vấn phân mảnh, ta có thể có được nhiều truy vấn tương đương.

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. Đặc biệt khâu tốn kém thời gian trong cơ sở dữ liệu phân tán là khâu truyền dữ liệu do tốc độ và băng thông giới hạn.

Trong trường hợp nhân bản thì còn phải tính xem nhân bản nào được sử dụng nhằm giảm chi phí truyền thông.

Một khía cạnh quan trọng của tối ưu hoá truy vấn là *thứ tự thực hiện các phép kết phân tán.* Nhờ tính giao hoán của các phép kết, chúng ta có thể làm giảm chi phí thực hiện các phép kết này. Một kỹ thuật cơ bản để tối ưu hoá một chuỗi các phép kết phân tán là sử dụng phép nửa kết nhằm làm giảm chi phí truyền thông giữa các vị trí và tăng tính xử lý cục bộ tại các vị trí.

$$R \blacktriangleright\!\blacktriangleleft_{A=B} S = S \blacktriangleright\!\blacktriangleleft_{A=B} (R \blacktriangleright\!<_{A=B} \Pi_B S)$$

Ví dụ: Giả sư có sự phân tán dữ liệu sau:

- mảnh *sinhvien1* đặt tại vị trí 1 và
- mảnh *lop1* đặt tại vị trí 2

Chúng ta cần thực hiện phép kết phân tán sau:

$$Sinhvien1 \blacktriangleright\!\blacktriangleleft lop1$$

Bằng cách áp dụng phép nửa kết biểu thức trên tương đương với:

$$Lop1 \blacktriangleright\!\blacktriangleleft (sinhvien1 \blacktriangleright\!< \Pi_{malop}(lop1))$$

Do đó ta có một chiến lược thực hiện cho phép kết phân tán này với các tác vụ truyền thông sau:

1) Thực hiện $T_1 = \Pi_{malop}(lop1)$ cục bộ tại vị trí 2.
2) Truyền T_1 từ vị trí 2 qua vị trí 1.
3) Thực hiện $T_2 = sinhvien1 \blacktriangleright\!< T_1$ cục bộ tại vị trí 1.
4) Truyền T_2 từ vị trí 1 qua vị trí 2.
5) Thực hiện $T_3 = lop1 \blacktriangleright\!< T_2$ cục bộ tại vị trí 2.
6) Truyền T_3 từ vị trí 2 qua vị trí của ứng dụng cần thực hiện của phép kết này.

4.3.4 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.