

**PHƯƠNG PHÁP TỐI ƯU HOÁ TRUY VẤN ĐỐI TƯỢNG BẰNG CÁC PHÉP
BIẾN ĐỔI BIỂU THỨC ĐẠI SỐ ĐỐI TƯỢNG OQL**

Lê Mạnh Thạnh, Đại học Huế

Hoàng Bảo Hùng

Sở Thông tin và Truyền thông tỉnh Thừa Thiên Huế

TÓM TẮT

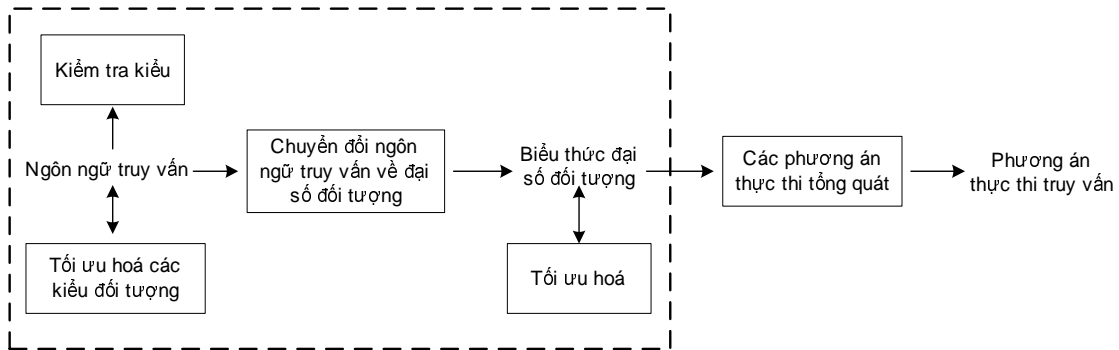
Tối ưu hóa truy vấn là vấn đề được quan tâm nghiên cứu về lý thuyết cơ sở dữ liệu. Những kết quả khá trọn vẹn về tối ưu hóa truy vấn trên mô hình cơ sở dữ liệu quan hệ đã là lời giải cho nhiều lớp bài toán quản lý nói chung và lý thuyết cơ sở dữ liệu nói riêng. Từ cơ sở đó, việc nghiên cứu mở rộng các phương pháp tối ưu hóa truy vấn quan hệ trên mô hình cơ sở dữ liệu hướng đối tượng là phương pháp được đề xuất trong bài báo này. Điều khác biệt so với mô hình quan hệ là việc tối ưu hóa truy vấn đối tượng dựa trên tập luật – biến đổi biểu thức truy vấn bằng các phép biến đổi đại số đối tượng, được thực hiện qua việc chuyển đổi tương đương giữa truy vấn viết bằng OQL (Object Query Language) và đại số đối tượng tương ứng, kết quả giữa truy vấn viết bằng OQL và biểu thức đại số đối tượng tương ứng là tương đương. Bài báo đưa ra các quy tắc tổng quát để tối ưu hóa truy vấn đối tượng, trên cơ sở đó đề xuất thuật toán tối ưu hóa truy vấn đối tượng dựa trên tập luật.

I. Giới thiệu

Khi thực thi một truy vấn, chúng ta có nhiều phương án mà hệ thống cơ sở dữ liệu (CSDL) cho phép xử lý để có câu trả lời. Các phương án đều có kết quả cuối cùng là tương đương nhưng khác nhau trong chi phí thực hiện, tức là tổng thời gian cần để thực hiện một truy vấn. Tối ưu hoá truy vấn là lựa chọn phương án để có tổng thời gian thực hiện truy vấn là nhỏ nhất. Như vậy, vấn đề chúng ta phải quan tâm là làm sao cực tiểu tần suất sử dụng của CPU, bộ nhớ, chi phí vào/ra và các nguồn tài nguyên về lĩnh vực truyền thông. Với kỹ thuật phần cứng hiện nay (khả năng của các chip nhớ), việc tối ưu thực thi một truy vấn chỉ còn là vấn đề làm cực tiểu thời gian trả lời của truy vấn, trong khi đó, các hệ thống lại chịu sự chi phối chủ yếu ở thời gian trao đổi vào/ra. Do đó, các kỹ thuật tối ưu hoá truy vấn chủ yếu tập trung giải quyết vấn đề cực tiểu chi phí xử lý vào/ra khi một truy vấn được thực thi. Phương pháp tối ưu hoá truy vấn đối tượng trong bài báo cũng được đề xuất theo hướng tiếp cận này.

Phương pháp tối ưu hoá truy vấn đối tượng được nghiên cứu và giải quyết các vấn đề với đặc trưng của mô hình hướng đối tượng. Phương pháp này được trình bày trên cơ sở của mô hình dữ liệu ODMG và ngôn ngữ truy vấn đối tượng OQL, nhưng không mất tính tổng quát đối với các mô hình dữ liệu hướng đối tượng có hỗ trợ các đặc trưng này [2], [4], [6].

Tiến trình tổng quát tối ưu hoá truy vấn đối tượng dựa trên tập luật được mô tả trong hình 1, đầu vào của tiến trình xử lý là các truy vấn được viết bằng ngôn ngữ truy vấn đối tượng, chuyển đổi các truy vấn thành các biểu thức đại số đối tượng tương đương. Sau đó, áp dụng các luật biến đổi trên các phép toán đại số như chọn, chiếu, kết nối đối với các lớp đối tượng, loại bỏ trùng lặp trong các đa tập,... Cuối cùng, chúng ta có kết quả là phương án thực thi được chọn trong tiến trình tối ưu truy vấn.



Hình 1. Tiến trình khung xử lý truy vấn

Nội dung của bài báo được cấu trúc như sau: chi tiết của phương pháp tối ưu hoá truy vấn đối tượng bằng các phép biến đổi biểu thức đại số đối tượng OQL được trình bày trong phần 2; phần 3 là ví dụ minh hoạ cho phương pháp đã được trình bày. Kết luận về các đóng góp của bài báo cũng như hướng phát triển được trình bày trong phần 4.

Để thống nhất trong trình bày các ví dụ trong bài báo, chúng ta sử dụng lược đồ đối tượng được định nghĩa trong OQL như sau:

Ví dụ 1.1. Cho lược đồ đối tượng TruongDaihoc như sau:

class NhanSu

type tuple (maso: int, hoten: string, pho: string, tpho: string, matinh: int,
ngaysinh: tuple (ngay: int, thang: int, nam:int))

class SinhVien inherits NhanSu

type tuple (gvhd: string, dtb: float, hocbong: float, tenkhoa: Khoa)

class GiangVien inherits NhanSu

type tuple (bomon: string, mabomon: int, chucvu: string,
tenkhoa: Khoa, luong: int, con: set(NhanSu))

class Khoa

type tuple (makhoa: int, tenkh: string, diadiem: string, ngansach: float,
cbgd: set(GiangVien))

II. Tối ưu hoá truy vấn đối tượng bằng các phép biến đổi biểu thức đại số đối tượng OQL

2.1. Sự biểu diễn tương đương giữa truy vấn OQL và đại số đối tượng

Định nghĩa. Nếu E là biểu thức đại số đối tượng và Q là truy vấn đối tượng OQL cùng xác định một tập đối tượng thì ta nói E biểu diễn Q hay Q biểu diễn E , ta gọi E tương đương với Q , ký hiệu $E \approx Q$.

Sự biểu diễn tương đương giữa truy vấn viết bằng ngôn ngữ OQL và đại số đối tượng được thể hiện qua hai định lý 1 và 2 sau:

Định lý 1. [3] Mọi biểu thức đại số đối tượng đều biểu diễn được bằng các truy vấn đối tượng trong OQL.

Định lý 2. [3] Mọi truy vấn đối tượng trong OQL đều biểu diễn được bằng các biểu thức đại số đối tượng.

Như vậy, việc viết lại một truy vấn đã cho thành các biểu thức đại số với tập phép toán đại số đối tượng là tương đương. Các biểu thức đại số này có thể được ước lượng với các chi phí xử lý khác nhau. Vì vậy, về mặt lý thuyết chúng ta mong muốn tìm được biểu thức đại số tương đương với một truy vấn sao cho có thể đạt được một phương án thực thi hiệu quả hơn. Tuy nhiên, trong các giải pháp cài đặt, vì số lượng các truy vấn tương đương quá lớn, trong lúc đó chúng ta chỉ cần một tập con các truy vấn này. Do đó, để tìm ra các truy vấn tương đương khác, chúng ta sẽ cần một tập luật nhằm biến đổi các biểu thức đại số tương đương. Tuy nhiên, trong mô hình dữ liệu hướng đối tượng lại không có một đại số đối tượng chuẩn áp dụng được cho tất cả các mô hình hướng đối tượng, cho nên sự kỳ vọng để có một tập chuẩn tắc gồm các luật biến đổi bảo toàn tương đương là không tồn tại. Vì vậy, chúng ta mong muốn chứng tỏ rằng sự biến đổi bảo toàn tương đương trên một cơ sở đại số đối tượng là có thể chấp nhận được. Một số luật biến đổi được trình bày trong phần 2.2.

2.2. Các luật biến đổi đại số đối tượng

Ký hiệu S, S_1, S_2, S_3 là các tập đối tượng; e, f, g, h là các biểu thức đại số, phép toán $op \in \{union, diff\}$. Những luật này chỉ áp dụng trên các phép toán đối tượng, phép toán bộ, phép toán tập hợp và các phép toán đa tập (*bag*). Về mặt ký hiệu, chúng ta chỉ sử dụng các ký hiệu phép toán một cách hình thức [6], [7], [8], các phép toán này có thể được cài đặt với một số thay đổi trong các mô hình khác nhau.

L1. Hoán vị phép chọn: $\sigma_{\lambda_t, g}(\sigma_{\lambda_s, f}(S)) = \sigma_{\lambda_s, f}(\sigma_{\lambda_t, g}(S))$

L2. Tổ hợp các phép chọn:

$$\sigma_{\lambda_s, (f \wedge g \wedge \dots \wedge h)}(S) = \sigma_{\lambda_s, f}(\sigma_{\lambda_t, g}(\dots (\sigma_{\lambda_u, h}(S)) \dots))$$

L3. Thu gọn dãy các phép chiếu:

$$\pi_{(a_1..a_n)}(\pi_{(b_1..b_m)}(S)) = \pi_{(a_1..a_n)}(S), \text{ với } \{a_1, \dots, a_n\} \subset \{b_1, \dots, b_m\}$$

L4. Hoán vị phép chọn và phép chiếu

$$\sigma_{\lambda_s.e}(\pi_{(a_1..a_n)}(S)) = \pi_{(a_1..a_n)}(\sigma_{\lambda_s.e}(S))$$

L5. Hoán vị một phép chiếu với phép hợp, hiệu trên tập/đa tập

$$\pi_{(a_1..a_n)}(S_1 \text{ op } S_2) = \pi_{(a_1..a_n)}(S_1) \text{ op } \pi_{(a_1..a_n)}(S_2)$$

L6. Phân phối phép chọn với phép hợp và phép hiệu trên tập/đa tập

$$\sigma_{\lambda_s.f}(S_1 \text{ op } S_2) = \sigma_{\lambda_s.f}(S_1) \text{ op } S_2, \text{ nếu } f \text{ chỉ liên quan với } S_1.$$

Tổng quát: $\sigma_{\lambda_s.(f \wedge g \wedge h)}(S_1 \text{ op } S_2) = \sigma_{\lambda_u.h}(\sigma_{\lambda_s.f}(S_1) \text{ op } \sigma_{\lambda_t.g}(S_2))$, nếu f liên quan S_1 , g liên quan S_2 và h liên quan cả S_1 và S_2 .

L7. Hoán vị giữa phép apply và phép chọn: nếu điều kiện chọn chỉ chứa các thuộc tính do phép toán apply trả về thì:

$$\text{apply}_{\lambda_s.e}(\sigma_{\lambda_t.f}(S)) = \sigma_{\lambda_t.f}(\text{apply}_{\lambda_s.e}(S))$$

L8. Hoán vị giữa phép làm phẳng (flat) và phép apply trên tập/đa tập: giả sử S là thể hiện của một lớp và X là một tập thuộc tính phức của lớp.

$$\text{flat}(\text{apply}_{\lambda_s.(\text{apply}_{\lambda_t.e}(\pi_X(\pi_V(S))))}(S)) = \text{apply}_{\lambda_t.e}(\text{flat}(\text{apply}_{\lambda_s.\pi_X}(\pi_V(S))))(S))$$

Biểu thức ở vế trái, có biểu thức e tác động trước tập các tập (thu được bởi π_X) sau đó làm phẳng thành một tập; biểu thức ở vế phải có phép toán làm phẳng được tác động trước (kết quả thu được là một tập), sau đó thực hiện phép toán apply.

L9. Tính kết hợp của phép hợp

$$(S_1 \text{ union } S_2) \text{ union } S_3 = S_1 \text{ union } (S_2 \text{ union } S_3)$$

L10. Các luật kế thừa đối với phép chọn và phép apply: nếu S_2 là một lớp con của S_1 , thì thể hiện của S_2 là một tập con của thể hiện của S_1 :

$$\sigma_{\lambda_s.f}(S_1) \text{ union } \sigma_{\lambda_s.f}(S_2) = \sigma_{\lambda_s.f}(S_1)$$

$$\text{apply}_{\lambda_s.e}(S_1) \text{ union } \text{apply}_{\lambda_s.e}(S_2) = \text{apply}_{\lambda_s.e}(S_1)$$

2.3. Các quy tắc tổng quát trong tối ưu hoá truy vấn đối tượng

Tiếp theo, với tập luật biến đổi đại số đối tượng trong phần 2.2, chúng ta sẽ đưa ra các quy tắc cho phép chọn lựa các luật thích hợp áp dụng trên các biểu thức đại số đầu vào nhằm tạo ra các bước ước lượng trên các biểu thức đại số đối tượng có chi phí xử lý thấp hơn tương đương với biểu thức đã cho.

(R1) Thực hiện các phép chọn, phép chiếu trên đối tượng, phép bagtoset trước các phép kết nối, tích Đề các, nhóm bộ (đối tượng) trên các lớp theo thứ tự trên: Nhằm

làm giảm số lượng các đối tượng tham gia trong các phép toán. Phép chiếu được áp dụng cho trường hợp tập thuộc tính của các lớp quá lớn nhưng không tham gia trong kết quả của truy vấn, do đó cho phép giảm kích thước lưu trữ của mỗi lớp.

(R2) Tổ hợp dãy các phép chọn và phép chiếu: Dãy các phép toán chọn và chiếu có thể nhóm gộp bằng một phép chọn hoặc một phép chiếu (luật L3). Với phép biến đổi này chúng ta sẽ làm giảm số lần truy xuất trên các lớp.

(R3) Làm phẳng các cấu trúc phức với các phép toán set_flat, bag_flat, list_flag: chuyển các cấu trúc phức về các kiểu tập, bộ và các danh sách với các phần tử đơn trị (lồng nhau). Với phép biến đổi này chúng ta loại bỏ được các tham chiếu lồng, lặp (tự trở) trong các cấu trúc phức làm giảm độ phức tạp tính toán trong quá trình xử lý truy vấn.

(R4) Xử lý trước các lớp đối tượng bằng các phép toán một ngôi: làm giảm kích thước các lớp đối tượng khi tham gia kết nối hay lập nhóm.

(R5) Tính các thành phần cơ sở trong một biểu thức đại số đối tượng: Xác định thành phần cơ sở chung nhất trên các biến vùng, nếu tồn tại một thành phần cơ sở chung nhất thì chúng ta sẽ tính trước các biểu thức con chung này, chúng được xem là đầu vào cho các bước truy vấn tiếp theo.

Thủ tục xác định thành phần cơ sở chung nhất trên các biến vùng: Để thuận tiện trong trình bày, chúng ta biểu diễn lại cú pháp của truy vấn đối tượng OQL:

```
Truy vấn:      select [distinct] < danh_sách_đích >
               from   < mệnh_đề_from >
               where  < mệnh_đề_where >
```

trong đó, các thành phần được thể hiện cụ thể như sau:

< danh_sách_đích > ::= < ds_đích > // danh sách kết quả của truy vấn.

< ds_đích > ::= (< kq_đích > ,) < kq_đích >

< kq_đích > ::= (< kq_thphần > .) < kq_thphần > : < kq_đích >

(có thể chứa các kết nối ẩn, sử dụng dấu “.”)

< kq_thphần > ::= (< đối_tượng > | < biến_tham_chiếu >)

< mệnh_đề_from > ::= < kiểu/lớp > [< toán_tử_tập > < kiểu/lớp >] as

< biến_tham_chiếu >: danh sách các lớp (có thể chứa các toán tử tập hợp) được tham chiếu (định nghĩa qua biến vùng) với **< biến_tham_chiếu >**

< kiểu/lớp > ::= (< thphần_kiểu > .) < thphần_kiểu >

< thphần_kiểu > ::= < đối_tượng > ([< thể_hiện_hàng > [: < thể_hiện_hàng >])

< mệnh_đề_where > ::= < mệnh_đề_where > (and | or) < mệnh_đề_where >

Cho hai <kq_đích> R_1 và R_2 , thì thành phần cơ sở chung nhất, ký hiệu $GC(R_1, R_2)$ là “tiền tố” chung nhất của các <kq_đích>. Chúng ta xác định thành phần cơ sở của mỗi <kq_đích> trong <ds_đích> bằng thuật toán:

Thuật toán 2.1: *Xác định thành phần cơ sở chung nhất*

Vào: R_1, R_2 là các kết quả thuộc <ds_đích> của truy vấn

Ra: Thành phần cơ sở chung nhất của các kết quả

Phương pháp:

- (1) **for** mỗi R_1 in <ds_đích> **do**
- (2) **for** mỗi R_2 in <ds_đích> **do**
- (3) $G := GC(R_1, R_2)$
- (4) **if** ($G > PC(R_1)$) **then** $PC(R_1) := G$

trong đó, GC là thành phần cơ sở chung nhất của R_1 và R_2 , PC : thành phần cơ sở của R_1 .

2.4. Thuật toán tối ưu hoá truy vấn đối tượng dựa trên tập luật

Thuật toán được giới thiệu trong phần này tập trung xử lý các phép toán chiếu, chọn, áp dụng biểu thức đại số (set_apply) trên các kiểu đối tượng và phép toán loại bỏ trùng lặp trên các đa tập, lớp đối tượng.

Thuật toán 2.2: Tối ưu hoá các biểu thức đại số đối tượng dựa trên tập luật.

Vào: Biểu thức đại số đối tượng.

Ra: Một dãy các bước ước lượng biểu thức đại số đối tượng.

Phương pháp:

- (1) Khởi tạo cây phân tích cú pháp từ biểu thức đại số đối tượng.
- (2) Sử dụng luật (L2) tách phép chọn $\sigma_{\lambda_s.(f \wedge g \wedge \dots \wedge h)}(S)$ thành chuỗi các phép chọn:
 $\sigma_{\lambda_s.f}(\sigma_{\lambda_t.g}(\dots(\sigma_{\lambda_u.h}(S))\dots))$
- (3) Sử dụng các luật kế thừa đối với các phép chiếu (L3), phép chọn và phép apply (L10) tổ hợp dãy các phép chiếu, chọn thành một phép chiếu và một phép chọn
- (4) Đối với mỗi phép chọn, sử dụng các luật (L4, L6, L7, L10) “đẩy” các phép chọn xuống các lớp thành phần hoặc “qua” các nút kết nối và phép tạo nhóm.
- (5) Đối với mỗi phép chiếu (đối tượng, tập, bộ), sử dụng luật (L3, L4, L5) để di chuyển phép chiếu xuống càng sâu càng tốt. Nếu tập thuộc tính được chiếu bao gồm tất cả các thuộc tính của biểu thức thì chúng ta loại bỏ phép chiếu đó.
- (6) Sử dụng các luật (L8, L9, L10) trên các lớp đối tượng, để loại bỏ các phần

tử trùng lặp trong các lớp đối tượng; di chuyển phép làm phẳng (flat), phép loại bỏ trùng lặp trong các đa tập (bagtose) lên trước các phép toán nhóm hoặc kết nối.

- (7) Tạo ra dãy các bước biến đổi để ước lượng mỗi nhóm theo một thứ tự sao cho không có nhóm nào được ước lượng trước các nhóm con của nó.

Mệnh đề. Thuật toán trên là đúng đắn và kết quả của thuật toán là phương án truy vấn có chi phí thấp hơn chi phí ước lượng của biểu thức đầu vào thuật toán.

Chứng minh.

Theo định lý 1, 2 và tập các luật biến đổi biểu thức đại số đối tượng, chúng ta suy ra các bước thực thi trong thuật toán cho kết quả đúng và tương đương.

Áp dụng các phép toán chiếu trên các lớp đối tượng làm giảm kích thước các lớp và các đối tượng tham gia trong biểu thức, điều này sẽ làm giảm chi phí nạp lớp vào bộ nhớ trong (IO_Load). Mặt khác, phép toán làm phẳng (set_flat) và loại bỏ trùng lặp (bagtose) áp dụng cho đa tập, lớp đối tượng sẽ làm giảm một cách đáng kể các biến thể của các lớp tham gia trong các phép kết nối, nhóm tương đương (IO_Eval).

Độ phức tạp tính toán của thuật toán có thời gian đa thức theo kích thước (số các biến thể) của các lớp tham gia trong biểu thức.

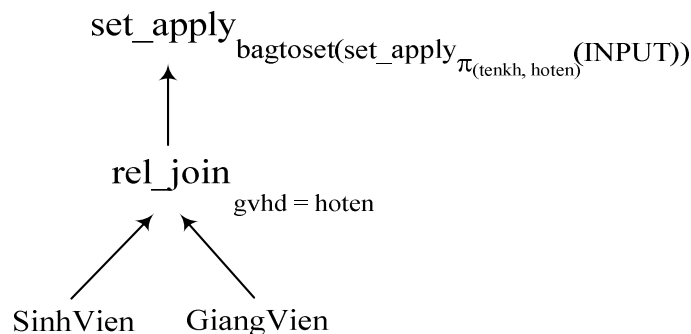
III. Ví dụ minh họa

Chúng ta sử dụng cây phân tích cú pháp để thực hiện việc biến đổi các biểu thức đại số đối tượng.

Ví dụ 3.1. Ta có truy vấn:

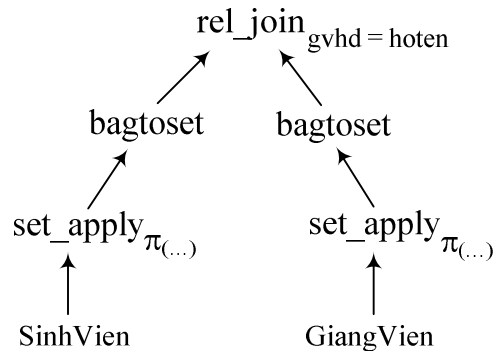
```
define SinhVien as s, GiangVien as e
select distinct (s.tenkhoa.tenkh, e.hoten)
where s.gvhd = e.hoten
```

Hình 2 là biểu diễn truy vấn của ví dụ 2 trên cây phân tích cú pháp.



Hình 2. Khởi tạo cây phân tích cú pháp của ví dụ 2

Hình 3 biểu diễn cây phân tích cú pháp khi chúng ta áp dụng luật “đẩy” phép toán *bagtoiset* (luật R1) lên trước toán tử kết nối *rel_join*, luật này áp dụng khi các thành phần trùng lặp quá lớn tồn tại trong các đa tập, như vậy toán tử *bagtoiset* chỉ thực hiện trên $|s| + |e|$ biến thể (trong trường hợp xấu nhất) tốt hơn $|s|*|e|$ biến thể (đối với trường hợp phép kết nối được thực hiện trước). Và tiếp tục đẩy toán tử π qua nút “*join*”.

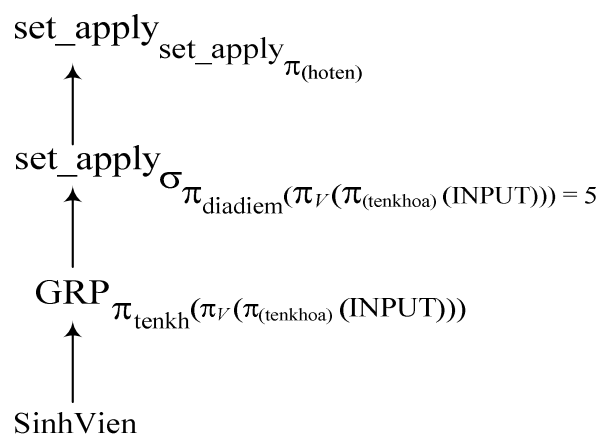


Hình 3. Bước chuyển đổi đầu tiên của hình 2

Ví dụ 3.2. Tìm tên các sinh viên của khoa có văn phòng khoa đặt ở tầng 5 (diadiem). Tên các sinh viên được nhóm theo khoa với thuộc tính tenkh (ví dụ: Sinh học, Công nghệ thông tin):

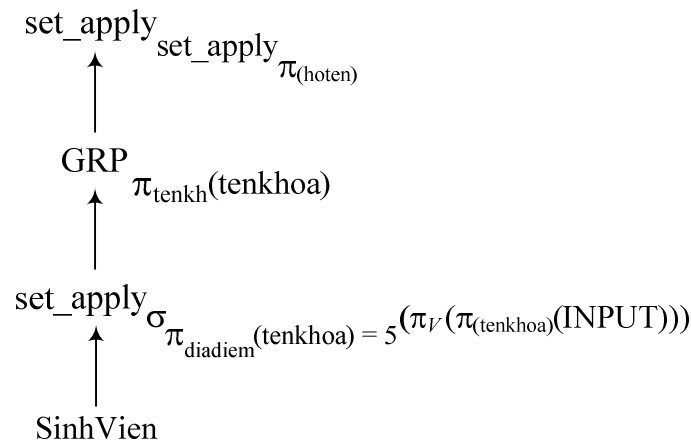
```
define SinhVien as s
select (s.hoten)
group by s.tenkhoa.tenkh
where s.tenkhoa.diadiem = 5
```

Hình 4 biểu diễn cây phân tích cú pháp đại số cho ví dụ 3, ta nhóm đa tập trên thuộc tính tenkh của thuộc tính tenkhoa, sau đó loại bỏ các sinh viên của các khoa không ở tầng 5, cuối cùng chiếu lấy thuộc tính hoten.



Hình 4. Khởi tạo cây của ví dụ 3

Một phương pháp tối ưu ví dụ 3 được suy trực tiếp từ hình 4. Trước hết, ta đưa phép chọn lên trước phép tạo nhóm GRP và sử dụng các phép chiếu trên đối tượng (π_V) để loại bỏ tham chiếu tên thuộc tính tenkhoa, sau đó trích chiếu lấy giá trị của thuộc tính diadiem.



Hình 5. Cây kết quả sau khi áp dụng các luật chuyển đổi

IV. Kết luận

Bài báo đã tập trung vào việc trình bày tập luật biến đổi các phép toán đại số đối tượng, đề xuất các quy tắc tổng quát trong tối ưu truy vấn đối tượng, đồng thời mở rộng và cải tiến thuật toán tối ưu hoá các biểu thức đại số đối tượng dựa vào tập luật.

Phương pháp tối ưu hoá truy vấn đối tượng dựa vào các luật biến đổi biểu thức đại số được áp dụng cho các mô hình dữ liệu hướng đối tượng có hỗ trợ tập luật. Phương pháp này xử lý cho lớp các truy vấn trên các kiểu đối tượng phức như kiểu bag, đa tập, danh sách, bộ,... nhưng khi thực hiện trên các lớp đối tượng, các biểu thức đại số đối tượng lồng thì chi phí ước lượng và các phương án thực thi truy vấn chưa hiệu quả. Cho nên, nếu chúng ta sử dụng siêu đồ thị kết nối đối tượng để giải quyết cho lớp các truy vấn đối tượng lồng với các thuật toán ước lượng trên các siêu cạnh đối tượng sẽ có chi phí xử lý truy vấn hiệu quả hơn so với phương pháp tối ưu các truy vấn đối tượng lồng bằng các phép biến đổi đại số trên các biểu thức đại số đối tượng lồng nhau.

TÀI LIỆU THAM KHẢO

1. Bierman G.M. and Trigoni A. *Towards A Formal Type System For ODMG OQL*, Technical Report 497, University of Cambridge, Computer Laboratory, 2000.
2. Cattel R.G.G., Barry D.K. *The Object Database Standard: ODMG 3.0*, Morgan Kaufmann Publishers, 2000.

3. Đoàn Văn Ban, Lê Mạnh Thạnh và Hoàng Bảo Hùng. *Sự tương đương trong biểu diễn giữa ngôn ngữ truy vấn OQL và đại số đối tượng*, Tạp chí Tin học và Điều khiển học, 20(3), (2004), 257–269.
4. Lê Mạnh Thạnh, Hoàng Bảo Hùng, *Ngôn ngữ truy vấn hướng đối tượng và tối ưu hoá truy vấn trên CSDL hướng đối tượng bằng phương pháp biến đổi đại số*, Kỷ yếu Hội nghị khoa học kỷ niệm 25 năm thành lập Viện Công nghệ Thông tin, Hà Nội, (2001), 175–185.
5. Lê Mạnh Thạnh, Hoàng Bảo Hùng, *Mô hình ước lượng chi phí xử lý truy vấn đối tượng trong cơ sở dữ liệu hướng đối tượng*, Kỷ yếu Hội thảo Quốc gia, lần thứ VIII, “Một số vấn đề chọn lọc về CNTT và truyền thông”, chủ đề “Mã nguồn mở”, 25/8-27/8/2005, Hải Phòng, Nhà xuất bản Khoa học và Kỹ thuật, Hà Nội, (2006), 568-579
6. Trigoni A. *Semantic Optimization of OQL Queries*, Technical Report, Number 547, University of Cambridge, Computer Laboratory, UCAM-CL-TR-547, ISSN 1476-2986, 2002.
7. Vanderberg, Scott Lee, *Algebras for Object - Oriented Query Languages*, Ph.D. Dissertation, University of Wisconsin-Madison, 1993.
8. Yu, Clement T., Meng, Weiyi. *Principles of Database Query Processing for Advanced Applications*, Morgan Kaufmann Publishers, Inc. San Francisco, California, 1998.

AN OPTIMIZATION METHOD FOR OBJECT QUERIES USING THE OQL OBJECT ALGEBRA EXPRESSION TRANSFORMATION

Le Manh Thanh

Hue University

Hoang Bao Hung

Thua Thien Hue Department of Information and Communications

SUMMARY

The query optimization is an attracting issue in the research on database theory. The complete results of the query optimization in relational databases model are solutions for classes of management problems in general and for the database theory in particular. On that basis, our approach is to expand methods of the query optimization in object-oriented databases, and that is presented in this article. The difference from the relational database model is that the object query optimization is based on the rules set which transforms the query expression using object algebra transformations, and this is carried out by the equivalent transformation between OQL-written queries and object algebra, and it's proven that the OQL query and the object algebra expression is equivalent. This article proposes general principles to optimize object queries, and from this basis we also propose an algorithm for object query optimization based on the rules set.