TỐI ƯU HOÁ TRUY VẤN

I. Các bảng dữ liệu và lược đồ phân mảnh

1. Các bảng dữ liệu

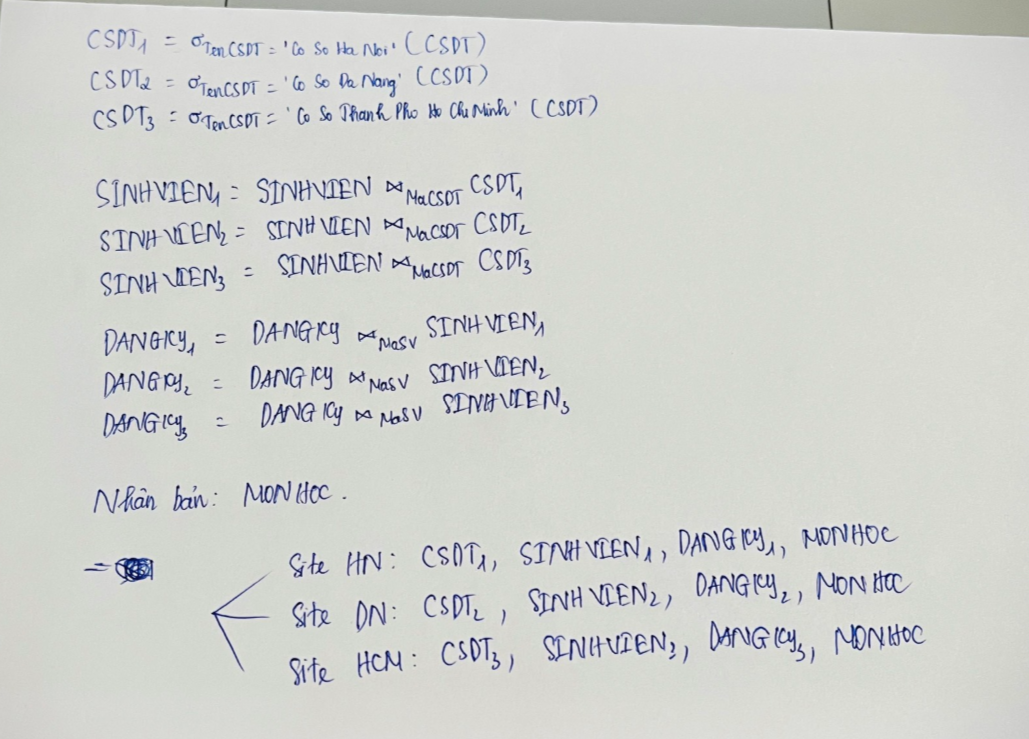
Bảng csdt(maCsdt, tenCsdt): (ảnh)

Bảng sinhvien(maSv, tenSv, maCsdt, gpa, tongTinchi): (ảnh)

Bảng monhoc(maMh, tenMh, soTinchi): (ảnh)

Bảng dangky(maSv, maMh, kyHoc) : (ảnh)

2. Lược đồ phân mảnh:



II. Xử lý truy vấn trong môi trường tập trung

***Thuật toán INGRES***

***Ý tưởng thuật toán****:* Thuật toán tổ hợp hai giai đoạn phân rã và tối ưu hoá.

* Đầu tiên phân rã câu truy vấn dạng phép toán quan hệ thành các phần nhỏ hơn.
* Câu truy vấn được phân rã thành một chuỗi các truy vấn có một quan hệ chung duy nhất.
* Sau đó mỗi câu truy vấn đơn quan hệ được xử lí bởi một “*thể xử lý truy vấn một biến*” (one variable query processor-OVQP)
* Trước tiên OVQP sẽ thực hiện các phép toán đơn ngôi và cố gắng giảm thiểu kích thước của các kết quả trung gian bằng các **phép tách** (detachment) và **Phép thế** (substitution)
* Kí hiệu qi-1→qi để chỉ câu truy vấn q được phân rã thành hai câu truy vấn con qi-1và qi, trong đó qi-1 được thực hiện trước và kết quả sẽ được qi sử dụng.
* **Phép tách**: OVQP sử dụng để tách câu truy vấn q thành các truy vấn q’→q” dựa trên một quan hệ chung là kết quả của q’.

**VD** **minh hoạ:** Xét CSDL đã biểu diễn ở trên

Xét câu truy vấn q1: Liệt kê tên các sinh viên có GPA > 1 đăng ký môn CSDLPT ở kỳ 1 hoặc kỳ 2.

- Dịch truy vấn:

q1: SELECT sinhvien.tenSv

FROM sinhvien, dangky, monhoc

WHERE sinhvien.maSv = dangky.maSv

AND dangky.maMh = monhoc.maMh

AND sinhvien.gpa > 1

AND dangky.kyhoc < 3

AND monhoc.tenMh = ‘CSDLPT’

**-** Thuật toán Ingres:

+ Tách q1 thành : q11 -> q’, trong đó TMP1 là quan hệ trung gian.

q11: SELECT monhoc.maMh INTO TMP1

FROM monhoc

WHERE monhoc.tenMh = ‘CSDLPT’

q’: SELECT sinhvien.tenSv

FROM sinhvien, dangky, TMP1

WHERE sinhvien.maSv = dangky.maSv

AND sinhvien.gpa > 1

AND dangky.kyhoc < 3

AND dangky.maMh = TMP1.maMh

+ Tách tiếp q’:

q12: SELECT dangky.maSv INTO TMP2

FROM dangky, TMP1

WHERE dangky.kyhoc < 3

AND dangky.maMh = TMP1.maMh

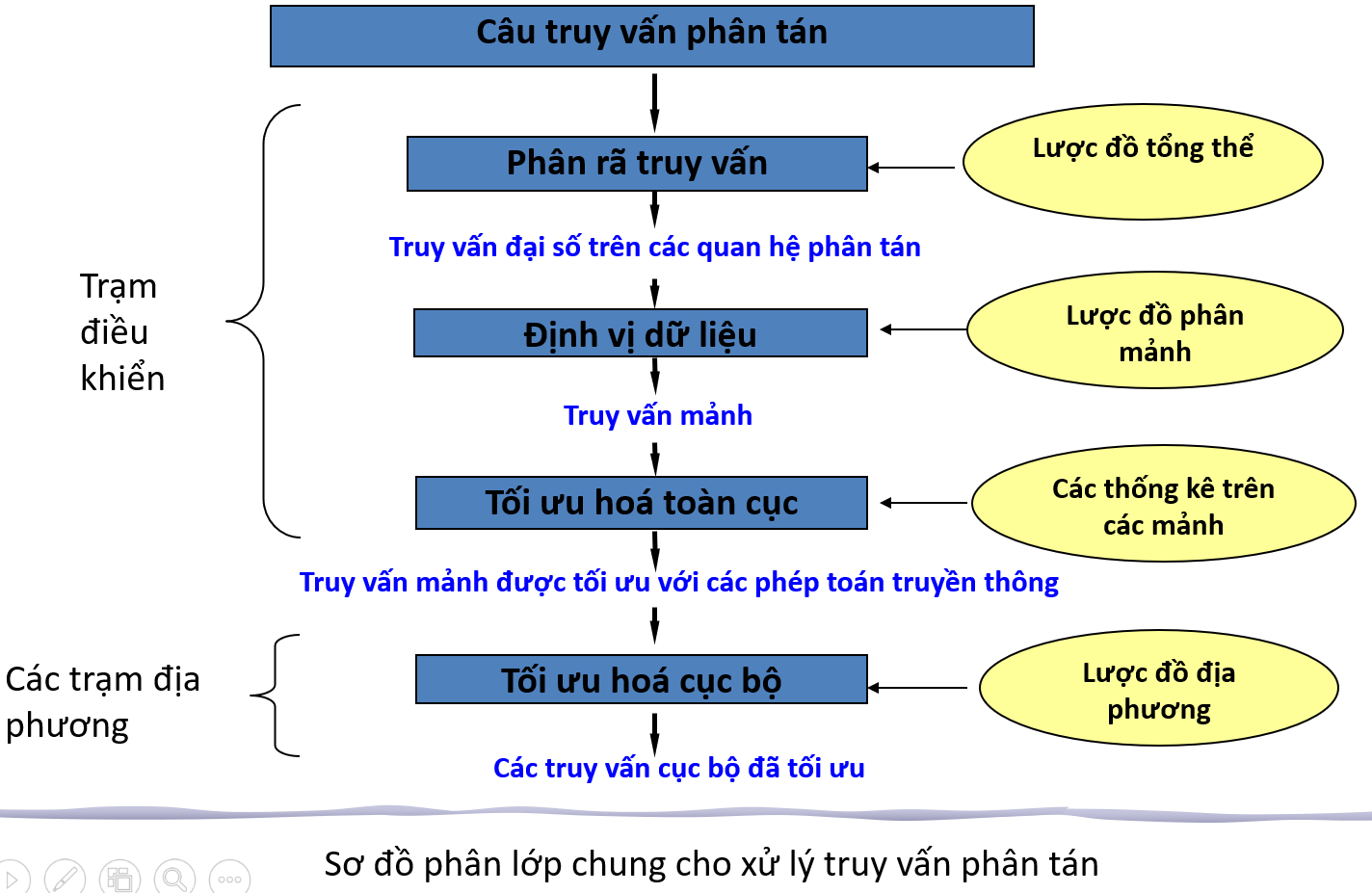
q13: SELECT sinhvien.tenSv

FROM sinhvien, TMP2

WHERE sinhvien.maSv = TMP2.maSv

AND sinhvien.gpa > 1

III. Xử lý truy vấn trong môi trường phân tán



1. Phân rã truy vấn (4 bước)

a, Chuẩn hoá

**Mục đích:** chuyển đổi truy vấn thành một dạng chuẩn để thuận lợi cho các xử lý tiếp theo.

Với SQL, có hai dạng chuẩn cho các vị từ trong mệnh đề WHERE là:

***Dạng chuẩn hội***  là hội (∧) của những phép toán tuyển (∨):

(p11∨ p12∨ ... ∨ p1n) ∧ ... ∧ (pm1∨ pm2∨ ... ∨ pmn)

***Dạng chuẩn tuyển*** là tuyển (∨) của những phép toán hội (∧):

(p11 ∧ p12 ∧ ... ∧ p1n) ∨ ... ∨ (pm1 ∧ pm2 ∧ ... ∧pmn), trong đó pij là các biểu thức nguyên tố.

VD: xét câu truy vấn gốc

SELECT sinhvien.tenSv

FROM sinhvien, dangky, monhoc

WHERE sinhvien.maSv = dangky.maSv

AND dangky.maMh = monhoc.maMh

AND sinhvien.gpa > 1

AND dangky.kyhoc < 3

AND monhoc.tenMh = ‘CSDLPT’

* Điều kiện trong dạng chuẩn hội: (sinhvien.maSv = dangky.maSv) ∧ (dangky.maMh = monhoc.maMh) ∧ sinhvien.gpa > 1 ∧ dangky.kyhoc < 3 ∧ (monhoc.tenMh = ‘CSDLPT’)

b, Phân tích

**Mục đích**: Phát hiện ra những thành phần không đúng (sai kiểu hoặc sai ngữ nghĩa) và loại bỏ chúng sớm nhất nếu có thể.

***Truy vấn sai kiểu****:* nếu một thuộc tính bất kỳ hoặc tên quan hệ của nó không được định nghĩa trong lược đồ tổng thể, hoặc phép toán áp dụng cho các thuộc tính sai kiểu.

VD: Truy vấn sau:

SELECT sinhvien.gioitinh

FROM sinhvien

WHERE sinhvien.tenSv > 100

* Truy vấn sai kiểu do 2 lý do:

+Thuộc tính gioitinh không được khai báo trong lược đồ

+ Phép toán “>100” không thích hợp với kiểu chuỗi của thuộc tính tenSv

***Truy vấn sai ngữ nghĩa****:* nếu các thành phần của nó không tham gia vào việc tạo ra kết quả.

Để xác định truy vấn có sai về ngữ nghĩa hay không, ta dựa trên việc biểu diễn truy vấn như một đồ thị gọi là *đồ thị truy vấn*. Đồ thị này được xác định bởi các truy vấn liên quan đến phép chọn, chiếu và nối. Nếu đồ thị truy vấn mà không liên thông thì truy vấn là sai ngữ nghĩa

***Đồ thị truy vấn:***

* Có một nút dùng để biểu diễn cho quan hệ kết quả
* Các nút khác biểu diễn cho các toán hạng trong câu truy vấn (các quan hệ)
* Cạnh nối giữa hai nút mà không phải là nút kết quả thì biểu diễn một **phép nối.**
* Cạnh có nút đích là nút kết quả thì biểu diễn một **phép chiếu**.
* Một nút không phải là nút kết quả có thể được gán nhãn bởi **phép chọn** hoặc **phép tự nối** (seft-join: nối của quan hệ với chính nó).

VD: xét câu truy vấn gốc:

SELECT sinhvien.tenSv

FROM sinhvien, dangky, monhoc

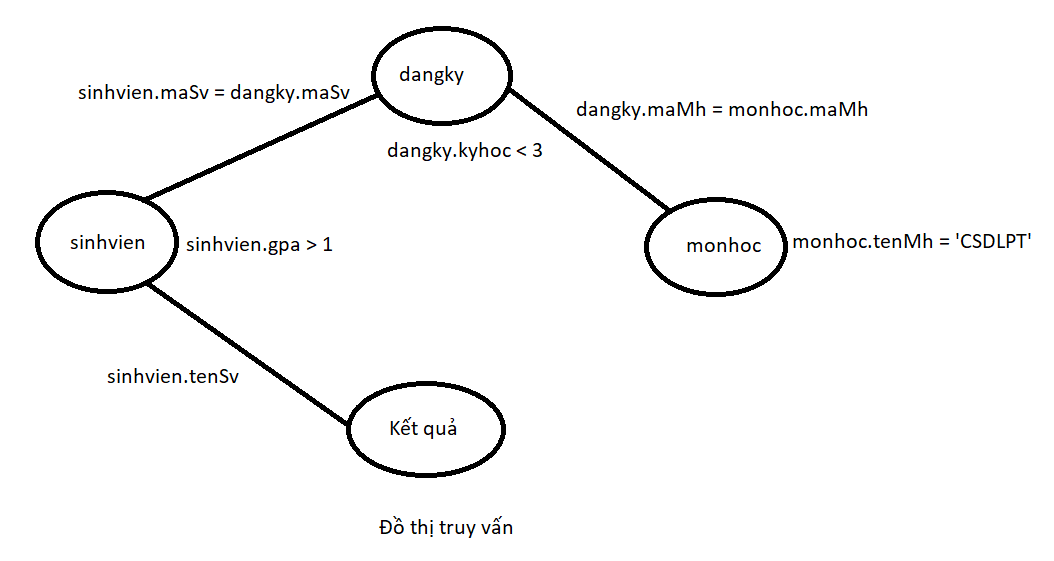
WHERE sinhvien.maSv = dangky.maSv

AND dangky.maMh = monhoc.maMh

AND sinhvien.gpa > 1

AND dangky.kyhoc < 3

AND monhoc.tenMh = ‘CSDLPT’



* Đồ thị truy vấn liên thông
* Câu truy vấn đúng ngữ nghĩa

c, Loại bỏ dư thừa

Điều kiện trong các truy vấn có thể có chứa các vị từ dư thừa.

Một đánh giá sơ sài về một điều kiện dư thừa có thể dẫn đến lặp lại một số công việc.

Sự dư thừa vị từ và dư thừa công việc có thể được loại bỏ bằng cách làm đơn giản hoá các điều kiện thông qua các luật luỹ đẳng sau:

1. p ∧ p⇔ p 2. p ∨ true ⇔ true

3. p ∨ p⇔ p 4. p ∧ ¬ p ⇔ false

5. p ∧ true ⇔ p 6. p ∨ ¬ p ⇔ true

7. p ∨ false ⇔ p 8. p1 ∧ (p1 ∨ p2) ⇔ p1

9. p ∧ false ⇔ false 10.p1 ∨ (p1 ∧ p2) ⇔ p1

VD: Xét câu truy vấn gốc:

SELECT sinhvien.tenSv

FROM sinhvien, dangky, monhoc

WHERE sinhvien.maSv = dangky.maSv

AND dangky.maMh = monhoc.maMh

AND sinhvien.gpa > 1

AND dangky.kyhoc < 3

AND monhoc.tenMh = ‘CSDLPT’

Đặt p1: <sinhvien.maSv = dangky.maSv>

p2: <dangky.maMh = monhoc.maMh>

p3: <sinhvien.gpa > 1>

p4: <dangky.kyhoc < 3>

p5: <monhoc.tenMh = ‘CSDLPT’>

Các vị từ sau mệnh đề WHERE được mô tả: p1 ∧ p2 ∧ p3 ∧ p4 ∧ p5

* Không có dư thừa

d, Viết lại

Bước này được chia làm hai bước con như sau:

* Biến đổi trực tiếp truy vấn phép tính sang đại số quan hệ.
* Cấu trúc lại truy vấn đại số quan hệ để cải thiện hiệu quả thực hiện. Đại số quan hệ là một cây mà nút lá biểu diễn một quan hệ trong CSDL, các nút không lá là các quan hệ trung gian được sinh ra bởi các phép toán đại số quan hệ.

Cách chuyển một truy vấn phép tính quan hệ thành một cây đại số quan hệ:

* Các nút lá khác nhau được tạo cho mỗi biến bộ khác nhau (tương ứng một quan hệ). Trong SQL các nút lá chính là các quan hệ trong mệnh đề FROM.
* Nút gốc được tạo ra bởi một phép chiếu lên các thuộc tính kết quả. Trong SQL nút gốc được xác định qua mệnh đề SELECT.
* Điều kiện (mệnh đề WHERE trong SQL) được biến đổi thành dãy các phép toán đại số thích hợp (phép chọn, nối, phép hợp, v.v...) đi từ lá đến gốc, có thể thực hiện theo thứ tự xuất hiện của các vị từ và các phép toán.

VD: Xét câu truy vấn gốc:

SELECT sinhvien.tenSv

FROM sinhvien, dangky, monhoc

WHERE sinhvien.maSv = dangky.maSv

AND dangky.maMh = monhoc.maMh

AND sinhvien.gpa > 1

AND dangky.kyhoc < 3

AND monhoc.tenMh = ‘CSDLPT’

**Quá trình xử lý cây đại số quan hệ:**

