



Chương 4 Khai thác cơ sở dữ liệu

Giáo viên: Nguyễn Mậu Uyên

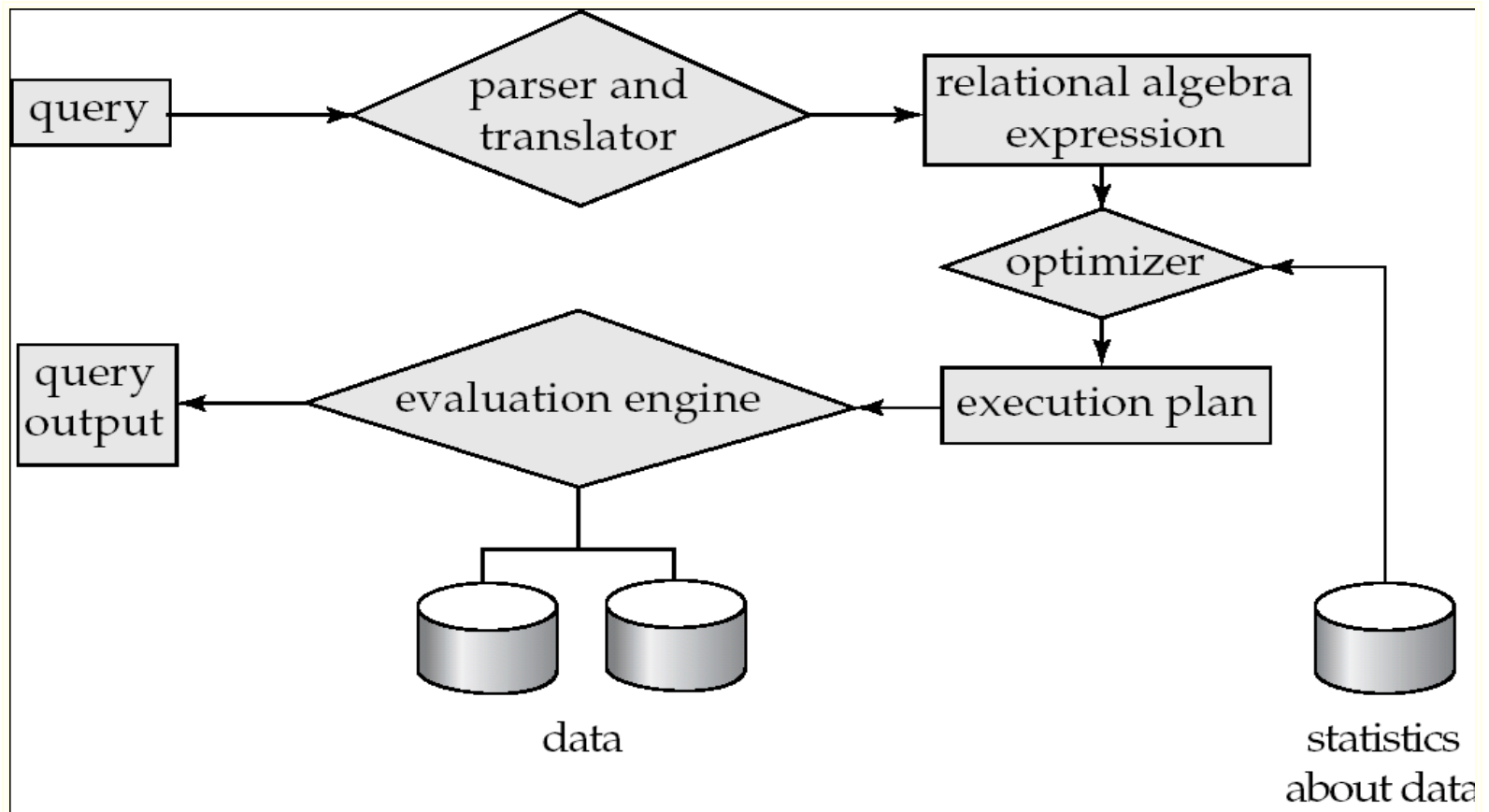
Bộ môn: Hệ thống thông tin

Biên soạn: Đỗ Thị Mai Hương

4.1 Tối ưu hóa câu truy vấn

Tối ưu hóa câu truy vấn

❖ Quá trình thực hiện một câu truy vấn



Tối ưu hóa câu truy vấn

❖ Quá trình thực hiện một câu truy vấn

❖ Bước 1:

- Bộ quét: duyệt truy vấn để biết truy vấn được viết bằng ngôn ngữ nào.
- Bộ kiểm tra: kiểm tra cú pháp của truy vấn xem có hợp lệ hay không?
- Xác nhận tính hợp lệ: các quan hệ, thuộc tính sử dụng trong truy vấn đã được khai báo hay chưa?
- Biểu diễn truy vấn bằng một biểu thức đại số quan hệ

Tối ưu hóa câu truy vấn

❖ Bước 2:

- Bộ tối ưu: Tìm ra phương pháp thực hiện tối ưu cho truy vấn.
- Sau bước này sẽ cho ra một biểu thức đại số quan hệ với chi phí thực hiện nhỏ nhất.

❖ Bước 3:

- Bộ tạo mã sẽ tạo ra chương trình bằng ngôn ngữ trong để thực hiện truy vấn.
- Thực thi chương trình để lấy về kết quả.

Tối ưu hóa câu truy vấn

❖ Biến đổi câu truy vấn về đại số quan hệ

- Select chuyển thành phép chiếu
- Mệnh đề From chuyển thành Tích đề các
- Phép Join chuyển thành nối theo điều kiện
- Mệnh đề Where chuyển thành phép chọn

Tối ưu hóa câu truy vấn

Các phép biến đổi tương đương

1.
$$\sigma_{\theta_1 \wedge \theta_2}(E) = \sigma_{\theta_1}(\sigma_{\theta_2}(E))$$

2.
$$\sigma_{\theta_1}(\sigma_{\theta_2}(E)) = \sigma_{\theta_2}(\sigma_{\theta_1}(E))$$

3.
$$\Pi_{L_1}(\Pi_{L_2}(\dots(\Pi_{L_n}(E))\dots)) = \Pi_{L_1}(E)$$

4. a.
$$\sigma_{\theta}(E_1 \bowtie E_2) = E_1 \bowtie_{\theta} E_2$$

b.
$$\sigma_{\theta_1}(E_1 \bowtie_{\theta_2} E_2) = E_1 \bowtie_{\theta_1 \wedge \theta_2} E_2$$

Tối ưu hóa câu truy vấn

Các phép biến đổi tương đương

5.
$$E_1 \bowtie_{\theta} E_2 = E_2 \bowtie_{\theta} E_1$$

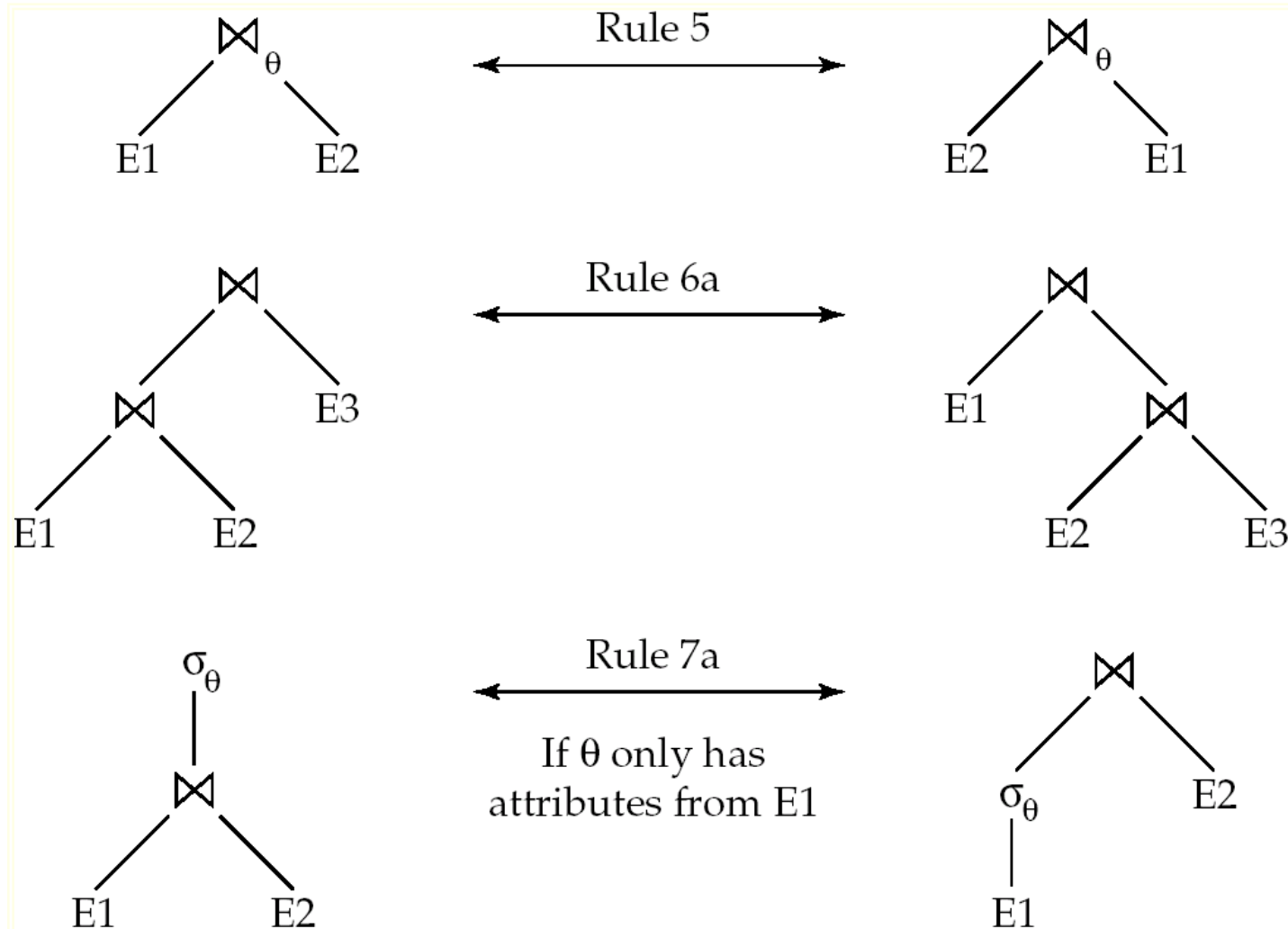
6. (a)
$$(E_1 \bowtie E_2) \bowtie E_3 = E_1 \bowtie (E_2 \bowtie E_3)$$

(b)
$$(E_1 \bowtie_{\theta_1} E_2) \bowtie_{\theta_2 \wedge \theta_3} E_3 = E_1 \bowtie_{\theta_1 \wedge \theta_3} (E_2 \bowtie_{\theta_2} E_3)$$

7. (a)
$$\sigma_{\theta_0}(E_1 \bowtie_{\theta} E_2) = (\sigma_{\theta_0}(E_1)) \bowtie_{\theta} E_2$$

(b)
$$\sigma_{\theta_1 \wedge \theta_2}(E_1 \bowtie_{\theta} E_2) = (\sigma_{\theta_1}(E_1)) \bowtie_{\theta} (\sigma_{\theta_2}(E_2))$$

Tối ưu hóa câu truy vấn



Tối ưu hóa câu truy vấn

$$8.(a) \quad \Pi_{L_1 \cup L_2}(E_1 \bowtie_{\theta} E_2) = (\Pi_{L_1}(E_1)) \bowtie_{\theta} (\Pi_{L_2}(E_2))$$

$$(b) \quad \Pi_{L_1 \cup L_2}(E_1 \bowtie_{\theta} E_2) = \Pi_{L_1 \cup L_2}((\Pi_{L_1 \cup L_3}(E_1)) \bowtie_{\theta} (\Pi_{L_2 \cup L_4}(E_2)))$$

$$9. \quad \begin{aligned} E_1 \cup E_2 &= E_2 \cup E_1 \\ E_1 \cap E_2 &= E_2 \cap E_1 \end{aligned}$$

$$10. \quad \begin{aligned} (E_1 \cup E_2) \cup E_3 &= E_1 \cup (E_2 \cup E_3) \\ (E_1 \cap E_2) \cap E_3 &= E_1 \cap (E_2 \cap E_3) \end{aligned}$$

$$11. \quad \sigma_{\theta}(E_1 - E_2) = \sigma_{\theta}(E_1) - \sigma_{\theta}(E_2)$$

$$12. \quad \Pi_L(E_1 \cup E_2) = (\Pi_L(E_1)) \cup (\Pi_L(E_2))$$

Tối ưu hóa câu truy vấn

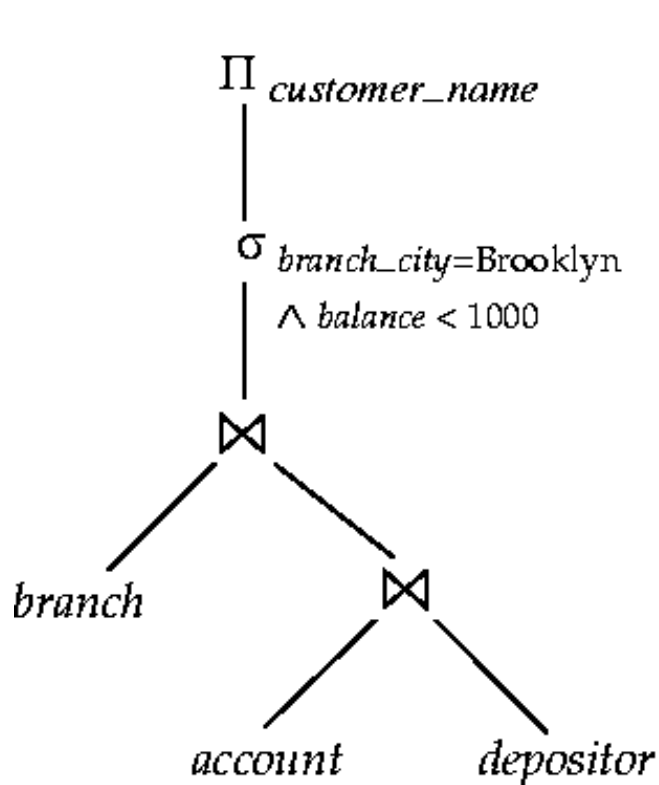
A decorative header image featuring a close-up of a fountain pen nib on the right, with a blurred background of a document and a logo that says "LOGO" in green capital letters.

❖ Quá trình tối ưu

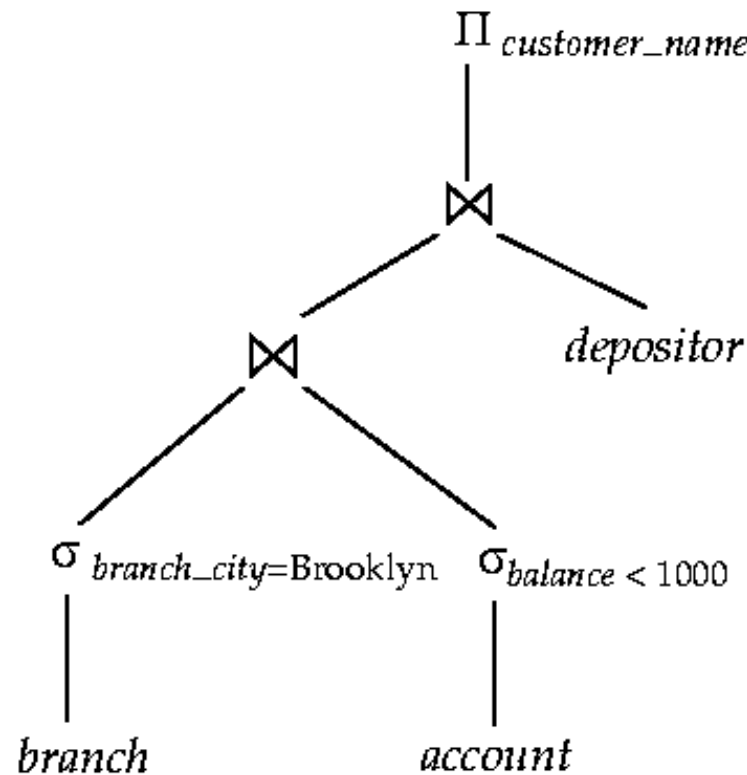
- Nhằm đạt được tốc độ nhanh nhất và bộ nhớ chiếm dụng ít.
- Biến đổi thứ tự thực hiện các phép toán của biểu thức đại số quan hệ sao cho các phép toán một ngôi được thực hiện trước các phép toán hai ngôi.
- Giải thuật:
 - Phân rã điều kiện chọn, đẩy từng điều kiện thành viên xuống dưới nút lá khi còn có thể.
 - Chiều lấy thuộc tính cần thiết.

Tối ưu hóa câu truy vấn

❖ Quá trình tối ưu – Ví dụ:



(a) Initial expression tree



(b) Tree after multiple transformations

Tối ưu hóa câu truy vấn

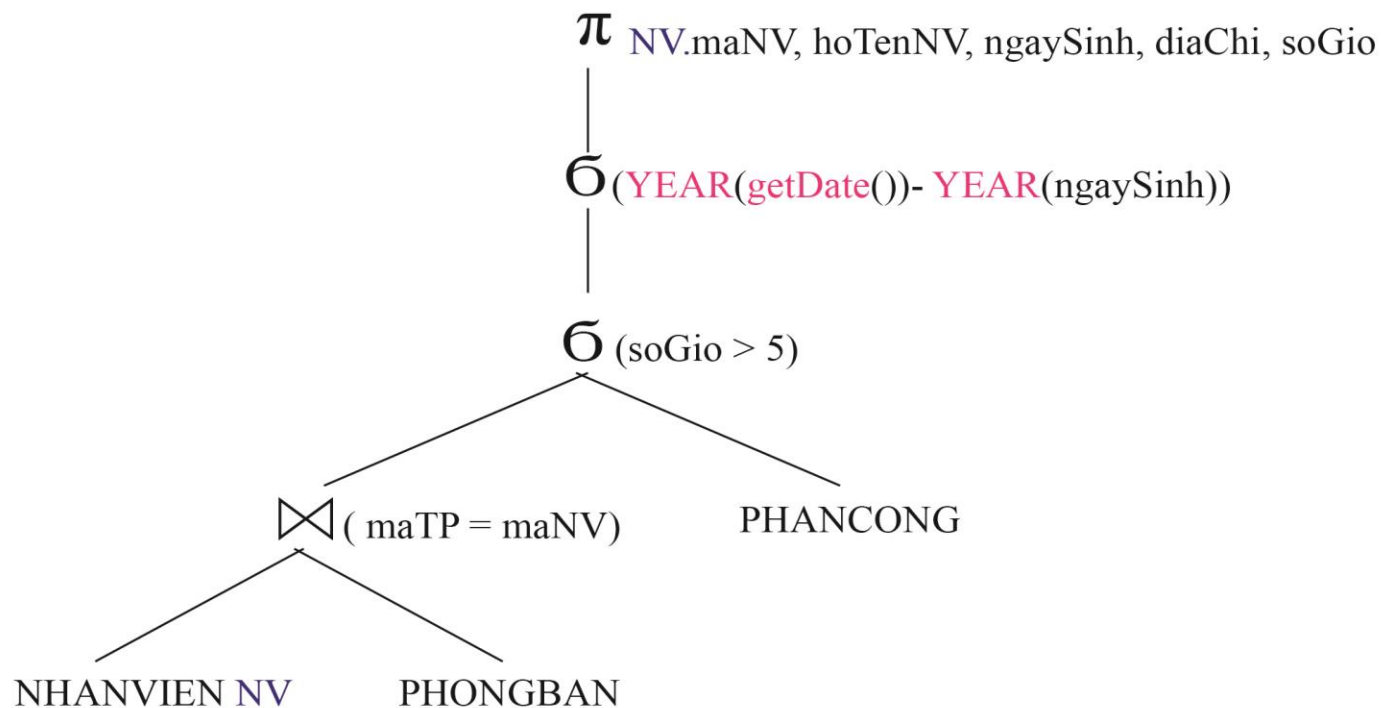


❖ Viết câu lệnh truy vấn đưa ra nhân viên là trưởng phòng, tham gia dự án có số giờ làm việc >5 giờ và có tuổi <40?

```
Select  NV.maNV, hoTenNV, ngaySinh, gioiTinh, diaChi, soGio
From    NhanVien NV, PhongBan PB, PhanCong PC
Where   maNV=maTP
        and    NV.maNV=PC.maNV
        and    soGio>5
        and    YEAR(getdate())- YEAR(ngaySinh )<40
```

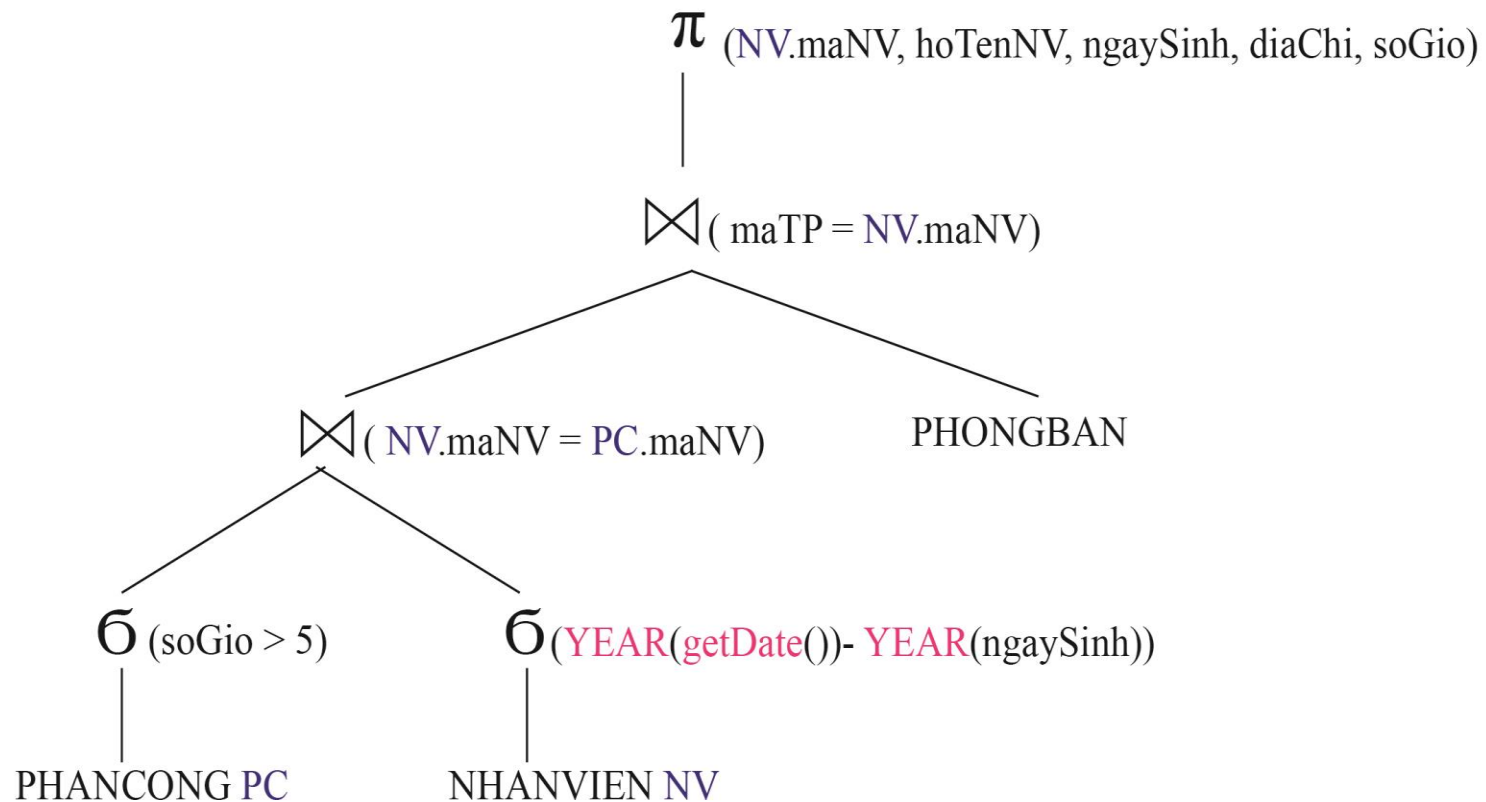
Tối ưu hóa câu truy vấn

LOGO



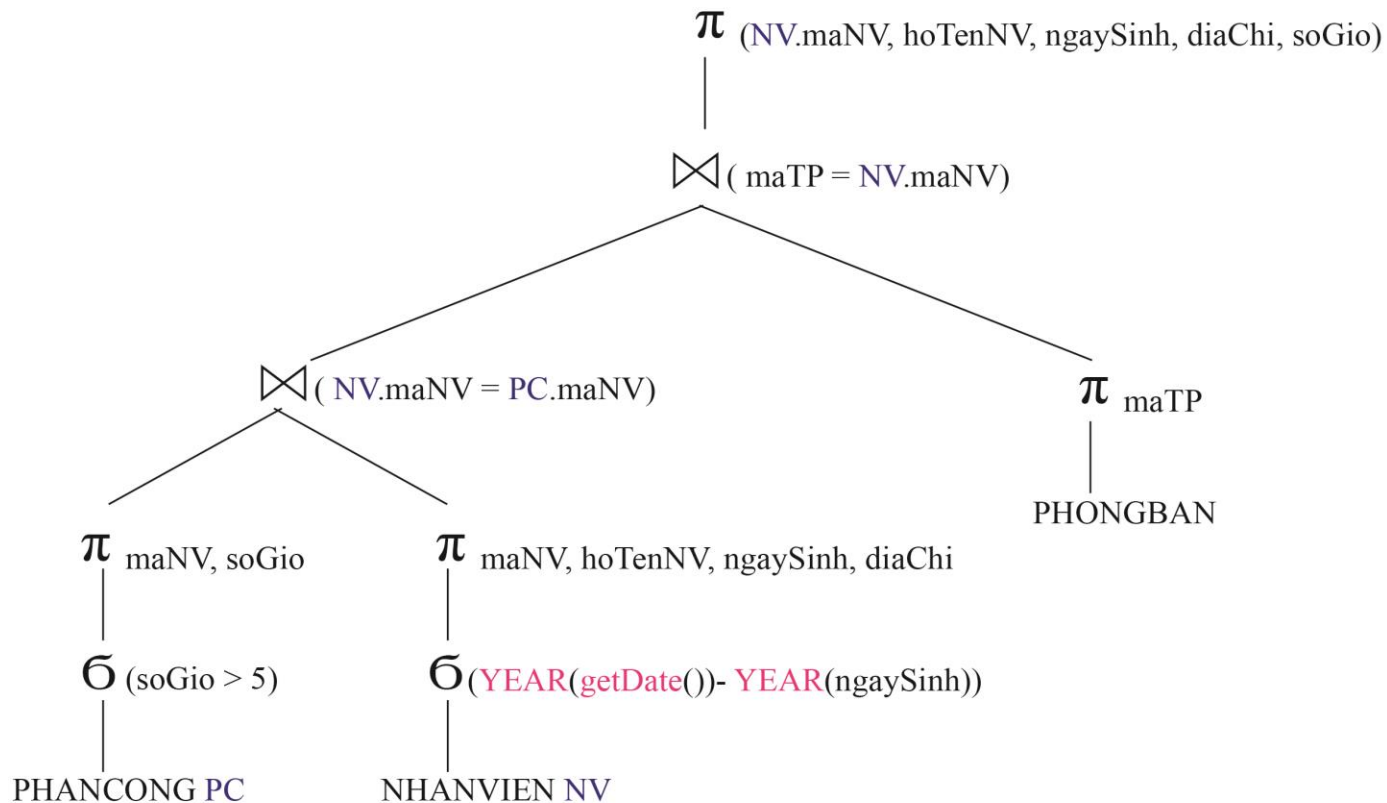
Tối ưu hóa câu truy vấn

LOGO



Tối ưu hóa câu truy vấn

LOGO



Tối ưu hóa câu truy vấn



- ❖ Thiết lập chế độ xem thời gian thực hiện truy vấn
- ❖ SET STATISTICS IO ON
- ❖ SET STATISTICS TIME ON
- ❖ SQLSERVER sẽ trả về thời gian thực thi là theo đơn vị milisecond cho mỗi lần thực thi

Tối ưu hóa câu truy vấn

❖ Điều chỉnh câu truy vấn

- Truy vấn truy cập dữ liệu quá nhiều.
- Chỉ mục được tạo ra nhưng đôi khi không được sử dụng trong câu truy vấn.

Tối ưu hóa câu truy vấn

❖ Điều chỉnh câu truy vấn: Các trường hợp điển hình:

1. Nhiều bộ tối ưu truy vấn (query optimizers) không sử dụng chỉ mục:
 - ❖ Có biểu thức toán học
 - ✓ Ví dụ: **SALARY/365 > 10.50**
 - ❖ So sánh số học của các thuộc tính khác nhau về kích thước.
 - ✓ Ví dụ: **AQTY = BQTY** với AQTY kiểu INTEGER và BQTY kiểu SMALLINTEGER
 - ❖ So sánh NULL
 - ✓ Ví dụ: **BDATE IS NULL**
 - ❖ So sánh chuỗi con
 - ✓ Ví dụ: **LNAME LIKE "%MANN"**

Tối ưu hóa câu truy vấn

LOGO

❖ Điều chỉnh truy vấn

2. Chỉ mục thường không được sử dụng trong những truy vấn lồng nhau.

❖ Ví dụ:

```
SELECT  SSN
FROM    EMPLOYEE
WHERE   DNO IN (SELECT  DNUMBER
                  FROM    DEPARTMENT
                  WHERE   MGRSSN = '333445555');
```

❖ Không sử dụng chỉ mục trên DNO trong EMPLOYEE.

Tối ưu hóa câu truy vấn

Điều chỉnh truy vấn

3. Tối thiểu toán tử DISTINCT.
4. Có thể bỏ những bảng kết quả tạm thời không cần thiết.
5. Trong một số trường hợp, bảng tạm thời có ích.

```
SELECT      SSN
FROM        EMPLOYEE AS E
WHERE        SALARY = SELECT MAX (SALARY)
                        FROM EMPLOYEE AS M
                        WHERE M.DNO = E.DNO
```

➤ Để hiệu quả hơn:

```
SELECT MAX (SALARY) AS HIGHSALARY, DNO INTO TEMP
FROM EMPLOYEE
GROUP BY DNO;
```

```
SELECT SSN
FROM EMPLOYEE, TEMP
WHERE SALARY = HIGHSALARY AND EMPLOYEE.DNO = TEMP.DNO;
```

Tối ưu hóa câu truy vấn



Điều chỉnh truy vấn:

6. Nếu có nhiều lựa chọn điều kiện liên kết chọn điều kiện sử dụng clustering index.
7. Một đặc tính của bộ tối ưu truy vấn là thứ tự của bảng trong mệnh đề FROM có thể ảnh hưởng đến quá trình liên kết.
8. Một số bộ tối ưu truy vấn trong trường hợp truy vấn lồng nhau không tốt bằng truy vấn tương đương không lồng nhau

Tối ưu hóa câu truy vấn

Hướng dẫn thêm về điều chỉnh truy vấn

❖ Truy vấn với nhiều điều kiện chọn kết nối qua OR:

```
SELECT FNAME, LNAME, SALARY, AGE  
FROM EMPLOYEE  
WHERE AGE > 45 OR SALARY < 50000;
```

➤ Để hiệu quả hơn:

```
SELECT FNAME, LNAME, SALARY, AGE  
FROM EMPLOYEE  
WHERE AGE > 45  
UNION  
SELECT FNAME, LNAME, SALARY, AGE  
FROM EMPLOYEE  
WHERE SALARY < 50000;
```

Tối ưu hóa câu truy vấn

Hướng dẫn thêm về điều chỉnh truy vấn

- ❖ Điều kiện trong mệnh đề **WHERE** có thể được viết lại để sử dụng chỉ mục trên nhiều cột.

```
SELECT REGION#, PRODUCT__TYPE, MONTH, SALES  
FROM SALES_STATISTICS  
WHERE REGION# = 3 AND  
((PRODUCT__TYPE BETWEEN 1 AND 3) OR (PRODUCT__TYPE  
    BETWEEN 8 AND 10));
```

➤ Để hiệu quả hơn:

```
SELECT REGION#, PRODUCT__TYPE, MONTH, SALES  
FROM SALES_STATISTICS  
WHERE (REGION# = 3 AND  
(PRODUCT__TYPE BETWEEN 1 AND 3)) OR  
(REGION# = 3 AND (PRODUCT__TYPE BETWEEN 8 AND 10));
```


Tối ưu hóa câu truy vấn

❖ Chú ý:

- Thứ tự điều kiện trong Where: Chọn điều kiện trả về bản ghi ít nhất sắp xếp ở thứ tự trước, các điều kiện trả về nhiều bản ghi nhất sắp xếp ở cuối thứ tự.
- Trong biểu thức điều kiện nên thay toán tử OR bằng IN. Bộ dịch SQL trả về nhanh hơn với IN.
- Thứ tự lựa chọn các bảng trong JOIN hoặc FROM có ảnh hưởng tới tốc độ truy vấn: chọn bảng nhỏ nhất trước, bảng lớn nhất sau.
- Nên sử dụng các toán tử thay thế kí tự trong biểu thức sau LIKE hợp lý.

Tối ưu hóa câu truy vấn

❖ Chú ý:

- Hạn chế dùng mệnh đề HAVING
- Tránh các câu lệnh sắp xếp ORDER BY đối với lượng dữ liệu lớn vì sẽ bắt bộ biên dịch SQL phải mất thời gian sắp xếp khi trả về kết quả.
- Không nên dùng View thay cho bảng gốc
- Trigger không nên sử dụng nếu không cần thiết. (Thay xử lý trong trigger bằng thủ tục)
- Chỉ Select những cột cần thiết, hạn chế Select *
- Hạn chế sử dụng con trỏ

Bài tập

- ❖ Bài 1: Thực hiện truy vấn dữ liệu trên “Hệ thống quản lý bán sách online” theo các kịch bản khác nhau đưa ra so sánh thời gian thực hiện. (thực hiện lặp 1000 lần – hạn chế vấn đề ngẫu nhiên)
- ❖ Bài 2: Thực hiện truy vấn dữ liệu trên “Hệ thống quản lý đào tạo LQDUNI” theo các kịch bản khác nhau đưa ra so sánh thời gian thực hiện. (thực hiện lặp 1000 lần – hạn chế vấn đề ngẫu nhiên)

Bài tập

- ❖ Bài 3: Thực hiện cập nhật “Hệ thống quản lý bán sách online” theo kịch bản có trigger, stored procedure, các câu lệnh insert trực tiếp.
- ❖ Bài 2: Thực hiện cập nhật “Hệ thống quản lý bán sách online” theo kịch bản có trigger, stored procedure, các câu lệnh insert trực tiếp.