VI

Xử lý Câu truy vấn

Nội dung chi tiết

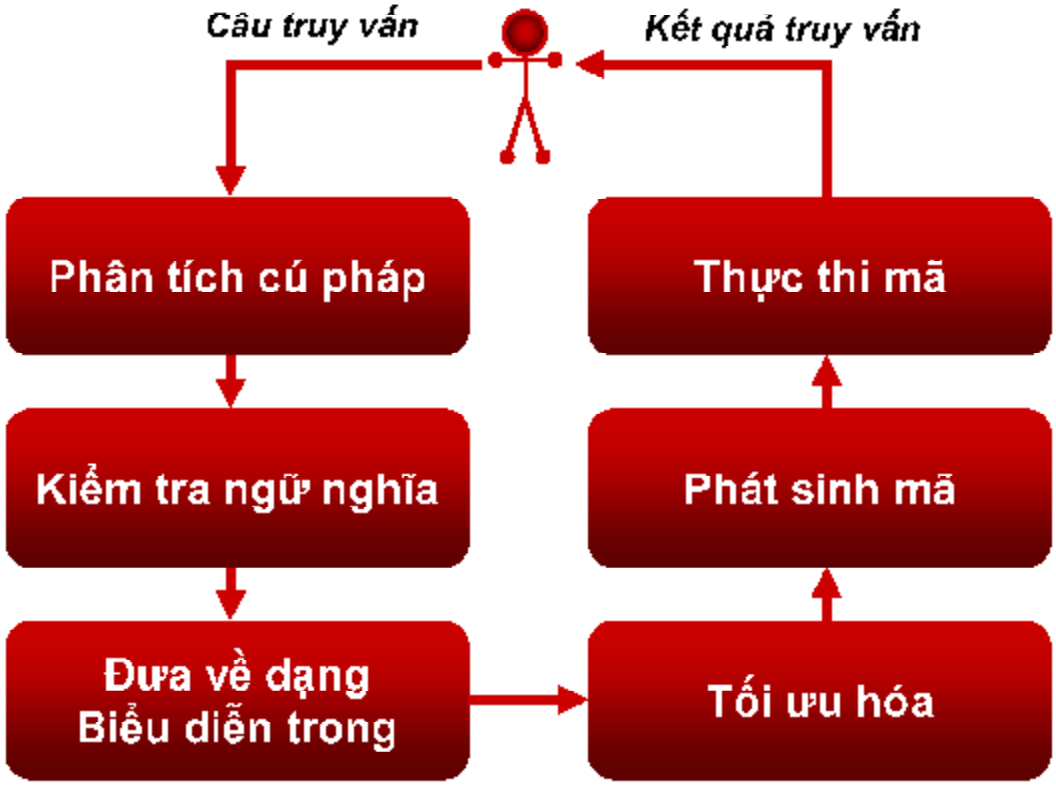
- *Giới thiệu*
- *Phân tích cú pháp - ngữ nghĩa*
- *Biến đổi sang Đại số Quan hệ*
- *Tối ưu hóa cây truy vấn*
- *Ước lượng kích thước cây truy vấn*
- *Phát sinh và thực thi mã lệnh*

- ***Xét hai quan hệ R và S như sau :***
  - R(A, B, C)
  - S(C, D, E)
- ***Xét câu truy vấn sau đây trên R và S***
  - Select R.B, S.D
    From R, S
    Where R.A='c' And S.E=2 And R.C=S.C
- ***Nhận xét***
  - Một câu truy vấn có rất nhiều cách thực hiện
  - Tùy trường hợp mà các cách thực hiện được đánh giá là tốt hay dở

Giới thiệu (tt)

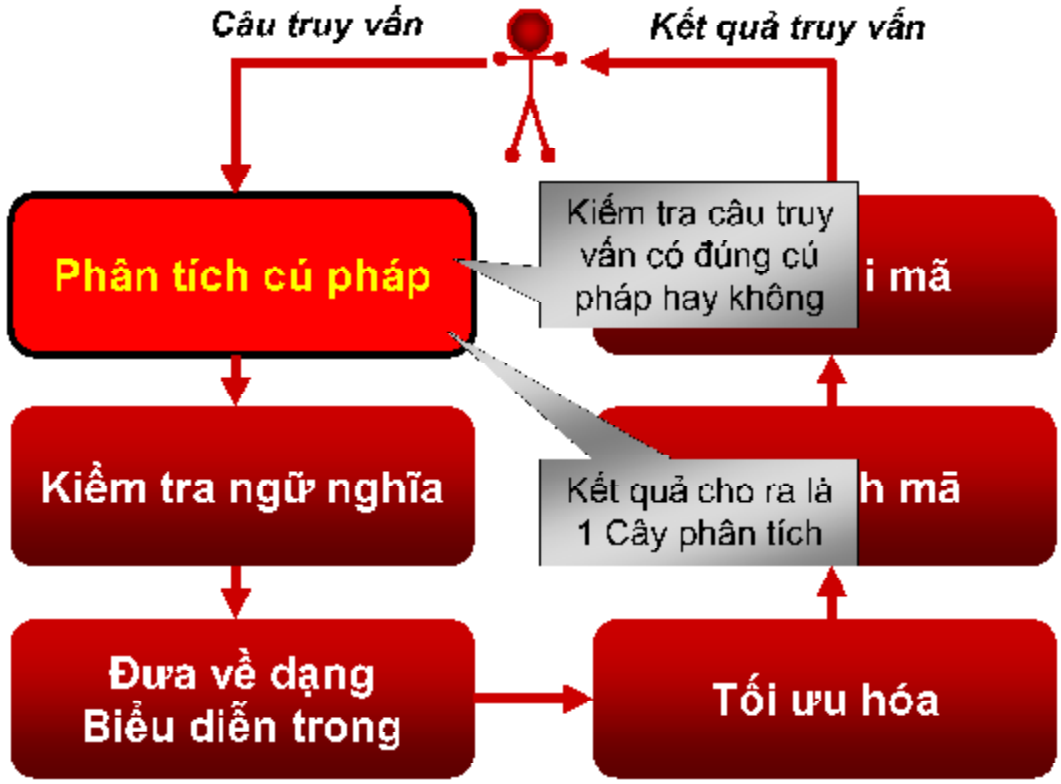
- ***Xử lý của DBMS***
  - Cách 1 :
    - $\Pi_{B,D}\,[\,\sigma_{R.A='c' \wedge S.E=2 \wedge R.C = S.C}\,(R \times S)]$
  - Cách 2 :
    - $\Pi_{B,D}\,[\,\sigma_{R.A='c'}\,(R) \bowtie \sigma_{S.E=2}\,(S)]$
  - Cách 3 : Sử dụng chỉ mục trên R.A và S.C
    - Tìm các bộ trong R thỏa R.A='c'
    - Với mỗi bộ tìm thấy, tìm tiếp các bộ trong S thỏa R.C=S.C
    - Bỏ đi những bộ S.E ≠ 2
    - Chiếu trên thuộc tính B và D
- ***DBMS chọn cách nào ?***
- ***Chương này tập trung vào xử lý truy vấn của RDBMS***

- ***Quy trình xử lý câu truy vấn***

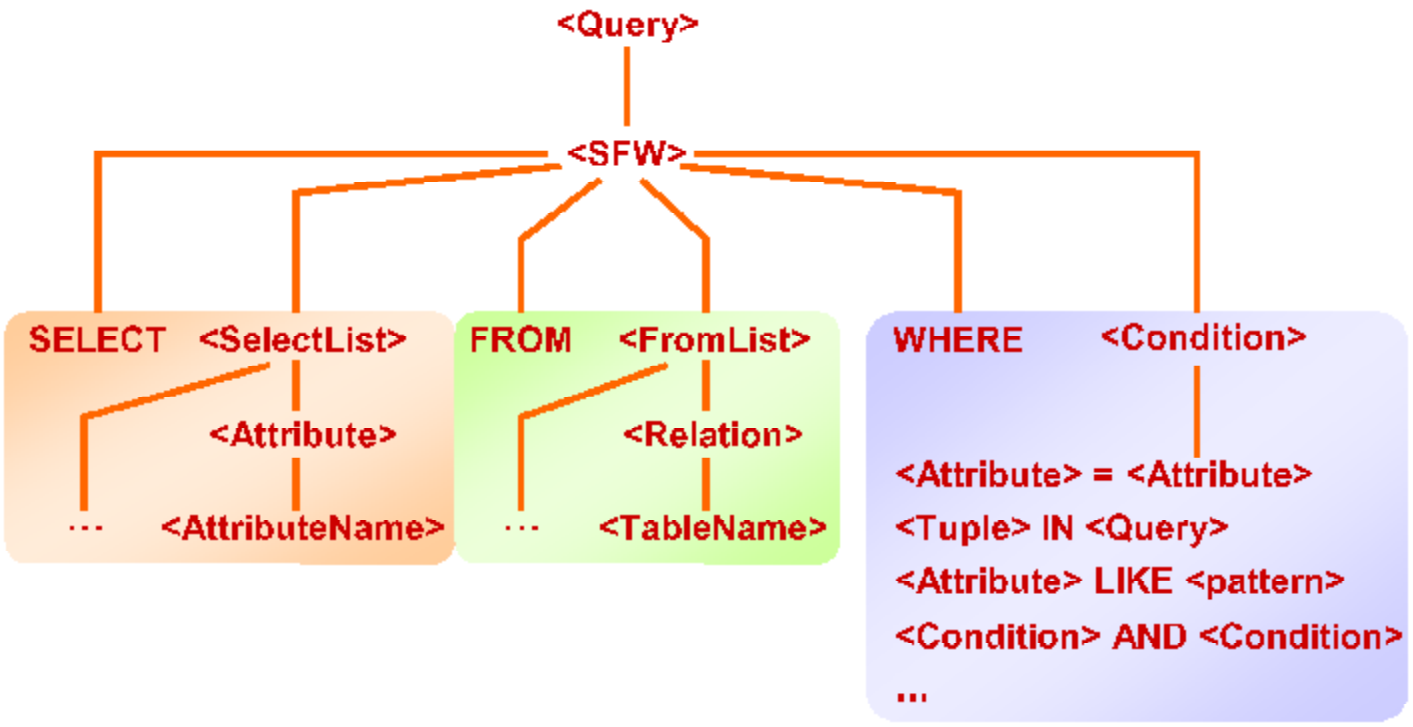


Nội dung chi tiết

- ***Giới thiệu***
- ***Phân tích cú pháp - ngữ nghĩa***
- ***Biến đổi sang Đại số Quan hệ***
- ***Tối ưu hóa cây truy vấn***
- ***Ước lượng kích thước cây truy vấn***
- ***Phát sinh và thực thi mã lệnh***



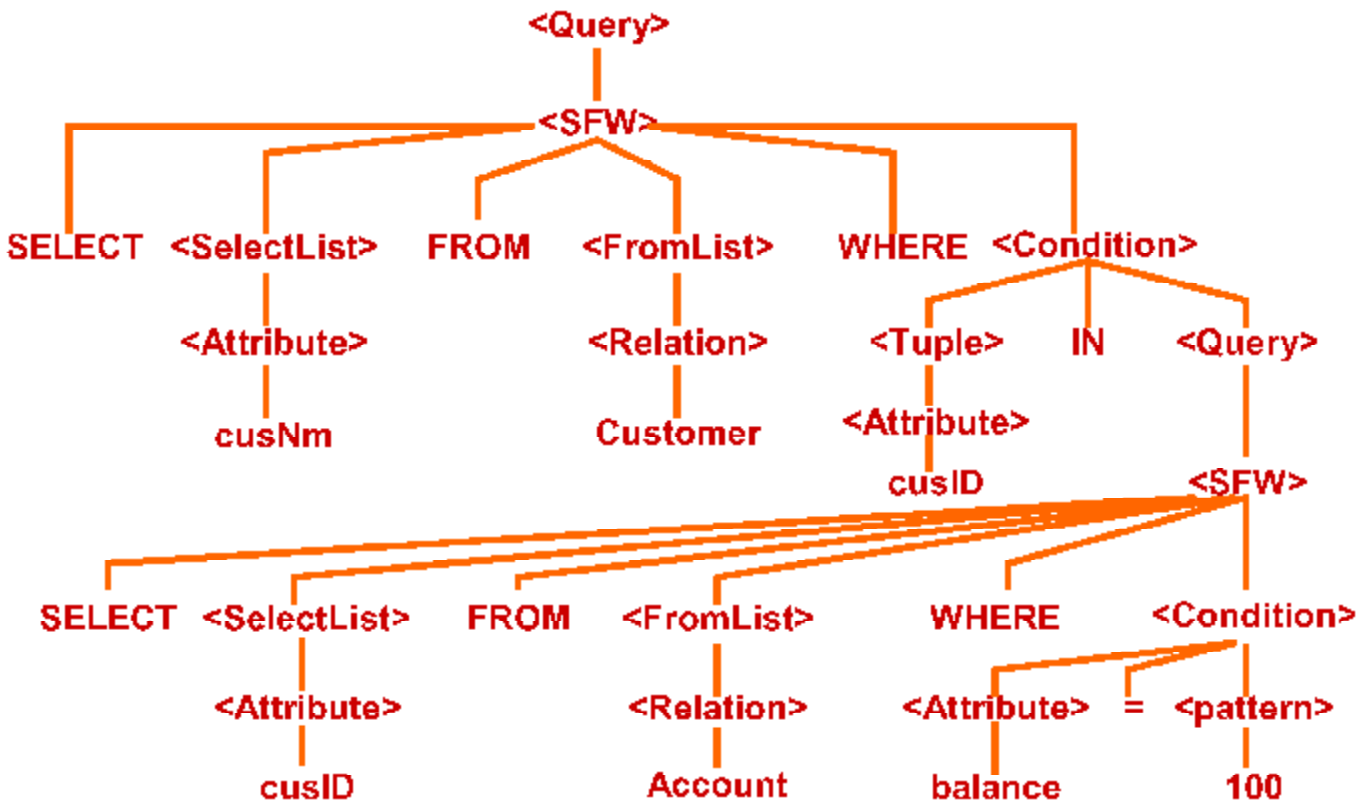
Phân tích cú pháp và ngữ nghĩa (tt)



- ***Xét hai quan hệ sau :***
  - Customer(cusID, cusNm, cusStreet, cusCity)
  - Account(accID, cusID, balance)
- ***Và câu truy vấn***

```
SELECT cusNm
FROM Customer
WHERE cusID IN (
        SELECT cusID
        FROM Account
        WHERE balance = 100)
```

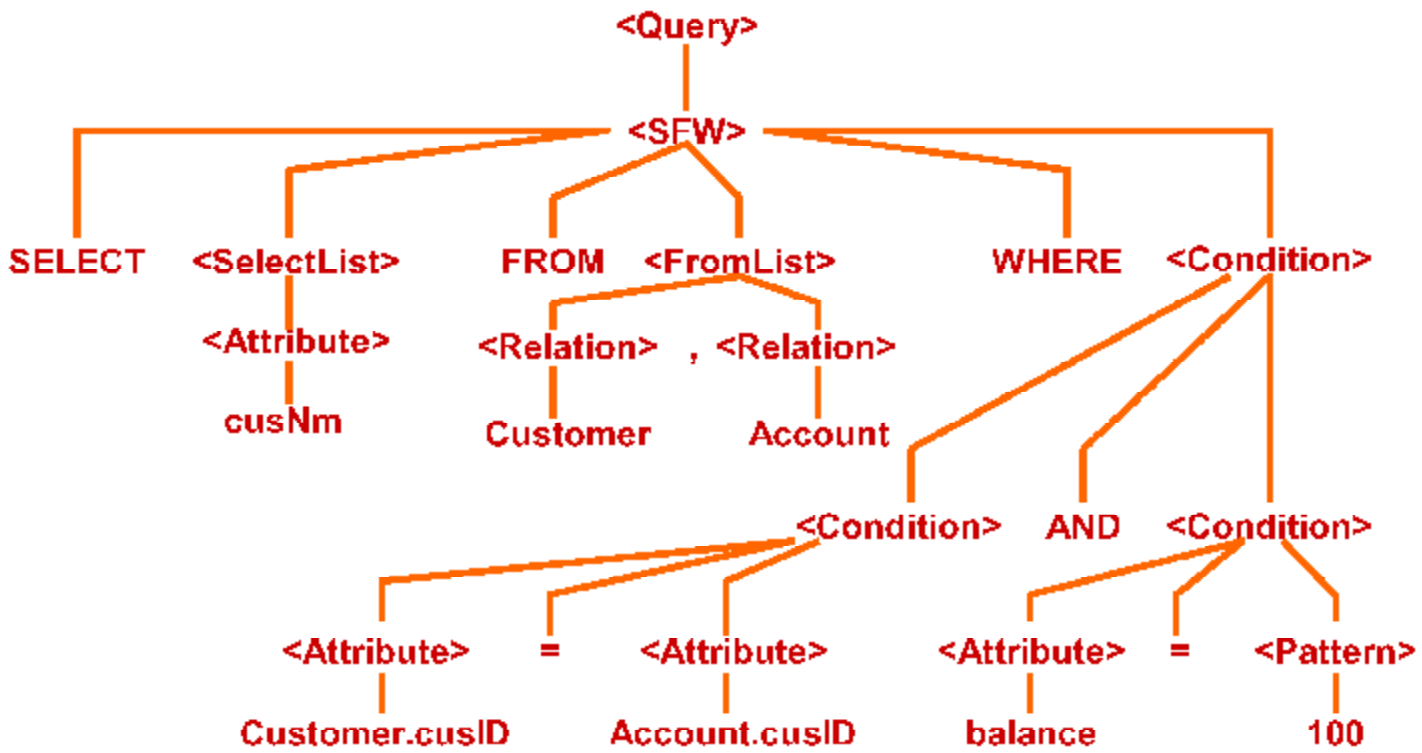
Ví dụ 1 (tt)



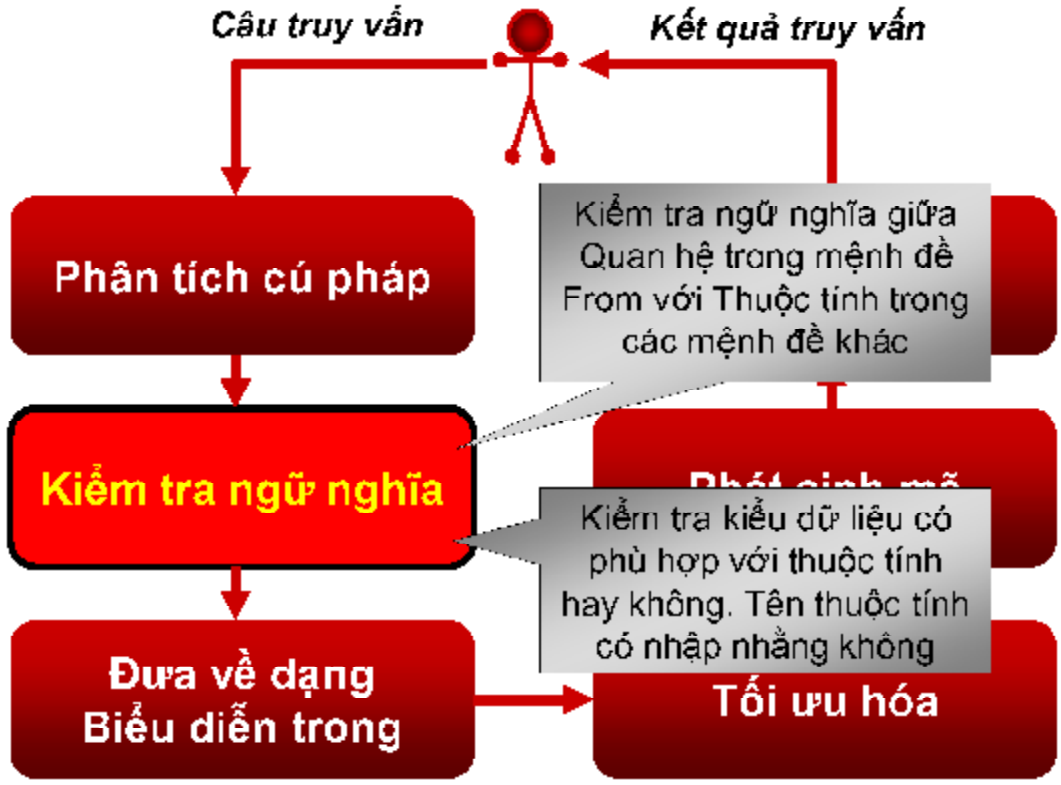
- ***Xét hai quan hệ sau đây :***
  - Customer(cusID, cusNm, cusStreet, cusCity)
  - Account(acclD, cusID, balance)
- ***Và câu truy vấn trên hai quan hệ ấy***

```
SELECT cusNm
FROM Customer, Account
WHERE Customer.cusID = Account.cusID
AND balance = 100
```

Ví dụ 2 (tt)

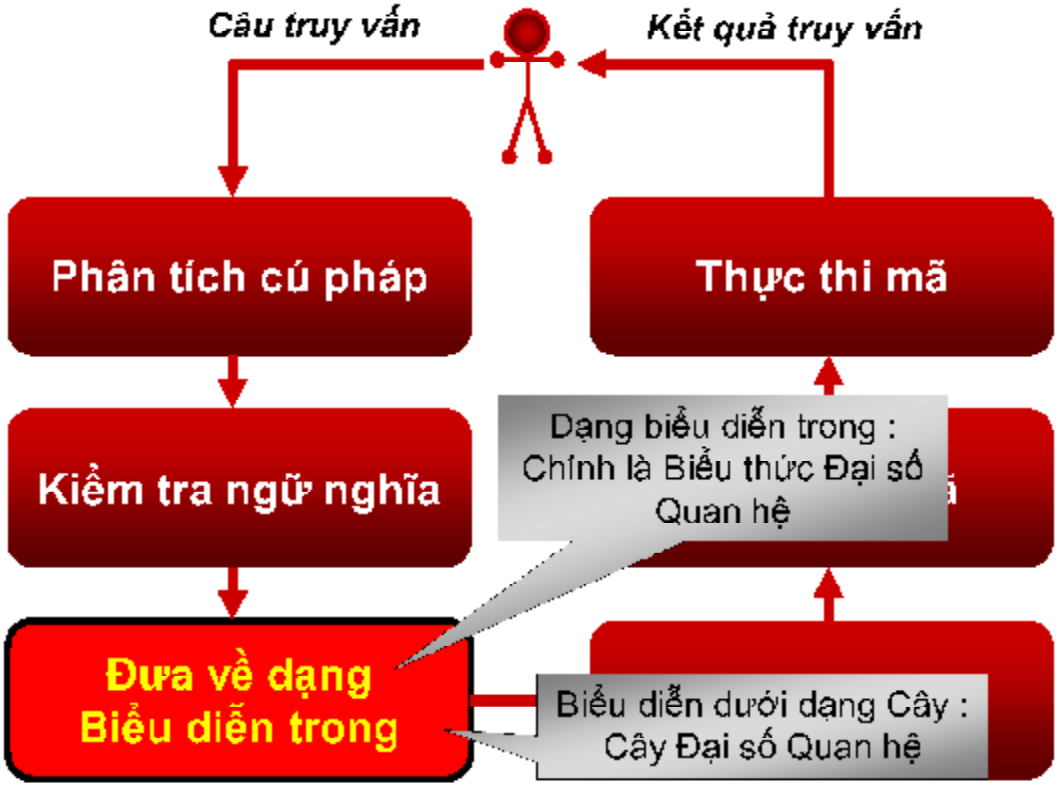


- ***Sau khi có cây phân tích, DBMS kiểm tra ngữ nghĩa***



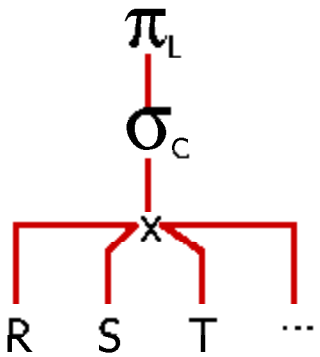
Nội dung chi tiết

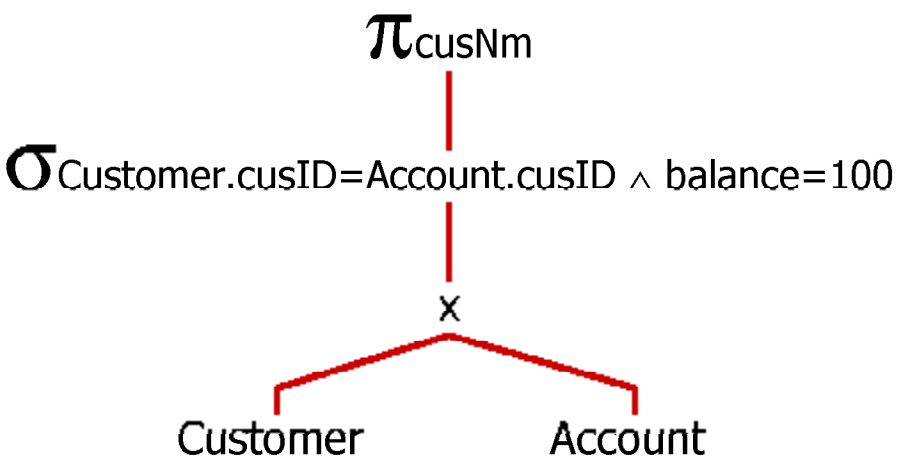
- ***Giới thiệu***
- ***Phân tích cú pháp - ngữ nghĩa***
- ***Biến đổi sang Đại số Quan hệ***
- ***Tối ưu hóa cây truy vấn***
- ***Ước lượng kích thước cây truy vấn***
- ***Phát sinh và thực thi mã lệnh***



Biến đổi sang ĐSQH (tt)

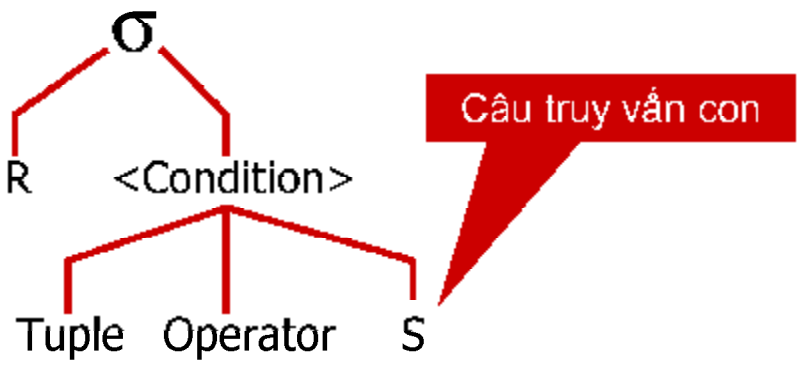
- ***Truy vấn đơn***
  - Xét câu trúc <SFW>, sử dụng quy tắc <SFW>
    - Thay thế <FromList> thành các biến quan hệ
      - *Sử dụng phép tích cartesian cho các biến quan hệ*
    - Thay thế <Condition> thành phép chọn σ_C
    - Thay thế <SelectList> thành phép chiếu π_L
  - Kết quả lá một Cây truy vấn

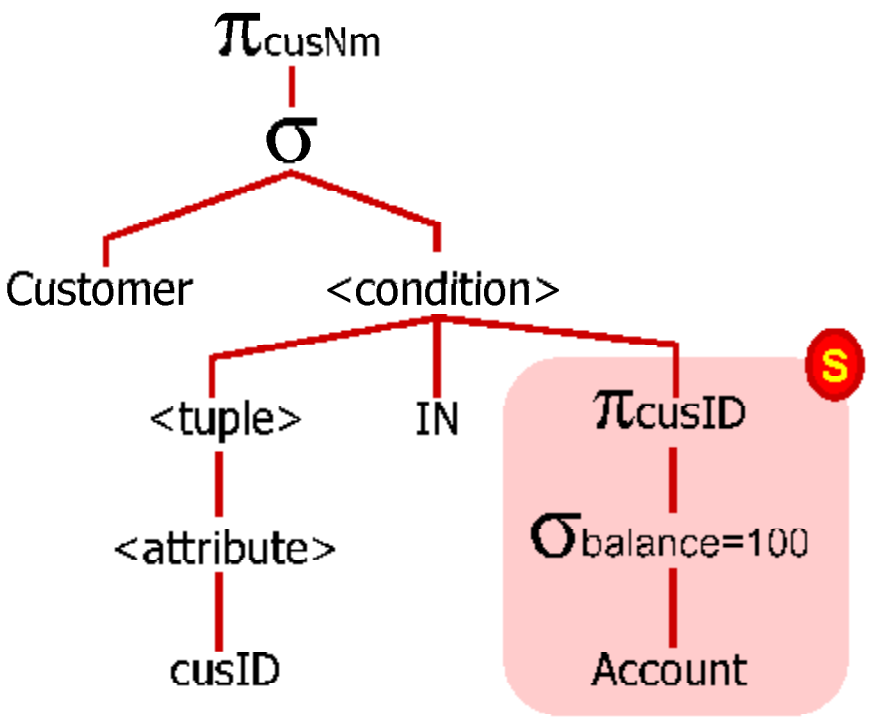




Biến đổi sang ĐSQH (tt)

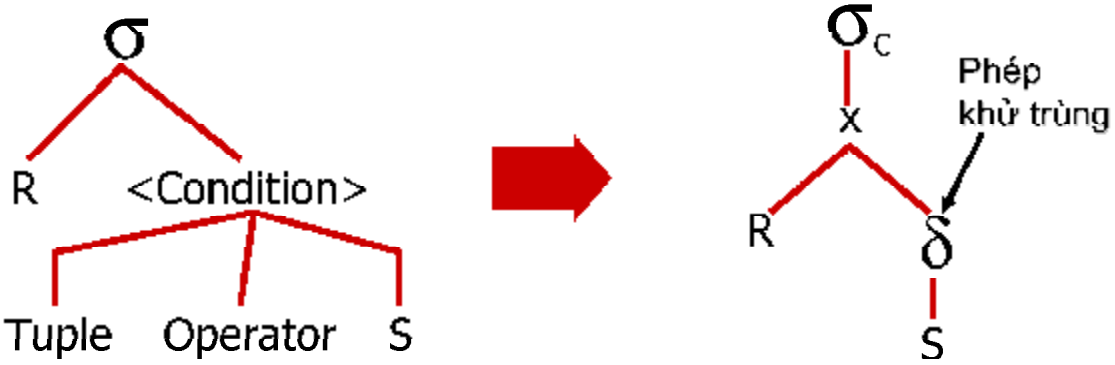
- ***Truy vấn lồng***
  - Tồn tại câu truy vấn con S trong <Condition>
  - Áp dụng qui tắc <SFW> cho truy vấn con S
  - Phép chọn 2 biến (two-argument selection)
    - Nút là phép chọn không có tham số
    - Nhánh con trái là biến quan hệ R
    - Nhánh con phải là <condition> áp dụng cho **<u>mỗi bộ trong R</u>**

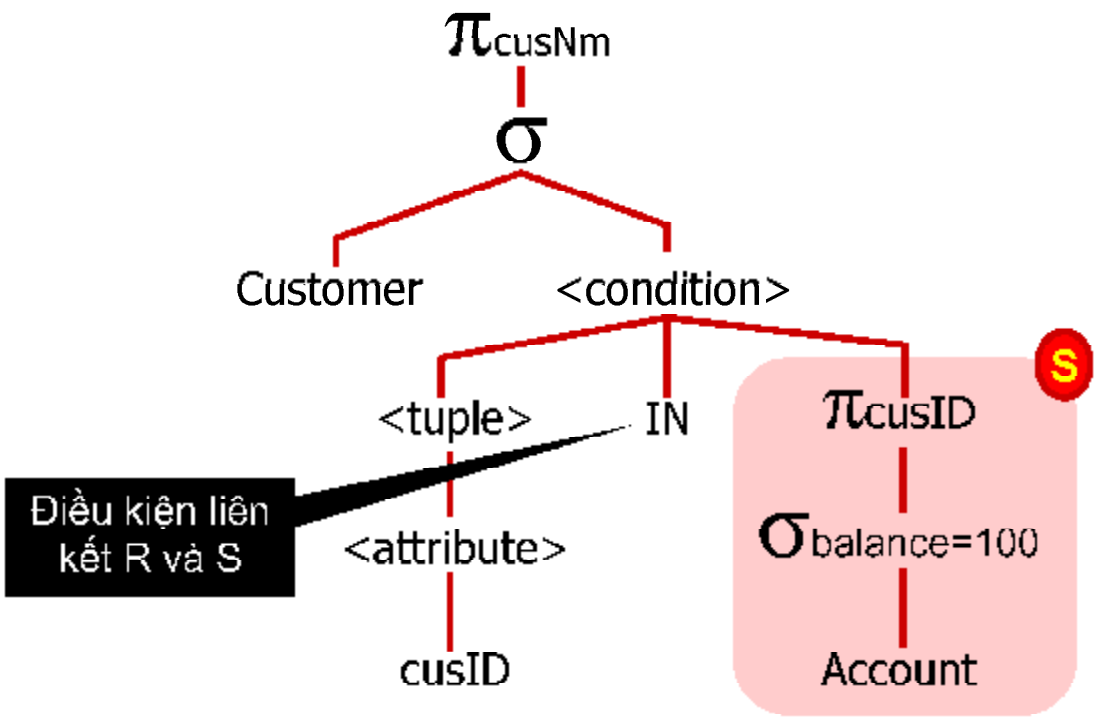




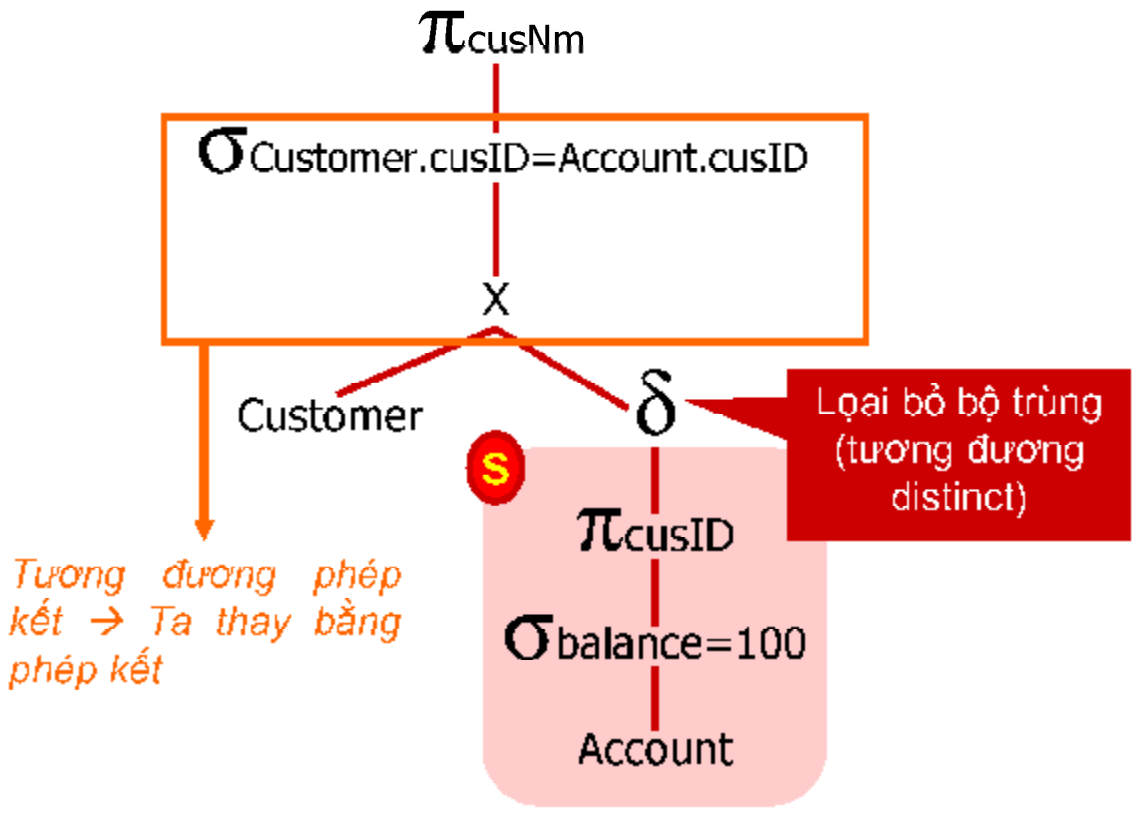
Biến đổi sang ĐSQH (tt)

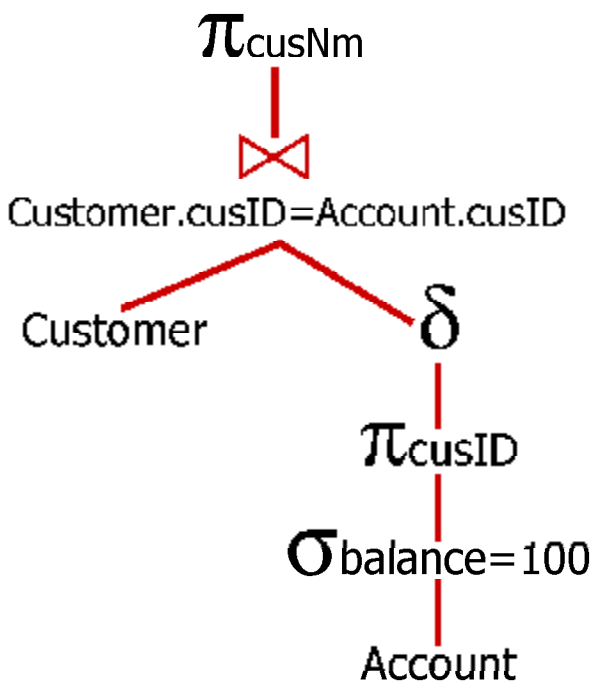
- ***Truy vấn lồng***
  - Biến đổi phép chọn 2 biến
    - Thay thế <Condition> bằng 1 cây có ngọn là S
      - *Nếu S có các bộ trùng nhau thì phải lược bỏ bớt bộ trùng nhau đi. Sử dụng phép δ để lược bỏ (giống Distinct)*
    - Thay thế phép chọn 2 biến thành σ_C với C là điều kiện liên kết (không đơn thuần là kết) R với S
    - σ_C làm trên kết quả của phép cartesian của R và S




πcusNm
σ
Customer
<condition>
<tuple>
IN
Điều kiện liên kết R và S
<attribute>
cusID
S
πcusID
σbalance=100
Account

Xét ví dụ 1 (tt)

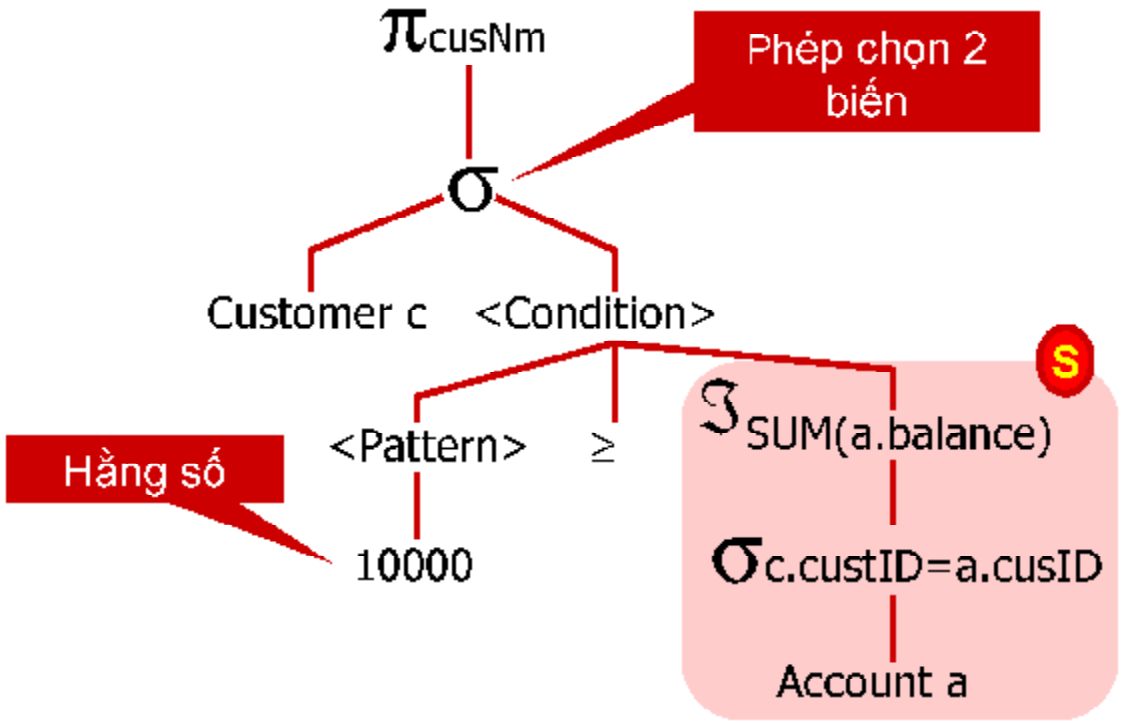

πcusNm
σCustomer.cusID=Account.cusID
X
Customer
δ
Loại bỏ bộ trùng (tương đương distinct)
S
πcusID
σbalance=100
Account
Tương đương phép kết → Ta thay bằng phép kết



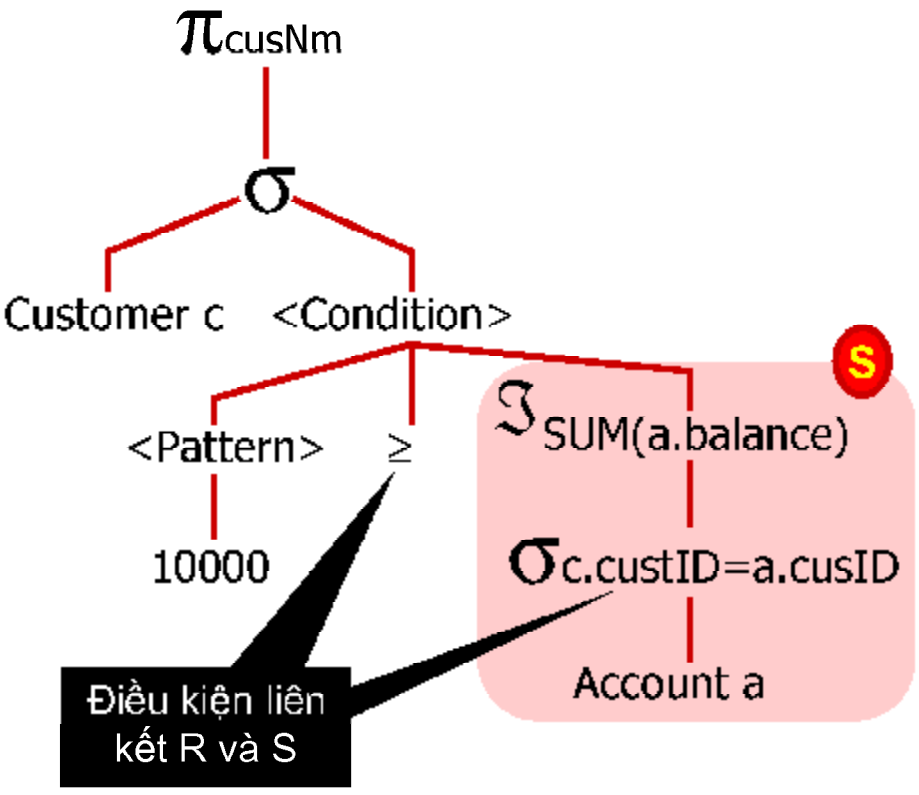
Ví dụ 3 (Lồng tương quan)

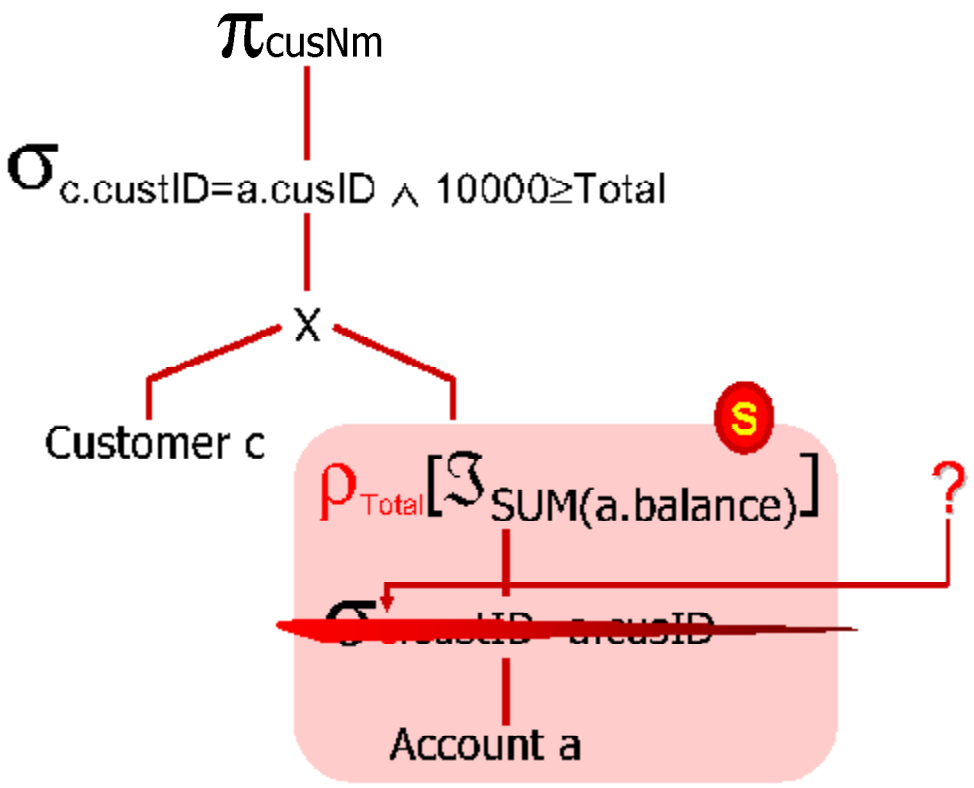
- ***Xét hai quan hệ sau đây :***
  - Customer(cusID, cusNm, cusStreet, cusCity)
  - Account(acclD, cusID, balance)
- ***Xét câu truy vấn sau đây :***

```
SELECT c.cusNm
FROM Customer c
WHERE 10000 >= (
        SELECT SUM(a.balance)
        FROM Account a
        WHERE a.cusID=c.cusID)
```


πcusNm
Phép chọn 2 biến
σ
Customer c
<Condition>
<Pattern>
≥
Hằng số
10000
S
ℑSUM(a.balance)
σc.custID=a.cusID
Account a

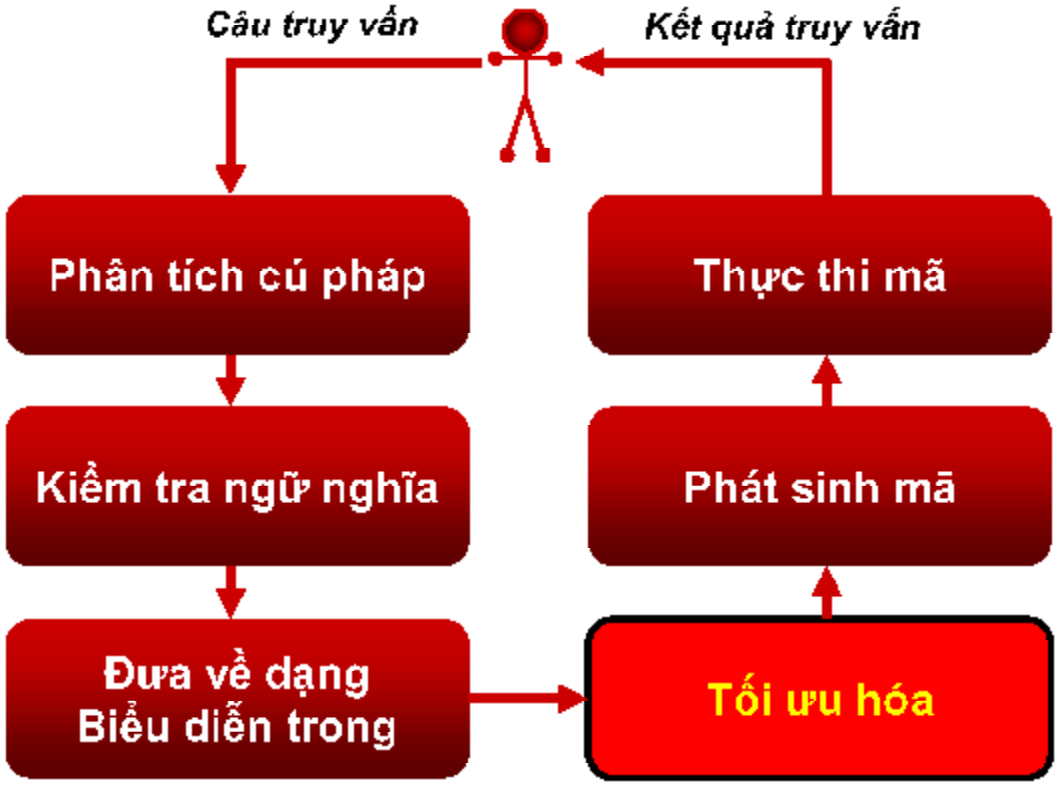
Ví dụ 3 (tt)


πcusNm
σ
Customer c
<Condition>
<Pattern>
≥
10000
S
ℑSUM(a.balance)
σc.custID=a.cusID
Account a
Điều kiện liên kết R và S



Nội dung chi tiết

- *Giới thiệu*
- *Phân tích cú pháp - ngữ nghĩa*
- *Biến đổi sang Đại số Quan hệ*
- *Tối ưu hóa cây truy vấn*
- *Ước lượng kích thước cây truy vấn*
- *Phát sinh và thực thi mã lệnh*



Tối ưu hóa cây truy vấn (tt)

- ***Chiến lược tối ưu hóa***
  - Chiến lược
    - Tốc độ thức thi câu truy vấn nhanh nhất có thể
    - Việc xử lý câu truy vấn chiếm dụng bộ nhớ ít nhất có thể
  - Nhận xét
    - Hai yêu cầu trên mâu thuẫn nhau
    - Cần phải dung hòa, thỏa hiệp
- ***Chiến thuật***
  - Thực hiện các phép toán quan hệ 1 ngôi trước (nếu có thể)
  - Sau đó thực hiện các phép toán 2 ngôi và các phép toán 1 ngôi còn lại

- ***Qui tắc: Kết tự nhiên, tích cartesian, hội***
  - $R \times S = S \times R$
  - $(R \times S) \times T = R \times (S \times T)$
  - $R \bowtie S = S \bowtie R$
  - $(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$
  - $R \cup S = S \cup R$
  - $R \cup (S \cup T) = (R \cup S) \cup T$

Áp dụng quy tắc (tt)

- ***Qui tắc Phép chọn σ***
  - Cho
    - p là vị từ chỉ có các thuộc tính của R
    - q là vị từ chỉ có các thuộc tính của S
    - m là vị từ có các thuộc tính của R và S
  - $\sigma_{p1 \wedge p2}(R) = \sigma_{p1}[\sigma_{p2}(R)]$
  - $\sigma_{p1 \vee p2}(R) = [\sigma_{p1}(R)] \cup [\sigma_{p2}(R)]$

Quan hệ R là tập hợp
∪ là phép hội trên tập hợp

Áp dụng quy tắc (tt)

- ***Qui tắc: σ, ⋈***
  - $\sigma_p (R \bowtie S) = [\sigma_p(R)] \bowtie S$
  - $\sigma_q (R \bowtie S) = R \bowtie [\sigma_q (S)]$
  - $\sigma_{p \wedge q} (R \bowtie S) = [\sigma_p (R)] \bowtie [\sigma_q (S)]$
  - $\sigma_{p \wedge q \wedge m} (R \bowtie S) = \sigma_m [\sigma_p(R) \bowtie \sigma_q (S)]$
  - $\sigma_{p \vee q} (R \bowtie S) = [\sigma_p (R) \bowtie S] \cup [R \bowtie \sigma_q (S)]$

p là điều kiện chỉ liên quan thuộc tính của R và q là điều kiện chỉ liên quan thuộc tính của S

m là điều kiện liên quan thuộc tính của R và S

- ***Qui tắc: σ, ∪ và σ, –***
  - $\sigma_c(R \cup S) = \sigma_c(R) \cup \sigma_c(S)$
  - $\sigma_c(R - S) = \sigma_c(R) - S = \sigma_c(R) - \sigma_c(S)$

- ***Qui tắc: Phép chiếu π***
  - Cho
    - X = tập thuộc tính con của R
    - Y = tập thuộc tính con của R
  - Ta có
    - $XY = X \cup Y$
  - Ta KHÔNG có
    - $\pi_{XY}(R) = \pi_X[\pi_Y(R)]$

Áp dụng quy tắc (tt)

- ***Qui tắc: π, ⋈***
  - Cho
    - X = tập thuộc tính con của R
    - Y = tập thuộc tính con của S
    - Z = tập giao thuộc tính của R và S
  - Ta có
    - $\pi_{XY}(R \bowtie S) = \pi_{XY}[\pi_{XZ}(R) \bowtie \pi_{YZ}(S)]$

- ***Qui tắc: σ, π***
  - Cho
    - X = tập thuộc tính con của R
    - Z = tập thuộc tính con của R xuất hiện trong vị từ p
  - Ta có
    - $\pi_X [\sigma_p (R)] = \pi_X \{\sigma_p [\pi_{XZ} (R)]\}$

Áp dụng quy tắc (tt)

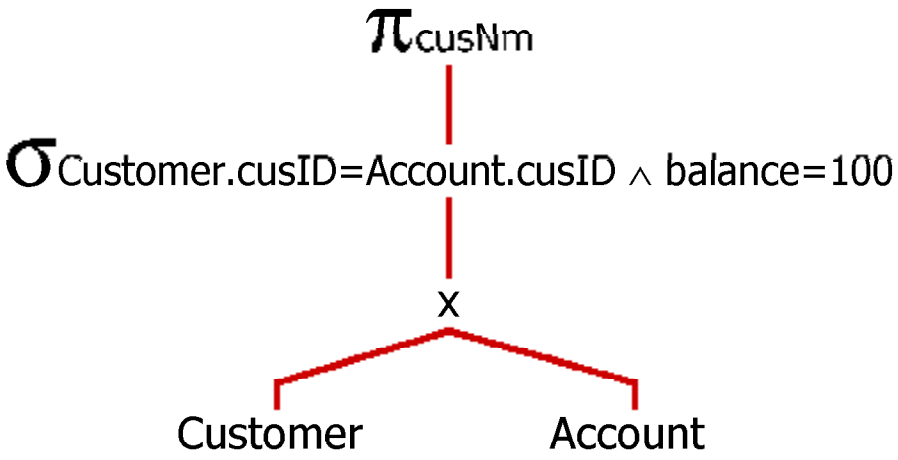
- ***Qui tắc: σ, π, ⋈***
  - Cho
    - X = tập thuộc tính con của R
    - Y = tập thuộc tính con của S
    - Z = tập giao thuộc tính của R và S
    - Z' = Z ∪ {các thuộc tính xuất hiện trong vị từ p}
  - Ta có
    - $\pi_{XY} [\sigma_p (R \bowtie S)] = \pi_{XY} \{\sigma_p [\pi_{XZ'} (R) \bowtie \pi_{YZ'} (S)]\}$

- ***Qui tắc: ×, ⋈***
  - $\sigma_C (R \bowtie S) = R \bowtie_C S$
  - $R \times S = \pi_L [\sigma_C (R \times S)]$

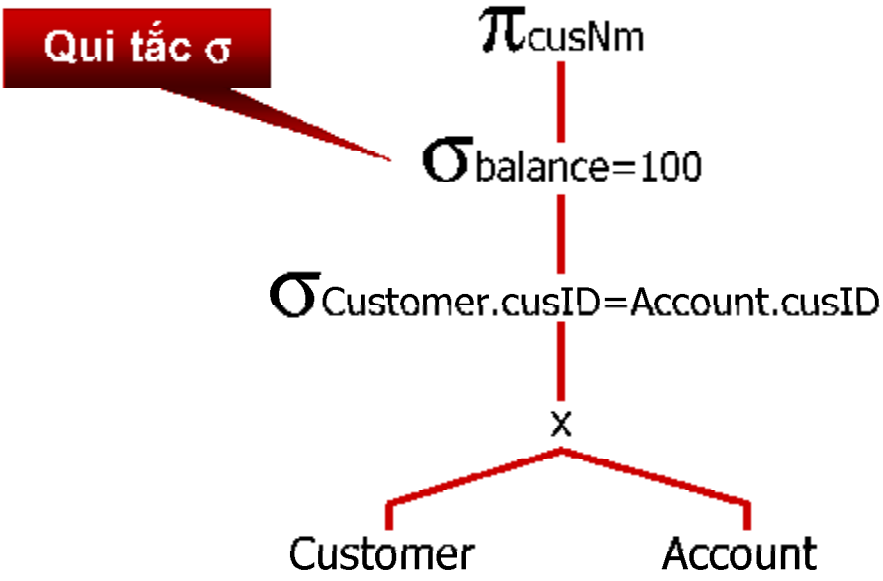
- ***Qui tắc: δ***
  - $\delta(R \bowtie S) = \delta(R) \bowtie \delta(S)$
  - $\delta(R \times S) = \delta(R) \times \delta(S)$
  - $\delta[\sigma_C (R)] = \sigma_C [\delta(R)]$
  - $\delta(R \cap_B S) = \delta(R) \cap_B S = R \cap_B \delta(S) = \delta(R) \cap_B \delta(S)$

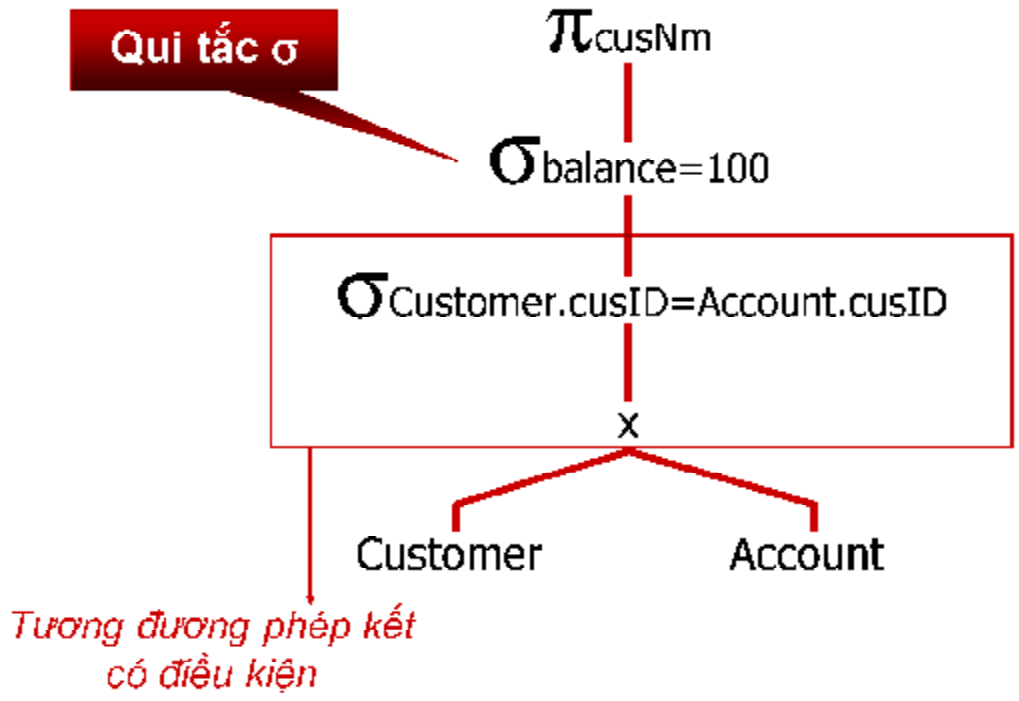
Áp dụng quy tắc (tt)

- ***Qui tắc ℑ***
  - Cho
    - X = tập thuộc tính trong R được gom nhóm
    - Y = X ∪ {một số thuộc tính khác của R}
  - Ta có
    - $\delta[{}_X\Im(R)] = {}_X\Im(R)$
    - ${}_X\Im(R) = {}_X\Im [\pi_Y (R)]$

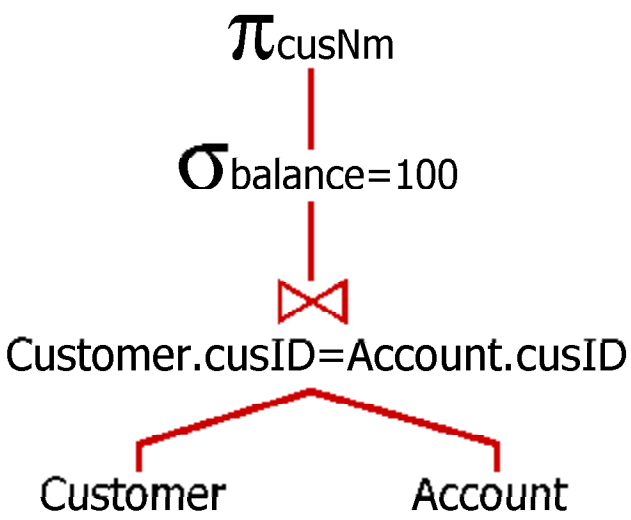

πcusNm
σCustomer.cusID=Account.cusID ∧ balance=100
x
Customer
Account

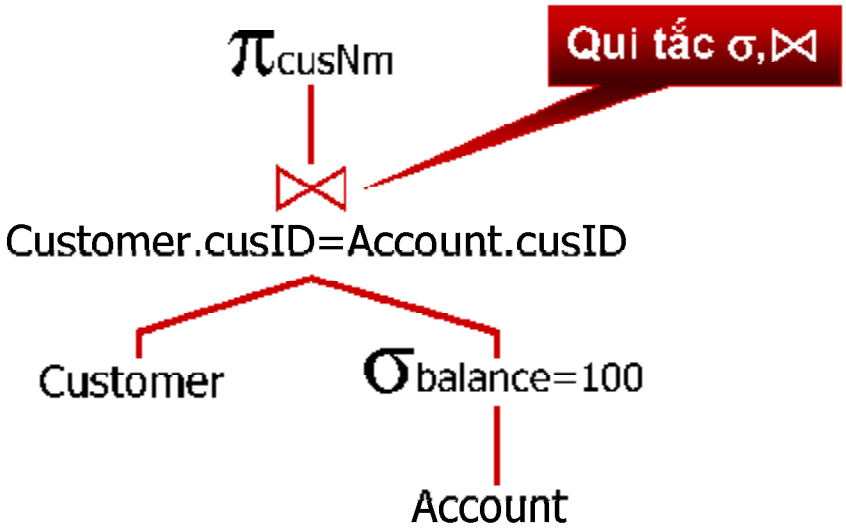
Xét ví dụ 2


Qui tắc σ
πcusNm
σbalance=100
σCustomer.cusID=Account.cusID
x
Customer
Account

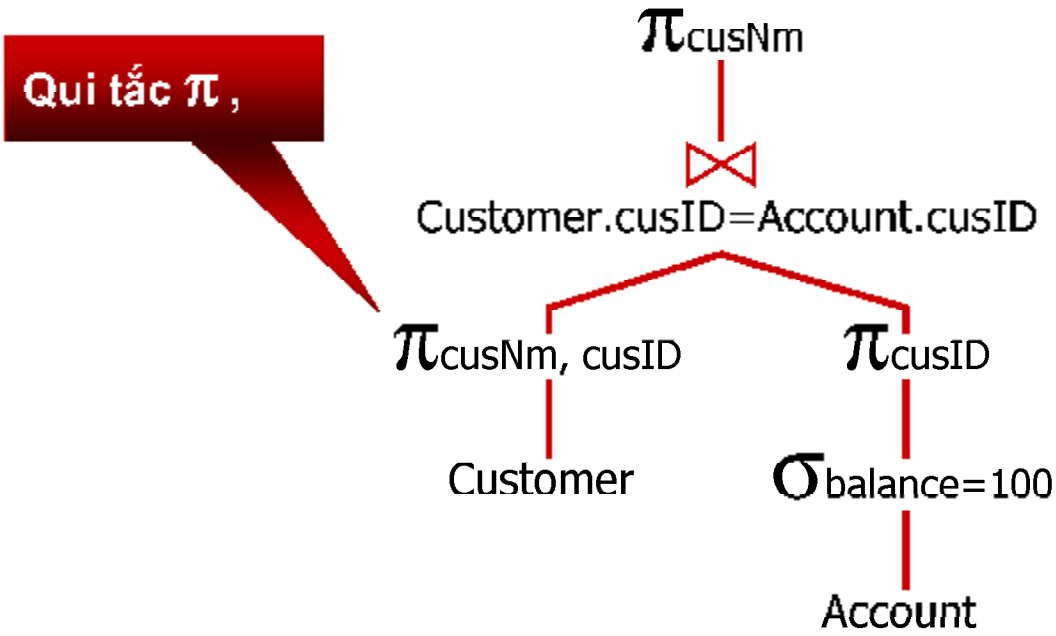

Qui tắc σ
πcusNm
σbalance=100
σCustomer.cusID=Account.cusID
x
Customer
Account
Tương đương phép kết
có điều kiện

Xét ví dụ 2 (tt)


πcusNm
σbalance=100
⋈
Customer.cusID=Account.cusID
Customer
Account


πcusNm
Qui tắc σ,⋈
Customer.cusID=Account.cusID
Customer
σbalance=100
Account

Xét ví dụ 2 (tt)


Qui tắc π ,
πcusNm
Customer.cusID=Account.cusID
πcusNm, cusID
πcusID
Customer
σbalance=100
Account

- *Giới thiệu*
- *Phân tích cú pháp - ngữ nghĩa*
- *Biến đổi sang Đại số Quan hệ*
- *Tối ưu hóa cây truy vấn*
- *Ước lượng kích thước cây truy vấn*
- *Phát sinh và thực thi mã lệnh*

Ước lượng kích thước cây truy vấn

- ***Trong quá trình tối ưu hóa câu truy vấn, có thể có nhiều giải pháp khác nhau***
  - Các giải pháp này ngang nhau về mặt chiến thuật tối ưu hóa
  - Chỉ được chọn 1 giải pháp để thực thi
  - Việc lựa chọn không được thực hiện theo cảm tính
- ***Do đó, cần một cách đánh giá bằng định lượng → Ước lượng kích thước cây truy vấn***
  - Cây truy vấn A tốt hơn cây truy vấn B khi kích thước A nhỏ hơn kích thước B
  - Cây truy vấn được chọn để thực thi là cây truy vấn có kích thước nhỏ nhất trong các ứng viên

- ***Thống kê quan hệ R***
  - T(R): số bộ trong R
  - S(R): tổng số byte của 1 bộ trong R
  - B(R): tổng số block chứa tất cả các bộ của R
  - V(R, A): số giá trị khác nhau mà thuộc tính A trong R có thể có

Ví dụ

- ***Cho quan hệ R như sau***
  - A: chuỗi
  - 20 bytes
  - B: số nguyên 4 bytes
  - C: ngày 8 bytes
  - D: chuỗi 68 bytes
  - 1 block = 1024 bytes
  - block header: 24 bytes
- ***Vậy***
  - T(R) = 5
  - S(R) = 100
  - B(R) = 1
  - V(R, A) = 3, V(R, B) = 1
  - V(R, C) = 5, V(R, D) = 4

R

A	B	C	D
x	1	1	a
x	1	2	b
y	1	3	a
y	1	4	c
z	1	5	d

- *Ước lượng:* $W = R1 \times R2$
  - S(W) = S(R1) + S(R2)
  - T(W) = T(R1) x T(R2)
- *Ước lượng:* $W = \sigma_{Z=val}(R)$
  - S(W) = S(R)
  - T(W) = T(R) / V(R, Z)
- *Ước lượng:* $W = \sigma_{Z=val}(R)$
  - T(W) = T(R) / 2
  - Hoặc T(W) = T(R) / 3

Ví dụ

- *Cho*
  - R(A, B, C)
  - T(R) = 10000
  - V(R, A) = 50
- *Ước lượng kích thước biểu thức* $S = \sigma_{A=10 \wedge B<20}(R)$
  - T(S) = T(R) / [V(R, A) x 3] = 10000 / [50 x 3] = 67
- *Ước lượng kích thước biểu thức* $S = \sigma_{A=10 \vee B<20}(R)$
  - Giả sử :
    - m1 là số bộ thỏa A=10 trong R
    - m2 là số bộ thỏa B<20 trong R
    - Đặt n = T(R)
  - T(S) = n[1 - (1 - m1/n)(1 – m2/n)]

- ***Ước lượng:*** $W = R1 \bowtie R2$
- ***Cho***
  - X = tập thuộc tính của R1
  - Y = tập thuộc tính của R2
- ***Xét trường hợp*** $X \cap Y = \varnothing$
  - Tương tự W = R1 x R2
- ***Xét trường hợp*** $X \cap Y = A$
  - Nếu mọi giá trị của A trong R1 đều có trong R2
    - T(W) = T(R1) [T(R2) / V(R2,A)]
  - Nếu mọi giá trị của A trong R2 đều có trong R1
    - T(W) = T(R2) [T(R1) / V(R1,A)]
  - Tổng quát
    - T(W) = T(R1).T(R2) / Max[V(R1,A) , V(R2,A)]

Ví dụ

- ***Cho***
  - R1
    - T(R1) = 1000
    - V(R1, A) = 50
    - V(R1, B) = 100
  - R2
    - T(R2) = 2000
    - V(R2, B) = 200
    - V(R2, C) = 300
  - R3
    - T(R3) = 3000
    - V(R3, C) = 90
    - V(R3, D) = 500

- ***Hãy ước lượng U = R1(A, B) ⋈ R2(B, C)***
  - T(U) = (1000 x 2000)/Max(100,200) = 10000
  - V(U, A) = 50
  - V(U, B) = 100
  - V(U, C) = 300
- ***Hãy ước lượng Z = R1(A, B) ⋈ R2(B, C) ⋈ R3(C, D)***
  - Nhận xét : Z = U(A,B,C) ⋈ R3(C, D)
  - Vậy
    - T(Z) = (10000 x 3000)/Max(300,90)=100000
    - V(Z, A) = 50
    - V(Z, B) = 100
    - V(Z, C) = 90
    - V(Z, D) = 500

Ước lượng kích thước (tt)

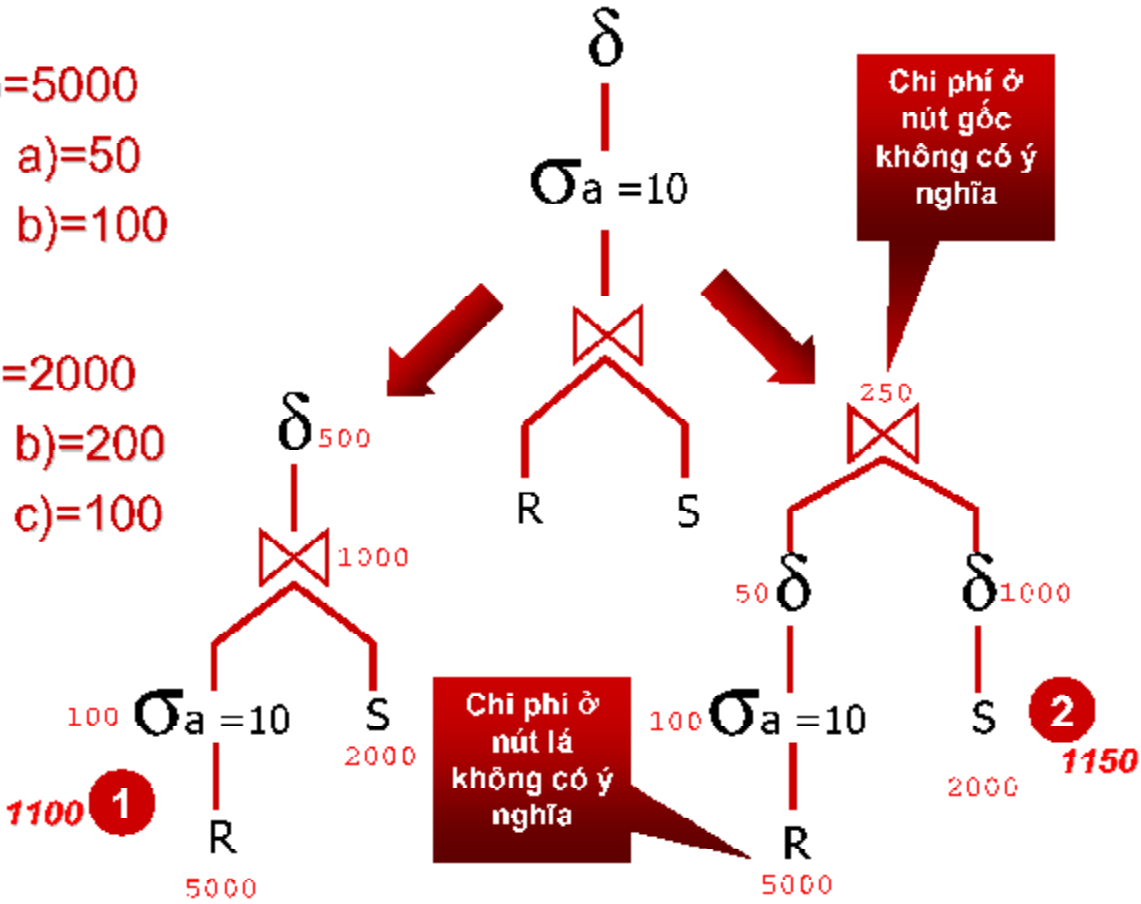
- ***Ước lượng: W = R1 ∪ R2***
  - Nếu R1 và R2 chấp nhận giá trị lặp
    - T(W) = T(R1) + T(R2)
  - Nếu R1 và R2 không chấp nhận giá trị lặp
    - TH1: R1∪ R2 không tạo giá trị lặp $T_1(W)$ =T(R1) + T(R2)
    - TH2: R1∪ R2 có tạo giá trị lặp $T_2(W)$ < T(R1) + T(R2)
    - Tổng quát : T(W) = [$T_1(W)$ + $T_2(W)$]/2
- ***Ước lượng: W = R1 ∩ R2***
  - TH1 : (trường hợp nhỏ nhất) R1 ∩ R2 = ∅ thì
    - $T_1(W)$ = 0
  - TH2 : (trường hợp lớn nhất) R1 ∩ R2 = R1 hay R2 thì
    - $T_2(W)$ = T(R1) hay T(R2)
  - Tổng quát : T(W) = [$T_1(W)$+$T_2(W)$] / 2

- ***Ước lượng: W = R1 – R2***
  - TH1 : (trường hợp lớn nhất) R1 – R2 = R1 thì
    - $T_1(W) = T(R1)$
  - TH2 : (trường hợp nhỏ nhất) R1 ∩ R2 = R2 thì
    - $T_2(W) = T(R1) – T(R2)$
  - Tổng quát : $T(W) = [T_1(W)+T_2(W)] / 2 = T(R1) – T(R2)/2$
- ***Ước lượng: W = δ(R)***
  - Giả sử $R(a_1,a_2,a_3,\ldots,a_n)$
  - Vậy số bộ phân biệt tối đa là $\Pi_{i\in[1,n]}V(R,a_i)$
  - Trường hợp nhỏ nhất : R rỗng → T(W) = 0
  - $T(W) = Min(T(R)/2 , \Pi_{i\in[1,n]}V(R,a_i))$

Ước lượng kích thước (tt)

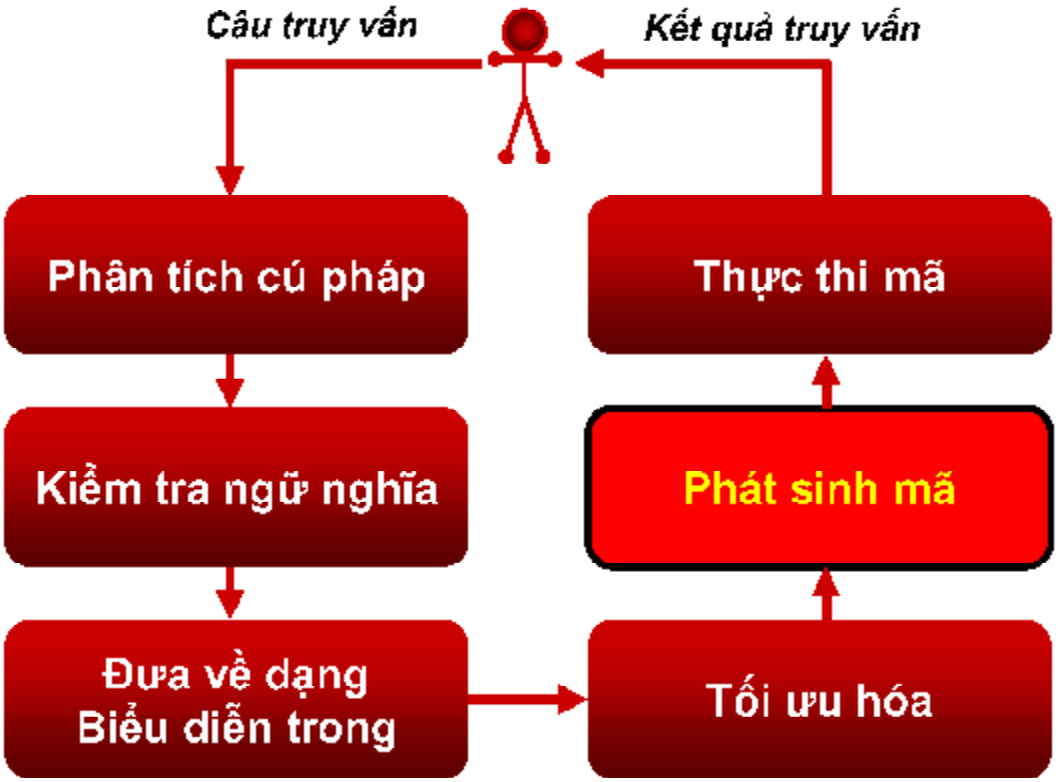
- ***Ước lượng: W = ℑ(R)***
  - TH1 : (trường hợp lớn nhất) số bộ phân biệt trong R cũng là số nhóm
    - $T_1(W) = T(\delta(R))$
  - TH2 : (trường hợp nhỏ nhất) R rỗng
    - $T_2(W) = 0$
  - TH3 : Toàn bộ R tạo 1 nhóm
    - $T_3(W) = 1$
  - Tổng quát : $T(W) = Min(T(R)/2 , \Pi_{i\in[1,n]}V(R,a_i))$
- ***Kích thước sau cùng của cây truy vấn***
  - Là tổng kích thước của phép toán ở tất cả các node, ngoại trừ node lá và node gốc.

- *R(a, b)*
  - T(R)=5000
  - V(R, a)=50
  - V(R, b)=100
- *S(b, c)*
  - T(S)=2000
  - V(S, b)=200
  - V(S, c)=100



Nội dung chi tiết

- *Giới thiệu*
- *Phân tích cú pháp - ngữ nghĩa*
- *Biến đổi sang Đại số Quan hệ*
- *Tối ưu hóa cây truy vấn*
- *Ước lượng kích thước cây truy vấn*
- *Phát sinh và thực thi mã lệnh*



Phát sinh mã (tt)

- ***Từ cây Truy vấn sau bước tối ưu hóa DBMS sẽ***
  - Phát sinh mã lệnh của ngôn ngữ chủ (ngôn ngữ dùng để viết chính DBMS) để thực thi cây truy vấn ấy
  - Các phép toán của Đại số quan hệ
    - Được cài đặt trước thành một bộ các hàm (với hệ thống tham số đầy đủ).
    - Ví dụ
      - *Projection (R: Relation,A: Array of Attribute) As Relation*
      - *Selection (R: Relation,C: Array of Condition) As Relation*
      - …
  - Việc phát sinh mã lệnh thực chất là việc phát sinh các lời gọi các hàm trên và truyền cho chúng đối số cụ thể

Sắp xếp ngoài

- Việc sắp xếp là cần thiết cho thực thi truy vấn (Vd : Order by, join, union, distinct…)
- Có trường hợp yêu cầu truy vấn liên quan thuộc tính không có chỉ mục trên ấy
- Tập tin CSDL lớn → không chứa đủ trong bộ nhớ chính để sắp xếp → Cần phải sắp xếp ngoài (dùng file tạm trên đĩa)
- Thuật toán : merge short
  - Ban đầu sắp xếp trong các run nhỏ của tập tin chính
  - Sau đó trộn các run nhỏ và lại sắp xếp để có run lớn hơn
  - Lặp lại quá trình đến khi chỉ còn 1 run
  - Số lượng run khởi điểm (nR) tùy thuộc vào số block cần sắp xếp (b) và không gian trống trong buffer (nF)

$$nR = b/nF$$

Phát sinh mã (tt)

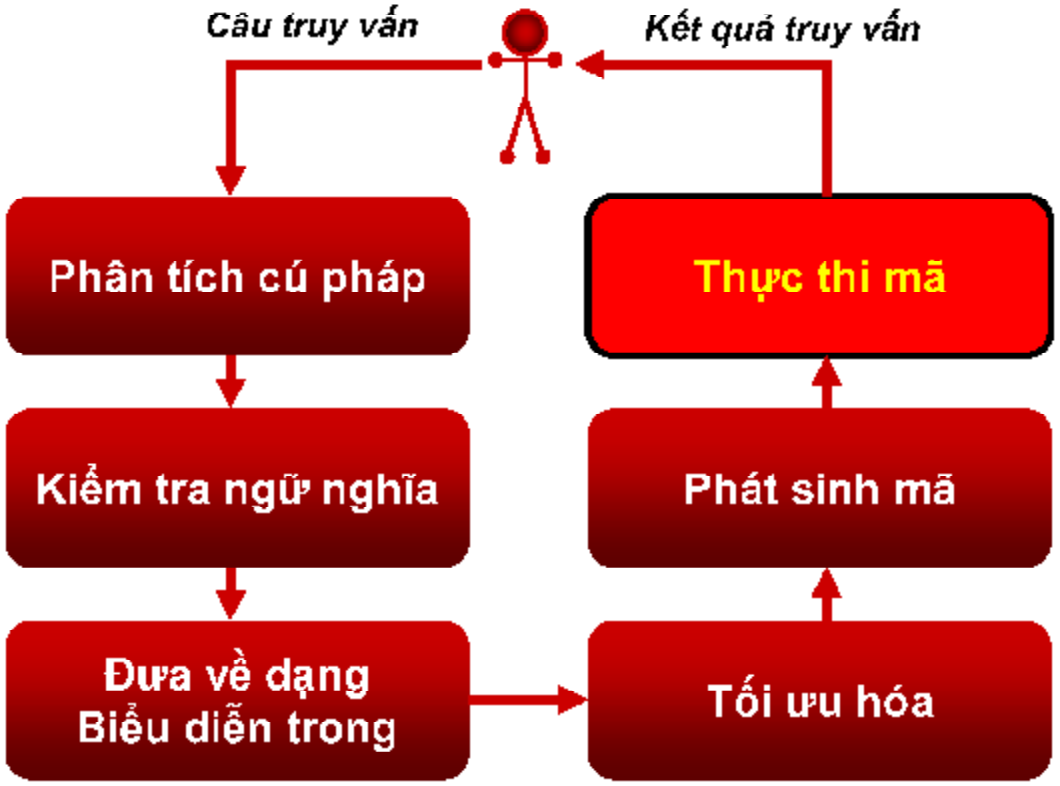
Cài đặt hàm phép chọn 1 điều kiện

- Tìm tuyến tính : Đọc từng mẫu tin và kiểm tra điều kiện chọn
- Nếu điều kiện chọn là so sánh bằng trên thuộc tính là khóa sắp xếp file → tìm nhị phân
- Nếu điều kiện chọn là so sánh bằng trên thuộc tính là khóa có primary index / hash key → dùng primary index / hash key
- Nếu điều kiện chọn là so sánh bằng trên thuộc tính không là khóa nhưng có clustering index → dùng clustering index
- Nếu điều kiện chọn không phải so sánh bằng → dùng Secondary Index
- Nếu điều kiện là so sánh ≤, ≥ thì tìm cho điều kiện = trước

- ***Cài đặt hàm phép chọn nhiều điều kiện (nối bởi AND)***
  - Chọn 1 điều kiện để thực hiện như phép chọn đơn. Khi có kết quả, loại dần những bộ không thỏa các điều kiện còn lại
  - Thực hiện từng điều kiện như từng phép chọn đơn và giao kết quả với nhau

Phát sinh mã (tt)

- ***Cài đặt hàm phép kết*** $R \bowtie_{R.A=S.B} S$
  - Dùng 2 vòng lặp lồng nhau : Duyệt mỗi bộ r trong R, duyệt mỗi bộ s trong S và kiểm tra điều kiện r.A=s.B
  - Nếu có chỉ mục trên B → dùng 1 vòng lặp : Với mỗi bộ r trong R, truy cập trực tiếp (bằng chỉ mục) các bộ s trong S thỏa s.B = r.A
  - Nếu R và S đều được sắp xếp vật lý theo A và B thì duyệt trên file tương ứng và so khớp các giá trị A và B
  - Dùng hàm băm
    - Băm trên khóa A → phân các dòng r trong R vào các lô R_i
    - Băm trên khóa B → phân các dòng s trong S vào các lô S_i
    - Quét qua R_i và S_i và tìm các lô mà $R_i.A = S_i.B$



Thực thi mã lệnh (tt)

- ***Hiệu quả của việc thực thi mã lệnh đã phát sinh ở bước trước phụ thuộc vào 2 yếu tố***
  - Mức độ tối ưu của cây truy vấn
  - Mức độ tối ưu của các hàm cài đặt các phép toán đại số quan hệ
- ***Tối ưu hóa cây truy vấn***
  - Áp dụng các quy tắc (đã học trong chương này)
- ***Mức độ tối ưu của các hàm***
  - Vận dụng các cấu trúc lưu trữ Dữ liệu (chương 5) và các thuật toán truy xuất, tìm kiếm trên các cấu trúc Dữ liệu (môn Cấu trúc Dữ liệu 1 & 2)
  - Đặc biệt quan tâm cài đặt cho phép chọn và phép kết