

Chương 3

Xử lý câu truy vấn phân tán



ptndiem@cit.ctu.edu.vn

Nội dung



- **Giới thiệu tổng quan về xử lý câu truy vấn**
- Xử lý câu truy vấn tập trung
- Xử lý truy vấn phân tán

CSDL mẫu

CUSTOMER (CID, CNAME, STREET, CCITY);

BRANCH (BNAME, ASSETS, BCITY);

ACCOUNT (A#, CID, BNAME, BAL);

LOAN (L#, CID, BNAME, AMT);

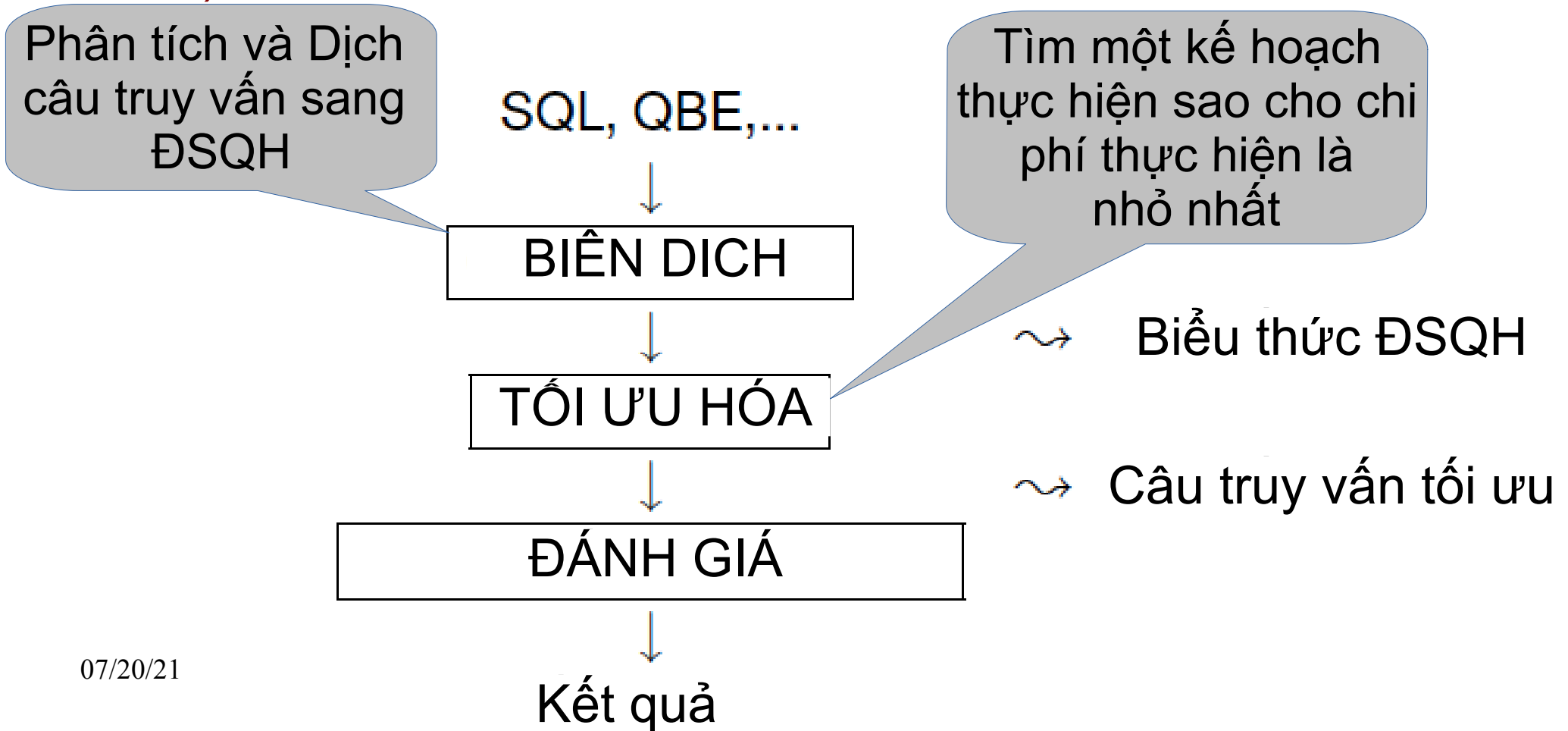
TRANSACTION (TID, CID, A#, Date, AMOUNT);

Mục tiêu xử lý câu truy vấn

- Cho một truy vấn. *Mục tiêu là đánh giá tính hiệu quả.*
 - Làm thế nào để chuyển từ SQL sang cây biểu thức đại số quan hệ?
 - Làm thế nào bộ phận tối ưu hóa có được nhiều kế hoạch thực thi (execution plan) có thể.
 - Làm thế nào lựa chọn trong số các kế hoạch này.
 - Làm thế nào để câu truy vấn sau đó được đánh giá.
- => Đây là những kỹ thuật cơ bản, được cài đặt trong bất kỳ DBMS quan hệ nào

Xử lý câu truy vấn

*Xử lý câu truy vấn (Query Processing): Là quá trình 3 bước chuyển đổi câu truy vấn cấp cao (ví dụ SQL) sang một câu truy vấn cấp thấp hơn **tương đương và hiệu quả hơn** (đại số quan hệ).*



Xử lý câu truy vấn

- Trong một hệ thống tập trung, mục tiêu của **bộ xử lý câu truy vấn** (query processor) có thể bao gồm:
 - Tối thiểu hóa thời gian trả lời truy vấn.
 - Tối đa hoá tính song song trong hệ thống.
 - Tối đa hoá thông lượng hệ thống.
 - Tối thiểu hóa tổng số tài nguyên được sử dụng (số lượng bộ nhớ, không gian đĩa, bộ nhớ cache, vv).
 - Các mục tiêu khác

=> Hệ thống có thể không thỏa mãn tất cả những mục tiêu này

Nội dung



- Giới thiệu tổng quan về xử lý câu truy vấn
- **Xử lý câu truy vấn tập trung**
- Xử lý truy vấn phân tán

3 bước xử lý câu truy vấn

1. Phân tích và dịch câu truy vấn.

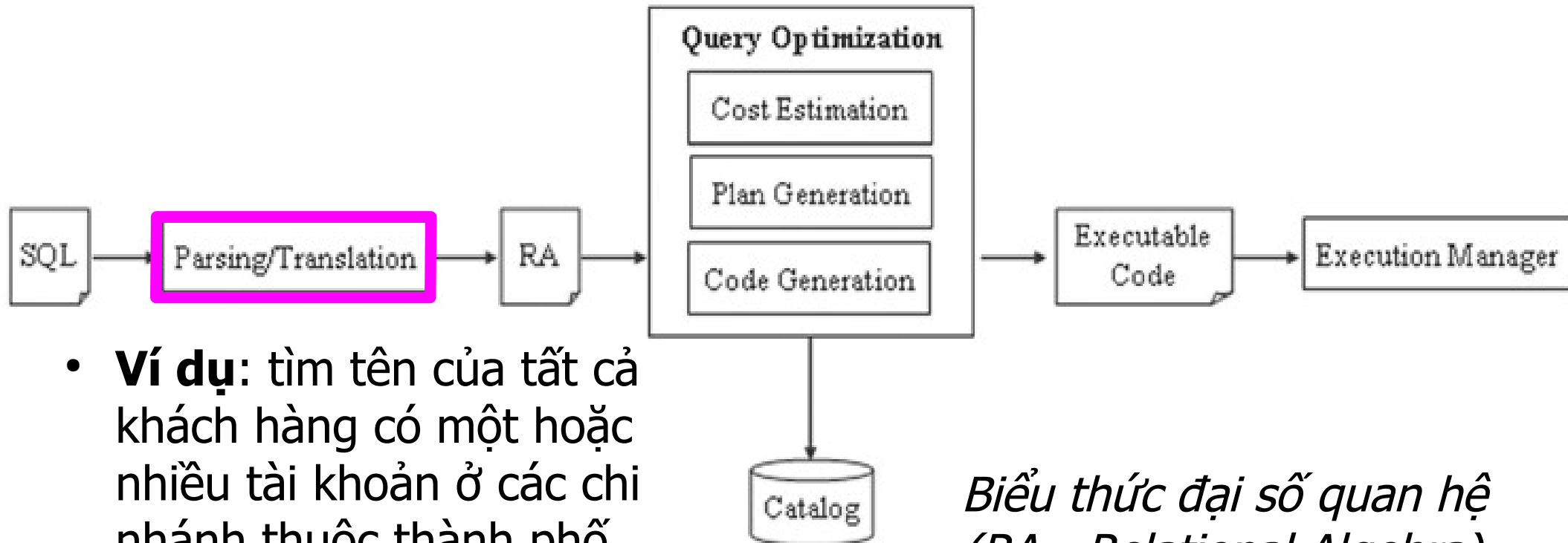
- Kiểm tra cú pháp và các quan hệ
- Nếu câu truy vấn chính xác, nó được viết lại dưới dạng biểu thức đại số quan hệ tương đương.

2. Tối ưu hóa:

- Viết lại câu truy vấn thành các biểu thức tương đương
- Lựa chọn thuật toán riêng biệt cho mỗi phép toán
- Thu được các *kế hoạch thực thi* (execution plan) và chi phí ước lượng
- Chọn *kế hoạch* tốt nhất.

3. Thực thi/Đánh giá: *kế hoạch thực thi* được biên dịch, được thực hiện và trả về kết quả.

Phân tích và dịch câu truy vấn



- **Ví dụ:** tìm tên của tất cả khách hàng có một hoặc nhiều tài khoản ở các chi nhánh thuộc thành phố Edina ?

Select c.Cname
 From Customer c, Branch b, Account a
 Where c.CID = a.CID
 AND a.Bname = b.Bname
 AND b.Bcity = 'Edina';

SQL

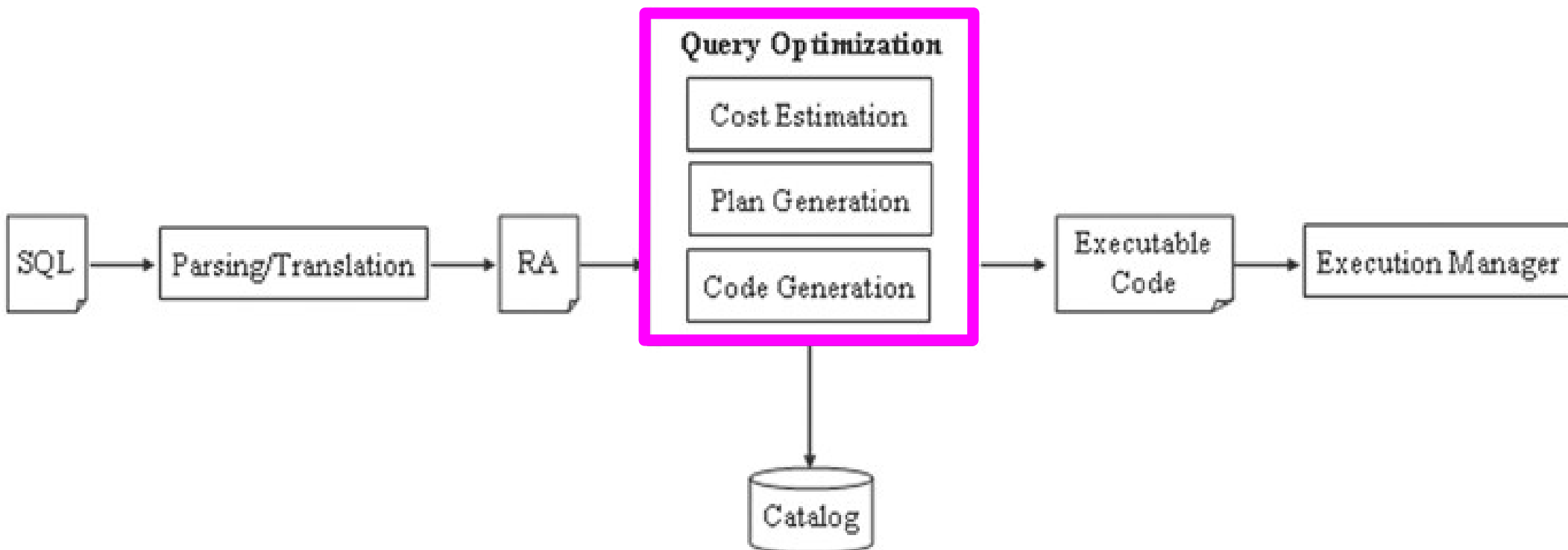
ĐSQH

Biểu thức đại số quan hệ (RA - Relational Algebra) mà bộ phân tích cú pháp có thể tạo ra :

$\pi_{cname}(\sigma_{Bcity='Edina'} (Customer * (Account * Branch)))$

Phân tích và dịch câu truy vấn

- DBMS **không thực thi ngay** biểu thức ĐSQH trên.
- Biểu thức này phải đi qua một tập các phép biến đổi và tối ưu hóa trước khi nó đã sẵn sàng để chạy
- **Bộ phận tối ưu hoá truy vấn** (query optimizer) là thành phần chịu trách nhiệm thực hiện điều này.



Nội dung



- Giới thiệu tổng quan về xử lý câu truy vấn
- **Tối ưu hoá câu truy vấn**
- Xử lý truy vấn phân tán

Tối ưu hoá câu truy vấn

- Tối ưu hoá gồm ba bước:
 - Ước lượng chi phí,
 - Sinh ra kế hoạch (plan) thực thi và
 - Sinh mã câu truy vấn.
- Trong một số DBMS (ví dụ: DB2), có một bước bổ sung được gọi là "**viết lại câu truy vấn** (query Rewrite)"
 - Được thực hiện trước khi tối ưu hóa được thực hiện.
 - Bộ phận tối ưu hóa viết lại câu truy vấn bằng cách:
 - loại bỏ điều kiện dư thừa,
 - loại bỏ các biểu thức con thừa và
 - đơn giản hóa các biểu thức phức tạp như lồng nhau

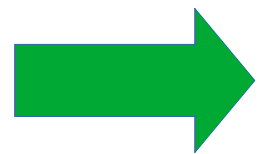
Tối ưu hoá câu truy vấn

- Những sửa đổi trong **câu truy vấn viết lại** được thực hiện bất kể **thống kê cơ sở dữ liệu**.
- Thống kê được sử dụng trong bước tối ưu để tạo ra một kế hoạch tối ưu.
- **Nhắc lại**: một kế hoạch tối ưu có thể không nhất thiết là kế hoạch tốt nhất cho truy vấn.

Ví dụ: $\pi_{\text{cname}}(\sigma_{\text{Bcity}='Edina'}(\text{Customer} \times (\text{Account} \times \text{Branch})))$
không hiệu quả vì tích Cartesian sinh ra quan hệ trung gian lớn
 \Rightarrow dùng kết nối tự nhiên

$\pi_{\text{cname}}((\text{Customer} * \text{Account}) * (\text{Account} * (\sigma_{\text{Bcity}='Edina'}(\text{Branch}))))$
 \Rightarrow loại bỏ dư thừa

$\pi_{\text{cname}}(\text{Customer} * (\text{Account} * \sigma_{\text{Bcity}='Edina'}(\text{Branch})))$



Tối ưu hoá câu truy vấn

Customer : 1000 dòng

Branch : 50 dòng, Có 1 chi nhánh ở Edina

Trung bình, mỗi khách hàng có 2 tài khoản

Trung bình mỗi chi nhánh có 40 tài khoản.

=> Account có ? Dòng => 2000

$\pi_{\text{cname}}(\sigma_{\text{Bcity}='Edina'}(\text{Customer} \times (\text{Account} \times \text{Branch})))$

R1 \leftarrow Account \times Branch => 100.000 dòng

R2 \leftarrow Customer \times R1 => 100.000.000 dòng

Mỗi dòng 100B => R2 ~ 10GB

$\pi_{\text{cname}}(\sigma_{\text{Bcity}='Edina'}((\text{Customer} * \text{Account}) * \text{Branch}))$

R1 \leftarrow Customer * Account => 2000

R2 \leftarrow R1 * Branch => 2000

Mỗi dòng 100B

=> R2 ~ 200KB

Tích Cartesian sinh ra quan hệ trung gian lớn
=> dùng kết nối tự nhiên

Tối ưu hoá câu truy vấn

- ***Tối ưu hoá cái gì ?***

- Tối ưu hoá việc sử dụng tài nguyên: bộ xử lý, truy xuất đĩa, giao tiếp (D-DB)
- Tối ưu hoá:
 - thời gian trả lời câu truy vấn;
 - số lượng câu truy vấn được xử lý trên một đơn vị thời gian (tốc độ).
- Tối ưu hóa truy vấn nhằm mục đích giảm tối thiểu hàm chi phí:

Chi phí I / O + chi phí CPU \rightarrow Min

=> Sử dụng các thông tin nào cho tối ưu ?

Tối ưu hoá câu truy vấn

- **Các thông tin sử dụng để tối ưu**

- Lược đồ luận lý của cơ sở dữ liệu, mô tả các bảng, các ràng buộc toàn vẹn
- Lược đồ vật lý của cơ sở dữ liệu, chỉ mục và các đường dẫn, kích thước của khối (block)
- **Thông kê**: kích thước của các bảng, của chỉ mục, sự phân phối các giá trị, tỷ lệ cập nhật ...
- Đặc điểm của hệ thống: tính song song, bộ vi xử lý chuyên dụng
- **Giải thuật**: chúng có thể khác nhau tùy thuộc vào hệ thống, ví dụ giải thuật kết nối, chọn, sắp xếp
 - Tất cả các phép toán ĐSQH đều có thể tốn chi phí

Tối ưu hoá câu truy vấn

- Cho biểu thức:

$\pi_{\text{name}}(\text{Customer} * (\text{Account} * \sigma_{\text{Bcity}='Edina'}(\text{Branch})))$

- Có thể có nhiều biểu thức khác tương đương với biểu thức đã cho

=> Tất cả các biểu thức khác nhau cho câu truy vấn được đánh giá bởi bộ phận tối ưu hóa truy vấn để tìm biểu thức truy vấn tối ưu ?

Xây dựng biểu thức ĐSQH tương đương

- Cho một câu truy vấn với nhiều toán tử đại số quan hệ:
 - Có nhiều lựa chọn có thể được sử dụng để thể hiện câu truy vấn.
 - Những lựa chọn được tạo ra bằng cách áp dụng **các tính chất ĐSQH**:
 - kết hợp,
 - giao hoán,
 - Idempotent (luỹ đẳng),
 - phân phối,...

Các tính chất của ĐSQH

- Toán tử một ngôi (Uop - chọn) có tính giao hoán :

$$Uop1(Uop2(R)) \equiv Uop2(Uop1(R))$$

$$\sigma_{Bname='Main'} (\sigma_{Assets>12000000} (Branch)) \equiv \sigma_{Assets>12000000} (\sigma_{Bname='Main'} (Branch))$$

- Toán tử một ngôi có tính chất lũy đẳng

$$Uop((R)) \equiv Uop1(Uop2((R)))$$

$$\sigma_{Bname='Main'} \wedge \sigma_{Assets>12000000} (Branch) \equiv \sigma_{Bname='Main'} (\sigma_{Assets>12000000} (Branch))$$

- Toán tử 2 ngôi (Bop) có tính kết hợp

$$R \text{ Bop1 } (S \text{ Bop2 } T) \equiv (R \text{ Bop1 } S) \text{ Bop2 } T$$

- Toán tử 2 ngôi có tính giao hoán **trừ phép trừ**

$$R \text{ Bop1 } S \equiv S \text{ Bop1 } R$$

Các tính chất của ĐSQH

- Toán tử một ngôi phân phối đối với một số toán tử 2 ngôi

$$Uop(R \text{ Bop } S) \equiv (Uop(R)) \text{ Bop } (Uop(S))$$

$$\sigma_{sl>5000} (\pi_{cname, sal}(\text{Customer}) \text{ UNION } \pi_{Ename, sal} (\text{CUSloyee})) \equiv$$

$$\sigma_{sl>5000} (\pi_{cname, sal}(\text{Customer})) \text{ UNION } \sigma_{sl>5000} (\pi_{Ename, sal} (\text{CUSloyee}))$$

- Toán tử một ngôi có thể được tính toán (tính chất ngược lại phân phối) đối với một số toán tử hai ngôi:

$$(Uop(R)) \text{ Bop } (Uop(S)) \equiv Uop(R \text{ Bop } S)$$

$$\sigma_{sl>5000} (\pi_{cname, sal}(\text{Customer})) \text{ UNION } \sigma_{sl>5000} (\pi_{Ename, sal} (\text{CUSloyee}))$$

$$\equiv \sigma_{sl>5000} (\pi_{cname, sal}(\text{Customer}) \text{ UNION } \pi_{Ename, sal} (\text{CUSloyee}))$$

Tối ưu hoá câu truy vấn

- Tối ưu hoá gồm ba bước:
 - Ước lượng chi phí,
 - Sinh ra kế hoạch (plan) thực thi và
 - Sinh mã câu truy vấn.

Ước lượng chi phí

- Áp dụng các tính chất trên vào một biểu thức ĐSQH
=> có thể tạo nhiều biểu thức tương đương cho câu truy vấn ĐSQH

- **Ví dụ:** $(\sigma_{Bname='Main'}(Account)) * Branch$
 $Branch * (\sigma_{Bname='Main'}(Account))$

....

=> Bộ phận tối ưu hoá truy vấn chịu trách nhiệm lựa chọn biểu thức tối ưu nhất.

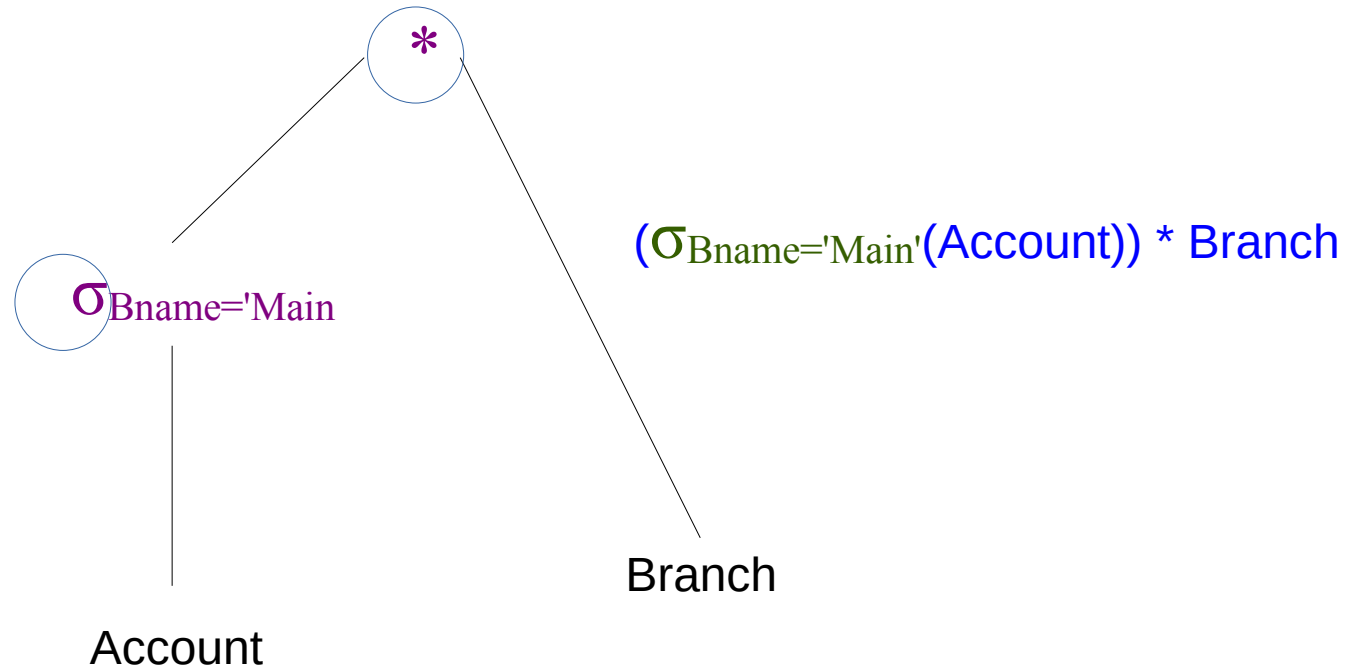
=> để thuận tiện trong việc ước lượng chi phí, khái niệm **cây biểu thức ĐSQH** (query tree) được sử dụng

Cây biểu thức ĐSQH

- Các nút lá là các quan hệ
- Các nút trong là các phép toán ĐSQH
- Toán tử một ngôi nhận vào 1 QH và trả về 1 QH
- Toán tử 2 ngôi nhận vào 2 QH và trả về 1 QH
- Các kết quả từ các toán tử của một mức được sử dụng bởi các toán tử của mức tiếp theo trong cây cho đến khi **kết quả cuối cùng được tập hợp ở nút gốc.**
- **Ví dụ:** Vẽ cây ĐSQH biểu diễn cho biểu thức:

$(\sigma_{Bname='Main'}(Account)) * Branch$

Cây biểu thức ĐSQH



Bài tập: Vẽ cây BT ĐSQH tương ứng với 2 biểu thức sau:

1. $\pi_{cname}((Customer * Account) * (Account * (\sigma_{Bcity='Edina'}(Branch))))$

2. $\pi_{cname}(Customer * (Account * \sigma_{Bcity='Edina'}(Branch)))$

Cây biểu thức ĐSQH

3. Tìm số tiền đã vay của khách hàng ở chi nhánh Cần thơ

$$\pi_{\text{AMT}}(\text{LOAN} * (\sigma_{\text{Bcity}='Cantho'}(\text{Branch})))$$

4. Tìm số tiền đã giao dịch của khách hàng Nguyễn Văn Linh ở chi nhánh Cần thơ.

$$A \leftarrow \text{Account} * (\sigma_{\text{Bcity}='Cantho'}(\text{Branch}))$$

$$B \leftarrow \sigma_{\text{Cname}='NVL'}(\text{Customer})$$

$$\pi_{\text{AMOUNT}}((A * B) * \text{Transaction})$$