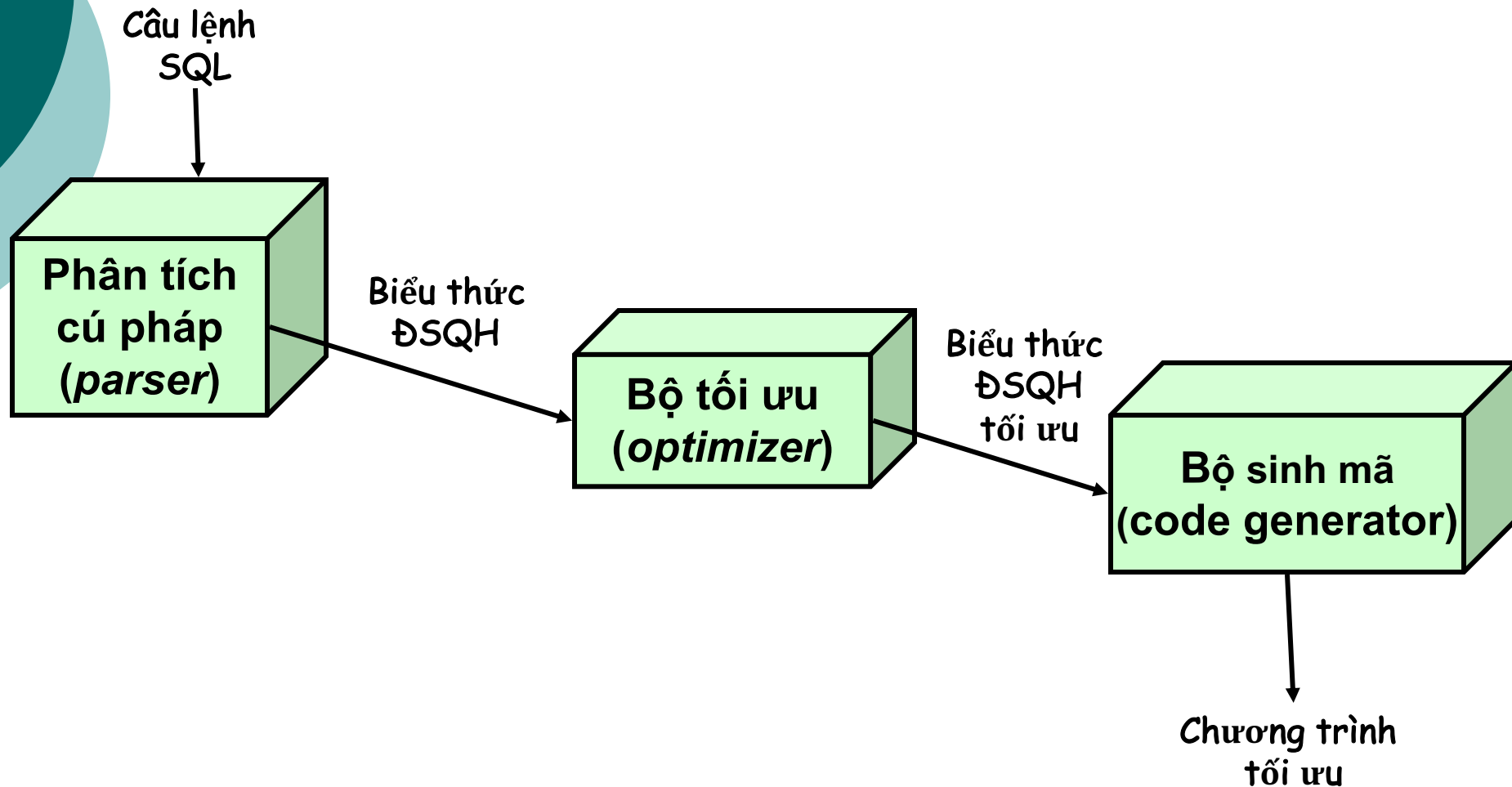




Tối ưu hoá câu hỏi

Xử lý câu hỏi truy vấn





Ngôn ngữ ĐSQH

Tổng quan

- Gồm các phép toán tương ứng với các thao tác trên các quan hệ
- Mỗi phép toán
 - Đầu vào: một hay nhiều quan hệ
 - Đầu ra: một quan hệ
- Biểu thức đại số quan hệ = chuỗi các phép toán
- Kết quả thực hiện một biểu thức đại số là một quan hệ
- Được cài đặt trong phần lớn các hệ CSDL hiện nay

Phân loại các phép toán

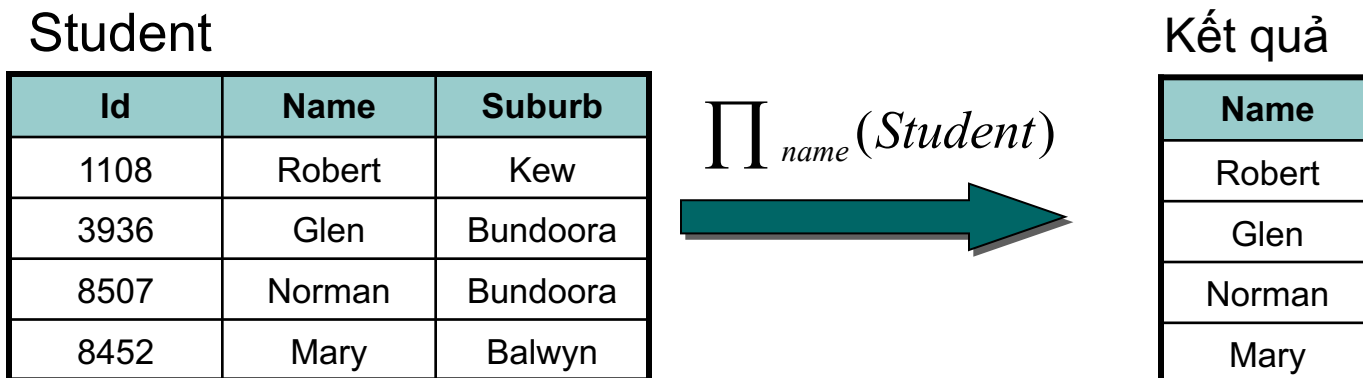
- Phép toán quan hệ
 - Phép chiếu (*projection*)
 - Phép chọn (*selection*)
 - Phép kết nối (*join*)
 - Phép chia (*division*)
- Phép toán tập hợp
 - Phép hợp (*union*)
 - Phép giao (*intersection*)
 - Phép trừ (*difference*)
 - Phép tích đề-các (*cartesian product*)

Phép chiếu

- Đ/n: Lựa chọn một số thuộc tính từ một quan hệ
- Cú pháp: $\Pi_{A_1, A_2, \dots}(R)$



- ❖ Ví dụ: đưa ra danh sách tên của tất cả các sinh viên



Phép chọn

- Đ/n: Lựa chọn các bộ trong một quan hệ thoả mãn điều kiện cho trước
- Cú pháp: $\sigma_{\langle condition \rangle}(R)$

R1
R2
R3
R4



R2
R3

- ❖ Ví dụ: đưa ra danh sách những sinh viên sống ở Bundoora

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

$$\sigma_{suburb="Bundoora"}(Student)$$



Kết quả

Id	Name	Suburb
3936	Glen	Bundoora
8507	Norman	Bundoora

Vi dụ - chọn và chiếu

- đưa ra tên của các sinh viên sống ở Bundoora

$$\Pi_{name}(\sigma_{suburb="Bundoora"}(Student))$$

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn



Kết quả

Name
Glen
Norman

Phép kết nối

- Đ/n: ghép các bộ từ 2 quan hệ thoả mãn điều kiện kết nối $R_1 \triangleright \triangleleft_{\langle join_condition \rangle} R_2$
- Cú pháp:



- ❖ Ví dụ: đưa ra danh sách các sinh viên và khoá học $Student \triangleright \triangleleft_{Id = SID} Enrol$

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

$\triangleright \triangleleft_{Id = SID}$

Enrol

SID	Course
3936	101
1108	113
8507	101



Kết quả

SID	Id	Name	Suburb	Course
1108	1108	Robert	Kew	113
3936	3936	Glen	Bundoora	101
8507	8507	Norman	Bundoora	101

Ví dụ - chọn, chiếu và kết nối

- đưa ra **tên** của các sinh viên sống ở Bundoora và **mã khoá học** mà sinh viên đó đăng ký

$$\Pi_{Name, Course} (\sigma_{Suburb="Bundoora"} (Student \triangleright \triangleleft_{Id=SID} Enrol))$$

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Enrol

SID	Course
3936	101
1108	113
8507	101

Kết quả

Name	Course
Glen	101
Norman	101

Phép kết nối tự nhiên

- Đ/n: là phép kết nối với điều kiện bằng trên các thuộc tính trùng tên

❖ Ví dụ:

Takes

SID	SNO
1108	21
1108	23
8507	23
8507	29

Enrol

SID	Course
3936	101
1108	113
8507	101

*



SID	SNO	Course
1108	21	113
1108	23	113
8507	23	101
8507	29	101

Phép kết nối ngoài

- Phép kết nối ngoài trái



- Phép kết nối ngoài phải




Ví dụ về phép kết nối ngoài

- Đưa ra danh sách mã số các sinh viên và mã khoá học mà sinh viên đó đăng ký nếu có

Student

ID	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn


Id = SID

Enrol

SID	Course
3936	101
1108	113
8507	101

Kết quả

ID	Name	Suburb	Course
1108	Robert	Kew	113
3936	Glen	Bundoora	101
8507	Norman	Bundoora	101
8452	Mary	Balwyn	null

Phép tích đề-các

- Đ/n: là kết nối giữa từng bộ của quan hệ thứ nhất và mỗi bộ của quan hệ thứ hai
- Cú pháp: $R_1 \times R_2$

a
b
c
d

\times

x
y



a	x
a	y
b	x
b	y
c	x
c	y
d	x
d	y

Ví dụ phép tích đề-các

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Sport

SportID	Sport
05	Swimming
09	Dancing

X

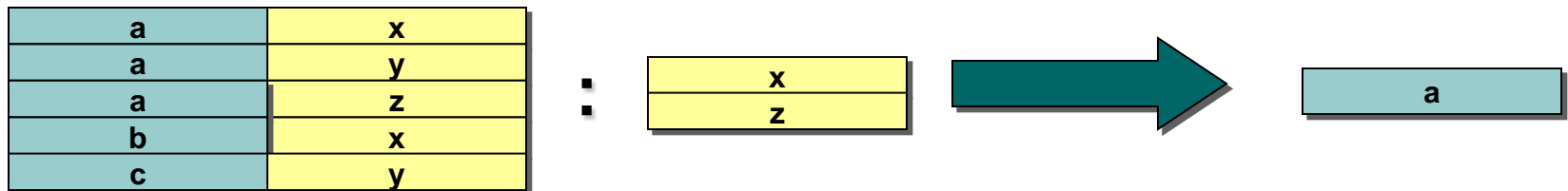
Student_Sport



Id	Name	Suburb	SportID	Sport
1108	Robert	Kew	05	Swimming
3936	Glen	Bundoora	05	Swimming
8507	Norman	Bundoora	05	Swimming
8452	Mary	Balwyn	05	Swimming
1108	Robert	Kew	09	Dancing
3936	Glen	Bundoora	09	Dancing
8507	Norman	Bundoora	09	Dancing
8452	Mary	Balwyn	09	Dancing

Phép chia

- Đ/n: cho R_1 và R_2 lần lượt là các quan hệ n và m ngôi. Kết quả của phép chia R_1 cho R_2 là một quan hệ $(n-m)$ ngôi
- Cú pháp: $R_1 : R_2$



❖ Ví dụ:

Subject

Name	Course
Systems	BCS
Database	BCS
Database	MCS
Algebra	MCS

Course

Course
BCS
MCS

:



Kết quả

Name
Database

Phép hợp

- Đ/n: gồm các bộ thuộc ít nhất một trong hai quan hệ đầu vào
 - 2 quan hệ khả hợp được xác định trên cùng miền giá trị
- Cú pháp: $R_1 \cup R_2$



❖ Ví dụ:

Subject

Name	Course
Systems	BCS
Database	BCS
Database	MCS
Algebra	MCS

∪

Subject2

Name	Course
DataMining	MCS
Writing	BCS

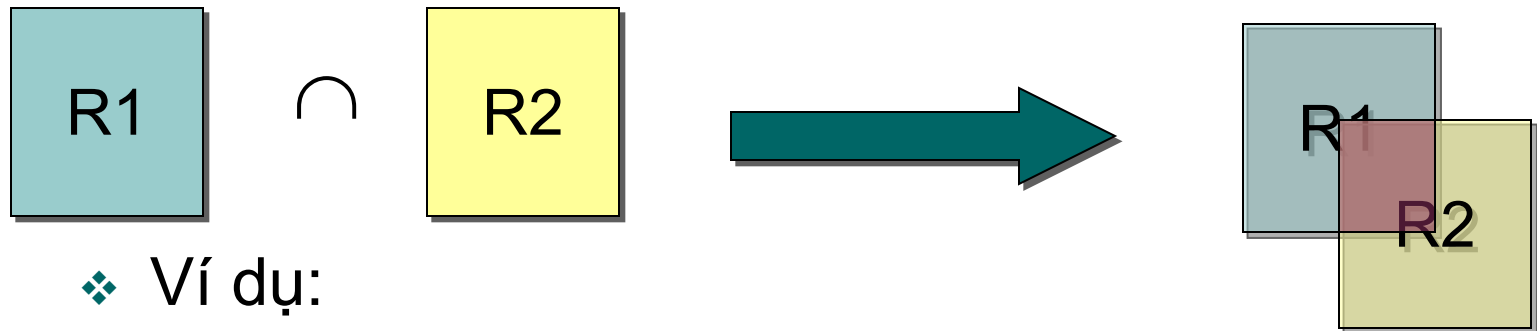
Kết quả



Name	Course
Systems	BCS
Database	BCS
Database	MCS
Algebra	MCS
DataMining	MCS
Writing	BCS

Phép giao

- Đ/n: gồm các bộ thuộc cả hai quan hệ đầu vào
- Cú pháp: $R_1 \cap R_2$



❖ Ví dụ:

Subject

Name	Course
Systems	BCS
Database	BCS
Database	MCS
Algebra	MCS

Subject2

Name	Course
DataMining	MCS
Database	MCS
Systems	BCS
Writing	BCS

Kết quả

Name	Course
Systems	BCS
Database	MCS

Phép trừ

- Đ/n: gồm các bộ thuộc quan hệ thứ nhất nhưng không thuộc quan hệ thứ hai
 - 2 quan hệ phải là khả hợp
- Cú pháp: $R_1 \setminus R_2$



❖ Ví dụ:

Subject

Name	Course
Systems	BCS
Database	BCS
Database	MCS
Algebra	MCS

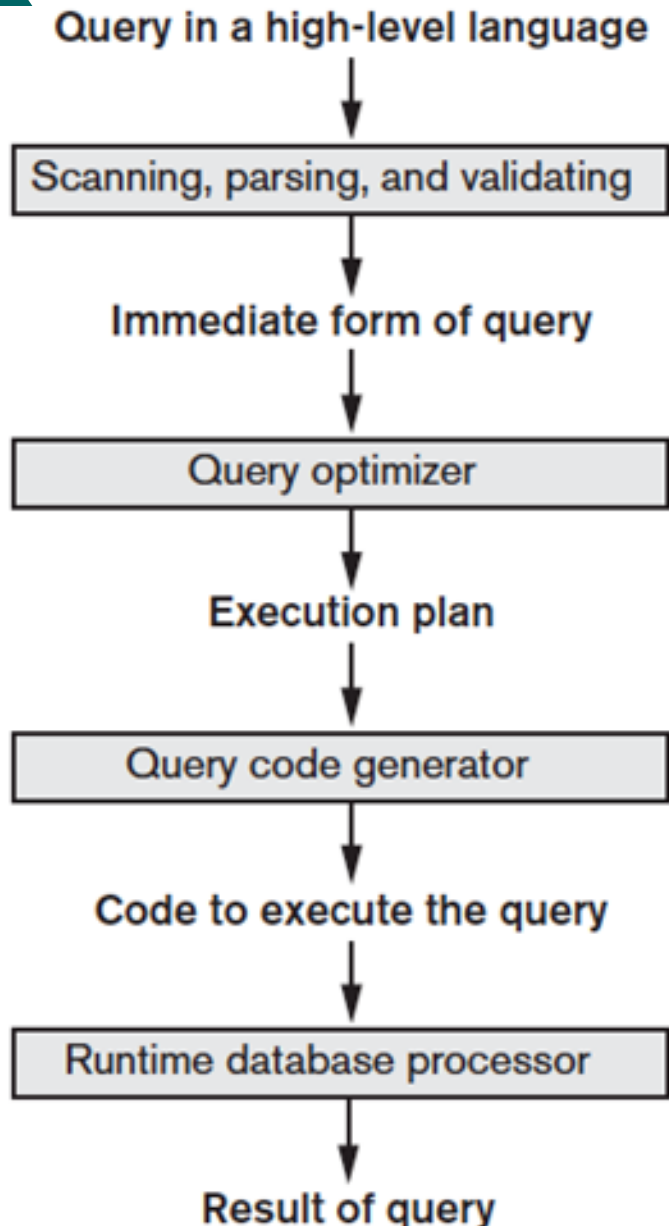
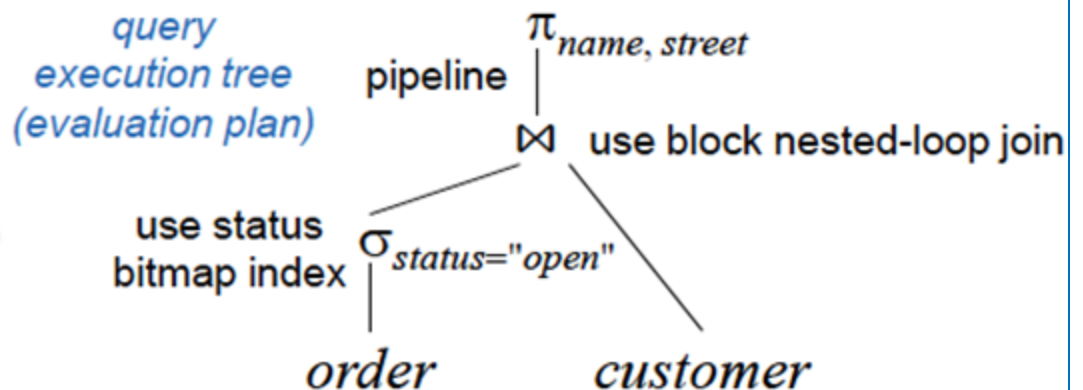
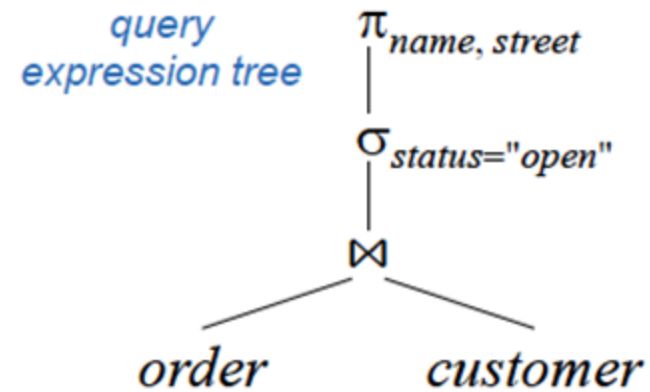
Subject2

Name	Course
DataMining	MCS
Database	MCS
Systems	BCS
Writing	BCS

Kết quả

Name	Course
Database	BCS
Algebra	MCS

```
SELECT name, street
FROM Customer, Order
WHERE Order.customerID = Customer.customerID
AND status = 'open';
```

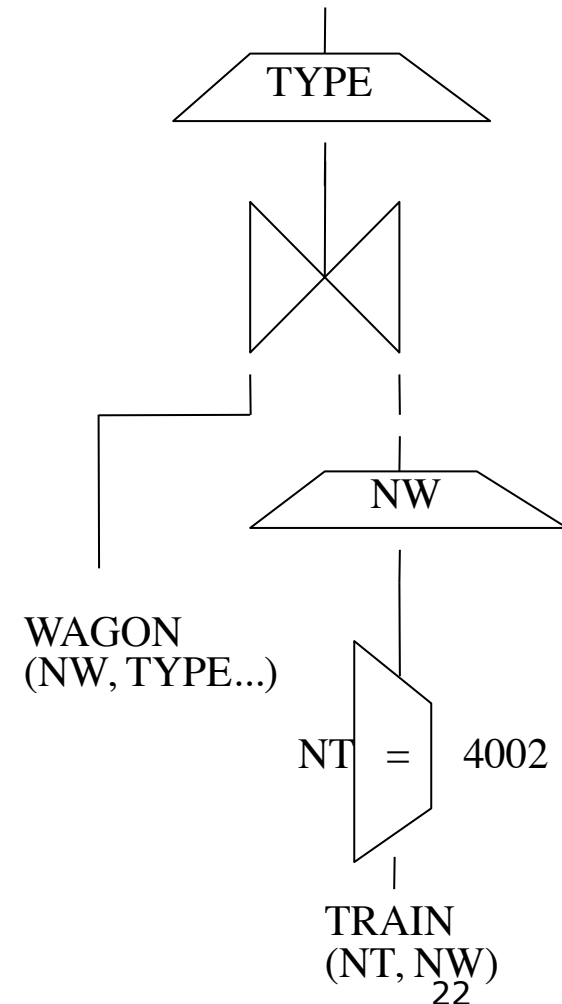
$$\pi_{name, street}(\sigma_{status="open"}(order \bowtie customer))$$


Tối ưu hoá

- Biến đổi biểu thức ĐSQH để tìm 1 biểu thức hiệu quả
- Tối ưu dựa trên cấu trúc và nội dung của dữ liệu
- Nâng cao hiệu quả thực hiện câu hỏi trên 1 hay nhiều tiêu chí: thời gian, sử dụng bộ nhớ, ...
- Lưu ý:
 - Không nhất thiết phải tìm biểu thức tối ưu nhất
 - Chú ý tới tài nguyên sử dụng cho tối ưu

Kỹ thuật tối ưu hoá

- 2 kỹ thuật chính
 - Tối ưu logic (rewriting)
 - Tối ưu vật lý (access methods)
- Mục đích của các kỹ thuật tối ưu
 - Giảm số bản ghi
 - Giảm kích thước bản ghi
- Ví dụ
WAGON (NW, TYPE, COND, STATION,
CAPACITY, WEIGHT)
TRAIN (NT, NW)





Nội dung

- Giới thiệu chung
- Tối ưu logic
- Tối ưu vật lý
- Mô hình giá

Tối ưu hoá logic

- Sử dụng các phép biến đổi tương đương để tìm ra biểu thức ĐSQH tốt
- Gồm 2 giai đoạn
 - Biến đổi dựa trên ngữ nghĩa
 - Biến đổi dựa trên tính chất của các phép toán ĐSQH

Biến đổi dựa trên ngữ nghĩa

- Mục đích:

- Dựa trên các ràng buộc dữ liệu để xác định các biểu thức tương đương
- Viết lại câu hỏi trên khung nhìn dựa trên các định nghĩa của khung nhìn

- Ví dụ:

EMPLOYEE (FirstName, LastName, SSN, Birthday, Adresse, NoDept)

DEPARTEMENT (DNO, DName, SSNManager)

PROJECT (PNO, PName, PLocation, DNo)

WORK-IN (ESSN, PNO, Heures)

Biến đổi dựa trên ngữ nghĩa ..

- Định nghĩa khung nhìn: $V = R * S$
 - Câu truy vấn của client : $Q = V * (T * U)$
- ➔ Viết lại câu truy vấn dựa trên định nghĩa khung nhìn:
- $$Q = (R * S) * (T * U)$$



Biến đổi dựa trên t/chất của ĐSQH

Tính chất của phép toán ĐSQH

$A \sim$ tập các thuộc tính, $F \sim$ biểu thức điều kiện

1. Phép chiếu và phép chọn

$$\Pi_A(R) \Rightarrow \Pi_A(\Pi_{A1}(R)) \quad \text{if} \quad A \subseteq A1$$

$$\sigma_F(R) \Rightarrow \sigma_{F1}(\sigma_{F2}(R)) \quad \text{if} \quad F = F1 \wedge F2$$

Tính chất của phép toán ĐSQH (2)

A ~ tập các thuộc tính, F ~ biểu thức điều kiện

2. Tính giao hoán đối với phép chọn và chiếu

$$\sigma_{F_1}(\sigma_{F_2}(R)) \Rightarrow \sigma_{F_2}(\sigma_{F_1}(R))$$

$$\sigma_{F_1}(\Pi_{A_2}(R)) \Rightarrow \Pi_{A_2}(\sigma_{F_1}(R))$$

Nếu các thuộc tính của F2 thuộc A1 :

$$\Pi_{A_1}(\sigma_{F_2}(R)) \Rightarrow \sigma_{F_2}(\Pi_{A_1}(R))$$

Nếu $A1 \subseteq A2$:

$$\Pi_{A_1}(\Pi_{A_2}(R)) \Rightarrow \Pi_{A_1}(R)$$

Tính chất của phép toán ĐSQH (3)

3. Tính giao hoán và kết hợp của các phép toán

$$R \times S \Rightarrow S \times R$$

$$R * S \Rightarrow S * R$$

$$R \cap S \Rightarrow S \cap R$$

$$R \cup S \Rightarrow S \cup R$$

$$(R \times S) \times T \Rightarrow R \times (S \times T)$$

$$(R \cap S) \cap T \Rightarrow R \cap (S \cap T)$$

$$(R \cup S) \cup T \Rightarrow R \cup (S \cup T)$$

$$(R_{F1} * S_{F2}) * T \Rightarrow R_{F1} * (S_{F2} * T) \quad \text{chỉ nếu } \text{Attr}(F2) \subseteq \text{Attr}(S) \cup \text{Attr}(T)$$

Tính chất của phép toán ĐSQH (4)

4. Tính phân phối σ và Π trên các phép toán $*$, \cap , \cup , $-$, \times

Nếu $F = (FR \wedge FS)$ và nếu $Attr(FR) \subseteq R$ và $Attr(FS) \subseteq S$ thì :

$$\sigma_F(R \underset{JC}{*} S) \Rightarrow \sigma_{FR}(R) \underset{JC}{*} \sigma_{FS}(S)$$

$$\sigma_F(R \times S) \Rightarrow \sigma_{FR}(R) \times \sigma_{FS}(S)$$

$$\sigma_F(R \cup S)$$

$$\sigma_F(R) \cup \sigma_F(S)$$

$$\sigma_F(R - S)$$

$$\sigma_F(R) \setminus \sigma_F(S)$$

$$\pi_Z(R(X) \times S(Y))$$

$$\pi_{X \cap Z}(R) \times \pi_{Y \cap Z}(S)$$

$$\pi_Z(R \cup S)$$

$$\pi_Z(R) \cup \pi_Z(S)$$

Biến đổi biểu thức ĐSQH

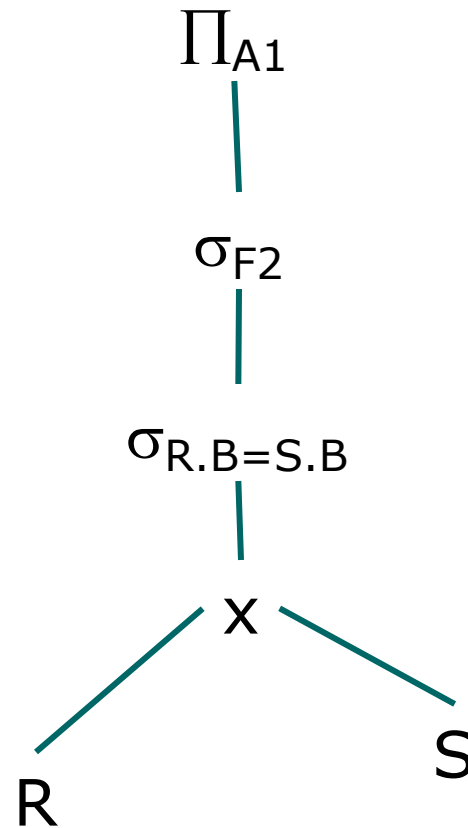
T1	$\sigma_{F1 \wedge F2 \wedge \dots F_n}(R)$	$\sigma_{F1}(\sigma_{F2} \dots (\sigma_{F_n}(R)))$
T2	$j_Z(j_Y(R))$	$j_Z(R) \text{ n''Aut } Z \subseteq Y$
T3	$\sigma_{F(X)}(j_Y(R))$	$j_Y(\sigma_{F(X)}(R)) \text{ n''Aut } X \subseteq Y$
T3'	$j_Y(\sigma_{F(X)}(R))$	$j_Y(\sigma_{F(X)}(j_{X \cup Y}(R))) \text{ n''Aut } X \not\subseteq Y$
T4	$\sigma_{F(Z)}(R(X) \times S(Y))$ $\sigma_{F(Z1) \wedge F(Z2)}(R(X) \times S(Y))$	$\sigma_{F(Z)}(R(X)) \times S(Y) \text{ n''Aut } Z \subseteq X$ $\sigma_{F(Z1)}(R(X)) \times \sigma_{F(Z2)}(S(Y))$ $\text{n''Aut } Z1 \subseteq X \text{ và } Z2 \subseteq Y$
T5	$\sigma_F(R \cup S)$	$\sigma_F(R) \cup \sigma_F(S)$
T6	$\sigma_F(R - S)$	$\sigma_F(R) \ominus \sigma_F(S)$
T7	$j_Z(R(X) \times S(Y))$	$j_{X \cap Z}(R) \times j_{Y \cap Z}(S)$
T8	$j_Z(R \cup S)$	$j_Z(R) \cup j_Z(S)$

Trình tự áp dụng

- Khai triển phép lựa chọn dựa trên nhiều điều kiện: T1
- Hoán vị phép chọn với tích đề-các, hợp, trừ: T3, T4, T5, T6 : *đẩy phép chọn để có thể thực hiện sớm nhất có thể*
- Hoán vị phép chiếu với tích đề-các, hợp : T2, T3', T7, T8
- Nhóm các điều kiện chọn bởi T1 và áp dụng T2 để loại các phép chiếu dư thừa

Biểu diễn dạng cây của ĐSQH

$$\begin{aligned} & \Pi_{A1}(\sigma_{F2}(R * S)) \\ &= \Pi_{A1}(\sigma_{F2} \sigma_{R.B=S.B}(R \times S)) \end{aligned}$$



Bài tập

EMPLOYEE (FirstName, LastName, SSN, Birthday, Adresse, NoDept)
DEPARTEMENT (DNO, DName, SSNManager)
PROJECT (PNO, PName, PLocation, DNo)
WORK (ESSN, PNO, Heures)

Tên của các nhân viên sinh sau ngày 30/01/70 và làm việc cho dự án "Esprit"

$$\Pi_{FirstName, LastName} \left(\begin{array}{l} \sigma_{Birthday > '30/01/70' \wedge PName = 'Esprit'} \\ \left(\sigma_{Employee.SSN = Work.ESSN} (EMPLOYEE \times WORK) \right) \\ * PROJECT \end{array} \right)$$

$$\Pi_{FirstName, LastName} \left(\begin{array}{l} \sigma_{Employee.SSN = Work.ESSN} (\sigma_{Birthday > '30/01/70'} (EMPLOYEE) \times WORK) \\ * \sigma_{PName = 'Esprit'} (PROJECT) \end{array} \right)$$



Nội dung

- Giới thiệu chung
- Tối ưu logic
- Tối ưu vật lý
- Mô hình giá

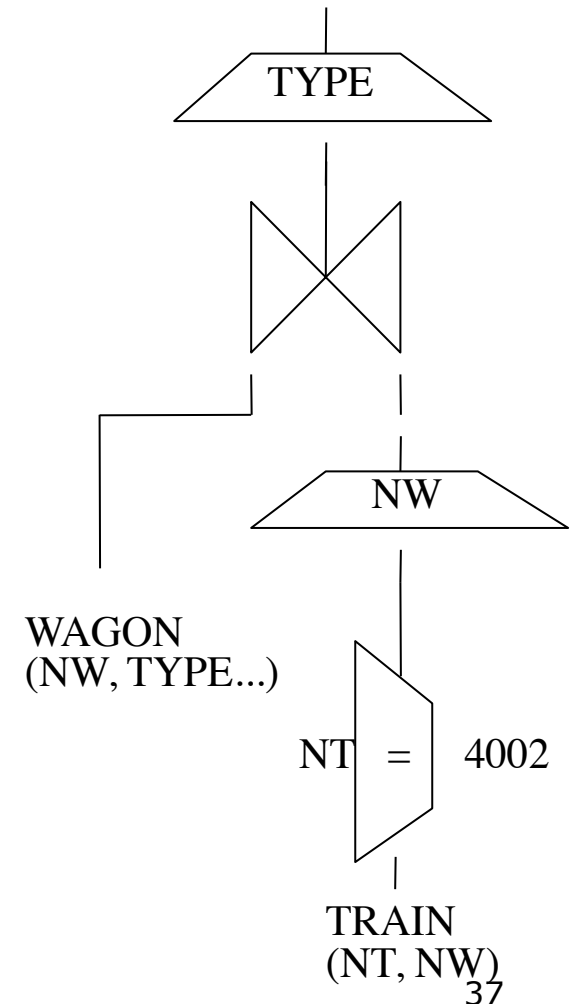
Lựa chọn cách truy nhập dữ liệu

○ Giả thiết

- TRAIN : có chỉ số trên NT
- WAGON : có chỉ số trên NW

○ Thực hiện phép kết nối

- Lựa chọn 1 giải thuật.
- Lựa chọn cách truy nhập các quan hệ



Lựa chọn với điều kiện phức AND

- Sử dụng đk với thuộc tính có index trước tiên
- Sử dụng composite index
- Nếu có nhiều index, sử dụng phép giao
- Query optimizer lựa chọn điều kiện đơn
 - Dựa vào điểm selectivity của từng điều kiện
 - Mục tiêu: hạn chế số lượng bản ghi phải truy xuất

Lựa chọn với điều kiện phức OR

- Không có nhiều khả năng tối ưu
- Chỉ sử dụng nếu có index cho tất cả các thuộc tính nằm trong điều kiện chọn
- Trái lại, phải thực thi linear scan

$\sigma_{Dno=5 \text{ OR } Salary > 30000 \text{ OR } Sex='F'}(EMPLOYEE)$



Thực thi phép toán JOIN

- Một trong những phép toán tốn chi phí thời gian nhất (time-consuming)
- Phương án thực thi:
 - Nested-loop JOIN
 - Single-loop JOIN
 - Sort-merge JOIN
 - Partition-hash JOIN

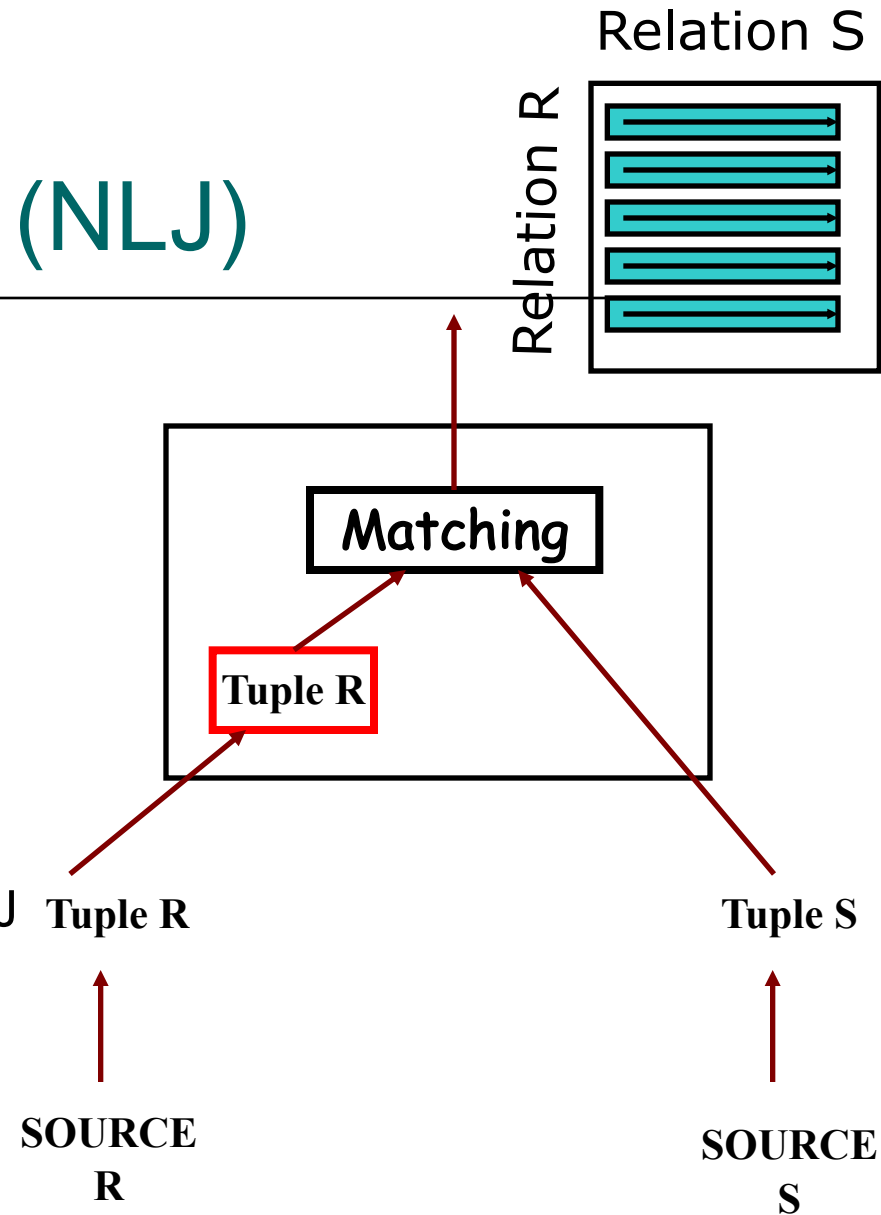
Nested-loop JOIN

```
for each tuple tr in r {  
  for each tuple ts in s {  
    if (tr and ts satisfy the join condition  $\theta$ ) {  
      add tuple  $tr \times ts$  to the result set  
    }  
  }  
}
```

- Không cần sử dụng index, có thể sử dụng cho bất kì điều kiện JOIN nào
- Chi phí cao $O(n^2)$ do mỗi cặp bản ghi (bộ) trong 2 bảng đều được xem xét

Nested-loop-join (NLJ)

- Nguyên tắc
 - Duyệt 1 lần trên quan hệ ngoài R & lặp trên quan hệ trong S
- Các mở rộng của thuật toán
 - Tuple-based NLJ, block-based NLJ, index-based NLJ

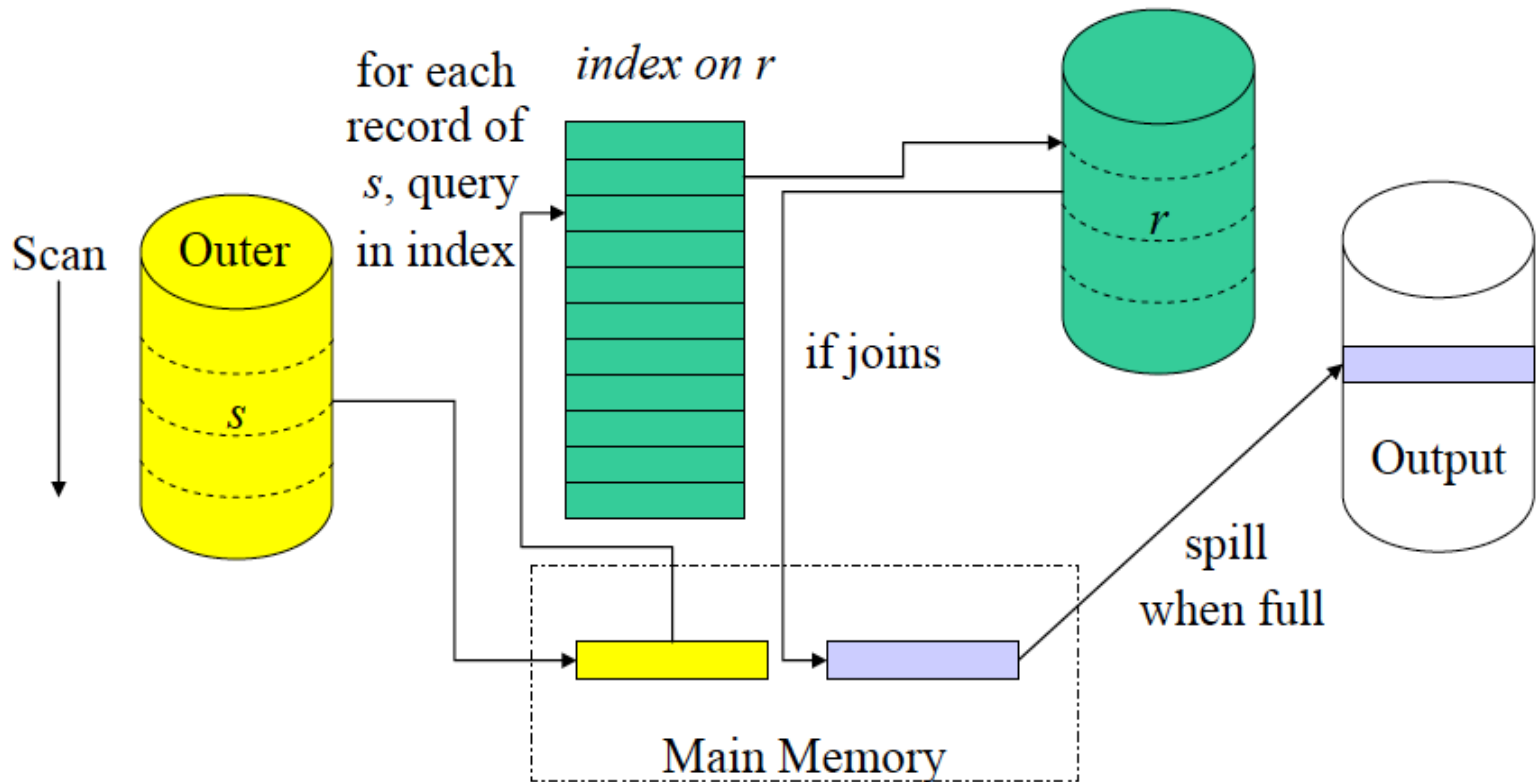


Single-loop (index-based) JOIN

- Sử dụng khi tồn tại index cho một hoặc cả hai thuộc tính trong điều kiện JOIN

```
for each tuple ts in S {  
  tr = index.get(tuple ts) {  
    if tr.exist() {  
      add ts x tr to the result set  
    }  
  }  
}
```

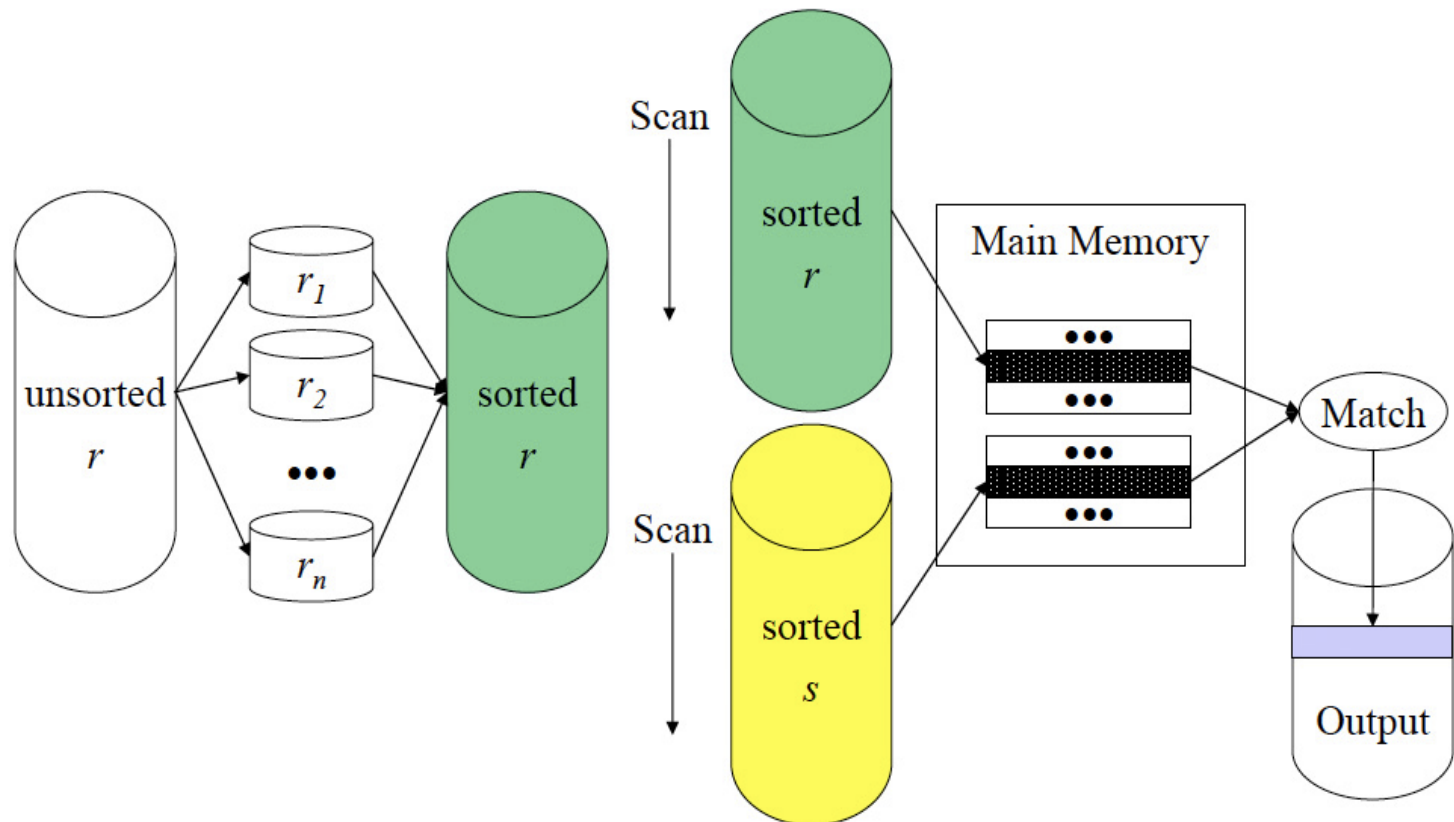
Single-loop (index-based) JOIN



Sort-merge JOIN

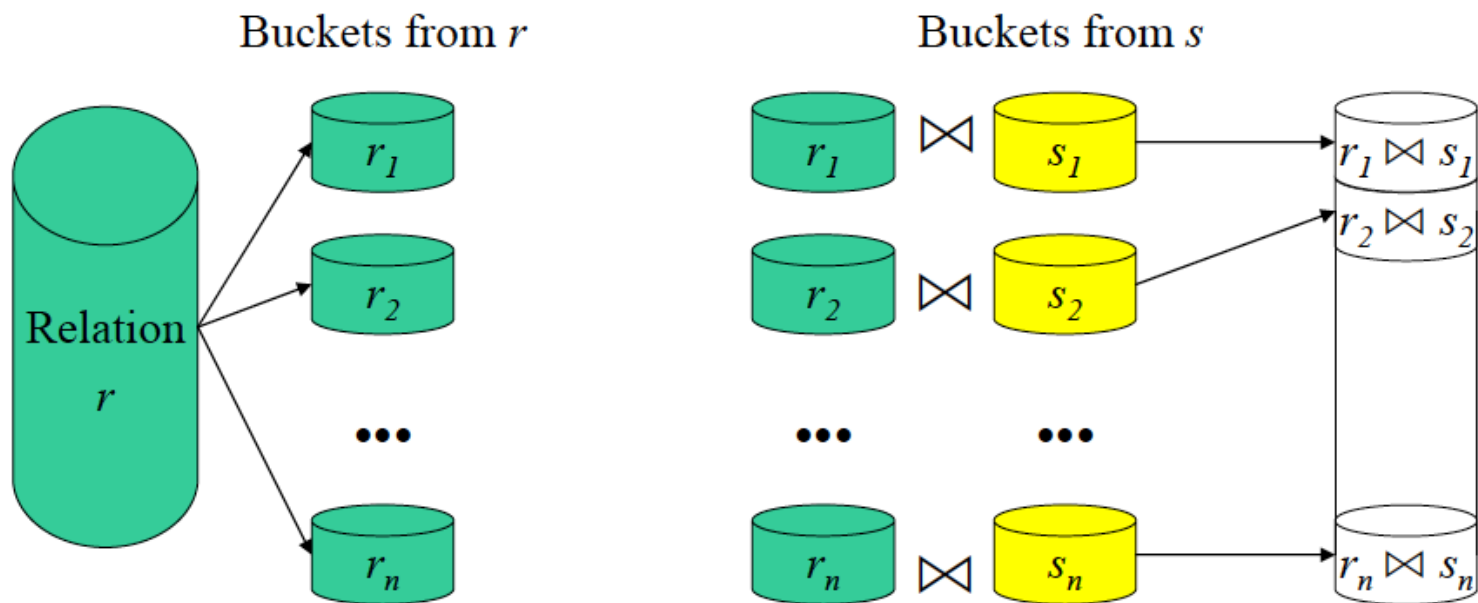
- Sử dụng khi R và S được lưu trữ dạng sorted file, sắp xếp theo thuộc tính thuộc điều kiện kết nối
 - 2 file được duyệt đồng thời
 - Chọn ra 2 bộ thỏa mãn điều kiện JOIN
- Nếu R và S chưa được lưu dạng sorted file, có thể tiến hành sắp xếp trước theo thuật toán external sorting

Sort-merge JOIN



Partition-hash JOIN

- Băm các quan hệ theo thuộc tính JOIN
- Join các giỏ (buckets) tương ứng



Hash r (same for s)

Join corresponding r and s buckets

Mô hình giá

- Chi phí thực hiện câu hỏi truy vấn phụ thuộc:
 - Đọc/ghi bộ nhớ ngoài (số trang nhớ)
 - Kích thước dữ liệu (trung gian) phải xử lý
- Chi phí :
 - Đọc/ghi dữ liệu + Xử lý + Truyền thông giữa các trạm

Tối ưu hoá dựa trên mô hình giá

- Mục đích: Chọn phương án thực hiện câu hỏi với **chi phí thấp nhất**
- Nhận xét:
 - Chi phí cho **liệt kê các phương án** trả lời câu hỏi
 - Chi phí cho **lượng hoá** các phương án theo mô hình giá
 - Có thể sử dụng các « mẹo » (heuristics) để giảm không gian tìm kiếm của câu hỏi
- Mỗi **quan hệ/ index** đều có các thông số được **thống kê sẵn**

Đánh giá các biểu thức ĐSQH

○ Vật chất hóa:

- Ghi các kết quả trung gian
- Chi phí đánh giá câu hỏi: + thời gian đọc/ghi DL trung gian

○ Đường ống (pipelining):

- Tổ chức các phép toán trong 1 đường ống
- Kết quả ra của phép toán này được lấy ngay làm đầu vào cho phép toán kế tiếp
- Không mất thời gian đọc/ghi DL trung gian
- Không phải trường hợp nào cũng có thể thực hiện được

Kết luận

- Tối ưu hoá nhằm tìm phương án tốt nhất để thực hiện một câu hỏi
 - Cần lưu ý: chi phí thực hiện tối ưu hoá và chi phí thực hiện câu hỏi
- Các kỹ thuật tối ưu
 - **Logic** : kiểm tra điều kiện ràng buộc của các thuộc tính/quan hệ và điều kiện lựa chọn trong câu hỏi, biến đổi tương đương các biểu thức ĐSQH
 - **Vật lý** : tổ chức vật lý của dữ liệu trên đĩa, mô hình giá
 - Không nhất thiết phải áp dụng tất cả các kỹ thuật trên khi thực hiện tối ưu hoá 1 câu hỏi