



CƠ SỞ DỮ LIỆU

Đại số quan hệ & Phép toán quan hệ

[1] Relational Algebra and Relational Calculus



Nội dung chương 5

- ❖ Cách tạo query trong đại số quan hệ
- ❖ Cách tạo query trong phép toán quan hệ trên bộ
- ❖ Cách tạo query trong phép toán quan hệ trên miền
- ❖ Các loại ngôn ngữ thao tác dữ liệu quan hệ



Ví dụ về phép chọn của đại số quan hệ

- ❖ Liệt kê danh sách tất cả nhân viên ở chi nhánh B003

$\sigma_{\text{branchNo} = \text{'B003'}} (\text{Staff})$



Ngôn ngữ quan hệ

- ❖ **Ngôn ngữ phi thủ tục** (non-procedural language) cho phép người dùng chỉ ra cái (what) họ muốn mà không cần chỉ ra cách thực hiện như thế nào (how) để được kết quả đó
 - SQL là ngôn ngữ phi thủ tục
- ❖ **Đại số quan hệ** là một ngôn ngữ thủ tục mức cao (high-level procedure language) cho phép DBMS tạo 1 quan hệ mới



- ❖ Đại số quan hệ và phép toán quan hệ là các ngôn ngữ chuẩn của mô hình quan hệ
- ❖ Có thể xem: **đại số quan hệ** là ngôn ngữ thủ tục (cấp cao), và **phép toán quan hệ** là ngôn ngữ phi thủ tục
- ❖ Tuy nhiên về mặt lý thuyết thì đại số quan hệ và phép toán quan hệ là tương đương nhau.



Đại số quan hệ

- ❖ Các toán tử của đại số quan hệ áp dụng trên một hay nhiều quan hệ để định nghĩa ra một quan hệ khác mà không làm thay đổi các quan hệ ban đầu
- ❖ Cả các toán hạng và các kết quả đều là các quan hệ, vì vậy đầu ra của một toán tử có thể là đầu vào của một toán tử khác
- ❖ Cho phép các biểu thức có thể lồng vào nhau giống như các biểu thức số học. Đặc tính này gọi là closure.



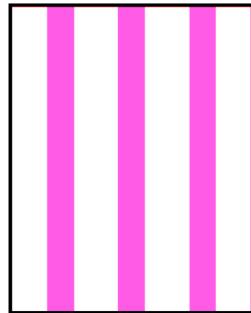
- ❖ 5 toán tử cơ bản trong ĐS quan hệ: Chọn (Selection), Chiếu (Projection), Tích Đè các (Cartesian product), Hội (Union), và Trừ (Set Difference)
- ❖ Các toán tử cơ bản thực hiện được hầu hết các thao tác truy xuất dữ liệu
- ❖ Các toán tử khác: Kết (Join), Giao (Intersection), và Chia (Division). Các toán tử này có thể được biểu diễn bằng 5 toán tử cơ bản trên



Các phép toán trong đại số quan hệ



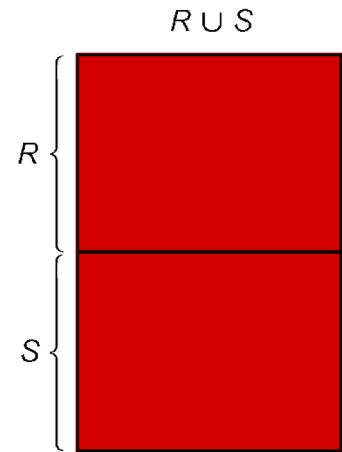
(a) Selection



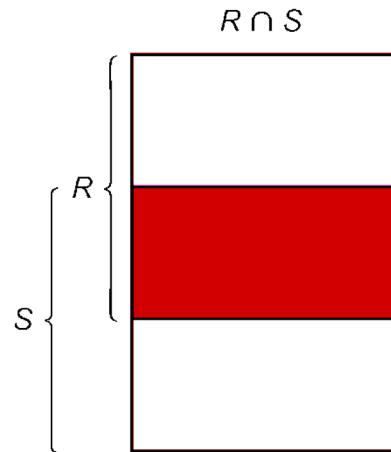
(b) Projection

$$R \quad S \quad R \times S$$
$$\begin{array}{|c|} \hline a \\ \hline b \\ \hline \end{array} \quad \begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline 3 \\ \hline \end{array} \quad = \quad \begin{array}{|c|c|} \hline & 1 \\ \hline a & 1 \\ \hline a & 2 \\ \hline a & 3 \\ \hline b & 1 \\ \hline b & 2 \\ \hline b & 3 \\ \hline \end{array}$$

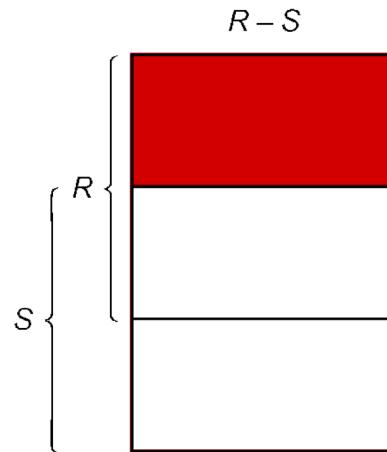
(c) Cartesian product



(d) Union



(e) Intersection



(f) Set difference



Các phép toán trong đại số quan hệ (tt)

T	
A	B
a	1
b	2

U	
B	C
1	x
1	y
3	z

$T \bowtie U$		
A B C		
a	1	x
a	1	y
a	1	y

$T \triangleright_B U$	
A B	
a	1

$T \bowtie_C U$		
A B C		
a	1	x
a	1	y
b	2	

(g) Natural join

(h) Semijoin

(i) Left Outer join

R
Remainder

S

$R \div S$

V	
A B	
a	1
a	2
b	1
b	2
c	1

W
B
1
2

$V \div W$	
A	
a	
b	

(j) Division (shaded area)

Example of division



5 phép toán cơ bản

- ❖ Các phép toán được thực hiện cho quan hệ r trên lược đồ quan hệ R và điều kiện F
- ❖ Điều kiện F là 1 biểu thức luận lý có giá trị true/false, nó bao gồm:
 - Các toán hạng là hằng hoặc tên thuộc tính
 - Các phép toán so sánh =, ≠, <, ≤, >, ≥
 - Các phép toán luận lý not (\neg), and (\wedge), or (\vee)



Phép chọn (Selection or Restriction)

❖ $\sigma_{\text{điều kiện}} (R)$

- Kết quả phép chọn tạo ra một quan hệ trong đó chỉ chứa các bộ của R, mà các bộ đó thỏa mãn điều kiện đã nêu trong phép chọn



Ví dụ về phép chọn

- ❖ **Liệt kê danh sách tất cả nhân viên có lương lớn hơn £10,000**

$\sigma_{\text{salary} > 10000} (\text{Staff})$

staffNo	fName	IName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24- Mar-58	18000	B003
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003



Phép chọn (Selection or Restriction)

❖ $\sigma_{điều\ kiện}(R)$

- Các mệnh đề có thể được nối lại nhờ các phép: \neg , \wedge , \vee
- Phép chọn có tính giao hoán:

$$\sigma_{đk1}(\sigma_{đk2}(R)) = \sigma_{đk2}(\sigma_{đk1}(R)) = \sigma_{đk1 \wedge đk2}(R)$$



Ví dụ về phép chọn

- ❖ Liệt kê danh sách tất cả nhân viên có lương lớn hơn £10,000 ở chi nhánh B003 hoặc các nhân viên có mức lương trên £30,000 ở chi nhánh B005

$\sigma_{(branchNo = 'B003' \wedge salary > 10000) \vee (branchNo = 'B005' \wedge salary > 30000)} (Staff)$



Phép chiếu (Projection)

❖ $\Pi_{\text{col}1, \dots, \text{col}n}(R)$

- Kết quả phép chiếu tạo ra một quan hệ trong đó chỉ chứa một tập con theo chiều dọc của R, tức trích ra theo chiều dọc các giá trị của dữ liệu của một số thuộc tính, và loại bỏ các dữ liệu trùng lắp



Ví dụ về phép chiếu

- ❖ Tạo danh sách lương của tất cả nhân viên, chỉ hiện các thuộc tính staffNo, fName, lName, và salary

$\Pi_{\text{staffNo}, \text{fName}, \text{lName}, \text{salary}}(\text{Staff})$

staffNo	fName	lName	salary
SL21	John	White	30000
SG37	Ann	Beech	12000
SG14	David	Ford	18000
SA9	Mary	Howe	9000
SG5	Susan	Brand	24000
SL41	Julie	Lee	9000



Phép hội (Union)

❖ $R \cup S$

- Hội của 2 quan hệ R và S tạo ra một quan hệ mà các bộ của nó hoặc thuộc R, hoặc thuộc S, hoặc vừa thuộc R vừa thuộc S
 - R và S phải có tính tương thích mới hội với nhau được
-
- ❖ Nếu R có i bộ và S có j bộ thì phép hội có tối đa ($i+j$) bộ



Ví dụ về phép hội

- ❖ **Liệt kê danh sách các thành phố mà ở đó có chi nhánh hay có tài sản cho thuê**

$\Pi_{\text{city}}(\text{Branch}) \cup \Pi_{\text{city}}(\text{PropertyForRent})$

city
London
Aberdeen
Glasgow
Bristol



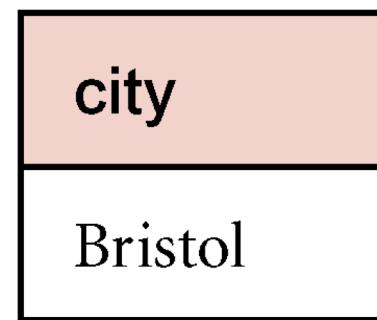
Phép trừ (Set Difference)

❖ R – S

- Tạo một quan hệ chứa các bộ thuộc R nhưng không thuộc S
- R và S phải có tính tương thích

- ❖ **Liệt kê các thành phố có chi nhánh nhưng không có tài sản cho thuê**

$\Pi_{\text{city}}(\text{Branch}) - \Pi_{\text{city}}(\text{PropertyForRent})$





Phép giao (Intersection)

❖ $R \cap S$

- Tạo ra một quan hệ chứa các bộ vừa thuộc R vừa thuộc S
- R và S phải có tính tương thích nhau

❖ Biểu diễn phép giao bằng phép trừ:

$$R \cap S = R - (R - S)$$



Ví dụ về phép giao

- ❖ **Liệt kê các thành phố mà ở đó có chi nhánh và có cả tài sản cho thuê**

$\Pi_{\text{city}}(\text{Branch}) \cap \Pi_{\text{city}}(\text{PropertyForRent})$

city
Aberdeen
London
Glasgow

Cho CSDL Địa lý có cấu trúc như sau:

TINH_TP (MA_T_TP, TEN_T_TP, DT, DS, MIEN)

BIENGIOI (NUOC, MA_T_TP)

LANGGIENG (MA_T_TP, LG)

Viết các câu phép tính quan hệ sau:

1.Xuất ra tên tỉnh, TP cùng với dân số của tỉnh, TP.

2.Xuất ra tên tỉnh, TP, DT cùng với dân số của tỉnh, TP có diện tích
 $\geq 5000 \text{ Km}^2$

3.Xuất ra tên tỉnh, TP cùng với dân số của tỉnh, TP thuộc miền Bắc

4.Xuất ra tên tỉnh, TP cùng với dân số của tỉnh, TP thuộc miền Bắc
mà có diện tích < 5000 .

5.Xuất ra tên các TP là láng giềng với Tp HCM



Phép tích Đề-các (Cartesian product)

❖ R X S

- Tạo một quan hệ trong đó các bộ của nó chính là sự nối kết của từng bộ của R với từng bộ của S



Ví dụ về phép tích Đề-các

- ❖ **Liệt kê các tên và lời nhận xét của các khách hàng đã xem nhà**

$(\Pi_{\text{clientNo}, \text{fName}, \text{IName}}(\text{Client})) \times (\Pi_{\text{clientNo}, \text{propertyNo}, \text{comment}}(\text{Viewing}))$

client.clientNo	fName	IName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR56	PA14	too small
CR76	John	Kay	CR76	PG4	too remote
CR76	John	Kay	CR56	PG4	
CR76	John	Kay	CR62	PA14	no dining room
CR76	John	Kay	CR56	PG36	
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR62	PA14	no dining room
CR56	Aline	Stewart	CR56	PG36	
CR74	Mike	Ritchie	CR56	PA14	too small
CR74	Mike	Ritchie	CR76	PG4	too remote
CR74	Mike	Ritchie	CR56	PG4	
CR74	Mike	Ritchie	CR62	PA14	no dining room
CR74	Mike	Ritchie	CR56	PG36	
CR62	Mary	Tregear	CR56	PA14	too small
CR62	Mary	Tregear	CR76	PG4	too remote
CR62	Mary	Tregear	CR56	PG4	
CR62	Mary	Tregear	CR62	PA14	no dining room
CR62	Mary	Tregear	CR56	PG36	



Ví dụ về phép tích Đè-các và phép chọn

❖ Dùng phép tích Đè-các, rồi phép chọn

$$\sigma_{\text{Client.clientNo}} = \text{viewing.clientNo}((\Pi_{\text{clientNo}, \text{fName}, \text{lName}}(\text{Client})) \times (\Pi_{\text{clientNo}, \text{propertyNo}, \text{comment}}(\text{Viewing})))$$

client.clientNo	fName	lName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR56	PG36	
CR62	Mary	Tregear	CR62	PA14	no dining room

♦ Phép tích Đè-các và phép chọn được làm chung lại gọi là phép kết (*Join*)



Phép Kết (Join)

- ❖ Phép kết phát sinh từ phép tích Đề-các
- ❖ Phép kết thực hiện tương đương với phép chọn trên kết quả của phép tích Đề-các, với một điều kiện chọn phải thỏa



Phép Kết (Join) (tt)

❖ Các dạng khác nhau của phép kết:

- Theta join (phép kết theta)
- Equijoin (phép kết tương đương – là 1 trường hợp đặc biệt của phép kết theta)
- Natural join (phép kết tự nhiên)
- Outer join (phép kết ngoài)
- Semijoin (phép nửa kết)



Phép kết Theta (θ -join)

❖ $R \bowtie_F S$

- Tạo ra 1 quan hệ có các bộ là kết quả từ phép chọn theo điều kiện F trên kết quả của phép tích Đè-các của R và S
- Điều kiện F có dạng $R.a_i \theta S.b_i$ trong đó θ là 1 trong 6 phép so sánh $<$, \leq , $>$, \geq , $=$, \neq



Phép kết Theta (θ -join) (tt)

- ❖ Có thể biểu diễn phép kết Theta:

$$R \bowtie_F S = \sigma_F(R \times S)$$

- ◆ Cấp của phép kết Theta là tổng của cấp 2 quan hệ tham gia
- ◆ Nếu trong điều kiện F sử dụng dấu $=$ thì có tên mới là phép kết tương đương



Ví dụ về phép kết tương đương

- ❖ **Liệt kê các tên và lời nhận xét của các khách hàng đã xem nhà**

$(\Pi_{clientNo, fName, lName}(Client) \bowtie$

$Client.clientNo =$

$Viewing.clientNo (\Pi_{clientNo, propertyNo, comment}(Viewing))$

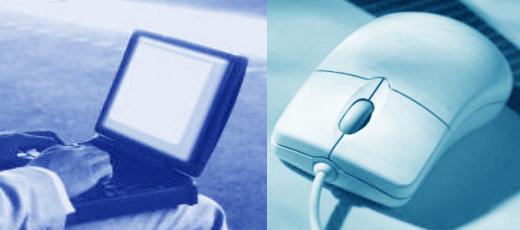
client.clientNo	fName	lName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR56	PG36	
CR62	Mary	Tregear	CR62	PA14	no dining room



Phép kết tự nhiên (Natural Join)

❖  S

- Là phép kết tương đương của 2 quan hệ trên tất cả các thuộc tính chung
- Trong kết quả sẽ bỏ bớt các thuộc tính chung trùng nhau



Ví dụ về phép kết tự nhiên

- ❖ **Liệt kê các tên và lời nhận xét của các khách hàng đã xem nhà**

$(\Pi_{clientNo, fName, lName}(\text{Client})) \bowtie$

$(\Pi_{clientNo, propertyNo, comment}(\text{Viewing}))$

clientNo	fName	lName	propertyNo	comment
CR76	John	Kay	PG4	
CR56	Aline	Stewart	PA14	too small
CR56	Aline	Stewart	PG4	
CR56	Aline	Stewart	PG36	
CR62	Mary	Tregear	PA14	no dining room



Phép kết ngoài (Outer join)

- ❖ Để hiển thị trong kết quả của phép kết các hàng không có các cột kết bằng nhau
- ❖ $R \bowtie S$
 - là phép kết ngoài bên trái: các bộ của R không trùng các giá trị trong các cột chung vẫn có mặt trong kết quả của phép kết



Ví dụ về phép kết ngoài bên trái

- ❖ Lập bảng báo cáo tình trạng các tài sản cho thuê

$\Pi_{\text{propertyNo}, \text{street}, \text{city}}(\text{PropertyForRent}) \setminus \times \text{ Viewing}$

propertyNo	street	city	clientNo	viewDate	comment
PA14	16 Holhead	Aberdeen	CR56	24-May-01	too small
PA14	16 Holhead	Aberdeen	CR62	14-May-01	no dining room
PL94	6 Argyll St	London	null	null	null
PG4	6 Lawrence St	Glasgow	CR76	20-Apr-01	too remote
PG4	6 Lawrence St	Glasgow	CR56	26-May-01	
PG36	2 Manor Rd	Glasgow	CR56	28-Apr-01	
PG21	18 Dale Rd	Glasgow	null	null	null
PG16	5 Novar Dr	Glasgow	null	null	null



Phép nửa kết (Semijoin)

❖ $R \triangleright_F S$

- Tạo ra 1 quan hệ có các bộ là kết quả từ R có tham gia vào phép kết với S
- ◆ Có thể viết lại phép nửa kết bằng phép Chiếu và phép kết:

$$R \triangleright_F S = \Pi_A(R \bowtie_F S)$$

A là các thuộc tính của R



Ví dụ phép nửa kết

- ❖ Lập danh sách chi tiết về các nhân viên làm việc ở chi nhánh ở Glasgow.

Staff ▷ Staff.brancNo = Branch.branchNo and branch.city = 'Glasgow'
Branch

staffNo	fName	IName	position	sex	DOB	salary	branchNo
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24- Mar-58	18000	B003
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003



Phép chia (Division)

❖ $R \div S$

- Tạo ra 1 quan hệ có tập thuộc tính C, mà kết quả từ phép kết từng bộ của C kết hợp với từng bộ của S đều thuộc quan hệ R
- ❖ Biểu diễn cách khác:

$$T_1 \leftarrow \Pi_C(R)$$

$$T_2 \leftarrow \Pi_C((S \times T_1) - R)$$

$$T \leftarrow T_1 - T_2$$



Ví dụ phép chia

- ❖ Tìm tất cả các khách hàng đã xem tất cả các nhà có 3 phòng

$$(\Pi_{\text{clientNo}, \text{propertyNo}}(\text{Viewing})) \div (\Pi_{\text{propertyNo}}(\sigma_{\text{rooms} = 3}(\text{PropertyForRent})))$$
$$\Pi_{\text{clientNo}, \text{propertyNo}}(\text{Viewing})$$

clientNo	propertyNo
CR56	PA14
CR76	PG4
CR56	PG4
CR62	PA14
CR56	PG36

$$\Pi_{\text{propertyNo}}(\sigma_{\text{rooms} = 3}(\text{PropertyForRent}))$$

propertyNo
PG4
PG36

RESULT

clientNo
CR56

❖ Cho các quan hệ sau:

- STUDENT(Id,Name,BirthDate)
- PROFESSOR(Id,Name,Qualification)
- TEACHING(ProfID,CrsCode,Semester)
- COURSE(CrsCode,Name)
- TRANSCRIPT(StudId,CrsCode,Semester, Year,Grade)

1. Hãy liệt kê tên sinh viên và giáo sư mà ID của sinh viên nhỏ hơn Id của giáo sư
2. Hãy liệt kê tên các giáo sư dạy môn học mùa thu 2007 (semester ='F2007')
3. Môn học mà tất cả giáo sư CS đều dạy



Phép toán quan hệ (Relational Calculus)

- ❖ Phép toán quan hệ để diễn đạt “muốn gì?” (what) chứ không diễn đạt phải làm gì (how) để có được điều mong muốn
- ❖ Trong phép toán luận lý tiền tố (hay phép toán vị từ), vị từ (predicate) là 1 hàm có đối số nằm phía sau
- ❖ Phép toán vị từ có kết quả là đúng (true) hay sai (false)
- ❖ Phép toán vị từ trên CSDL có 2 loại: phép toán trên bộ và phép toán trên miền của thuộc tính



Phép toán quan hệ trên bộ

- ❖ Có mục đích tìm ra các bộ làm cho vị từ có giá trị true.
Sử dụng các biến trên bộ để tìm
 - ❖ Biến trên bộ là biến nhận giá trị là 1 bộ của quan hệ
 - ❖ Ví dụ: giá trị của 1 biến S có thể là 1 bộ của quan hệ Staff:
 $\text{Staff}(S)$
 - ❖ Tìm tất cả giá trị S: biểu thức sau phải thỏa (true) :
 $\{S \mid P(S)\}$



Ví dụ phép toán quan hệ trên bộ

- ❖ Lập danh sách chi tiết về các nhân viên có lương lớn hơn 10000
 $\{S \mid \text{Staff}(S) \wedge S.\text{salary} > 10000\}$
- ❖ Lập danh sách 1 cột lương chỉ lấy giá trị lớn hơn 10000
 $\{S.\text{salary} \mid \text{Staff}(S) \wedge S.\text{salary} > 10000\}$
- ❖ Tìm ngày sinh và địa chỉ của nhân viên có tên là Ann



Phép toán quan hệ trên bộ

- ❖ Có thể sử dụng lượng từ (*quantifier*) để *cho biết vị từ sẽ áp dụng trên bao nhiêu thể hiện*:
 - Lượng từ “tồn tại”: \exists
 - Lượng từ “với mọi”: \forall



Ví dụ phép toán quan hệ trên bộ

- ❖ Lượng từ tồn tại dùng trong công thức để bảo đảm có ít nhất 1 giá trị đúng:

$$\text{Staff}(S) \wedge (\exists B)(\text{Branch}(B) \wedge (B.\text{branchNo} = S.\text{branchNo}) \wedge B.\text{city} = \text{'London'})$$

- ❖ Có nghĩa là: Tồn tại ít nhất 1 chi nhánh (1 bộ) trong bảng Branch có BranchNo bằng với BranchNo của bộ S đang xét.



Phép toán quan hệ trên bộ

- ❖ Lượng từ “với mọi” \forall được dùng trên mỗi bộ như:
 $(\forall B) (Branch(B) \wedge (B.city \neq 'Paris'))$
- ❖ Có nghĩa ‘Các bộ dữ liệu của các chi nhánh, mà các chi nhánh đó có địa chỉ thành phố không thuộc Paris’
- ❖ Ngược lại: $\sim(\exists B) (Branch(B) \wedge (B.city = 'Paris'))$ có nghĩa ‘không có chi nhánh nào có địa chỉ ở Paris’.



Phép toán quan hệ trên bộ

- ❖ Các công thức phải tránh sự mơ hồ và có ý nghĩa
- ❖ Một công thức tốt được cấu thành từ các thành tố:
 - $R(S_i)$, với S_i là 1 biến trên bộ và R là 1 quan hệ
 - $S_i.a_1 \theta S_j.a_2$
 - $S_i.a_1 \theta c$
- ❖ Một công thức có thể cấu thành bằng các công thức lồng nhau:
 - 1 thành tố là 1 công thức
 - Nếu F_1 và F_2 là công thức thì có các công thức: $F_1 \wedge F_2$, $F_1 \vee F_2$ và $\sim F_1$
 - Nếu F là công thức và X là 1 biến thì $(\exists X)(F)$ và $(\forall X)(F)$ cũng là công thức



Ví dụ về Phép toán quan hệ trên bộ

- a) Liệt kê tên tất cả các giám đốc kiểm được hơn £25,000 trong 1 năm:

$$\{S.\text{fName}, S.\text{lName} \mid \text{Staff}(S) \wedge S.\text{position} = \text{'Manager'} \wedge S.\text{salary} > 25000\}$$

- b) Liệt kê các nhân viên quản lý tài sản ở Glasgow:

$$\{S \mid \text{Staff}(S) \wedge (\exists P) (\text{PropertyForRent}(P) \wedge (P.\text{staffNo} = S.\text{staffNo}) \wedge P.\text{city} = \text{'Glasgow'})\}$$



Ví dụ về Phép toán quan hệ trên bộ

c) **Liệt kê tên các nhân viên hiện không quản lý bất kỳ tài sản nào:**

$$\{S.fName, S.lName \mid Staff(S) \wedge (\neg(\exists P) (PropertyForRent(P) \wedge (S.staffNo = P.staffNo)))\}$$

Hoặc:

$$\{S.fName, S.lName \mid Staff(S) \wedge (\forall P) (\neg PropertyForRent(P) \vee \neg(S.staffNo = P.staffNo)))\}$$



Ví dụ về Phép toán quan hệ trên bộ

- ❖ **Liệt kê tên khách hàng đã xem nhà ở Glasgow:**

{C.fName, C.lName | Client(C) \wedge (($\exists V$)($\exists P$)
(Viewing(V) \wedge PropertyForRent(P) \wedge (
C.clientNo = V.clientNo) \wedge
(V.propertyNo=P.propertyNo) \wedge P.city = 'Glasgow'))}



Chú ý về Phép toán quan hệ trên bộ

- ❖ Các biểu thức có thể tạo 1 tập vô hạn không xác định:
 $\{S \mid \sim\text{Staff}(S)\}$
- ❖ Để tránh lỗi này, cần thêm giới hạn về miền giá trị của kết quả biểu thức



Phép toán quan hệ trên miền (Domain Relational Calculus)

- Cơ sở dữ liệu - Đại số quan hệ
- ❖ Dùng các biến có giá trị từ miền chứ không phải từ bộ của quan hệ
 - ❖ Nếu $F(d_1, d_2, \dots, d_n)$ là công thức tạo thành từ các thành tố, và d_1, d_2, \dots, d_n là các biến trên miền thì:
$$\{d_1, d_2, \dots, d_n \mid F(d_1, d_2, \dots, d_n)\}$$
Là biểu thức của phép toán quan hệ trên miền



Ví dụ về Phép toán quan hệ trên miền

- a) Tìm tên của tất cả các giám đốc kiểm được hơn £25,000 trong 1 năm

{fN, IN | $\exists sN, posn, sex, DOB, sal, bN$)

(Staff (sN, fN, IN, posn, sex, DOB, sal, bN) \wedge
posn = ‘Manager’ \wedge sal > 25000)}



Ví dụ về Phép toán quan hệ trên miền

b) **Liệt kê các nhân viên quản lý tài sản ở Glasgow:**

{sN, fN, IN, posn, sex, DOB, sal, bN |
 $(\exists sN1, cty)(Staff(sN, fN, IN, posn, sex, DOB, sal, bN) \wedge$
 $(PropertyForRent(pN, st, cty, pc, typ, rms,$
 $rnt, oN, sN1, bN1) \wedge$
 $(sN = sN1) \wedge$
 $cty = 'Glasgow')$ }



Ví dụ về Phép toán quan hệ trên miền

- c) **Liệt kê tên các nhân viên hiện không quản lý bất kỳ tài sản nào:**

$\{fN, lN \mid (\exists sN)$
 $(Staff(sN, fN, lN, posn, sex, DOB, sal, bN) \wedge$
 $(\sim(\exists sN1) (PropertyForRent(pN, st, cty, pc, typ,$
 $rms, rnt, oN, sN1, bN1) \wedge (sN = sN1))))\}$



Ví dụ về Phép toán quan hệ trên miền

d) **Liệt kê tên khách hàng đã xem nhà ở Glasgow:**

{fN, IN | ($\exists cN, cN1, pN, pN1, cty$)
(Client(cN, fN, IN, tel, pT, mR) \wedge
Viewing(cN1, pN1, dt, cmt) \wedge
PropertyForRent(pN, st, cty, pc, typ,
 rms, rnt, oN, sN, bN) \wedge
(cN = cN1) \wedge (pN = pN1) \wedge cty = ‘Glasgow’)}}



Phép toán quan hệ trên miền

- ❖ Mọi biểu thức đại số đều có biểu thức phép toán quan hệ tương đương và ngược lại

- ❖ Cho các quan hệ sau:
 - STUDENT(Id,Name,BirthDate)
 - PROFESSOR(Id,Name,Qualification)
 - TEACHING(ProfID,CrsCode,Semester)
 - COURSE(CrsCode,Name)
 - TRANSCRIPT(StudId,CrsCode,Semester, Year,Grade)

 1. Liệt kê tất cả các môn học mà mọi sinh viên đều học
 2. Liệt kê tên của tất cả giáo sư đã dạy môn MGT123

- ♦ Liệt kê tên của tất cả giáo sư đã dạy môn MGT123
 $\{P.Name | \text{PROFESSOR}(P) \wedge \exists T (\text{TEACHING}(T) \wedge (P.Id = T.ProfId \wedge T.CrsCode = 'MGT123'))\}$
- ♦ Câu lệnh SQL tương ứng
- ```
SELECT P.Name
FROM PROFESSOR P, TEACHING T
WHERE P.Id= T.ProfId AND T.CrsCode = 'MGT123'
```



# Các ngôn ngữ khác

- ❖ Các ngôn ngữ phi thủ tục dùng các quan hệ đưa vào đầu vào để lấy được đầu ra theo ý muốn (ví dụ ngôn ngữ SQL).
- ❖ Các ngôn ngữ đồ họa trình bày với người dùng cấu trúc của các quan hệ, người dùng sẽ thao tác trên đó như là làm ví dụ cho chương trình hiểu cần phải trả về kết quả dữ liệu như thế nào cho đúng với mong muốn (ví dụ ngôn ngữ QBE trong MS Access)



# Các ngôn ngữ khác

- ❖ Ngôn ngữ thế hệ thứ tư (4GLs) ít dùng câu lệnh, thay vào đó là dùng giao diện trực quan.
- ❖ Ngôn ngữ thế hệ thứ năm (5GLs) sử dụng ngôn ngữ tự nhiên, nhưng ngôn ngữ này chưa hoàn thiện, còn đang nghiên cứu ở bước đầu.