

INT1313 – Cơ Sở Dữ Liệu

Mô hình Dữ Liệu Quan hệ và SQL
Đại số Quan hệ – Phép tính Quan hệ

Giảng viên: Lê Hà Thanh

lehathanh@ptithcm.edu.vn

Hành trình

- Mô hình Dữ liệu Quan hệ
- Các Ràng buộc CSDL Quan hệ
- SQL
- **Đại số quan hệ và Phép tính quan hệ**
- Thiết kế Lược đồ CSDL quan hệ từ lược đồ khái niệm

Nội dung

- Ngôn ngữ hình thức cho mô hình quan hệ
- Đại số quan hệ
 - Toán tử quan hệ đơn phân: SELECT và PROJECT
 - Các toán tử tập hợp
 - Phép JOIN và các toán tử nhị phân
 - Các hàm tổng hợp (aggregate), OUTER JOIN, OUT UNION
- Phép tính quan hệ
 - Phép tính quan hệ trên các bộ dữ liệu
 - Phép tính quan hệ trên các miền giá trị
 - Phép tính vị từ

CSDL COMPANY

COMPANY: Lược đồ CSDL quan hệ

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

COMPANY: Lược đồ CSDL quan hệ

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

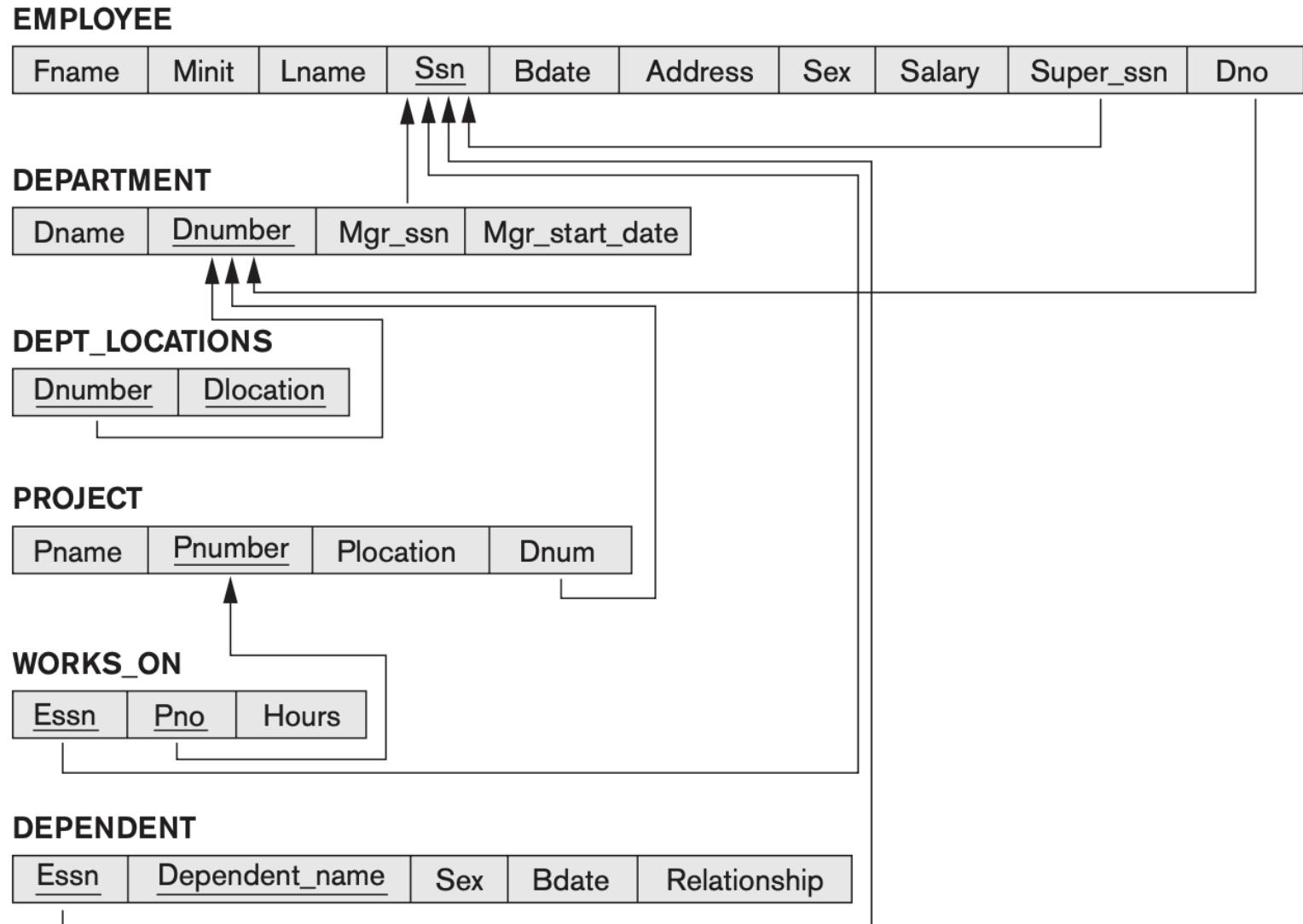
<u>Pname</u>	<u>Pnumber</u>	<u>Plocation</u>	<u>Dnum</u>
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	<u>Sex</u>	<u>Bdate</u>	<u>Relationship</u>
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

COMPANY:

Lược đồ CSDL quan hệ - Ràng buộc toàn vẹn tham chiếu



Đại số Quan hệ (Relational Algebra)

Ngôn ngữ truy vấn hình thức quy định cách thức truy xuất dữ liệu trong mô hình quan hệ.

Phép Chọn - SELECT

Chọn ra một tập con (*subset*) gồm các bộ trong một quan hệ thỏa mãn điều kiện lựa chọn (**selection condition - filter**).

$$\sigma_{<\text{selection condition}>}(R)$$

Trong đó,

- σ (sigma): biểu thị phép chọn SELECT
- $<\text{selection condition}>$: biểu thức Boolean (điều kiện) được chỉ định trên các thuộc tính của quan hệ R .

- Tương đương phép phân vùng theo chiều ngang chia một quan hệ thành hai phần: (1) phần gồm các bộ thỏa mãn điều kiện chọn và (2) phần gồm các bộ còn lại.

Phép Chọn – Ví dụ

VD 1: “*Chọn các bộ trong quan hệ EMPLOYEE có mã phòng ban là 4.*”

$$\sigma_{Dno=4}(EMPLOYEE)$$

VD 2: “*Chọn các bộ trong quan hệ EMPLOYEE có mức lương trên \$30,000.*”

$$\sigma_{Salary > 30000}(EMPLOYEE)$$

Biểu thức điều kiện

$< \text{attribute name} > < \text{comparison op} > < \text{constant value} >$

$< \text{attribute name} > < \text{comparison op} > < \text{attribute name} >$

Trong đó,

$< \text{attribute name} >$: tên thuộc tính trong R ,

$< \text{comparison op} >$: toán tử so sánh $\{=, <, \leq, >, \geq, \neq\}$,

$< \text{constant value} >$: giá trị hằng trong miền giá trị của thuộc tính.

- Dùng các toán tử luận lý: *and, or, not*

Biểu thức điều kiện – Ví dụ

“Chọn các bộ dữ liệu nhân viên đang làm ở phòng ban số 4 và có mức lương trên \$25,000, hoặc đang làm ở phòng ban số 5 và có mức lương trên \$30,000.”

$\sigma_{(Dno=4 \text{ AND } Salary > 25000) \text{ OR } (Dno=5 \text{ AND } Salary > 30000)}(EMPLOYEE)$

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5

Toán tử luận lý – Bảng chân trị

AND cond1 cond2	TRUE	FALSE
TRUE	TRUE	FALSE
FALSE	FALSE	FALSE

OR cond1 cond2	TRUE	FALSE
TRUE	TRUE	TRUE
FALSE	TRUE	FALSE

cond	TRUE	FALSE
NOT cond	FALSE	TRUE

Phép Chọn – Các tính chất

- Là phép toán đơn phân (**unary**), chỉ áp dụng lên một quan hệ.
- Được áp dụng cho từng bộ dữ liệu riêng biệt.
- Độ phức tạp (số thuộc tính) của kết quả tương đương với bậc của quan hệ R .
- Số bộ dữ liệu trong kết quả chọn luôn nhỏ hơn hoặc bằng số bộ trong R .

$$|\sigma_C(R)| \leq |R|, \text{ với } \forall C$$

Phép Chọn – Tính giao hoán

- Phép chọn có tính giao hoán, nghĩa là:

$$\sigma_{<cond1>}(\sigma_{<cond2>}(R)) = \sigma_{<cond2>}(\sigma_{<cond1>}(R))$$

→ Chuỗi các phép chọn có thể được thực hiện theo bất kỳ thứ tự nào.

- Có thể kết hợp một chuỗi các phép chọn thành một phép chọn duy nhất bằng cách dùng điều kiện kết hợp (AND) như sau:

$$\begin{aligned} &\sigma_{<cond1>}(\sigma_{<cond2>}(\dots (\sigma_{<condn>}(R)) \dots)) \\ &= \sigma_{<cond1>} AND <cond2> AND \dots AND <condn>(R) \end{aligned}$$

Biểu diễn Điều kiện của phép Chọn trong SQL

Điều kiện của phép chọn được chỉ định trong mệnh đề *WHERE* của lệnh truy vấn.

Ví dụ:

$$\sigma_{Dno=4 \text{ AND } Salary > 25000}(\text{EMPLOYEE})$$

Sẽ tương ứng với truy vấn SQL sau:

```
SELECT *
FROM EMPLOYEE
WHERE Dno = 4 AND Salary > 25000;
```

Phép Chiếu - PROJECT

Chọn ra một số cột (*columns*) nhất định từ bảng và loại bỏ các cột khác (còn gọi là chiếu quan hệ trên các thuộc tính này)

$$\pi_{<\text{attribute list}>}(R)$$

Trong đó,

- π (pi): biểu thị phép chiếu PROJECT.
- $<\text{attribute list}>$: danh sách con các thuộc tính của quan hệ R .
- Tương đương phép phân vùng một quan hệ theo chiều dọc thành hai phần: (1) phần gồm các cột (thuộc tính) được quan tâm và chứa kết quả của phép chiếu, và (2) phần chứa các cột (thuộc tính) khác.

Phép Chiếu – Ví dụ 1

“Liệt kê danh sách các nhân viên theo họ, tên và mức lương của từng người.”

$$\pi_{Lname, Fname, Salary}(EMPLOYEE)$$

Lname	Fname	Salary
Smith	John	30000
Wong	Franklin	40000
Zelaya	Alicia	25000
Wallace	Jennifer	43000
Narayan	Ramesh	38000
English	Joyce	25000
Jabbar	Ahmad	25000
Borg	James	55000

Phép Chiếu – Các tính chất (1/2)

- R là biểu thức đại số quan hệ có kết quả là một quan hệ, tương đương tên một quan hệ CSDL.
- Kết quả của phép chiếu chỉ bao gồm các thuộc tính được chỉ định trong $\langle attribute\ list \rangle$ và được trình bày theo đúng thứ tự được khai báo.
→ bậc của kết quả chiếu là số thuộc tính trong $\langle attribute\ list \rangle$
- Phép chiếu loại bỏ các bộ bị trùng lặp nếu danh sách thuộc tính của phép chiếu chỉ gồm các thuộc tính không phải khóa của R .
→ Kết quả chiếu chỉ bao gồm các bộ dữ liệu phân biệt và là một quan hệ hợp lệ.
- Số bộ dữ liệu trong kết quả chiếu luôn nhỏ hơn hoặc bằng số bộ trong R .

$$|\pi_{\langle attribute\ list \rangle}(R)| \leq |R|, \text{ với } \forall \langle attribute\ list \rangle$$

Phép Chiếu – Các tính chất (2/2)

- Nếu $\langle \text{attribute list} \rangle$ là siêu khóa của R :

$$|\pi_{\langle \text{attribute list} \rangle}(R)| = |R|$$

Hơn nữa, nếu $\langle list2 \rangle$ chứa các thuộc tính trong $\langle list1 \rangle$, ta có:

$$\pi_{\langle list1 \rangle}(\pi_{\langle list2 \rangle}(R)) = \pi_{\langle list1 \rangle}(R)$$

Điều ngược lại, nghĩa là có ít nhất một thuộc tính trong $\langle list1 \rangle$ không có trong $\langle list2 \rangle$, khiến vé bên trái không đúng.

- Phép chiếu không có tính giao hoán.

Phép Chiếu – Ví dụ 2

“Liệt kê thông tin giới tính của nhân viên và mức lương nhận được.”

$$\pi_{Sex, Salary}(EMPLOYEE)$$

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Sex	Salary
M	30000
M	40000
F	25000
F	43000
M	38000
M	25000
M	55000

Biểu diễn Danh sách thuộc tính của phép Chiếu trong SQL

Danh sách thuộc tính của phép chiếu được chỉ định trong mệnh đề *SELECT* của lệnh truy vấn.

Ví dụ:

$$\pi_{Sex, Salary}(EMPLOYEE)$$

Sẽ tương ứng với truy vấn SQL sau:

```
SELECT DISTINCT Sex, Salary  
FROM EMPLOYEE;
```

Chú ý: từ khóa DISTINCT để loại bỏ các bộ trùng lặp.

Trình tự thực hiện phép toán trong biểu thức đại số quan hệ

Để thực hiện liên tiếp nhiều phép toán quan hệ, ta có thể thực hiện:

- Lồng các phép toán vào nhau thành một biểu thức duy nhất – biểu thức nội tuyến (**in-line expression**).

Ví dụ:

$$\pi_{Fname,Fname,Salary}(\sigma_{Dno=5}(EMPLOYEE))$$

- Gán tên cho quan hệ là kết quả phép toán trung gian rồi thực hiện phép toán tiếp theo trên quan hệ trung gian này.

Ví dụ:

$$DEP5_EMPL \leftarrow \sigma_{(Dno=5)}(EMPLOYEE)$$
$$RESULT \leftarrow \pi_{(Fname,Lname,Salary)}(DEP5_EMPS)$$

Fname	Lname	Salary
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

Đổi tên thuộc tính trong quan hệ kết quả và quan hệ trung gian

Ví dụ:

$$TEMP \leftarrow \sigma_{Dno=5}(EMPLOYEE)$$
$$R(First_name, Last_name, Salary) \leftarrow \pi_{Fname, Lname, Salary}(TEMP)$$

TEMP										
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno	
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5	
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5	
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5	
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5	

R

First_name	Last_name	Salary
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

Phép Đổi tên – RENAME

Là phép toán đơn phân đổi tên quan hệ và/hoặc tên thuộc tính của quan hệ.

R là một quan hệ có bậc n , phép đổi tên ρ được biểu diễn bằng một trong các dạng sau:

$$\rho_{S(B_1, B_2, \dots, B_n)}(R)$$

hoặc,

$$\rho_S(R)$$

hoặc,

$$\rho_{(B_1, B_2, \dots, B_n)}(R)$$

trong đó,

- ρ (rho): biểu diễn phép đổi tên RENAME,
- S : tên mới của quan hệ,
- B_1, B_2, \dots, B_n : các tên mới của các thuộc tính.

- Nếu thứ tự các thuộc tính trong R là (A_1, A_2, \dots, A_n) thì mỗi A_i được tương ứng đổi tên thành B_i .

Phép Đổi tên trong SQL – AS

SQL dùng từ khóa **AS** để đổi tên mới cho quan hệ và các thuộc tính.

Ví dụ:

```
SELECT E.Fname AS First_name, E.Lname AS Last_name, E.Salary AS Salary  
FROM EMPLOYEE AS E  
WHERE E.Dno = 5;
```

Các phép toán Đại số Quan hệ từ Lý thuyết Tập hợp

Ví dụ 1

“Lấy số an sinh xã hội của tất cả nhân viên hoặc đang làm việc trong phòng ban số 5 hoặc đang trực tiếp quản lý một nhân viên khác đang làm việc trong phòng ban số 5.”

```
DEP5_EMPS ← σDno=5(EMPLOYEE)
RESULT1 ← πSsn(DEP5_EMPS)
RESULT2(Ssn) ← πSuper_ssн(DEP5_EMPS)
RESULT ← RESULT1 ∪ RESULT2
```

RESULT1	RESULT2	RESULT
Ssn	Ssn	Ssn
123456789	333445555	123456789
333445555	888665555	333445555
666884444		666884444
453453453		453453453
		888665555

Có thể viết thành,

$$Result ← π_{Ssn}(\sigma_{Dno=5}(EMPLOYEE)) ∪ π_{Super_ssн}(\sigma_{Dno=5}(EMPLOYEE))$$

Tính tương thích hợp nhất (Tương thích kiểu) giữa hai quan hệ

Định nghĩa: Hai quan hệ $R(A_1, A_2, \dots, A_n)$ và $S(B_1, B_2, \dots, B_n)$ được gọi là tương thích hợp nhất (tương thích kiểu) nếu chúng có cùng bậc n và $\text{dom}(A_i) = \text{dom}(B_i), 1 \leq i \leq n$.

Nghĩa là, hai quan hệ có cùng số lượng các thuộc tính và mỗi cặp thuộc tính tương ứng với nhau có cùng miền giá trị.

Phép Hợp (UNION), Giao (INTERSECTION) và Trừ tập hợp (MINUS)

- Trong lý thuyết tập hợp: Là các phép toán nhị phân, áp dụng cho hai tập hợp.
- Điều kiện áp dụng trong CSDL quan hệ: Hai quan hệ trong các phép toán này phải có cùng kiểu của các bộ dữ liệu
 - thỏa mãn điều kiện tương thích hợp nhất (*union compatibility*), hay tương thích kiểu (*type compatibility*).

Phép Hợp (UNION), Giao (INTERSECTION) và Trừ tập hợp (MINUS)

Định nghĩa: Cho hai quan hệ R và S tương thích hợp nhất,

- Phép hợp (UNION) giữa hai quan hệ, ký hiệu $R \cup S$, là quan hệ kết quả chứa tất cả các bộ xuất hiện trong cả R hoặc S , hoặc trong R và S . Các bộ trùng lặp sẽ bị loại bỏ.
- Phép giao (INTERSECTION) giữa hai quan hệ, ký hiệu $R \cap S$, là quan hệ kết quả chứa tất cả các bộ có trong cả R và S .
- Phép trừ (MINUS), ký hiệu $R - S$, là quan hệ kết quả gồm tất cả các bộ có trong R nhưng không có trong S .

Ví dụ 2 – minh họa các phép toán

- (a) Hai quan hệ STUDENT và INSTRUCTOR tương tính hợp nhất,
- (b) STUDENT \cup INSTRUCTOR,
- (c) STUDENT \cap INSTRUCTOR,
- (d) STUDENT – INSTRUCTOR,
- (e) INSTRUCTOR – STUDENT

(b)

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert
John	Smith
Ricardo	Browne
Francis	Johnson

(c)

Fn	Ln
Susan	Yao
Ramesh	Shah

(d)

Fn	Ln
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

(a) STUDENT

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

Fname	Lname
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

(e)

Fname	Lname
John	Smith
Ricardo	Browne
Francis	Johnson

Các tính chất

- Phép hợp và phép giao có tính giao hoán; nghĩa là,

$$R \cup S = S \cup R \text{ và } R \cap S = S \cap R$$

- Phép hợp và phép giao có tính kết hợp \rightarrow phép toán n-phân áp dụng để tính toán cho nhiều quan hệ; nghĩa là,

$$R \cup (S \cup T) = (R \cup S) \cup T \text{ và } R \cap (S \cap T) = (R \cap S) \cap T$$

- Phép trừ không có tính giao hoán; nghĩa là, một cách tổng quát,

$$R - S \neq S - R$$

- Phép giao có thể được viết lại dưới dạng:

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

Các phép toán trong SQL

Tương đương các phép:

- **UNION**
- **INTERSECT**
- **EXCEPT**

Ngoài ra, các phép tính đa tập hợp sau sẽ không loại bỏ các bộ bị trùng khỏi quan hệ kết quả:

- **UNION ALL**
- **INTERSECT ALL**
- **EXCEPT ALL**

Phép toán Tích Decartes (Tích chéo)

- Trong lý thuyết tập hợp: tích Decartes là phép toán nhị phân.
- Trong CSDL quan hệ, tích Decartes không yêu cầu các quan hệ phải tương tích hợp nhất với nhau.

Phép toán Tích Decartes (Tích chéo)

Định nghĩa: Cho hai quan hệ $R(A_1, A_2, \dots, A_n)$ và $S(B_1, B_2, \dots, B_m)$.

Tích Decartes $R(A_1, A_2, \dots, A_n) \times S(B_1, B_2, \dots, B_m)$ là một quan hệ Q bậc $(n + m)$ $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$, với các thuộc tính được sắp theo đúng thứ tự này.

- Trong Q có một bộ tương ứng cho mỗi tổ hợp các bộ – một từ R và một từ S .

Do đó, nếu R có n_R bộ ($|R| = n_R$) và S có n_S bộ ($|S| = n_S$), thì $R \times S$ sẽ có $n_R * n_S$ bộ.

Tích Decartes n-phân

Tạo các bộ mới bằng cách nối tất cả các tổ hợp có thể có của các bộ lấy từ n quan hệ thành tố.

Chú ý: kết quả của tích Decartes thường không có ý nghĩa.

→ Phải chú ý các ràng buộc để trích ra thông tin có ý nghĩa.

Ví dụ

“Lấy danh sách tên các người phụ thuộc của các nhân viên nữ.”

1. $FEMALE_EMPS \leftarrow \sigma_{Sex='F'}(EMPLOYEE)$
2. $EMPNAMES \leftarrow \pi_{Fname, Lname, Ssn}(FEMALE_EMPS)$
3. $EMP_DEPENDENTS \leftarrow EMPNAMES \times DEPENDENT$
4. $ACTUAL_DEPENDENTS \leftarrow \sigma_{Ssn=Essn}(EMP_DEPENDENTS)$
5. $RESULT \leftarrow \pi_{Fname, Lname, Dependent_name}(ACTUAL_DEPENDENTS)$

FEMALE_EMPS

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5

EMPNAMEs

Fname	Lname	Ssn
Alicia	Zelaya	999887777
Jennifer	Wallace	987654321
Joyce	English	453453453

EMP_DEPENDENTS

Fname	Lname	Ssn	Essn	Dependent_name	Sex	Bdate	...
Alicia	Zelaya	999887777	333445555	Alice	F	1986-04-05	...
Alicia	Zelaya	999887777	333445555	Theodore	M	1983-10-25	...
Alicia	Zelaya	999887777	333445555	Joy	F	1958-05-03	...
Alicia	Zelaya	999887777	987654321	Abner	M	1942-02-28	...
Alicia	Zelaya	999887777	123456789	Michael	M	1988-01-04	...
Alicia	Zelaya	999887777	123456789	Alice	F	1988-12-30	...
Alicia	Zelaya	999887777	123456789	Elizabeth	F	1967-05-05	...
Jennifer	Wallace	987654321	333445555	Alice	F	1986-04-05	...
Jennifer	Wallace	987654321	333445555	Theodore	M	1983-10-25	...
Jennifer	Wallace	987654321	333445555	Joy	F	1958-05-03	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...
Jennifer	Wallace	987654321	123456789	Michael	M	1988-01-04	...
Jennifer	Wallace	987654321	123456789	Alice	F	1988-12-30	...
Jennifer	Wallace	987654321	123456789	Elizabeth	F	1967-05-05	...
Jennifer	Wallace	987654321	333445555	Alice	F	1986-04-05	...
Jennifer	Wallace	987654321	333445555	Theodore	M	1983-10-25	...
Jennifer	Wallace	987654321	333445555	Joy	F	1958-05-03	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...
Jennifer	Wallace	987654321	123456789	Michael	M	1988-01-04	...
Jennifer	Wallace	987654321	123456789	Alice	F	1988-12-30	...
Jennifer	Wallace	987654321	123456789	Elizabeth	F	1967-05-05	...
Jennifer	Wallace	987654321	333445555	Alice	F	1986-04-05	...
Jennifer	Wallace	987654321	333445555	Theodore	M	1983-10-25	...
Jennifer	Wallace	987654321	333445555	Joy	F	1958-05-03	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...
Jennifer	Wallace	987654321	123456789	Michael	M	1988-01-04	...
Jennifer	Wallace	987654321	123456789	Alice	F	1988-12-30	...
Jennifer	Wallace	987654321	123456789	Elizabeth	F	1967-05-05	...
Joyce	English	453453453	123456789	Alice	F	1988-12-30	...
Joyce	English	453453453	123456789	Elizabeth	F	1967-05-05	...

ACTUAL_DEPENDENTS

Fname	Lname	Ssn	Essn	Dependent_name	Sex	Bdate	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...

RESULT

Fname	Lname	Dependent_name
Jennifer	Wallace	Abner

Tích Decartes trong SQL

- Tích Decartes tương đương khai báo **CROSS JOIN** trong câu truy vấn SQL.
- Nếu có hai bảng trong mệnh đề FROM và không có điều kiện kết (join) trong mệnh đề WHERE, thì kết quả sẽ là tích Decartes của hai bảng thành phần.

Các phép quan hệ nhị phân:
Phép kết (JOIN)
Phép chia (DIVISION)

Phép kết – JOIN

Phép kết (JOIN), ký hiệu \bowtie , được sử dụng để kết hợp các bộ có liên quan từ hai quan hệ thành các bộ “dài hơn”.

- Cho phép xử lý các mối quan hệ giữa các “quan hệ”

JOIN - Ví dụ

“Lấy tên người quản lý của từng phòng ban.”

$$\begin{aligned} \text{DEPT_MGR} &\leftarrow \text{DEPARTMENT} \bowtie_{\text{Mgr_ssn}=\text{Ssn}} \text{EMPLOYEE} \\ \text{RESULT} &\leftarrow \pi_{\text{Dname}, \text{Lname}, \text{Fname}}(\text{DEPT_MGR}) \end{aligned}$$

DEPT_MGR

Dname	Dnumber	Mgr_ssn	...	Fname	Minit	Lname	Ssn	...
Research	5	333445555	...	Franklin	T	Wong	333445555	...
Administration	4	987654321	...	Jennifer	S	Wallace	987654321	...
Headquarters	1	888665555	...	James	E	Borg	888665555	...

- Mgr_ssn là khóa ngoại của DEPARTMENT, liên kết tới Ssn là khóa chính của EMPLOYEE.

Phép kết – JOIN

Phép kết JOIN trên hai quan hệ $R(A_1, A_2, \dots, A_n)$ và $S(B_1, B_2, \dots, B_m)$ là

$$R \bowtie_{\langle join\ condition \rangle} S$$

Cho kết quả là quan hệ Q có $n + m$ thuộc tính
 $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ theo thứ tự này;

- Q có một bộ tương ứng với mỗi tổ hợp các bộ - một từ R và một từ S
– bất cứ khi nào tổ hợp đó thỏa mãn điều kiện kết.

Điều kiện kết – THETA JOIN

Điều kiện kết có dạng:

$< condition > AND < condition > AND \dots AND < condition >$

Trong đó, mỗi $< condition >$ là một biểu thức so sánh $A_i \theta B_j$, với A_i là một thuộc tính của R , B_j là một thuộc tính của S , A_i và B_j có chung miền giá trị, và θ là một phép toán so sánh $\{=, <, \leq, >, \geq, \neq\}$

➔ Phép kết với điều kiện kết tổng quát trên được gọi là phép kết Theta, **THETA JOIN**.

Phép kết “bằng” - EQUIJOIN

Định nghĩa: Phép kết JOIN chỉ có toán tử so sánh bằng = được gọi là phép kết bằng – EQUIJOIN.

$\text{DEPT_MGR} \leftarrow \text{DEPARTMENT} \bowtie_{\text{Mgr_ssn}=\text{Ssn}} \text{EMPLOYEE}$

DEPT_MGR

Dname	Dnumber	Mgr_ssn	...	Fname	Minit	Lname	Ssn	...
Research	5	333445555	...	Franklin	T	Wong	333445555	...
Administration	4	987654321	...	Jennifer	S	Wallace	987654321	...
Headquarters	1	888665555	...	James	E	Borg	888665555	...

EQUIJOIN - Dư thừa thuộc tính kết nối

$$\text{DEPT_MGR} \leftarrow \text{DEPARTMENT} \bowtie_{\text{Mgr_ssn}=\text{Ssn}} \text{EMPLOYEE}$$

- Mgr_ssn là khóa ngoại của DEPARTMENT, liên kết tới Ssn là khóa chính của EMPLOYEE.
- Kết quả của EQUIJOIN xuất hiện thuộc tính dư thừa (**Mgr_ssn** hoặc **Ssn**) do có giá trị giống hệt nhau trong từng bộ dữ liệu.

DEPT_MGR

Dname	Dnumber	Mgr_ssn	...	Fname	Minit	Lname	Ssn	...
Research	5	333445555	...	Franklin	T	Wong	333445555	...
Administration	4	987654321	...	Jennifer	S	Wallace	987654321	...
Headquarters	1	888665555	...	James	E	Borg	888665555	...

Phép kết tự nhiên – NATURAL JOIN

Phép kết tự nhiên (NATURAL JOIN), ký hiệu *, được sử dụng để loại bỏ thuộc tính thừa trong một điều kiện của EQUIJOIN.

- NATURAL JOIN yêu cầu hai thuộc tính nối (hoặc mỗi cặp thuộc tính nối) phải có cùng tên trong cả hai quan hệ.
- Nếu cặp thuộc tính nối khác tên, phải thực hiện đổi tên thuộc tính trước.

NATURAL JOIN – Ví dụ

"Kết mỗi bộ trong PROJECT với bộ trong DEPARTMENT đang kiểm soát dự án."

Cách 1:

PROJ_DEPT \leftarrow PROJECT* ρ (Dname, Dnum, Mgr_ssn, Mgr_start_date)(DEPARTMENT)

Cách 2:

DEPT \leftarrow ρ (Dname, Dnum, Mgr_ssn, Mgr_start_date)(DEPARTMENT)

PROJ_DEPT \leftarrow PROJECT *DEPT

Dnum: thuộc tính kết

(xem thêm ví dụ về NATURAL JOIN trong phần về SQL)

NATURAL JOIN – Ví dụ

Dòng kết quả chỉ có một giá trị của thuộc tính kết được giữ lại.

PROJ_DEPT

Pname	Pnumber	Plocation	Dnum	Dname	Mgr_ssn	Mgr_start_date
ProductX	1	Bellaire	5	Research	333445555	1988-05-22
ProductY	2	Sugarland	5	Research	333445555	1988-05-22
ProductZ	3	Houston	5	Research	333445555	1988-05-22
Computerization	10	Stafford	4	Administration	987654321	1995-01-01
Reorganization	20	Houston	1	Headquarters	888665555	1981-06-19
Newbenefits	30	Stafford	4	Administration	987654321	1995-01-01

Lực lượng của kết quả phép kết JOIN

Nếu quan hệ R có n_R bộ và S có n_S bộ, phép kết giữa hai bộ:

$$R \bowtie_{\langle \text{join condition} \rangle} S$$

cho kết quả có lực lượng trong khoảng $(0, n_R * n_S)$ bộ.

Định nghĩa: tỉ lệ giữa kích thước mong đợi của kết quả phép kết chia cho kích thước tối đa $n_R * n_S$ thuộc tính được gọi là **tính chọn lọc của phép kết (join selectivity)**, là một thuộc tính của mỗi điều kiện kết.

- Nếu không có điều kiện kết, mọi tổ hợp của các bộ đều đủ điều kiện. Khi đó phép JOIN suy biến thành tích chập DECARTES, và còn được gọi là tích chéo (CROSS PRODUCT) hay phép kết chéo (CROSS JOIN)

Phép kết nội (inner join) và phép kết n-chieu

- Là phép kết JOIN kết hợp dữ liệu từ hai quan hệ và tạo ra thông tin được trình bày trong một bảng duy nhất.
- Cách định nghĩa khác: một phép kết nội (inner join) là phép so khớp-kết hợp được định nghĩa một cách hình thức là sự kết hợp giữa tích chập Decartes với phép chọn SELECTION.
- Phép NATURAL JOIN (EQUIJOIN) có thể áp dụng cho nhiều bảng → kết n-chieu (n-way join)

Ví dụ:

$((\text{PROJECT} \bowtie_{Dnum=Dnumber} \text{DEPARTMENT}) \bowtie_{Mgr_ssn=ssn} \text{EMPLOYEE})$

Tập đầy đủ các phép đại số quan hệ

Tập các phép toán đại số quan hệ $\{\sigma, \pi, U, \rho, -, \times\}$ là một tập đầy đủ; nghĩa là, bất kỳ một phép toán đại số quan hệ nào khác cũng đều có thể được diễn giải thông qua một chuỗi các phép toán trong tập nói trên.

Tập đầy đủ các phép đại số quan hệ

Ví dụ 1: phép giao (INTERSECTION) được viết bằng phép UNION và MINUS như sau:

$$R \cap S \equiv (R \cup S) - ((R - S) \cup (S - R))$$

Ví dụ 2: phép JOIN tương đương với tích chập Decartes và phép SELECT như sau:

$$R \bowtie_{\langle condition \rangle} S \equiv \sigma_{\langle condition \rangle}(R \times S)$$

Ví dụ 3: phép NATURAL JOIN tương đương với phép đổi tên RENAME, tích chập Decartes và phép SELECT và PROJECT.

Phép chia – DIVISION

Phép chia (DIVISION), ký hiệu \div , áp dụng vào hai quan hệ $R(Z) \div S(X)$, trong đó các thuộc tính của S là tập con của các thuộc tính của R : $X \subseteq Z$.

Gọi Y là tập các thuộc tính của R và không là thuộc tính của S : $Y = Z - X$ (và do đó $Z = X \cup Y$).

Phép chia – DIVISION

Định nghĩa:

Kết quả của phép DIVISION giữa hai quan hệ $R(Z) \div S(X)$ là một quan hệ $T(Y)$ bao gồm một bộ t nếu các bộ t_R xuất hiện trong R với $t_R[Y] = t$, và với $t_R[X] = t_S$ cho mọi bộ t_S trong S .

Điễn giải: Để một bộ t xuất hiện trong kết quả T , các giá trị trong t phải xuất hiện trong R trong tổ hợp với mọi bộ trong S .

- Trong công thức của phép chia, các bộ trong quan hệ mẫu số S hạn chế quan hệ tử số R bằng cách chọn những bộ trong kết quả khớp với tất cả các giá trị có trong mẫu số.
- Không nhất thiết phải biết các giá trị này là gì vì chúng có thể được tính trong một phép tính khác.

DIVISION – Ví dụ

“Hãy lấy tên các nhân viên làm việc trong **tất cả** các dự án mà ‘John Smith’ đang tham gia.”

B1. Lấy danh sách các mã dự án mà ‘John Smith’ đang làm, lưu vào quan hệ trung gian SMITH_PNOS:

$$\begin{aligned} \text{SMITH} &\leftarrow \sigma_{Fname='John' \text{ AND } Lname='Smith'}(\text{EMPLOYEE}) \\ \text{SMITH_PNOS} &\leftarrow \pi_{Pno}(\text{WORK_ON} \bowtie_{Essn=Ssn} \text{SMITH}) \end{aligned}$$

B2. Tạo một quan hệ có chứa một bộ $\langle Pno, Essn \rangle$ bất cứ khi nào mà nhân viên có Ssn là Essn có làm việc trong dự án có mã dự án là Pno, lưu kết quả vào quan hệ trung gian SSN_PNOS:

$$\text{SSN_PNOS} \leftarrow \pi_{Essn, Pno}(\text{WORKS_ON})$$

B3. Áp dụng phép DIVISION vào hai quan hệ trung gian để có kết quả:

$$\begin{aligned} \text{SSNS}(Ssn) &\leftarrow \text{SSN_PNOS} \div \text{SMITH_PNOS} \\ \text{RESULT} &\leftarrow \pi_{Fname, Lname}(\text{SSNS} * \text{EMPLOYEE}) \end{aligned}$$

DIVISION – Ví dụ

SSN_PNOS

Essn	Pno
123456789	1
123456789	2
666884444	3
453453453	1
453453453	2
333445555	2
333445555	3
333445555	10
333445555	20
999887777	30
999887777	10
987987987	10
987987987	30
987654321	30
987654321	20
888665555	20

SMITH_PNOS

Pno
1
2

SSNS

Ssn
123456789
453453453

DIVISION – Ví dụ

Cho hai quan hệ R và S và các thuộc tính A, B .
Các tập thuộc tính sau: $X = \{A\}, Y = \{B\}$, và $Z = \{A, B\}$.

Tính T ?

R

A	B
a1	b1
a2	b1
a3	b1
a4	b1
a1	b2
a3	b2
a2	b3
a3	b3
a4	b3
a1	b4
a2	b4
a3	b4

S

A
a1
a2
a3

T

B
b1
b4

Biểu diễn DIVISION bằng các phép toán cơ sở

Phép DIVISION có thể được thực hiện như sau:

$$T1 \leftarrow \pi_Y(R)$$

$$T2 \leftarrow \pi_Y((S \times T1) - R)$$

$$T \leftarrow T1 - T2$$

DIVISION trong SQL

- Phép chia được định nghĩa để thuận tiện cho việc xử lý các truy vấn liên quan đến lượng hóa tổng quát hoặc điều kiện *tất cả*.
- Hầu hết RDBMS không cài đặt phép chia.

Bảng tổng kết các phép toán đại số quan hệ (1/2)

PHÉP TOÁN	MỤC ĐÍCH	KÝ HIỆU
SELECT	Chọn từ trong quan hệ R tất cả các bộ thỏa mãn điều kiện chọn.	$\sigma_{<selection\ condition>}(R)$
PROJECT	Tạo một quan hệ mới với một số thuộc tính của R , và bỏ đi các bộ bị trùng lặp.	$\pi_{<attribute\ list>}(R)$
THETA JOIN	Tạo ra tất cả các tổ hợp của các bộ từ R_1 và R_2 thỏa mạn điều kiện kết.	$R_1 \bowtie_{<join\ condition>} R_2$
EQUIJOIN	Tạo ra tất cả các tổ hợp của các bộ từ R_1 và R_2 thỏa mạn điều kiện kết chỉ có các phép so sánh bằng.	$R_1 \bowtie_{<join\ condition>} R_2$, hoặc $R_1 \bowtie_{(<join\ attribute\ 1>),(<join\ attribute\ 2>)} R_2$
NATURAL JOIN	Tương tự EQUIJOIN, ngoại trừ các thuộc tính kết của R_2 không được bao gồm trong quan hệ kết quả; nếu các thuộc tính kết có cùng tên thì không phải chỉ định chúng trong khai báo phép kết.	$R_1 *_{<join\ condition>} R_2$, hoặc $R_1 *_{(<join\ attributes\ 1>),(<join\ attribute\ 2>)} R_2$, hoặc $R_1 * R_2$

Bảng tổng kết các phép toán đại số quan hệ (2/2)

PHÉP TOÁN	MỤC ĐÍCH	KÝ HIỆU
UNION	Tạo ra một quan hệ bao gồm tất cả các bộ trong R_1 hoặc R_2 hoặc trong cả R_1 và R_2 ; R_1 và R_2 phải tương thích hợp nhất.	$R_1 \cup R_2$
INTERSECTION	Tạo ra một quan hệ bao gồm tất cả các bộ trong cả R_1 và R_2 ; R_1 và R_2 phải tương thích hợp nhất.	$R_1 \cap R_2$
DIFFERENCE	Tạo ra một quan hệ bao gồm tất cả các bộ trong R_1 nhưng không có trong R_2 ; R_1 và R_2 phải tương thích hợp nhất.	$R_1 - R_2$
CARTESIAN PRODUCT	Tạo ra một quan hệ có các thuộc tính của R_1 và R_2 và bao gồm mọi tổ hợp của các bộ từ R_1 và R_2 .	$R_1 \times R_2$
DIVISION	Tạo ra một quan hệ $R(X)$ bao gồm tất cả các bộ $t[X]$ trong $R_1(Z)$ có xuất hiện trong R_1 trong tổ hợp với mọi bộ từ $R_2(Y)$, trong đó $Z = XUY$.	$R_1(Z) \div R_2(Y)$

Ký hiệu cho các cây Truy vấn

Cây truy vấn (*query tree*) là ký hiệu dùng để biểu diễn nội bộ các truy vấn trong các RDBMS (Relational DBMS – hệ quản trị CSDL quan hệ)

Còn gọi là cây đánh giá truy vấn (*query evaluation tree*) hoặc là cây thực hiện truy vấn (*query execution tree*).

Cây Truy vấn – Định nghĩa

Định nghĩa:

Cây truy vấn là một cấu trúc dữ liệu dạng cây tương đương với một biểu thức đại số quan hệ.

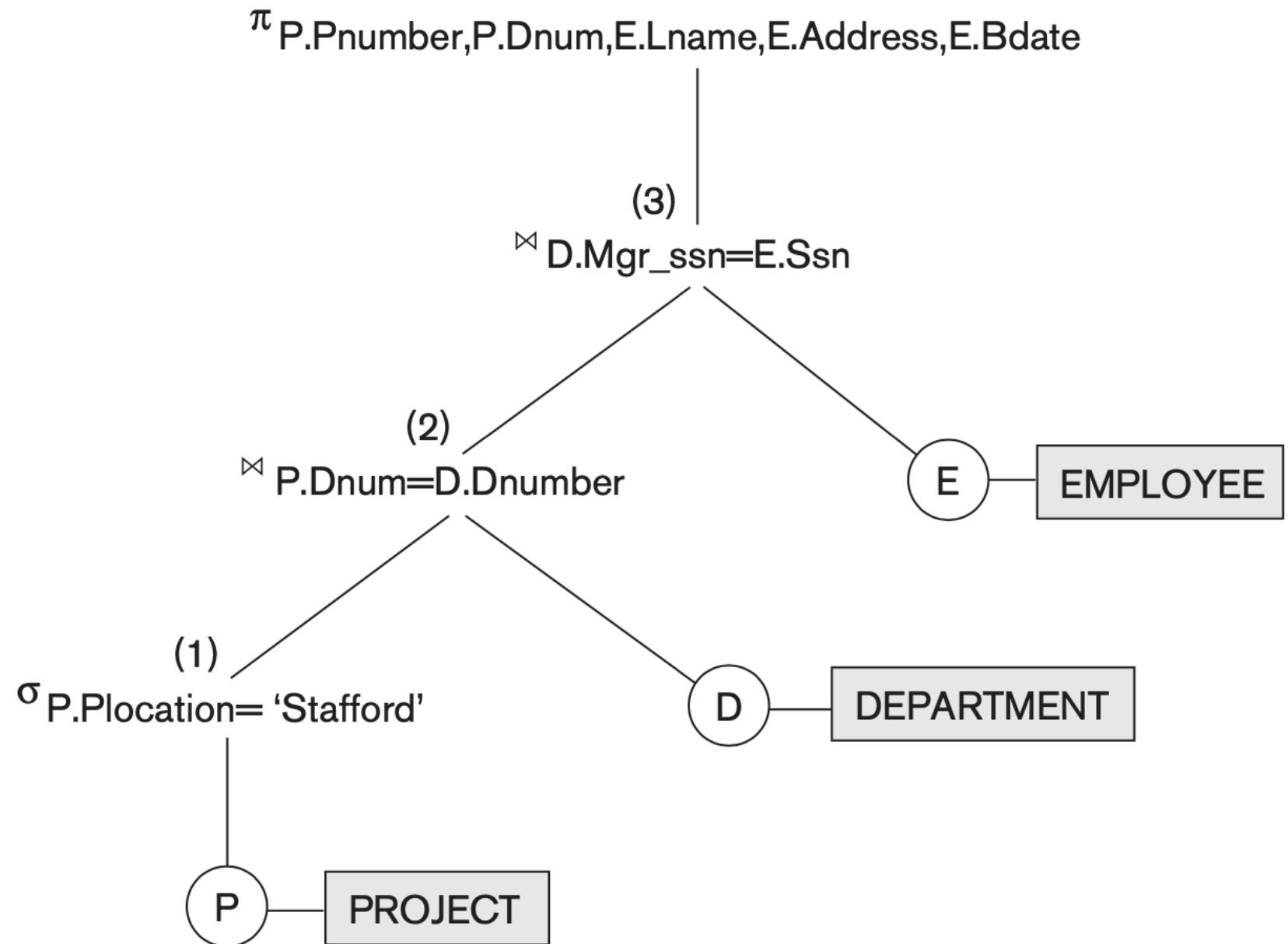
- Các nút lá biểu diễn các quan hệ đầu vào,
 - Các nút trong biểu diễn các toán tử đại số quan hệ.
-
- Một thực thi của cây truy vấn bao gồm việc thực hiện một phép toán tại nút trong bất kỳ khi nào các toán hạng của nó (biểu diễn bằng các nút con) sẵn sàng, và sau đó thay thế nút này bằng một quan hệ là kết quả của việc thực hiện phép toán đó.
 - Thực thi truy vấn kết thúc khi nút gốc được tính và cho ra kết quả của phép truy vấn.

Cây truy vấn – Ví dụ

“Với mọi dự án triển khai tại ‘Stafford’, hãy liệt kê mã số dự án, mã số phòng ban quản lý, và họ, địa chỉ, ngày sinh của người quản lý phòng ban này.”

$$\pi_{Pnumber, Dnum, Lname, Address, Bdate}(((\sigma_{Plocation='Stafford'}(PROJECT)) \bowtie_{Dnum=Dnumber}(DEPARTMENT) \bowtie_{Mgr_ssn=Ssn}(EMPLOYEE)))$$

Cây truy vấn – Ví dụ



Các phép tính quan hệ khác

Phép chiếu tổng quát

Phép chiếu tổng quát mở rộng phép chiếu bằng cách cho phép các hàm của thuộc tính được đưa vào danh sách chiếu.

Dạng tổng quát:

$$\pi_{F_1, F_2, \dots, F_n}(R)$$

trong đó, F_1, F_2, \dots, F_n là các hàm trên các thuộc tính trong quan hệ R , có thể bao gồm các phép toán số học và các giá trị hằng.

Ứng dụng: tạo các báo cáo có cột chứa các giá trị được tính toán sẵn.

Phép chiếu tổng quát – Ví dụ

Xét quan hệ: EMPLOYEE (Ssn, Salary, Deduction, Years_service)

Tạo báo cáo tính lương, thưởng, thuế được tính như sau:

$$\text{Net Salary} = \text{Salary} - \text{Deduction},$$

$$\text{Bonus} = 2000 * \text{Years_service}, \text{ai}$$

$$\text{Tax} = 0.25 * \text{Salary}$$

Truy vấn tạo báo cáo sẽ là:

```
REPORT ← ρ(Ssn,Net_salary,Bonus,Tax)(πSsn,Salary-Deduction, (EMPLOYEE))  
2000*Years service,  
0.25*Salary
```

Hàm tổng hợp và Gom nhóm

Gom nhóm theo giá trị một số thuộc tính, sau đó áp dụng hàm tổng hợp cho *từng nhóm riêng*.

Định nghĩa một phép toán hàm tổng hợp, ký hiệu \mathcal{I} , để chỉ định các kiểu yêu cầu như sau:

$$<\text{grouping attributes}>^{\mathcal{I}}<\text{function list}>(R)$$

trong đó,

- <grouping attributes>: danh sách các thuộc tính của quan hệ R
- <function list>: danh sách các cặp (<function> <attribute>).

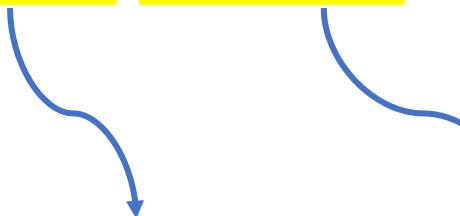
Trong mỗi cặp, <function> là một hàm áp dụng được, ví dụ SUM, AVERAGE, MAXIMUM, MINIMUM, COUNT; <attribute> là một thuộc tính trong quan hệ R .

Quan hệ kết quả có các thuộc tính gom nhóm và thêm một thuộc tính cho mỗi phần tử trong danh sách hàm.

Hàm tổng hợp và Gom nhóm – Ví dụ 1

“Hãy lấy ra mã số phòng ban, số nhân viên trong phòng ban đó, lương trung bình của họ, đồng thời đổi tên các thuộc tính kết quả.”

$$\rho_R(Dno, \text{No_of_employee}, \text{Average_sal})(\sigma_{Dno}^{\exists} \text{COUNT Ssn}, \text{AVERAGE Salary}(\text{EMPLOYEE}))$$



Dno	No_of_employees	Average_sal
5	4	33250
4	3	31000
1	1	55000

Hàm tổng hợp và Gom nhóm – Ví dụ 2

“Hãy lấy ra mã số phòng ban, số nhân viên trong phòng ban đó, lương trung bình của họ, đồng thời đổi tên các thuộc tính kết quả.”

Nếu không đổi tên thuộc tính, thì thuộc tính của kết quả sẽ được ghép từ tên hàm với tên thuộc tính là đối số của hàm.

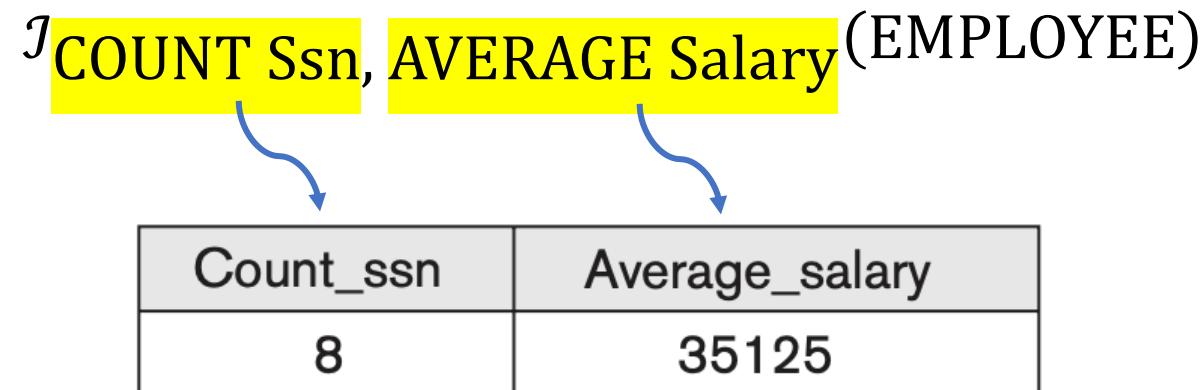
$Dno \rightarrow$ COUNT Ssn, AVERAGE Salary (EMPLOYEE)

Dno	Count_ssn	Average_salary
5	4	33250
4	3	31000
1	1	55000

Hàm tổng hợp và Gom nhóm – Ví dụ 3

“Hãy lấy ra mã số phòng ban, số nhân viên trong phòng ban đó, lương trung bình của họ, đồng thời đổi tên các thuộc tính kết quả.”

Nếu không chỉ định thuộc tính gom nhóm, thì các hàm tổng hợp được áp dụng cho *tất cả các bộ dữ liệu* trong quan hệ, và quan hệ kết quả *chỉ có một bộ duy nhất*.



Phép đóng đệ quy

Áp dụng trong trường hợp có mối liên kết đệ quy giữa các bộ cùng kiểu.

Ví dụ: liên kết giữa một nhân viên và người giám sát.

Yêu cầu:

"Truy xuất ra tất cả những người được giám sát bởi nhân viên e ở tất cả các cấp. Nghĩa là, mọi nhân viên e' được giám sát trực tiếp bởi e, mọi nhân viên e'' được giám sát trực tiếp bởi e', mọi nhân viên e''' được giám sát trực tiếp bởi e'', và tiếp tục ở cấp thấp hơn."

Phép đóng đệ quy – Ví dụ (1/2)

Giám sát cấp 1:

“Truy xuất ra tất cả những người e' được giám sát trực tiếp bởi nhân viên e có tên ‘James Borg’.”

$$\begin{aligned} \text{BORG_SSN} &\leftarrow \pi_{\text{Ssn}}(\sigma_{\text{Fname}=\text{'James'} \text{ AND } \text{Lname}=\text{'Borg'}}(\text{EMPLOYEE})) \\ \text{SUPERVISION}(\text{Ssn1}, \text{Ssn2}) &\leftarrow \pi_{\text{Ssn}, \text{Super_ssn}}(\text{EMPLOYEE}) \\ \text{RESULT1}(\text{Ssn}) &\leftarrow \pi_{\text{Ssn1}}(\text{SUPERVISION} \bowtie_{\text{Ssn2}=\text{Ssn}} \text{BORG_SSN}) \end{aligned}$$

Giám sát cấp 2:

“mọi nhân viên e'” được giám sát trực tiếp bởi các nhân viên e' ở trong truy vấn trên.”

$$\text{RESULT2}(\text{Ssn}) \leftarrow \pi_{\text{Ssn1}}(\text{SUPERVISION} \bowtie_{\text{Ssn2}=\text{Ssn}} \text{RESULT1})$$

Kết quả:

$$\text{RESULT} \leftarrow \text{RESULT2} \cup \text{RESULT1}$$

Phép đóng đệ quy – Ví dụ (2/2): kết quả

Lời giải đầy đủ yêu cầu phải xác định số lần đệ quy (số mức truy vấn liên kết giám sát).

Phép toán *đóng bắc cầu (transitive closure)* được dùng để tính toán mối quan hệ đệ quy cho trong suốt quá trình thực thi đệ quy.

SUPERVISION

(Borg's Ssn is 888665555)
(Ssn) (Super_ssn)

Ssn1	Ssn2
123456789	333445555
333445555	888665555
999887777	987654321
987654321	888665555
666884444	333445555
453453453	333445555
987987987	987654321
888665555	null

RESULT1

Ssn
333445555
987654321

(Supervised by Borg)

RESULT2

Ssn
123456789
999887777
666884444
453453453
987987987

(Supervised by
Borg's subordinates)

RESULT

Ssn
123456789
999887777
666884444
453453453
987987987
333445555
987654321

(RESULT1 \cup RESULT2)

Phép kết ngoại – OUTER JOIN

Tập các phép kết ngoại được sử dụng khi muốn giữ lại tất cả các bộ trong R , hoặc tất cả các bộ trong S , hoặc tất cả các bộ trong hai quan hệ trong kết quả của phép JOIN mà không quan tâm đến việc hai quan hệ này có các bộ khớp với nhau hay không.

Phép kết ngoại trái – LEFT OUTER JOIN

Phép kết ngoại trái, ký hiệu \bowtie , giữa R và S :

$$R \bowtie S;$$

sẽ giữ lại tất cả các bộ trong quan hệ R , ở bên *trái*, trong quan hệ kết quả; nếu không có bộ nào khớp được tìm thấy trong S thì các thuộc tính của S trong kết quả kết được điền bằng NULL.

LEFT OUTER JOIN – Ví dụ

“Lấy danh sách tên của tất cả nhân viên cùng tên của phòng ban mà họ quản lý, nếu nhân viên được cử làm quản lý của một phòng ban.

Trường hợp nhân viên không là quản lý phòng ban nào cả, ghi giá trị NULL.”

$\text{TEMP} \leftarrow (\text{EMPLOYEE} \bowtie_{\text{Ssn}=\text{Mgr_ssn}} \text{DEPARTMENT})$
 $\text{RESULT} \leftarrow \pi_{\text{Fname}, \text{Minit}, \text{Lname}, \text{Dname}}(\text{TEMP})$

Fname	Minit	Lname	Dname
John	B	Smith	NULL
Franklin	T	Wong	Research
Alicia	J	Zelaya	NULL
Jennifer	S	Wallace	Administration
Ramesh	K	Narayan	NULL
Joyce	A	English	NULL
Ahmad	V	Jabbar	NULL
James	E	Borg	Headquarters

Phép kết ngoại phải – RIGHT OUTER JOIN

Phép kết ngoại phải, ký hiệu \bowtie , giữa R và S :

$$R \bowtie S$$

sẽ giữ lại tất cả các bộ trong quan hệ S , ở bên *phải*, trong quan hệ kết quả; nếu không có bộ nào khớp được tìm thấy trong R thì các thuộc tính của R trong kết quả kết được điền bằng NULL.

Phép kết ngoại đầy đủ – FULL OUTER JOIN

Phép kết ngoại đầy đủ, ký hiệu \bowtie , giữa R và S :

$$R \bowtie S$$

sẽ giữ lại tất cả các bộ trong cả R ở bên *trái* và S ở bên *phải* trong quan hệ kết quả nếu không có bộ nào khớp được tìm thấy, và điền các thuộc tính tương ứng bằng NULL nếu cần thiết.

Phép hợp ngoại – OUTER UNION

Định nghĩa: Phép hợp ngoại lấy phần hợp các bộ từ hai quan hệ có cùng một số thuộc tính chung, nhưng không phải là tương thích hợp nhất.

- Hai quan hệ $R(X, Y)$ và $S(X, Z)$ có tính chất trên được gọi là tương thích một phần, nghĩa là chỉ một số thuộc tính của chúng (X) là tương thích hợp nhất.
- Các thuộc tính tương thích hợp nhất được trình bày duy nhất một lần trong quan hệ kết quả.
Các thuộc tính không tương thích hợp nhất được giữ lại trong quan hệ kết quả.
 $\Rightarrow T(X, Y, Z)$
- Tương đương phép kết ngoại đầy đủ trên các thuộc tính chung.

Phép hợp ngoại – OUTER UNION – Ví dụ

Áp dụng phép hợp ngoại cho hai quan hệ:

STUDENT(Name, Ssn, Department, Advisor)

INSTRUCTOR(Name, Ssn, Department, Rank)

Quan hệ kết quả:

STUDENT_OR_INSTRUCTOR(Name, Ssn, Department, Advisor, Rank)

Câu hỏi: Hãy thử diễn giải ý nghĩa của quan hệ kết quả?

Các ví dụ mẫu

Bài tập ví dụ 1

“Truy xuất tên và địa chỉ của tất cả nhân viên làm trong phòng ‘Research’.”

Đáp án:

```
RESEARCH_DEPT ← σDname='Research'(DEPARTMENT)
RESEARCH_EMPS ← (RESEARCH_DEPT ⋈Dnumber=Dno EMPLOYEE)
RESULT ← πFname, Lname, Address(RESEARCH_EMPS)
```

hoặc:

```
πFname, Lname, Address (σDname='Research'(DEPARTMENT ⋈Dnumber=Dno(EMPLOYEE)))
```

Bài tập ví dụ 2

“Với mọi dự án triển khai tại ‘Stafford’, hãy liệt kê danh sách mã số dự án, mã phòng ban kiểm soát dự án, và họ, địa chỉ, ngày sinh của người quản lý phòng ban này.”

Đáp án:

```
STAFFORD_PROJS ←  $\sigma_{\text{Plocation}=\text{'Stafford'}}(\text{PROJECT})$ 
CONTR_DEPTS ← (STAFFORD_PROJS  $\bowtie_{\text{Dnum}=\text{Dnumber}}$  DEPARTMENT)
PROJ_DEPT_MGRS ← (CONTR_DEPTS  $\bowtie_{\text{Mgr_ssn}=\text{SsnE}}$  MPLOYEE)
RESULT ←  $\pi_{\text{Pnumber}, \text{Dnum}, \text{Lname}, \text{Address}, \text{Bdate}}(\text{PROJ_DEPT_MGRS})$ 
```

Bài tập ví dụ 3

“Tìm tên của các nhân viên làm việc trong tất cả các dự án do phòng ban số 5 kiểm soát.”

Đáp án:

```
DEPT5_PROJS ← ρ(Pno)(πPnumber(σDnum=5(PROJECT)))
EMP_PROJ ← ρ(Ssn, Pno)(πEssn, Pno(WORKS_ON))
RESULT_EMP_SSNS ← EMP_PROJ ÷ DEPT5_PROJS
RESULT ← πLname, Fname(RESULT_EMP_SSNS * EMPLOYEE)
```

Bài tập ví dụ 4

“Lập danh sách các mã số của các dự án có sự tham gia của nhân viên tên ‘Smith’ với vai trò là công nhân hoặc là người quản lý của phòng ban đang kiểm soát dự án.”

Đáp án:

$$\begin{aligned} \text{SMITHS}(\text{Essn}) &\leftarrow \pi_{\text{Ssn}}(\sigma_{\text{Lname}=\text{'Smith'}}(\text{EMPLOYEE})) \\ \text{SMITH_WORKER_PROJS} &\leftarrow \pi_{\text{Pno}}(\text{WORKS_ON} * \text{SMITHS}) \\ \text{MGRS} &\leftarrow \pi_{\text{Lname}, \text{Dnumber}}(\text{EMPLOYEE} \bowtie_{\text{Ssn}=\text{Mgr_ssn}} \text{DEPARTMENT}) \\ \text{SMITH_MANAGED_DEPTS}(\text{Dnum}) &\leftarrow \pi_{\text{Dnumber}}(\sigma_{\text{Lname}=\text{'Smith'}}(\text{MGRS})) \\ \text{SMITH_MGR_PROJS}(\text{Pno}) &\leftarrow \pi_{\text{Pnumber}}(\text{SMITH_MANAGED_DEPTS} * \text{PROJECT}) \\ \text{RESULT} &\leftarrow (\text{SMITH_WORKER_PROJS} \cup \text{SMITH_MGR_PROJS}) \end{aligned}$$

Bài tập ví dụ 4 – đáp án gọn hơn

“Lập danh sách các mã số của các dự án có sự tham gia của nhân viên tên ‘Smith’ với vai trò là công nhân hoặc là người quản lý của phòng ban đang kiểm soát dự án.”

Đáp án:

$$\begin{aligned} & \pi_{Pno} (\text{WORKS_ON} \bowtie_{E\text{ssn}=S\text{sn}} (\pi_{Ssn} (\sigma_{Lname='Smith'}(\text{EMPLOYEE}))) \cup \pi_{Pno} \\ & ((\pi_{Dnumber} (\sigma_{Lname='Smith'}(\pi_{Lname, Dnumber}(\text{EMPLOYEE}))) \bowtie_{Ssn=Mgr_ssn} \text{DEPARTMENT})) \bowtie_{Dnum-ber=Dnum} \text{PROJECT} \end{aligned}$$

Bài tập ví dụ 5

“Lập danh sách tất cả nhân viên có từ hai người phụ thuộc trở lên.”

Đáp án:

$$\begin{aligned} T1(\text{Ssn}, \text{No_of_dependents}) &\leftarrow \text{Essn } \exists \text{ COUNT Dependent_name(DEPENDENT)} \\ T2 &\leftarrow \sigma_{\text{No_of_dependents} > 2}(T1) \\ \text{RESULT} &\leftarrow \pi_{\text{Lname, Fname}}(T2 * \text{EMPLOYEE}) \end{aligned}$$

Bài tập ví dụ 6

“Lập danh sách các nhân viên không có người phụ thuộc.”

Đáp án:

```
ALL_EMPS ← πSsn(EMPLOYEE)
EMPS_WITH_DEPS(Ssn) ← πEssn(DEPENDENT)
EMPS_WITHOUT_DEPS ← (ALL_EMPS – EMPS_WITH_DEPS)
RESULT ← πLname, Fname(EMPS_WITHOUT_DEPS * EMPLOYEE)
```

hoặc:

$$\pi_{Lname, Fname}((\pi_{Ssn}(EMPLOYEE) - \rho_{Ssn}(\pi_{Essn}(DEPENDENT))) * EMPLOYEE)$$

Bài tập ví dụ 7

“Lập danh sách tên các người quản lý có ít nhất một người phụ thuộc.”

Đáp án:

$$\begin{aligned} \text{MGRS}(\text{Ssn}) &\leftarrow \pi_{\text{Mgr_ssn}}(\text{DEPARTMENT}) \\ \text{EMPS_WITH_DEPS}(\text{Ssn}) &\leftarrow \pi_{\text{Essn}}(\text{DEPENDENT}) \\ \text{MGRS_WITH_DEPS} &\leftarrow (\text{MGRS} \cap \text{EMPS_WITH_DEPS}) \\ \text{RESULT} &\leftarrow \pi_{\text{Lname, Fname}}(\text{MGRS_WITH_DEPS} * \text{EMPLOYEE}) \end{aligned}$$

Phép tính Quan hệ (Relational Calculus)

Một ngôn ngữ truy vấn hình thức khác cho mô hình quan hệ.

Giới thiệu chung

- Biểu thức tính toán xác định đối tượng cần truy xuất, không phải cách thức truy xuất đối tượng.
→ Phép tính quan hệ là ngôn ngữ mô tả hình thức phi thủ tục.
- Định nghĩa: Ngôn ngữ truy vấn L được coi là hoàn chỉnh về mặt quan hệ (*relationally complete*) nếu có thể biểu diễn trong L bất kỳ câu truy vấn nào có thể được biểu thị bằng phép tính quan hệ.

Các biến bộ và Các quan hệ miền giá trị

Dạng thức câu truy vấn tính toán quan hệ trên bộ:

$$\{t \mid \text{COND}(t)\}$$

trong đó,

- t : biến bộ (*tuple variable*)
- $\text{COND}(t)$: biểu thức điều kiện có liên quan đến t , trả về TRUE hoặc FALSE cho các phép gán các bộ khác nhau cho biến t .

Kết quả của câu truy vấn là một tập hợp tất cả các bộ t sao cho $\text{COND}(t)$ là TRUE. Các bộ này được gọi là thỏa mãn điều kiện $\text{COND}(t)$.

Ví dụ định nghĩa quan hệ miền giá trị

“Tìm tất cả nhân viên có mức lương trên \$50,000.”

Đáp án:

$$\{t \mid \text{EMPLOYEE}(t) \text{ AND } t.\text{Salary} > 50000\}$$

Nếu quy định thuộc tính cần truy xuất, ví dụ chỉ lấy họ và tên nhân viên:

$$\{t.\text{Fname}, t.\text{Lname} \mid \text{EMPLOYEE}(t) \text{ AND } t.\text{Salary} > 50000\}$$

- EMPLOYEE: quan hệ miền giá trị (**range relation**) trong đó biến bộ t được kiểm tra điều kiện truy xuất.

Các biểu thức trong phép tính quan hệ bộ

Biểu thức tổng quát có dạng:

$$\{t_1 \cdot A_j, t_2 \cdot A_k, \dots, t_n \cdot A_m \mid \text{COND}(t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m})\}$$

trong đó,

- $t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m}$: các biến bộ,
- Mỗi A_i là một thuộc tính của quan hệ xác định miền giá trị của t_i ,
- COND: một điều kiện hoặc công thức trong phép tính quan hệ bộ.

Công thức trong phép tính quan hệ bộ

Một công thức được tạo thành từ các nguyên tử phép tính vị từ có các dạng sau:

1. Nguyên tử dạng $R(t_i)$, trong đó R là tên quan hệ và t_i là một biến bộ.
 - Dùng để xác định miền giá trị của t_i là R .
 - TRUE: nếu $t_i \in R$, FALSE: nếu ngược lại.
2. Nguyên tử dạng $t_i. A \text{ op } t_j. B$,
trong đó **op** là phép toán so sánh trong tập $\{=, <, \leq, >, \geq, \neq\}$, t_i, t_j là các biến bộ, A là một thuộc tính của quan hệ xác định miền giá trị của t_i và B là một thuộc tính của quan hệ xác định miền giá trị của t_j .
3. Nguyên tử dạng $t_i. A \text{ op } c$ hoặc $c \text{ op } t_j. B$,
trong đó **op** là phép toán so sánh trong tập $\{=, <, \leq, >, \geq, \neq\}$, t_i, t_j là các biến bộ, A là một thuộc tính của quan hệ xác định miền giá trị của t_i và B là một thuộc tính của quan hệ xác định miền giá trị của t_j , và c là một giá trị hằng.

Công thức và các toán tử logic, Định nghĩa đệ quy công thức

Một công thức được tạo thành từ các nguyên tử nối nhau bằng các toán tử logic **AND**, **OR**, **NOT** và được định nghĩa một cách đệ quy bằng luật 1 và luật 2 như sau:

- **Luật 1** : Mọi nguyên tử là một công thức.
- **Luật 2** : Nếu F_1, F_2 là các công thức, thì
 $(F_1 \text{ AND } F_2), (F_1 \text{ OR } F_2), \text{NOT}(F_1), \text{NOT}(F_2)$ cũng là các công thức.

Bảng chân trị của các công thức logic

Giá trị chân trị (truth value) của các công thức như sau:

- a. $(F_1 \text{ AND } F_2)$ là TRUE nếu cả F_1 và F_2 là TRUE; ngược lại, kết quả là FALSE.
- b. $(F_1 \text{ OR } F_2)$ là FALSE nếu cả F_1 và F_2 là FALSE; ngược lại, kết quả là TRUE.
- c. $\text{NOT}(F_1)$ là TRUE nếu F_1 là FALSE; là FALSE nếu F_1 là TRUE.
- d. $\text{NOT}(F_2)$ là TRUE nếu F_2 là FALSE; là FALSE nếu F_2 là TRUE.

Biến bộ tự do và biến bộ bị ràng buộc trong một công thức (1/3)

- Lượng từ tồn tại (\exists) và lượng từ tất cả (\forall)

Định nghĩa (không hình thức): Một biến bộ t là bị ràng buộc nếu nó được lượng hóa, nghĩa là nó xuất hiện trong mệnh đề $(\exists t)$ hoặc $(\forall t)$; ngược lại thì t là tự do.

Biến bộ tự do và biến bộ bị ràng buộc trong một công thức (2/3)

Định nghĩa: Trong một công thức, một biến bộ là tự do hoặc bị ràng buộc tuân theo các luật sau:

- Một xuất hiện của biến bộ trong công thức F mà là một nguyên tử thì biến này là tự do trong F .
- Một xuất hiện của biến bộ t là tự do hoặc bị ràng buộc trong công thức tạo thành từ các liên kết logic - ($F_1 \text{ AND } F_2$), ($F_1 \text{ OR } F_2$), $\text{NOT}(F_1)$, $\text{NOT}(F_2)$ – phụ thuộc vào việc nó có là tự do hay bị ràng buộc trong F_1 hay F_2 (nếu nó xuất hiện ở một trong hai công thức).

Chú ý: Trong công thức có dạng $F = (F_1 \text{ AND } F_2)$ hay $F = (F_1 \text{ OR } F_2)$, một biến bộ có thể là tự do trong F_1 và bị ràng buộc trong F_2 hoặc ngược lại; trong trường hợp này, một biến bộ sẽ là bị ràng buộc và biến còn lại là tự do trong F .

Biến bộ tự do và biến bộ bị ràng buộc trong một công thức (3/3)

- Tất cả các xuất hiện tự do của biến bộ t trong F là bị ràng buộc trong công thức F' có dạng $F' = (\exists t)(F)$ hoặc $F' = (\forall t)(F)$.

Biến bộ là bị ràng buộc trong lượng từ được xác định trong F' .

Ví dụ: Xem các công thức sau:

$$F_1: d.\text{Dname} = \text{'Research'}$$

$$F_2: (\exists t)(d.\text{Dnumber} = t.\text{Dno})$$

$$F_3: (\forall d)(d.\text{Mgr_ssn} = \text{'333445555'})$$

Biến d là tự do trong cả F_1 và F_2 , trong khi nó lại bị ràng buộc với lượng từ (\forall) trong F_3 . Biến t là bị ràng buộc với lượng từ (\exists) trong F_2 .

Định nghĩa công thức (tt.)

- **Luật 3 :** Nếu F là một công thức, thì $(\exists t)(F)$, trong đó t là biến bộ, cũng là một công thức.

Công thức $(\exists t)(F)$ là TRUE nếu F là TRUE với *một số bộ* (ít nhất là một) được gán cho các xuất hiện tự do của t trong F ; ngược lại, thì $(\exists t)(F)$ là FALSE.

- **Luật 4 :** Nếu F là một công thức, thì $(\forall t)(F)$, trong đó t là biến bộ, cũng là một công thức.

Công thức $(\forall t)(F)$ là TRUE nếu F là TRUE với *mọi bộ* được gán cho các xuất hiện tự do của t trong F ; ngược lại, thì $(\forall t)(F)$ là FALSE.

Bài tập ví dụ 1

Query 1. “*Liệt kê tên và địa chỉ của tất cả nhân viên làm trong phòng ‘Research’.*”

Q1:

$$\{t.Fname, t.Lname, t.Address \mid \text{EMPLOYEE}(t) \text{ AND } (\exists d)(\text{DEPARTMENT}(d) \text{ AND } d.Dname = 'Research' \text{ AND } d.Dnumber = t.Dno)\}$$

- Các biến bộ tự do nằm ở bên trái của \mid trong biểu thức toán quan hệ.

Biến tự do t trong **Q1** sẽ được liên kết lần lượt với mỗi bộ dữ liệu. Nếu bộ thỏa mãn các điều kiện phía phải của \mid , các thuộc tính Fname, Lname, Address được truy xuất cho mỗi bộ tương ứng.

- Các điều kiện $\text{EMPLOYEE}(t)$ và $\text{DEPARTMENT}(d)$ xác định các quan hệ miền giá trị cho t và d .
- Điều kiện $d.Dname = 'Research'$ là điều kiện chọn, tương đương phép SELECT,
- Điều kiện $d.Dnumber = t.Dno$ là điều kiện kết, tương đương phép INNER JOIN.

Bài tập ví dụ 2

Query 2. “Với mọi dự án triển khai tại ‘Stafford’, hãy liệt kê danh sách mã số dự án, mã phòng ban kiểm soát dự án, và họ, địa chỉ, ngày sinh của người quản lý phòng ban này.”

Q2:

$$\{p.Pnumber, p.Dnum, m.Lname, m.Bdate, m.Address \mid \text{PROJECT}(p) \text{ AND } \text{EMPLOYEE}(m) \text{ AND } p.Plocation = \text{'Stafford'} \text{ AND } ((\exists d)(\text{DEPARTMENT}(d) \text{ AND } p.Dnum = d.Dnumber \text{ AND } d.Mgr_ssn = m.Ssn))\}$$

Bài tập ví dụ 3'

Query 3'. “*Tìm tên của các nhân viên làm việc trong một vài dự án do phòng ban số 5 kiểm soát.*”

Q3':

$$\{e.\text{Lname}, e.\text{Fname} \mid \text{EMPLOYEE}(e) \text{ AND } ((\exists x)(\exists w)(\text{PROJECT}(x) \text{ AND } \text{WORKS_ON}(w) \text{ AND } x.\text{Dnum}=5 \text{ AND } w.\text{Essn}=e.\text{Ssn} \text{ AND } x.\text{Pnumber}=w.\text{Pno}))\}$$

Bài tập ví dụ 4

Query 4. “Lập danh sách các mã số của các dự án có sự tham gia của nhân viên tên ‘Smith’ với vai trò là công nhân hoặc là người quản lý của phòng ban đang kiểm soát dự án.”

Q4:

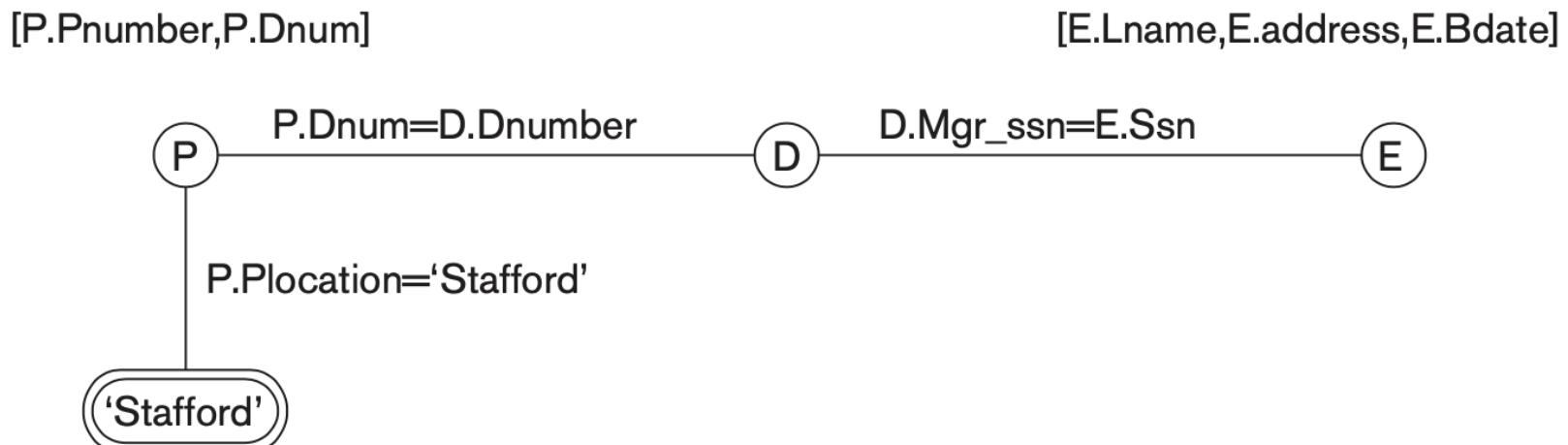
```
{ p.Pnumber | PROJECT(p) AND (((∃e)(∃w)(EMPLOYEE(e)  
AND WORKS_ON(w) AND w.Pno=p.Pnumber  
AND e.Lname='Smith' AND e.Ssn=w.Essn) )  
OR  
((∃m)(∃d)(EMPLOYEE(m) AND DEPARTMENT(d)  
AND p.Dnum=d.Dnumber AND d.Mgr_ssn=m.Ssn  
AND m.Lname='Smith')))}
```

Đồ thị truy vấn

Biểu diễn dạng đồ thị các truy vấn dạng **select-project-join**.

Ký hiệu:

-  : nút quan hệ
-  : nút hằng
- D.Mgr_ssn=E.Ssn : điều kiện chọn hoặc kết
- [P.Pnumber,P.Dnum] : thuộc tính cần truy xuất



Chuyển đổi lượng từ

- Các chuyển đổi lượng từ sau là tương đương:

$$(\forall x) (P(x)) \equiv \text{NOT} (\exists x) (\text{NOT} (P(x)))$$

$$(\exists x) (P(x)) \equiv \text{NOT} (\forall x) (\text{NOT} (P(x)))$$

$$(\forall x) (P(x) \text{ AND } Q(x)) \equiv \text{NOT} (\exists x) (\text{NOT} (P(x)) \text{ OR NOT } (Q(x)))$$

$$(\forall x) (P(x) \text{ OR } Q(x)) \equiv \text{NOT} (\exists x) (\text{NOT} (P(x)) \text{ AND NOT } (Q(x)))$$

$$(\exists x) (P(x) \text{ OR } Q(x)) \equiv \text{NOT} (\forall x) (\text{NOT} (P(x)) \text{ AND NOT } (Q(x)))$$

$$(\exists x) (P(x) \text{ AND } Q(x)) \equiv \text{NOT} (\forall x) (\text{NOT} (P(x)) \text{ OR NOT } (Q(x)))$$

- Các phép suy diễn sau là TRUE:

$$(\forall x)(P(x)) \Rightarrow (\exists x)(P(x))$$

$$\text{NOT} (\exists x)(P(x)) \Rightarrow \text{NOT} (\forall x)(P(x))$$

Sử dụng lượng từ trong câu truy vấn (1/4)

Query 3. “Tìm tên của các nhân viên làm việc trong tất cả các dự án do phòng ban số 5 kiểm soát.”

Cách giải 1

Q3: $\{e.\text{Lname}, e.\text{Fname} \mid \text{EMPLOYEE}(e) \text{ AND } ((\forall x)(\text{NOT}(\text{PROJECT}(x)) \text{ OR NOT } (x.\text{Dnum}=5) \text{ OR } ((\exists w)(\text{WORKS_ON}(w) \text{ AND } w.\text{Essn}=e.\text{Ssn} \text{ AND } x.\text{Pnumber}=w.\text{Pno}))))\}$

Tách từng bước,

Q3: $\{e.\text{Lname}, e.\text{Fname} \mid \text{EMPLOYEE}(e) \text{ AND } F'\}$

$F' = ((\forall x)(\text{NOT}(\text{PROJECT}(x)) \text{ OR } F_1))$

$F_1 = \text{NOT}(x.\text{Dnum}=5) \text{ OR } F_2$

$F_2 = ((\exists w)(\text{WORKS_ON}(w) \text{ AND } w.\text{Essn}=e.\text{Ssn} \text{ AND } x.\text{Pnumber}=w.\text{Pno}))$

Sử dụng lượng tử trong câu truy vấn (2/4)

Truy vấn Q3 có thể được viết lại với lượng tử tồn tại (\exists) như sau:

Q3A: $\{e.\text{Lname}, e.\text{Fname} \mid \text{EMPLOYEE}(e) \text{ AND } (\text{NOT } (\exists x) (\text{PROJECT}(x) \text{ AND } (x.\text{Dnum}=5) \text{ and } (\text{NOT } (\exists w) (\text{WORKS_ON}(w) \text{ AND } w.\text{Essn}=e.\text{Ssn} \text{ AND } x.\text{Pnumber}=w.\text{Pno}))))\}$

Câu hỏi: Hãy diễn giải câu truy vấn này.

Sử dụng lượng từ trong câu truy vấn (3/4)

Query 6. “Lên danh sách tên các nhân viên không có người phụ thuộc.”

Q6: { $e.Fname, e.Lname \mid EMPLOYEE(e) \text{ AND } (\text{NOT } (\exists d)(\text{DEPENDENT}(d) \text{ AND } e.Ssn=d.Essn))\}$ }

Truy vấn Q6 có thể được viết lại với lượng từ “với mọi” (\forall) như sau:

Q6A: { $e.Fname, e.Lname \mid EMPLOYEE(e) \text{ AND } ((\forall d)(\text{NOT}(\text{DEPENDENT}(d)) \text{ OR NOT}(e.Ssn=d.Essn)))\}$ }

Câu hỏi: Hãy diễn giải câu truy vấn này.

Sử dụng lượng tử trong câu truy vấn (4/4)

Query 7. “Lên danh sách tên các quản lý có ít nhất một người phụ thuộc.”

Q7: { $e.\text{Fname}, e.\text{Lname} \mid \text{EMPLOYEE}(e) \text{ AND } ((\exists d)(\exists p)(\text{DEPARTMENT}(d) \text{ AND } \text{DEPENDENT}(p) \text{ AND } e.\text{Ssn}=d.\text{Mgr_ssn} \text{ AND } p.\text{Essn}=e.\text{Ssn}))}$ }

Câu hỏi: Hãy diễn giải câu truy vấn này.

Biểu thức an toàn (1/2)

Định nghĩa:

Trong phép tính quan hệ , một biểu thức **an toàn** là biểu thức được đảm bảo sẽ cho kết quả là một tập hợp các bộ có số phần tử hữu hạn.

Ngược lại, biểu thức là **không an toàn**.

Ví dụ: biểu thức $\{t \mid \text{NOT}(\text{EMPLOYEE}(t))\}$ là không an toàn vì có kết quả gồm tất cả các bộ không phải là bộ EMPLOYEE \rightarrow tập hợp có vô số phần tử.

Miền của một biểu thức phép tính quan hệ trên bộ

Định nghĩa: Miền của một biểu thức phép tính quan hệ trên bộ là tập hợp tất cả các giá trị xuất hiện dưới dạng là các giá trị hằng trong biểu thức hoặc tồn tại trong bất kỳ bộ nào trong các quan hệ được tham chiếu trong biểu thức.

Ví dụ:

- Miền của biểu thức $\{t \mid \text{NOT}(\text{EMPLOYEE}(t))\}$ gồm tất cả các giá trị thuộc tính xuất hiện trong các bộ của EMPLOYEE.
- Miền của biểu thức Q3A gồm tất cả các giá trị xuất hiện trong EMPLOYEE, PROJECTS, WORKS_ON

Biểu thức an toàn (2/2)

Định nghĩa:

Một biểu thức được gọi là **an toàn** nếu mọi giá trị trong kết quả của biểu thức nằm trong miền giá trị của biểu thức đó.

Ví dụ: biểu thức $\{t \mid \text{NOT}(\text{EMPLOYEE}(t))\}$ là không an toàn vì có kết quả gồm các bộ (và theo đó, là các giá trị) nằm ngoài phạm vi của quan hệ EMPLOYEE.

Phép tính quan hệ trên miền giá trị

Khái niệm và Định nghĩa

- Biến trong biểu thức có phạm vi là các giá trị duy nhất trong các miền giá trị của các thuộc tính.
- Phải có n biến miền giá trị, mỗi biến tương ứng với một thuộc tính, để biểu diễn quan hệ cấp n trong kết quả truy vấn.

Biểu thức phép tính trên miền có dạng:

$$\{x_1, x_2, \dots, x_n \mid \text{COND}(x_1, x_2, \dots, x_n, x_{n+1}, x_{n+2}, \dots, x_{n+m})\}$$

trong đó,

- $x_1, x_2, \dots, x_n, x_{n+1}, x_{n+2}, \dots, x_{n+m}$ là các biến miền có phạm vi trên các miền giá trị của các thuộc tính,
- COND: điều kiện hay công thức của phép tính quan hệ miền.

Nguyên tử thành phần trong công thức

Một công thức được tạo thành từ các nguyên tử. Có các dạng nguyên tử sau:

1. Nguyên tử dạng $R(x_1, x_2, \dots, x_j)$, trong đó R là quan hệ cấp j và $x_i, 1 \leq i \leq j$ là một biến miền.

Nguyên tử dạng này cho biết một danh sách các giá trị $\langle x_1, x_2, \dots, x_j \rangle$ phải là một bộ trong quan hệ R , trong đó x_i là giá trị của giá trị thuộc tinh thứ i trong bộ.

TRUE nếu các biến miền được gán giá trị tương ứng với một bộ của R .

2. Nguyên tử dạng $x_i \text{ op } x_j$,
trong đó **op** là phép toán so sánh trong tập $\{=, <, \leq, >, \geq, \neq\}$, x_i, x_j là các biến miền.

TRUE nếu các biến miền được gán giá trị thỏa mãn điều kiện so sánh.

3. Nguyên tử dạng $x_i \text{ op } c$ hoặc $c \text{ op } x_j$,
trong đó **op** là phép toán so sánh trong tập $\{=, <, \leq, >, \geq, \neq\}$, x_i, x_j là các biến miền, và c là một giá trị hằng.

TRUE nếu các biến miền được gán giá trị thỏa mãn điều kiện so sánh.

Ví dụ (1/4)

Query 0. “*Liệt kê ngày sinh và địa chỉ của nhân viên có tên ‘John B. Smith’.*”

Q0: $\{u, v \mid (\exists q) (\exists r) (\exists s) (\exists t) (\exists w) (\exists x) (\exists y) (\exists z)$
 $(\text{EMPLOYEE}(qrstuvwxyz) \text{ AND } q=\text{'John'} \text{ AND } r=\text{'B'} \text{ AND } s=\text{'Smith'})\}$

Hoặc,

Q0A: $\{u, v \mid \text{EMPLOYEE}(\text{'John'}, \text{'B'}, \text{'Smith'}, t, u, v, w, x, y, z)\}$

Ví dụ (2/4)

Query 1. “Truy xuất tên và địa chỉ của tất cả nhân viên đang làm trong phòng ‘Research’.”

Q1: $\{q, s, \nu \mid (\exists z) (\exists l) (\exists m) (\text{EMPLOYEE}(qrstuvwxyz) \text{ AND } \text{DEPARTMENT}(lmno) \text{ AND } l=\text{'Research'} \text{ AND } m=z)\}$

Query 2. “Với mọi dự án triển khai tại ‘Stafford’, hãy liệt kê mã số dự án, mã số phòng ban quản lý dự án, và họ, địa chỉ, ngày sinh của người quản lý phòng ban này.”

Q2: $\{i, k, s, u, \nu \mid (\exists j)(\exists m)(\exists n)(\exists t)(\text{PROJECT}(hijk) \text{ AND } \text{EMPLOYEE}(qrstuvwxyz) \text{ AND } \text{DEPARTMENT}(lmno) \text{ AND } k=m \text{ AND } n=t \text{ AND } j=\text{'Stafford'})\}$

Ví dụ (3/4)

Query 6. “*Lập danh sách các nhân viên không có người phụ thuộc.*”

Q6: $\{q, s \mid (\exists t)(\text{EMPLOYEE}(qrstuvwxyz) \text{ AND } (\text{NOT}(\exists l)(\text{DEPENDENT}(lmnop) \text{ AND } t=l)))\}$

Hoặc,

Q6A: $\{q, s \mid (\exists t)(\text{EMPLOYEE}(qrstuvwxyz) \text{ AND } ((\forall l)(\text{NOT}(\text{DEPENDENT}(lmnop)) \text{ OR NOT}(t=l))))\}$

Ví dụ (4/4)

Query 7. “*Lập danh sách tên các người quản lý có ít nhất một người phụ thuộc.*”

Q7: $\{s, q \mid (\exists t)(\exists j)(\exists l)(\text{EMPLOYEE}(qrstuvwxyz) \text{ AND } \text{DEPARTMENT}(hijk) \text{ AND } \text{DEPENDENT}(lmnop) \text{ AND } t=j \text{ AND } l=t)\}$

Hết phần này!

Bài tập chương 8.