

Trường Đại học Công nghệ thông tin

Ôn thi cao học năm 2009

Đại số quan hệ và ngôn ngữ SQL
Relational Algebra & SQL

PGS.TS. Đỗ Phúc

Khoa Hệ thống thông tin

Năm phép toán cơ bản

- Selection (σ) : chọn.
- Projection (π) : chiếu.
- Descartes-product (\times) tích Descartes.
- Set-difference ($-$) hiệu
- Union (\cup) hợp.

Chiếu (projection)

<u>sid</u>	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

S2

Ví dụ:

Chỉ giữ lại các thuộc tính được
chỉ định trong danh sách chiếu.

sname	rating
yuppy	9
lubber	8
guppy	5
rusty	10

$\pi_{sname, rating}(S2)$

age
35.0
55.5

$\pi_{age}(S2)$

Chọn -selection (σ)

- Chọn các dòng thỏa điều kiện.

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

$\sigma_{rating > 8}(S2)$

sname	rating
yuppy	9
rusty	10

$\pi_{sname, rating}(\sigma_{rating > 8}(S2))$

Hợp và hiệu

- Các phép toán này yêu cầu 2 quan hệ nhập phải tương thích
 - Cùng số thuộc tính.
 - Cùng tên và kiểu.

Hợp- Union

<u>sid</u>	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S1

<u>sid</u>	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

S2

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0
44	guppy	5	35.0
28	yuppy	9	35.0

$S1 \cup S2$

Hiệu- Set Difference

<u>sid</u>	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S1

<u>sid</u>	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

S2

sid	sname	rating	age
22	dustin	7	45.0

$S1 - S2$

<u>sid</u>	sname	rating	age
28	yuppy	9	35.0
44	guppy	5	35.0

$S2 - S1$

Toán tử giao (Intersection)

- Các quan hệ phải tương thích
- Có thể biểu diễn qua phép hợp và phép hiệu

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

Giao - Intersection

<u>sid</u>	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S1

<u>sid</u>	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

S2

sid	sname	rating	age
31	lubber	8	55.5
58	rusty	10	35.0

$S1 \cap S2$

Tích Descartes

- Kết từng dòng của quan hệ đầu (R) với từng dòng của quan hệ thứ hai (S):

$$R \times S$$

Ví dụ về tích Descartes

Students

<i>stud#</i>	<i>name</i>	<i>course</i>
100	Fred	PH
200	Dave	CM
300	Bob	CM

Courses

<i>course#</i>	<i>name</i>
PH	Pharmacy
CM	Computing

Students X Courses =

<i>stud#</i>	<i>Students.name</i>	<i>course</i>	<i>course#</i>	<i>Courses.name</i>
100	Fred	PH	PH	Pharmacy
100	Fred	PH	CM	Computing
200	Dave	CM	PH	Pharmacy
200	Dave	CM	CM	Computing
300	Bob	CM	PH	Pharmacy
300	Bob	CM	CM	Computing

Theta kết

- Tích Descartes có áp dụng điều kiện:

$$R \bowtie \langle \text{condition} \rangle S$$

Ví dụ về Theta kết

Students

<i>stud#</i>	<i>name</i>	<i>course</i>
100	Fred	PH
200	Dave	CM
300	Bob	CM

Courses

<i>course#</i>	<i>name</i>
PH	Pharmacy
CM	Computing

Students ⋈_{*stud# = 200*} **Courses**

<i>stud#</i>	<i>Students.name</i>	<i>course</i>	<i>course#</i>	<i>Courses.name</i>
200	Dave	CM	PH	Pharmacy
200	Dave	CM	CM	Computing

Phép kết trong :Inner Join (Equijoin)

- **Phép kết Theta với điều kiện $\langle \text{condition} \rangle$ là bằng nhau (=) giữa khóa chính và khóa ngoại.**

$$\mathbf{R} \bowtie \langle \text{R.primary_key} = \text{S.foreign_key} \rangle \mathbf{S}$$

Ví dụ về Inner Join

Students

<i>stud#</i>	<i>name</i>	<i>course</i>
100	Fred	PH
200	Dave	CM
300	Bob	CM

Courses

<i>course#</i>	<i>name</i>
PH	Pharmacy
CM	Computing

Students ⋈ *course = course#* **Courses**

<i>stud#</i>	<i>Students.name</i>	<i>course</i>	<i>course#</i>	<i>Courses.name</i>
100	Fred	PH	PH	Pharmacy
200	Dave	CM	CM	Computing
300	Bob	CM	CM	Computing

Phép kết tự nhiên (natural join)

- Phép kết trong (inner join) sinh ra dữ liệu dư (trong ví dụ trước ta có: course và course#). Để loại bỏ trường hợp trùng này, ta dùng:

π *< stud#, Students.name, course, Courses.name >*
(Students \bowtie *<course = course#>* Courses)

hay

R1 = Students \bowtie *<course = course#>* Courses

R2 = π *< stud#, Students.name, course, Courses.name >* R1

Kết quả này được gọi là kết tự nhiên (natural join) giữa Students và Courses

Ví dụ về kết tự nhiên (Natural Join)

Students

<i>stud#</i>	<i>name</i>	<i>course</i>
100	Fred	PH
200	Dave	CM
300	Bob	CM

Courses

<i>course#</i>	<i>name</i>
PH	Pharmacy
CM	Computing

R1 = Students ⋈ **<course = course#> Courses**

R2 = π **< stud#, Students.name, course, Courses.name >** **R1**

<i>stud#</i>	<i>Students.name</i>	<i>course</i>	<i>Courses.name</i>
100	Fred	PH	Pharmacy
200	Dave	CM	Computing
300	Bob	CM	Computing

Phép kết ngoài (Outer Joins)

- **Phép kết trong + các dòng của bảng không thỏa <condition>.**
- **Phép kết ngoài trái: $R \bowtie_{<R.primary_key = S.foreign_key>} S$**
Tất cả các dòng trong R được giữ lại và các dòng không thỏa của S được điền các trị NULL
- **Phép kết ngoài phải: $R \bowtie_{<R.primary_key = S.foreign_key>} S$**
Tất cả các dòng trong S được giữ lại và các dòng không thỏa của R được điền các trị NULL

Ví dụ về phép kết ngoài trái


Left Outer Join

Students

<i>stud#</i>	<i>name</i>	<i>course</i>
100	Fred	PH
200	Dave	CM
400	Peter	EN

Courses

<i>course#</i>	<i>name</i>
PH	Pharmacy
CM	Computing
CH	Chemistry

Students  **<course = course#> Courses**

<i>stud#</i>	<i>Students.name</i>	<i>course</i>	<i>course#</i>	<i>Courses.name</i>
100	Fred	PH	PH	Pharmacy
200	Dave	CM	CM	Computing
400	Peter	EN	NULL	NULL

Ví dụ về phép kết ngoài phải

Right Outer Join

Students

<i>stud#</i>	<i>name</i>	<i>course</i>
100	Fred	PH
200	Dave	CM
400	Peter	EN

Courses

<i>course#</i>	<i>name</i>
PH	Pharmacy
CM	Computing
CH	Chemistry

Students  **Courses**
<course = course#>

<i>stud#</i>	<i>Students.name</i>	<i>course</i>	<i>course#</i>	<i>Courses.name</i>
100	Fred	PH	PH	Pharmacy
200	Dave	CM	CM	Computing
NULL	NULL	NULL	CH	Chemistry

Phép chia: Division

- Dùng cho biểu thức “tất cả ” trong truy vấn sau:
Tìm mã thủy thủ (sids) đặt tất cả (all) các thuyền (boats).
- Vd: cho A có 2 thuộc tính x và y ; B chỉ có một thuộc tính y :

$$A/B = \{ \langle x \rangle \mid \forall \langle y \rangle \in B (\exists \langle x, y \rangle \in A) \}$$

A/B (chia) chứa tất cả các bộ (x) sao cho mọi bộ y trong B đều có bộ xy trong A .

Ví dụ phép chia A/B

sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

A

pno
p2

B1

sno
s1
s2
s3
s4

A/B1

pno
p2
p4

B2

sno
s1
s4

A/B2

pno
p1
p2
p4

B3

sno
s1

A/B3

Ví dụ về phép chia

- **Tìm tất cả khách hàng có tài khoản tại tất cả chi nhánh nằm ở ChVille**
 - Branch (bname, assets, bcity)
 - Account (bname, acct#, cname, balance)

Ví dụ về phép chia

R1: Tìm tất cả các chi nhánh tại Chville

R2: Tìm cặp (bname, cname) từ Account

R3: Các khách của r2 có tài khoản tại tất cả chi nhánh có tên trong r1

$$r1 = \pi_{bname} (\sigma_{bcity = 'Chville', Branch})$$

$$r2 = \pi_{bname, cname} (Account)$$

$$r3 = r2 \div r1$$

Biểu diễn phép chia A/B dùng các toán tử cơ bản

$$\pi_x((\pi_x(A) \times B) - A)$$

Cơ sở dữ liệu

- **Cho CSDL câu lạc bộ thuyền buồm**
- Sailors(Sid,Sname, rating, age):thủy thủ
- Boat(Bid,color): thuyền
- Reserves(Sid,Bid,ReservedDate):đặt chỗ

Tìm tên thủy thủ (sailors) đặt thuyền có mã (bid) #103

- **Solution 1:** $\pi_{sname}((\sigma_{bid=103} Reserves) \bowtie Sailors)$
- **Solution 2:** $\pi_{sname}(\sigma_{bid=103}(Reserves \bowtie Sailors))$

Sailors(Sid,Sname, rating, age):thủy thủ

Boat(Bid,color): thuyền

Reserves(Sid,Bid,ReservedDate):đặt chỗ

Tìm tên thủy thủ (sailors) đặt thuyền màu đỏ (red boat)

- Thông tin về **màu đỏ** chỉ có trong quan hệ **Boats**, ta cần thêm một phép kết:

$$\pi_{sname}((\sigma_{color='red'} Boats) \bowtie Reserves \bowtie Sailors)$$

- ❖ Giải pháp tốt hơn (**tối ưu truy vấn**):

$$\pi_{sname}(\pi_{sid}((\pi_{bid} \sigma_{color='red'} Boats) \bowtie Res) \bowtie Sailors)$$

Sailors(Sid,Sname, rating, age):thủy thủ

Boat(Bid,color): thuyền

Reserves(Sid,Bid,ReservedDate):đặt chỗ

Tìm tên thủy thủ (sailors) đặt thuyền màu đỏ và màu xanh lục

$$\rho (Tempred, \pi_{sid}((\sigma_{color='red'} Boats) \bowtie Reserves))$$

$$\rho (Tempgreen, \pi_{sid}((\sigma_{color='green'} Boats) \bowtie Reserves))$$

$$\pi_{sname}((Tempred \cap Tempgreen) \bowtie Sailors)$$

Tìm tên thủy thủ (sailors) đặt tất cả các thuyền

- **Phép chia:**

$$\rho (Tempsids, (\pi_{sid,bid}^{Reserves}) / (\pi_{bid}^{Boats}))$$

$$\pi_{sname} (Tempsids \bowtie Sailors)$$

Sailors(Sid,Sname, rating, age):thủy thủ

Boat(Bid,color): thuyền

Reserves(Sid,Bid,ReservedDate):đặt chỗ

Chứng minh biểu thức đại số quan hệ

Cho q là quan hệ trên tập thuộc tính RS ($R \cup S$), chứng minh: $q \subseteq \Pi_R(q) \bowtie \Pi_S(q)$

Chứng minh:

Cho $t \in q$, theo đ/n của phép chiếu $t[R] \in \Pi_R(q)$ và $t[S] \in \Pi_S(q)$

Ta có theo định nghĩa của phép kết:

$$t[RS] \in \Pi_R(q) \bowtie \Pi_S(q)$$

Do vậy $t = t[RS] \in \Pi_R(q) \bowtie \Pi_S(q)$

Chứng minh biểu thức đại số quan hệ

- Cho quan hệ $r(R)$, A và B là hai thuộc tính của quan hệ R , chứng minh:

$$\sigma_{A=a}(\sigma_{B=b}(r)) = \sigma_{B=b}(\sigma_{A=a}(r))$$

- Chứng minh:

$$\begin{aligned}\sigma_{A=a}(\sigma_{B=b}(r)) &= \sigma_{A=a}(\{t \in r \mid t[B]=b\}) = \\ &\{t' \in \{t \in r \mid t[B]=b\} \mid t'[A]=a\} = \\ &\{t \in r \mid t[A]=a \wedge t[B]=b\} = \\ &\{t' \in \{t \in r \mid t[A]=a\} \mid t'[B]=b\} = \\ &\sigma_{B=b}(\sigma_{A=a}(r))\end{aligned}$$

Chứng minh biểu thức đại số quan hệ

- Cho 2 quan hệ r, s , A là thuộc tính của quan hệ R, S . Chứng minh:

$$\sigma_{A=a}(r \cap s) = \sigma_{A=a}(r) \cap \sigma_{A=a}(s)$$

- Chứng minh:

$$\begin{aligned}\sigma_{A=a}(r \cap s) &= \sigma_{A=a}(t \in r \wedge t \in s) = \\ \{ t' \in \{ t | t \in r \wedge t \in s \} \mid t[A]=a \} &= \\ \{ t | t \in r \text{ và } t[A]=a \} \cap \{ t | t \in s \text{ và } t[A]=a \} &= \\ \sigma_{A=a}(r) \cap \sigma_{A=a}(s) &\end{aligned}$$

Đại số quan hệ và ngôn ngữ SQL

Toán tử một ngôi (unary operations)

Selection

σ course = 'Computing' **Students**

In SQL:

Select *

From Students

Where course = 'Computing';

Projection

π stud#, name **Students**

In SQL:

Select stud#, name

From Students;

Selection & Projection

π stud#, name (σ course = 'Computing' **Students**)

In SQL:

Select stud#, name

From students

Where course = 'Computing';

Toán tử hai ngôi/kết Binary Operations/Joins

Tích Descartes: Students X Courses

In SQL:

Select *

From Students, Courses;

Theta kết: Students ⋈ <stud# =200> Courses

In SQL:

Select *

From Students, Courses

Where stud# = 200;

Toán tử hai ngôi/kết

Inner Join (Equijoin): $\text{Students} \bowtie \langle \text{course} = \text{course\#} \rangle \text{Courses}$

In SQL:

Select *

From Students, Courses

Where course=course#;

Natural Join:

$R1 = \text{Students} \bowtie \langle \text{course} = \text{course\#} \rangle \text{Courses}$

$R2 = \pi \langle \text{stud\#}, \text{Students.name}, \text{course}, \text{Courses.name} \rangle R1$

In SQL:

Select stud#, Students.name, course, Courses.name

From Students, Courses

Where course=course#;

Phép kết ngoài (Outer Joins)

Left Outer Join

Students ⋈_{<course = course#>} **Courses**

In SQL:

Select *

From Students, Courses

Where course = course#(+)

Right Outer Join

Students ⋈_{<course = course#>} **Courses**

In SQL:

Select *

From Students, Courses

Where course(+)= course#

Tổ hợp các phép toán một ngôi và hai ngôi

$R1 = \text{Students} \bowtie \langle \text{course} = \text{course\#} \rangle \text{Courses}$

$R2 = \sigma \langle \text{address} = \text{"Aberdeen"} \rangle R1$

$R3 = \pi \langle \text{Students.name, Course.name} \rangle R2$

In SQL:

Select Students.name, Courses.name

From Students, Courses

Where course=course#

AND address="Aberdeen";

Các toán tử tập hợp

Union: $R \cup S$

In SQL:

Select * From R

Union

Select * From S;

Intersection: $R \cap S$

In SQL:

Select * From R

Intersect

Select * From S;

Difference: $R - S$

In SQL:

Select * From R

Minus

Select * From S;

Các toán tử trong SQL

Between, In, Like, Not

Các toán tử SQL

```
SELECT *  
FROM Book  
WHERE catno BETWEEN 200 AND 400;
```

```
SELECT *  
FROM Product  
WHERE prod_desc BETWEEN 'C' AND 'S';
```

```
SELECT *  
FROM Book  
WHERE catno NOT BETWEEN 200 AND 400;
```

Các toán tử SQL

```
SELECT Catno  
FROM Loan  
WHERE Date-Returned IS NULL;
```

```
SELECT Catno  
FROM Loan  
WHERE Date-Returned IS NOT NULL;
```

Các toán tử SQL

```
SELECT Name  
FROM Member  
WHERE memno IN (100, 200, 300, 400);
```

```
SELECT Name  
FROM Member  
WHERE memno NOT IN (100, 200, 300, 400);
```

Các toán tử SQL

```
SELECT Name  
FROM Member  
WHERE address NOT LIKE '%Aberdeen%';
```

```
SELECT Name  
FROM Member  
WHERE Name LIKE '_ES%';
```

Note: In MS Access, use * and # instead of % and _

Chọn giá trị phân biệt

Student

<i>stud#</i>	<i>name</i>	<i>address</i>
100	Fred	Aberdeen
200	Dave	Dundee
300	Bob	Aberdeen

**SELECT Distinct address
FROM Student;**

<i>address</i>
Aberdeen
Dundee

Các bài tập về SQL

Lược đồ CSDL

- **Professor(ssn, profname, status, salary)**
- **Course(crscode, crsname, credits)**
- **Taught(crscode, semester, ssn)**

Giá định (1) Mỗi khóa học chỉ có một giáo sư phụ trách trong mỗi học kỳ; (2) tất cả giáo sư đều có lương khác nhau; (3) tất cả giáo sư có tên khác nhau; (4) tất cả khóa học có tên khác nhau; (5) thuộc tính status có thể có các giá trị “Full”, “Associate”, và “Assistant”.

Truy vấn 1

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê tất cả giáo sư đã dạy khóa học có mã 'CSC6710' nhưng không dạy khóa học có mã 'CSC7710'.

Lời giải ĐSQH

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{\text{ssn}}(\sigma_{\text{crscode}='CSC6710'}(\text{Taught})) -$

$\pi_{\text{ssn}}(\sigma_{\text{crscode}='CSC7710'}(\text{Taught}))$

Lời giải SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
(SELECT ssn  
From Taught  
Where crscode = 'CSC6710')  
EXCEPT  
(SELECT ssn  
From Taught  
Where crscode = 'CSC7710'))
```

Truy vấn 2

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê các giáo sư đã dạy các khóa học có mã 'CSC6710' và 'CSC7710'.

Đại số quan hệ

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{\text{ssn}}(\sigma_{\text{crscode}='CSC6710' \wedge \text{crscode}='CSC7710'}(\text{Taught}), \text{SAI!})$

$\pi_{\text{ssn}}(\sigma_{\text{crscode}='CSC6710'}(\text{Taught})) \cap$
 $\pi_{\text{ssn}}(\sigma_{\text{crscode}='CSC7710'}(\text{Taught})), \text{ĐÚNG!})$

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT T1.ssn  
From Taught T1, Taught T2,  
Where T1.crscode = 'CSC6710' AND T2.crscode='CSC7710' AND  
T1.ssn=T2.ssn
```

Truy vấn 3

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê các giáo sư không dạy môn có mã số 'CSC7710'.

Đại số quan hệ

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{\text{ssn}}(\sigma_{\text{crscode} \neq \text{'csc7710'}}(\text{Taught}))$, **SAI!**

$\pi_{\text{ssn}}(\text{Professor}) - \pi_{\text{ssn}}(\sigma_{\text{crscode} = \text{'csc7710'}}(\text{Taught}))$,
ĐÚNG!

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
(SELECT ssn  
From Professor)  
EXCEPT  
(SELECT ssn  
From Taught T  
Where T.crscode = 'CSC7710')
```

Truy vấn 4

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê các giáo sư đã dạy môn học có mã ‘CSC6710’ và ‘CSC7710’ trong cùng một học kỳ

Đại số quan hệ

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{ssn}(\sigma_{crscode1='CSC6710'}(Taught[crscode1, ssn, semester])) \bowtie$

$\sigma_{crscode2='CSC7710'}(Taught[crscode2, ssn, semester]))$

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

SELECT T1.ssn

From Taught T1, Taught T2,

Where T1.crscode = 'CSC6710' AND T2.crscode='CSC7710' AND

T1.ssn=T2.ssn AND T1.semester=T2.semester

Truy vấn 5

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê các giáo sư đã dạy môn có mã
'CSC6710' hay môn học có mã
'CSC7710" nhưng không dạy cả hai
môn.

Đại số quan hệ

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{ssn}(\sigma_{crscode='CSC6710' \vee$
 $crscode='CSC7710', (Taught)) -$
 $(\pi_{ssn}(\sigma_{crscode='CSC6710', (Taught))) \cap$
 $\pi_{ssn}(\sigma_{crscode='CSC7710', (Taught)))$

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
(SELECT ssn
FROM Taught T
WHERE T.crscode='CSC6710' OR T.crscode='CSC7710')
Except
(SELECT T1.ssn
From Taught T1, Taught T2,
Where T1.crscode = 'CSC6710') AND T2.crscode='CSC7710' AND
T1.ssn=T2.ssn)
```

Truy vấn 6

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Trả về các khóa học không bao giờ mở.

Đại số quan hệ

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$$\pi_{\text{crscode}}(\text{Course}) - \pi_{\text{crscode}}(\text{Taught})$$

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
(SELECT crscode  
FROM Course)  
EXCEPT  
(SELECT crscode  
FROM TAUGHT  
)
```

Truy vấn 7

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê các khóa học được dạy tối thiểu trong 2 học kỳ.

Đại số quan hệ

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{\text{crscode}}(\sigma_{\text{semester1} \neq \text{semester2}}($

Taught[crscode, ssn1, semester1] \bowtie

Taught[crscode, ssn2, semester2]))

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

SELECT T1.crscode

FROM Taught T1, Taught T2

WHERE T1.crscode=T2.crscode AND T1.semester <> T2.semester

Truy vấn 8

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê các khóa học được dạy trong ít nhất là 10 học kỳ.

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT crscode
FROM Taught
GROUP BY crscode
HAVING COUNT(*) >= 10
```

Truy vấn 9

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê các khóa học được dạy trong ít nhất là 5 giáo sư khác nhau.

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT crscode
FROM (SELECT DISTINCT crscode, ssn FROM TAUGHT)
GROUP BY crscode
HAVING COUNT(*) >= 5
```

```
SELECT crscode
FROM Course C
WHERE (SELECT COUNT(DISTINCT *) FROM Taught T
      WHERE T.crscode = C.crscode
      ) >=5.
```

Truy vấn 10

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê tên các giáo sư đã dạy khóa học có mã 'CSC6710'.

Đại số quan hệ

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{\text{profname}}(\sigma_{\text{crscode}='CSC6710'}(\text{Taught}) \bowtie \text{Professor})$

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT P.profname  
FROM Professor P, Taught T  
WHERE P.ssn = T.ssn AND T.crscode = 'CSC6710'
```

Truy vấn 11

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê tên của các full professors đã dạy môn học có mã 'CSC6710'.

Đại số quan hệ

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{\text{profname}}(\sigma_{\text{crscode}='csc6710'}(\text{Taught}) \bowtie$
 $\sigma_{\text{status}='full'}(\text{Professor}))$

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT P.profname  
FROM Professor P, Taught T  
WHERE P.status = 'full' AND P.ssn = T.ssn AND T.crscode = 'CSC6710'
```

Truy vấn 12

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê tên của các full professors đã dạy ít nhất là 2 khóa học trong một học kỳ.

Đại số quan hệ

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{\text{profname}}(\pi_{\text{ssn}}(\sigma_{\text{crscode1} \neq \text{crscode2}}($

Taught[crscode1, ssn, semester] 

Taught[crscode2, ssn, semester])) 

$\sigma_{\text{status}='full'}(\text{Professor}))$

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT P.profname  
FROM Professor P, Taught T1, Taught T2  
WHERE P.status = 'Full' AND P.ssn = T1.ssn AND T1.ssn = T2.ssn  
AND T1.crscode <> T2.crscode AND T1.semester = T2.semester
```

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT P.profname
FROM Professor P
WHERE status = 'Full' AND ssn IN(
SELECT ssn
FROM Taught
GROUP BY ssn, semester
HAVING COUNT(*) >= 2
)
```

Truy vấn 13

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Xóa các giáo sư không dạy khóa học nào.

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
DELETE FROM Professor
WHERE ssn NOT IN
(SELECT ssn
FROM Taught
)
```

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
DELETE FROM Professor
WHERE ssn IN
(
  (SELECT ssn FROM Professor)
  EXCEPT
  (SELECT ssn FROM Taught)
)
```

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
DELETE FROM Professor P
WHERE NOT EXISTS
(
  SELECT * FROM Taught T
  WHERE T.ssn = P.ssn
)
```

Truy vấn 14

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Đổi tất cả tín chỉ (credits) sang 4 cho các khóa học được giảng dạy trong học kỳ mùa thu năm 2006 (f2006 semester).

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
UPDATE Course
SET credits = 4
WHERE crscode IN
(
    SELECT crscode
    FROM Taught
    WHERE semester = 'f2006'
)
```

Truy vấn 15

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê tên của các giáo sư đã dạy hơn 30 tín chỉ khóa học.

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT profname
FROM Professor
WHERE ssn IN
(
    SELECT T.ssn
    FROM Taught T, Course C
    WHERE T.crscode = C.crscode
    GROUP BY T.ssn
    HAVING SUM(C.credits) > 30
)
```

Truy vấn 16

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê tên của các giáo sư đã dạy nhiều khóa học nhất trong học kỳ mùa xuân năm 2006 (S2006).

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT profname
FROM Professor
WHERE ssn IN(
    SELECT ssn FROM Taught
    WHERE semester = 'S2006'
    GROUP BY ssn
    HAVING COUNT(*) =
        (SELECT MAX(Num)
         FROM
            (SELECT ssn, COUNT(*) as Num
             FROM Taught
             WHERE semester = 'S2006'
             GROUP BY ssn)
        )
)
```

Truy vấn 17

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê tên các khóa học mà giáo sư
‘Smith’ đã dạy trong học kỳ mùa thu
năm 2007.

Đại số quan hệ

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{\text{crsname}}(\sigma_{\text{profname}='Smith'}(\text{Professor})) \bowtie$

$\sigma_{\text{semester}='f2007'}(\text{Taught}) \bowtie$

Course)

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

SELECT crsname

FROM Professor P, Taught T, Course C

WHERE P.profname = 'Smith' AND P.ssn = T.ssn AND
T.semester = 'F2007' AND T.crscode = C.crscode

Truy vấn 18

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Hãy liệt kê theo thứ tự thời gian các khóa học mà giáo sư có mã số ssn = 123456789 đã dạy trong từng học kỳ.

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT semester, COUNT(*)  
FROM Taught  
WHERE ssn = '123456789'  
GROUP BY semester  
ORDER BY semester ASC
```

Truy vấn 19

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Hãy liệt kê theo thứ tự từ điển tên của từng giáo sư và số các khóa học đã giảng dạy.

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT P.profname, COUNT(*)  
FROM Professor P, Taught T  
WHERE P.ssn = T.ssn  
GROUP BY P.ssn, P.profname  
ORDER BY P.profname ASC
```

Truy vấn 20

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Xóa các giáo sư đã dạy ít hơn 10 khóa học.

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
DELETE FROM Professor
WHERE ssn IN(
    SELECT ssn
    FROM Taught
    GROUP BY ssn
    HAVING COUNT(*) < 10
)
```

Truy vấn 21

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Xóa các giáo sư đã dạy ít hơn 40 tín chỉ.

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
DELETE FROM Professor
WHERE ssn IN(
    SELECT T.ssn
    FROM Taught T, Course C
    WHERE T.crscode = C.crscode
    GROUP BY ssn
    HAVING SUM(C.credits) < 40
)
```


Truy vấn 22

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê các giáo sư không dạy bất kỳ khóa học nào trong 3 học kỳ (F2006, W2007, F2007).

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT *  
FROM Professor P  
WHERE NOT EXISTS(  
    SELECT *  
    FROM Taught  
    WHERE P.ssn = T.ssn AND (T.semester = 'F2006' OR  
        T.semester = 'W2007' OR T.semester='F2007'))  
)
```

Truy vấn 23

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê tên các khóa học mà giáo sư Smith không dạy.

Đại số quan hệ

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{\text{crsname}}(\text{Course}) -$

$\pi_{\text{crsname}}(\sigma_{\text{profname}='Smith'}(\text{Professor})) \bowtie$
 $(\text{Taught}) \bowtie \text{Course}$

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT crsname
FROM Course C
WHERE NOT EXISTS
  SELECT *
  FROM Professor P, Taught T
  WHERE P.profname='Smith' AND P.ssn = T.ssn AND
  T.crscode = C.crscode
)
```

Truy vấn 24

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê tên các khóa học được giảng dạy bởi **tất cả** giáo sư.

Đại số quan hệ

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$$\pi_{\text{crscode, ssn}}(\text{Taught}) / \pi_{\text{ssn}}(\text{Professor})$$

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT crscode
FROM Taught T1
WHERE NOT EXISTS(
    (SELECT ssn
     FROM Professor)
    EXCEPT
    (SELECT ssn
     FROM Taught T2
     WHERE T2.crscode = T1.crscode)
)
```


Truy vấn 25

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê tên của các khóa học được giảng dạy trong tất cả các học kỳ.

Đại số quan hệ

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$$\pi_{\text{crscode, semester}}(\text{Taught}) / \pi_{\text{semester}}(\text{Taught})$$

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT crscode
FROM Taught T1
WHERE NOT EXISTS(
    (SELECT semester
     FROM Taught)
    EXCEPT
    (SELECT semester
     FROM Taught T2
     WHERE T2.crscode = T1.crscode)
)
```

Truy vấn 26

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê các khóa học CHỈ được giảng dạy bởi các trợ lý giáo sư (assisitant) professors.

Đại số quan hệ

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{\text{crscode}}(\text{Course}) - \pi_{\text{crscode}}$
 $(\sigma_{\text{status} \neq \text{'Assistant'}}(\text{Professor}) \bowtie \text{Taught})$

SQL

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT crscode
FROM Course C
WHERE c.crscode NOT IN(
    (SELECT crscode
     FROM Taught T, Professor P
     WHERE T.ssn = P.ssn AND P.status='Junior'
    )
)
```

Truy vấn 27

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê tên của các giáo sư giảng dạy nhiều khóa học nhất trong học kỳ Fall 2001.

SQL Solution

```
SELECT *
FROM Professor P1
WHERE Not EXISTS
(
    SELECT *
    FROM Professor P2
    WHERE(
        (SELECT COUNT(*)
        FROM Taught
        WHERE Taught.ssn = P2.ssn AND
Taught.semester='F2001')
        >
        (SELECT COUNT(*)
        FROM Taught
        WHERE Taught.ssn = P1.ssn AND
Taught.semester='F2001')
    )
)
```


Truy vấn 28

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê tên của giáo sư có lương cao nhất.

SQL Solution

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT *  
FROM Professor  
WHERE salary = (  
    (SELECT MAX(salary)  
    FROM Professor P  
)
```

Truy vấn 29

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Liệt kê tên của giáo sư có lương cao xếp thứ nhì.

SQL Solution

Professor(ssn, profname, status, salary)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

```
SELECT *  
FROM Professor P1  
WHERE 1 = (  
    (SELECT COUNT(*)  
     FROM Professor P2  
     WHERE P2.salary > P1.salary  
    )  
)
```