

Relational Algebra and SQL Exercises

Revised based on the notes from Dr. Shiyong Lu of Wayne State University
Database Systems: An Application Oriented Approach, Edition 2
by Michael Kifer, Arthur Bernstein, Philip M. Lewis
ISBN: 0321268458
ISBN13: 9780321268457
Release Date: March 2005

Relational Algebra and SQL Exercises

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

Query 1

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

Return those professors who have taught 'CSCI203' but never 'CSCI240'.

Relational Algebra Solution

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

$$\pi_{ssn}(\sigma_{crsname='CSCI203'}(Taught)) - \pi_{ssn}(\sigma_{crsname='CSCI240'}(Taught))$$

SQL Solution

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

```
SELECT ssn
FROM Taught
WHERE crsname = 'CSCI203'
EXCEPT
SELECT ssn
FROM Taught
WHERE crsname = 'CSCI240';
```

Query 2

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

Return those professors who have taught both 'CSCI203' and 'CSCI204'.

Relational Algebra Solution

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

$\pi_{ssn}(\sigma_{crsname='CSCI203' \wedge crsname='CSCI204'}(Taught))$, wrong!

$\pi_{ssn}(\sigma_{crsname='CSCI203'}(Taught)) \cap \pi_{ssn}(\sigma_{crsname='CSCI204'}(Taught))$, correct!

SQL Solution

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

```
SELECT T1.ssn
From Taught T1, Taught T2,
Where T1.crsname = 'CSCI203' AND T2.crsname='CSCI204' AND
T1.ssn=T2.ssn;
```

Query 3

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

Return those professors who have never taught 'CSCI204'.

Relational Algebra Solution

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

$\pi_{ssn}(\sigma_{crsname \neq 'CSCI204'}(Taught))$, wrong answer!

$\pi_{ssn}(Professor) - \pi_{ssn}(\sigma_{crsname='CSCI204'}(Taught))$, correct answer!

SQL Solution

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

```
SELECT ssn
From Professor
EXCEPT
SELECT ssn
From Taught T
Where T.crsname = 'CSCI204';
```

Query 4

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

Return those professors who taught 'CSCI203' and 'CSCI204' in the same semester

Relational Algebra Solution

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

$\pi_{ssn}(\sigma_{crsname1='CSCI203'}(Taught[crsname1, ssn, semester])) \bowtie$
 $\sigma_{crsname2='CSCI204'}(Taught[crsname2, ssn, semester]))$

SQL Solution

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

```

SELECT T1.ssn
From Taught T1, Taught T2,
Where T1.crsname = 'CSCI203' AND T2.crsname='CSCI204' AND
T1.ssn=T2.ssn AND T1.semester=T2.semester;

```

Query 5

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

Return those professors who taught 'CSCI204' or 'CSCI315' but not both.

Relational Algebra Solution

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

$\pi_{ssn}(\sigma_{crsname \neq 'CSCI204' \vee$
 $crsname='CSCI315'}(Taught)) -$
 $(\pi_{ssn}(\sigma_{crsname='CSCI204'}(Taught))) \cap$
 $\pi_{ssn}(\sigma_{crsname='CSCI315'}(Taught)))$

SQL Solution

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

```

SELECT ssn
FROM Taught T
WHERE T.crsname='CSCI204' OR T.crsname='CSCI315'
Except
SELECT T1.ssn
From Taught T1, Taught T2
Where T1.crsname = 'CSCI204' AND T2.crsname='CSCI315' AND
T1.ssn=T2.ssn;

```

Query 6

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

Return those courses that have never been taught.

Relational Algebra Solution

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

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

SQL Solution

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

```
SELECT crsname
FROM Course)
EXCEPT
SELECT crsname
FROM Taught;
```

Query 7

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

Return those courses that have been taught in two different semesters.

Relational Algebra Solution

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

$\pi_{\text{crscode}}(\sigma_{\text{semester1} \neq \text{semester2}}(\text{Taught}[\text{crsname}, \text{ssn1}, \text{semester1}] \bowtie \text{Taught}[\text{crsname}, \text{ssn2}, \text{semester2}])))$

SQL Solution

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

```
SELECT distinct T1.crsname
FROM Taught T1, Taught T2
WHERE T1.crsname=T2.crsname AND T1.semester <> T2.semester;
```

Query 8

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

Return those courses that have been taught at least twice.

SQL Solution

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

This is an example of queries in which relational algebra can't express accurately as RA doesn't have aggregate functions.

```
SELECT crsname
FROM Taught
GROUP BY crsname
HAVING COUNT(*) >= 2;
```

Query 9

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

Return those courses that have been taught by at least 3 different professors.

SQL Solution

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

This is an example of queries in which relational algebra can't express accurately as RA doesn't have aggregate functions.

```
SELECT crsname
FROM (SELECT DISTINCT crsname, ssn FROM TAUGHT)
GROUP BY crsname
HAVING COUNT(*) >= 3
```

Query 10

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

Return the names of the professors who ever taught 'CSCI204'.

Relational Algebra Solution

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

$\pi_{\text{profname}}(\sigma_{\text{crscode}='CSCI204'}(\text{Taught}) \bowtie \text{Professor})$

SQL Solution

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

```
SELECT P.profname
FROM Professor P, Taught T
WHERE P.ssn = T.ssn AND T.crsname = 'CSCI204';
```

Query 11

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

Return the names of the senior professors who ever taught 'CSCI204'.

Relational Algebra Solution

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

$$\pi_{\text{profname}}(\sigma_{\text{crsname}='CSCI204'}(\text{Taught}) \bowtie \sigma_{\text{status}='senior'}(\text{Professor}))$$

SQL Solution

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

```
SELECT P.profname
FROM Professor P, Taught T
WHERE P.status = 'senior' AND P.ssn = T.ssn AND T.crsname = 'CSCI204';
```