Relational Algebra and SQL Exercises

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

Slides are adapted from presentation by Dr. Shiyong Lu at Wayne State University

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Assumption:

(1) Each course has only one instructor in each semester.

(2) all professors have different names.

(3) all courses have different names.

(4) status can take values from “Full”, “Associate”, and “Assistant”.

Query 1

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

Return those professors who have taught ‘csc6710’ but never ‘csc7710’.

Relational Algebra Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{ssn}(\sigma_{crscode=`csc6710'}(Taught))-$
$\pi_{ssn}(\sigma_{crscode=`csc7710'}(Taught))$

SQL Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

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

Query 2

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

Return those professors who have taught both ‘csc6710’ and ‘csc7710’.

Relational Algebra Solution

Professor(<u>ssn</u>, profname, status)
Course(<u>crscode</u>, crsname, credits)
Taught(<u>crscode, semester</u>, ssn)

$\pi_{ssn}(\sigma_{crscode=\text{`csc6710'} \wedge crscode=\text{`csc7710'}}(\text{Taught})$, wrong!

$\pi_{ssn}(\sigma_{crscode=\text{`csc6710'}}(\text{Taught})) \cap$
$\pi_{ssn}(\sigma_{crscode=\text{`csc7710'}}(\text{Taught}))$, correct!

SQL Solution

Professor(ssn, profname, status)

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
```

Query 3

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

Return those professors who have never taught ‘csc7710’.

Relational Algebra Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{ssn}(\sigma_{crscode<>'csc7710'}(Taught))$, wrong answer!

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

SQL Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

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

Query 4

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

Return those professors who taught ‘CSC6710’ and ‘CSC7710” in the same semester

Relational Algebra Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

Relational Algebra Solution

$\pi_{ssn}(\sigma_{crscode1='csc6710'}(Taught[crscode1, ssn, semester]) \bowtie$

$\sigma_{crscode2='csc7710'}(Taught[crscode2, ssn, semester]))$

SQL Solution

Professor(ssn, profname, status)

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
```

Query 5

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

Return those professors who taught ‘CSC6710’ or ‘CSC7710” but not both.

Relational Algebra Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$$\pi_{ssn}(\sigma_{crscode<>\text{'csc7710'} \vee crscode=\text{'csc7710'}}(Taught)) - (\pi_{ssn}(\sigma_{crscode=\text{'csc6710'}}(Taught)) \cap \pi_{ssn}(\sigma_{crscode=\text{'csc7710'}}(Taught)))$$

SQL Solution

Professor(ssn, profname, status)

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)
```

Query 6

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

Return those courses that have never been taught.

Relational Algebra Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

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

SQL Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

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

Query 7

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

Return those courses that have been taught at least in two semesters.

Relational Algebra Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{\text{crscode}}(\sigma_{\text{semester1} <> \text{semester2}}($

Taught[crscode, ssn1, semester1] $\bowtie$
Taught[crscode, ssn2, semester2]))

SQL Solution

Professor(ssn, profname, status)

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
```

Query 8

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

Return those courses that have been taught at least in 10 semesters.

SQL Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

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

Query 9

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

Return those courses that have been taught by at least 5 different professors.

SQL Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

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

Query 10

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

Return the names of professors who ever taught ‘CSC6710’.

Relational Algebra Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

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

SQL Solution

Professor(ssn, profname, status)

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'
```

Query 11

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

Return the names of full professors who ever taught ‘CSC6710’.

Relational Algebra Solution

Professor(ssn, profname, status)

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 Solution

Professor(ssn, profname, status)

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'
```

Query 12

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

Return the names of full professors who ever taught more than two courses in one semester.

SQL Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

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

Query 13

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

Delete those professors who never taught a course.

SQL Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

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

Query 14

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

Change all the credits to 4 for those courses that are taught in f2006 semester.

SQL Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

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

Query 15

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

Return the names of the professors who have taught more than 30 credits of courses.

SQL Solution

Professor(ssn, profname, status)

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
)
```

Query 16

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

Return the name(s) of the professor(s) who taught the most number of courses in S2006.

SQL Solution

Professor(ssn, profname, status)

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)
          )
)
```

Query 17

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

List all the course names that professor ‘Smith” taught in Fall of 2007.

Relational Algebra Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{crsname}(\sigma_{profname='Smith'}(Professor) \bowtie \sigma_{semester='f2007'}(Taught) \bowtie Course)$

SQL Solution

Professor(ssn, profname, status)

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
```

Query 18

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

In chronological order, list the number of courses that the professor with ssn ssn = 123456789 taught in each semester.

SQL Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

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

Query 19

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

In alphabetical order of the names of professors, list the name of each professor and the total number of courses she/he has taught.

SQL Solution

Professor(ssn, profname, status)

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
```

Query 20

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

Delete those professors who taught less than 10 courses.

SQL Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

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

Query 21

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

Delete those professors who taught less than 40 credits.

SQL Solution

Professor(ssn, profname, status)

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
)
```

Query 22

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

List those professors who have not taught any course in the past three semesters (F2006, W2007, F2007).

SQL Solution

Professor(ssn, profname, status)

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'))
)
```

Query 23

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

List the names of those courses that professor Smith have never taught.

Relational Algebra Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

$\pi_{crsname}$(Course)-

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

Course)

SQL Solution

Professor(ssn, profname, status)

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
)
```

Query 24

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

Return those courses that have been taught by all professors.

Relational Algebra Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

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

SQL Solution

Professor(ssn, profname, status)

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)
)
```

Query 25

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

Return those courses that have been taught in all semesters.

Relational Algebra Solution

Professor(ssn, profname, status)

Course(crscode, crsname, credits)

Taught(crscode, semester, ssn)

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

SQL Solution

Professor(ssn, profname, status)

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)
)
```

Query 25

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

Return those courses that have been taught ONLY by junior professors.

Relational Algebra Solution

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

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

SQL Solution

Professor(ssn, profname, status)

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'
)
```