

CS1555 Recitation 5 - Solution

Objectives: 1. To practice more relational algebra.
2. To practice SQL queries.

Consider the following relation schemas:

Student (SID, Name, Class, Major)

Student_Dir (ID, Address, Phone)

FK: (ID) \rightarrow Student (SID)

Courses_taken (Course_No, Term, SID, Grade)

FK: (Course_No) \rightarrow Course (Course_No); (SID) \rightarrow Student (SID)

Course (Course_No, Course_Name, Level)

Instructor (ID, Fname, Lname)

Courses_offered(Course_No, Term, InstructorID)

FK: (Course_No) \rightarrow Course (Course_No); (InstructorID) \rightarrow Instructor (ID)

Part 1: Relational Algebra

Write a relational algebra query for each of the queries below:

1. List the students who did not enroll in any course in Fall 18.

$$SID_enroll_fall18 \leftarrow \pi_{SID}(\sigma_{Term = 'Fall 18'}(Courses_Taken))$$
$$SID_all \leftarrow \pi_{SID}(Student)$$
$$RSLT \leftarrow SID_all - SID_enroll_fall18$$

If we were interested in all of the student attributes then :

$$\pi_{Student.SID, Student.name, Student.class, Student.major} (Student \bowtie_{Student.SID = RSLT.SID} RSLT)$$

2. (Optional) Find the total number of students.

$$\mathcal{F}_{COUNT\ SID} (Student)$$

3. Find the total number of students who have enrolled in the course “Operating Systems”.

$$OS_Taking \leftarrow \pi_{SID}(\sigma_{Course.Name = 'Operating Systems'}(Course_Taken * Course))$$
$$RSLT \leftarrow \mathcal{F}_{COUNT\ SID} (OS_Taking)$$

(or you can combine the two steps into one expression (nested operations))

4. List the SID, name, and address (if available) of all students.

$\pi_{SID, name, address} (Student \bowtie_{Student.SID=StudentDir.ID} Student_Dir)$

(note the left outer join)

5. Find the SID(s) of the student(s) who has/have the highest GPA

$Student_GPA(SID, GPA) \leftarrow \sigma_{SID} \mathcal{F}_{AVERAGE\ GRADE} (Courses_taken)$

$Highest_GPA (Max_GPA) \leftarrow \mathcal{F}_{MAX\ GPA}(Student_GPA)$

$RSLT \leftarrow \pi_{SID} (Student_GPA \bowtie_{GPA = Max_GPA} (Highest_GPA))$

6. Find the SID(s) of the student(s) who has/have taken all courses at the UGrad level

$Course_Denominator \leftarrow \pi_{Course_No} (\sigma_{Level = 'UGrad'} Course)$

$RSLT \leftarrow (\pi_{SID, Course_No}(Course_Taken)) \div Course_Denominator$

7. Find for each instructor, the course names of the courses he/she was teaching in Fall 18.
List in addition to the course name, the first name and the last names of the instructor.

$\pi_{Course.Name, Instructor.fname, Instructor.Lname} (\sigma_{term='Fall 18'} (Instructor \bowtie_{Instructor.ID = Courses_offered.InstructorID} (Course * Courses_offered)))$

8. (Optional) Find for each instructor the number of courses he/she has taught or is teaching.
List the first name and the last name of each instructor along with his/her ID and number of courses.

$Courses_taught(ID, N_courses) \leftarrow \pi_{InstructorID} \mathcal{F}_{Count\ course_no} (Courses_offered)$

$RSLT \leftarrow Courses_taught * Instructor$

Part 2: SQL

1. List the student ID and course number for every student who took a course in Fall 18 but has not received a grade yet.

```
select ct.sid, ct.course_no
from course_taken ct
where ct.term = 'Fall 18'
       and ct.grade is null;
```

2. List the SIDs and names of all students and the number of courses they have taken.

```
select s.sid, s.name, count(distinct course_no) as num_courses
from student s, course_taken ct
where s.sid = ct.sid
group by s.sid, s.name;
```

Or using the new Join syntax:

```
select s.sid, s.name, count(distinct course_no) as num_courses
from student s join course_taken ct on ct.sid = s.sid
group by s.sid, s.name;
```

3. List the SIDs names and GPAs of the students whose GPAs are greater than 3.7. List them in the descending order of the GPAs.

```
select s.SID, s.name, avg(grade) as GPA
from course_taken ct join student s on ct.sid = s.sid
group by s.sid, s.name
having avg(grade) > 3.7
order by GPA desc;
```