

CS1555 Recitation 5 - Solution

Objectives: 1. To practice SQL queries.

Consider the following relation schemas:

Student (SID, Name, Class, Major)

Student_Dir (ID, Address, Phone)

FK: (ID) → Student (SID)

Courses_taken (Course_No, Term, SID, Grade)

FK: (Course_No) → Course (Course_No); (SID) → Student (SID)

Course (Course_No, Course_Name, Level)

Instructor (ID, Fname, Lname)

Courses_offered(Course_No, Term, InstructorID)

FK: (Course_No) → Course (Course_No); (InstructorID) → Instructor (ID)

1. List the student ID and course number for every student who took a course in Fall 19 but has not received a grade yet.

```
select ct.sid, ct.course_no
from course_taken ct
where ct.term = 'Fall 19'
       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;

```

4. List the students who did not enroll in any course in Fall 19.

```

select SID
from student
where sid not in (
    select SID
    from course_taken
    where term = 'Fall 19');

```

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

```

select sid, course_no -- Postgres
from course_taken ct1
where not exists(
    (select course_no
     from course
     where course_level = 'UGrad')
    except
    (select course_no
     from course_taken ct2
     where ct1.sid = ct2.sid));

```

```

select sid, course_no -- Oracle
from course_taken ct1
where not exists(
    (select course_no
     from course
     where course_level = 'UGrad')
    MINUS
    (select course_no
     from course_taken ct2
     where ct1.sid = ct2.sid));

```

6. List 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.

```

select * from
(select instructorID as id, count(course_no) as N_courses
 from course_offered
 group by instructorID) as ic natural join instructor i

```