# CS1555 Recitation 6 - Solution

**Objective:** To practice SQL queries on PostgreSQL.

Before we start:

- Download the SQL script studentdb.sql through an SFTP client (such as FileZilla) from the machine "class3.cs.pitt.edu" at the directory:
  - `/afs/pitt.edu/home/r/a/raa88/public/studentdb.sql`

---

1. Find the address and phone number of the student whose student id is 123

```
SELECT address, phone
FROM student_dir
WHERE sid = 123;
```

---

2. List all the courses offered in 'Spring 21'.

```
SELECT distinct course_no
FROM course_taken
WHERE term = 'Spring 21';
```

---

3. 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;
```

---

4. List the sid(s) and gpa(s) of the students whose gpa(s) are greater than 3.7. List them in the descending order of the gpa(s).

```
SELECT sid, AVG(grade) as gpa
FROM course_taken
GROUP BY sid
HAVING AVG(grade) > 3.7
ORDER BY AVG(grade) DESC;
```

---

5. List the sid(s) of all the students and the number of courses they have taken.

```
SELECT sid, COUNT(DISTINCT course_no) AS num_courses
FROM course_taken
GROUP BY sid;
```

What if we want names too?

```
SELECT s.sid, s.name, COUNT (DISTINCT course_no) AS num_courses
FROM student s JOIN course_taken ct ON s.sid = ct.sid
GROUP BY s.sid, s.name;
```

How about another way?

```
SELECT sid, name, COUNT (DISTINCT course_no) AS num_courses
FROM student s NATURAL JOIN course_taken ct
GROUP BY sid, name;
```

---

6. Now insert a tuple into the Student table:

```
insert into student values (130, 'Peter', 1,'CS', '????');
```

Then run the query 5 again. How can we include this new student in the result, with 0 as the number of classes he has taken?

```
SELECT s.sid, s.name, COUNT (DISTINCT course_no) AS num_courses
FROM student s LEFT OUTER JOIN course_taken ct ON s.sid = ct.sid
GROUP BY s.sid, s.name;
```

---

7. For each course a student has repeated, list the sid and course number.

```
SELECT sid, course_no, COUNT(*)
FROM course_taken
GROUP BY sid, course_no
HAVING COUNT(*) > 1 ;
```

---

8. Assuming there is another table for outreach students who want to major in certificates:

```
CREATE TABLE student_outreach (
        sid int not null,
        name     varchar(15) not null,
        class    int,
        major    varchar (10),
        ssn      varchar (16) not null,
        CONSTRAINT PK_OUTREACH PRIMARY KEY(sid)
);
```

Insert the following student in the outreach table:

```
insert into student_outreach values ('130', 'Zach', 1,'CS',
'abcd');
```

List all the students in your organization?

```
(SELECT *
FROM student)
UNION
(SELECT *
FROM student_outreach);
```