CS1555 Recitation 7

Objective:

1. To practice more SQL queries on PostgreSQL.
2. To practice Views

PART 1:

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

```
create table student_outreach (
        sid integer not null,
        name     varchar(15) not null,
        class    integer,
        major    varchar (10),
        ssn      varchar (16) not null,
        constraint pk_stud_bad 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?

2. For each course a student from ‘CS’ major has repeated, list the studentID and course number.

---

3. List the SIDs and names of the students who have not taken the course “Web Applications”.

---

4. Find the top 3 students with the highest GPAs.

```
--note that if all the grades of a student is null, the average
(GPA) will be null. Ordering by GPA, those with null GPA will
appear first. Therefore, we specify a condition "avg(grade) is
not null" in order to eliminate those tuples with null GPA to
appear in the result set.
```

---

5. Find the SID and GPA of the top 1 student whose GPA is greater than the student whose SID is 123.

6. Rank the students (student ID and name) based on their GPA. Can we do something simpler?

PART 2:

1. Create a view called student_courses that lists the SIDs, student names, number of courses in the Course_taken table.

2. Create a materialized view called mv_student_courses that lists the SIDs, student names, number of courses in the Course_taken table.

3. Execute the following commands. Compare the query results and time used of the two select statements.

```
insert into course_taken (course_no, sid, term, grade)
values ('CS1555', '129','Fall 19', null);

--REFRESH MATERIALIZED VIEW mv_student_courses;
select * from mv_student_courses;
select * from student_courses;
```