**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;