CS1555 Recitation 6 - Solution

Objective:
1. To practice top-k and ranking queries in PostgreSQL.
2. To practice Views

PART 1:

Before we start:

- Download and run the below to build the database:
  - Recitation_6_PostgreSQL.sql
  - Recitation_6_Oracle.sql

1. List the student IDs and names of the 2 students with the lowest GPA.

```
-- Oracle and Postgres
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
order by GPA
FETCH FIRST 2 ROWS ONLY;
--or
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
order by GPA
limit 2;
```

```
--or
SELECT  x.SID, x.name,x.GPA
FROM (
select s.SID, s.name, avg(grade) as GPA,
ROW_NUMBER() OVER (ORDER BY avg(grade))
from course_taken ct
        join student s on ct.sid = s.sid
group by s.sid, s.name) x
WHERE ROW_NUMBER <=2;
-- In oracle
select * from (
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
order by GPA)
WHERE rownum <= 2;
```

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

```
select sid,     name,
     (1 + (select count(*)
          from (select s.sid, s.name, avg(grade) as gpa
               from course_taken ct
                    join student s on ct.sid = s.sid
                    where grade is not null
                    group by s.sid, s.name
                    having avg(grade) > i.gpa
               order by gpa) as e)
        ) as rank
from (select s.sid, s.name, avg(grade) as gpa
     from course_taken ct
          join student s on ct.sid = s.sid
       where grade is not null
       group by s.sid, s.name
       order by gpa) as i
order by rank;
```

```
-- Simplify
create or replace view student_gpa as
select s.sid, s.name, avg(grade) as gpa
from course_taken ct
      join student s on ct.sid = s.sid
where grade is not null
group by s.sid, s.name
order by gpa;

-- Now the query
select sid, name,
     (1 + (select count(*)
          from student_gpa as e
          where e.gpa > i.gpa)
        ) as rank
from student_gpa as i
order by rank;
```

PART 2:

Before we start:

- Download and run the below to build the database:
  - Recitation_6_PostgreSQL.sql
  - Recitation_6_Oracle.sql

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

```
create or replace view student_courses as
select s.sid, s.name, count(course_no) as num_courses
from student s, course_taken ct
where s.sid = ct.sid
group by s.sid, s.name;
```

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

```
drop materialized view if exists mv_student_courses;
create materialized view mv_student_courses
as
select s.sid, s.name, count(course_no) as num_courses
from student s, course_taken ct
where s.sid = ct.sid
group by s.sid, s.name;
```

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);
commit;

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

- The result from the materialized view is incorrect because the materialized view was not refreshed after the insert statement.

- The result from the view is correct because what a normal view does is rewriting the query. It does not store a snapshot of the query result like the materialized view.
- The running time of the materialized view is shorter, because it does not need to rewrite the query and run the rewritten query on the original Course_taken table.

4. Reset the database by running the `Recitation_6_PostgreSQL.sql` and recreate the views. Comment back the line beginning with "REFRESH" in the above commands and execute the commands. Compare the query results of the two select statements.

- The user can request a refresh of the materialized view by running the command:
  - `refresh materialized view <view_name>;`
- The result from the materialized view is correct this time, because we refreshed the materialized view before the select statement.