Write the following queries in SQL (of course, the queries must work on all data, not just the sample one):

1. Compute the GPA for the academic year 2015/16 for all students who have taken at least one class during this academic year. The answer should have attributes \( \{ \text{sid}, \text{gpa} \} \).

   ```sql
   select sid, avg(grade) as gpa
   from record
   where (year = 2015 and qtr = 'F') or (year = 2016 and qtr = 'W')
   or (year = 2016 and qtr = 'S')
   group by sid
   ```

2. List all pairs of students who have taken at least one course together in Fall 2016. The answer should have attributes \( \{ \text{sid1}, \text{sid2} \} \). Avoid listing pairs of the form \(<a, a>\), or listing both \(<a, b>\) and \(<b, a>\).

   ```sql
   select distinct x.sid as sid1, y.sid as sid2
   from record x, record y
   where x.sid < y.sid and x.qtr = 'F' and x.year = 2016 and y.qtr = 'F' and y.year = 2016
   and x.cid = y.cid
   ```

3. List, for each quarter, the number of courses with fewer than 5 students enrolled that quarter. The answer should have attributes \( \{ \text{qtr}, \text{year}, \text{num} \} \).

   ```sql
   select s.qtr, s.year, count(*) as num
   from (select qtr, year, cid from record
   group by qtr, year, cid
   having count(sid) < 5) s
   group by s.qtr, s.year;
   ```

4. Find the number of courses each student has taken in Fall 2016. If a student has taken no class that quarter, the number of classes should be zero. The answer should have attributes \( \{ \text{sid}, \text{num} \} \).
select sid, count(cid) as num
from record
where qtr = 'F' and year = 2016
group by sid
union
select r.sid, 0 as num
from record r
where not exists
(select * from record
where year = 2016 and qtr = 'F' and sid = r.sid);

5. List the students who have taken all prerequisites for CSE132X with a grade of 2 or higher. The answer should have one attribute sid. Note that, if CSE132X has no prerequisites, then all students should be in the answer. Provide two SQL queries, using nested sub-queries in different ways:

- with NOT IN tests only;
  select sid from student
  where sid not in
  (select s.sid from student s, prerequisite p
   where p.cid = 'CSE132X' and s.sid not in
   (select sid from record
    where cid = p.pre-cid and grade >= 2))

- with NOT EXISTS tests only;
  select s.sid from student s
  where not exists
  (select * from prerequisite p
   where p.cid = 'CSE132X' and not exists
   (select * from record
    where sid = s.sid and cid = p.pre-cid and grade >= 2));

6. Find the students whose quarterly GPA has gone up every quarter they have been enrolled (their GPA in each quarter is strictly higher than their GPA in previous quarters). Recall that students may have gaps in enrollment. **Hint:** It may be helpful to observe that the order of occurrence of quarters in a given calendar year is in reverse alphabetical...
order (‘W’ > ‘S’ > ‘F’). For example, the quarters occurring in 2016 are, in chronological order, W, S and F.

select distinct r.sid from record r
where not exists
(select * from record x, record y
 where x.sid = r.sid and y.sid = r.sid and
 (x.year < y.year or (x.year = y.year and x.qtr > y.qtr))
 and ((select AVG(grade) from record where sid = r.sid and qtr = x.qtr and year = x.year)
 >= (select AVG(grade) from record where sid = r.sid and qtr = y.qtr and year = y.year)))

7. Update relation record by swapping enrollments in CSE132A and CSE132B in Fall 2016, without explicitly naming the students involved. That is, all students enrolled in CSE132A should be dropped from that class and enrolled in CSE132B, and conversely. You may use several update commands if needed.

    insert into course values (’ZZZ’)
    update record
    set cid = ’ZZZ’
    where qtr = ’F’ and year = 2016 and cid = ’CSE132A’;
    update record
    set cid = ’CSE132A’
    where qtr = ’F’ and year = 2016 and cid = ’CSE132B’;
    update record
    set cid = ’CSE132B’
    where qtr = ’F’ and year = 2016 and cid = ’ZZZ’;
    delete from course where cid = ’ZZZ’