

- When describing the results of queries involving `UNION ALL`, `INTERSECT ALL`, and `EXCEPT ALL`, address (the meaning of) how many times a row could be repeated — don't just observe that there could be duplicates.
 - For #2, a student appears twice if they scored 95 or higher on both a (midterm) exam and a final. (Over all their courses — the exam and final don't have to have been in the same course.)
 - `INTERSECT ALL` can be awkward to explain but don't be too broad in the description. For #4 “the grading schemes Bell and Gassert use” is not correct — capture both that these are grading schemes used by both Bell and Gassert (the `INTERSECT`) and what the number of duplicates of a particular row reflects (the `ALL`).
- Be precise with language.
 - #3 (and others) selects from `GRADE`, which records student grades on assignments. The result involves students who have received a grade in a course, which is not necessarily the same as students who have taken a course.
 - #6 isn't just “the difference between the number of CPSC and MATH courses taught” — the difference is not reported directly (there's not a count reported) and only those teaching more CPSC than MATH are included.
 - #7 isn't just “instructors who taught in both terms” — it is specific course and instructor combinations which occurred in both terms.
 - In #8 and #9, note that the various sets are separate — the final with a score > 90 can be in any course, not necessarily a CPSC or MATH course.
- Note that #15 is very similar to #6... Approaches involving computing the actual difference and then picking out the cases where that difference is in the right direction (more CPSC than MATH) are valid, but it is worthwhile to recognize this as an `EXCEPT ALL` pattern.
- Both a `NOT EXISTS` subquery and the anti-join pattern (`LEFT JOIN ... IS NULL`) are appropriate for #20. A `NOT IN` subquery would fail, however, if there are any MATH courses where the instructor is `NULL` — `NOT IN` applied to a list containing `NULL` silently returns no rows. (That's why `NOT IN` wasn't listed as a potential subquery in the problem instructions.)