

- “More English-like” means to give the meaning rather than just writing what the SQL is doing in words. For example, “... where the term is equal to 'F22'” is just writing `term = 'F22'` in words and “where the title is not null” is just writing `title IS NOT NULL` in words. Instead, convey what the comparison means — “[sections] in Fall 2022” or “[sections] in the F22 term” and “where a title is defined”.
- Be precise with your language — expressing queries in natural language doesn’t mean being sloppy. Use the language of the schema and reflect the concepts there.
 - The grades database has both courses and sections (and these are different things!), so describing #1 as “courses offered in F22” is incorrect.
 - “Classes” could refer to both courses or sections. While “sections” is perhaps the more likely meaning, it is better to avoid the ambiguity.
 - #3 involves courses whose department is “CPSC” — while “CPSC instructors” could refer to “instructors who teach CPSC courses”, `INSTRUCTOR` also has a `dept` attribute and so “CPSC instructors” more naturally refers to that. “CPSC courses” is the correct description.
- In #2, the third column in the result is `COUNT(title)`, not `title` — this query isn’t returning the number and titles of courses in each department.
- Identify what is in the groups resulting from `GROUP BY` before expressing what an aggregation computes. In #4, the `FROM` has one row per section — sections are combined with the course the section is a section of and there is only one such course for each section (why?). Each group thus consists of the sections offered by a particular department in a particular term, so `SUM(COURSE.credits)` is the total credit value of the sections offered by a particular department in a particular term.
- The order of rows in a result is arbitrary unless there’s an `ORDER BY` — they might happen to come out alphabetically anyway, but it’s not a valid description of a query to say there’s a particular ordering of rows in the absence of `ORDER BY`.
- Be careful to account for everything in the `SELECT` clause in your description of the result e.g.

```
SELECT SECTION.instructor, SECTION.term,  
       AVG(GRADE.score/GRADECOMP.maxpts*100)
```

isn’t only returning instructors and terms.

- It is good practice to name result columns that include aggregation or other expressions — it makes the query clearer and allows the column to be referred

to should something be making use of the query's results.

- Be careful to use `DISTINCT` only when it is needed — while an unnecessary `DISTINCT` does not produce incorrect results, it does require additional duplicate-removal work by the query processor. It can also hide muddled thinking about the structure of the database.
- “There’s a better idiom for counting rows” — `COUNT(*)` specifically means “count the rows” while `COUNT(s)` means “count the number of non-NULL values in column `s`”. This works out to the samething when `s` is a primary key or `NOT NULL` column, but an extra level of thinking is required to realize that. If you intend to count rows, use `COUNT(*)`.