

CS 325 SQL and SQL*Plus Coding Standards so far

last modified: 2021-10-25

Here is an evolving set of CS 325 SQL and SQL*Plus Style Standards -- your CS 325 SQL and SQL*Plus code is expected to conform to these standards.

General

- When a SQL statement or SQL*Plus command is long (more than 80 characters), continue it on the next line(s) as needed, indenting the continuations by at least 3 spaces.
 - If a SQL statement clause is longer than one line, INDENT the continuation on the next line by at least three spaces (so it is clear which clause it "belongs" to).
- THOU SHALT NOT have more than one statement per line.
- THOU SHALT put a blank line **before** and **after** each SELECT statement, and usually before and after each multi-line SQL statement, for better readability.
 - And logically group SQL*Plus statements within a script as well.
- When in doubt, follow the style of posted class examples, AND ask me.

SELECT-specific

- THOU SHALT write the beginning of a SELECT statement's FROM clause on its OWN line.
- THOU SHALT write the beginning of a SELECT statement's WHERE clause on its OWN line.
- When a SELECT statement has N tables/relations in its FROM clause (OR for joins written using the ASCII JOIN syntax), THOU SHALT have (N-1) join conditions (unless one REALLY, TRULY wants a true Cartesian product, a rare occurrence!).
- THOU SHALT use mnemonic table aliases (d and e for tables dept and emp1, for example, not x and y or a and b - there should be some obvious relationship between the alias and the table name).
- THOU SHALT only use an ORDER BY clause for an outermost SELECT (not within any sub-select), and it shall be indented to make clear that it "belongs" to the outermost SELECT.
- THOU SHALT only use a GROUP BY clause when one has a good reason (usually one or more aggregate function computations that you wish done to those groups).

nested SELECTS/sub-selects

- THOU SHALT surround each nested select statement with a set of parentheses ().
 - (this is syntactically required in some contexts, such as after the IN operator. But it is apparently syntactically permitted to omit such parentheses when the nested select is, for example, one of the operands for union, intersect, or minus.)
 - (so, it is a class style standard that such nested select statements always be surrounded by parentheses, even when such parentheses are not required)

- THOU SHALT indent nested `select` statements by at least 3 spaces.
- ...except when the nested `select` is one of the operands for `union`, `intersect`, or `minus`).
That is,

- ...nested `selects` (sub-`selects`) should be indented within their outer `select`, **STILL** with their `from` and `where` clauses each on their own line -- for example, **both** of the following meet class style standards:

```
select empl_last_name, salary
from   empl
where  dept_num IN
      (select dept_num
       from   dept
       where  dept_loc = 'Dallas');
```

```
select empl_last_name, salary
from   empl
where  dept_num IN (select dept_num
                   from   dept
                   where  dept_loc = 'Dallas');
```

- But, this is not required for nested `selects` (sub-`selects`) that are operands to `union`, `intersect`, or `minus`:

```
(select empl_last_name, salary "Total Compensation"
 from   empl
 where  commission is null)
union
(select empl_last_name, salary + commission
 from   empl
 where  commission is not null)
order by "Total Compensation";
```

- When using `EXISTS` or `NOT EXISTS`
 - its sub-`select` argument is **EXPECTED** to include an appropriate **correlation condition**.
 - its sub-`select` argument is **EXPECTED** to project a literal (since these predicates only "care" if any rows exist in that sub-`select`'s results, **NOT** those rows' contents, and why bother doing much work projecting those contents, then?)

...and I reserve the option to add to this list over the course of the semester.