Style

3 min. read · View original

Here's a familiar simple example from the complex queries chapter:

```
select user_id,
  count(*) as how_many from bboard
where not exists (select 1 from
bboard_authorized_maintainers
bam where bam.user_id =
bboard.user_id) and
posting_time + 60 > sysdate group by user_id
order
by how_many desc;
```

Doesn't seem so simple, eh? How about if we rewrite it:

```
select user_id, count(*) as how_many
from bboard
where not exists (select 1 from
                  bboard_authorized_maintainers
bam
                  where bam.user_id =
bboard.user_id)
and posting_time + 60 > sysdate
group by user_id
order by how_many desc;
```

If your code isn't properly indented then you will never be able to debug it. How can we justify using the word "properly"? After all, the SQL parser doesn't take extra spaces or newlines into account.

> **Software is indented properly when the structure of the software is revealed *and* when the indentation style is familiar to a community of programmers.**

Rules for All Queries

If it fits on one line, it is okay to leave on one line:

```
select email from users where user_id = 34;
```

If it doesn't fit nicely on one line, give each clause a separate line:

```
select *
from news
where sysdate > expiration_date
and approved_p = 't'
order by release_date desc, creation_date desc
```

If the stuff for a particular clause won't fit on one line, put a newline immediately after the keyword that opens the clause. Then indent the items underneath. Here's an example from the ArsDigita Community System's static .html page administration section. We're querying the `static_pages` table, which holds a copy of any .html files in the Unix file system:

```
select
  to_char(count(*),'999G999G999G999G999') as
n_pages,

to_char(sum(dbms_lob.getlength(page_body)),'999G999G999G999G999')
as n_bytes
from static_pages;
```

In this query, there are two items in the select list, a count of all the rows and a sum of the bytes in the `page_body` column (of type CLOB, hence the requirement to use `dbms_lob.getlength` rather than simply `length`). We want Oracle to format these numbers with separation characters between every three digits. For this, we have to use the `to_char` function and a mask of `'999G999G999G999G999'` (the "G" tells Oracle to use the appropriate character depending on the country where it is installed, e.g., comma in the U.S. and period in Europe). Then we have to give the results correlation names so that they will

be easy to use as Tcl variables. By the time we're done with all of this, it would be confusing to put both items on the same line.

Here's another example, this time from the top-level comment administation page for the ArsDigita Community System. We're going to get back a single row with a count of each type of user-submitted comment:

```
select
  count(*) as n_total,

sum(decode(comment_type,'alternative_perspective',1,0))
as n_alternative_perspectives,
  sum(decode(comment_type,'rating',1,0)) as
n_ratings,

sum(decode(comment_type,'unanswered_question',1,0))
as n_unanswered_questions,

sum(decode(comment_type,'private_message_to_page_authors',1,0))
as n_private_messages
from comments
```

Notice the use of `sum(decode` to count the number of each type of comment. This gives us similar information to what we'd get from a GROUP BY, but we get a sum total as well as category totals. Also, the numbers come out with the column names of our choice. Of course, this kind of query only works when you know in advance the possible values of `comment_type`.

Rules for GROUP BY queries

When you're doing a GROUP BY, put the columns that determine the group identity first in the select list. Put the aggregate columns that compute a function for that group afterwards:

```
select links.user_id, first_names, last_name,
count(links.page_id) as n_links
```

```
from links, users
where links.user_id = users.user_id
group by links.user_id, first_names, last_name
order by n_links desc
```

Next: procedural

---

philg@mit.edu

Reader's Comments

The where clause is in two lines in the example above though the text suggests one line:
"If it doesn't fit nicely on one line, give each clause a separate line:
select *
from news
where sysdate > expiration_date
and approved_p = 't'
order by release_date desc, creation_date desc"
In this case, one line would be the better, of course.

-- Peter Tury, June 12, 2002

Add a comment | Add a link