TWENTY YEARS OF PERFORMANCE TUNING
by Chris Lawson

TIP #9
Consider Query Subfactoring

Have you ever tried to analyze a complicated sql, but there were so many clauses or inline views, that you hardly knew where to start? Well, I've been saddled with that type of code many times. But there is a better alternative. It's called "Query Subfactoring," and it's a great way to make code more manageable.

Some folks call this the "With" syntax. Using this feature makes your code much easier to follow, and much easier to debug--both functionality-wise and performance-wise. Use of a lot of inline-views can make the code nearly impossible to understand.

Here's what it looks like--note that each "view" is enclosed within parentheses:

```
With View1 as (Select * from Table1 Where Col1 = 'ABC'),
     View2 as (Select * from Table2 Where Col1 = 'DEF')
Select Count(*) from View1, View2 Where View1.Col1 = View2.Col1;
```

Using this approach, you will find it much easier to tune complicate sql. When this format, you can simply tune each part, piece by piece. So, for the above example, I would get the "View1" running smoothly, then "View2." When all the preliminary parts are running well, you can then concentrate on tuning the main body. It's often helpful to use the MATERIALIZE hint in the preliminary views. This way, the optimizer won't try to change execution plans at the end, since the early views are already "materialized" behind the scenes.

Chris Lawson is the author of *The Art & Science of Oracle Performance Tuning*, and *Snappy Interviews: 100 Questions to Ask Oracle DBAs*. When he's not solving performance problems, Chris is an avid hiker, runner, chorister, Amazon reviewer, and geocacher. Chris writes using the pen name, "*Bassocantor*."

Twenty Years of Performance Tuning is a series of tips based on the author's experience solving performance problems over the last 20 years.