

NoCOUG presents  
**Jonathan Lewis Training Days**

Register at [www.speak-tech.com](http://www.speak-tech.com)

**TROUBLESHOOTING AND TUNING**

**November 11, 2008**

**Troubleshooting or Tuning:** What's the difference? What are the strategies? Why tuning is hard but troubleshooting is easy. Key targets, indicators, and mechanisms for producing a well-tuned system on day one. Strategies for dealing with badly performing systems after go-live.

**Frequently Occurring Problems:** Some of the most commonly occurring issues that affect performance after a system has gone into production. Methods for spotting them, measuring the impact, and dealing with the cost/risk/benefit triangle involved in fixing them. Getting into the habit of pre-emptive analysis and proactive fixing.

**Quick Fixes:** Methods, workarounds, dirty tricks, and parameters for dealing with classic performance problems when the system is in production. There really aren't many quick fixes that can be applied across the board—each one needs careful examination of costs, risk, and benefits. In this session we consider some of the options that are most likely to be worthwhile.

**V\$ and X\$:** It's a good idea to be familiar with just a few of the dynamic performance views—and there are a couple of items still hidden away in the X\$ objects that can add a little value. This session will describe the views that are most commonly of use, and explain the meaning of some of the more useful items.

**WRITING OPTIMAL SQL**

**November 12, 2008**

**Background:** We will examine the need for a global viewpoint and a general strategy for minimizing work while recognizing that there will always be a conflict between local and global optimization. A quick review of the use of indexes, including some of the less-common uses and common errors in index design. Closing with the need to be able to investigate and understand the data and business requirements.

**Basic Practices:** How to approach the task of translating business requirements into SQL. The need for clarity in code, with a suggested set of standards for presentation to improve ease of understanding. An introduction to the essential strategy for writing SQL that gets its result by doing the minimum work, and a model for investigating and re-engineering poorly performing statements.

**Sundry Methods:** A few examples demonstrating classic problems and their solutions that use some of the less-well-known features of SQL. A brief discussion of the mechanics (and costing) of nested loop and hash joins. Warnings about the abuse of views and the problems of predicates. And a few closing comments about tuning distributed queries.

**More Methods:** Fixing the code without touching the code—the latest option. Problems and options with knowing internal mechanisms and taking advantage of them. A little time with OEM, and a look at the threats of hints and how to use them properly. ▲

**EARLY-BIRD PRICING (UNTIL SEPTEMBER 19)**

**One day:** \$350 for non-members,  
\$300 for NoCOUG members

**Both days:** \$650 for non-members,  
\$550 for NoCOUG members

**REGULAR PRICING**

**One day:** \$375 for non-members,  
\$325 for NoCOUG members

**Both days:** \$700 for non-members,  
\$600 for NoCOUG members