



Jonathan Lewis

Jonathan Lewis is a well-known figure in the Oracle world with more than 22 years experience of using the database (and several years on other products). He has published two books, contributed to three others, runs a couple of websites and contributes fairly regularly to newsgroups, forums, User Group magazines, and speaking events around the world. Jonathan has been self-employed for most of his time in the IT industry. He specialises in short-term assignments, typically of a design, review, or trouble-shooting nature.

Optimising Oracle

ABOUT THIS SEMINAR

When and where: November 1-2, 2011 Crowne Plaza Helsinki, Mannerheimintie 50, Finland

Registration: by email to JonathanLewis@miracleoy.fi, no later than October the 25th.

Fee: 1200e+VAT, members of Oracle User Group Finland 1000e+VAT

Organizer: Miracle Finland Oy



Optimising Oracle – Foundations: Agenda Day 1

This tutorial is a compilation from the other tutorial sessions. The first day is mainly about how the Cost Based Optimizer does it's arithmetic and how you can compare your data with the optimizer's "picture" of your data. The last session of the day will be about finding and interpreting execution plans.

Session 1 1.5 hours	CBO session 1: Why isn't Oracle using my index? In this session we aim to acquire a visual, rather than highly mathematical, understanding of when Oracle will ignore an index. We start with a discussion of the strategic direction that we should be going with the cost based optimizer and then examine a simple example to pin-point the traditional reasons why the cost based optimizer in versions prior to 9.0 could so easily produce inappropriate execution plans. We examine a couple of commonly used tuning parameters which give Oracle better information about our system and allow it to identify the appropriate execution path more frequently, identifying the risks of using these parameters too extravagantly, and end with a demonstration of how using CPU costing solves the traditional problems with much less risk of error.
------------------------	--

Break – coffee and informal discussion: 30 minutes

Session 2 1.5 hours	CBO session 2: Mechanisms of Joins We start by asking why it is so important to examine and understand join mechanisms and examine some of the transformations that convert complex queries to simple join forms. Then we look at details (and anomalies) of the nested loop, sort-merge, and hash joins; identifying strengths, weaknesses, and costs of each in turn. In the case of sort-merge and hash joins, we examine trace events, and their output, that allow us to investigate what is happening when response times become extreme.
------------------------	---

Break – Lunch and informal discussion: 1 hour

Session 3 1.5 hours	CBO session 3: Selectivity, Joins, and hints In this session we move on from the simple example of the first session to investigate what happens in the more general cases of using an index. We extend the arithmetic to
------------------------	---

MIRACLE

the calculations involved in joining tables, and note some of the anomalies and problems that joins can cause. Finally we ask the question "What is a hint?" and try to answer the question by examining what really happens inside the optimizer when we start adding hints to our SQL.

Break – coffee and informal discussion: 30 minutes

Explain Plan session 1 and 2: Finding and Reading Plans

Session 4
1.5 hours

There are several different methods for generating, or examining, execution paths, from the simple autotrace in SQL*Plus through to interrogating the dynamic performance views such as v\$sql_plan_statistics_all in great detail. The different methods have their own strengths and weaknesses, and these vary with version of Oracle. In this first session, we discuss the different options, what's visible in them, and to choose the most appropriate one for the task in hand.

Once you've got an execution plan, how do you decide what it means? An often-quoted clue is that 'it starts near the top and over to the right'. This hint is a consequence of a simple algorithm for reading execution plans – no matter how complex – in a fashion that can be described in just two steps.

Optimising Oracle – Foundations: Agenda Day 2

We start with a session on the problems that we can run into when looking for, or interpreting, execution plans, followed by some time looking at indexing strategies, the way Oracle can use indexes, and some common misunderstandings about the strengths and weaknesses of Oracle's B*Tree implementation.

We end the day by revisiting the optimizer to ask (and answer) the question – how do we make sure the optimizer gets a good picture of what our data looks like.

Explain plan session 3: Problems with plans

Session 1
1.5 hours

Although there are several ways of acquiring execution plans, none of them is perfect. Some options simply omit critical details; some options are unable of guaranteeing the truth. In this session, we look at the various omissions and errors that can make you jump to the wrong conclusion when you are trying to understand how a query might run.

Break – coffee and informal discussion: 30 minutes



Session 2 1.5 hours	Indexes session 2: The uses of indexes How many different ways can Oracle take advantage of an index. We have primary keys and unique keys – do we need unique indexes to enforce them. We may have problems with foreign keys if we don't have related indexes. We need optimal access paths – but will indexing help and what will it cost? When will Oracle use an index to accelerate a query? Can we assist performance by understanding the mechanisms.
Break – Lunch and informal discussion: 1 hour	
Session 3 1.5 hours	Optimal SQL: 2 / 3 – Methods and practices for optimal SQL. 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 badly performing statements. A few examples demonstrating classic problems, and solutions that take advantage of some of the slightly exotic options of SQL and Oracle's available features. And a few closing comments about the need for the front-end code to co-operate with the database engine in certain circumstances.
Break – coffee and informal discussion: 30 minutes	
Session 4 1.5 hours	CBO session 4: Maximising the truth If there is any information we can give the optimizer about our data, we should do so; otherwise it will be less able to produce the appropriate execution plan. We start this session by showing how a little extra information can help the optimizer find new execution paths. Then we examine the optimizer's dependence on statistics, and its need to get a correct numeric representation of your data. We move on to see how strange data patterns, and bad database design, can stop the optimizer from choosing a sensible execution path, and end with a couple of techniques for encouraging the optimizer to do what we want.