ORACLE UNIVERSITY

Datenbank - Workshops mit Jonathan Lewis

Nutzen Sie die Chance !
Treffen Sie 15 Jahre Oracle Datenbankerfahrung.

03. – 07. Juli 2006, Hotel Holiday Inn in Düsseldorf

Jonathan Lewis has 18 years database software experience, 15 years using Oracle. He has worked as an independent Design and Trouble-shooting Consultant. Jonathan's systems experience ranges from a single NT up to 128-node Pyramid RM1000, databases ranging from 50 Mb to 4 Tb with transaction rates from 60 a day to 4M an hour. He specialises in understanding how the database engine works, and can advise how to make systems run faster, cost-effectively. He presents to UK Oracle User Group and the Danish Database Forum. He is also the author of Practical Oracle8*i* – Building Efficient Databases.

3 Seminare buchen und Jonathan Lewis persönlich kennenlernen.
Fragen Sie nach dem Backstage-Pass!
Die Teilnehmerzahl für diesen exklusiven Backstage-Event ist limitiert.

Tagungssprache ist Englisch.

DBA-Seminars
These DBA-Seminars are designed to help DBAs make better use of their time when dealing with production database systems.

Developer-Seminar
Focused on the programmers needs of the Oracle RDBMS.

DBA-Topics:
- Cost Based Optimisation 03. Juli 06
- Indexing Strategies 04. Juli 06
- Trouble-shooting and Tuning 05. Juli 06
- Explain Plan 06. Juli 06

Developer-Seminar:
- Oracle Developer 07. Juli 06

Tagespreis: EUR 570,00 + MwSt.

Jedes Seminar kann einzeln gebucht werden!

Buchen Sie jetzt. Es steht nur eine begrenzte Anzahl von Plätzen zur Verfügung!

Bei Fragen wählen Sie 0180/2000-526 (0,06 EUR/Anruf).

Nachfolgend die Seminarinhalte:

July 03, 2006
Seminartopics: Understanding and Assisting the Cost Based Optimizer

Why isn't Oracle using my index?

Session 1
1.5 hours

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 had could so easily produce inappropriate execution plans. We introduces a couple of ***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

Mechanisms of Joins

Session 2
1.5 hours

We start by asking why it is so important to examine and understand ***join mechanisms*** and examine some of the transformations that convert sub-queries to joins. 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

Selectivity, Joins, and hints

Session 3
1.5 hours

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 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

Maximising the Truth

Session 4
1.5 hours

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. 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 forcing the optimizer to do what we want.

July 04, 2006
Seminartopics: Indexing Strategies

Session 1 1.5 hours	**The Structure of Indexes** What is a ***B-tree*** index, what is a ***Bitmap*** index and why is it so different? How does Oracle build indexes? Are there differences in ***reverse*** indexes, ***function-based*** indexes, ***cluster*** indexes, ***global partitioned*** indexes, ***indexed organized*** tables, and ***secondary*** indexes.With the aid of a few block dumps and carefully monitored test results we examine the way that different types of indexes are built and behave..
Break – coffee and informal discussion: 30 minutes	
Session 2 1.5 hours	**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 fast 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	**Popular misconceptions** There are several surprising ideas floating around the Internet about how Oracle indexes work. Sometimes these ideas are irrelevant and have no use or impact. Sometimes they have a serious impact on your effectiveness. They may cause you to do work that need not, or should not, be done. They may introduce ***down-time*** that you do not need. They may make you avoid a feature that is really the perfect feature for your application. This session examines some of the more popular and potentially ***damaging misconceptions*** about indexes.
Break – coffee and informal discussion: 30 minutes	
Session 4 1.5 hours	**Maximum benefit, minimum cost** We all know that indexes are supposed to make things go faster - but sometimes we forget that they have ***maintenance*** costs - of two types. How do you identify the ***optimum set*** of indexes for your application? How do you identify ***redundant indexes?*** How do you check whether or not an index has become ***inefficient*** and needs to be rebuilt - and how do you ***rebuild*** it most cost-effectively? This session discusses strategies for ensuring that you don't waste effort supporting indexes that shouldn't exist, and don't waste valuable ***batch time*** rebuilding indexes that don't need it.

July 05, 2006
Seminartopics: Trouble Shooting and Tuning

Trouble-shooting or Tuning

Session 1
1.5 hours

What's the difference? What are the strategies. Why tuning is hard but trouble-shooting is easy. Key targets, indicators and mechanism for producing a well-tuned system on day one. Strategies for dealing with badly performing systems after go-live.

Break – coffee and informal discussion: 30 minutes

Frequently Occurring Problems

Session 2
1.5 hours

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 pro-active fixing.

Break – Lunch and informal discussion: 1 hour

Quick Fixes

Session 3
1.5 hours

Methods, workaround and dirty tricks 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.

Break – coffee and informal discussion: 30 minutes

V$ and X$ -

Session 4
1.5 hours

It's a good idea to be familiar with just a few of the dynamic performance views - and there are a couple of items in the 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 obscurely named, but potentially useful items.

July 06, 2006
Seminartopics: Explain Plan

How to find execution paths.

Session 1
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 in great detail. The different methods have their own strengths and weaknesess, 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.

Break – coffee and informal discussion: 30 minutes

Intepreting Plans

Session 2
1.5 hours

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.

Break – Lunch and informal discussion: 1 hour

Problems with Plans

Session 3
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

Advanced Features

Session 4
1.5 hours

Once you have mastered the standard features of execution plans, you can easily extend your understanding to any new feature that Oracle throws at you. In this session, we take a look some of the less common features currently in use, such as parallel query, partitioned tables, and distributed queries, and then move on to a general approach to coping with plans that contain structures that you haven't seen before.

July 07, 2006
Seminartopics: The one-day Developer Course

Session 1 1.5 hours	**Overheads** A basic introduction to the costs of the infrastructure that makes Oracle what it is. Topics include connection costs, network traffic, parsing and optimising costs, undo generation and use, and redo.
Break – coffee and informal discussion: 30 minutes	
Session 2 1.5 hours	**Coding Strategies** There are good ways to code and bad ways to code. This session demonstrates the problems with the bad coding strategies and shows how much difference it makes when you do it right.
Break – Lunch and informal discussion: 1 hour	
Session 3 1.5 hours	**Indexing** A description of indexes, largely devoted to B-tree indexes, talking about structure, how they work, and the paths Oracle is able to take through them. Indications of how to get the most out of indexes, and minimise the cost of indexing, with warnings about side effects of the different strategies for making an index more useful.
Break – coffee and informal discussion: 30 minutes	
Session 4 1.5 hours	**Execution paths** In this session, we take an introductory view on how the cost based optimizer works, including a brief overview of the different join mechanisms. Then we examine execution paths and the developer-oriented ways of acquiring them. Finally we learn how to read and interpret execution paths.