

ORACLE®

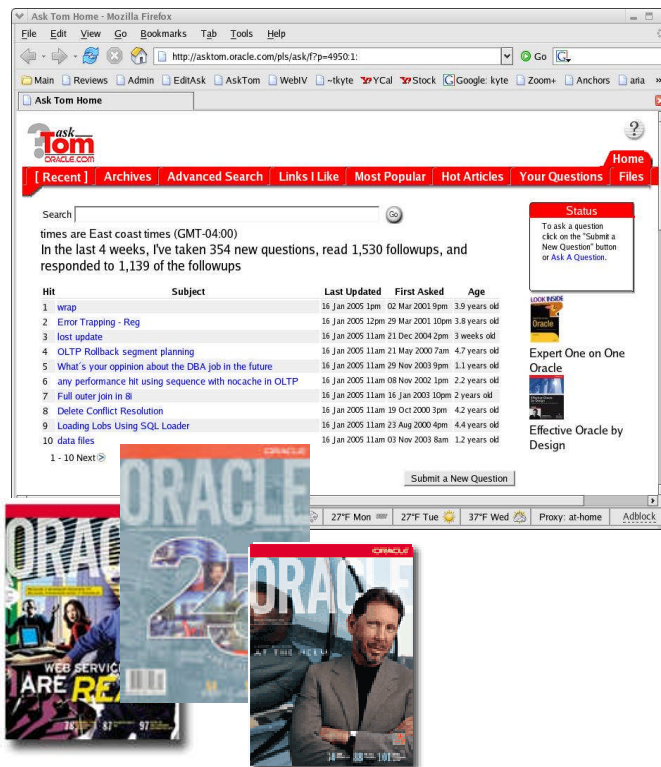


The Best Way...

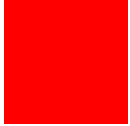
Thomas Kyte
<http://asktom.oracle.com/>

What we know, shapes how we do things...

Who am I

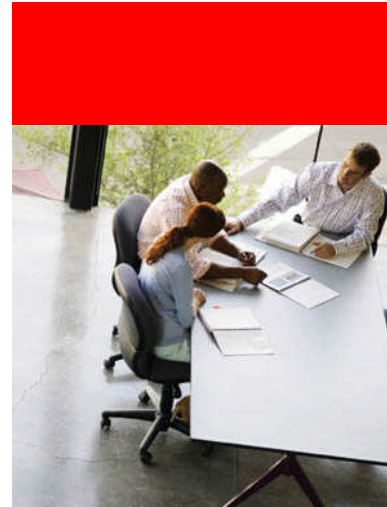


- Been with Oracle since 1993
- User of Oracle since 1987
- The “Tom” behind AskTom in Oracle Magazine
www.oracle.com/oramag
- Expert Oracle Database Architecture
- Effective Oracle by Design
- Expert One on One Oracle
- Beginning Oracle



**New Ideas – Almost always
come from those uncommitted
to the status quo – from
outsiders**

*People are committed to whatever has worked for
them in the past.*





Best Practices defined –

Consensus of expert opinions, based on actual customer experiences in practice.

Lessons learned.

Proven practices associated with a particular usage profile.

Baseline configuration rules - prerequisite to tuning.



Sounds all good...

Best Practices – It is easy with Best Practices to forget that once a practice has been branded as "Best", that it may **represent certain tradeoffs** and may **involve noteworthy downside potential**. It is also easy to forget the *context* for which any given practice was promoted as "Best", and therefore **apply it in some inappropriate context**.

– *Bob Sneed, Sun Microsystems*





Prescribing best practice principles for programming any 3GL is phenomenally difficult. One of the hardest challenges is the safety of the assumption that the reader starts out with these qualities

- **Required for the "Right to Play" (RTP) (Self) Protection** to develop the three pillars of good code, manage the test specifications and aggressive threats that arise along the way?)

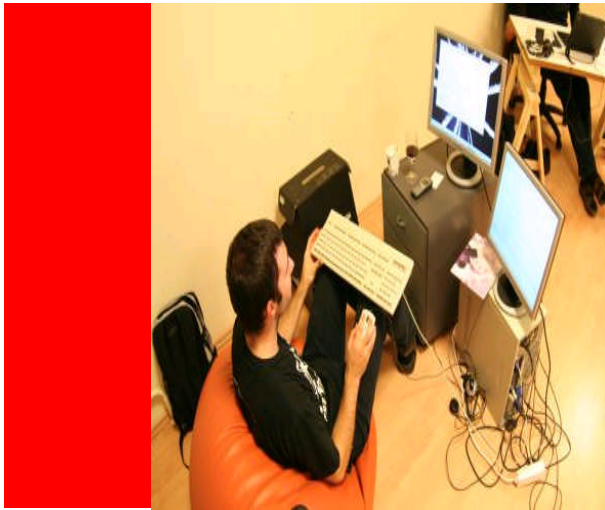


**“What is the best way..?” –
Questions that begin and end
with that can drive you nuts.**



*If there was a universal best way to do something,
we would not have implemented the other ways...*

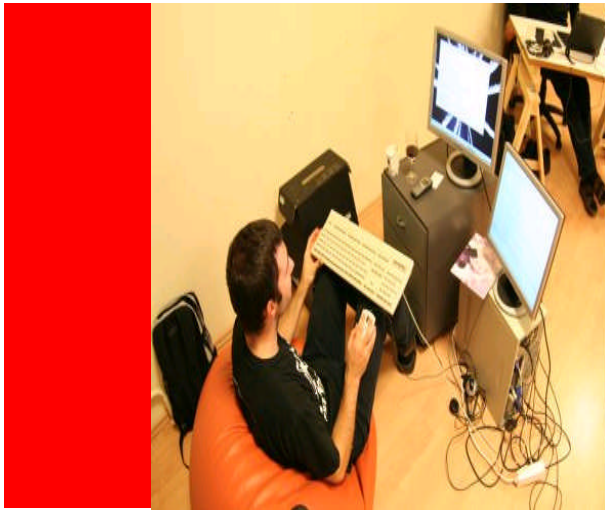
AskTom “What is The best way?”



Is there
a *best* way
to do something – every
time?

```
select *  
  from t1, t2  
 where t1. id = t2. id  
       and t1.small_distinct = :x
```

- T1 is large, where small_distinct = :x returns much of the table
- T2 is large



Is there
a *best* way
to do something – every
time?

```
select * from t1, t2
where t1. id = t2. Id and t1.small_distinct = :x
```

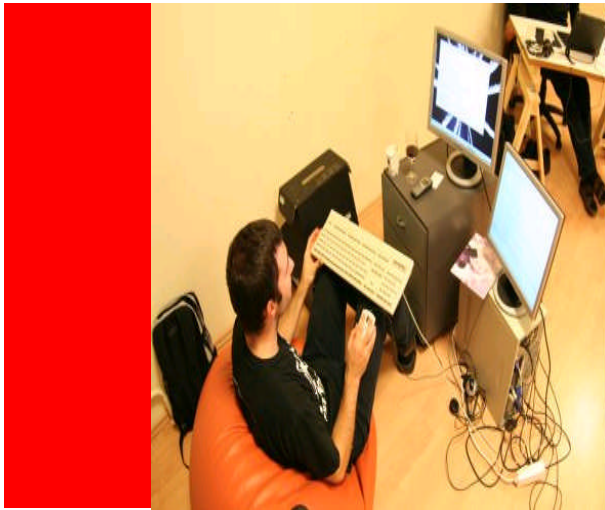
HASH JOIN

TABLE ACCESS FULL T1
TABLE ACCESS FULL T2

SELECT STATEMENT

NESTED LOOPS

TABLE ACCESS BY INDEX ROWID(T1)
INDEX RANGE SCAN T1_IDX
TABLE ACCESS BY INDEX ROWID(T2)
INDEX UNIQUE SCAN T2_PK



Is there
a *best* way
to do something – every
time?

HASH JOIN

TABLE ACCESS FULL T1
TABLE ACCESS FULL T2

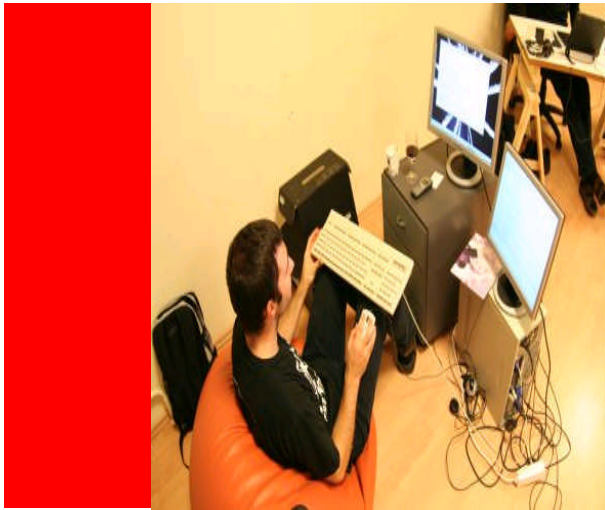
SELECT STATEMENT

NESTED LOOPS

TABLE ACCESS BY INDEX ROWID(T1)
INDEX RANGE SCAN T1_IDX
TABLE ACCESS BY INDEX ROWID(T2)
INDEX UNIQUE SCAN T2_PK

call	count	cpu	elapsed	disk	query
Fetch	35227	5.63	9.32	23380	59350
Fetch	35227	912.07	3440.70	1154555	121367981

ORACLE®



Is there
a *best* way
to do something – every
time?

HASH JOIN

TABLE ACCESS FULL T1
TABLE ACCESS FULL T2

SELECT STATEMENT

NESTED LOOPS

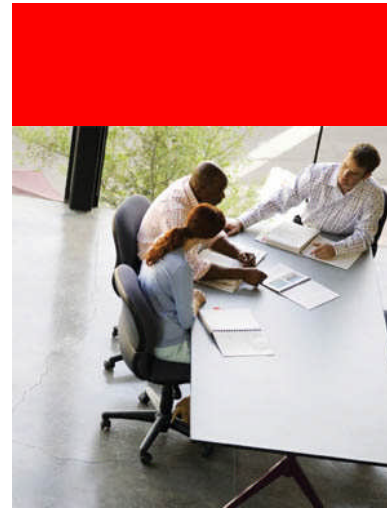
TABLE ACCESS BY INDEX ROWID(T1)
INDEX RANGE SCAN T1_IDX
TABLE ACCESS BY INDEX ROWID(T2)
INDEX UNIQUE SCAN T2_PK


call	count	cpu	elapsed	disk	query
Fetch	1	4.55	5.16	12152	12456
Fetch	1	0.05	0.09	12	15

ORACLE

It takes a context – It is also easy to forget the *context* for which any given practice was promoted as "Best", and therefore apply it in *some inappropriate context*

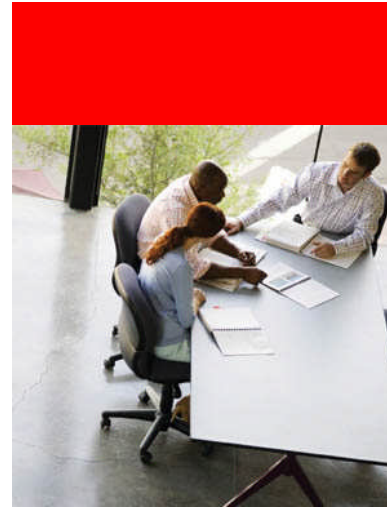
Indexes are 'best', everyone knows that...

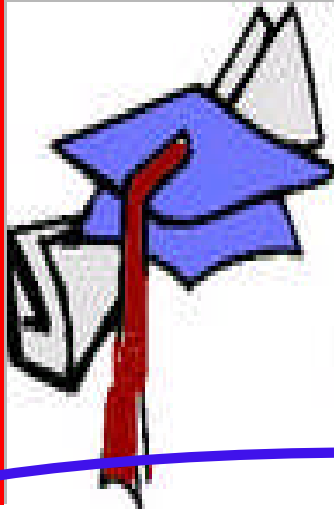




It takes understanding too –
You need to take the facts,
coupled with your knowledge,

How do I tune with tkprof...





What can
you do with this
information?

```
select count(subobject_name) from big_table.big_table
```

```
COUNT(SUBOBJECT_NAME)
```

```
-----
```

```
688250
```

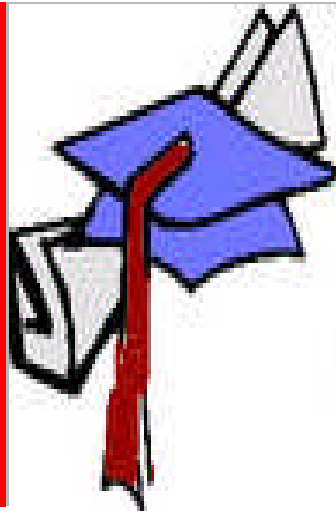
call	count	cpu	elapsed	disk	query
total	4	99.36	262.11	1840758	1840800

Rows Row Source Operation

1	SORT AGGREGATE (cr=1840797 pr=1840758 pw=0 time=262104893 us)
128000000	TABLE ACCESS FULL BIG TABLE (cr=1840797 pr=1840758 pw=0 time=384004887 us)

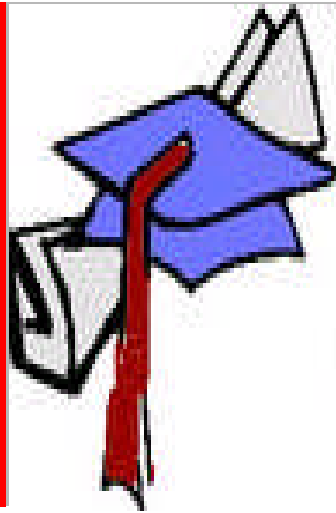
Event waited on	Times	Max. Wait	Total Waited
db file scattered read	14425	0.22	195.87

ORACLE



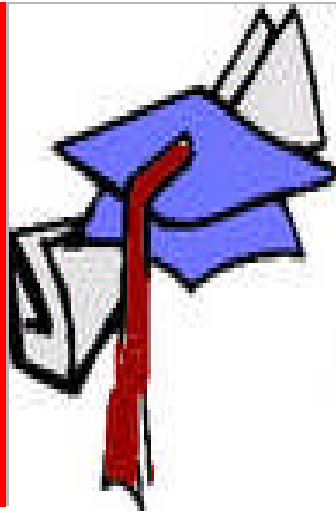
First, there
are a bunch of
facts...

- Query took a long time – *if we make it fast...*
- We did a ton of physical IO – *and that is slow*
- We did a ton of logical IO – *and that is not 'free'*
- There is a big difference between elapsed and cpu – *we were waiting for something*
- We can see our query and plan – *we know the answer to the query*



**There are
things we have
knowledge of...**

- We know the data (it is ours after all)
- How Oracle works (hopefully!)



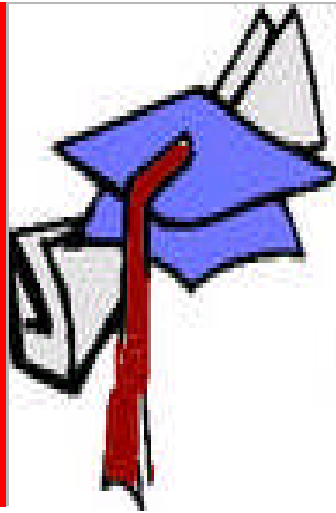
**What are
some obvious
things to think
about here?**

- We needed a very small subset of the rows – 700k out of 128m
- The table looks well packed – simple math, divide IO's (cr=1,840,797) by rows (128,000,000), about 70 rows/block and given we know the average row width (it is our data after all...) that sounds nicely packed
 - What can we rule out now? Shrink and Rebuild...



What are Some possible options?

- Make full scan faster
 - Maybe by compressing the table
 - Maybe by including subobject_name in some index (to avoid the table)
- Remove Full Scan
 - We are interested in only 0.6% of the data
 - Maybe a new index would help
- Don't do it or do it differently...



**In Real
Life it
Will be more
*Complex...***

- It will be more complex in general
- But the process is the same
 - Get *facts*
 - *Infer* more facts
 - Build your *context!*
 - Rule things *out*
 - Ruling something out is as good as ruling something in
 - Many best practices will fall by the wayside here

Educated Incapacity – A barrier to creative ideas can be experience, ‘the best way’



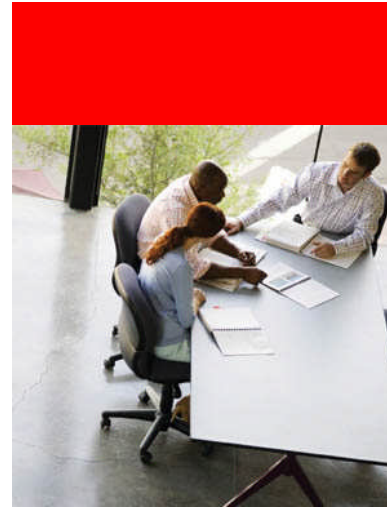
"This 'telephone' has too many shortcomings to be seriously considered as a means of communication. The device is inherently of no value to us." – Western Union internal memo, 1876.

"The concept is interesting and well-formed, but in order to earn better than a 'C,' the idea must be feasible." – Yale University management professor in response to Fred Smith's paper proposing reliable overnight delivery service. Smith went on to found Federal Express Corp.

Although experience is often valuable, it can be a liability in a search for creative ideas.



So, What is the point?



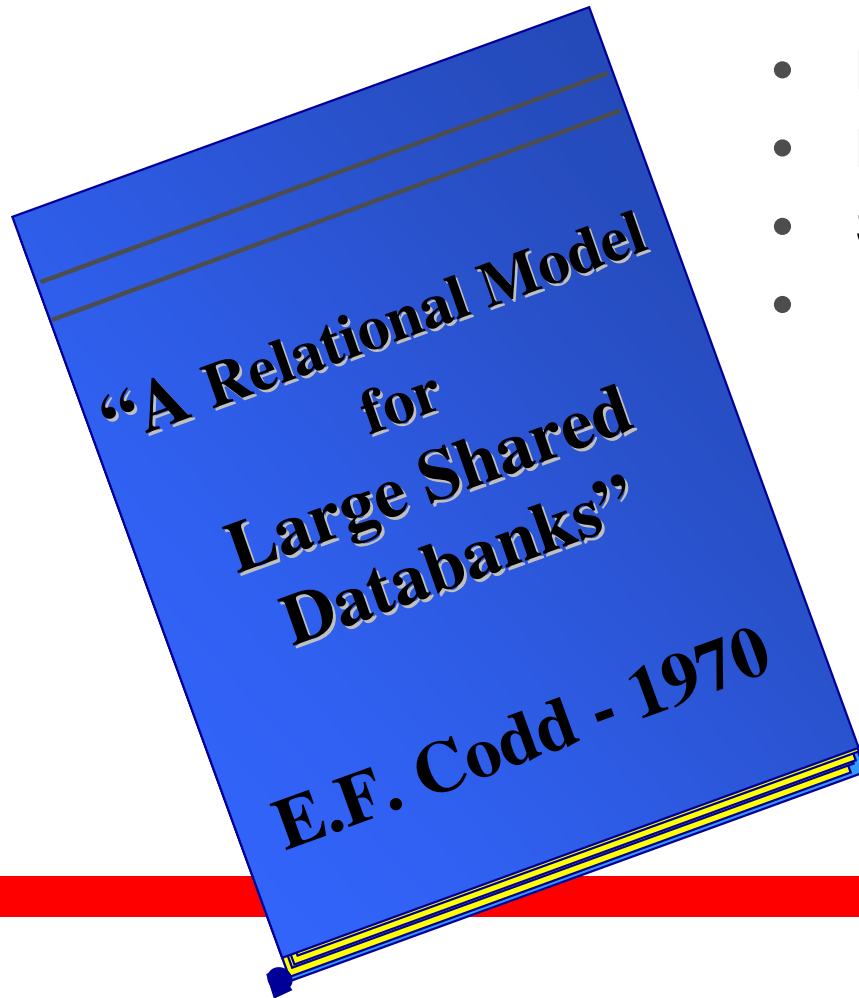


The Beginning...

**When you were 12,
did you know what
you wanted to be
when you grew up?**



The Beginning...



- Data Model with Structure
- Data Independent of Code
- Set-oriented
- 1977 the work begins



Continuous Rethinking



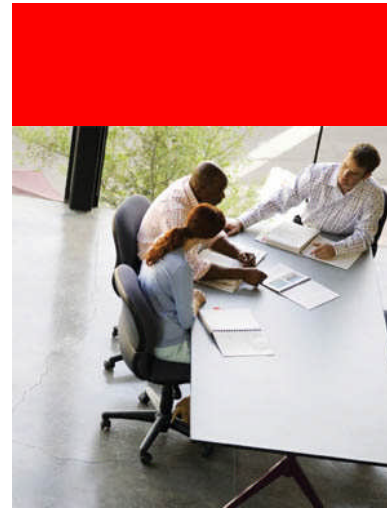


Continuous Change



**Learn a new language – else
everything will look like a nail.**

***C, C++, PL/I, Rexx, Exec, JCL, SAS, Pascal,
Cobol, Java, Ada, PL/SQL, T-SQL, Prolog,
Lisp, Scheme, Various Assemblers, many SQL
dialects, many scripting languages, ...***





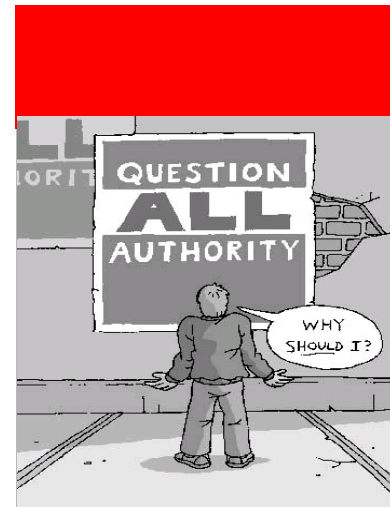
Don't tune a query – tune a process, an algorithm, the entire approach.

Don't fall into the sunk cost theory...



**Always Question Everything –
in a non-annoying way of
course!**

Question Authority...



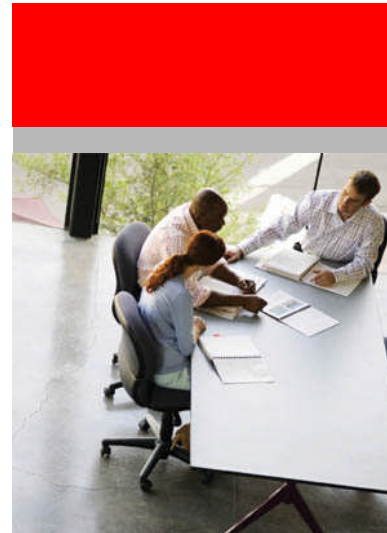
Never stop testing ideas –
Things change, I still
remember how it worked in
version 5.

The tuning techniques (best practices)
of yesterday are today's performance /
maintenance headaches...



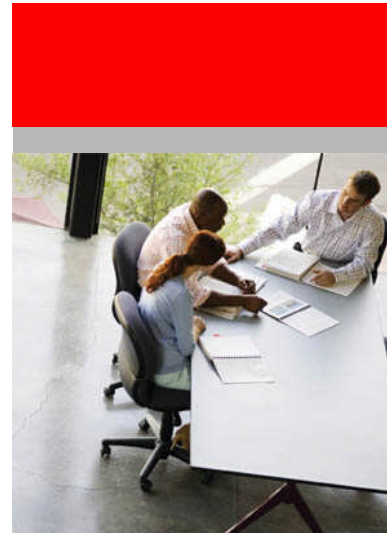
Don't rule anything out until you try it – How do you *know* it won't work unless you try it. Just because it didn't work in a different *context* doesn't mean it won't work now.

You can't use any feature until it is 3 releases old – this is software, not fine wine...



The simple solution is usually right – There are many ways to build an application. Not all of them need 14 tiers. Many applications are quite simple – there is room for many tools.

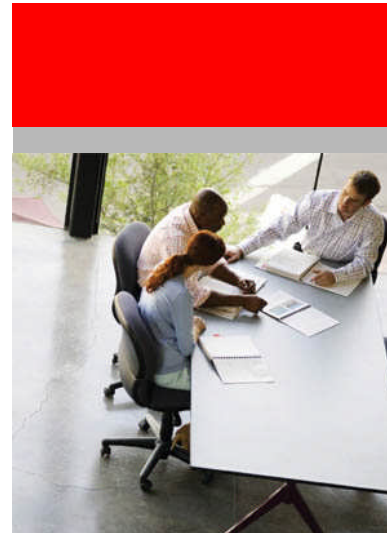
We get caught up in the novelty of the idea that we ignore how practical it really is...

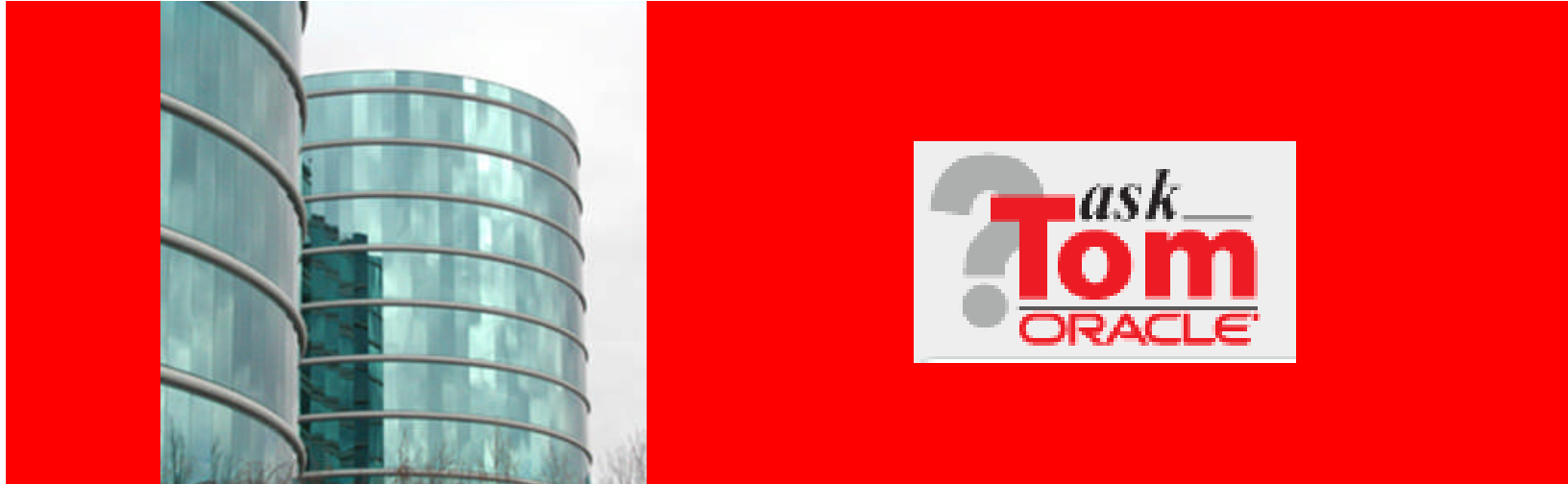




**Collaborate – Participate,
Network, Exchange ideas.**

*I learn something new every day about
Oracle – from the questions I get about
Oracle...*





ORACLE®



The Best Way...

Thomas Kyte
<http://asktom.oracle.com/>

What we know, shapes how we do things...



Q&A

- 1) What is the optimum number of nodes for RAC. Is it better to use more nodes with low end servers or have fewer nodes using high end servers.
- 2) I want to use event tracing to create resource profiles for database tuning. Is there a free tool that reads trace files (besides tkprof) to create resource profiles? See Cary Milsap's book on performance tuning.
- 3) We hear plenty about "licensed cost" items for Oracle; are there any new exciting (or overlooked) 'extra' programs that don't have marketing clout that we should know about? (Apex was a good example.)