Indexing and CBO

Help the system do its job

Piet de Visser

The <u>Simple</u> Oracle DBA

Commit Your ORACLE Knowledge

Keywords: Simplicity. Is that CBO looking at us… ?
Prevent burn-out.   (just had to add the ace-thingy….)

1

CBO is about Efficiency   **(Hah …)**
  Mostly OLTP stuff…

Lead a horse to Water   **(Indexes !)**

Make him drink   **(CBO, spfile, stats, ...)**

Conclusions   **(Simple messages)**

3 min Discussion   **(Do Challenge!)**

Commit Your ORACLE Knowledge

Agenda. Efficiency starts with Design. CBO is just a way to get ppls Attention.
I hate the CBO. But lead the CBO to efficient indexes and it may just do its job.

2
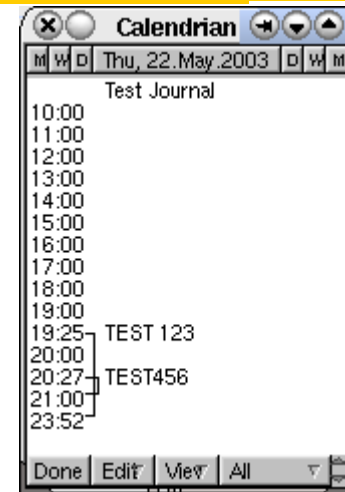
Fast - and Scaleable

- **Individual actions; must be efficient**
  - **C reate / Insert (1 x)**
  - **R ead / Queries (Nx, which fields, why?)**
  - **U pdate   (Nx, which fields ?)**
  - **D elete   (1x, bulk/del old data?)**
  - **…. Efficient ? Indexes ?**

- **Concurrent actions; must remain efficient**
  - **Limit locks (no blocking of others)**
  - **No unusable indexes (exhange part..!)**
  - **No hot-blocks (buffer busy waits).**

Commit Your ORACLE Knowledge

Measure individual activity (of Critical components) – can be a bit of work, but once you "get it", its not hard)
Then think & measure your concurrency issues : does it remain efficient, doest it scale…

When is something "efficient"

- **OLTP typical numbers ?**   **(check your StatsPack)**
  - **How many LIOs (gets) per transaction ?**
  - **how many gets/execute ?**
  - **How many gets/row ?**

- **IMHO (no golden rules):**   **(but check v$sql …)**
  - **< 1000 gets/trans**
  - **< 100 gets/exe**
  - **< 10 gets/row**   **(B-level + table-access… <5)**

- **Why these numbers ? Easy to remember.**
  - **NO Rules   -   YMMV!**

Commit Your ORACLE Knowledge

Just my opinion on what could be "efficient".
And tuning by Ratios is soo 1990s…

Your Milage May Vary!
Wonder what this will be like in 10 years time…

4

What to look for (random examples)….

	Per Second	Per
Redo size:	629,039.51	33.37
Logical reads:	120,998.82	1,474.09
Block changes:	3,56	43.47

Top 5 Timed Events

Event	Waits	Time(		Total Call Time	Wait Class
CPU time				58.9	
db file sequential read	83		6	18.7	User I/O
log file sync			6	18.7	Commit
log file parallel w	3		3	17.3	System I/O
db file pa	149		12	5.8	System I/O

| Trans | 82.08 | |

And … Is your Customer Happy ?

Reading AWR is a topic in itself. The numbers are important, but the customer is more important (duh)
Serious Performance nowadays is about measuring time and resources, the top-5 AWR and the MilliSeconds….!

- **OLTP: Anticipate, Plan (+ Verify) Access-Paths ….**
  - **Indexes!**
  - **Clusters (indexed), Hash Clusters.**
  - **IOTs**

- **The only way to Data... (index or…. FTS)**
- **But anything Fast… (<100 LIOs)**

  - **Needs a good Index (or a hash).**

- **Blog on Indexes: Richard Foote.**

Commit Your ORACLE Knowledge

Anything that needs ms responses on any amount of data… needs an Index.
The index is your Road to Data!   Indexes Show the Way. (CBO just reads the map, or uses TomTom)

- **Any (Btree) index is: Data + pointers, stored in Order.**

- **Index + statement, (DBA and Dev) must work together:**
  - **Good: Leading columns in the Where-clause**
  - **Better: All where-conditions in the index (smallest slice)**
  - **Even Better: Order-by from Index, Prevent sort**
  - **Best: All data from Index, don't visit the Table**

- **Various books, but … Tapio Lahdenmäki !**
  - **All you need to know about "good" indexing.**
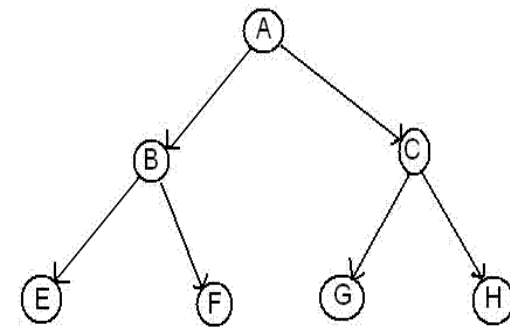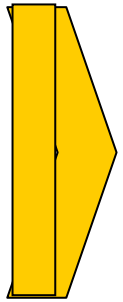
Commit Your **ORACLE** Knowledge

- **Table: Records in random order.**

- **PK: Unique value(s), sorted.**

- **UK: other Unique set(s), sorted.**

- **Other Indexes (e.g. postcode, zip…)**
  •Table   •Index   •Index   •Index
           •(PK)   •(UK)   •(other)
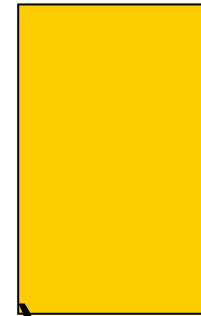
  - **Sorted.**

- **So how often is that data stored and sorted now … ?**

- **And when I update something… (yep – the downside!)**

- **Indexes slooooooow down DML,**

- **Indexes generate more Redo.**

A quick illustration of indexes…. And the ups and downs
Nothing you don't already Know.

- **Index-overloading; a known trick:**
  - – **More fields in the index, Favor Index only-access.**
- **You can also do this for Constraint-indexes!**
  - – **PK, UK or FK Index _Can_ contain more data…**
  - – **Useful to load a "status" or "date_effective" column**
  - – **And reduces Density! (recs/blck) - Buff-busy waits on PK…**

- **Beware of the downsides (DML + Redo) !**

- **Demo: `Owner`, `Table_name`, `num_rows, row_len, …`**
  - – **PK constraint with extra fields in Index…** (SQL > @pk_overload_1.sql)

Commit Your ORACLE Knowledge

Often-used "satus" fields can be added to the most-used index and save in two ways: less index-data to scan, and less table-access required. Another bonus: Denisty decreases! – google : Richard Foote!

- **Index-Reverse:**
  - **Only for Equal Predicates!**
  - **To Fix Buff-Busy-Waits or to help "RAC"**

- **Most Selective Columns first…?? (partly myth)**
  - **Only If you have choice…and If Relevant for Queries**

- **Compression of indexes…Yes! (less IO, less work)**
  - **If Non-selective columns come first.**
  - **Use Analyze-validate struct: index_stats**
  - **Caveat: upserts wreck havoc…. TEST!**
  - **Wishlist: Compress Indexes per partition.**

Commit Your **ORACLE** Knowledge

Reverse: only if you have Buff-busy-waits. Column order is a bit of a myth, not always relevant, and often not much choice. Compression: Works wonders on read-only Varchar-data, but sometimes turns against you.

10

SOLUTIONS THAT MATTER

- # Clusters (indexed)
  - – **Group data with same cluster (parent-)key (=Often Good!)**
  - – **Need regular, predictable data and good size-estimates.**

- # Clusters (hash, and hash-ordered)
  - – **Potentially <u>The Fastest Way</u> into Data (google: Joze)**
  - – **Equal-Predicate only (or use index)**
  - – **Spread + Group the data, children of one Parent..**
  - – **Size Estimate, requires regular and predictable data.**

- # Anyone Using Clusters ?
  - – **(benchmarking, <u>predictable</u> <u>data</u> and <u>predictable</u> <u>operations</u>!)**

Commit Your ORACLE Knowledge

Clusters often seem promising, but the "size" estimate requires predictable data.
Willing to learn on this one! Someone do a ppt on this next Conference ? (demo_sthc.sql if time)

- **Index Organized Tables**
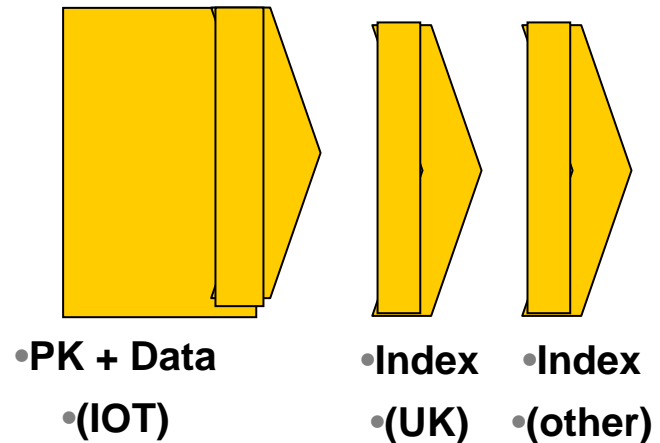 - Overloading to the extreme: all data in the PK.
- **Group and Order data by leading columns**
  - Ideal for Parent-child tables: Children Forced together.
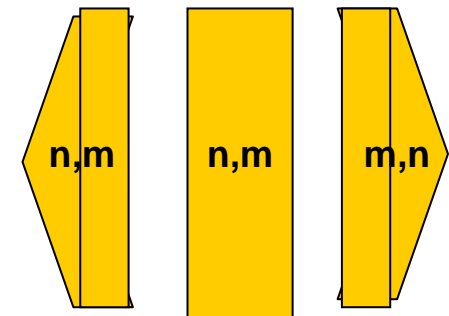- **Also Good for (small) Lookup-tables (TomK, RichardF)**

- **IOT : one less segment..:**

- PK + Data
- (IOT)

- Index
- (UK)

- Index
- (other)

Commit Your ORACLE Knowledge

IOTs simply don't have "tables", all data is in the PK.
You have one less segment, there is no more table, no more un-ordered heap of records…

12

SOLUTIONS THAT MATTER

- **Bonus-feature on IOTs: Fat Indexes**
  - **2ndary indexes are "overloaded"**
  - **contain the PK-values (as rowid) to allow Access to PK (+data)**

- **Good for n:m relationships and join-only access**
  - **Normally, you need TBL + PK + FK**
  - **(you can "overload" to get index-only-access)**
  - **The IOT does the overloading for you…**
  - **And removes the "table" segment altogether.**

  n,m   **n,m**   **m,n**

- **Show grouping + bonus-feature.**   **SQL > @ Demo_iot.sql**

Commit Your ORACLE Knowledge

2ndary indexes merit a good look! Neat Trick nr #2: automatic overloading.
Deep-down-wishlist: use 2ndary indexes as overflow-segment…. Too complicated for Fast-ppt ?

13

- **Slow DML (ins/upd)**
  - **Especially Bulk & Batches, no "direct" insert**
  - **(people are used to bulk-loading without any indexes…)**
- **Overflow Segment with extra columns.**
  - **Plan it – don't get taken by surprise.**

- **2ndary indexes are bigger**
- **"Table" access is slower, Row-Guess(?), more "hops"**

- **Statistics take forever (9.x only?).**
- **Bugs… (ML** 471479.1 **+ more)**

Commit Your ORACLE Knowledge

Downsides are non–negigable, as one customer noted… (ML:) Suspect extra redo, need to test.
Wishlist: Prevent Overflow segments, or control overflow to go into 2ndary Index (e.g. avoid any heap–segment)

14

- **Verify Access Paths (especially on OLTP):**
  - **Explain, (auto)trace and check v$sql and v$sql_plan**

- **Good Indexing: Tapio!**
  - **will help you more then anything**

- **Overloading is useful**
- **Index-Compression is useful (but test)**
- **Clusters and IOTs … If applicable (but test)**

- **Now for the C-B-O… (paracetamol…)**

Commit Your **ORACLE** Knowledge

Just to stress – and this is striclty my own humble opinion, but I will shout it aloud!
Do you notice we havnt lost any time / sleep over CBO yet ?

15

Thank You !
Stay Tuned
(not done yet…)

Find out from the audience.
Bonus question: who prefers init-files ?

- **CBO is like TomTom (= Garmin); Very Clever, but…**
  - **Do you ever mess with your Tomtom ?**

- **You need good Roads - hence my rant on Indexes.**

- **TomTom needs "the map"**

- **TomTom needs good "settings"**
  - Image:
  - common sense

- **Sometimes it needs common sense**
  - **Hints, outlines, sqlplans**

- **And Somtimes it needs a Spanking.**

Commit Your ORACLE Knowledge
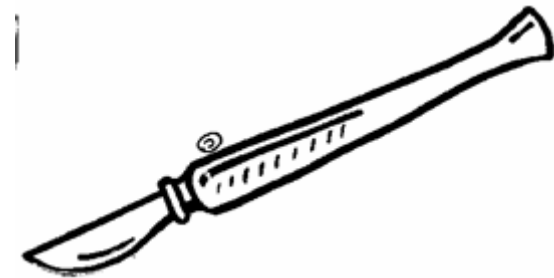
CBO… sooo much to it! I can only begin to scratch the surface…
When TomTom goes bananas, you don't 10053 on it, nor change the settings: you use a map + brains…

17

SOLUTIONS THAT MATTER

- **SQL > COMMENT on table emp is 'you are it';**

- **SQL > GRANT select on emp to perfstat;**

- **SQL > ANALYZE table emp estimate statistics ;**

- **SQL > alter system flush shared_pool;**

- **SQL > .. You may have more of these …**

- **Pre-11 shooting of a cursor!**

- **Never quite "precise", but they generally work.**

- **New: dbms_shared_pool.purge(cursor)   (c/o "Prutser")**

Commit Your ORACLE Knowledge

Early morning shooting… In ora 11, there is a method.. Check Harald v Brederode.
"Precise, targeted Fixes" are always better the "generic" settings.

18

- **Spfile-parameters**

- **System stats**

- **Object Stats**

- **Session-parameters**

- **Outlines or SQLPlans**

- **Hints (if you really have to…)**

- **(and … sometimes it "needs to get lucky" – c/o JL)**

- **Realize the hierarchy: Order**
  - **Troubleshoot from bottom to top!**

There is a hierarchy in the information used by CBO. Troubleshooting should take this into account.
Start with changes that are "precise", aimed at your problem–only, prevent wider impact of "tweaks"

SOLUTIONS THAT MATTER

- **Spfile: Simplify; get rid of ALL init.ora "history".**
  - **Any change = system wide   (Don't Mess here!)**
  - **Optimizer_mode…   (dflt Choose is fine)**
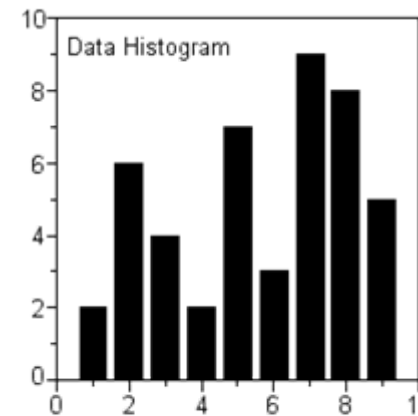 - **Hash-/Sort-area-size   (Session level, if at all)**

- **System Statistics (often overlooked):**
  - **Gather on your hardware   (CPUs, disk-behaviour)**
  - **Tip: Collect + Plot over time, get a feel for your system.**
  - **Set system-stats manual ?…**
  - **(See book by Christian Antognini, but Need more Info)**

Commit Your ORACLE Knowledge

System stats tend to vary, notably if hardware changes (vmware, dynamic hardware provisioning) and SAN behaviour over the day – re-check the book and the docu! Funny: ML **368252.1 (for EBS users)**

SOLUTIONS THAT MATTER

- **Session-parameters**
  - Override Spfile-parameters – for duration of session.
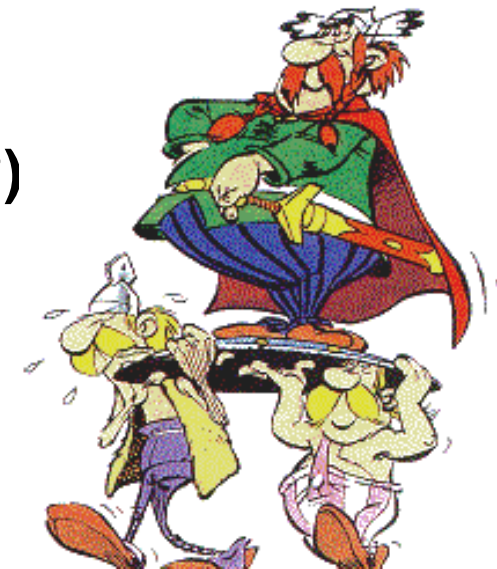 - (I don't mess with these, but you can…, optimizer_mode)

- **Object Statistics**
  - DBMS_STATS (But I "analyze" when in a hurry)
  - Can be "Set", copied, tweaked.
  - In doubt: use worst-case stats and LOCK.
    - if it works on 60M, it works on 120 records too.
  - New Month, New Partitions: Stale.

Commit Your ORACLE Knowledge

Session settings will (only) last for the session. Beware how long your session lasts.
Table/index_stats will auto–gather… do you want that ? What about new, "empty" objects ?

- **<u>Simplify</u> (be Lazy): Use default gather_stats_job**
  - **Gather_database_stats_job (internal use only … ?)**

- **DBMS_STATS = Heavy (and unpredictable)**
  - **Check the Maintenance Window**
  - **(and learn to use the scheduler)**
- **Save Stats you trust – for re-import (=effort?)**

- **10g: Restore-stats: Safety-net.**
  - **Retention of 31 days…**

Long and hard discussions on "when + how to analyze" Just do auto–gather + stale… but…
Beware of the Job, control the window. Nb: Restore is different from exp/imp stats !

22

- **Stale, 10%... : Lock stats you trust! (but how long…?)**
  - **Check for stale anyway (=work…).**
  - **Locking of stats: for any use of the segment**
  - **Locking of SQL (hint, outline, sqlplan) : per stmnt…)**

- **By Exception only: set or tweak stats.**
  - **Volatile tables, GTTs … maybe… (I hestiate, ... more work…)**

- **"Upgrade took two weeks to stabilize…" (Thx!)**
  - **You need an upgrade-strategy,**
  - **Whitepaper + outlines/SQLplans!**

Complications around gathering stats… sigh… the "automation" is good, but beware of unpredictable changes. 10%–stale is a bit low. Lock the ones you trust. Upgrade was my intro into "outlines"

23

Outlines: an "Emergency" strategy

- **I never liked outlines: too much hassle, but..**
  - I Discovered their use on upgrades from 9 to 10.

- **IF you can afford to do this (1 hr work):**
  - Get outlines of all major queries when "Good"
  - Then either lock m in place.
  - Or keep m for use and reference when needed.

- **This is SQL-Plan management by another name..**
  - But I'm not on 11g yet…

Commit Your ORACLE Knowledge

Collect Outlines when the system is running "as intended" and keep those just-in-case.
You can then activate one or more outlines when queries go out of control (non-intrusive!)

24

- **Hints are EVIL**
  - **Gremlins, time-bombs (job security?)**
- **Maybe: on GTTs**
  - **Dynamic sampling (tt, 1) (c/o JLewis)**
- **Possibly on "The-Cast-table" in PL/SQL**
  - **Tell CBO what is in your array**

- **You can "catch" a hint from dbms_xplan…**
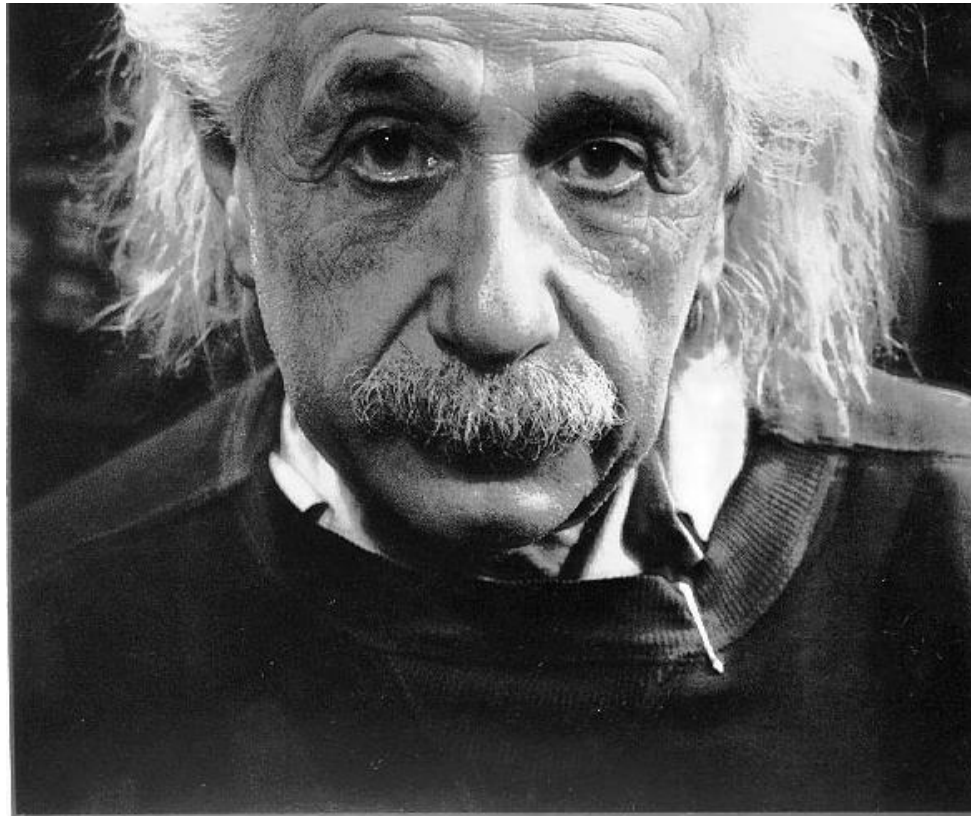 - **(demo_outline_hint.sql – how to get in trouble…)**

Commit Your ORACLE Knowledge

Hints… ongoing work… on GTT: dynamic–smple, or cardinality or ordered…
Hint (outlines) can be "gotten" form dbms_Xplan now!   (**11gr2: IGNORE_ROW_ON_DUPKEY_INDEX .. ? Later!**)

Logica

- **Leonardo da Vinci:**
  - **Simplicity is the ultimate sophistication.**

- **Goethe:**
  - **In der Beschränkung zeigt sich der Meister".**

- **EW Dijkstra:**
  - **Simplicity is a pre-requisite for reliability.**
  - **The sore truth is that Complexity sells better**

Commit Your ORACLE Knowledge

"If you can't explain it simply, you don't understand it well enough"

As Simple as Possible, but not too simple
Simplicity is a Requirement – but Comlexity just sells better (EWD).

Takeaways for "Architects" + DBAs

- **Define Efficiency:**
  - **Get a feeling for your system – Statspack, AWR, v$sql**
- **INDEXING : all Access should be "efficient" – check!**
- **CBO: be lazy, dont tweak (not too much).**
- **Stats: Auto-gather, and use a window**
  - **Trust but Verify : save outlines for a rainy day**
  - **Inspect your stats on problem-qries/tables.**
- **Tweaks (set-stats, use outlines, use hints)**
  - **Sometimes you have to…**
  - **Minimize and simplify**

Commit Your **ORACLE** Knowledge

Oh, and "Oracle is Great, and Larry is its CEO" … (I have to say this now that I got "ace")

- ## Simplicity
  - In case of doubt: Simplify!
  - Less Tweaks, Less work,
  - less procedures, less scripts…

- ## Can you explain it ? To me ?
  - Can I take your system without paper documentation

Commit Your ORACLE Knowledge

In case there is time: SIMPLICITY!
If me, or someone like me can take over your job… (so you can do more interesting things)

29

Don't Take my word for it…

Tahiti: start with concept-guides

Technet (but be critical)

Oracle-L : real world stuff

www . Bloggingaboutoracle . org (official ramblings)

SimpleOracleDba . Blogspot . com (private ramblings)

Goethe (simplicity…)

Commit Your **ORACLE** Knowledge

Majority of times, I have been WRONG.
So go see for yourself – but don't complicate life.

30

- **Questions ?**

- **Reactions ?**

- **Experiences from the audience ?**
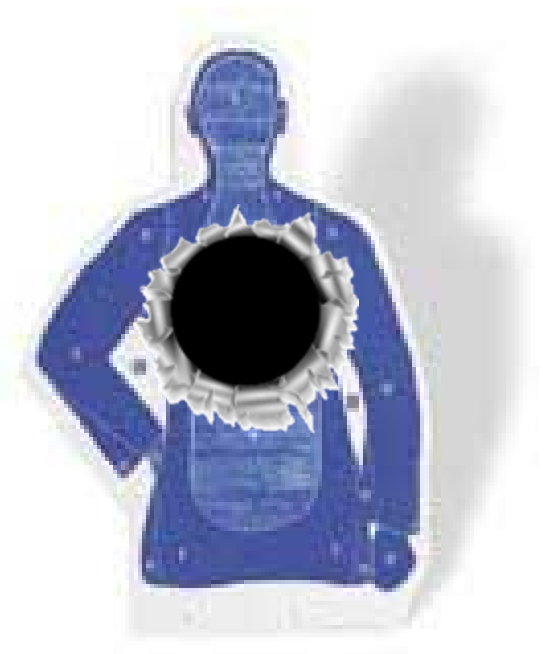
Commit Your ORACLE **Knowledge**

Question and Answer time. Discussion welcome   (what about that Razor?)
Teach me something: Tell me where you do NOT AGREE.

31

- **Road + map are good metophores.**

- **Occams Razor**
  - **Least possible ASSumptions**
  - **Simplest Solution.**
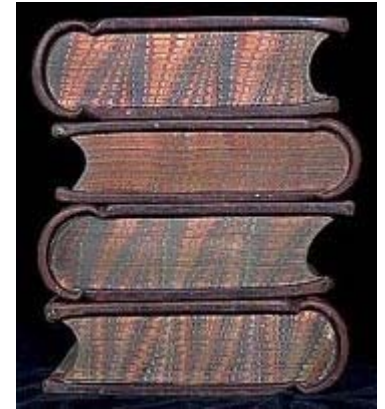
Outlines (and plans)

- **There is a lot to outlines and "plan management"**

- **It it becomes (complicated) tweaking….**

- **It is probably too … complicated.**

- **Image ?**

- **Think of an easier way !**
  - **Good indexing – good INDEXING…**
  - **Shoot qrys (but cant do that forever)**
  - **Try New, better stats**
  - **Outlines – just in case**

Commit Your **ORACLE** Knowledge

Time.. You cant tweak CBO forever, no matter how much Oracle wants you to…
Did I mention "good Indexing" ?

33

- **Frozen Plans.**

- **Thu, 18Feb, 11:15 (one one one five...)**
- **Hall …,**
- **the SIMPLE approach to Indexing and CBO**

- **Same time: many real celebrities presenting**

Commit Your ORACLE Knowledge

User Managed: Flexible, but "You run it" Dataguard: covers almost any-outage, user-errors, machine-failure, site-disasters, and great for "other usages"