

# Indexing and CBO

**Help the system do its job**

Piet de Visser

The Simple Oracle DBA



Commit Your **ORACLE** Knowledge

# Agenda (45min - 2 parts of 20 minutes)

PDVBV

SOLUTIONS THAT MATTER

**This is about Efficiency  
Mostly OLTP stuff...**

**(Hah ...)**

**Lead a horse to Water**

**(Indexes !)**

**Make him drink**

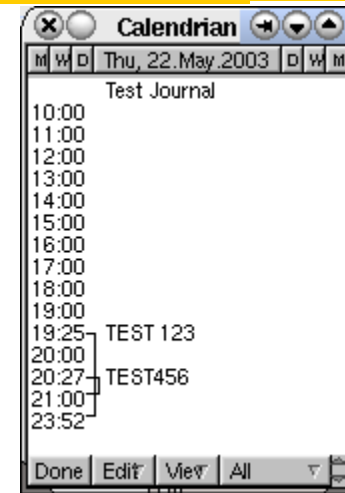
**(CBO, spfile, stats, ... )**

**Conclusions**

**(Simple messages)**

**3 min Discussion**

**(Do Challenge!)**



Commit Your **ORACLE** Knowledge

- **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** (exchange part..!)
  - **No hot-blocks** (buffer busy waits).



Commit Your **ORACLE** Knowledge

# 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 ...)
  - < 10 gets/row (B-level + table-access... <5)
  - < 100 gets/exe
  - < 1000 gets/trans
  
- **Why these numbers ? Easy to remember.**
  - NO Rules - YMMV!

Commit Your **ORACLE** Knowledge

# What to look for (random examples)....

PDVBV

SOLUTIONS THAT MATTER

	Per Second	Per
Redo size:	629,039.51	33.37
Logical reads:	120,998.82	1,474.09
Block changes:	3,565	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 write		3	17.3	System I/O
db file parallel read	149	12	5.8	System I/O

**And ... Is your Customer Happy ?**

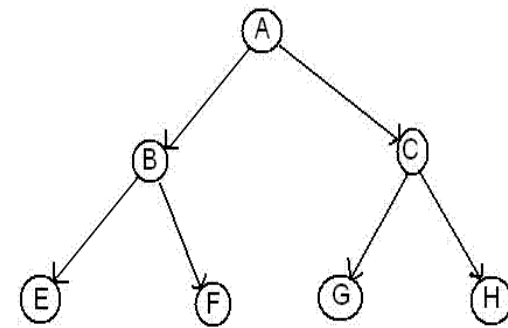
Commit Your **ORACLE** Knowledge

- **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.**



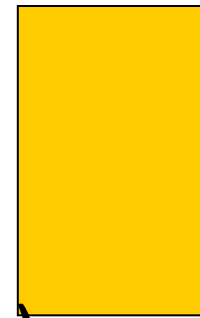
Commit Your **ORACLE** Knowledge

- 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.

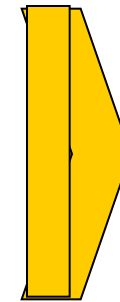


Commit Your **ORACLE** Knowledge

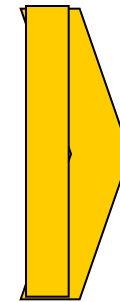
- Table: Records in random order.
- PK: Unique value(s), sorted.
- UK: other Unique set(s), sorted.
- Other Indexes (e.g. postcode, zip...)
  - Sorted.



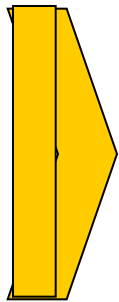
•Table



•Index  
•(PK)



•Index  
•(UK)



•Index  
•(other)

- So how often is that data stored and sorted now ... ?
- And when I update something... (yep – the downside!)
- Indexes sloooooow down DML,
- Indexes generate more Redo.



# Neat trick #1 : Overloading.

- **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 )



- **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

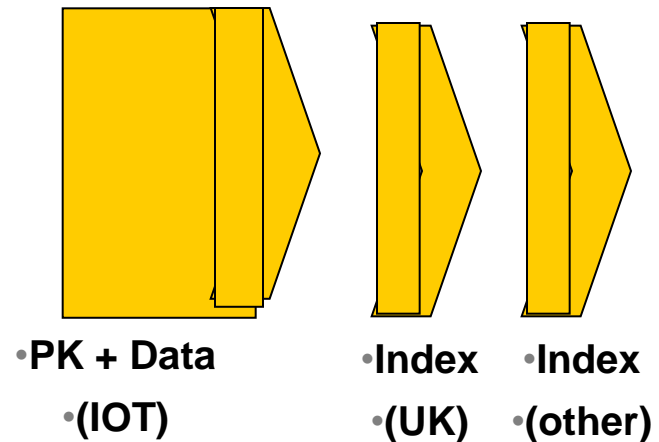
- **Clusters (indexed)**
  - Group data with same cluster (parent-)key (=Often Good!)
  - Need regular, predictable data and good size-estimates.
- **Clusters (hash, and hash-ordered)**
  - Potentially The Fastest Way into Data (google: Joze)
  - Equal-Predicate only (or use index)
  - Spread + Group the data, children of one Parent..
  - Size Estimate, requires regular and predictable data.
- **Anyone Using Clusters ?**
  - (benchmarking, predictable data and predictable operations!)



- **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)**

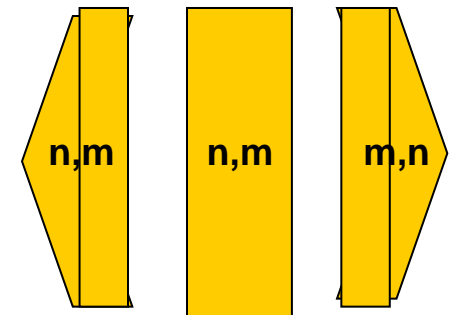


- **IOT : one less segment..:**



Commit Your **ORACLE** Knowledge

- **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.



- **Show grouping + bonus-feature.**

SQL > @ Demo\_iot.sql

Commit Your **ORACLE** Knowledge

- **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

- **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

•

**Thank You !**  
**Stay Tuned**  
**(not done yet...)**



Commit Your **ORACLE** Knowledge



- **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”**
- **Sometimes it needs common sense**
  - Hints, outlines, sqlplans
- **And Somtimes it needs a Spanking.**

- **Image:**
- **common sense**

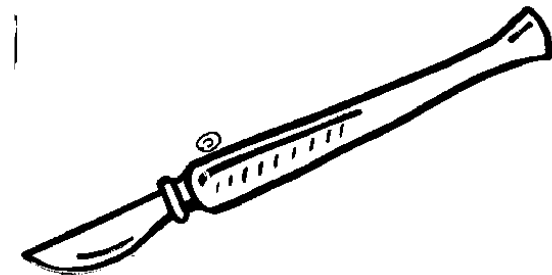


Commit Your **ORACLE** Knowledge

## What do these have in common...

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”)

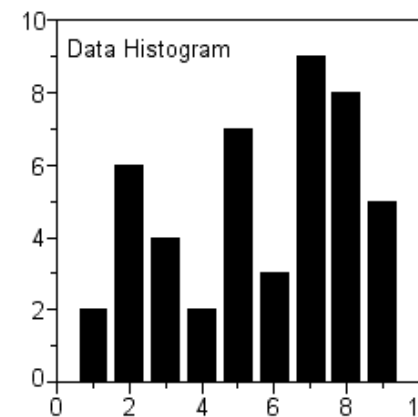
Commit Your **ORACLE** Knowledge

- 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!

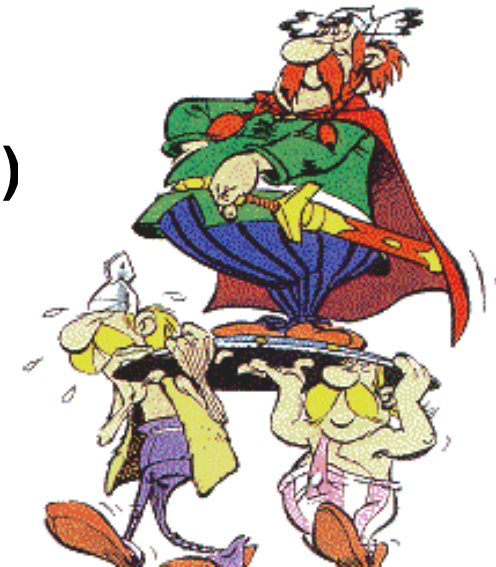


- **Spfile: Simplify; get rid of ALL init.ora “history”.**
  - Any change = system wide (Don't Mess here!)
  - Optimizer\_mode... (dflt Choose is fine)
  - Hash-/Sort-area-size (Session level, if at all)
  
- **System Statistics (often overlooked):**
  - Gather on your hardware (CPUs, disk-behaviour)
  - Tip: Collect + Plot over time, get a feel for your system.
  - Set system-stats manual ?...
  - (See book by Christian Antognini, but Need more Info)

- **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.**



- **Simplify ( 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...



Commit Your **ORACLE** Knowledge

- **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 stmtmnt...)
- **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!



# 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...





- 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...)



# Eh, just to remind you ... Simplicity

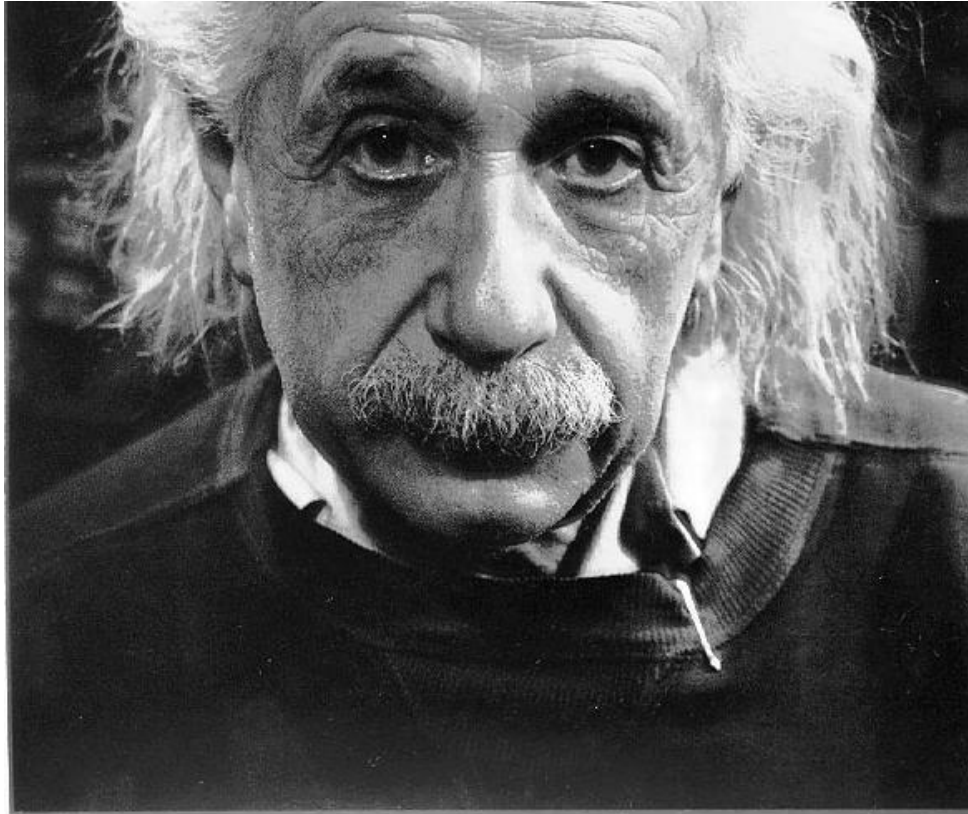
- **Leonardo da Vinci:**
  - **Simplicity is the ultimate sophistication.**
- **Goethe:**
  - **In der Beschränkung zeigt sich der Meister".**
- **EW Dijkstra:**
  - **Simplicity is a pre-requisite for reliability.**
  - **The sore truth is that Complexity sells better**



He got it ...

PDV BV

SOLUTIONS THAT MATTER



**“If you can't explain it simply, you don't understand it well enough”**

Commit Your **ORACLE** Knowledge

- **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

# Advice.. (the hardest slides...)

PDVBV

SOLUTIONS THAT MATTER

- **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

Don't  
Take my  
word for  
it...

SOLUTIONS THAT MATTER

Click to edit Master title style

Tahiti: start with concept-guides

Technet (but be critical)

Oracle-L : real world stuff

[www . Bloggingaboutoracle . org](http://www.Bloggingaboutoracle.org) (official ramblings)

[SimpleOracleDb . Blogspot . com](http://SimpleOracleDb.blogspot.com) (private ramblings)

Goethe (simplicity...)



Commit Your **ORACLE** Knowledge

Quick Q & A (3 min ;-)

3 .. 2 .. 1 .. Zero

PDVBV

SOLUTIONS THAT MATTER

- Questions ?
- Reactions ?
- Experiences from the audience ?



Commit Your **ORACLE** Knowledge