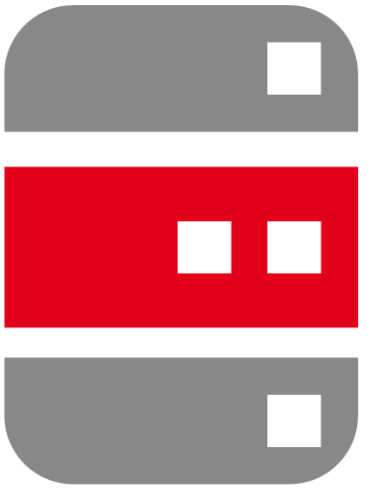
performing databases



Your reliability. Our concern.

performing databases
Your reliability. Our concern.

#1899
The Bad One Into Your Crop - SQL Tuning Analysis for DBAs

Martin Klier

Performing Databases GmbH
Mitterteich / Germany

Speaker

performing databases
Your reliability. Our concern.

- Martin Klier
- Solution Architect and Database Expert

- My focus
  - Performance Optimization
  - High Availability
  - Architecture DBMS

- Linux since 1997
- Oracle Database since 2003

ORACLE ACE

Speaker

performing databases
Your reliability. Our concern.

- Meet & Greet

DOAG
Regionalgruppen
Fachkonferenzen

ORACLE OPEN WORLD

IOUG
independent oracle users group

COLLABORATE
April 2016

- Contact: martin.klier@performing-db.com
- Weblog: http://www.usn-it.de (English)

performing databases
Your reliability. Our concern.

DOAG

Conference + Exhibition
#DOAG2016

2000+ Visitors

Nuremberg, Germany
November 15.-18. 2016

Performing Databases

performing databases
Your reliability. Our concern.

ORACLE Gold Partner

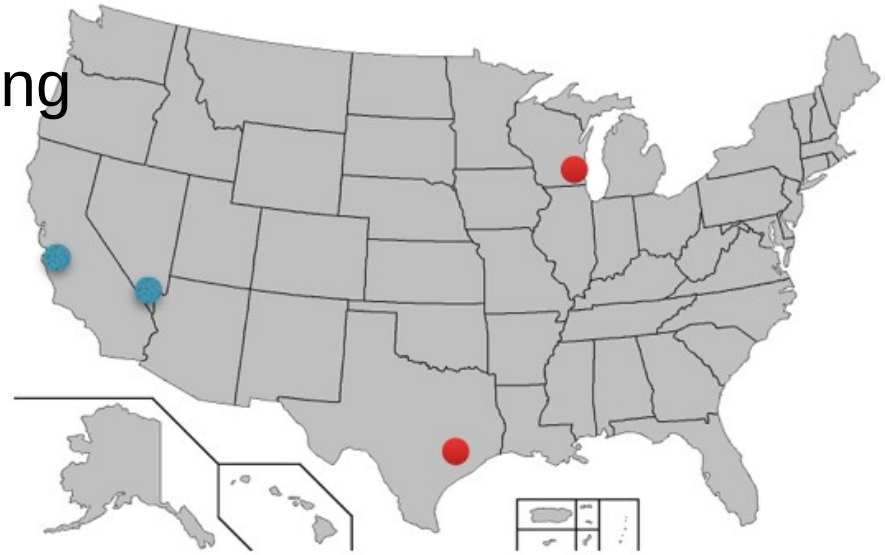
- Experts for Database Technology
  - Concept
  - Planning & Sizing
  - Licensing
  - Implementation and Troubleshooting
- Get in touch
  - Performing Databases GmbH
    Wiesauer Straße 27
    95666 Mitterteich, GERMANY
  - Web: http://www.performing-databases.com
  - Twitter: @PerformingDB

International

performing databases
Your reliability. Our concern.

ORACLE Gold Partner

- Design
- Licensing
- Implementation
- Tuning
- Troubleshooting
- Service
- Upgrade
- Migration



http://www.performing-databases.com

performing databases
Your reliability. Our concern.

Warm-Up

simplicity

means

simple

performing databases
Your reliability. Our concern.

awesome

is the new

awful

knowing

matters

proof

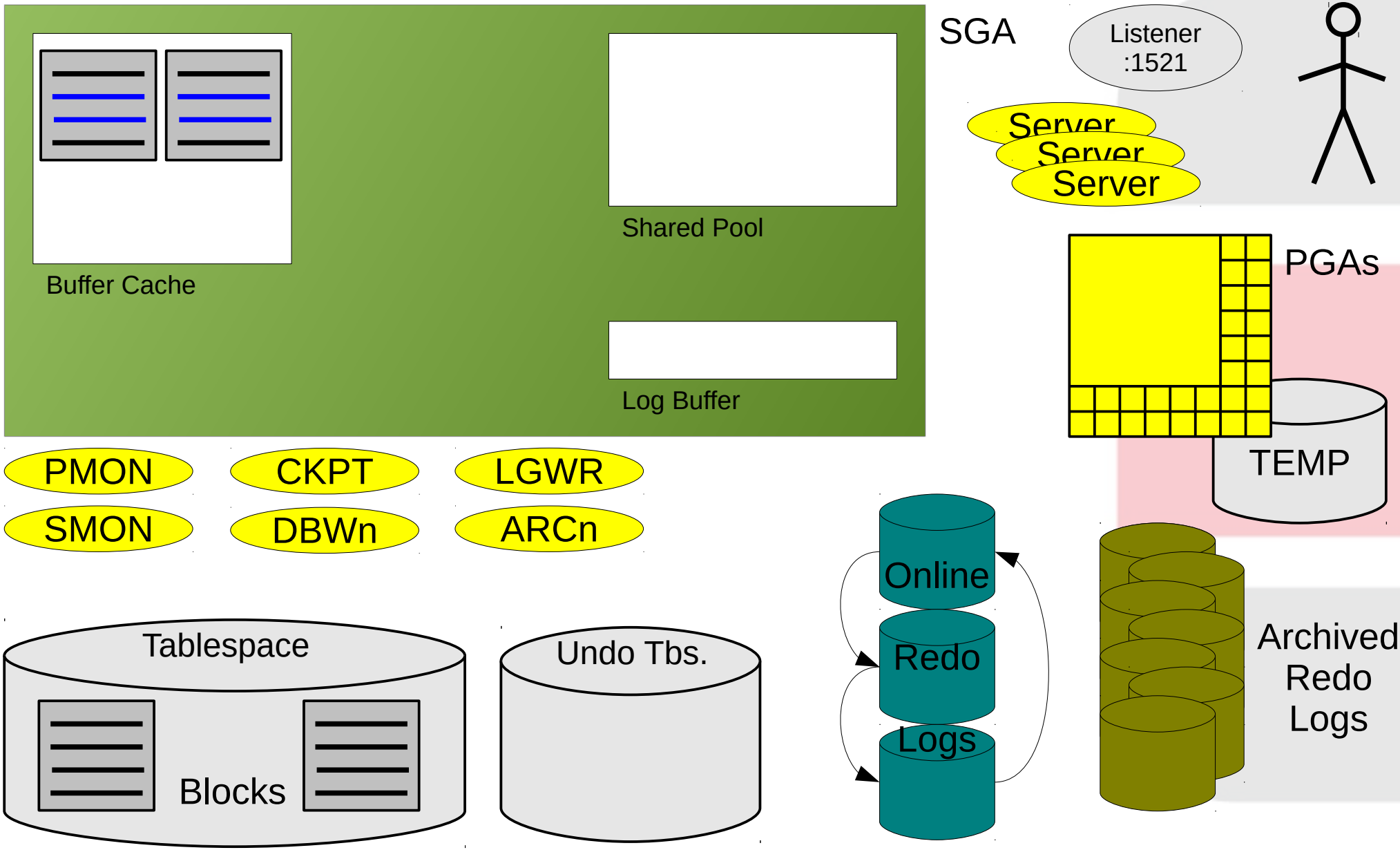
matters more

performing databases
Your reliability. Our concern.

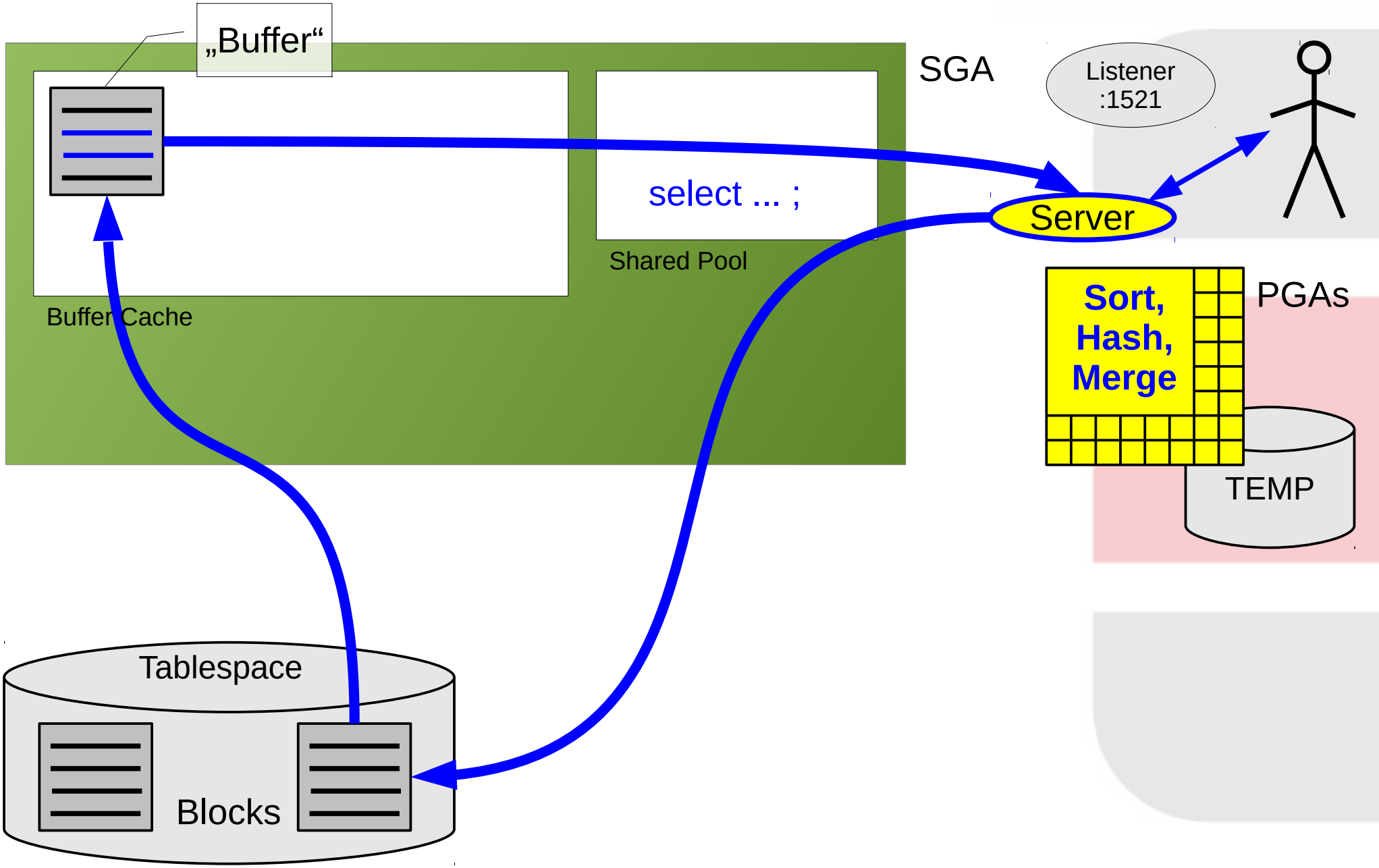
Basics

Oracle Architecture
(simplified)

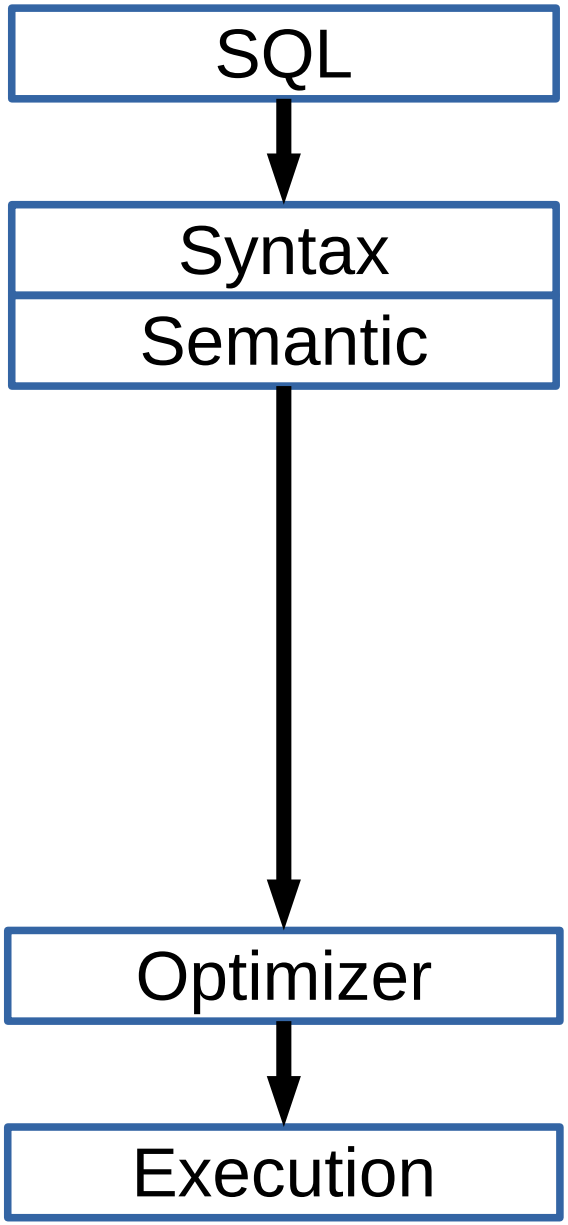
performing databases
Your reliability. Our concern.



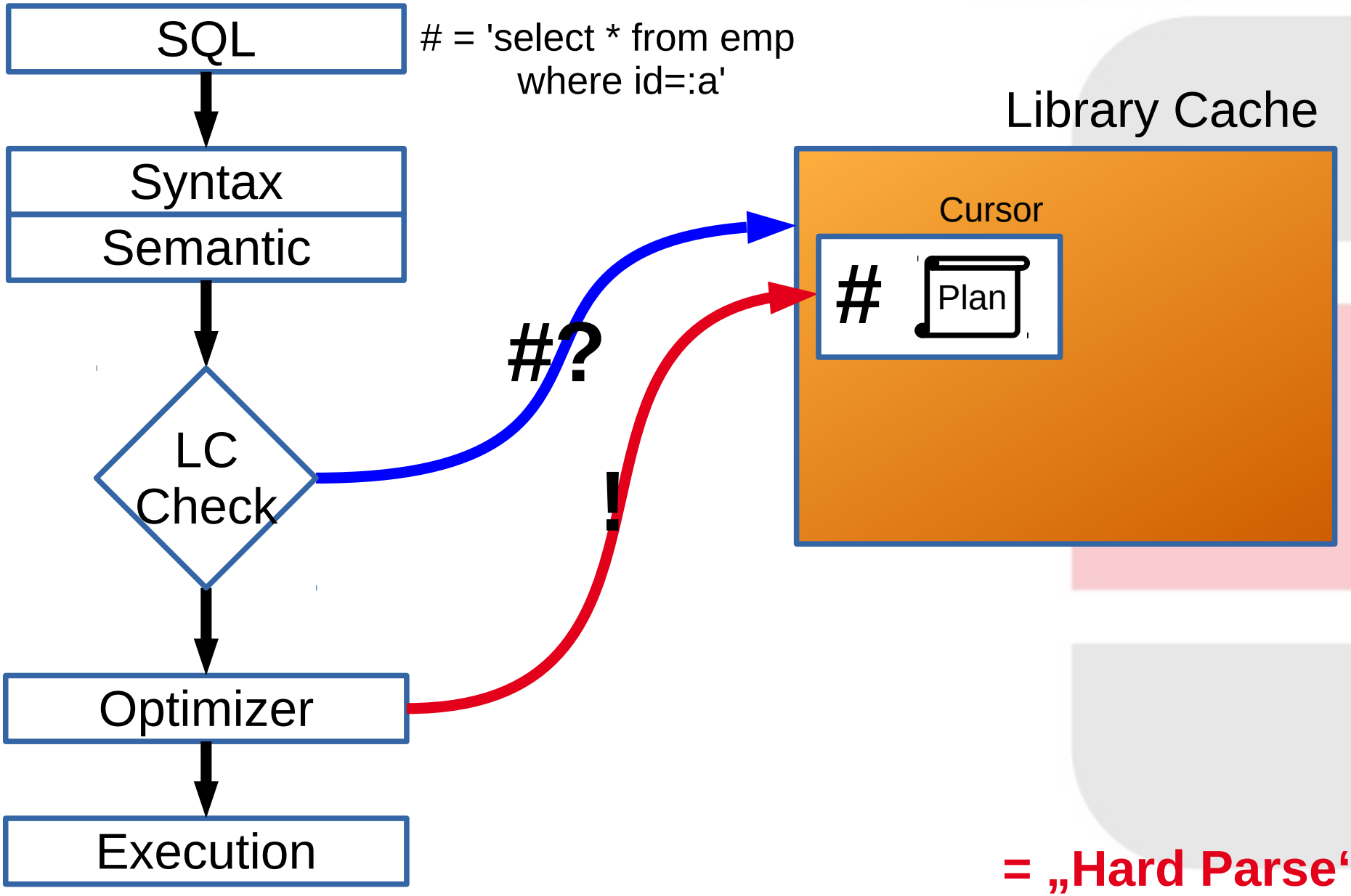
Reading


„Buffer“
SGA
Listener
:1521
select ... ;
Server
Shared Pool
Buffer Cache
Sort,
Hash,
Merge
PGAs
TEMP
Tablespace
Blocks

Parsing

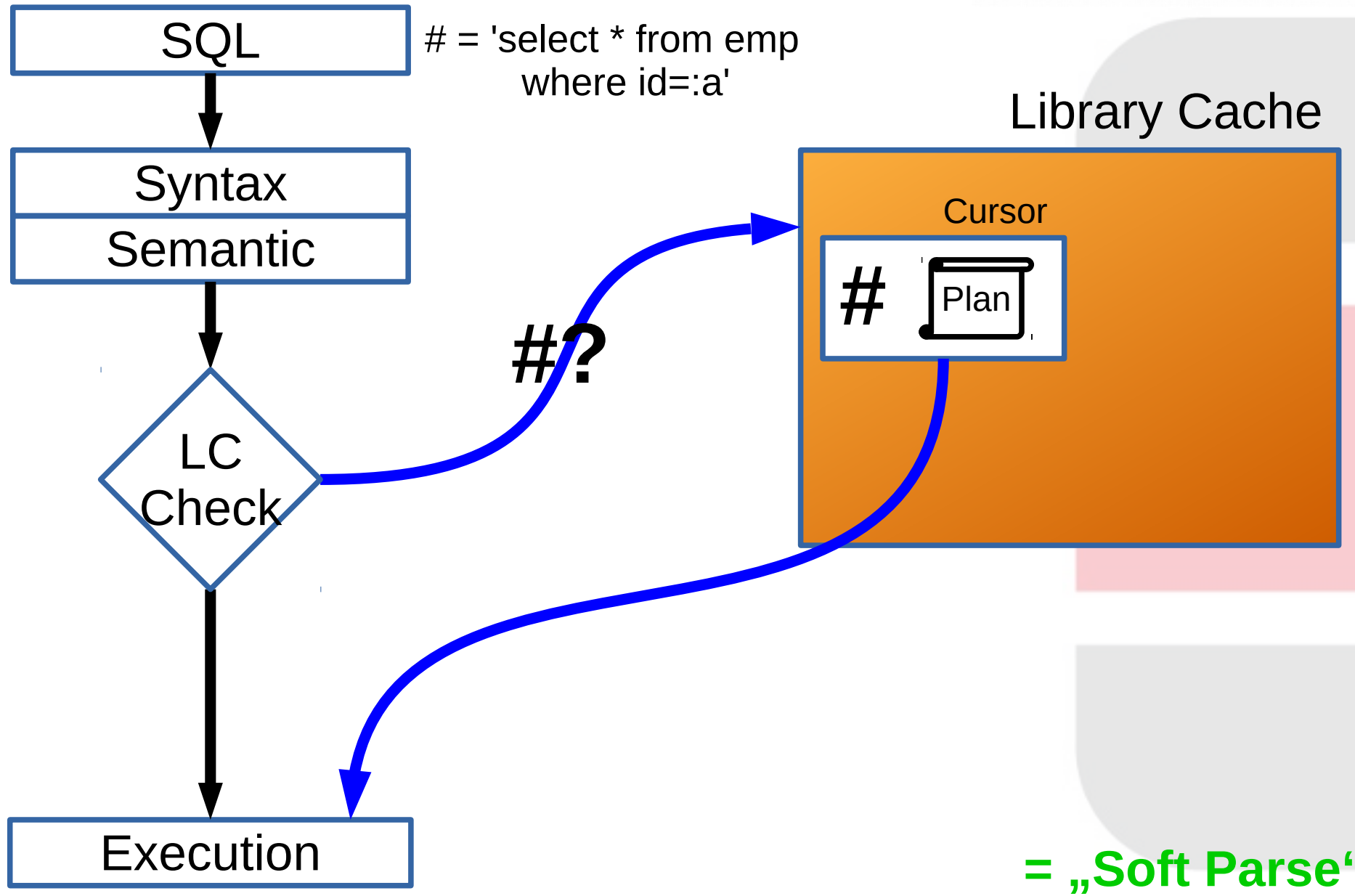


Parsing



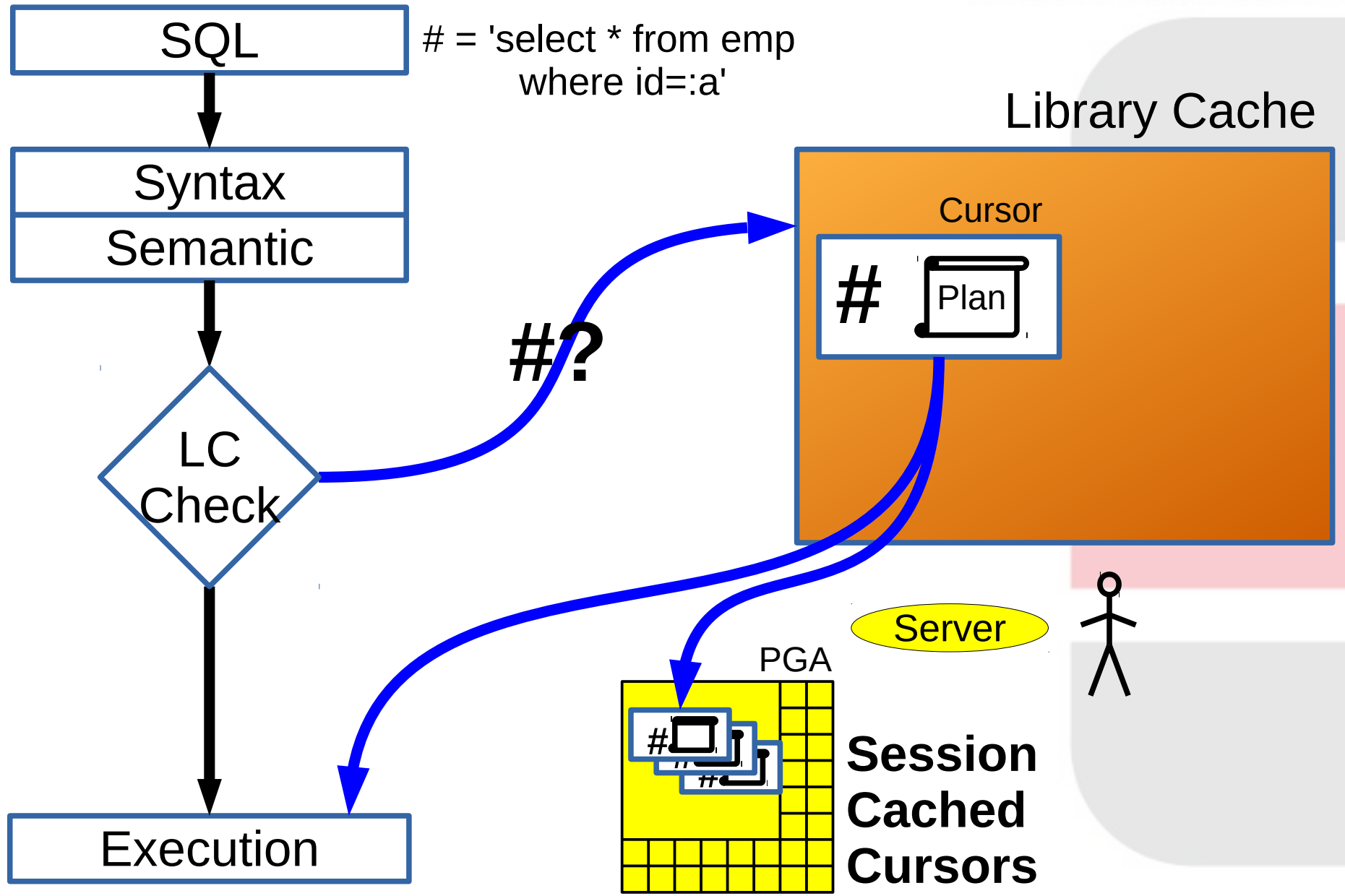
= „Hard Parse“

Cursor Sharing

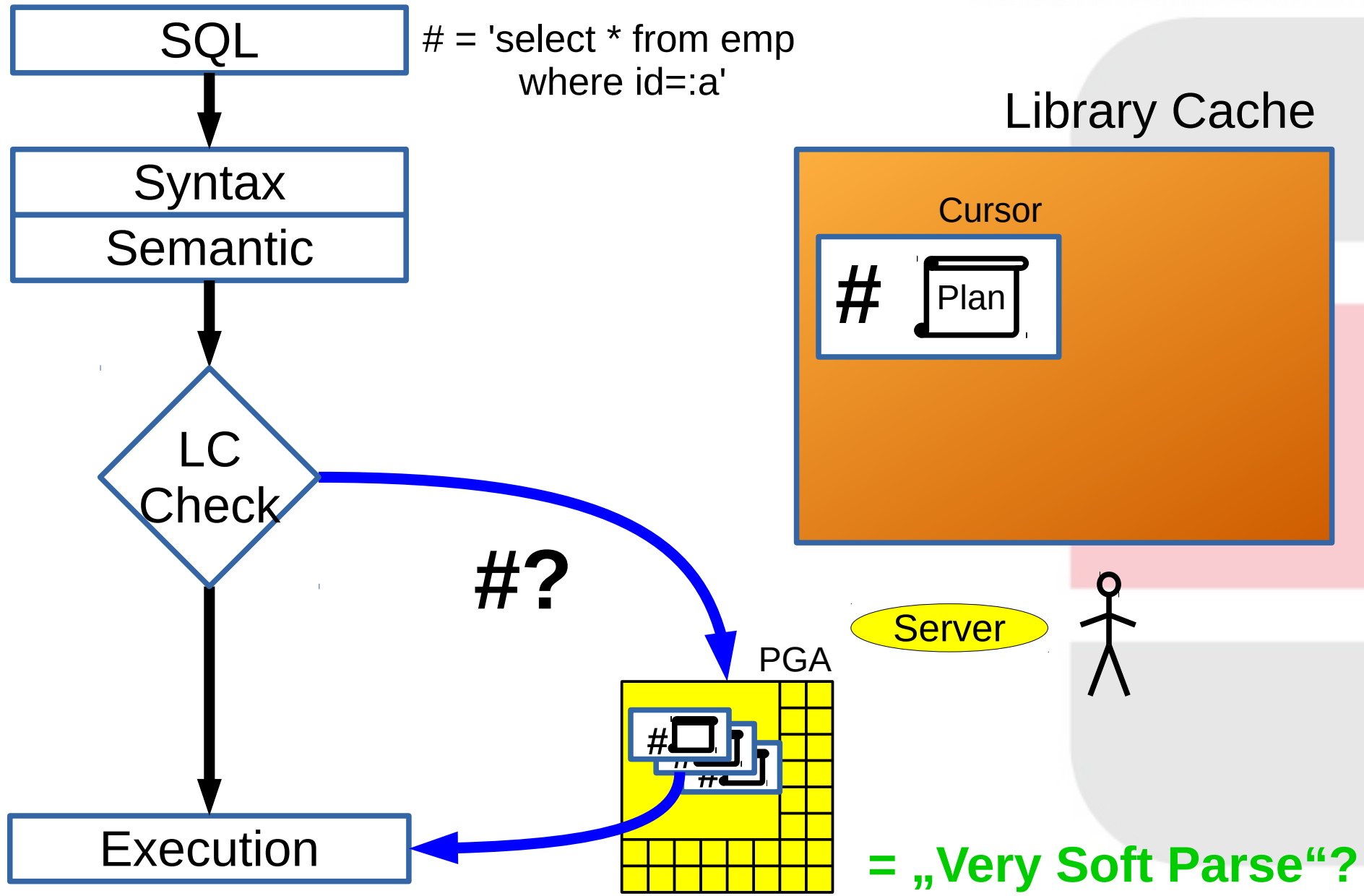


= „Soft Parse“

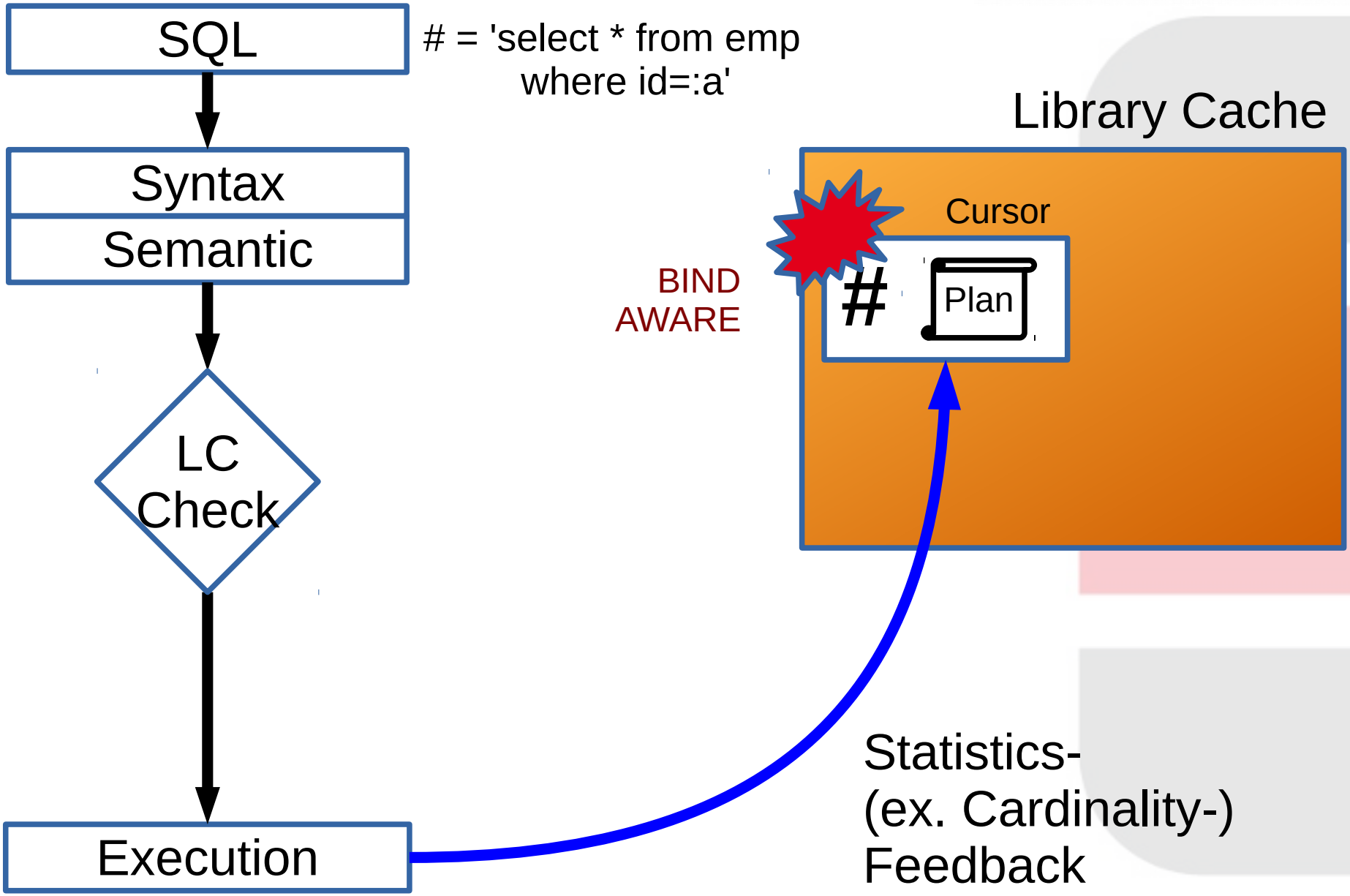
Soft Parse


SQL
= 'select * from emp
where id=:a'
Syntax
Semantic
LC
Check
#?
Library Cache
Cursor
#
Plan
Server
PGA
#
Session
Cached
Cursors
Execution

Session Cached Cursors

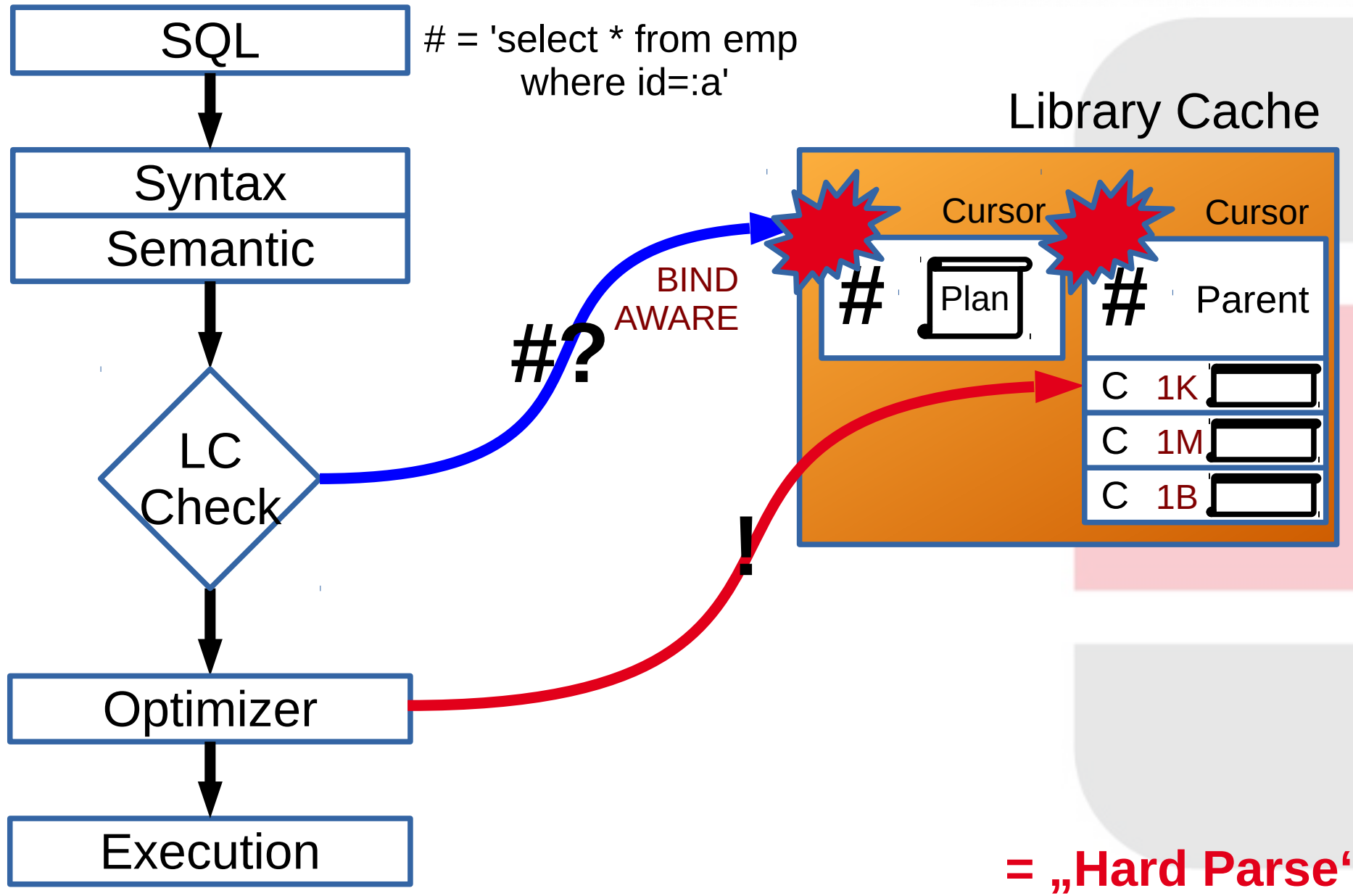


Statistics Feedback



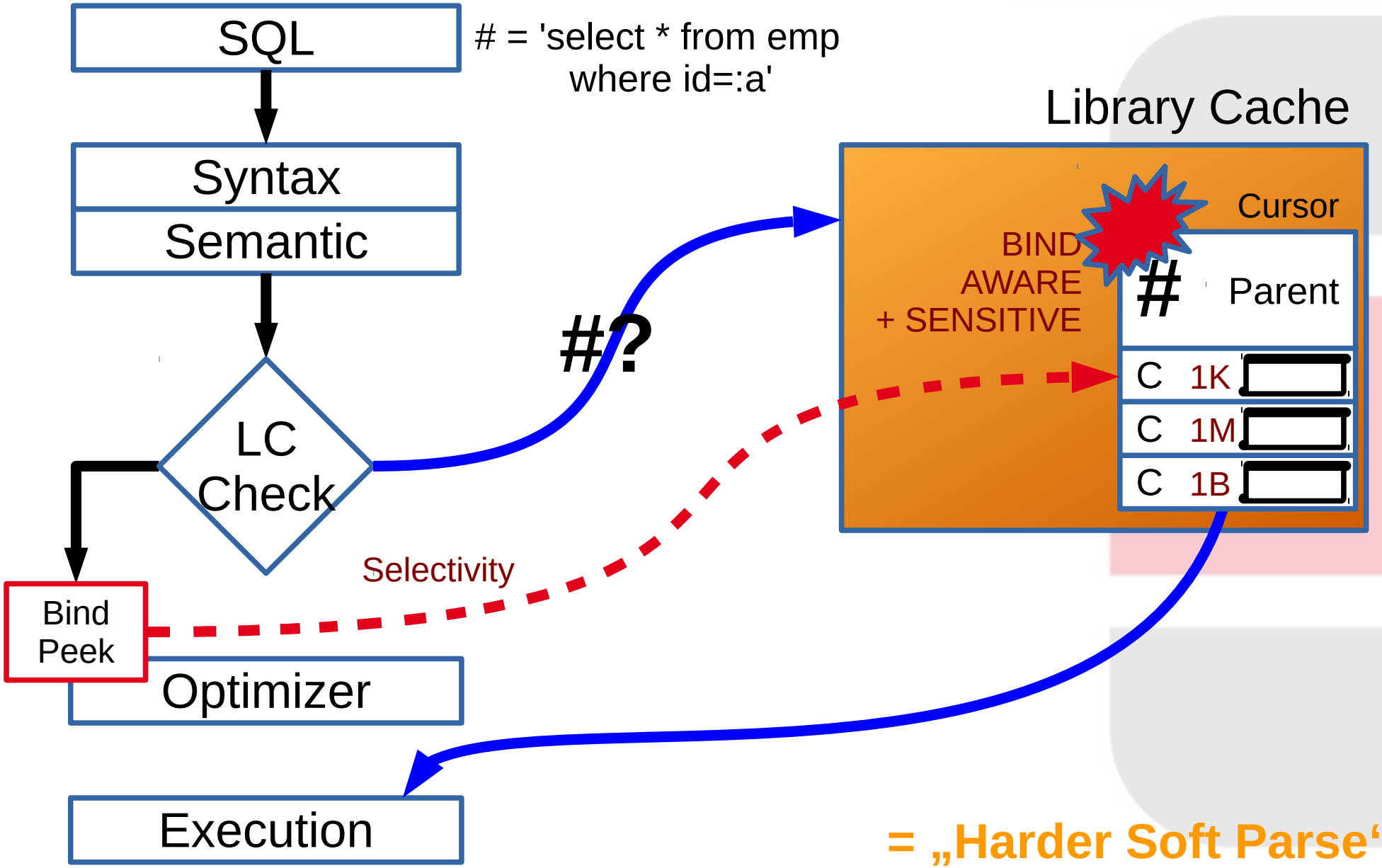
Child Cursor Plans

performing databases
Your reliability. Our concern.



= „Hard Parse“

Adaptive Cursor Sharing



= „Harder Soft Parse“

Adaptive Cursor Sharing

performing databases
Your reliability. Our concern.

Repeat all over:

- Execution
- Feedback statistics
- Hard Parse once
- Add new Child Cursor (reason „Feedback Stats“)
- Harder Soft Parse => Selectivity of bind
- Re-use Child Cursor
- Buffer in PGA (Session Cached Cursors)
- Re-use Session Cached Cursor

performing databases
Your reliability. Our concern.

Analysis

starts with asking

„What's the problem?“

performing databases
Your reliability. Our concern.

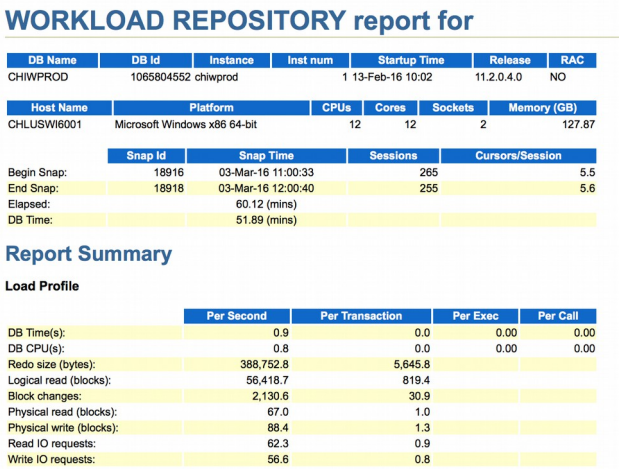
My style

Analysis à la Martin

performing databases
Your reliability. Our concern.

Simplify your Toolset

- AWR or Statspack
- SQL*Plus / SQLcl / SQL Developer (most recent)
- dbms_xplan.display_cursor()



WORKLOAD REPOSITORY report for

DB Name	DB Id	Instance	Inst num	Startup Time	Release	RAC
CHIWPROD	1065804552	chiwprod	1	13-Feb-16 10:02	11.2.0.4.0	NO

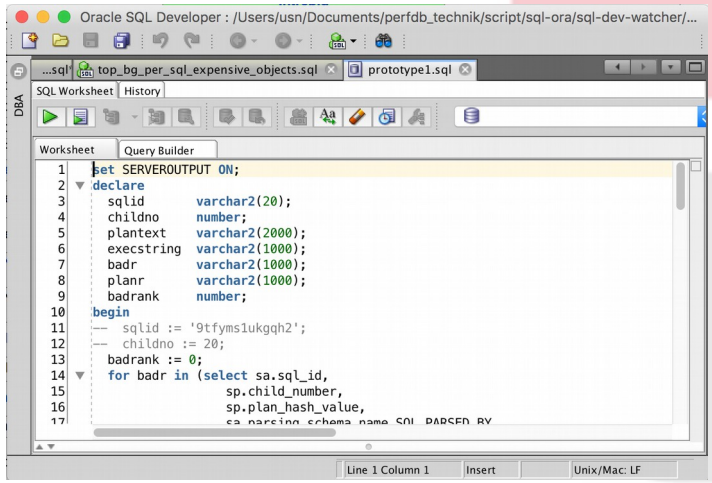
Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
CHLUSWI6001	Microsoft Windows x86 64-bit	12	12	2	127.87

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	18916	03-Mar-16 11:00:33	265	5.5
End Snap:	18918	03-Mar-16 12:00:40	255	5.6
Elapsed:		60.12 (mins)		
DB Time:		51.89 (mins)		

Report Summary

Load Profile

	Per Second	Per Transaction	Per Exec	Per Call
DB Time(s):	0.9	0.0	0.00	0.00
DB CPU(s):	0.8	0.0	0.00	0.00
Redo size (bytes):	388,752.8	5,645.8		
Logical read (blocks):	56,418.7	819.4		
Block changes:	2,130.6	30.9		
Physical read (blocks):	67.0	1.0		
Physical write (blocks):	88.4	1.3		
Read IO requests:	62.3	0.9		
Write IO requests:	56.6	0.8		



Analysis à la Martin

performing databases
Your reliability. Our concern.

Be yourself, use your

- Suspiciousness (don't believe the obvious)
- Mark-A eyeball for facts (look up thoroughly)
- Situation awareness (look beyond the horizon)
- Seat-of-your-pants-feeling and intuition
  (trust instinct where to look first - often true, not always)

Analysis à la Martin

performing databases
Your reliability. Our concern.

Go down to the second

- One hour has 3600 seconds
- Transactions per Second
- IOs per Second
- SQL Execs per Second
- ...

Finding the culprit

Customer complains about multiple dialogs being slow - not all.

Reading a Report

Real-life example, Mar 13, 2016, Switzerland

performing databases
Your reliability. Our concern.

DB Name	DB Id	Instance	Inst num	Startup Time	Release	RAC
CHIWPROD	1065804552	chiwprod	1	13-Feb-16 10:02	11.2.0.4.0	NO

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
CHLUSWI6001	Microsoft Windows x86 64-bit	12	12	2	127.87

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	18916	03-Mar-16 11:00:33	265	5.5
End Snap:	18918	03-Mar-16 12:00:40	255	5.6
Elapsed:		60.12 (mins)		
DB Time:		51.89 (mins)		

We observed 60min of clock time
We had 12 cores for 60min = 720min plus x of time
DB used only 51.89 min (nearly 100% of 1 core)

Can this be overload? Check for single-threaded problem.

Reading a Report

performing databases
Your reliability. Our concern.

Report Summary

Load Profile

	Per Second	Per Transaction	Per Exec	Per Call
DB Time(s):	0.9	0.0	0.00	0.00
DB CPU(s):	0.8	0.0	0.00	0.00
Redo size (bytes):	388,752.8	5,645.8		
Logical read (blocks):	56,418.7	819.4		
Block changes:	2,130.6	30.9		
Physical read (blocks):	67.0	1.0		
Physical write (blocks):	88.4	1.3		
Read IO requests:	62.3	0.9		
Write IO requests:	56.6	0.8		
Read IO (MB):	0.5	0.0		
Write IO (MB):	0.7	0.0		
User calls:	4,905.5	71.2		
Parses (SQL):	2,046.4	29.7		
Hard parses (SQL):	1.1	0.0		
SQL Work Area (MB):	11.4	0.2		
Logons:	0.4	0.0		
Executes (SQL):	2,199.4	31.9		
Rollbacks:	30.3	0.4		
Transactions:	68.9			

CPU

IO

CPU

Reading a Report

performing databases
Your reliability. Our concern.

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
DB CPU		2971,4		95.4	
log file sync	140,337	232,1	2	7.5	Commit
direct path read	152,563	30	0	1.0	User I/O
db file sequential read	62,612	23,2	0	.7	User I/O
SQL*Net more data to client	35,544	11,6	0	.4	Network
SQL*Net message to client	10,348,482	3,9	0	.1	Network
enq: TX - row lock contention	5	3,8	766	.1	Application
db file scattered read	7,948	2	0	.1	User I/O
SQL*Net more data from client	15,015	1,9	0	.1	Network
direct path write	7,600	1,4	0	.0	User I/O

IO

We had 95.4 % of 51.89 min = 49.5 min CPU activity.

Why? And is it related to what hurts the user?

Reading a Report

performing databases
Your reliability. Our concern.

SQL ordered by Gets

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- %Total - Buffer Gets as a percentage of Total Buffer Gets
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Total Buffer Gets: 203,500,151
- Captured SQL account for 80.9% of Total

Buffer Gets	Executions	Gets per Exec	%Total	Elapsed Time (s)	%CPU	%IO	SQL Id	SQL Module	SQL
39,836,966	294	135,499.88	19.58	180.92	98,4	0	c5bpttzczmj5r	iWACS	SELECT mailingId, m
39,161,524	290	135,039.74	19.24	172.27	99,9	0	403dzpcdzg3pt	iWACS	SELECT mailingId, m
9,589,820	121	79,254.71	4.71	16.63	99,5	0	drzjvrr83hu0q	rueckmto@DEBSCTC0098	SELECT goodsInPos
8,392,022	240,880	34.84	4.12	44.51	101,3	0	fvgsk34swhm9b	blumca@chluzctc1431	SELECT tuId, tuNo, t
6,211,310	240,891	25.78	3.05	14.68	102,5	0	70rxm16vmf54r	blumca@chluzctc1431	SELECT LU.SKUID F
6,071,773	7	867,396.14	2.98	23.96	99,7	0	d5msj347tupum	cukicjo@chluzcwm1108	SELECT colliId, ware
4,886,489	11,742	416.15	2.40	13.36	95,2	,2	brz36vgcc762v	iWACS	SELECT clientId, flag
4,209,341	60	70,155.68	2.07	17.45	100,8	0	g7s9p5dyx5uu2	iWACS	SELECT clientIdPrj, v
4,182,909	59,693	70.07	2.06	11.52	99,8	0	ag48gvq4rf6fb	iWACS	SELECT clientId, dtsI
3,341,004	63	53,031.81	1.64	12.23	99,7	0	b2q63rnz0xts1	iselich@chluzcwm1181	SELECT warehouseS
3,022,973	133	22,729.12	1.49	84.83	70,2	30,2	2301utna4pd58	nedderra@DEBSCTC0099	SELECT storageLoca
2,623,385	2,186	1,200.08	1.29	7.27	103,4	0	aj99nfbdyuvgt	blumca@chluzctc1431	SELECT tuId, storage
2,456,341	821,471	2.99	1.21	16.59	94,1	0	0xfa1r30mk0v3	iWACS	SELECT rackId, rackI
2,423,654	294	8,243.72	1.19	7.06	100,2	0	96v0ussgxx860	broemeha@DEBSCTC0139	SELECT goodsInId, v

Buffer Gets cause CPU load
We have two SQLs, looking similar, causing 40% gets
Check their full SQL text
Ask: Can they be related to $PROBLEM ?

Reading a Report

performing databases
Your reliability. Our concern.

SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 29.4% of Total DB Time (s): 3,114
- Captured PL/SQL account for 0.3% of Total DB Time (s): 3,114

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
180.92	294	0.62	5.81	98.45	0.00	c5bpttzczmj5r	iWACS	SELECT mailingId, mailingNo, s...
172.27	290	0.59	5.53	99.88	0.00	403dzpcdzg3pt	iWACS	SELECT mailingId, mailingNo, s...
84.83	133	0.64	2.72	70.21	30.15	2301utna4pd58	nedderra@DEBSCTC0099	SELECT storageLocationId, rack...
44.51	240,880	0.00	1.43	101.32	0.00	fvgsk34swhm9b	blumca@chluzctc1431	SELECT tuId, tuNo, tuTypeId, x...
23.96	7	3.42	0.77	99.69	0.00	d5msj347tupum	cukicjo@chluzcwm1108	SELECT colliId, warehouseSiteI...
19.38	810,384	0.00	0.62	97.30	0.00	1abbugqmwaswd	iWACS	SELECT tuTypeId, tuTypeGroup, ...
17.45	60	0.29	0.56	100.82	0.00	g7s9p5dyx5uu2	iWACS	SELECT clientIdPrj, warehouseS...
16.63	121	0.14	0.53	99.51	0.00	drzjvrr83hu0q	rueckmto@DEBSCTC0098	SELECT goodsInPosId, warehouse...
16.59	821,471	0.00	0.53	94.11	0.00	0xfa1r30mk0v3	iWACS	SELECT rackId, rackNo, warehou...
16.57	29,027	0.00	0.53	81.46	18.92	b5q7njhqacg4c	langan@chluzctc1485	INSERT INTO SnJournalPrj (snJo...

Our two Top-Get SQLs are fast

But 'd5msj…' runs >3sec - bad if interactive
Ask: Is it part of the $PROBLEM?

Reading a Report

performing databases
Your reliability. Our concern.

Also checking

- SQL executions
- Shared memory usage / Version Count (think: Child Cursors)
- Instance Parameters
- NLS settings
- Specials (Queues etc.)

Elaborate

My #1 Tool

performing databases
Your reliability. Our concern.

```
SELECT *
FROM table(
          DBMS_XPLAN.DISPLAY_CURSOR(
              '&&SQL_ID',
              null,
              'COST,IOSTATS,LAST,ADVANCED,ADAPTIVE'
              )
          );
```

Execution Plan

Example -1-

Finding the Culprit

Real-life example, Mar 13, 2016, Switzerland

performing databases
Your reliability. Our concern.

SQL ordered by Gets

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- %Total - Buffer Gets as a percentage of Total Buffer Gets
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Total Buffer Gets: 203,500,151
- Captured SQL account for 80.9% of Total

Buffer Gets	Executions	Gets per Exec	%Total	Elapsed Time (s)	%CPU	%IO	SQL Id	SQL Module	SQL Text
39,836,966	294	135,499.88	19.58	180.92	98,4	0	c5bpttzczmj5r	iWACS	SELECT mailingId, mailingNo, s...
39,161,524	290	135,039.74	19.24	172.27	99,9	0	403dzpcdzg3pt	iWACS	SELECT mailingId, mailingNo, s...
9,589,820	121	79,254.71	4.71	16.63	99,5	0	drzjvrr83hu0q	rueckmto@DEBSCTC0098	SELECT goodsInPosId, warehouse...
8,392,022	240,880	34.84	4.12	44.51	101,3	0	fvgsk34swhm9b	blumca@chluzctc1431	SELECT tuId, tuNo, tuTypeId, x...
6,211,310	240,891	25.78	3.05	14.68	102,5	0	70rxm16vmf54r	blumca@chluzctc1431	SELECT LU.SKUID FROM TU SHUTTL...
6,071,773	7	867,396.14	2.98	23.96	99,7	0	d5msj347tupum	cukicjo@chluzcwm1108	SELECT collId, warehouseSiteI...
4,886,489	11,742	416.15	2.40	13.36	95,2	,2	brz36vgcc762v	iWACS	SELECT clientId, flagLuDispose...
4,209,341	60	70,155.68	2.07	17.45	100,8	0	g7s9p5dyx5uu2	iWACS	SELECT clientIdPrj, warehouseS...
4,182,909	59,693	70.07	2.06	11.52	99,8	0	ag48gvq4rf6fb	iWACS	SELECT clientId, dtsRestrictio...
3,341,004	63	53,031.81	1.64	12.23	99,7	0	b2q63rnz0xts1	iselich@chluzcwm1181	SELECT warehouseSiteId, pickin...
3,022,973	133	22,729.12	1.49	84.83	70,2	30,2	2301utna4pd58	nedderra@DEBSCTC0099	SELECT storageLocationId, rack...
2,623,385	2,186	1,200.08	1.29	7.27	103,4	0	aj99nfbdyuvgt	blumca@chluzctc1431	SELECT tuId, storageLocationId...
2,456,341	821,471	2.99	1.21	16.59	94,1	0	0xfa1r30mk0v3	iWACS	SELECT rackId, rackNo, warehou...
2,423,654	294	8,243.72	1.19	7.06	100,2	0	96v0ussgxx860	broemeha@DEBSCTC0139	SELECT goodsInId, warehouseSit...

Examining the Culprit

performing databases
Your reliability. Our concern.

```
SQL_ID  c5bpttzczmj5r, child number 0
-------------------------------------
SELECT mailingId, mailingNo, senderId, invoiceRecipientId, recipientId,
...

Plan hash value: 2681284623
```

Id	Operation	Name	E-Rows	E-Bytes	E-Temp	Cost (%CPU)	E-Time
0	SELECT STATEMENT					33988 (100)	
* 1	FILTER						
2	SORT GROUP BY		10204	5829K	122M	33988 (1)	00:06:48
* 3	HASH JOIN		204K	113M	17M	24307 (1)	00:04:52
4	TABLE ACCESS FULL	MAILINGPOS	824K	8052K		15425 (1)	00:03:06
* 5	TABLE ACCESS BY INDEX ROWID	MAILING	48469	26M		6674 (1)	00:01:21
* 6	INDEX RANGE SCAN	XIF2MAILINGPRJ	48521			393 (0)	00:00:05

```
Peeked Binds (identified by position):
--------------------------------------

   1 - (NUMBER): 109
   2 - (VARCHAR2(30), CSID=873): 'y'

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter(SUM("QTYORDERED")>:3)
   3 - access("MP"."MAILINGID"="MAILING"."MAILINGID")
   5 - filter(NLSSORT("FLAGDISPATCHNOTEPRJ",'nls_sort=''BINARY_CI''')=NLSSORT(:2,'nls_sort=''BINAR
           Y_CI'''))
   6 - access("CLIENTIDPRJ"=:1)
```

Improve!

performing databases
Your reliability. Our concern.

Create two-column, function based index

```
Create index I_MAILING_TUNING_4
    on MAILING (
            NLSSORT("FLAGDISPATCHNOTEPRJ",'nls_sort="BINARY_CI"'),
            CLIENTIDPRJ
        );
```

Check back

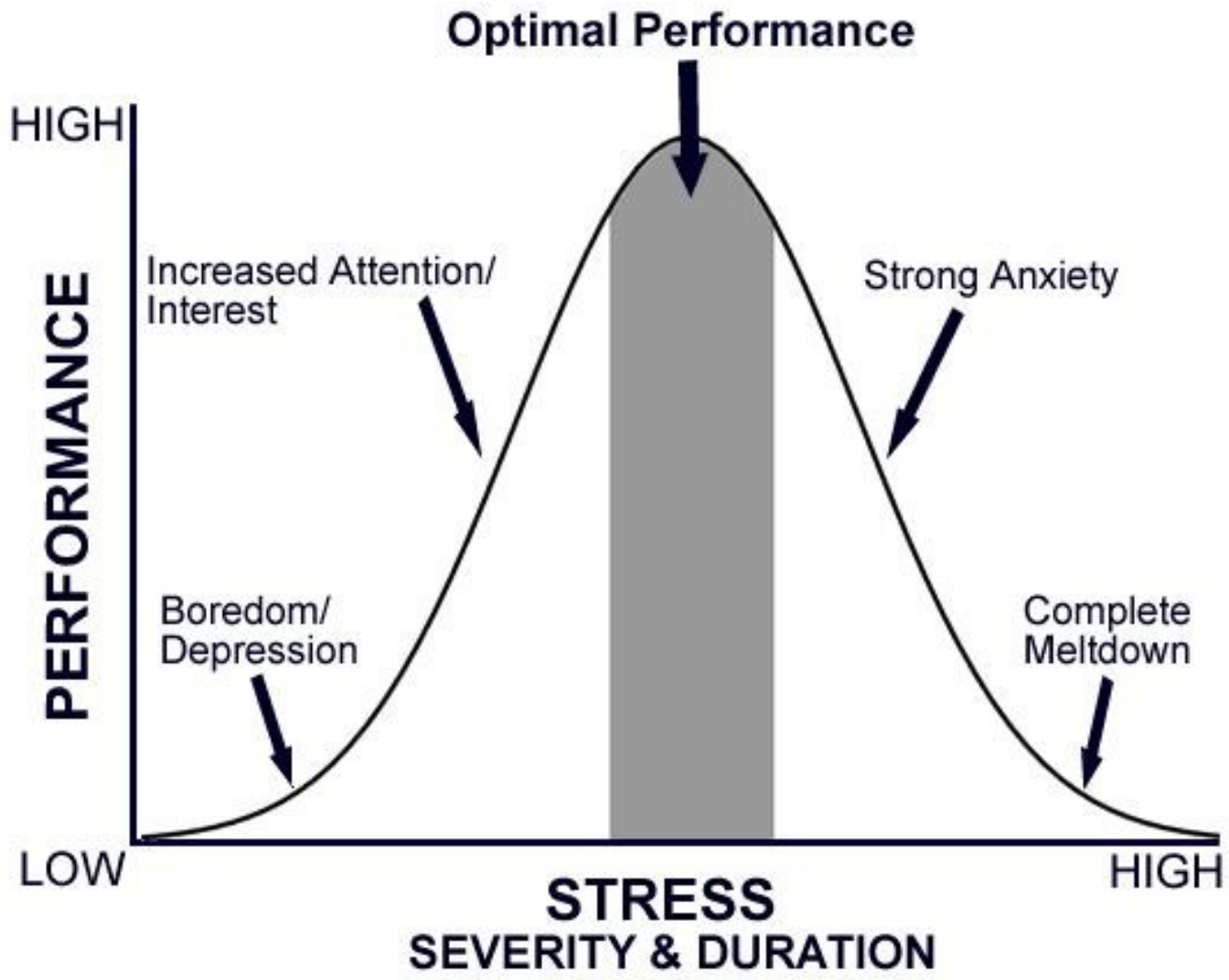
performing databases
Your reliability. Our concern.

```
SQL_ID  c5bpttzczmj5r, child number 0
-------------------------------------
SELECT mailingId, mailingNo, senderId, invoiceRecipientId, recipientId,
...

Plan hash value: 2016964577
-----------------------------------------------------------------------------------------------------------
| Id  | Operation                           | Name                | E-Rows |E-Bytes| Cost (%CPU)| E-Time   |
-----------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |                     |        |       |     8 (100)|          |
|*  1 |  FILTER                             |                     |        |       |            |          |
|   2 |   SORT GROUP BY                     |                     |      1 |   600 |     8  (13)| 00:00:01 |
|   3 |    NESTED LOOPS                     |                     |      1 |   600 |     7   (0)| 00:00:01 |
|   4 |     TABLE ACCESS BY INDEX ROWID     | MAILING             |      1 |   590 |     4   (0)| 00:00:01 |
|*  5 |      INDEX RANGE SCAN               | I_MAILING_TUNING_4  |      1 |       |     3   (0)| 00:00:01 |
|   6 |     TABLE ACCESS BY INDEX ROWID     | MAILINGPOS          |      4 |    40 |     3   (0)| 00:00:01 |
|*  7 |      INDEX RANGE SCAN               | XIF1MAILINGPOS      |      4 |       |     2   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------

Peeked Binds (identified by position):
--------------------------------------

   1 - (NUMBER): 401
   2 - (VARCHAR2(30), CSID=873): 'y'

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter(SUM("QTYORDERED")>:3)
   5 - access("MAILING"."SYS_NC00090$"=NLSSORT(:2,'nls_sort=''BINARY_CI''') AND
              "CLIENTIDPRJ"=:1)
   7 - access("MP"."MAILINGID"="MAILING"."MAILINGID")
```

Improvement

performing databases
Your reliability. Our concern.

Estimated Rows	824k	to	4
Estimated Bytes	113M	to	500
Estimated Temp	122M	to	0
Estimated Time	6:48s	to	0:01s
Real BG / Exec	135k	to	1k

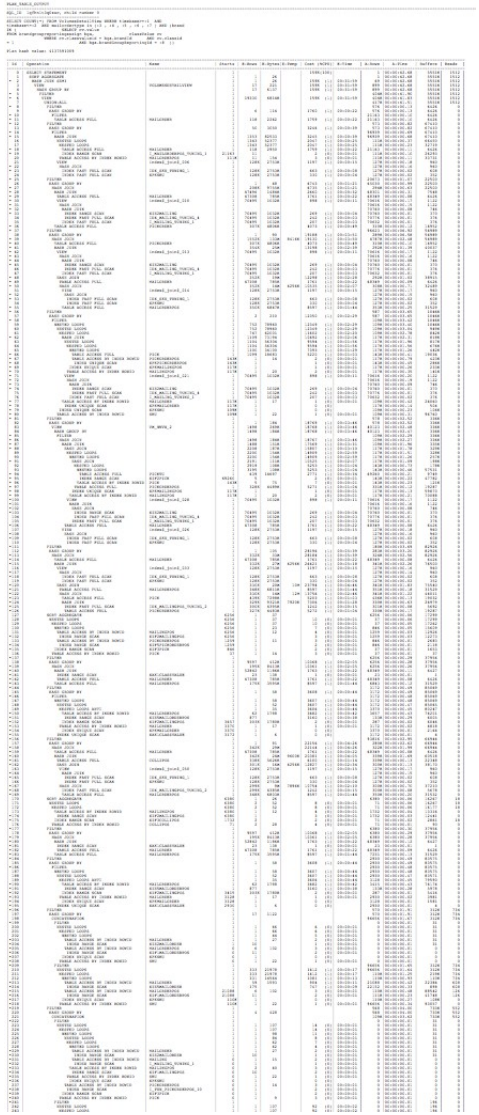

Optimal Performance
HIGH
PERFORMANCE
LOW
Increased Attention/
Interest
Strong Anxiety
Boredom/
Depression
Complete
Meltdown
HIGH
STRESS
SEVERITY & DURATION

Execution Plan

Example -2-

Impressions

Real-life example, Mar 23, 2016, United Kingdom



```
SQL_ID  1qfhn1t3q0xac, child number 0
-------------------------------------
SELECT COUNT(*) FROM VolumesDetailView WHERE timebase>=:1  AND
timebase<=:2  AND mailordertype in (:3 , :4 , :5 , :6 , :7 ) AND (brand
IN (                SELECT cv.value
FROM brandgroupreportingassign bga,                classValue cv
                WHERE cv.classvalueid = bga.brandId       AND cv.classid
= 1                         AND bga.brandGroupReportingId = :8  ))

Plan hash value: 4137591009
```

Id	Operation	Name	Starts	E-Rows	E-Bytes	E-Temp	Cost (%CPU)	E-Time	A-Rows	A-Time	Buffers	Reads
0	SELECT STATEMENT		1				159K(100)		1	00:00:42.68	5551K	1512
1	SORT AGGREGATE		1	1	26				1	00:00:42.68	5551K	1512
* 2	HASH JOIN SEMI		1	1	26		159K (1)	00:31:59	69	00:00:42.68	5551K	1512
3	VIEW	VOLUMESDETAILVIEW	1	17	221		159K (1)	00:31:59	899	00:00:42.68	5551K	1512
4	HASH GROUP BY		1	17	6137		159K (1)	00:31:59	899	00:00:42.68	5551K	1512
* 5	FILTER		1						404K	00:00:41.90	5551K	1512
* 6	VIEW		1	19330	6814K		159K (1)	00:31:59	404K	00:00:41.83	5551K	1512
7	UNION-ALL		1						417K	00:00:41.51	5551K	1512
* 8	FILTER		1						576	00:00:00.13	6426	0
9	HASH GROUP BY		1	6	114		1760 (1)	00:00:22	576	00:00:00.13	6426	0
* 10	FILTER		1						21163	00:00:00.10	6426	0
* 11	TABLE ACCESS FULL	MAILORDER	1	118	2242		1759 (1)	00:00:22	21163	00:00:00.10	6426	0
* 12	FILTER		1						573	00:00:00.82	67410	0
13	HASH GROUP BY		1	50	3050		3246 (1)	00:00:39	573	00:00:00.82	67410	0
* 14	FILTER		1						96939	00:00:00.69	67410	0
* 15	HASH JOIN		1	1353	82533		3245 (1)	00:00:39	96939	00:00:00.68	67410	0
16	NESTED LOOPS		1	1343	52377		2047 (1)	00:00:25	111K	00:00:00.37	66450	0
17	NESTED LOOPS		1	1343	52377		2047 (1)	00:00:25	111K	00:00:00.23	32719	0
* 18	TABLE ACCESS FULL	MAILORDER	1	118	2950		1759 (1)	00:00:22	21163	00:00:00.11	6426	0
* 19	INDEX RANGE SCAN	I_MAILORDERPOS_TUNING_1	21163	11			2 (0)	00:00:01	111K	00:00:00.10	26293	0
20	TABLE ACCESS BY INDEX ROWID	MAILORDERPOS	111K	11	154		3 (0)	00:00:01	111K	00:00:00.11	33731	0
21	VIEW	index$_join$_006	1	128K	2753K		1197 (1)	00:00:15	127K	00:00:00.18	960	0
* 22	HASH JOIN		1						127K	00:00:00.16	960	0
* 23	INDEX FAST FULL SCAN	IDX_SKU_TUNING_1	1	128K	2753K		663 (1)	00:00:08	127K	00:00:00.02	608	0
24	INDEX FAST FULL SCAN	XPKSKU	1	128K	2753K		330 (1)	00:00:04	127K	00:00:00.02	352	0
* 25	FILTER		1						20673	00:00:01.07	22500	0
26	HASH GROUP BY		1	1	48		6743 (1)	00:01:21	65030	00:00:00.99	22500	0
* 27	HASH JOIN		1	208K	9755K		6735 (1)	00:01:21	294K	00:00:00.63	22500	0
* 28	HASH JOIN		1	47496	1484K		2660 (1)	00:00:32	68301	00:00:00.31	7548	0
* 29	TABLE ACCESS FULL	MAILORDER	1	47308	785K		1761 (1)	00:00:22	68349	00:00:00.08	6426	0
* 30	VIEW	index$_join$_010	1	70495	1032K		898 (1)	00:00:11	70616	00:00:00.17	1122	0
* 31	HASH JOIN		1						70616	00:00:00.15	1122	0
* 32	HASH JOIN		1						70760	00:00:00.08	746	0
* 33	INDEX RANGE SCAN	XIE2MAILING	1	70495	1032K		269 (1)	00:00:04	70760	00:00:00.01	370	0
34	INDEX FAST FULL SCAN	IDX_MAILING_TUNING_4	1	70495	1032K		242 (1)	00:00:03	70776	00:00:00.01	376	0
35	INDEX FAST FULL SCAN	I_MAILING_TUNING_1	1	70495	1032K		207 (1)	00:00:03	70632	00:00:00.01	376	0
* 36	TABLE ACCESS FULL	PICKORDER	1	307K	4806K		4073 (1)	00:00:49	310K	00:00:00.12	14952	0
* 37	FILTER		1						96623	00:00:06.92	54989	0
38	HASH GROUP BY		1	1	90		19188 (1)	00:03:51	289K	00:00:06.58	54989	0
* 39	HASH JOIN		1	1552K	133M	8416K	19130 (1)	00:03:50	4787K	00:00:02.48	54989	0
* 40	TABLE ACCESS FULL	PICKORDER	1	307K	4806K		4073 (1)	00:00:49	310K	00:00:00.10	14952	0
* 41	HASH JOIN		1	354K	25M		13198 (1)	00:02:39	292K	00:00:01.39	40037	0
* 42	VIEW	index$_join$_013	1	70495	1032K		898 (1)	00:00:11	70616	00:00:00.17	1122	0
* 43	HASH JOIN		1						70616	00:00:00.16	1122	0
* 44	HASH JOIN		1						70760	00:00:00.08	746	0
* 45	INDEX RANGE SCAN	XIE2MAILING	1	70495	1032K		269 (1)	00:00:04	70760	00:00:00.01	370	0
46	INDEX FAST FULL SCAN	IDX_MAILING_TUNING_4	1	70495	1032K		242 (1)	00:00:03	70776	00:00:00.01	376	0
47	INDEX FAST FULL SCAN	I_MAILING_TUNING_1	1	70495	1032K		207 (1)	00:00:03	70632	00:00:00.01	376	0
* 48	HASH JOIN		1	352K	19M		12298 (1)	00:02:28	292K	00:00:01.02	38915	0
* 49	TABLE ACCESS FULL	MAILORDER	1	47308	785K		1761 (1)	00:00:22	68349	00:00:00.09	6426	0
* 50	HASH JOIN		1	352K	14M	4256K	10535 (1)	00:02:07	308K	00:00:00.71	32489	0
51	VIEW	index$_join$_014	1	128K	2753K		1197 (1)	00:00:15	127K	00:00:00.17	960	0
* 52	HASH JOIN		1						127K	00:00:00.15	960	0

Expensive

performing databases

```
SQL_ID  1qfhn1t3q0xac, child number 0
-------------------------------------
SELECT COUNT(*) FROM VolumesDetailView WHERE timebase>=:1  AND
timebase<=:2  AND mailordertype in (:3 , :4 , :5 , :6 , :7 ) AND (brand
IN (                SELECT cv.value
FROM brandgroupreportingassign bga,              classValue cv
               WHERE cv.classvalueid = bga.brandId      AND cv.classid
= 1                             AND bga.brandGroupReportingId = :8  ))

Plan hash value: 413759109
```

5M Buffer Gets
42 seconds

Id	Operation	Name	Starts	E-Rows	E-Bytes	E-Temp	Cost (%CPU)	E-Time	A-Rows	A-Time	Buffers	Reads
0	SELECT STATEMENT		1				159K(100)		1	00:00:42.68	5551K	1512
1	SORT AGGREGATE		1	1	26				1	00:00:42.68	5551K	1512
* 2	HASH JOIN SEMI		1	1	26		159K (1)	00:31:59	69	00:00:42.68	5551K	1512
3	VIEW	VOLUMESDETAILVIEW	1	17	221		159K (1)	00:31:59	899	00:00:42.68	5551K	1512
4	HASH GROUP BY		1	17	6137		159K (1)	00:31:59	899	00:00:42.68	5551K	1512
* 5	FILTER		1						404K	00:00:41.90	5551K	1512
* 6	VIEW		1	19330	6814K		159K (1)	00:31:59	404K	00:00:41.83	5551K	1512
7	UNION-ALL		1						417K	00:00:41.51	5551K	1512
* 8	FILTER		1						576	00:00:00.13	6426	0
9	HASH GROUP BY		1	6	114		1760 (1)	00:00:22	576	00:00:00.13	6426	0
* 10	FILTER		1						21163	00:00:00.10	6426	0
* 11	TABLE ACCESS FULL	MAILORDER	1	118	2242		1759 (1)	00:00:22	21163	00:00:00.10	6426	0
* 12	FILTER		1						573	00:00:00.82	67410	0
13	HASH GROUP BY		1	50	3050		3246 (1)	00:00:39	573	00:00:00.82	67410	0
* 14	FILTER		1						96939	00:00:00.69	67410	0
* 15	HASH JOIN		1	1353	82533		3245 (1)	00:00:39	96939	00:00:00.68	67410	0
16	NESTED LOOPS		1	1343	52377		2047 (1)	00:00:25	111K	00:00:00.37	66450	0
17	NESTED LOOPS		1	1343	52377		2047 (1)	00:00:25	111K	00:00:00.23	32719	0
* 18	TABLE ACCESS FULL	MAILORDER	1	118	2950		1759 (1)	00:00:22	21163	00:00:00.11	6426	0
* 19	INDEX RANGE SCAN	I_MAILORDERPOS_TUNING_1	21163	11			2 (0)	00:00:01	111K	00:00:00.10	26293	0
20	TABLE ACCESS BY INDEX ROWID	MAILORDERPOS	111K	11	154		3 (0)	00:00:01	111K	00:00:00.11	33731	0
21	VIEW	index$_join$_006	1	128K	2753K		1197 (1)	00:00:15	127K	00:00:00.18	960	0
* 22	HASH JOIN		1						127K	00:00:00.16	960	0
* 23	INDEX FAST FULL SCAN	IDX_SKU_TUNING_1	1	128K	2753K		663 (1)	00:00:08	127K	00:00:00.02	608	0
24	INDEX FAST FULL SCAN	XPKSKU	1	128K	2753K		330 (1)	00:00:04	127K	00:00:00.02	352	0
* 25	FILTER		1						20673	00:00:01.07	22500	0
26	HASH GROUP BY		1	1	48		6743 (1)	00:01:21	65030	00:00:00.99	22500	0
* 27	HASH JOIN		1	208K	9755K		6735 (1)	00:01:21	294K	00:00:00.63	22500	0
* 28	HASH JOIN		1	47496	1484K		2660 (1)	00:00:32	68301	00:00:00.31	7548	0
* 29	TABLE ACCESS FULL	MAILORDER	1	47308	785K		1761 (1)	00:00:22	68349	00:00:00.08	6426	0
* 30	VIEW	index$_join$_010	1	70495	1032K		898 (1)	00:00:11	70616	00:00:00.17	1122	0
* 31	HASH JOIN		1						70616	00:00:00.15	1122	0
* 32	HASH JOIN		1						70760	00:00:00.08	746	0
* 33	INDEX RANGE SCAN	XIE2MAILING	1	70495	1032K		269 (1)	00:00:04	70760	00:00:00.01	370	0
34	INDEX FAST FULL SCAN	IDX_MAILING_TUNING_4	1	70495	1032K		242 (1)	00:00:03	70776	00:00:00.01	376	0
35	INDEX FAST FULL SCAN	I_MAILING_TUNING_1	1	70495	1032K		207 (1)	00:00:03	70632	00:00:00.01	376	0
* 36	TABLE ACCESS FULL	PICKORDER	1	307K	4806K		4073 (1)	00:00:49	310K	00:00:00.12	14952	0
* 37	FILTER		1						96623	00:00:06.92	54989	0
38	HASH GROUP BY		1	1	90		19188 (1)	00:03:51	289K	00:00:06.58	54989	0
* 39	HASH JOIN		1	1552K	133M	8416K	19130 (1)	00:03:50	4787K	00:00:02.48	54989	0
* 40	TABLE ACCESS FULL	PICKORDER	1	307K	4806K		4073 (1)	00:00:49	310K	00:00:00.10	14952	0
* 41	HASH JOIN		1	354K	25M		13198 (1)	00:02:39	292K	00:00:01.39	40037	0
* 42	VIEW	index$_join$_013	1	70495	1032K		898 (1)	00:00:11	70616	00:00:00.17	1122	0
* 43	HASH JOIN		1						70616	00:00:00.16	1122	0
* 44	HASH JOIN		1						70760	00:00:00.08	746	0
* 45	INDEX RANGE SCAN	XIE2MAILING	1	70495	1032K		269 (1)	00:00:04	70760	00:00:00.01	370	0
46	INDEX FAST FULL SCAN	IDX_MAILING_TUNING_4	1	70495	1032K		242 (1)	00:00:03	70776	00:00:00.01	376	0
47	INDEX FAST FULL SCAN	I_MAILING_TUNING_1	1	70495	1032K		207 (1)	00:00:03	70632	00:00:00.01	376	0
* 48	HASH JOIN		1	352K	19M		12298 (1)	00:02:28	292K	00:00:01.02	38915	0
* 49	TABLE ACCESS FULL	MAILORDER	1	47308	785K		1761 (1)	00:00:22	68349	00:00:00.09	6426	0
* 50	HASH JOIN		1	352K	14M	4256K	10535 (1)	00:02:07	308K	00:00:00.71	32489	0
51	VIEW	index$_join$_014	1	128K	2753K		1197 (1)	00:00:15	127K	00:00:00.17	960	0
* 52	HASH JOIN		1						127K	00:00:00.15	960	0

Deeper

performing databases
Your reliability. Our concern.

Index Join over 3 indexes

* 37	FILTER		1						96623	00:00:06.92	54989	0
38	HASH GROUP BY		1	1	90		19188 (1)	00:03:51	289K	00:00:06.58	54989	0
* 39	HASH JOIN		1	1552K	133M	8416K	19130 (1)	00:03:50	4787K	00:00:02.48	54989	0
* 40	TABLE ACCESS FULL	PICKORDER	1	307K	4806K		4073 (1)	00:00:49	310K	00:00:00.10	14952	0
* 41	HASH JOIN		1	354K	25M		13198 (1)	00:02:39	292K	00:00:01.39	40037	0
* 42	VIEW	index$_join$_013	1	70495	1032K		898 (1)	00:00:11	70616	00:00:00.17	1122	0
* 43	HASH JOIN		1						70616	00:00:00.16	1122	0
* 44	HASH JOIN		1						70760	00:00:00.08	746	0
* 45	INDEX RANGE SCAN	XIE2MAILING	1	70495	1032K		269 (1)	00:00:04	70760	00:00:00.01	370	0
46	INDEX FAST FULL SCAN	IDX_MAILING_TUNING_4	1	70495	1032K		242 (1)	00:00:03	70776	00:00:00.01	376	0
47	INDEX FAST FULL SCAN	I_MAILING_TUNING_1	1	70495	1032K		207 (1)	00:00:03	70632	00:00:00.01	376	0
* 48	HASH JOIN		1	352K	19M		12298 (1)	00:02:28	292K	00:00:01.02	38915	0
* 49	TABLE ACCESS FULL	MAILORDER	1	47308	785K		1761 (1)	00:00:22	68349	00:00:00.09	6426	0
* 50	HASH JOIN		1	352K	14M	4256K	10535 (1)	00:02:07	308K	00:00:00.71	32489	0
51												0
* 52												0
* 53												0
54												0
* 55												0

```
NLSSORT(MIN("MO"."MAILORDERTYPE"),'nls_sort=''GERMAN''
39 - access("M"."MAILINGID"="PO"."MAILINGID")
40 - filter(("PO"."MAILINGID" IS NOT NULL AND "PO"."STATUS"<=99))
41 - access("MO"."MAILORDERID"="M"."MAILORDERIDPRJ")
42 - filter("M"."STATUS"<=99)
43 - access(ROWID=ROWID)
44 - access(ROWID=ROWID)
45 - access("M"."STATUS"<=99)
48 - access("MO"."MAILORDERID"="MOP"."MAILORDERID")
```

Indexes

performing databases
Your reliability. Our concern.

Index Join over 3 indexes
+ one useless column

INDEX_NAME	COLUMNS
XPKMAILING	MAILINGID
XAK1MAILING	MAILINGNO
XIE2MAILING	STATUS
XIE3MAILING	STATUS, FLAGREADYTODISPOSEPRJ
I_MAILING_TUNING_1	MAILORDERIDPRJ
I_MAILING_TUNING_2	MAILINGID, STATUS
I_MAILING_TUNING_5	MAILINGID, MAILINGNO
IDX_MAILING_TUNING_3	SYS_NC00040$
IDX_MAILING_TUNING_4	MAILINGID, WAREHOUSESITEIDPRJ
I_MAILING_MAILINGTYPE	MAILINGTYPE

```
              NLSSORT(MIN("MO"."MAILORDERTYPE"),'nls_sort=''GERMAN''
39 - access("M"."MAILINGID"="PO"."MAILINGID")
40 - filter(("PO"."MAILINGID" IS NOT NULL AND "PO"."STATUS"<=99))
41 - access("MO"."MAILORDERID"="M"."MAILORDERIDPRJ")
42 - filter("M"."STATUS"<=99)
43 - access(ROWID=ROWID)
44 - access(ROWID=ROWID)
45 - access("M"."STATUS"<=99)
48 - access("MO"."MAILORDERID"="MOP"."MAILORDERID")
```

Indexes

One big 3-column index instead

INDEX_NAME	COLUMNS
XPKMAILING	MAILINGID
XAK1MAILING	MAILINGNO
XIE2MAILING	STATUS
XIE3MAILING	STATUS, FLAGREADYTODISPOSEPRJ
I_MAILING_TUNING_1	MAILORDERIDPRJ
I_MAILING_TUNING_2	MAILINGID, STATUS
I_MAILING_TUNING_5	MAILINGID, MAILINGNO
I_MAILING_TUNING_6	MAILINGID, MAILORDERIDPRJ, STATUS
IDX_MAILING_TUNING_3	SYS_NC00040$
IDX_MAILING_TUNING_4	MAILINGID, WAREHOUSESITEIDPRJ
I_MAILING_MAILINGTYPE	MAILINGTYPE

```
30 - filter((TRUNC(MIN("PO"."GENDAT"))>=:1 AND TRUNC(MIN("PO"."GENDAT"))<=:2 .
32 - access("M"."MAILINGID"="PO"."MAILINGID")
33 - filter(("PO"."MAILINGID" IS NOT NULL AND "PO"."STATUS"<=99))
34 - access("MO"."MAILORDERID"="M"."MAILORDERIDPRJ")
35 - filter("M"."STATUS"<=99)
36 - access("MO"."MAILORDERID"="MOP"."MAILORDERID")
37 - filter((INTERNAL_FUNCTION("MO"."MAILORDERTYPE") AND "MO"."STATUS"<=99))
38 - access("MOP"."SKUID"="SKU"."SKUID")
40 - access(ROWID=ROWID)
41 - filter("SKU"."SKUNO"<>'PRINTSKU')
43 - filter("MOP"."STATUS"<=99)
45 - filter(TO_DATE(:2)>=TO_DATE(:1))
```

```
|* 30 |        FILTER                   |                      |     1 |        |        |        |             |          |    251K|00:00:04.11 |   54029 |      0 |
|  31 |         HASH GROUP BY           |                      |     1 |     10 |    810 |        | 18043    (1)| 00:03:37 |    290K|00:00:03.62 |   54029 |      0 |
|* 32 |          HASH JOIN              |                      |     1 |  1552K|   119M|  8416K| 17985    (1)| 00:03:36 |   4799K|00:00:02.14 |   54029 |      0 |
|* 33 |           TABLE ACCESS FULL     | PICKORDER            |     1 |   307K|  4806K|        |  4073    (1)| 00:00:49 |    311K|00:00:00.10 |   14952 |      0 |
|* 34 |           HASH JOIN             |                      |     1 |   354K|    21M|        | 12203    (1)| 00:02:27 |    293K|00:00:01.10 |   39077 |      0 |
|* 35 |            INDEX FAST FULL SCAN | I_MAILING_TUNING_6   |     1 |  70495 |  1032K|        |    72    (2)| 00:00:01 |  70893 |00:00:00.01 |     264 |      0 |
|* 36 |            HASH JOIN            |                      |     1 |   352K|    16M|        | 12130    (1)| 00:02:26 |    293K|00:00:00.90 |   38813 |      0 |
|* 37 |             TABLE ACCESS FULL   | MAILORDER            |     1 |  68223 |  1132K|        |  1757    (1)| 00:00:22 |  68480 |00:00:00.06 |    6426 |      0 |
|* 38 |             HASH JOIN           |                      |     1 |   352K|    11M|  3136K| 10371    (1)| 00:02:05 |    309K|00:00:00.64 |   32387 |      0 |
|  39 |              VIEW               | index$_join$_014     |     1 |   128K|  1627K|        |  1087    (1)| 00:00:14 |    127K|00:00:00.16 |     858 |      0 |
|* 40 |               HASH JOIN         |                      |     1 |        |        |        |             |          |    127K|00:00:00.14 |     858 |      0 |
|* 41 |                INDEX FAST FULL SCAN | XAK1SKU          |     1 |   128K|  1627K|        |   594    (1)| 00:00:08 |    127K|00:00:00.02 |     506 |      0 |
|  42 |                INDEX FAST FULL SCAN | XPKSKU           |     1 |   128K|  1627K|        |   330    (1)| 00:00:04 |    127K|00:00:00.02 |     352 |      0 |
|* 43 |              TABLE ACCESS FULL  | MAILORDERPOS         |     1 |   350K|  6847K|        |  8597    (1)| 00:01:44 |    354K|00:00:00.22 |   31529 |      0 |
|  44 |        HASH GROUP BY            |                      |     1 |    128 | 12288 |        | 11854    (1)| 00:02:23 |   1463 |00:00:07.14 |   2656K|    785 |
```

Indexes

performing databases
Your reliability. Our concern.

* 37	FILTER		1						96623	00:00:06.92	54989	0
38	HASH GROUP BY		1	1	90		19188	(1)	00:03:51	289K	00:00:06.58	54989	0
* 39	HASH JOIN		1	1552K	133M	8416K	19130	(1)	00:03:50	4787K	00:00:02.48	54989	0
* 40	TABLE ACCESS FULL	PICKORDER	1	307K	4806K		4073	(1)	00:00:49	310K	00:00:00.10	14952	0
* 41	HASH JOIN		1	354K	25M		13198	(1)	00:02:39	292K	00:00:01.39	40037	0
* 42	VIEW	index$_join$_013	1	70495	1032K		898	(1)	00:00:11	70616	00:00:00.17	1122	0
* 43	HASH JOIN		1							70616	00:00:00.16	1122	0
* 44	HASH JOIN		1							70760	00:00:00.08	746	0
* 45	INDEX RANGE SCAN	XIE2MAILING	1	70495	1032K		269	(1)	00:00:04	70760	00:00:00.01	370	0
46	INDEX FAST FULL SCAN	IDX_MAILING_TUNING_4	1	70495	1032K		242	(1)	00:00:03	70776	00:00:00.01	376	0
47	INDEX FAST FULL SCAN	I_MAILING_TUNING_1	1	70495	1032K		207	(1)	00:00:03	70632	00:00:00.01	376	0
* 48	HASH JOIN		1	352K	19M		12298	(1)	00:02:28	292K	00:00:01.02	38915	0
* 49	TABLE ACCESS FULL	MAILORDER	1	47308	785K		1761	(1)	00:00:22	68349	00:00:00.09	6426	0
* 50	HASH JOIN		1	352K	14M	4256K	10535	(1)	00:02:07	308K	00:00:00.71	32489	0
51	VIEW	index$_join$_014	1	128K	2753K		1197	(1)	00:00:15	127K	00:00:00.17	960	0
* 52	HASH JOIN		1							127K	00:00:00.15	960	0
* 53	INDEX FAST FULL SCAN	IDX_SKU_TUNING_1	1	128K	2753K		663	(1)	00:00:08	127K	00:00:00.02	608	0
54	INDEX FAST FULL SCAN	XPKSKU	1	128K	2753K		330	(1)	00:00:04	127K	00:00:00.02	352	0
* 55	TABLE ACCESS FULL	MAILORDERPOS	1	350K	6847K		8597	(1)	00:01:44	353K	00:00:00.25	31529	0

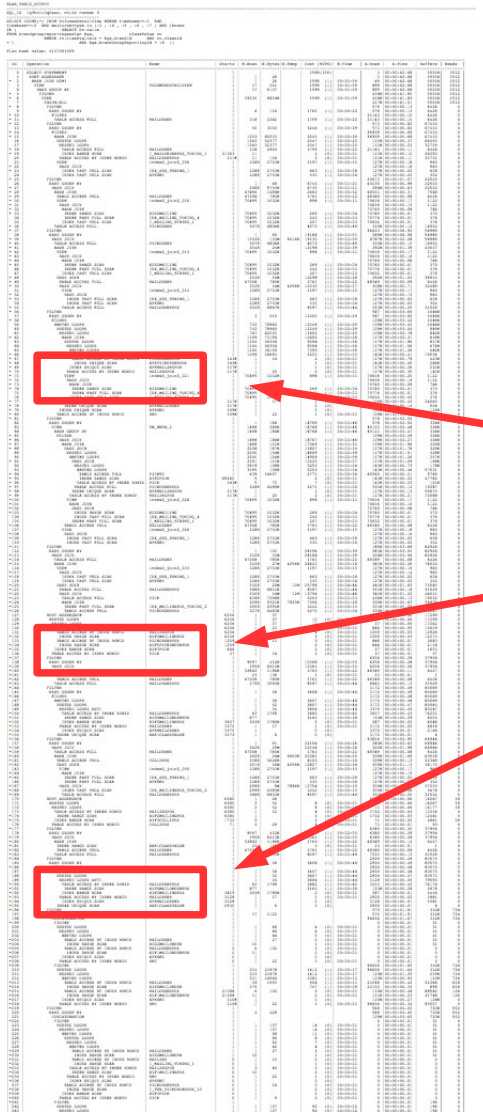
Same cardinality

Buffer Gets - 858

* 30	FILTER		1						251K	00:00:04.11	54029	0
31	HASH GROUP BY		1	10	810		18043	(1)	00:03:37	290K	00:00:03.62	54029	0
* 32	HASH JOIN		1	1552K	119M	8416K	17985	(1)	00:03:36	4799K	00:00:02.14	54029	0
* 33	TABLE ACCESS FULL	PICKORDER	1	307K	4806K		4073	(1)	00:00:49	311K	00:00:00.10	14952	0
* 34	HASH JOIN		1	354K	21M		12203	(1)	00:02:27	293K	00:00:01.10	39077	0
* 35	INDEX FAST FULL SCAN	I_MAILING_TUNING_6	1	70495	1032K		72	(2)	00:00:01	70893	00:00:00.01	264	0
* 36	HASH JOIN		1	352K	16M		12130	(1)	00:02:26	293K	00:00:00.90	38813	0
* 37	TABLE ACCESS FULL	MAILORDER	1	68223	1132K		1757	(1)	00:00:22	68480	00:00:00.06	6426	0
* 38	HASH JOIN		1	352K	11M	3136K	10371	(1)	00:02:05	309K	00:00:00.64	32387	0
39	VIEW	index$_join$_014	1	128K	1627K		1087	(1)	00:00:14	127K	00:00:00.16	858	0
* 40	HASH JOIN		1							127K	00:00:00.14	858	0
* 41	INDEX FAST FULL SCAN	XAK1SKU	1	128K	1627K		594	(1)	00:00:08	127K	00:00:00.02	506	0
42	INDEX FAST FULL SCAN	XPKSKU	1	128K	1627K		330	(1)	00:00:04	127K	00:00:00.02	352	0
* 43	TABLE ACCESS FULL	MAILORDERPOS	1	350K	6847K		8597	(1)	00:01:44	354K	00:00:00.22	31529	0
44	HASH GROUP BY		1	128	12288		11854	(1)	00:02:23	1463	00:00:07.14	2656K	785

A long way ...

Real-life example, Mar 23, 2016, United Kingdom



2 Index Joins
to
1 three-column index
=
1 hour of work

42 seconds => 30 seconds

Do it my way

performing databases
Your reliability. Our concern.

- **Don't wait for the next crash - act NOW!**
- Be yourself
- Simplify your Toolset
- Go down to the second
- Prove your findings

performing databases
Your reliability. Our concern.

DOAG

Conference + Exhibition
#DOAG2016

2000+ Visitors

Nuremberg, Germany
November 15.-18. 2016

performing databases
Your reliability. Our concern.

Oracle Core: 12c InMemory Column Store

Tomorrow
3:30 pm

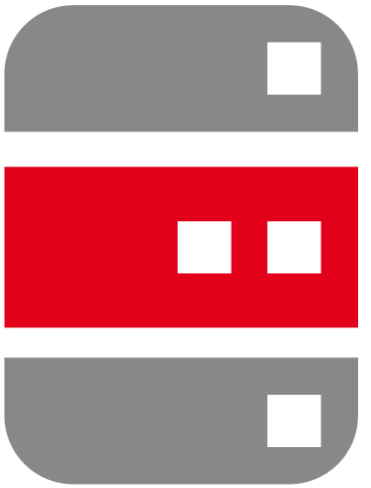
here in Palm-A

performing databases
Your reliability. Our concern.

Q & A

Download my Presentations and Whitepapers
http://www.performing-databases.com

performing databases



Your reliability. Our concern.