

# The Sherlock Holmes Sql Mystery

- 11. – 14.10.2022
- Rovinj, Croatia

Lothar Flatz

Oracle Performance Tuning





## May I introduce myself?

- Over 30 years Oracle Database Experience (starting with Version 5)
- Oak Table Member
- Ex-Real World Performance Group
- Oracle ACE
- Signatur Project: PVSS (CERN)
- Patent US 8103658 B2 together with Björn Engsig



# About DBConcepts GmbH

THE ORACLE EXPERTS



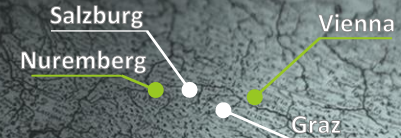
Founded: 2000, in Vienna

DBC Group – AT & GER

Employees : 50

Oracle | Partner

30++ Oracle Product Specialisation



# Just imagine





# The Silver Bullet



# Typical Question on OTN

„We just migrated to a new dev environment in Linux REDHAT5, and now the query is very slow.

I checked toad monitor, it said need to increase db\_buffer\_cache and shared pool too small.“

# The Answer...

---

“If you want more help here, let's start with some DATA. Without this, people here can only guess at the actual cause of your problem, although if enough of us guess for a long enough time, somebody will undoubtedly find the right answer eventually.”

Senior DBA, Calagry





## However

“Why? Guessing is fun. It's like fantasy football, only with other people's databases.”

Senior DBA, New York



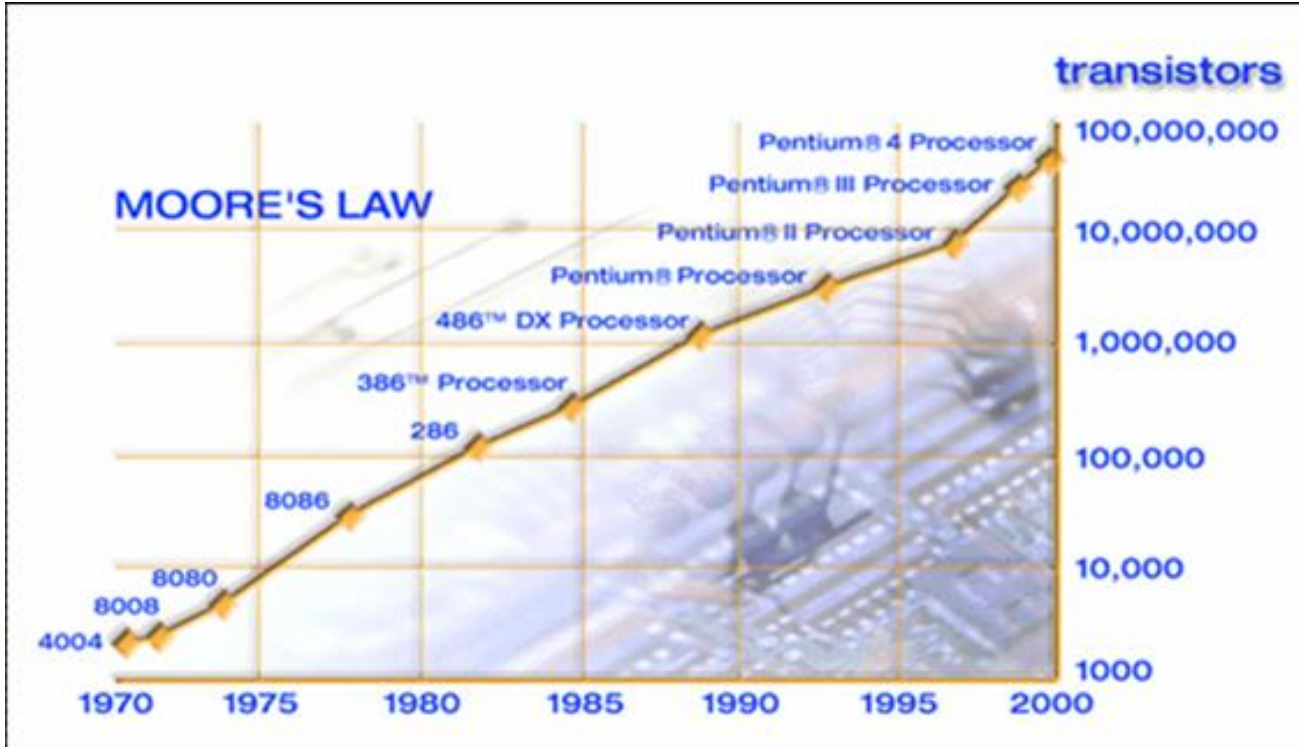
# Expert Level

People tend to see themselves as Experts, when they have anecdotal success.

However: an expert is someone widely recognized as a **reliable** source of technique or skill whose faculty for judging or deciding rightly, justly, or wisely is accorded authority and status by peers or the public in a specific well-distinguished domain.

Wikipedia : [Expert](#)

# Kill it with iron !



# Sherlock Holmes, more than fiction?



Dr. Joseph Bell, The role model  
for Sherlock Holmes

„In which the hero would treat crime as Dr Bell treated disease and where science would take the place of chance.“

(Arthur Conan Doyle)

# Real Case

## Customer on alert

- I/O Subsystem is under heavy load
- Oracle Partner has worked on the issue
- So far without visible success

The junior consultant suggested



**CAN'T GO WRONG WITH THAT:**

- 1. INCREASE SGA**
- 2. RECALCULATE STATISTICS**

**THOUGH IT DID NOT HURT,  
IT WAS NOT HELPFUL  
EITHER ..**



# The Senior Consultants Findings

One statement is the source of all trouble:

SELECT...

We have to ask the application vendor what he changed.

He also confirms that increasing the SGA and new statistics did not change anything.

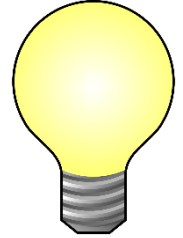
# Can we start now?



By Melbourneian - Self-photographed, CC BY 2.5,  
<https://commons.wikimedia.org/w/index.php?curid=1259804>

Datum: 14.10.2022

# What is the first thing to do?



Francis Bacon,  
Philosoph and Scientist

„Man, being the servant and interpreter of nature, can do and understand **so much and so much only as he has observed** in fact or in thought of the course of nature.“

# Collect Facts first !



„It is a capital mistake to theorize before you have all the evidence. It biases the judgment.“

Sherlock Holmes Quote  
-A Study in Scarlet

“ SHERLOCK HOLMES WAS STANDING SMILING AT ME ACROSS MY STUDY TABLE.”

# Runtime Statistics show the numbers you need

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	Reads
0	SELECT STATEMENT		1		12342	00:00:17.85	306K	237K
1	SORT ORDER BY		1	771	12342	00:00:17.85	306K	237K
2	SORT UNIQUE		1	771	12342	00:00:17.72	306K	237K
3	UNION-ALL		1		19469	00:00:17.68	306K	237K
* 4	HASH JOIN		1	297	1038	00:00:00.91	25441	78
* 5	HASH JOIN		1	298	1038	00:00:00.69	24558	78
* 6	TABLE ACCESS BY INDEX ROWID	MEDI_B	1	298	40557	00:00:00.59	24033	78
* 7	INDEX RANGE SCAN	I2	1	5356	40557	00:00:00.16	105	78
* 8	TABLE ACCESS BY INDEX ROWID	MEDI_C	1	2309	721	00:00:00.01	525	0
* 9	INDEX RANGE SCAN	I1	1	2317	721	00:00:00.01	5	0
10	INDEX FAST FULL SCAN	I3	1	279K	279K	00:00:00.07	883	0
* 11	HASH JOIN		1	474	18431	00:00:16.73	280K	237K
* 12	HASH JOIN		1	475	18431	00:00:16.58	279K	237K
* 13	TABLE ACCESS BY INDEX ROWID	MEDI_C	1	2309	721	00:00:00.01	525	0
* 14	INDEX RANGE SCAN	I1	1	2317	721	00:00:00.01	5	0
* 15	HASH JOIN		1	7202	957K	00:00:16.23	279K	237K
* 16	TABLE ACCESS FULL	MEDI_A	1	7161	957K	00:00:12.78	237K	237K
* 17	TABLE ACCESS FULL	MEDI_B	1	598K	598K	00:00:01.99	41476	0
18	INDEX FAST FULL SCAN	I3	1	279K	279K	00:00:00.04	883	0



# Systematical Approach

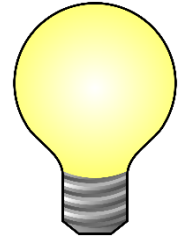
1. Is the Statement optimizable?
2. Where should be my focus?
3. What causes the problem?
4. How to solve the problem?

# Is the statement optimizable?



Martin Berg, Oak Table Member,  
„Tuning Queries by Eliminating Throwaway“  
Internet whitepaper

„Most badly performing queries usually exhibits a high amount of rows processed and then thrown away because they were not needed in the output.“



# Any Throw away ?

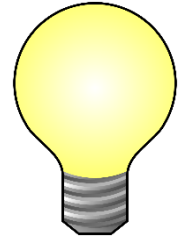
Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	Reads
0	SELECT STATEMENT		1		12342	00:00:17.85	306K	237K
1	SORT ORDER BY		1	771	12342	00:00:17.85	306K	237K
2	SORT UNIQUE		1	771	12342	00:00:17.72	306K	237K
3	UNION-ALL		1		19469	00:00:17.68	306K	237K
* 4	HASH JOIN		1	297	1038	00:00:00.91	25441	78
* 5	HASH JOIN		1	298	1038	00:00:00.69	24558	78
* 6	TABLE ACCESS BY INDEX ROWID	MEDI_B	1	298	40557	00:00:00.59	24033	78
* 7	INDEX RANGE SCAN	I2	1	5356	40557	00:00:00.16	105	78
* 8	TABLE ACCESS BY INDEX ROWID	MEDI_C	1	2309	721	00:00:00.01	525	0
* 9	INDEX RANGE SCAN	I1	1	2317	721	00:00:00.01	5	0
10	INDEX FAST FULL SCAN	I3	1	279	279K	00:00:00.07	883	0
* 11	HASH JOIN		1	474	18431	00:00:16.73	280K	237K
* 12	HASH JOIN		1	475	18431	00:00:16.58	279K	237K
* 13	TABLE ACCESS BY INDEX ROWID	MEDI_C	1	2309	721	00:00:00.01	525	0
* 14	INDEX RANGE SCAN	I1	1	2317	721	00:00:00.01	5	0
* 15	HASH JOIN		1	7202	957K	00:00:16.23	279K	237K
* 16	TABLE ACCESS FULL	MEDI_A	1	7161	957K	00:00:12.78	237K	237K
* 17	TABLE ACCESS FULL	MEDI_B	1	598	598K	00:00:01.99	41476	0
18	INDEX FAST FULL SCAN	I3	1	279	279K	00:00:00.04	883	0

# Where should be my focucs?



Dr. Wolfgang Breitling,  
Oak Table Member,  
Tuning with Cardinality Feedback,  
2006 Hotsos Symposium

„Whenever the CBO chooses a bad plan it can be traced back to an error in the estimation of the cardinality of one or more row sources.“



# Cardinality Feedback

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	Reads
0	SELECT STATEMENT		1		12342	00:00:17.85	306K	237K
1	SORT ORDER BY		1	771	12342	00:00:17.85	306K	237K
2	SORT UNIQUE		1	771	12342	00:00:17.72	306K	237K
3	UNION-ALL		1		19469	00:00:17.68	306K	237K
* 4	HASH JOIN		1	297	1038	00:00:00.91	25441	78
* 5	HASH JOIN		1	298	1038	00:00:00.69	24558	78
* 6	TABLE ACCESS BY INDEX ROWID	MEDI_B	1	298	40557	00:00:00.59	24033	78
* 7	INDEX RANGE SCAN	I2	1	5356	40557	00:00:00.16	105	78
* 8	TABLE ACCESS BY INDEX ROWID	MEDI_C	1	2309	721	00:00:00.01	525	0
* 9	INDEX RANGE SCAN	I1	1	2317	721	00:00:00.01	5	0
10	INDEX FAST FULL SCAN	I3	1	279K	279K	00:00:00.07	883	0
* 11	HASH JOIN		1	474	18431	00:00:16.73	280K	237K
* 12	HASH JOIN		1	475	18431	00:00:16.58	279K	237K
* 13	TABLE ACCESS BY INDEX ROWID	MEDI_C	1	2309	721	00:00:00.01	525	0
* 14	INDEX RANGE SCAN	I1	1	2317	721	00:00:00.01	5	0
* 15	HASH JOIN		1	598K	957K	00:00:16.23	279K	237K
* 16	TABLE ACCESS FULL	MEDI_A	1	7161	957K	00:00:12.78	237K	237K
* 17	TABLE ACCESS FULL	MEDI_B	1	598K	598K	00:00:01.99	41476	0
18	INDEX FAST FULL SCAN	I3	1	279K	279K	00:00:00.04	883	0

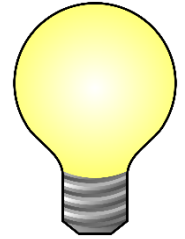


# Focus confirmed



Cary Millsap, Oak Table Member,  
entrepreneur and Oracle software performance  
specialist

„Focus on the step that  
takes the most time.”

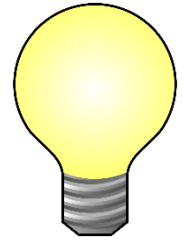


# The prime evil

```
16 - filter (CreationDate>:loaddate  
OR lastChangeDate>:loaddate)  
AND TO_NUMBER(effective(id))=1
```

# Isolation of Variables

When determining what independent variables to change in an experiment, it is very important that you **isolate the effects of each independent variable**. You do not want to change more than one variable at once, for if you do it becomes more difficult to analyze the effects of each change on the dependent variable.



## The Scientific Method/Independent and Dependent Variables

# Estimation Error , first Part

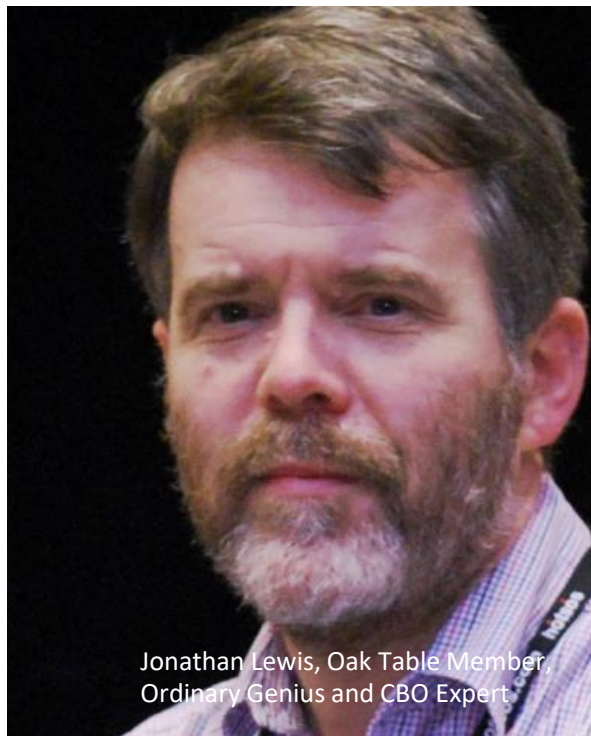
```
16 - filter (CreationDate>:loaddate  
OR lastChangeDate>:loaddate)  
AND TO_NUMBER(effective(id))=1
```

# Result of systematical Tests

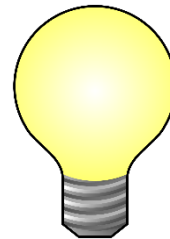
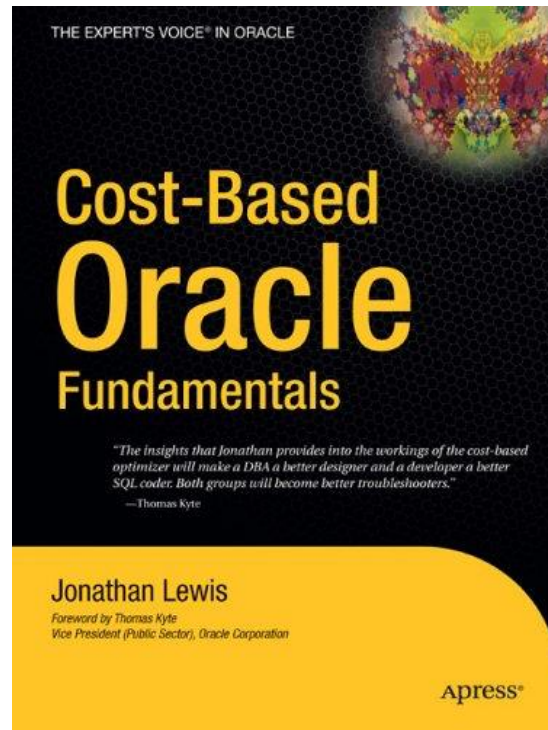
Condition	Estimate	Real	Error Factor
CD & LUD	806,000	4,924,104	6
CD	403,000	4,784,580	12
LUD	426,000	4,924,104	12



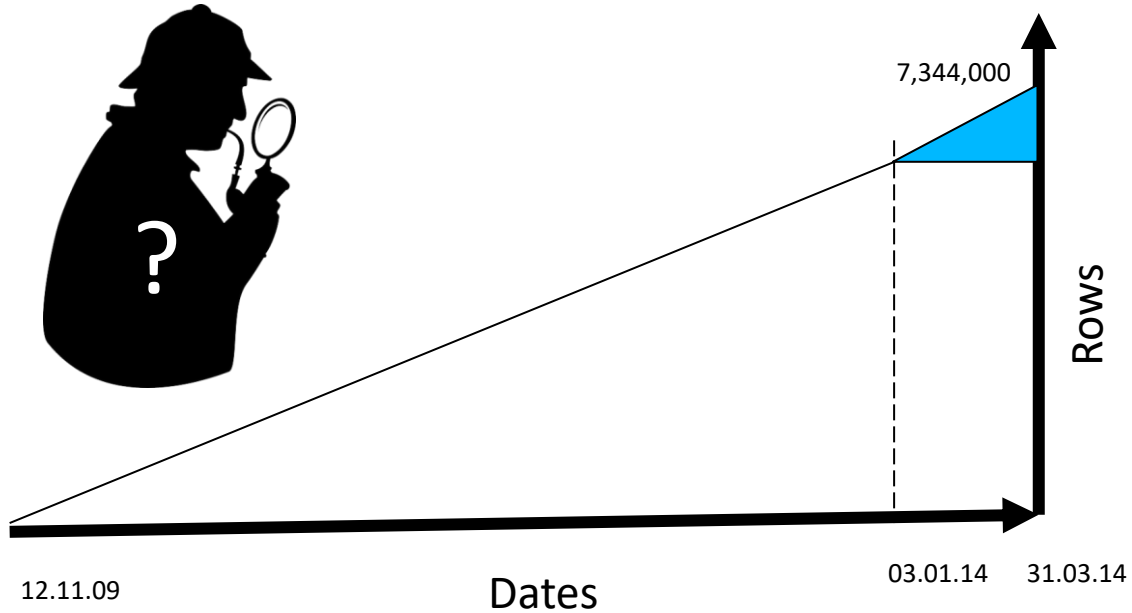
# How is the estimate calculated?



$$(x + a)^n = \sum_{k=0}^n \binom{n}{k} x^k a^{n-k}$$

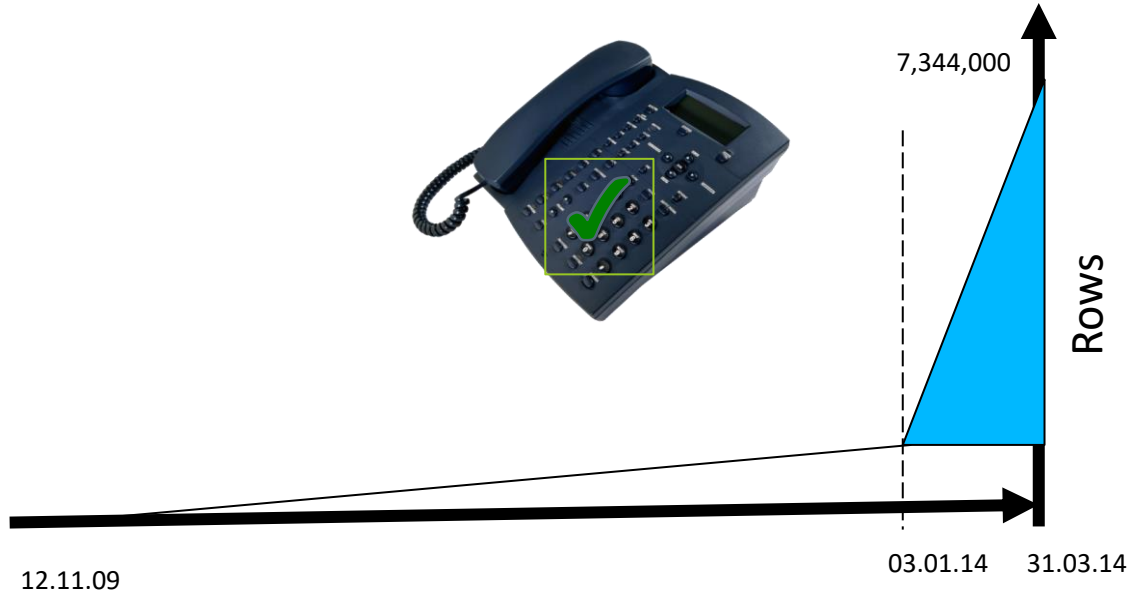


# The estimation Formular explained



# Was wenn die Annahme falsch ist?

New hypothesis:  
recent increase of  
data

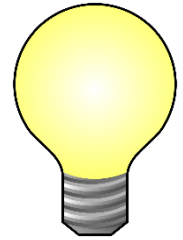


# Try the generic Solution first



Christian Antognini,  
Senior Principal Consultant,  
Oak Table Member

„Before you apply tuning techniques make sure the initialization parameters are correctly set and all the necessary system and object statistics are in place.”



# Better Statistics are generally beneficial

```
BEGIN
```

```
  dbms_stats.gather_schema_stats (user,  
  estimate_percent=> 100,  
  method_opt=> 'for all columns size  
skewonly');
```

```
END;
```

```
/
```

# The isolated Test shows improved estimates

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers
0	SELECT STATEMENT		1		1	00:00:01.68	21984
1	SORT AGGREGATE		1	1	1	00:00:01.68	21984
* 2	INDEX FAST FULL SCAN	MEDI_A_I01	1	6470K	4924K	00:00:01.26	21984

Predicate Information (identified by operation id):

```
2 - filter(("LAST_UPDATE_DATE">:LADEDATUMZEIT OR "CREATION_DATE">:LADEDATUMZEIT))
```



# Estimation Error, Part 2

```
16 - filter (CreationDate>:loaddate  
OR lastChangeDate>:loaddate)  
AND TO_NUMBER(effective(id))=1
```

# Results of the Systematical Tests

Condition	Estimate	Real	Error Factor
TO_NUMBER (effective(id))=1	73,440	7,344,000	100

# The Function's Code

```

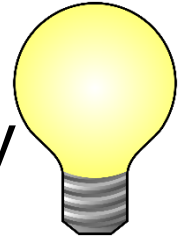
FUNCTION effective (p_id IN VARCHAR2)
RETURN VARCHAR2 IS
BEGIN
-- IF      p_id = '0'
--      OR p_id = current_id
--      OR current_id = '0'
-- THEN
RETURN ('1');
-- ELSE
--      RETURN ('0');
-- END IF;
END;
  
```

# What causes the Estimation Error?



Joze Senegacnik,  
Oak Table Member

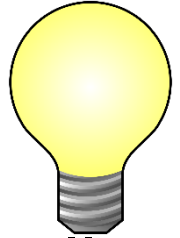
„The default selectivity  
for a unknown  
functions is 1%.“



# Correct Estimation

```
ASSOCIATE STATISTICS WITH  
FUNCTIONS effective DEFAULT  
SELECTIVITY 100;
```

# Dan Tow's Rule (Sql Diagramming)



Dr. Daniel Tow, former head of Oracle Applications tuning was also lecturing sql tuning at Stanford University, USA.

„The best execution plan usually starts with the best filter condition and minimizes reads to the largest tables.“

# Dan Tow's Rule intuitively explained:



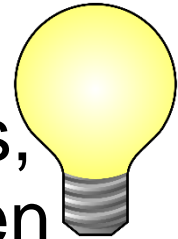
How Water is flowing down the mountains by going through the deepest and steepest path, so should an execution plan go for least number of rows to retrieve.

# Speed Version of SQL Diagramming



Lothar Flatz,  
Oak Table Member,  
Oracle Performance Architect

SQL Diagramming is labourious,  
but here is a shortcut that often  
work:



„The source of biggest throw  
away will often indicate the point  
of best selectivity”



# Throw away, that rings a bell ...

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	Reads
0	SELECT STATEMENT		1		12342	00:00:17.85	306K	237K
1	SORT ORDER BY		1	771	12342	00:00:17.85	306K	237K
2	SORT UNIQUE		1	771	12342	00:00:17.72	306K	237K
3	UNION-ALL		1		19469	00:00:17.68	306K	237K
* 4	HASH JOIN		1	297	1038	00:00:00.91	25441	78
* 5	HASH JOIN		1	298	1038	00:00:00.69	24558	78
* 6	TABLE ACCESS BY INDEX ROWID	MEDI_B	1	298	40557	00:00:00.59	24033	78
* 7	INDEX RANGE SCAN	I2	1	5356	40557	00:00:00.16	105	78
* 8	TABLE ACCESS BY INDEX ROWID	MEDI_C	1	2309	721	00:00:00.01	525	0
* 9	INDEX RANGE SCAN	I1	1	2317	721	00:00:00.01	5	0
10	INDEX FAST FULL SCAN	I3	1	279	279K	00:00:00.07	883	0
* 11	HASH JOIN		1	474	18431	00:00:16.73	280K	237K
* 12	HASH JOIN		1	475	18431	00:00:16.58	279K	237K
* 13	TABLE ACCESS BY INDEX ROWID	MEDI_C	1	2309	721	00:00:00.01	525	0
* 14	INDEX RANGE SCAN	I1	1	2317	721	00:00:00.01	5	0
* 15	HASH JOIN		1	7202	957K	00:00:16.23	279K	237K
* 16	TABLE ACCESS FULL	MEDI_A	1	7161	957K	00:00:12.78	237K	237K
* 17	TABLE ACCESS FULL	MEDI_B	1	598	598K	00:00:01.99	41476	0
18	INDEX FAST FULL SCAN	I3	1	279	279K	00:00:00.04	883	0

# Final result with Hints

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers
0	SELECT STATEMENT		1		2618	00:00:03.68	51765
1	SORT ORDER BY		1	2814	2618	00:00:03.68	51765
2	SORT UNIQUE		1	2814	2618	00:00:03.66	51765
3	UNION-ALL		1		70207	00:00:03.45	51765
* 4	HASH JOIN		1	297	0	00:00:00.01	544
* 5	HASH JOIN		1	297	0	00:00:00.01	544
* 6	TABLE ACCESS BY INDEX ROWID	MEDI_B	1	297	73	00:00:00.01	26
* 7	INDEX RANGE SCAN	I2_X	1	5352	73	00:00:00.01	3
* 8	TABLE ACCESS BY INDEX ROWID	MEDI_C	1	2309	712	00:00:00.01	518
* 9	INDEX RANGE SCAN	I1	1	2318	712	00:00:00.01	5
10	INDEX FAST FULL SCAN	I3	0	278K	0	00:00:00.01	0
11	CONCATENATION		1		70207	00:00:02.95	51221
12	NESTED LOOPS		1	1228	0	00:00:00.39	23200
13	NESTED LOOPS		1	1230	0	00:00:00.39	23200
14	NESTED LOOPS		1	2500	7915	00:00:00.19	7481
* 15	TABLE ACCESS BY INDEX ROWID	MEDI_C	1	2309	712	00:00:00.01	518
* 16	INDEX RANGE SCAN	I1	1	2318	712	00:00:00.01	5
17	TABLE ACCESS BY INDEX ROWID	MEDI_B	712	1	7915	00:00:00.15	6963
* 18	INDEX RANGE SCAN	I2_X	712	1	7915	00:00:00.08	1570
19	TABLE ACCESS BY INDEX ROWID	MEDI_A	7915	1	0	00:00:00.17	15719
* 20	INDEX RANGE SCAN	I4_X	7915	1	0	00:00:00.07	15719
* 21	INDEX UNIQUE SCAN	I3	0	1	0	00:00:00.01	0
22	NESTED LOOPS		1	1488	70207	00:00:02.38	28021
23	NESTED LOOPS		1	1491	70207	00:00:01.55	27339
24	NESTED LOOPS		1	2500	7915	00:00:00.18	7626
* 25	TABLE ACCESS BY INDEX ROWID	MEDI_C	1	2309	712	00:00:00.01	518
* 26	INDEX RANGE SCAN	I1	1	2318	712	00:00:00.01	5
27	TABLE ACCESS BY INDEX ROWID	MEDI_B	712	1	7915	00:00:00.14	7108
* 28	INDEX RANGE SCAN	I2_X	712	1	7915	00:00:00.07	1715
* 29	TABLE ACCESS BY INDEX ROWID	MEDI_A	7915	1	70207	00:00:01.14	19713
* 30	INDEX RANGE SCAN	I4_X	7915	1	70207	00:00:00.60	15976
* 31	INDEX UNIQUE SCAN	I3	70207	1	70207	00:00:00.36	682

# Final result with Hints

No longer overload of I/O sub system  
=> logical reads reduced by a factor > 50  
=> physical reads almost at zero

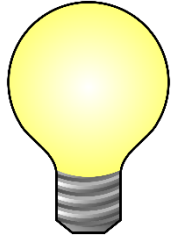
**Mission completed (?)**

# How to estimate the efficiency of SQL?



Christian Antognini,  
Senior Principal Consultant,  
Oak Table Member

„Access paths that lead to up to 10-15 logical reads per returned row are probably acceptable . Access paths that lead to more than 15-20 logical reads per returned row are probably inefficient.“



Quote: „Troubleshooting Oracle Performance“ Page 341

# Are we efficient now?

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers
0	SELECT STATEMENT		1		2618	00:00:03.68	51765
1	SORT ORDER BY		1	2814	2618	00:00:03.68	51765
2	SORT UNIQUE		1	2814	2618	00:00:03.66	51765
3	UNION-ALL		1		70207	00:00:03.45	51765
* 4	HASH JOIN		1	297	0	00:00:00.01	544
* 5	HASH JOIN		1	297	0	00:00:00.01	544
* 6	TABLE ACCESS BY INDEX ROWID	MEDI_B	1	297	73	00:00:00.01	26
* 7	INDEX RANGE SCAN	I2_X	1	5352	73	00:00:00.01	3
* 8	TABLE ACCESS BY INDEX ROWID	MEDI_C	1	2309	712	00:00:00.01	518
* 9	INDEX RANGE SCAN	I1	1	2318	712	00:00:00.01	5
10	INDEX FAST FULL SCAN	I3	0	278K	0	00:00:00.01	0
11	CONCATENATION		1		70207	00:00:02.95	51221
12	NESTED LOOPS		1	1228	0	00:00:00.39	23200
13	NESTED LOOPS		1	1230	0	00:00:00.39	23200
14	NESTED LOOPS		1	2500	7915	00:00:00.19	7481
* 15	TABLE ACCESS BY INDEX ROWID	MEDI_C	1	2309	712	00:00:00.01	518
* 16	INDEX RANGE SCAN	I1	1	2318	712	00:00:00.01	5
* 17	TABLE ACCESS BY INDEX ROWID	MEDI_B	712	1	7915	00:00:00.15	6963
* 18	INDEX RANGE SCAN	I2_X	712	1	7915	00:00:00.08	1570
* 19	TABLE ACCESS BY INDEX ROWID	MEDI_A	7915	1	0	00:00:00.17	15719
* 20	INDEX RANGE SCAN	I4_X	7915	1	0	00:00:00.07	15719
* 21	INDEX UNIQUE SCAN	I3	0	1	0	00:00:00.01	0
22	NESTED LOOPS		1	1488	70207	00:00:02.38	28021
23	NESTED LOOPS		1	1491	70207	00:00:01.55	27339
24	NESTED LOOPS		1	2500	7915	00:00:00.18	7626
* 25	TABLE ACCESS BY INDEX ROWID	MEDI_C	1	2309	712	00:00:00.01	518
* 26	INDEX RANGE SCAN	I1	1	2318	712	00:00:00.01	5
* 27	TABLE ACCESS BY INDEX ROWID	MEDI_B	712	1	7915	00:00:00.14	7108
* 28	INDEX RANGE SCAN	I2_X	712	1	7915	00:00:00.07	1715
* 29	TABLE ACCESS BY INDEX ROWID	MEDI_A	7915	1	70207	00:00:01.14	19713
* 30	INDEX RANGE SCAN	I4_X	7915	1	70207	00:00:00.60	15976
* 31	INDEX UNIQUE SCAN	I3	70207	1	70207	00:00:00.36	682

# Very likely a flawed business model



Thank you for listening !



**Lothar Flatz**

[Lothar.flatz@dbconcepts.at](mailto:Lothar.flatz@dbconcepts.at)

[www.dbconcepts.at](http://www.dbconcepts.at)