

OPTIMIZACIJA SQL UPITA NA NAČIN KAKO TO RADI ORACLE SUPPORT

Josip Pojatina
mStart d.o.o. (Agrokor ICT)
josip.pojatina@agrokor.hr

Sadržaj

- O tvrtki mStart
- Uvod
- Tkprof
- Trace analyzer
- SQLTXPLAIN
- SQL Health Check
- Demo
- Pitanja i odgovori

O tvrtki mStart

- Agrokor ICT 1.7.2010. promijenio naziv u mStart d.o.o.
- Djeluje kao samostalni subjekt unutar Agrokor koncerna s ciljem pružanja podrške za 100+ kompanija unutar Agrokor grupacije

Ulje i margarina						
Sladoledi i zamrznuta hrana						
Vode i pića						
Meso i mesne prerađevine						
Poljoprivreda						
Maloprodaja, veleprodaja i distribucija						
Ostale djelatnosti						

Uvod

- SQL je fundamentalna komponenta svih poslovnih sustava
- Loš SQL plan onemogućuje uspješno izvršavanje poslovnih procesa
- Problemi sa izvršavanjem SQL-ova su prisutni i nakon stabilizacije sustava
- Mnogobrojni su razlozi zbog kojih dolazi do problema s izvršavanjem SQL upita (statistika, temp i objektne tablice, view-ovi, ETL/batch obrade, neravnomjerna razdioba vrijednosti (skew distribution) unutar stupca, međusobno ovisni stupci u where uvjetu, ograničenja u implementaciji CBO-a...)

Uvod

- Iako naizgled jednostavan za korištenje, SQL engine je najkompliciraniji dio Oracle baze (CBO)
- SQL engine sa svakom verzijom postaje još moćniji i kompleksniji
- SQL reference manual u 12c je gotovo dvostruko veći od 8i
- Unatoč jednostavnosti, stvarno iskorištavanje SQL engine-a zahtjeva ekspertno poznavanje svih komponenti sustava (DBA/OS/Storage/development (PL/SQL, Java, C...))

Uvod

- Ukoliko SQL upit/DML postane do te mjere neefikasan da ozbiljno ugrožava sposobnost sustava da izvršava poslovni proces u zadanom vremenskom roku, treba odmah otvoriti SR
- Oracle Support tehničar je u bitno nepovoljnijem položaju od DBA
 - ne zna ništa o tvrtki koja je otvorila SR (Oracle ima više stotina tisuća klijenata)
 - ne zna ništa o poslovnom procesu koji se neefikasno izvršava
 - ne zna ništa o korisničkim podacima
 - nema pristup poslovnom sustavu korisnika

Uvod

- Kako bi smanjio broj nepotrebnih iteracija na relaciji korisnik – Support inženjer, Oracle je razvio niz alata koji smanjuju korespondenciju na najmanju moguću mjeru a samim time i ubrzavaju rješavanje SR u najkraćem mogućem roku
- Svi alati su besplatni za korisnike pretplaćene na Oracle Support
- Najveći dio metoda koje koristi Oracle Support može koristiti i krajnji korisnik (DBA) prilikom rješavanja problema sa SQL upitima
- Cilj ove prezentacije je predstavljanje dijela metoda koje koristi Oracle Support a koje svaki DBA može koristiti

Uvod

- U većini slučajeva u nekoliko iteracija (2 -3) Oracle Support inženjer može ponuditi rješenje
- Potrebni podaci za Oracle Support inženjera su:
 - trace file
 - sqltxplan report
- Trace file se analizira korištenjem tkprof-a i/ili trace analyzer-a

Tkprof

- tkprof alat ima slijedeće karakteristike
 - brzina (4.2 Gb trace file obradi za 15-ak sec)
 - samostalan alat (ne zahtjeva spajanje na bazu)
 - tkprof dolazi u sklopu instalacije Oracle baze i Oracle Client-a
 - rezultat analize trace datoteke je izvještaj u tekstualnom formatu

Tkprof

TKPROF: Release 11.2.0.2.0 – Development on Thu Sep 19 22:12:47 2013

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

Trace file: orcl_ora_2774_HROUG_2013.trc

Sort options: exeela prsela fchela

```
*****
count   = number of times OCI procedure was executed
cpu     = cpu time in seconds executing
elapsed = elapsed time in seconds executing
disk    = number of physical reads of buffers from disk
query   = number of buffers gotten for consistent read
current = number of buffers gotten in current mode (usually for update)
rows    = number of rows processed by the fetch or execute call
*****
```

SQL ID: gk4twjgf4kz0h Plan Hash: 568005898

```
select *
from
emp, dept where dept.deptno = :b1
```

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.00	0	0	0	0
Execute	1	0.00	0.05	0	0	0	0
Fetch	1	0.00	0.00	0	4	0	14
total	3	0.00	0.05	0	4	0	14

Misses in library cache during parse: 1

Optimizer mode: ALL_ROWS

Parsing user id: 83

Number of plan statistics captured: 1

Rows (1st) Rows (avg) Rows (max) Row Source Operation

14	14	14	NESTED LOOPS (cr=4 pr=0 pw=0 time=114 us cost=4 size=812 card=14)
1	1	1	TABLE ACCESS BY INDEX ROWID DEPT (cr=2 pr=0 pw=0 time=22 us cost=1 size=20 card=1)
1	1	1	INDEX UNIQUE SCAN PK_DEPT (cr=1 pr=0 pw=0 time=16 us cost=0 size=0 card=1)(object id 75334)
14	14	14	TABLE ACCESS FULL EMP (cr=2 pr=0 pw=0 time=50 us cost=3 size=532 card=14)

Elapsed times include waiting on following events:

Event waited on	Times	Max. Wait	Total Wait
		Waited	Waited
SQL*Net message to client	1	0.00	0.00
SQL*Net message from client	1	0.00	0.00

Trace analyzer

- Trace analyzer utility ima slijedeće karakteristike
 - potrebno ga je skinuti s My Oracle Support portala
 - sporiji je od tkprof-a (najvećim dijelom zato jer prikazuje sveobuhvatnu analizu trace datoteke)
 - nije pogodan za analizu velikih trace datoteka
 - preduvjet za dobivanje izvještaja je spajanje na Oracle bazu (bilo koju bazu, ne mora nužno biti baza gdje je nastao trace file)
 - rezultat analize trace datoteke je izvještaj u html formatu (lakše za čitanje)

Trace analyzer - nastavak

- prikazuje stvarne vrijednosti bind varijabli
- prikazuje “hottest” blokove, statistiku za indekse i tablice
- razlikuje korisničke rekurzivne pozive od internih rekurzivnih poziva
- prikazuje puno detaljnije wait event informacije

Trace analyzer - nastavak

224270.1 TRCA Trace Analyzer 11.4.5.8 Report: trca_e49102.html

orcl_ora_2774_HROUG_2013.trc (3474 bytes)

Total Trace Response Time: 31.562 secs.
 2013-SEP-19 21:58:40.103 (start of first db call in trace 1379620720.06668).
 2013-SEP-19 21:59:11.665 (end of last db call in trace 1379620751.628303).

- [Glossary of Terms Used](#)
- [Response Time Summary](#)
- [Overall Time and Totals](#)
- [Non-Recursive Time and Totals](#)
- [Recursive Time and Totals](#)
- [Top SQL](#)
- [Non-Recursive SQL](#)
- [SQL Genealogy](#)
- [Individual SQL](#)
- [Overall Segment I/O Wait Summary](#)
- [Hot I/O Blocks](#)
- [Gaps in Trace](#)
- [ORA errors in Trace](#)
- [Transactions Summary](#)
- [Non-default Initialization Params](#)
- [Trace Header](#)
- [Tool Data Dictionary](#)
- [Tool Execution Environment](#)
- [Tool Configuration Parameters](#)

Glossary of Terms Used

[+]

Response Time Summary

Response Time Component	Time (in secs)	pct of total resp time	Time (in secs)	pct of total resp time	Time (in secs)	pct of total resp time
CPU Time:	0.005	0.0%				
Non-idle Wait Time:	0.000	0.0%				
ET Unaccounted-for Time:	55.786	176.8%				
Total Elapsed Time¹:			55.791	176.8%		
Idle Wait Time:			55.732	176.6%		
RT Unaccounted-for Time:			-79.962	-253.4%		
Total Response Time²:					31.562	100.0%

(1) Total Elapsed Time = "CPU Time" + "Non-Idle Wait Time" + "ET Unaccounted-for Time".
 (2) Total Response Time = "Total Elapsed Time" + "Idle Wait Time" + "RT Unaccounted-for Time".
 Total Accounted-for Time = "CPU Time" + "Non-Idle Wait Time" + "Idle Wait Time" = 111.524 secs.

SQLTXPLAIN

- najvažniji alat za analizu SQL upita
- besplatan za korisnike koji su pretplaćeni na Oracle Support
- Carlos Sierra je autor ovog izuzetnog alata
- alat se već više od 15 godina kontinuirano unapređuje
- SQLT je glavni alat kojeg koriste Oracle Support inženjeri za analizu problema prilikom izvršavanja SQL upita
- Pored analize općih SQL upita, SQLT je posebno prilagođen za optimizaciju Oracle eBS i Siebel CRM aplikacija

SQLTXPLAIN - nastavak

- Glavne metode/analize su:
 - sqltextract
 - sqltexecute
 - sqlttrxec
 - sqltxplore
 - sqltcompare
 - sqltxtrsby

SQLTXPLAIN - nastavak

- SQLTXTRACT
 - proizvodi izvještaj bez da izvršava problematičan SQL
 - izvlači sve raspoložive podatke o problematičnom SQL-u (AWR, library cache, ADDM, STA, ASH, SQL Monitor...)

- SQLTXECUTE
 - izvršava (jednom) problematičan SQL
 - proizvodi detaljnije izvještaje (estimate/actual usporedba)
 - ne može se primijeniti u svim situacijama

SQLTXPLAIN - nastavak

- SQLTXTRXEC
 - kombinacija sqltextract i sqltexecute
- SQLTXPLORE
 - po principu “brute force” napada problematičan SQL isprobavajući sve kombinacije parametara
 - dizajniran je za pronalaženje bugova prouzročenih upgrade-om ili patchiranjem
 - koristi se kad i nakon analiza xtract/xecute i nadalje ne znamo uzrok promjene plana izvršavanja SQL upita

SQLTXPLAIN - nastavak

- SQLTCOMPARE
 - za usporedbu execution planova između testnog i produkcijskog okruženja
- SQLTXTRSBY
 - dizajniran za s read-only bazama kad se ne mogu koristiti SQLTXTRACT i SQLTXECUTE (zahtjevaju read-write)
 - koristi se za optimiziranje SQL-a koji se izvršavaju na Data Guard-u (najčešće za generiranje izvještaja)

SQLTXPLAIN - nastavak

- Dva “killer feature-a” SQLTXPLAIN-a su:
 - izrada SQL Profile-a na temelju najboljeg plana u memoriji/AWR-u ili testnom okruženju kao privremeni quick fix
 - prebacivanje produkcijskog okruženja na testno okruženje
 - u slučajevima kad xtract i xecute ne mogu dati rješenje problema
 - testno okruženje omogućuje vjerodostojnu replikaciju ponašanja produkcijskog, u kojem se mogu mijenjati parametri, dodavati hintovi, dodavati indexi, statistika...

SQLTXPLAIN - nastavak

215187.1 SQLT XTRACT 11.4.5.9 Report: sqlt_s84390_main.html

Global

- [Observations](#)
- [SQL Text](#)
- [SQL Identification](#)
- [Environment](#)
- [CBO Environment](#)
- [Fix Control](#)
- [CBO System Statistics](#)
- [DBMS_STATS Setup](#)
- [Initialization Parameters](#)
- [NLS Parameters](#)
- [I/O Calibration](#)
- [Tool Configuration Parameters](#)

Plans

- [Summary](#)
- [Performance Statistics](#)
- [Performance History \(delta\)](#)
- [Performance History \(total\)](#)
- [Execution Plans](#)

Plan Control

- [Stored Outlines](#)
- [SQL Patches](#)
- [SQL Profiles](#)
- [SQL Plan Baselines](#)

Tables

- [Tables](#)
- [Statistics](#)
- [Statistics Versions](#)
- [Modifications](#)
- [Properties](#)
- [Physical Properties](#)
- [Constraints](#)
- [Columns](#)
- [Indexed Columns](#)
- [Histograms](#)
- [Partitions](#)
- [Indexes](#)

Cursor Sharing and Binds

- [Cursor Sharing](#)
- [Adaptive Cursor Sharing](#)
- [Peeked Binds](#)
- [Captured Binds](#)

SQL Tuning Advisor

- [STA Report](#)
- [STA Script](#)

SQL Execution

- [Active Session History](#)
- [AWR Active Session History](#)
- [SQL Statistics](#)
- [SQL Detail ACTIVE Report](#)
- [Monitor Statistics](#)
- [Monitor ACTIVE Report](#)
- [Monitor HTML Report](#)
- [Monitor TEXT Report](#)
- [Segment Statistics](#)
- [Session Statistics](#)
- [Session Events](#)
- [Parallel Processing](#)

Objects

- [Objects](#)
- [Dependencies](#)
- [Fixed Objects](#)
- [Fixed Object Columns](#)
- [Nested Tables](#)
- [Policies](#)
- [Audit Policies](#)
- [Tablespaces](#)
- [Metadata](#)

This report may include some content provided by the Oracle Diagnostic and/or the Oracle Tuning Packs (in particular SQL Tuning Advisor "STA", SQL Tuning Sets "STS", SQL Monitoring and/or Automatic Workload Repository "AWR"). Be aware that using this extended functionality requires a license for the corresponding pack. If you need to disable SQLT access to one of these packages, please execute one of the following commands: SQL> EXEC sqltadmin.sqlt\$a.disable_tuning_pack_access; or SQL> EXEC sqltadmin.sqlt\$a.disable_diagnostic_pack_access;

sqlt_start: 2013-07-12/12:30:57

Observations

SQLTXPLAIN - nastavak

Execution Plans

List ordered by phv and source.

#	Plan Hash Value	SQL Plan Hash Value ¹	SQL Plan Hash Value ²	Src	Source	Plan Info	Plan Stability	Is Bind Sensitive	Optimizer	Optimizer Cost	Estimated Cardinality E-Rows	Rows Processed A-Rows	Plan Timestamp	Child Plans ²	Plan ID	Task ID	Attribute
1	449374452	39888	39888	AWR	DBA_HIST_SQL_PLAN				CHOOSE	11		39	2013-07-05/05:55:02				
2	621178183	23499	13466	XPL	PLAN_TABLE				CHOOSE	28630	16631		2013-07-12/12:31:02		5069		
3	1946316727 [W]	1075	95290	MEM	GV\$SQL_PLAN			N	CHOOSE	13		1757	2013-07-11/22:02:46	1			
4	1946316727 [W]	1075	1075	AWR	DBA_HIST_SQL_PLAN				CHOOSE	14		33	2013-07-10/05:55:02				
5	4076144913 [B]	35535	35535	AWR	DBA_HIST_SQL_PLAN				CHOOSE	25		10	2013-06-29/07:05:04				

(1) SQLT PHV considers id, parent_id, operation, options, index_columns and object_name. SQLT PHV2 includes also access and filter predicates.

(2) Display of child plans is restricted up to 100 per phv as per tool parameter "r_rows_table_s".

[Go to Plan Performance Statistics](#)

[Go to Plans Summary](#)

[Go to Top](#)

Execution Plan phv:449374452 sqit_phv:39888 sqit_phv2:39888 source:DBA_HIST_SQL_PLAN timestamp:2013-07-05/05:55:02 oldest_snapshot:2013-07-05/08:30:21

SQL Text: [-]

```

MERGE INTO MATS_TEVENT_DATA TDATA USING ( SELECT TEVENT.MESSAGE_ID, DATAFLOW.TARGET_SYSTEM, DATAFLOW.ENTITY, DATAFLOW.SOURCE_SYSTEM FROM ( SELECT /*+ PARALLEL(tt,4)
*/ TT.MESSAGE_ID,
TT.DATAFLOW_ID FROM MATS_TEVENT TT WHERE TT.PUBLISH_DATETIME > SYSDATE - :B1 AND EXISTS( SELECT NULL FROM MATS_TEVENT_DATA TD1 WHERE TD1.MESSAGE_ID = TT.MESSAGE_ID AND TD1.ROW_STATUS = 'N' AND
TD1.TARGET_SYSTEM IS NOT NULL AND TD1.ENTITY IS NULL AND TD1.SOURCE_SYSTEM IS NULL )
TEVENT, MATS_DATAFLOW DATAFLOW WHERE DATAFLOW.ID = TEVENT.DATAFLOW_ID GROUP BY TEVENT.MESSAGE_ID, DATAFLOW.TARGET_SYSTEM, DATAFLOW.ENTITY, DATAFLOW.SOURCE_SYSTEM )
SEL ON ( TDATA.MESSAGE_ID = SEL.MESSAGE_ID AND TDATA.TARGET_SYSTEM = SEL.TARGET_SYSTEM AND TDATA.ROW_STATUS = 'N' AND TDATA.TARGET_SYSTEM IS NOT NULL )
WHEN MATCHED THEN UPDATE SET TDATA.ENTITY = SEL.ENTITY, TDATA.SOURCE_SYSTEM = SEL.SOURCE_SYSTEM
    
```

SQL: [+]

ID	Exec Ord	Operation	Go To	More	Cost ²	Estim Card	PStart	PStop
0	20	MERGE STATEMENT			11			
1	19	MERGE MATS_TEVENT_DATA			13			
2	18	. VIEW			13			
3	17	.. NESTED LOOPS			13			
4	15	... NESTED LOOPS			11	1		
5	13 VIEW			9	1		
6	12+ SORT GROUP BY			9	1		
7	11+. NESTED LOOPS			9			
8	9+. NESTED LOOPS			8	1		
9	7+. NESTED LOOPS			7	1		
10	3+. SORT UNIQUE			2	1		
11	2+. TABLE ACCESS BY INDEX ROWID MATS_TEVENT_DATA	[+]		2	1		
12	1+. INDEX FULL SCAN MATS_TEVENT_DATA_IX_4	[+]		2	1		
13	6+. PARTITION RANGE ITERATOR		[+]	4	1	KFY	27

SQLTXPLAIN - nastavak

SQL: [\[+\]](#)

ID	Exec Ord	Operation	Go To	More	Cost ²	Estim Card	PStart	PStop
0	20	MERGE STATEMENT			11			
1	19	MERGE MATS_TEVENT_DATA			13			
2	18	. VIEW			13			
3	17	.. NESTED LOOPS			13			
4	15	... NESTED LOOPS			11	1		
5	13 VIEW			9	1		
6	12+ SORT GROUP BY			9	1		
7	11+. NESTED LOOPS			9			
8	9+.. NESTED LOOPS			8	1		
9	7+... NESTED LOOPS			7	1		
10	3+.... SORT UNIQUE			2	1		
11	2+....+ TABLE ACCESS BY INDEX ROWID <u>MATS_TEVENT_DATA</u>	[+]		2	1		
12	1+....+. INDEX FULL SCAN <u>MATS_TEVENT_DATA_IX_4</u>	[+]		2	1		
13	6+.... PARTITION RANGE ITERATOR		[+]	4	1	KEY	27
14	5+....+ TABLE ACCESS BY LOCAL INDEX ROWID <u>MATS_TEVENT</u>	[-]	[-]	4	1	KEY	27
			Table Columns	Partition ID [Start] [Stop]				
			Col Statistics	13 [KEY] [27]				
			Stats Versions					
			Column Usage					
			Col Properties					
			Histograms					
			Table					
			Constraints					
			Indexed Cols					
			Indexes					
			Partitions					
			Metadata					
15	4+....+. INDEX RANGE SCAN <u>MATS_TEVENT_IND_1</u>	[+]	[+]	2	4	KEY	27
16	8+.... INDEX UNIQUE SCAN <u>MATS_DATAFLOW_PK</u>	[+]	[+]	0	1		
17	10+... TABLE ACCESS BY INDEX ROWID <u>MATS_DATAFLOW</u>	[+]		1	1		
18	14 INDEX RANGE SCAN <u>MATS_TEVENT_DATA_IX_4</u>	[+]	[+]	2	1		
19	16	... TABLE ACCESS BY INDEX ROWID <u>MATS_TEVENT_DATA</u>	[+]		2	1		

(1) If estim_card * starts < output_rows then under-estimate. If estim_card * starts > output_rows then over-estimate. Color highlights when exceeding * 10x, ** 100x and *** 1000x over/under-estimates.

(2) Largest contributors for cumulative-statistics columns are shown in red.

Other XML (id=1): [\[+\]](#)

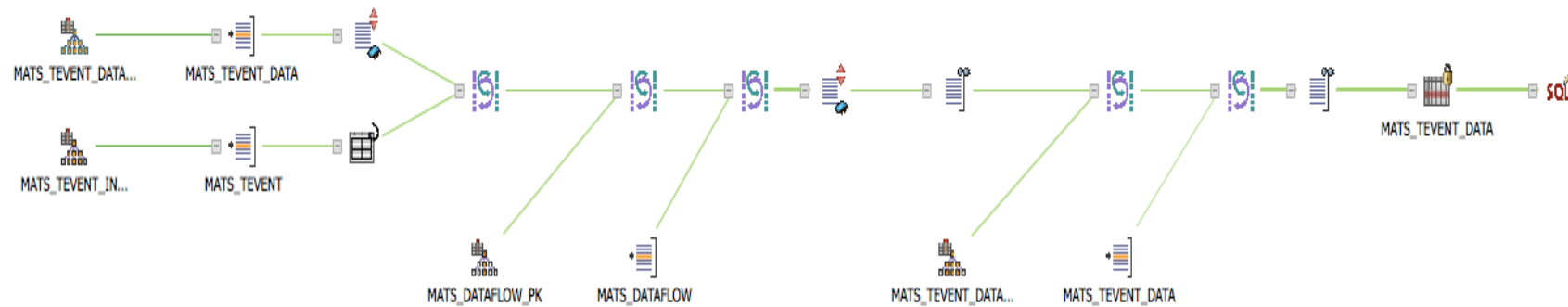
Outline Data (id=1): [\[+\]](#)

...

SQLTXPLAIN - nastavak

Execution Statistics | Historical Statistics | **Plan** | Activity | Top Sessions | Cursor Details | Monitor

Graphical Tabular



SQLTXPLAIN - nastavak

Execution Statistics | Historical Statistics | [Plan](#) | Activity | Top Sessions | Cursor Details | Monitor

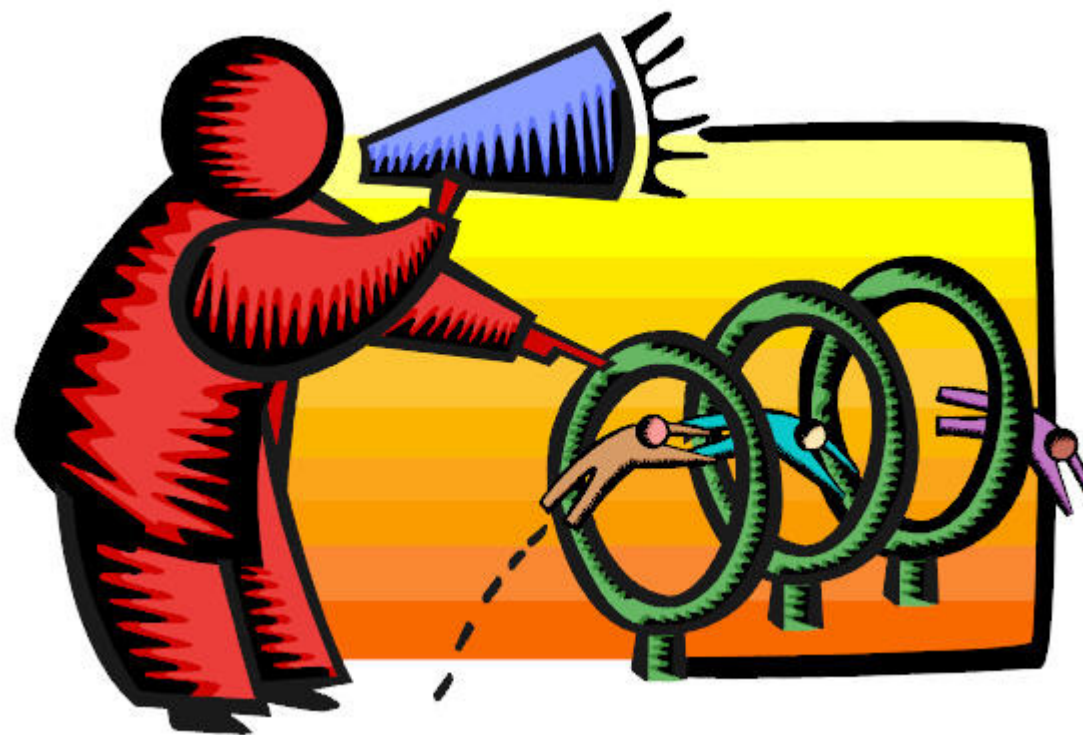
Graphical Tabular

Operation	Object	Predicate	Pruning	Operation Cost	Estimated Rows	Estimated Bytes	Activity %
[-] MERGE STATEMENT							
[-] MERGE	MATS_TEVENT_DATA						.02
[-] VIEW							
[-] NESTED LOOPS							
[-] NESTED LOOPS				1	1	2,250	
[-] VIEW					1	51	
[-] SORT GROUP BY					1	59	
[-] NESTED LOOPS							
[-] NESTED LOOPS				1	1	59	
[-] NESTED LOOPS				1	1	39	
[-] SORT UNIQUE					1	30	
[-] TABLE ACCESS BY INDEX ROWID	MATS_TEVENT_DATA			1	1	30	
INDEX FULL SCAN	MATS_TEVENT_DATA_IX_4			3	2		.06
[-] PARTITION RANGE ITERATOR			KEY .. 29		1	21	
[-] TABLE ACCESS BY LOCAL INDEX ROWID	MATS_TEVENT		KEY .. 29	1	1	21	81
INDEX RANGE SCAN	MATS_TEVENT_IND_1		KEY .. 29	2	4		19
INDEX UNIQUE SCAN	MATS_DATAFLOW_PK				1		
TABLE ACCESS BY INDEX ROWID	MATS_DATAFLOW			1	1	28	
INDEX RANGE SCAN	MATS_TEVENT_DATA_IX_5			2	1		.02
TABLE ACCESS BY INDEX ROWID	MATS_TEVENT_DATA			3	1	2,215	

SQL Health Check

- SQLHC je razvijen iz SQLTXPLAIN-a
- koristi se u situacijama kad SQLTXPLAIN nije moguće instalirati (politika tvrtke, security, mogući bugovi u SQLTXPLAIN-u)
- za razliku od SQLTXPLAIN-a, radi se o skriptama koje ne kreiraju bazne objekte (nije potrebno ništa instalirati na bazu)
- SQLHC pri izvođenju analize problematičnog SQL-a obavlja 100-ak provjera/testova (SQLTXPLAIN 300-ak)

Demo



Optimizacija SQL-a na Oracle Support način

