OPTIMIZACIJA SQL UPITA BEZ IZMJENE PROGRAMSKOG KODA

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

Sadržaj

- O tvrtki mStart
- Uvod
- Odnos krajnjeg korisnika i kupljenog SW-a
- Tipovi procesa i mogući problemi
- Pronalaženje problematičnih SQL-ova
- Načini rješavanja problema sa SQL-ovima
- 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 svih 50+ kompanija unutar Agrokor grupacije

Uvod

- SQL je ključna komponenta svih poslovnih sustava
- Efikasni SQL kôd smanjuje potrošnju resursa, bitno utječe na brzinu sustava i zadovoljstvo krajnjih korisnika, te utječe na smanjenje potrebnog broja licenci
- Neefikasan SQL kôd i promjena plana izvršavanja (execution plan) su dva najčešća problema pri održavanju kupljenog SW rješenja

Odnos krajnjeg korisnika i kupljenog SW-a

- Krajnji korisnik ima zadatak održavati kupljen SW (prva i druga razina podrške)
- Krajnji korisnik najčešće nema pristup izvornom kodu kupljenog SW-a, a ako ima, onda je taj pristup ograničen (najčešće paketi u bazi)
- Čak i kad je izvorni kôd dostupan, korisnici najčešće, zbog nedostatka kompetencije i mogućnosti gubitka podrške, ne mijenjaju kôd kupljenog SW-a, ili rade samo one izmjene koje ne utječu na funkcionalnost SW-a (kreiranje izvještaja)

Tipovi procesa i mogući problemi

- OLTP (Online Transaction Processing)
- ETL procesi punjenja DW-a
- Batch obrade
- Integracijske obrade

Tipovi procesa i mogući problemi

- OLTP - karakteristike
  - Precizni upiti
  - Uglavnom se radi o Nested Loop preko indeksiranih stupaca
  - ER dijagram baziran na 3 normalnoj formi
  - Tablice su povezane PK->FK
  - Nema bulk punjenja i pražnjenja tablica
  - Statistika je uglavnom dobra
  - Rijetko dolazi do problema i promjena execution planova (uglavnom zbog bind variable peeking-a i RLS-a)

Tipovi procesa i mogući problemi

- ETL - karakteristike
  - Proces punjenja DW-a može biti kompleksan
  - Koriste se staging i temp (i objektne) tablice
  - Bulk punjenje i pražnjenje tablica je uobičajeno
  - Upiti su složeniji u odnosu na OLTP
  - Ograničeno vrijeme trajanja ETL-a (Timeframe)
  - Povremeno dolazi do problema s promjenama execution planova
  - Posljedice probijanja predviđenog vremena trajanja najčešće nisu kritične (DW neće imati najsvježije podatke), osim u slučaju kad se na DW-u baziraju ovisni poslovni procesi (primjerice planiranje)

Tipovi procesa i mogući problemi

- Batch procesi - karakteristike
  - Kompleksni procesi (brojne kalkulacije i složeni SQL upiti)
  - Koriste se staging i temp (i objektne) tablice
  - Bulk punjenje i pražnjenje tablica je uobičajeno
  - Upiti su složeniji u odnosu na OLTP
  - Ograničeno vrijeme završetka
  - Često dolazi do problema s promjenama execution planova
  - Posljedice probijanja predviđenog vremena trajanja su kritične jer OLTP ERP sustav neće imati točne podatke

Tipovi procesa i mogući problemi

- Integracijske obrade
  - Kompleksni procesi (složeni SQL upiti)
  - Koriste se staging i temp (i objektne) tablice
  - Bulk punjenje i pražnjenje tablica je uobičajeno
  - Upiti su složeniji u odnosu na OLTP
  - Ograničeno vrijeme završetka
  - Često dolazi do problema s promjenama execution planova
  - Posljedice probijanja predviđenog vremena trajanja su kritične jer razni sustavi neće biti sinhronizirani

Tipovi procesa i mogući problemi

- Razlozi zbog kojih dolazi do promjene execution planova
  - Za temp i objektne tablice ne postoji statistika, pa CBO koristi "*hardkodirane*" vrijednosti
  - Proizvođač SW-a nije skupljanje statistike i hintove uključio u kôd koji izvršava određen proces (ETL, batch...)
  - Proizvođač SW-a ne poznaje korisnikove podatke (distribucija podataka, važne vrijednosti, količina podataka i omjeri...)
  - Testiranja na manjoj količini podataka i nereprezentativnom uzorku
  - Cijeli niz environment varijabli (vsql/vses/v$sys_optimizer_env)
  - Postavke/parametrizacija ("*setup*") sustava

Pronalaženje problematičnih SQL-ova

- Oracle je izvrsno instrumentalizirana baza zbog čega je vrlo lako pronaći problematične upite
- Postoji cijeli niz alata za monitoriranje baze od Oracle-ovih (DB konzola i EM Grid Control) do "*third party*" proizvođača alata (Quest Spotlight, Embarcadero DB Optimizer...)
- Standardni alati ugrađeni u bazu (Statspack, AWR/ASH/ADDM)
- Dinamički view-ovi (v$sql_monitor, v$sql_plan_monitor)
- report_sql_monitor procedura unutar dbms_sqltune paketa

Pronalaženje problematičnih SQL-ova

- prikaz execution plana u realnom vremenu

SQL Plan Monitoring Details (Plan Hash Value=3408134514)

Id	Operation	Name	Estimated Rows	Cost	Active Period (787s)	Execs	Rows	Memory	Temp	IO Requests	CPU Activity	Wait Activity
0	INSERT STATEMENT					1						
1	MULTI-TABLE INSERT					1						
2	VIEW		3M	40982		1						
-> 3	SORT UNIQUE		3M	40982		1	0	26.5MB			.89%	
-> 4	UNION-ALL					1	860K					
-> 5	NESTED LOOPS					1	860K				.38%	
-> 6	NESTED LOOPS		1	229		1	44M				1.1%	
-> 7	NESTED LOOPS		1	227		1	44M					
-> 8	MERGE JOIN CARTESIAN		1	223		1	3M					
-> 9	NESTED LOOPS					1	2337					
-> 10	NESTED LOOPS		1	205		1	2337					
-> 11	TABLE ACCESS FULL	RPM_PROMO_DTL_MERCH_NODE	1	175		1	3					
-> 12	INDEX RANGE SCAN	ITEM_MASTER_I3	1	29		3	2337					
-> 13	TABLE ACCESS BY INDEX ROWID	ITEM_MASTER	1	30		2337	2337					
-> 14	BUFFER SORT		1156	193		2337	3M				.13%	
15	TABLE ACCESS FULL	RPM_PROMO_COMP	1156	18		1	1233					
-> 16	TABLE ACCESS BY INDEX ROWID	RPM_PROMO_DTL	9	4		3M	44M				44%	
-> 17	INDEX RANGE SCAN	RPM_PROMO_DTL_I1	96	1		3M	281M				8.9%	
-> 18	INDEX RANGE SCAN	RPM_PROMO_ZONE_LOCATION_I1	1	1		44M	44M				36%	100%
-> 19	TABLE ACCESS BY INDEX ROWID	RPM_PROMO_ZONE_LOCATION	1	2		44M	860K				8.0%	
20	HASH JOIN		1M	4639								
21	VIEW	index$_join$_010	312K	2589								
22	HASH JOIN											
23	INDEX FAST FULL SCAN	ITEM_MASTER_I3	312K	1645								
24	INDEX FAST FULL SCAN	PK_ITEM_MASTER	312K	1583								
25	MERGE JOIN CARTESIAN		1M	2037								
26	NESTED LOOPS											
27	NESTED LOOPS		11	129								
28	NESTED LOOPS		12	117								
29	TABLE ACCESS FULL	RPM_PROMO_ZONE_LOCATION	12	105								
30	TABLE ACCESS BY INDEX ROWID	RPM_PROMO_DTL	1	1								
31	INDEX UNIQUE SCAN	PK_RPM_PROMO_DTL	1									
32	INDEX UNIQUE SCAN	PK_RPM_PROMO_COMP	1									
33	TABLE ACCESS BY INDEX ROWID	RPM_PROMO_COMP	1	1								
34	BUFFER SORT		114K	2036								
35	TABLE ACCESS FULL	RPM_PROMO_DTL_MERCH_NODE	114K	173								
36	HASH JOIN		2M	5356								
37	TABLE ACCESS FULL	ITEM_MASTER	158K	3305								
38	MERGE JOIN CARTESIAN		1M	2037								

Pronalaženje problematičnih SQL-ova

- AWR report – prikaz SQL-a koji mijenja execution plan

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 61.3% of Total DB Time (s): 56,453
- Captured PL/SQL account for 54.3% of Total DB Time (s): 56,453

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
23,744.62	5,590	4.25	42.06	15.12	4.44	fq3rrdwwr510v	fcexec	declare L_error_message VARCHA...
7,189.03	5,592	1.29	12.73	14.63	2.25	4k7h3trqkznkf	fcexec	INSERT INTO DEAL_ITEM_LOC_EXPL...
5,442.32	2,572	2.12	9.64	19.58	0.00	45s36gc8cymvk	fcexec	MERGE INTO DEAL_ITEM_LOC_EXPLO...
3,121.22	5,593	0.56	5.53	11.49	2.84	6fv85fz1av22a	fcexec	INSERT INTO FUTURE_COST_WORKIN...
2,642.42	1,382	1.91	4.68	29.67	0.00	bjqvs3h8q721h	JDBC Thin Client	UPDATE V_ITEMRANGING SET ROW_S...
1,931.18	9,313	0.21	3.42	15.18	2.70	dtd6hw8ahs76g	fcexec	SELECT FC.ROWID FROM FUTURE_CO...
1,239.29	5,589	0.22	2.20	18.76	0.04	cj53ntg3f24s9	fcexec	MERGE INTO FUTURE_COST_GTT FC ...
1,122.92	5,589	0.20	1.99	10.86	40.04	7sfb24n86ua8d	fcexec	MERGE INTO FUTURE_COST FC USIN...
857.17	140	6.12	1.52	19.99	25.71	7kt0ur6vbzyr4	java@rtl-mom-as3-1p1.konzum.hr (TNS V1-V3)	begin :1 := RPM_PRICE_EVENT_SQ...
767.34	1	767.34	1.36	32.63	36.28	4h0vhg9m0ctk6	SQL*Plus	BEGIN RMS_BASE_PRICE_SQL.FILL_...
705.27	1	705.27	1.25	15.21	63.83	57jr7zavmppv3	Z_ITEM_RANGING_N	declare O_error_message VARCH...
583.70	5,621	0.10	1.03	16.59	0.00	cktwj4hq9thj5	fcexec	MERGE INTO FUTURE_COST_GTT GTT...

Pronalaženje problematičnih SQL-ova

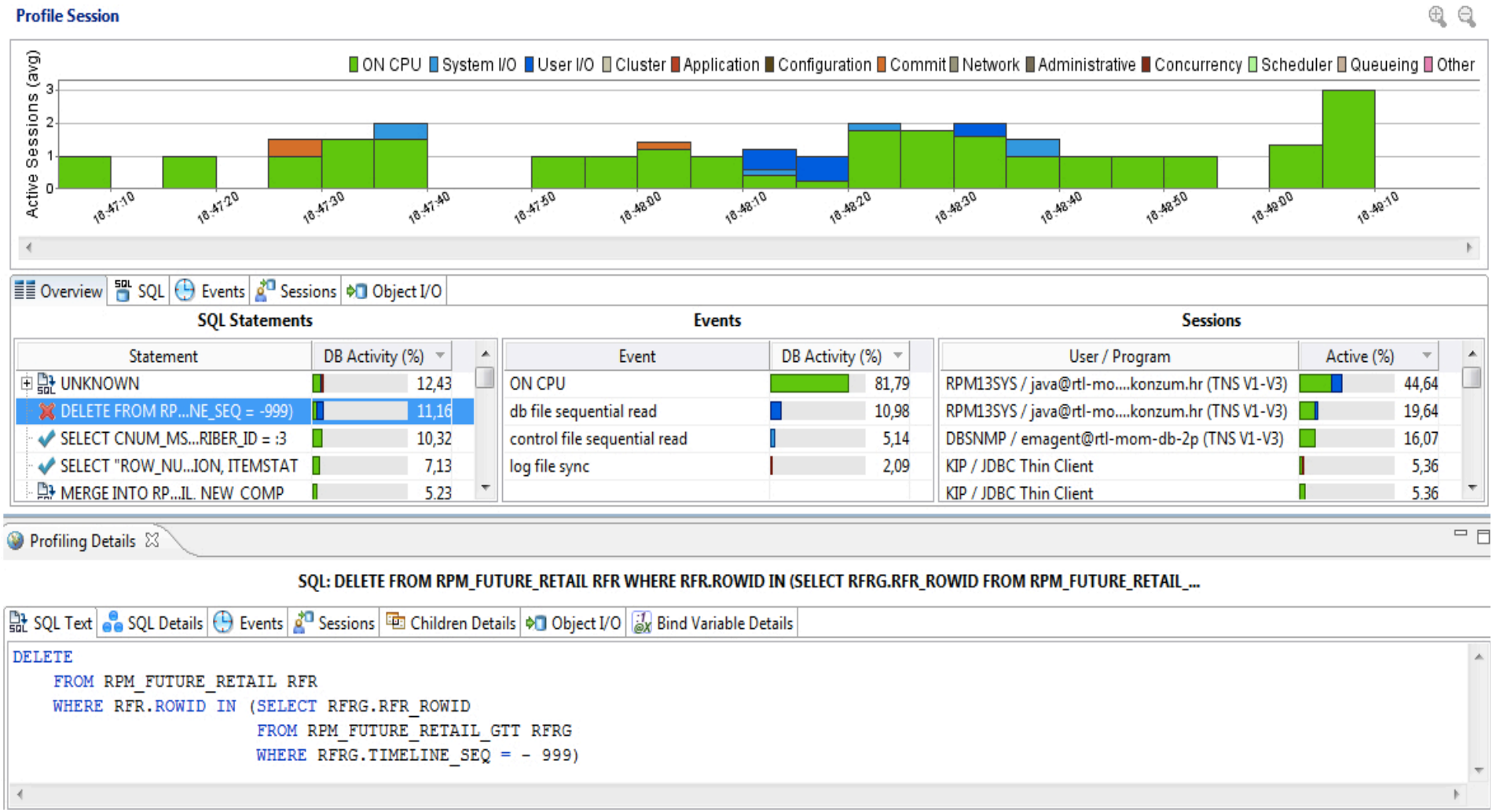
- AWR report – prikaz SQL-a koji mijenja execution plan

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 65.1% of Total DB Time (s): 31,534
- Captured PL/SQL account for 60.4% of Total DB Time (s): 31,534

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
13,558.06	4,570	2.97	43.00	13.93	2.08	fq3rrdwwr510v	fcexec	declare L_error_message VARCHA...
4,671.11	288	16.22	14.81	16.91	0.02	45s36gc8cymvk	fcexec	MERGE INTO DEAL_ITEM_LOC_EXPLO...
2,707.33	4,570	0.59	8.59	9.88	3.35	6fv85fz1av22a	fcexec	INSERT INTO FUTURE_COST_WORKIN...
2,256.58	4,570	0.49	7.16	16.38	0.01	cj53ntg3f24s9	fcexec	MERGE INTO FUTURE_COST_GTT FC ...
1,415.78	8,963	0.16	4.49	14.11	0.02	dtd6hw8ahs76g	fcexec	SELECT FC.ROWID FROM FUTURE_CO...
1,235.70	1	1,235.70	3.92	41.02	8.02	bd4ujrp6cfzj3	java@rtl-mom-as3-1p1.konzum.hr (TNS V1-V3)	begin RPM_ITEM_RECLASS_SQL.REC...
1,207.88	1	1,207.88	3.83	8.44	81.01	57jr7zavmppv3	Z_ITEM_RANGING_N	declare O_error_message VARCH...
976.64	2,776	0.35	3.10	47.54	0.00	9g2xsdyttfn5d	java@rtl-mom-as3-1p1.konzum.hr (TNS V1-V3)	SELECT DISTINCT TRC.ROW_NUMBER...
818.33	44	18.60	2.60	3.87	92.62	b67skgpjv252w	Z_KIRE_ITEM_RANGING.Z_LOC_PROFILE	UPDATE Z_ITEM_RANGING SET LOC_...
624.52	1	624.52	1.98	41.91	0.77	1xx5f61hz2vwu	java@rtl-mom-as3-1p1.konzum.hr (TNS V1-V3)	INSERT INTO RPM_PROMO_ITEM_LOC...
474.03	4,569	0.10	1.50	8.05	32.25	7sfb24n86ua8d	fcexec	MERGE INTO FUTURE_COST FC USIN...
460.49	291	1.58	1.46	56.38	2.09	03bmr33pvu2qt	fcthreadexec	declare L_error_message VARCHA...
454.15	276,959	0.00	1.44	25.21	0.10	gqcj7r3w6ugqp	SQL*Plus	INSERT INTO SKULIST_DETAIL(SKU...
423.69	4,682	0.09	1.34	15.55	0.00	cktwj4hq9thj5	fcexec	MERGE INTO FUTURE_COST_GTT GTT...
422.39	111	3.81	1.34	56.82	1.55	1mnuj5ybbuz3d	fcthreadexec	INSERT INTO COST_EVENT_THREAD ...
411.44	1	411.44	1.30	36.56	26.45	4h0vhg9m0ctk6	SQL*Plus	BEGIN RMS_BASE_PRICE_SQL.FILL_...
408.76	1	408.76	1.30	36.62	26.53	4cqrj52d7bw3y	RMS_BASE_PRICE_SQL.POPULATE_STG	INSERT INTO KIP.BASEPRICE_STG_...
407.21	1	407.21	1.29	53.93	0.42	82gzs7557vw03	java@rtl-mom-as3-1p1.konzum.hr (TNS V1-V3)	DELETE FROM RPM_PROMO_ITEM_LOC...
348.35	6,885,034	0.00	1.10	26.53	0.00	ggra[illegible]wkwm39z	SQL*Plus	SELECT 'x' FROM SKULIST_DETAIL...

Pronalaženje problematičnih SQL-ova



Načini rješavanja problema sa SQL-ovima

- Ručna manipulacija statistikom
- Stored outlines
- SQL Profile
- Kreiranje promjena access struktura (indeks, tip indeksa, tip tablice, IOT, materijalizirani view-ovi...)
- Promjena execution environmenta (cursor_sharing, optimizer_dynamic sampling, optimizer_features_enable, optimizer_index_caching, optimizer_mode, optimizer_index_cost_adj...)

Načini rješavanja problema sa SQL-ovima

- SQL Plan Baseline
- Nedokumentirane metode promjene execution plana
- Promjena SQL-a (SQL koji daje isti rezultat ali je efikasnije napisan) (zahtjeva promjenu koda)
- Promjena SQL-a dodavanjem hintova (*zahtjeva promjenu koda)
- Promjena Schema dizajna (zahtjeva promjenu kôda)

- *osim ako se ne koristi trik s SQL Plan Baseline-om

Načini rješavanja problema sa SQL-ovima

- Ručna manipulacija statistikom pomoću dbms_stats paketa
  - set_column_stats
  - set_table_stats
  - set_index_stats
  - gather_table_stats→method_opt
- Koristi se kad mi znamo više o podacima nego Oracle-ov defaultni mehanizam skupljanja statistike, ili za brzo skupljanje statistike
- Neupotrebljivo kod objektnih i temp tablica (on commit delete rows)

Načini rješavanja problema sa SQL-ovima

- Stored Outlines
  - funkcionalnost dostupna dugi niz godina
  - set hintova koji bi trebao prisiliti CBO da generira jednak plan za određen SQL upit bez obzira na promjene u okruženju (execution environment)
  - u praksi, bez obzira na postojanje Outlines-a, dolazi do promjene plana
  - ova opcija se napušta u korist SQL Plan Baseline-a

Načini rješavanja problema sa SQL-ovima

- SQL Profile
  - najčešće u kombinaciji sa SQL Tuning Advisor-om se koristi
  - pomaže CBO-u u generiranju efiksanog execution plana (precizna statistika, podaci o environment-u, korekcijom procjena CBO-a → opt_estimate hint)
  - potrebna licenca za Diagnostic i licenca za Tuning pack

Načini rješavanja problema sa SQL-ovima

```
GENERAL INFORMATION SECTION
-------------------------------------------------------------------------------
Tuning Task Name   : staName87440
Tuning Task Owner  : RMS
Tuning Task ID     : 41542
Workload Type      : Single SQL Statement
Execution Count    : 1
Current Execution  : EXEC_41582
Execution Type     : TUNE SQL
Scope              : COMPREHENSIVE
Time Limit(seconds): 1800
Completion Status  : COMPLETED
Started at         : 08/07/2012 10:27:11
Completed at       : 08/07/2012 10:27:26

-------------------------------------------------------------------------------
Schema Name: RMS
SQL ID     : 39xp5f0xpgsw2
SQL Text   : select distinct deals.*, 'N'
             from jp_future_cost_gtt il,
             deal_item_loc_explode deals
             where il.item              = deals.item
             and il.supplier            = deals.suppli
             and il.origin_country_id   = deals      country_id
             and il.location            = deals.l       n

-------------------------------------------------------------------------------
FINDINGS SECTION (1 finding)
-------------------------------------------------------------------------------

1- Index Finding (see explain plans section below)
--------------------------------------------------
  The execution plan of this statement can be improved by creating one or more
  indices.

  Recommendation (estimated benefit: 97.44%)
  ------------------------------------------
  - Consider running the Access Advisor to improve the physical schema design
    or creating the recommended index.
    create index RMS.IDX$$_A2460001 on RMS.DEAL_ITEM_LOC_EXPLODE("ITEM","SUPPLI
    ER","ORIGIN_COUNTRY_ID","LOCATION");

  Rationale
  ---------
    Creating the recommended indices significantly improves the execution plan
    of this statement. However, it might be preferable to run "Access Advisor"
    using a representative SQL workload as opposed to a single statement. This
    will allow to get comprehensive index recommendations which takes into
    account index maintenance overhead and additional space consumption.
```

Preporuke SQL Tuning Advisora

Načini rješavanja problema sa SQL-ovima

```
--------------------------------------------------------------------------------
EXPLAIN PLANS SECTION
--------------------------------------------------------------------------------

1- Original
-----------
Plan hash value: 1636952077

-------------------------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name                  | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                       |    10 |   660 |  1294   (1)| 00:00:01 |       |       |
|   1 |  HASH UNIQUE                  |                       |    10 |   660 |  1294   (1)| 00:00:01 |       |       |
|   2 |   NESTED LOOPS                |                       |       |       |            |          |       |       |
|   3 |    NESTED LOOPS               |                       |    10 |   660 |  1293   (1)| 00:00:01 |       |       |
|   4 |     VIEW                      | VW_DTP_94EC1700       |    10 |   230 |     2  (50)| 00:00:01 |       |       |
|   5 |      HASH UNIQUE              |                       |    10 |   230 |     2  (50)| 00:00:01 |       |       |
|   6 |       INDEX FULL SCAN         | JP_FUTURE_COST_GTT    |   138 |  3174 |     1   (0)| 00:00:01 |       |       |
|   7 |     PARTITION HASH ALL        |                       |     1 |   128 |   (0)| 00:00:01 |     1 |    64 |
|* 8  |      INDEX RANGE SCAN         | PK_DEAL_ITEM_LOC_EXPLODE |  1 |   128 |   (0)| 00:00:01 |     1 |    64 |
|   9 |    TABLE ACCESS BY LOCAL INDEX ROWID| DEAL_ITEM_LOC_EXPLODE |  1 |    89 |   129   (0)| 00:00:01 |     1 |     1 |
-------------------------------------------------------------------------------------------------------------------

2- Using New Indices
--------------------
Plan hash value: 4267730136

-------------------------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name                  | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                       |    10 |   660 |    33   (7)| 00:00:01 |       |       |
|   1 |  HASH UNIQUE                  |                       |    10 |   660 |    33   (7)| 00:00:01 |       |       |
|   2 |   NESTED LOOPS                |                       |       |       |            |          |       |       |
|   3 |    NESTED LOOPS               |                       |    10 |   660 |    32   (4)| 00:00:01 |       |       |
|   4 |     VIEW                      | VW_DTP_94EC1700       |    10 |   230 |     2  (50)| 00:00:01 |       |       |
|   5 |      HASH UNIQUE              |                       |    10 |   230 |     2  (50)| 00:00:01 |       |       |
|   6 |       INDEX FULL SCAN         | JP_FUTURE_COST_GTT    |   138 |  3174 |     1   (0)| 00:00:01 |       |       |
|* 7  |     INDEX RANGE SCAN          | IDX$$_A2460001        |     1 |       |     2   (0)| 00:00:01 |       |       |
|   8 |    TABLE ACCESS BY GLOBAL INDEX ROWID| DEAL_ITEM_LOC_EXPLODE |  1 |    89 |     3   (0)| 00:00:01 | ROWID | ROWID |
-------------------------------------------------------------------------------------------------------------------
```

Načini rješavanja problema sa SQL-ovima

- SQL Plan Baseline
  - poboljšana verzija Stored Outlines
  - osigurava stabilan execution plan bez obzira na promjene u execution environment-u ili statistici objekata u SQL upitu
  - najmoćniji način promjene SQL-a bez promjene programskog kôda
  - ako imam više SQL Plan Baseline-a za određen upit, CBO uzima onaj najeftiniji (najmanji cost)
  - plan mora biti Accepted da bi ga CBO koristio

Načini rješavanja problema sa SQL-ovima

- SQL Plan Baseline
  - može se kreirati iz library cache-a ili iz SQL Tuning Set-a (AWR)
  - može se kopirati baseline sa jednog okruženja na drugo
  - ne traži promjenu kôda i nije direktno vezan za SQL
  - ukoliko CBO misli da je plan bolji od onog u Baseline-u, on će ga automatski dodat u Baseline, ali ga neće koristiti dok nije u statusu Accepted
  - ukoliko je status u Baseline-u Fixed, neće se generirati novi planovi

Načini rješavanja problema sa SQL-ovima

- SQL Plan Baseline
  - najmoćnija opcija je dodavanje SQL-a s lošim *execution planom* u Baseline, ručno kreiranje istog SQL upita uz dodatak hintova koji daje efikasan *execution plan* i konačno pridruživanje efikasnog *execution plana* originalnom SQL-u.
  - druga najkorisnija opcija je korištenje postojećeg *execution plana* iz library cache-a ili AWR-a

Načini rješavanja problema sa SQL-ovima

- SQL Plan Baseline – kako vidjeti execution plan spremljen u baselineu

```
select * from table(dbms_xplan.display_sql_plan_baseline (sql_handle => 'SQL_4ba3eee4a9cd62fa', plan_name => 'SQL_PLAN_4r8zfwknwusru046851ee', format => 'typical'))
```

```
--------------------------------------------------------------------------------
SQL handle: SQL_4ba3eee4a9cd62fa
SQL text: SELECT FC.ROWID FROM FUTURE_COST FC, FUTURE_COST_GTT GTT WHERE GTT.ITEM
          = FC.ITEM AND GTT.SUPPLIER = FC.SUPPLIER AND GTT.ORIGIN_COUNTRY_ID =
          FC.ORIGIN_COUNTRY_ID AND GTT.LOCATION = FC.LOCATION FOR UPDATE OF
          FC.ITEM NOWAIT
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Plan name: SQL_PLAN_4r8zfwknwusru046851ee         Plan id: 73945582
Enabled: YES     Fixed: YES     Accepted: YES     Origin: MANUAL-LOAD
--------------------------------------------------------------------------------

Plan hash value: 3802570450

-----------------------------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name             | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                  | 71764 |  2172K|   144K  (1)| 00:00:04 |       |       |
|   1 |  FOR UPDATE                  |                  |       |       |            |          |       |       |
|   2 |   BUFFER SORT                |                  |       |       |            |          |       |       |
|   3 |    NESTED LOOPS              |                  | 71764 |  2172K|   144K  (1)| 00:00:04 |       |       |
|   4 |     TABLE ACCESS FULL        | FUTURE_COST_GTT  | 71764 |  3013K|   396   (1)| 00:00:01 |       |       |
|   5 |     PARTITION HASH ITERATOR  |                  |     1 |    36 |     2   (0)| 00:00:01 |   KEY |   KEY |
|*  6 |      INDEX RANGE SCAN        | PK_FUTURE_COST   |     1 |    36 |     2   (0)| 00:00:01 |   KEY |   KEY |
-----------------------------------------------------------------------------------------------------------------------

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

   6 - access("GTT"."ITEM"="FC"."ITEM" and "GTT"."SUPPLIER"="FC"."SUPPLIER" and
              "GTT"."ORIGIN_COUNTRY_ID"="FC"."ORIGIN_COUNTRY_ID" AND "GTT"."LOCATION"="FC"."LOCATION")
```

Načini rješavanja problema sa SQL-ovima

- SQL Plan Baseline – explain plan pokazuje da se koristi baseline

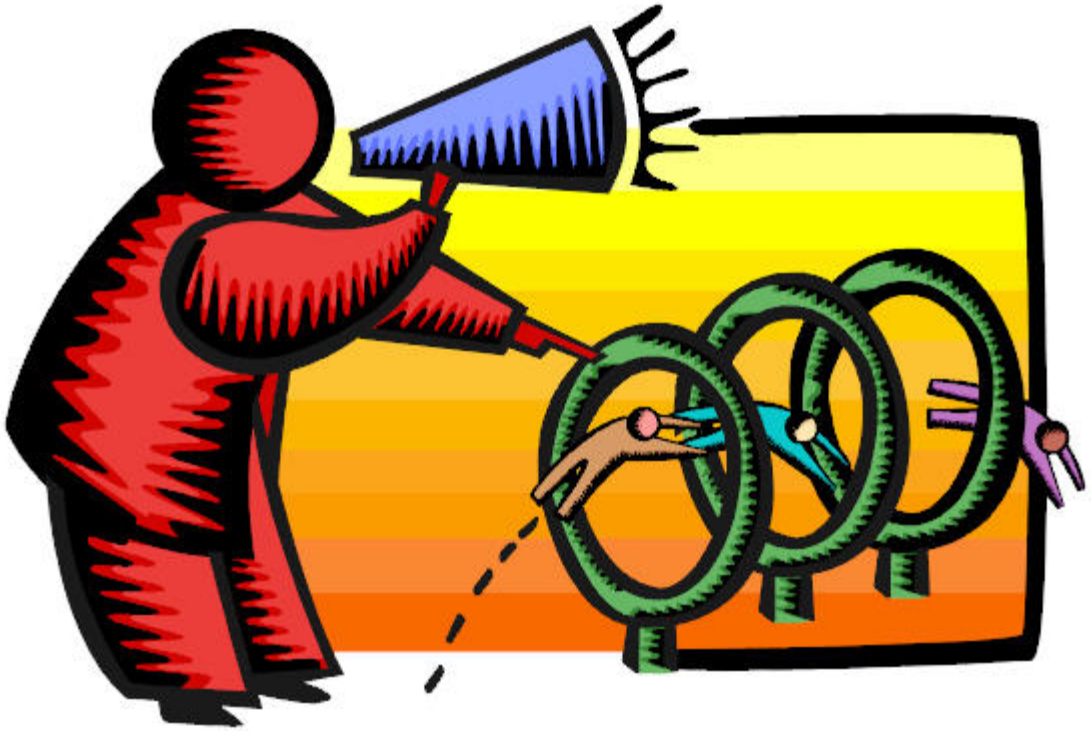
```
explain plan for
select fc.rowid from future_cost fc, future_cost_gtt gtt
where gtt.item = fc.item and gtt.supplier = fc.supplier and gtt.origin_country_id = fc.origin_country_id and gtt.location = fc.location
for update of fc.item nowait;


select * from table(dbms_xplan.display(null,null,'basic +note'));

Plan hash value: 3802570450

-----------------------------------------------------
| Id  | Operation                    | Name           |
-----------------------------------------------------
|   0 | SELECT STATEMENT             |                |
|   1 |  FOR UPDATE                  |                |
|   2 |   BUFFER SORT                |                |
|   3 |    NESTED LOOPS              |                |
|   4 |     TABLE ACCESS FULL        | FUTURE_COST_GTT |
|   5 |     PARTITION HASH ITERATOR|                |
|   6 |      INDEX RANGE SCAN        | PK_FUTURE_COST  |
-----------------------------------------------------

Note
-----
   - SQL plan baseline "SQL_PLAN_4r8zfwknwusru046851ee" used for this statement
```

Načini rješavanja problema sa SQL-ovima

- SQL Plan Baseline – stvarni plan potvrđuje korištenje baseline-a

```
select * from table(DBMS_XPLAN.DISPLAY_CURSOR('dtd6hw8ahs76g', 1, 'ALLSTATS LAST'));

SQL_ID  dtd6hw8ahs76g, child number 1
-------------------------------------
SELECT FC.ROWID FROM FUTURE_COST FC, FUTURE_COST_GTT GTT WHERE GTT.ITEM
= FC.ITEM AND GTT.SUPPLIER = FC.SUPPLIER AND GTT.ORIGIN_COUNTRY_ID =
FC.ORIGIN_COUNTRY_ID AND GTT.LOCATION = FC.LOCATION FOR UPDATE OF
FC.ITEM NOWAIT

Plan hash value: 3802570450

-----------------------------------------------------------------------------------------------
| Id  | Operation                    | Name             | E-Rows |  OMem |  1Mem | Used-Mem |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                  |        |       |       |          |
|   1 |  FOR UPDATE                  |                  |        |       |       |          |
|   2 |   BUFFER SORT                |                  |        |  1024 |  1024 |          |
|   3 |    NESTED LOOPS              |                  |  71764 |       |       |          |
|   4 |     TABLE ACCESS FULL        | FUTURE_COST_GTT  |  71764 |       |       |          |
|   5 |     PARTITION HASH ITERATOR|                  |      1 |       |       |          |
|*  6 |      INDEX RANGE SCAN        | PK_FUTURE_COST   |      1 |       |       |          |
-----------------------------------------------------------------------------------------------

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

   6 - access("GTT"."ITEM"="FC"."ITEM" AND "GTT"."SUPPLIER"="FC"."SUPPLIER" AND
              "GTT"."ORIGIN_COUNTRY_ID"="FC"."ORIGIN_COUNTRY_ID" AND
              "GTT"."LOCATION"="FC"."LOCATION")

Note
-----
   - SQL plan baseline SQL_PLAN_4r8zfwknwusru046851ee used for this statement
   - Warning: basic plan statistics not available. These are only collected when:
       * hint 'gather_plan_statistics' is used for the statement or
       * parameter 'statistics_level' is set to 'ALL', at session or system level
```

Demo



Optimizacija SQL-a bez promjene koda