DBPLUS

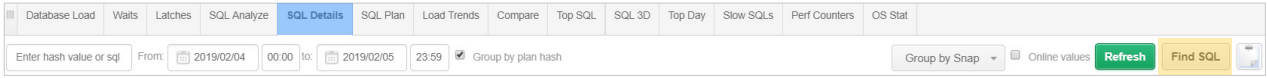
better performance

Performance Monitor for Oracle

Change of the execution plan

Find SQL - search for queries

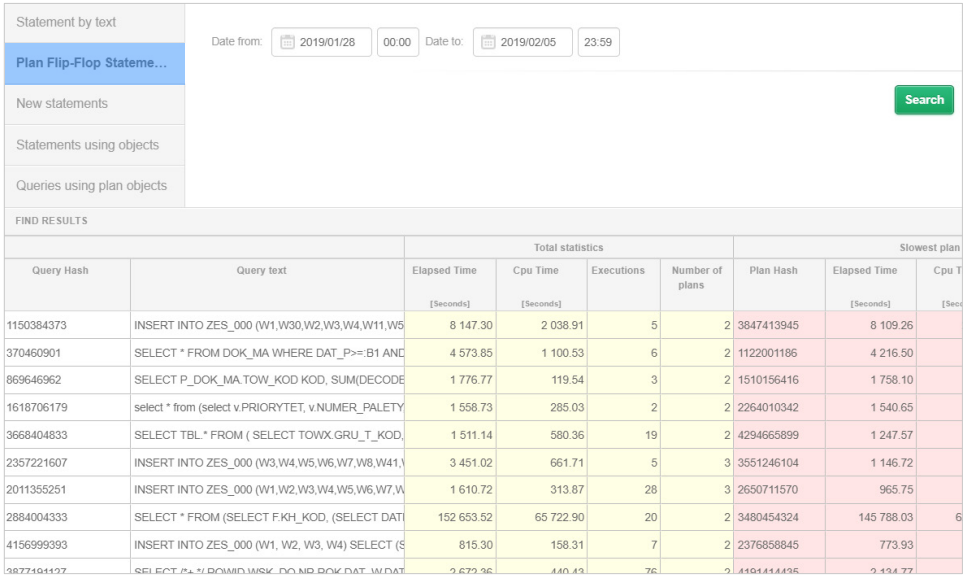
DBPLUS
better performance



Changing the execution plan is one of the most common reasons for the degradation of database performance.

Searching for queries that change the plan is possible in the SQL Details screen in **Find SQL**.

In the **Plan Flip-Flop Statement** tab, we select the time period we are interested in and receive queries with changed plans.



		Total statistics				Slowest plan		
Query Hash	Query text	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Number of plans	Plan Hash	Elapsed Time [Seconds]	Cpu T
1150384373	INSERT INTO ZES_000 (W1,W30,W2,W3,W4,W11,W5	8 147.30	2 038.91	5	2	3847413945	8 109.26	
370460901	SELECT * FROM DOK_MA WHERE DAT_P>=:B1 AND	4 573.85	1 100.53	6	2	1122001186	4 216.50	
869646962	SELECT P_DOK_MA.TOW_KOD KOD, SUM(DECODE	1 776.77	119.54	3	2	1510156416	1 758.10	
1618706179	select * from (select v.PRIORYTET, v.NUMER_PALETY	1 558.73	285.03	2	2	2264010342	1 540.65	
3668404833	SELECT TBL.* FROM (SELECT TOWX.GRU_T_KOD,	1 511.14	580.36	19	2	4294665899	1 247.57	
2357221607	INSERT INTO ZES_000 (W3,W4,W5,W6,W7,W8,W41,\	3 451.02	661.71	5	3	3551246104	1 146.72	
2011355251	INSERT INTO ZES_000 (W1,W2,W3,W4,W5,W6,W7,W	1 610.72	313.87	28	3	2650711570	965.75	
2884004333	SELECT * FROM (SELECT F.KH_KOD, (SELECT DATI	152 653.52	65 722.90	20	2	3480454324	145 788.03	6
4156999393	INSERT INTO ZES_000 (W1, W2, W3, W4) SELECT (S	815.30	158.31	7	2	2376858845	773.93	
3877191127	SELECT /*+ */ ROWID,WSK_DO,NR_ROK,DAT_W,DAT	2 672.26	440.43	76	2	4191414435	2 134.77	

Find SQL

DBPLUS
better performance

As a result of the search, we get the performance statistics of the query plans. On their basis, we can estimate the impact of ***changing the plan*** on the database performance.

FIND RESULTS

	Fastest plan statistics					Slowest vs Fastest		Estimation statistics	
Elapsed Time Per 1 exec [Seconds]	Plan Hash	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Elapsed Time Per 1 exec [Seconds]	Times faster	Elapsed Time Per 1 exec difference [Seconds]	Elapsed Time to reduce [Seconds]	Cpu Time to reduce [Seconds]
3 209.6195	2213930537	576.61	248.96	16	36.0384	89	3 173.5812	**50 777.2984**	**3 180.1229**
0.0002	3757328946	73.72	28.82	544 227	0.0001	1	0.0000	**5 924.9251**	**2 447.0847**
0.0288	789180689	8.69	3.73	513	0.0169	2	0.0119	**6 564.2172**	**2 847.9567**
4 903.2513	595788847	434.55	85.86	8	54.3186	90	4 848.9327	**14 546.7981**	**1 413.2916**
0.0013	195326030	41.13	17.81	50 827	0.0008	2	0.0005	**5 181.0533**	**2 012.3470**
0.3443	4169529607	113.58	44.42	32 406	0.0035	98	0.3408	**11 298.5179**	**4 537.1475**
1.7984	2082425496	4 424.43	1 597.07	2 519	1.7564	1	0.0419	**125.9913**	**107.3046**
398.0893	3782389031	15.04	6.68	2	7.5186	53	390.5707	**8 983.1265**	**555.3427**

After verification of the statistics, we proceed to a detailed analysis by clicking the **[+]** button in the column with the query identifier.

FIND RESULTS

		Total statistics				Slowest plan		
Query Hash	Query text	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Number of plans	Plan Hash	Elapsed Time [Seconds]	Cpu Ti [Seco
355833246	INSERT INTO ZES_000 (W1, W2, W3, W4, W30, W31	51 930.53	3 678.05	32	2	2169775936	51 353.91	
1230330421	SELECT GRU_T.KOD FROM GRU_T,TOW_GRU_T W	33 388.73	13 185.92	202 755 194	2	434371058	33 315.01	1
3830132343	SELECT * FROM (SELECT IT.TAG_NAME, INN.ID, INN	15 952.20	6 881.13	554 128	2	4238870180	15 943.51	
1394045510	SELECT DOK_MA.ROD_D_KOD||' '||DOK_MA.NR||' '|[	15 144.30	1 531.36	11	2	2634859585	14 709.75	
2542634967	OM ((SELECT NVL(S	13 564.35	5 641.50	10 359 904	2	1967382841	13 523.22	
19353795	R,ROK,DAT_W,DAT	11 528.29	4 627.00	65 556	2	2941142135	11 414.71	
2514954702	nr, pal_nr_zb, deco	9 826.73	3 608.94	5 523	2	1589438509	5 402.29	
171454451	INSERT INTO ZES_000 (W1,W2,W3,W4,W5,W6,W7,W	9 171.09	638.81	25	2	2884876068	9 156.05	

Query: 1394045510

View sql details

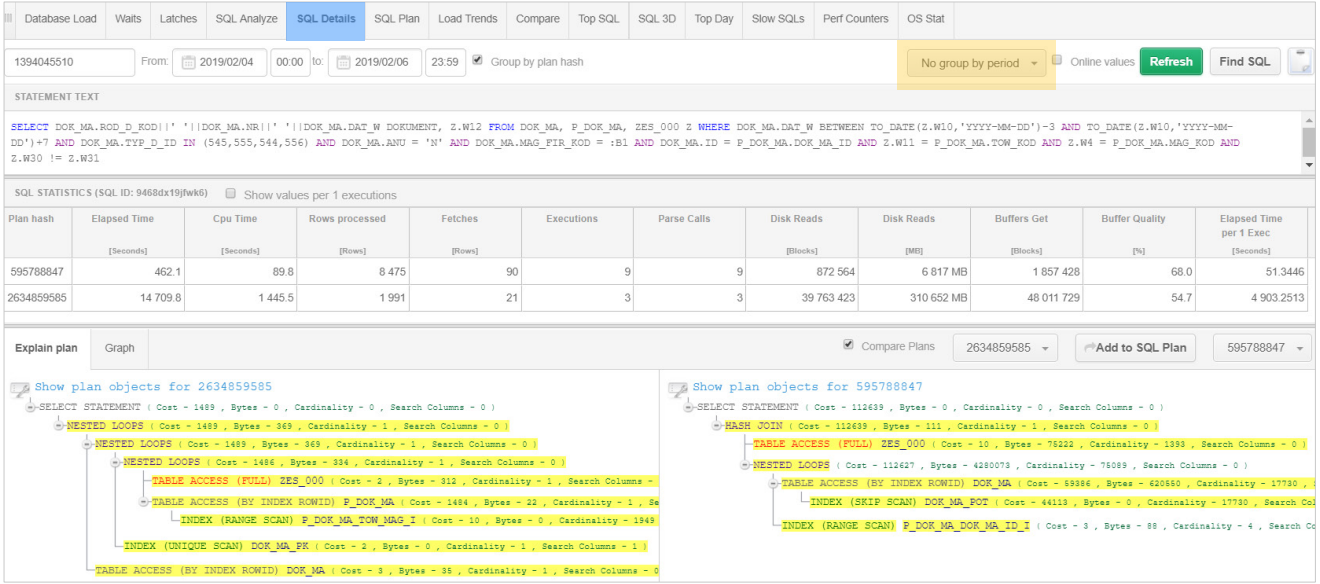
Add to query hash value list

SQL Details

DBPLUS
better performance

By choosing the **No group by period** grouping options, we get information about statistics for each of the plans in the selected period of time.

Thanks to the functionality of **Compare plans,** we have the opportunity to analyse plans in detail, indicate differences and suggest a solution to the problem.



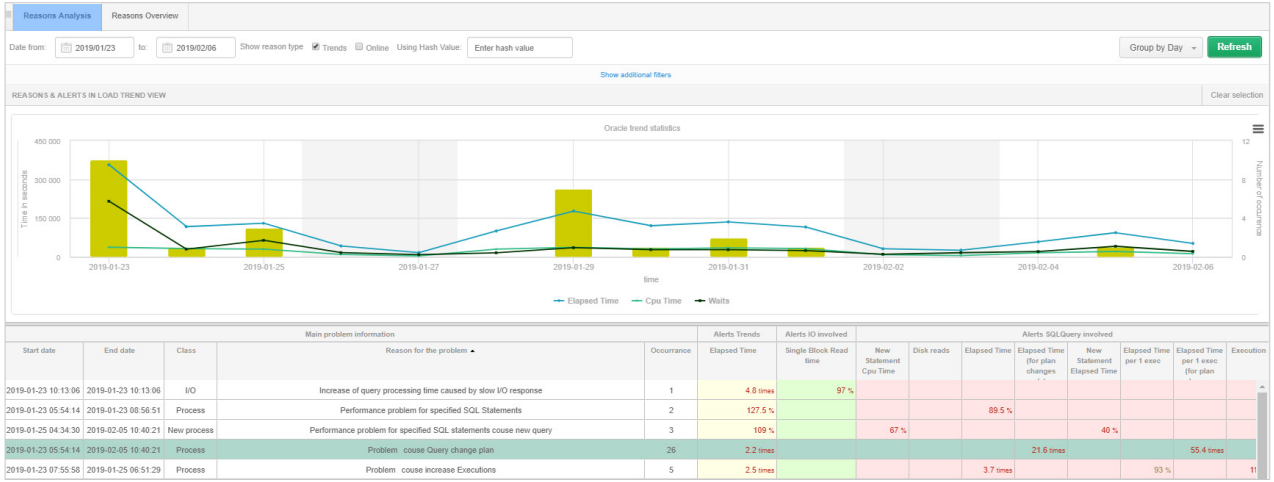
Anomaly Monitor

DBPLUS
better performance

Thanks to the extended alert module, we can check at any time how **often** there are problems in the database, including **changing the** query **plans**.

The information in the graph indicates **when** and **what** alert has occurred.

The table presents information on exceeding the **alert threshold** for statistics included in the **alert**.



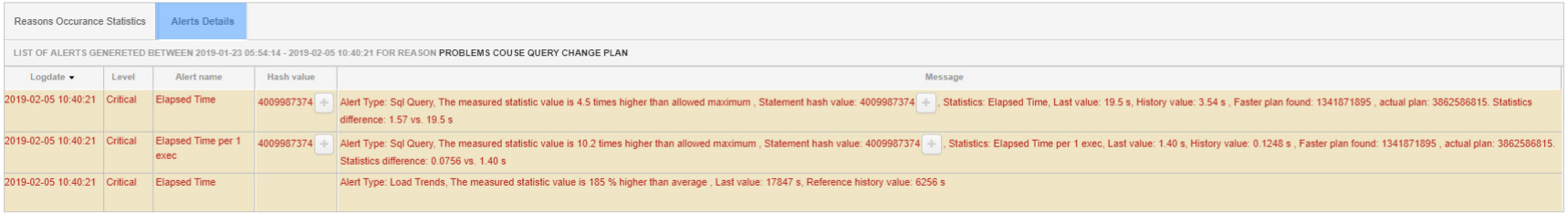
Main problem information					Alerts Trends	Alerts IO involved	Alerts SQLQuery involved							
Start date	End date	Class	Reason for the problem	Occurrance	Elapsed Time	Single Block Read time	New Statement Cpu Time	Disk reads	Elapsed Time	Elapsed Time (for plan changes	New Statement Elapsed Time	Elapsed Time per 1 exec	Elapsed Time per 1 exec (for plan	Execution
2019-01-23 10:13:06	2019-01-23 10:13:06	I/O	Increase of query processing time caused by slow I/O response	1	4.8 times	97 %								
2019-01-23 05:54:14	2019-01-23 08:56:51	Process	Performance problem for specified SQL Statements	2	127.5 %				89.5 %					
2019-01-25 04:34:30	2019-02-05 10:40:21	New process	Performance problem for specified SQL statements couse new query	3	109 %		67 %				40 %			
2019-01-23 05:54:14	2019-02-05 10:40:21	Process	Problem couse Query change plan	26	2.2 times					21.6 times			55.4 times	
2019-01-23 07:55:58	2019-01-25 06:51:29	Process	Problem couse increase Executions	5	2.5 times				3.7 times			93 %		11

Anomaly Monitor

DBPLUS
better performance

Each alert informing about a change to the execution plan contains information about:

- Query identifier
- What statistic has been exceeded
- An indication of a faster plan for the query, if it existed in the past.



Reasons Occurance Statistics | Alerts Details

LIST OF ALERTS GENERETED BETWEEN 2019-01-23 05:54:14 - 2019-02-05 10:40:21 FOR REASON PROBLEMS COUSE QUERY CHANGE PLAN

Logdate	Level	Alert name	Hash value	Message
2019-02-05 10:40:21	Critical	Elapsed Time	4009987374	Alert Type: Sql Query, The measured statistic value is 4.5 times higher than allowed maximum , Statement hash value: 4009987374 , Statistics: Elapsed Time, Last value: 19.5 s, History value: 3.54 s , Faster plan found: 1341871895 , actual plan: 3862586815. Statistics difference: 1.57 vs. 19.5 s
2019-02-05 10:40:21	Critical	Elapsed Time per 1 exec	4009987374	Alert Type: Sql Query, The measured statistic value is 10.2 times higher than allowed maximum , Statement hash value: 4009987374 , Statistics: Elapsed Time per 1 exec, Last value: 1.40 s, History value: 0.1248 s , Faster plan found: 1341871895 , actual plan: 3862586815. Statistics difference: 0.0756 vs. 1.40 s
2019-02-05 10:40:21	Critical	Elapsed Time		Alert Type: Load Trends, The measured statistic value is 185 % higher than average , Last value: 17847 s, Reference history value: 6256 s

DBPLUS
better performance

Thank you

www.dbplus.tech