DBPLUS
better performance

Performance Monitor for Oracle

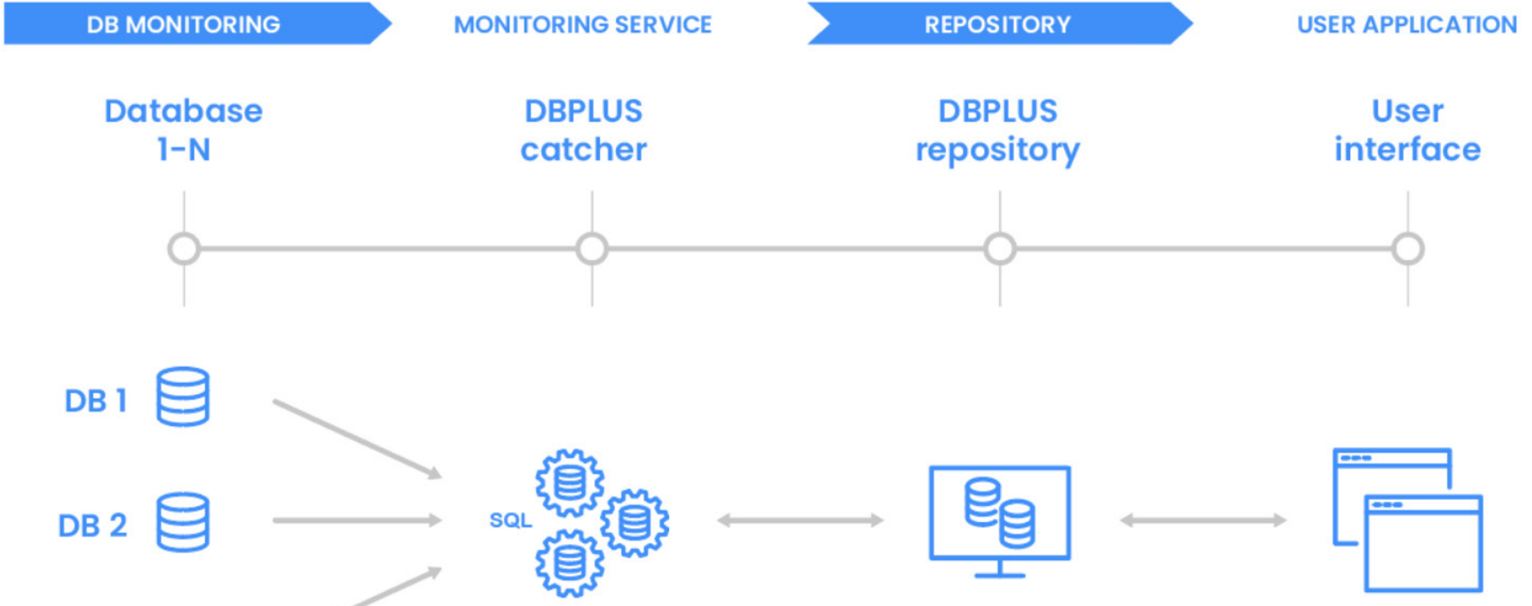
dbplus.tech

Agenda

DBPLUS
better performance

1. Solution architecture
2. Connection of the database to the monitoring service
3. The most important functionalities
4. Access management - Security module
5. Problem Monitor - Anomaly Monitor
6. Working with the program

System architecture



A set of SQL procedures responsible for collecting information about the performance of the monitored databases

Web application based on IIS technology

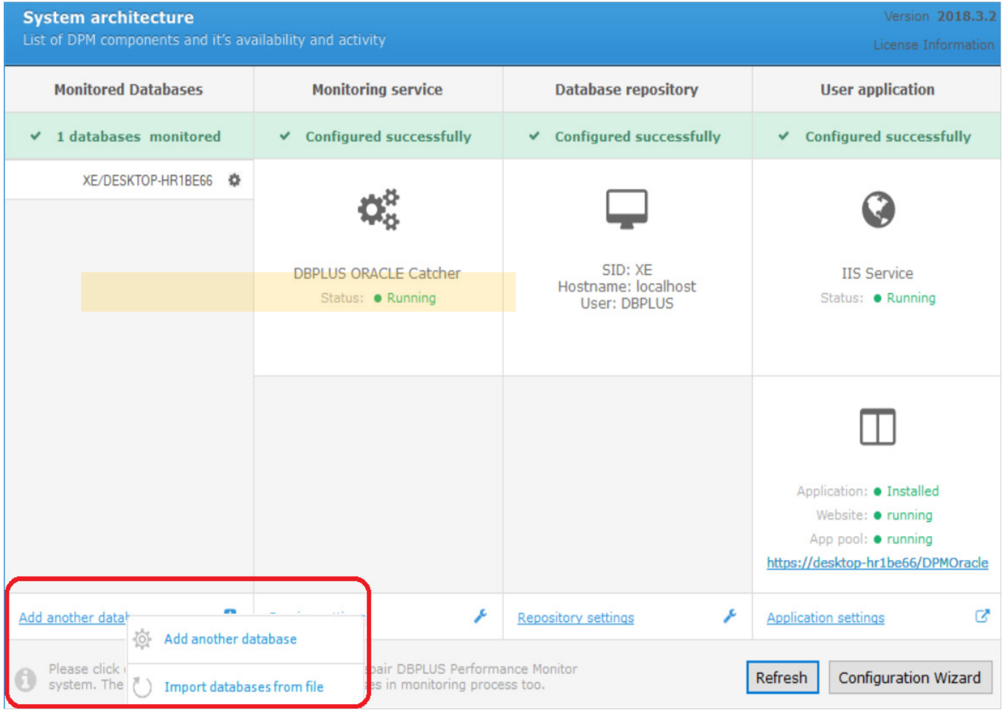
Connecting database to monitoring service

DBPLUS
better performance

In the main system configurator window (Configuration Wizard), click the [Add Another database] button.

Two ways to connect the database:

- Manual connection (a single database)
- Import from file



System architecture
List of DPM components and it's availability and activity

Version 2018.3.2
License Information

Monitored Databases	Monitoring service	Database repository	User application
✓ 1 databases monitored	✓ Configured successfully	✓ Configured successfully	✓ Configured successfully
XE/DESKTOP-HR1BE66	DBPLUS ORACLE Catcher Status: ● Running	SID: XE Hostname: localhost User: DBPLUS	IIS Service Status: ● Running
			Application: ● Installed Website: ● running App pool: ● running https://desktop-hr1be66/DPMOracle
Add another datal		Repository settings	Application settings

Please click system. The pair DBPLUS Performance Monitor es in monitoring process too.

Connecting database to monitoring service

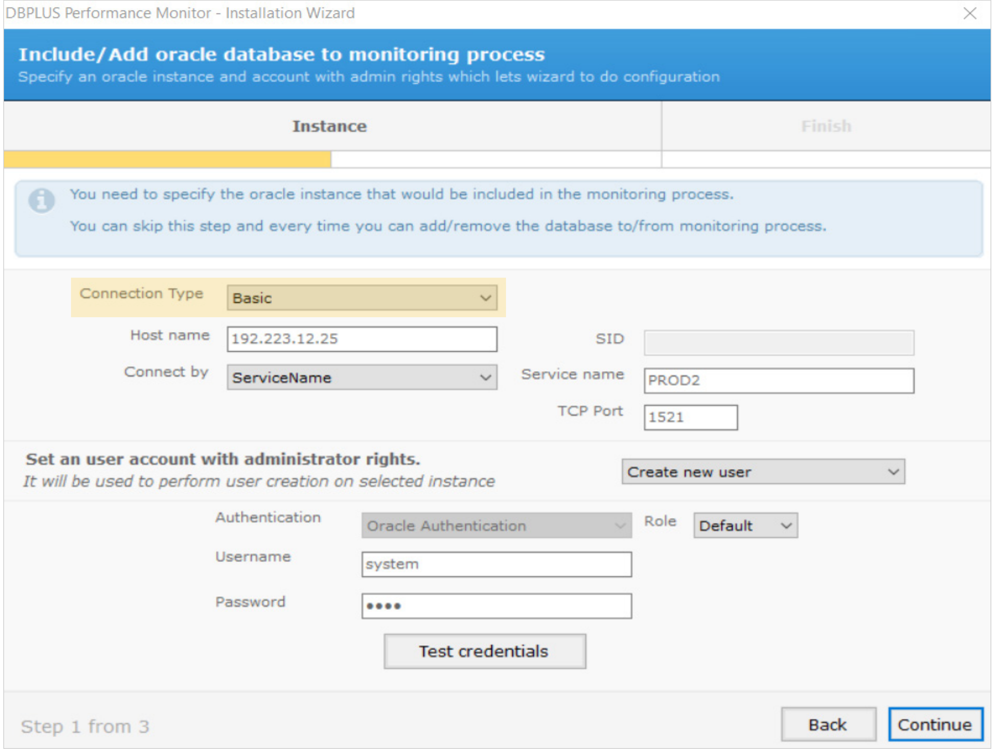
DBPLUS
better performance

Adding a single database:

- Choosing the connection type: Basic

Important!
You can create a DBPLUS user
on the monitored database
at the configurator level.

- Creating a new user in the monitored database is possible only by entering the username and password of a user with Administrator rights.



DBPLUS Performance Monitor - Installation Wizard

Include/Add oracle database to monitoring process
Specify an oracle instance and account with admin rights which lets wizard to do configuration

You need to specify the oracle instance that would be included in the monitoring process.
You can skip this step and every time you can add/remove the database to/from monitoring process.

Connection Type: Basic
Host name: 192.223.12.25
Connect by: ServiceName
SID:
Service name: PROD2
TCP Port: 1521

Set an user account with administrator rights.
It will be used to perform user creation on selected instance

Create new user

Authentication: Oracle Authentication
Role: Default
Username: system
Password: ••••

Step 1 from 3

Connecting database to monitoring service

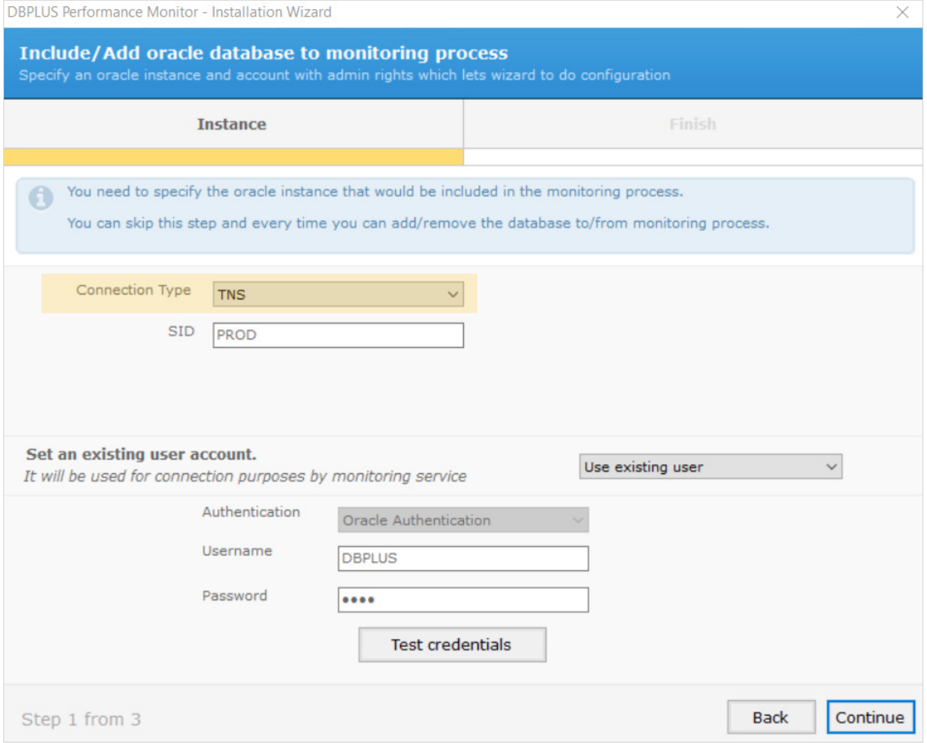
DBPLUS
better performance

Adding a single database:

- Choosing the connection type: TNS

Important!
You can create a DBPLUS user
on the monitored database
at the configurator level.

- Creating a new user in the monitored database is possible only by entering the username and password of a user with Administrator rights.



DBPLUS Performance Monitor - Installation Wizard

Include/Add oracle database to monitoring process
Specify an oracle instance and account with admin rights which lets wizard to do configuration

You need to specify the oracle instance that would be included in the monitoring process.

You can skip this step and every time you can add/remove the database to/from monitoring process.

Connection Type: TNS
SID: PROD

Set an existing user account.
It will be used for connection purposes by monitoring service

Use existing user

Authentication: Oracle Authentication
Username: DBPLUS
Password: ••••

Step 1 from 3

Connecting database to monitoring service

DBPLUS
better performance

Batch connection from a file:

File structure:

- USERNAME,PASSWORD, SID

or

- USERNAME,PASSWORD,SID, HOSTNAME,TCPPORT

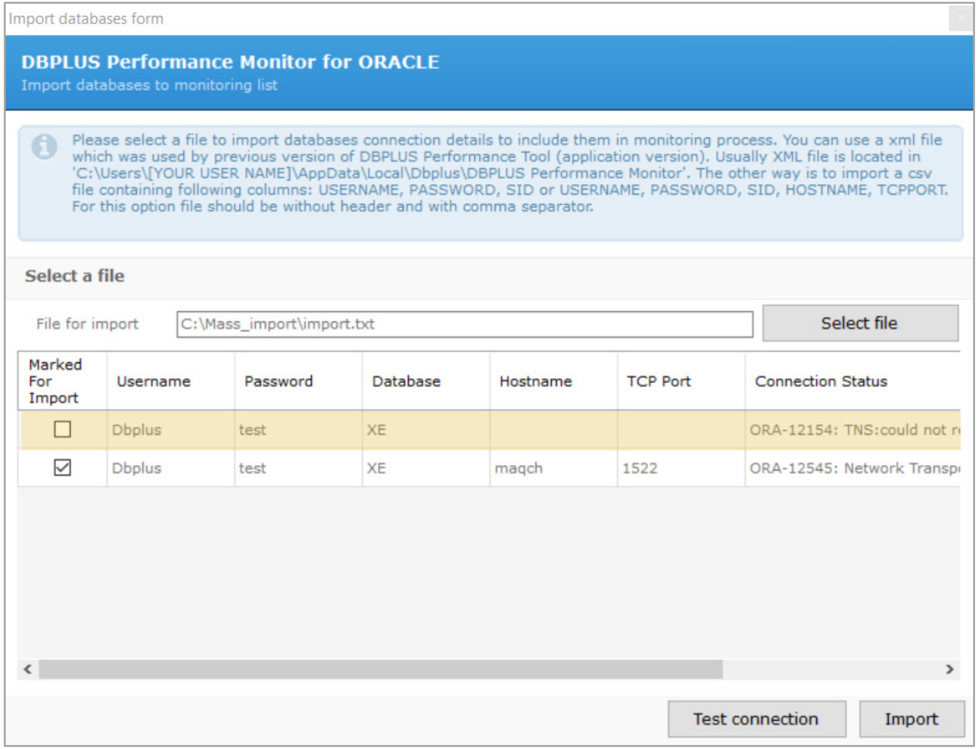
Sample file:

- Dbplus, test, XE

or

- Dbplus, test, XE, maqch, 1522

Important!
The DBPLUS user must have been previously created in the database for the connection to be successful.



Import databases form

DBPLUS Performance Monitor for ORACLE
Import databases to monitoring list

Please select a file to import databases connection details to include them in monitoring process. You can use a xml file which was used by previous version of DBPLUS Performance Tool (application version). Usually XML file is located in 'C:\Users\[YOUR USER NAME]\AppData\Local\Dbplus\DBPLUS Performance Monitor'. The other way is to import a csv file containing following columns: USERNAME, PASSWORD, SID or USERNAME, PASSWORD, SID, HOSTNAME, TCPPORT. For this option file should be without header and with comma separator.

Select a file

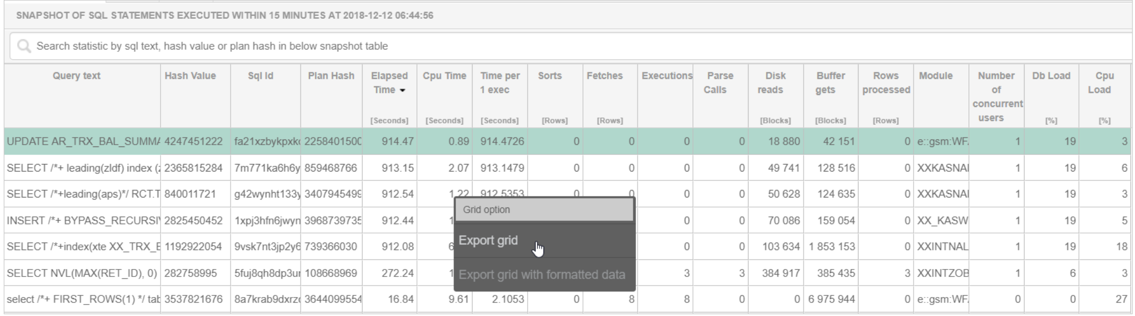
File for import: C:\Mass_import\import.txt

Marked For Import	Username	Password	Database	Hostname	TCP Port	Connection Status
□	Dbplus	test	XE			ORA-12154: TNS:could not r
☑	Dbplus	test	XE	maqch	1522	ORA-12545: Network Transp

Table options

DBPLUS
better performance

- The possibility of exporting to a CSV file.



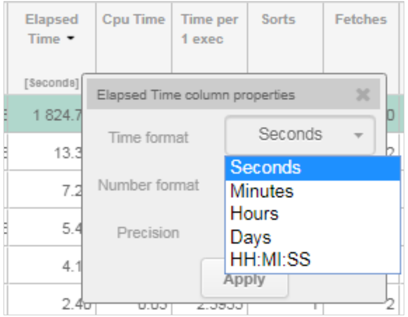
SNAPSHOT OF SQL STATEMENTS EXECUTED WITHIN 15 MINUTES AT 2018-12-12 06:44:56

Search statistic by sql text, hash value or plan hash in below snapshot table

Query text	Hash Value	Sql Id	Plan Hash	Elapsed Time ▾ [Seconds]	Cpu Time [Seconds]	Time per 1 exec [Seconds]	Sorts [Rows]	Fetches [Rows]	Executions	Parse Calls	Disk reads [Blocks]	Buffer gets [Blocks]	Rows processed [Rows]	Module	Number of concurrent users	Db Load [%]	Cpu Load [%]
UPDATE AR_TRX_BAL_SUMM/	4247451222	fa21xzbykpxk	225840150	914.47	0.89	914.4726	0	0	0	0	18 880	42 151	0	e::gsm:WF.	1	19	3
SELECT /*+ leading(zldf) index (:	2365815284	7m771ka6h6y	859468766	913.15	2.07	913.1479	0	0	0	0	49 741	128 516	0	XXKASNAI	1	19	6
SELECT /*+leading(aps)*/ RCT.T	840011721	g42wynht133y	340794549	912.54	1.22	912.5353	0	0	0	0	50 628	124 635	0	XXKASNAI	1	19	3
INSERT /*+ BYPASS_RECURSI	2825450452	1xpj3hfn6jwyn	396873973	912.44	1				0	0	70 086	159 054	0	XX_KASW	1	19	5
SELECT /*+index(xte XX_TRX_E	1192922054	9vsk7nt3jp2y6	739366030	912.08	6				0	0	103 634	1 853 153	0	XXINTNAL	1	19	18
SELECT NVL(MAX(RET_ID), 0)	282758995	5fuj8qh8dp3u	108668969	272.24	1				3	3	384 917	385 435	3	XXINTZOB	1	6	3
select /*+ FIRST_ROWS(1) */ tal	3537821676	8a7krab9dxrz	364409955	16.84	9.61	2.1053	0	8	8	0	0	6 975 944	0	e::gsm:WF.	0	0	27

Sorting and Formatting columns in tables:

- *unit* selection - e.g. Elapsed Time in seconds, minutes, days, etc.
- *selection of a shortcut for large numbers -* e.g. **k**ilo, **M**ega, ...
- *determination of the decimal place accuracy* of a number.



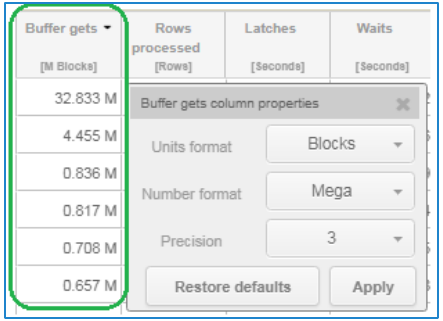
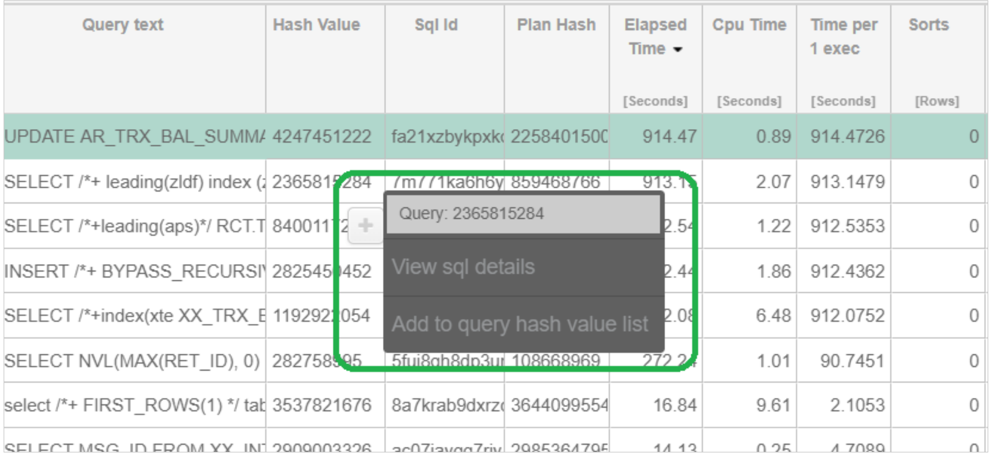
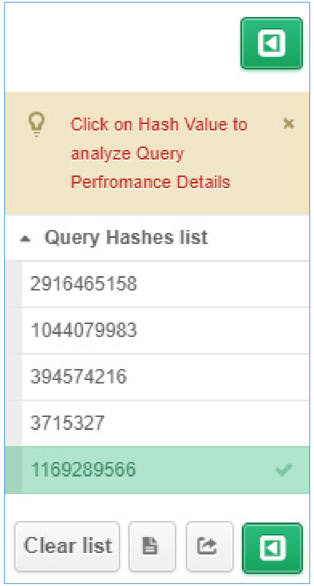


Table options

- The [+] button is presented in the Hash Value column
- It allows you to quickly go to the query details (SQL Details) or
- to add a query to the clipboard with a list of queries for later analysis (SQL Details)



Query text	Hash Value	Sql Id	Plan Hash	Elapsed Time ▾	Cpu Time	Time per 1 exec	Sorts
				[Seconds]	[Seconds]	[Seconds]	[Rows]
UPDATE AR_TRX_BAL_SUMM/	4247451222	fa21xzbykpxk(	225840150(	914.47	0.89	914.4726	0
SELECT /*+ leading(zldf) index (z	2365815284	7m771ka6h6y	859468766	913.1	2.07	913.1479	0
SELECT /*+leading(aps)*/ RCT.T	840011 2			2.54	1.22	912.5353	0
INSERT /*+ BYPASS_RECURSI\	282545 452			2.44	1.86	912.4362	0
SELECT /*+index(xte XX_TRX_E	1192922054			2.08	6.48	912.0752	0
SELECT NVL(MAX(RET_ID), 0)	282758 95	5fui8gh8dp3ur	108668969	272.2	1.01	90.7451	0
select /*+ FIRST_ROWS(1) */ tab	3537821676	8a7krab9dxrz(	3644099554	16.84	9.61	2.1053	0
SELECT MSG_ID FROM XX_IN	2909003326	ac07iavqg7riv	298536479	14.13	0.25	4.7089	0



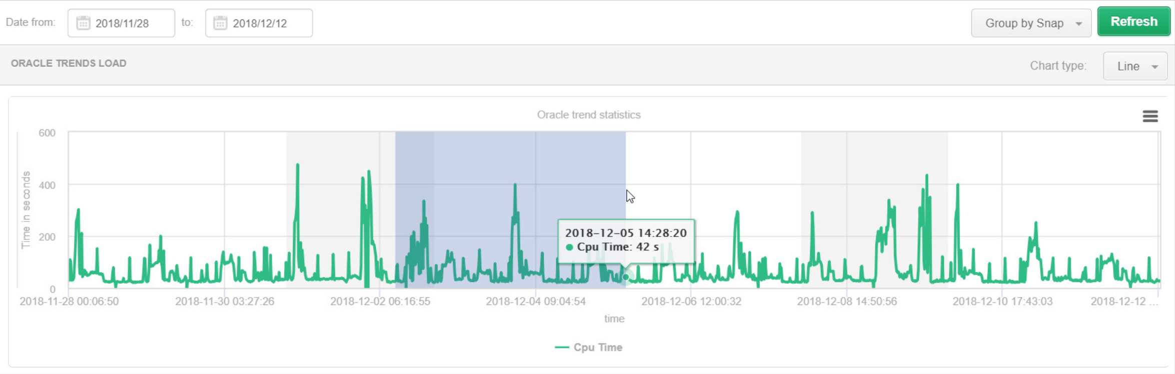
Click on Hash Value to analyze Query Perfromance Details

▴ Query Hashes list

2916465158
1044079983
394574216
3715327
1169289566 ✓

Table options

- Zooming in the selected area on the chart



Date from: 2018/11/28 to: 2018/12/12

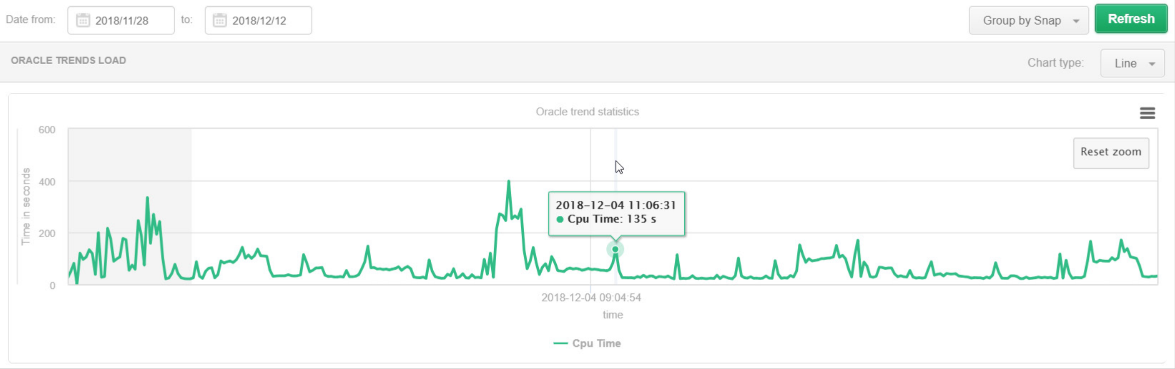
Group by Snap

Refresh

ORACLE TRENDS LOAD

Chart type: Line

- Option to return to the previous view via [Reset zoom]



Date from: 2018/11/28 to: 2018/12/12

Group by Snap

Refresh

ORACLE TRENDS LOAD

Chart type: Line

Table options

DBPLUS
better performance

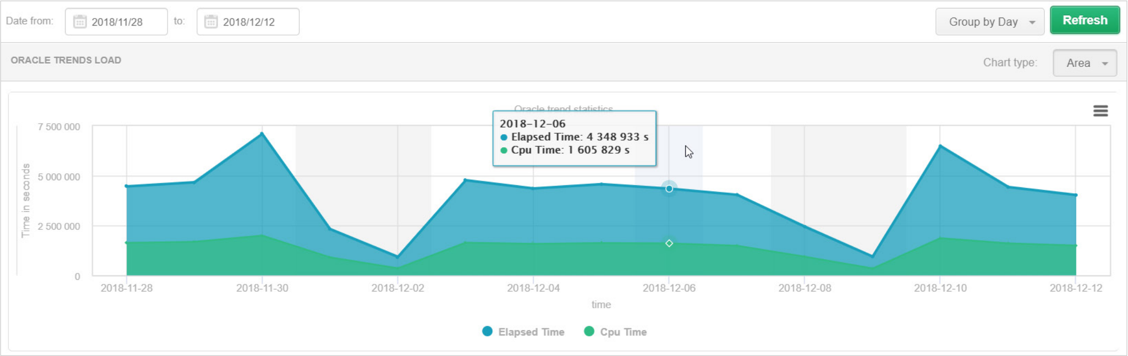
Different types of charts:

- *Line*
- *Area*
- *Column*

It is possible to mark and unmark the presented series on the chart.

Displaying information in a *Tooltip* after indicating the location on the chart.

The chart can be exported to a file in the following formats:
PNG, JPEG, PDF, SVG



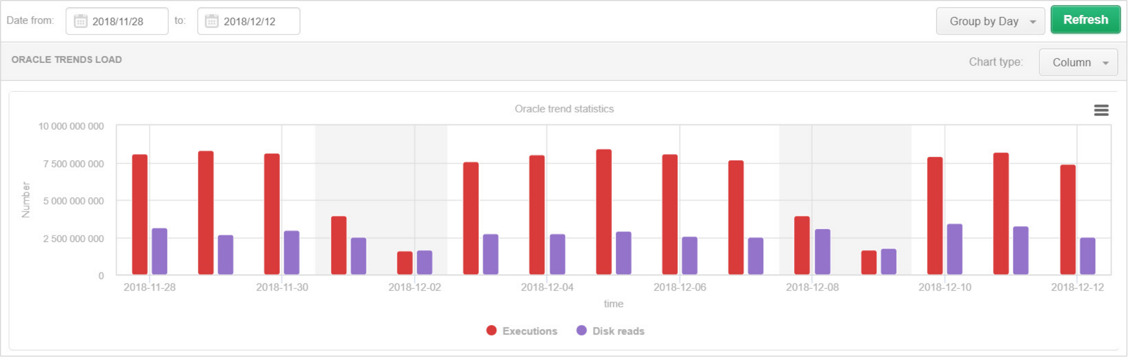
Date from: 2018/11/28 to: 2018/12/12

Group by Day

Chart type: Area

ORACLE TRENDS LOAD

2018-12-06
● Elapsed Time: 4 348 933 s
● Cpu Time: 1 605 829 s



Date from: 2018/11/28 to: 2018/12/12

Group by Day

Chart type: Column

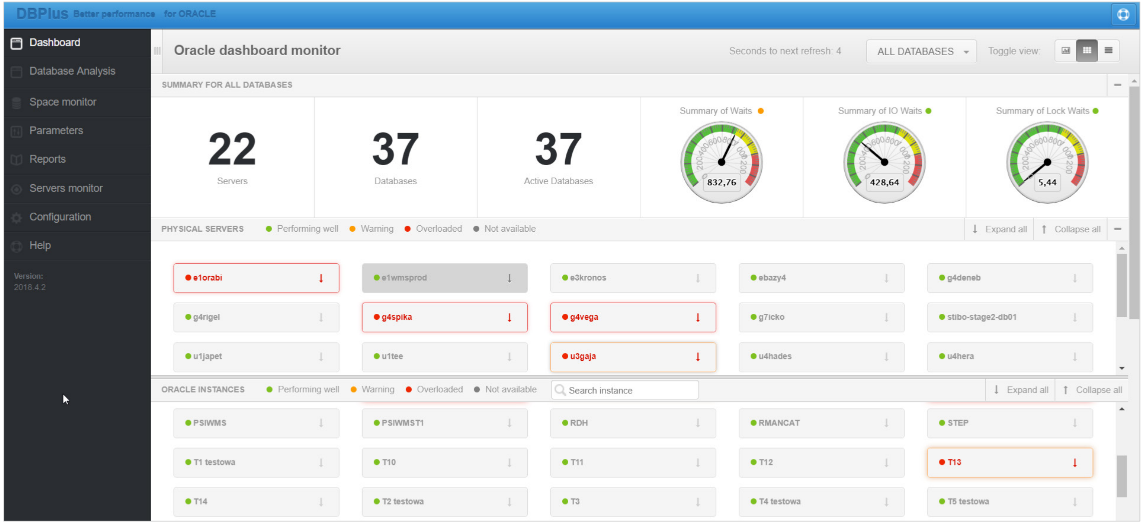
ORACLE TRENDS LOAD

Dashboard – Home screen

DBPLUS
better performance

Three different views
for presenting databases:

- *Icons view*

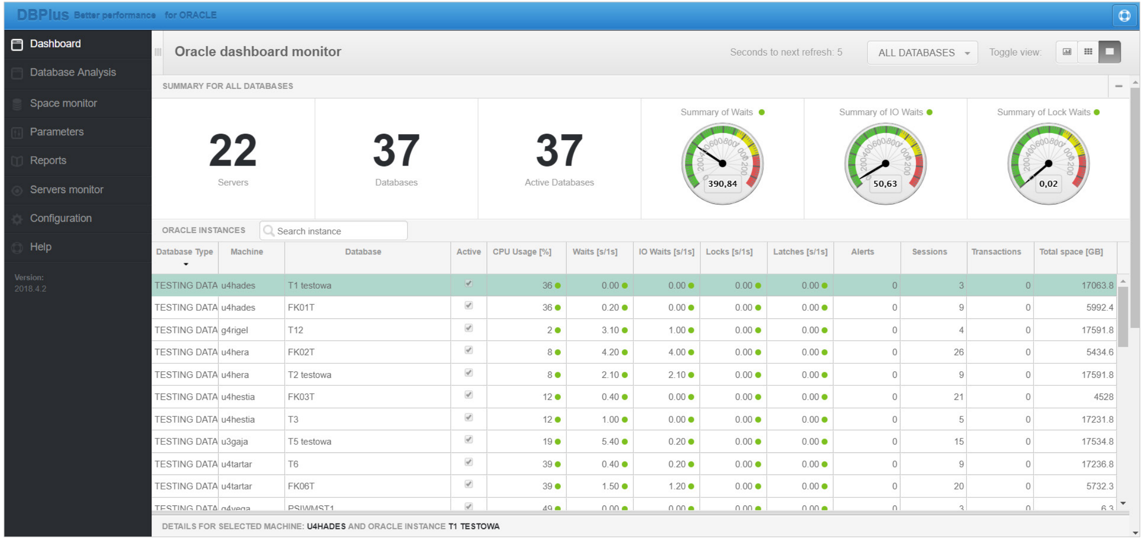


Dashboard – Home screen

DBPLUS
better performance

Three different views
for presenting databases:

- *Grid view*



Oracle dashboard monitor

Seconds to next refresh: 5

SUMMARY FOR ALL DATABASES

22 Servers

37 Databases

37 Active Databases

ORACLE INSTANCES Search instance

Database Type	Machine	Database	Active	CPU Usage [%]	Waits [s/1s]	IO Waits [s/1s]	Locks [s/1s]	Latches [s/1s]	Alerts	Sessions	Transactions	Total space [GB]
TESTING DATA	u4hades	T1 testowa	☑	36 ●	0.00 ●	0.00 ●	0.00 ●	0.00 ●	0	3	0	17063.8
TESTING DATA	u4hades	FK01T	☑	36 ●	0.20 ●	0.00 ●	0.00 ●	0.00 ●	0	9	0	5992.4
TESTING DATA	g4rigel	T12	☑	2 ●	3.10 ●	1.00 ●	0.00 ●	0.00 ●	0	4	0	17591.8
TESTING DATA	u4hera	FK02T	☑	8 ●	4.20 ●	4.00 ●	0.00 ●	0.00 ●	0	26	0	5434.6
TESTING DATA	u4hera	T2 testowa	☑	8 ●	2.10 ●	2.10 ●	0.00 ●	0.00 ●	0	9	0	17591.8
TESTING DATA	u4hestia	FK03T	☑	12 ●	0.40 ●	0.00 ●	0.00 ●	0.00 ●	0	21	0	4528
TESTING DATA	u4hestia	T3	☑	12 ●	1.00 ●	0.00 ●	0.00 ●	0.00 ●	0	5	0	17231.8
TESTING DATA	u3gaja	T5 testowa	☑	19 ●	5.40 ●	0.20 ●	0.00 ●	0.00 ●	0	15	0	17534.8
TESTING DATA	u4tartar	T6	☑	39 ●	0.40 ●	0.20 ●	0.00 ●	0.00 ●	0	9	0	17236.8
TESTING DATA	u4tartar	FK06T	☑	39 ●	1.50 ●	1.20 ●	0.00 ●	0.00 ●	0	20	0	5732.3
TESTING DATA	g4vega	PSIWMST1	☑	49 ●	0.00 ●	0.00 ●	0.00 ●	0.00 ●	0	3	0	6.3

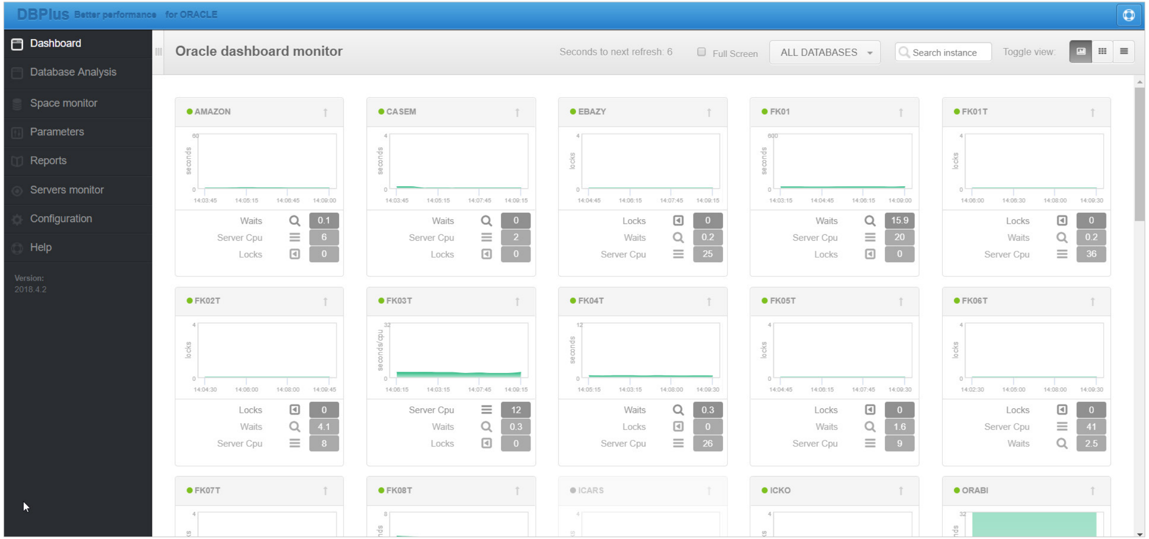
DETAILS FOR SELECTED MACHINE: **U4HADES** AND ORACLE INSTANCE **T1 TESTOWA**

Dashboard – Home screen

DBPLUS
better performance

Three different views
for presenting databases:

- *Television view*



Oracle dashboard monitor

Seconds to next refresh: 6

AMAZON
Waits 0.1
Server Cpu 6
Locks 0

CASEM
Waits 0
Server Cpu 2
Locks 0

EBAZY
Locks 0
Waits 0.2
Server Cpu 25

FK01
Waits 15.9
Server Cpu 20
Locks 0

FK01T
Locks 0
Waits 0.2
Server Cpu 36

FK02T
Locks 0
Waits 4.1
Server Cpu 8

FK03T
Server Cpu 12
Waits 0.3
Locks 0

FK04T
Waits 0.3
Locks 0
Server Cpu 26

FK05T
Locks 0
Waits 1.6
Server Cpu 9

FK06T
Locks 0
Server Cpu 41
Waits 2.5

FK07T

FK08T

ICARS

ICKO

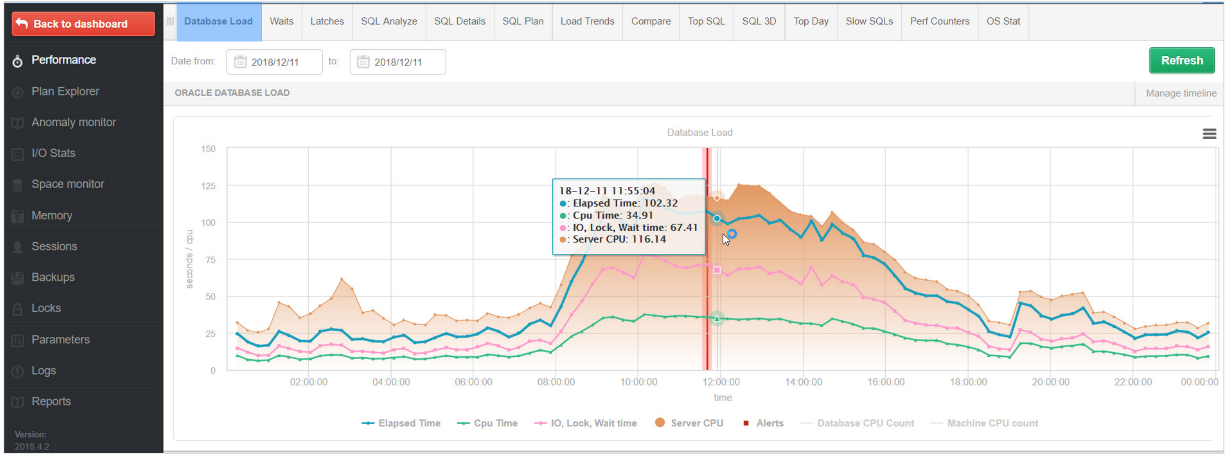
ORABI

Database Load – database details

DBPLUS
better performance

The chart presents information about the basic statistics of the database:

- *Elapsed Time*
- *CPU Time*
- *IO, Lock, Wait Time*
- *Server CPU*
- *Alerts*
- *CPU Usage*



Date from: 2018/12/11 to: 2018/12/11

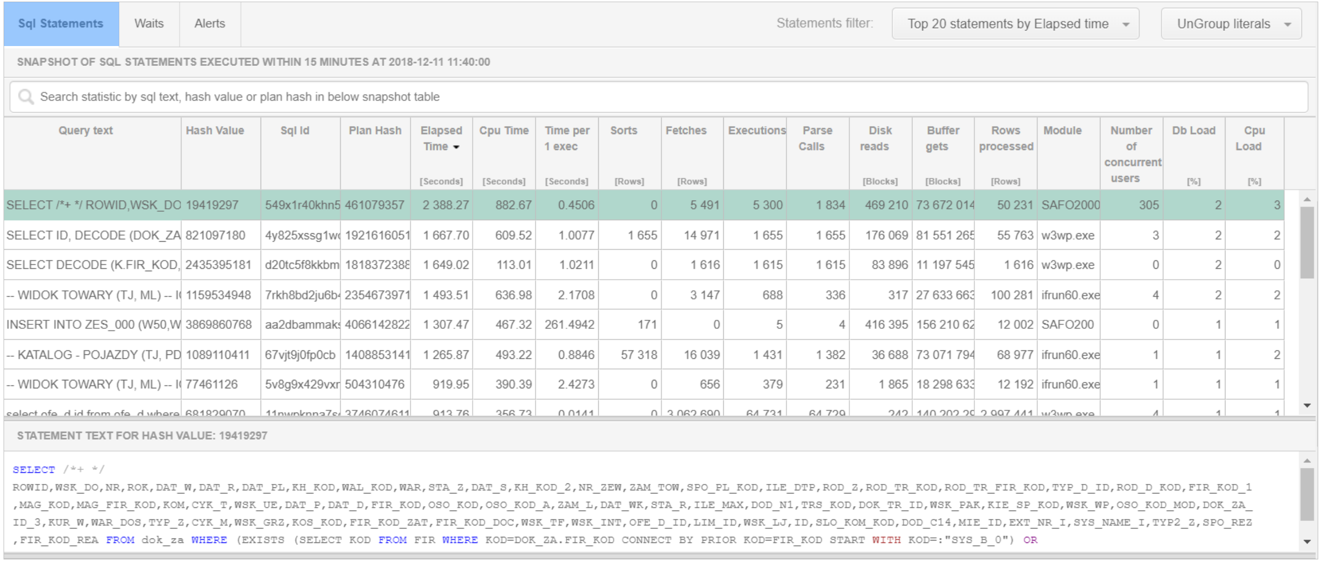
18-12-11 11:55:04
● : Elapsed Time: 102.32
● : Cpu Time: 34.91
● : IO, Lock, Wait time: 67.41
● : Server CPU: 116.14

Database Load – database details

DBPLUS
better performance

After clicking on a point on the chart, you can find information about:

- Queries run in a given period of time



Statements filter: Top 20 statements by Elapsed time

UnGroup literals

SNAPSHOT OF SQL STATEMENTS EXECUTED WITHIN 15 MINUTES AT 2018-12-11 11:40:00

Search statistic by sql text, hash value or plan hash in below snapshot table

Query text	Hash Value	Sql Id	Plan Hash	Elapsed Time	Cpu Time	Time per 1 exec	Sorts	Fetches	Executions	Parse Calls	Disk reads	Buffer gets	Rows processed	Module	Number of concurrent users	Db Load	Cpu Load
				[Seconds]	[Seconds]	[Seconds]	[Rows]	[Rows]			[Blocks]	[Blocks]	[Rows]			[%]	[%]
SELECT /*+ */ ROWID,WSK_DO	19419297	549x1r40khn5	461079357	2 388.27	882.67	0.4506	0	5 491	5 300	1 834	469 210	73 672 014	50 231	SAFO2000	305	2	3
SELECT ID, DECODE (DOK_ZA	821097180	4y825xssg1wc	1921616051	1 667.70	609.52	1.0077	1 655	14 971	1 655	1 655	176 069	81 551 265	55 763	w3wp.exe	3	2	2
SELECT DECODE (K.FIR_KOD,	2435395181	d20tc5f8kkbm	1818372388	1 649.02	113.01	1.0211	0	1 616	1 615	1 615	83 896	11 197 545	1 616	w3wp.exe	0	2	0
-- WIDOK TOWARY (TJ, ML) -- I	1159534948	7rkh8bd2ju6b	2354673971	1 493.51	636.98	2.1708	0	3 147	688	336	317	27 633 663	100 281	ifrun60.exe	4	2	2
INSERT INTO ZES_000 (W50,W	3869860768	aa2dbammaks	4066142822	1 307.47	467.32	261.4942	171	0	5	4	416 395	156 210 62	12 002	SAFO200	0	1	1
-- KATALOG - POJAZDY (TJ, PD	1089110411	67vjt9j0fp0cb	1408853141	1 265.87	493.22	0.8846	57 318	16 039	1 431	1 382	36 688	73 071 794	68 977	ifrun60.exe	1	1	2
-- WIDOK TOWARY (TJ, ML) -- I	77461126	5v8g9x429vxr	504310476	919.95	390.39	2.4273	0	656	379	231	1 865	18 298 633	12 192	ifrun60.exe	1	1	1
select ofe_d.id from ofe_d where	681829070	11nwpknna7s	3746074611	913.76	356.73	0.0141	0	3 062 690	64 731	64 729	242	140 202 29	2 997 441	w3wp.exe	4	1	1

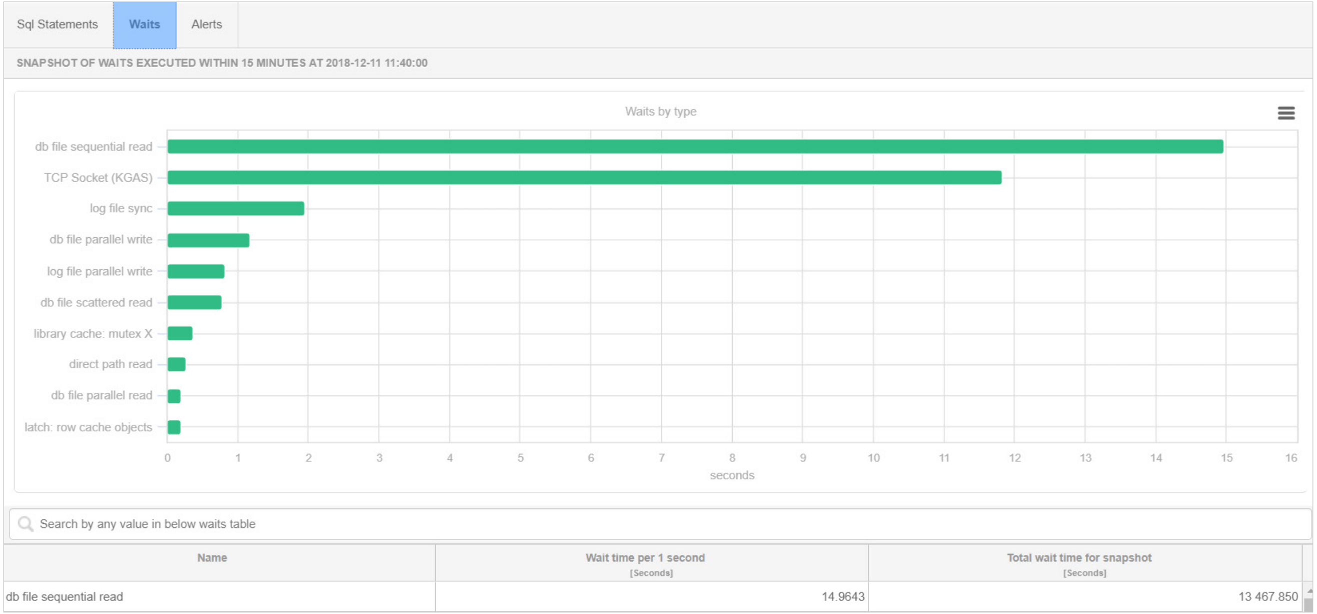
STATEMENT TEXT FOR HASH VALUE: 19419297

```
SELECT /*+ */
ROWID,WSK_DO,NR,ROK,DAT_W,DAT_R,DAT_PL,KH_KOD,WAL_KOD,WAR,STA_Z,DAT_S,KH_KOD_2,NR_ZEW,ZAM_TOW,SPO_PL_KOD,ILE_DTP,ROD_Z,ROD_TR_KOD,ROD_TR_FIR_KOD,TYP_D_ID,ROD_D_KOD,FIR_KOD_1
,MAG_KOD,MAG_FIR_KOD,KOM,CYK_T,WSK_UE,DAT_P,DAT_D,FIR_KOD,OSO_KOD,OSO_KOD_A,ZAM_L,DAT_WK,STA_R,ILE_MAX,DOD_N1,TRS_KOD,DOK_TR_ID,WSK_PAK,KIE_SP_KOD,WSK_WP,OSO_KOD_MOD,DOK_ZA_
ID_3,KUR_W,WAR_DOS,TYP_Z,CYK_M,WSK_GRZ,KOS_KOD,FIR_KOD_ZAT,FIR_KOD_DOC,WSK_TF,WSK_INT,OFE_D_ID,LIM_ID,WSK_LJ,ID,SLO_KOM_KOD,DOD_C14,MIE_ID,EXT_NR_I,SYS_NAME_I,TYP2_Z,SPO_REZ
,FIR_KOD_REA FROM dok_za WHERE (EXISTS (SELECT KOD FROM FIR WHERE KOD=DOK_ZA.FIR_KOD CONNECT BY PRIOR KOD=FIR_KOD START WITH KOD=:"SYS_B_0") OR
```

Database Load – database details

After clicking on a point on the chart, you can find information about:

- Wait level



SNAPSHOT OF WAITS EXECUTED WITHIN 15 MINUTES AT 2018-12-11 11:40:00

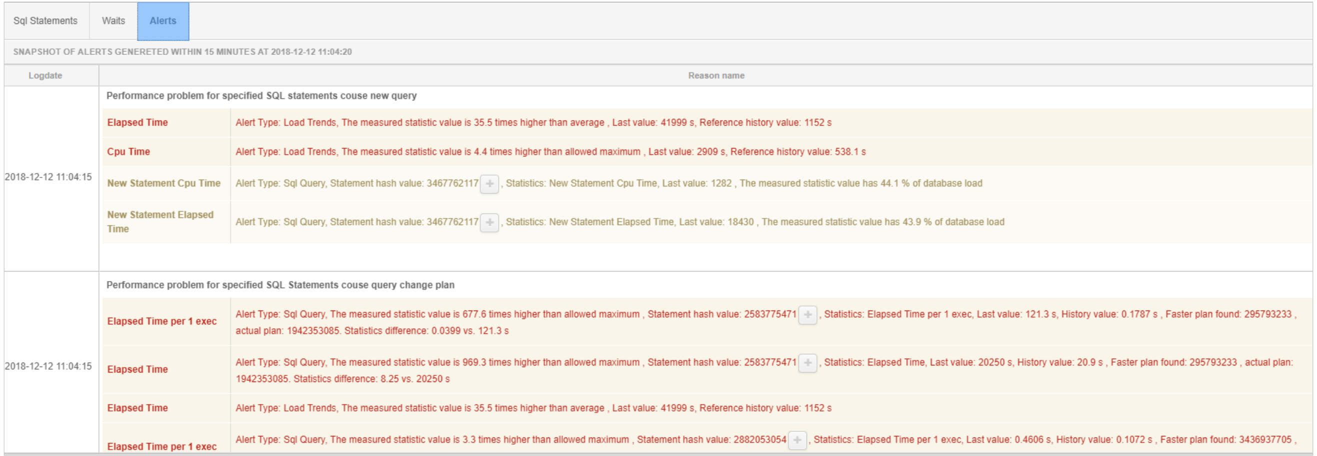
Name	Wait time per 1 second [Seconds]	Total wait time for snapshot [Seconds]
db file sequential read	14.9643	13 467.850

Database Load – database details

DBPLUS
better performance

After clicking on a point on the chart, you can find information about:

- Alerts (if any)



SNAPSHOT OF ALERTS GENERETED WITHIN 15 MINUTES AT 2018-12-12 11:04:20

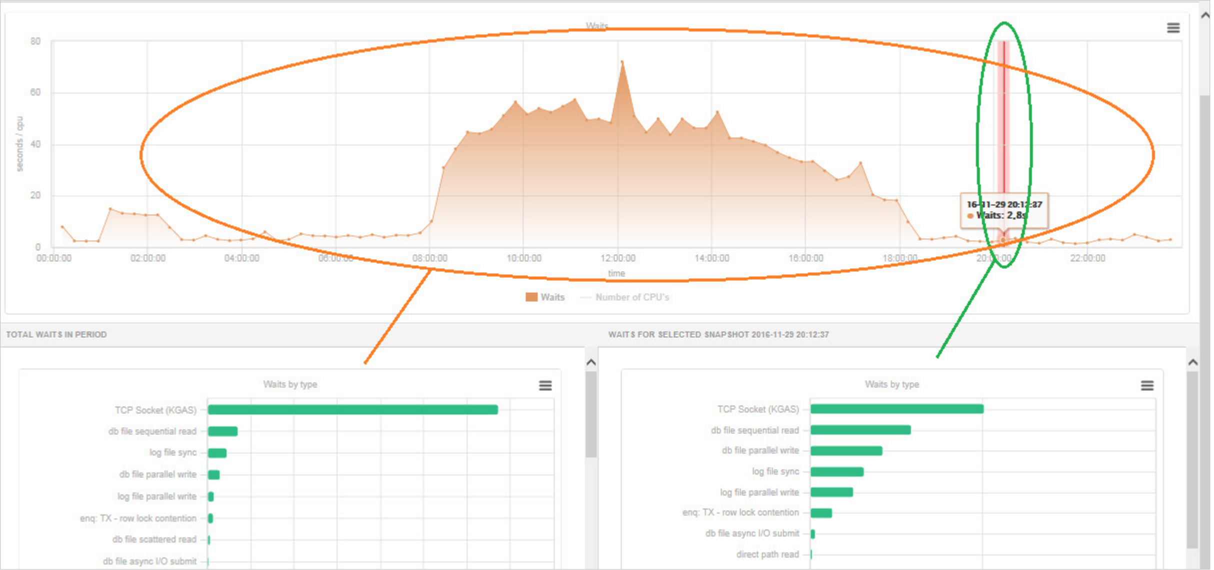
Logdate	Reason name	
2018-12-12 11:04:15	Performance problem for specified SQL statements couse new query	
	Elapsed Time	Alert Type: Load Trends, The measured statistic value is 35.5 times higher than average , Last value: 41999 s, Reference history value: 1152 s
	Cpu Time	Alert Type: Load Trends, The measured statistic value is 4.4 times higher than allowed maximum , Last value: 2909 s, Reference history value: 538.1 s
	New Statement Cpu Time	Alert Type: Sql Query, Statement hash value: 3467762117 + , Statistics: New Statement Cpu Time, Last value: 1282 , The measured statistic value has 44.1 % of database load
	New Statement Elapsed Time	Alert Type: Sql Query, Statement hash value: 3467762117 + , Statistics: New Statement Elapsed Time, Last value: 18430 , The measured statistic value has 43.9 % of database load
2018-12-12 11:04:15	Performance problem for specified SQL Statements couse query change plan	
	Elapsed Time per 1 exec	Alert Type: Sql Query, The measured statistic value is 677.6 times higher than allowed maximum , Statement hash value: 2583775471 + , Statistics: Elapsed Time per 1 exec, Last value: 121.3 s, History value: 0.1787 s , Faster plan found: 295793233 , actual plan: 1942353085. Statistics difference: 0.0399 vs. 121.3 s
	Elapsed Time	Alert Type: Sql Query, The measured statistic value is 969.3 times higher than allowed maximum , Statement hash value: 2583775471 + , Statistics: Elapsed Time, Last value: 20250 s, History value: 20.9 s , Faster plan found: 295793233 , actual plan: 1942353085. Statistics difference: 8.25 vs. 20250 s
	Elapsed Time	Alert Type: Load Trends, The measured statistic value is 35.5 times higher than average , Last value: 41999 s, Reference history value: 1152 s
	Elapsed Time per 1 exec	Alert Type: Sql Query, The measured statistic value is 3.3 times higher than allowed maximum , Statement hash value: 2882053054 + , Statistics: Elapsed Time per 1 exec, Last value: 0.4606 s, History value: 0.1072 s , Faster plan found: 3436937705 ,

Waits

The graph shows the wait time for all sessions in a database in a given time period.

The chart on the left shows the top waits for the day.

The graph on the right shows the top waits for the indicated point on the chart (snap).



TOTAL WAITS IN PERIOD

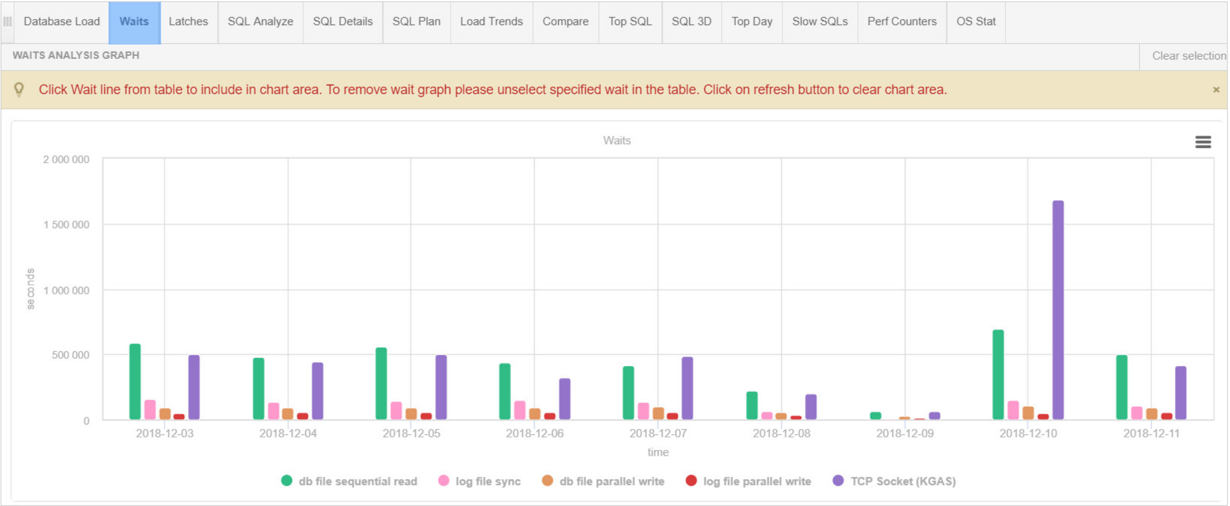
WAITS FOR SELECTED SNAPSHOT 2016-11-29 20:12:37

Waits Analyze

DBPLUS
better performance

As part of a detailed analysis, you can sort waits by:

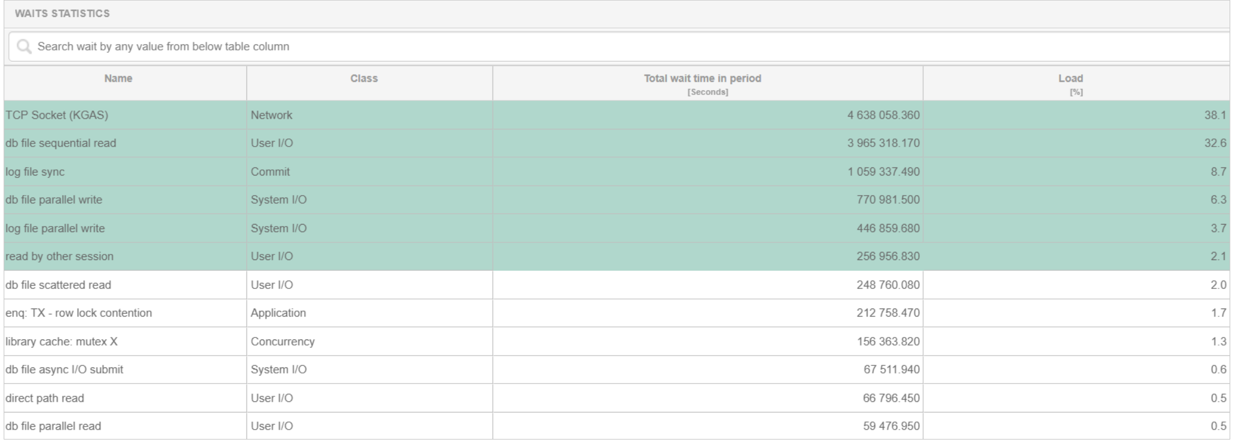
- Wait type
- Wait class
- Affecting performance



Click Wait line from table to include in chart area. To remove wait graph please unselect specified wait in the table. Click on refresh button to clear chart area.

Waits Analyze

All information on the chart is visible in the form of a table below it.



WAITS STATISTICS

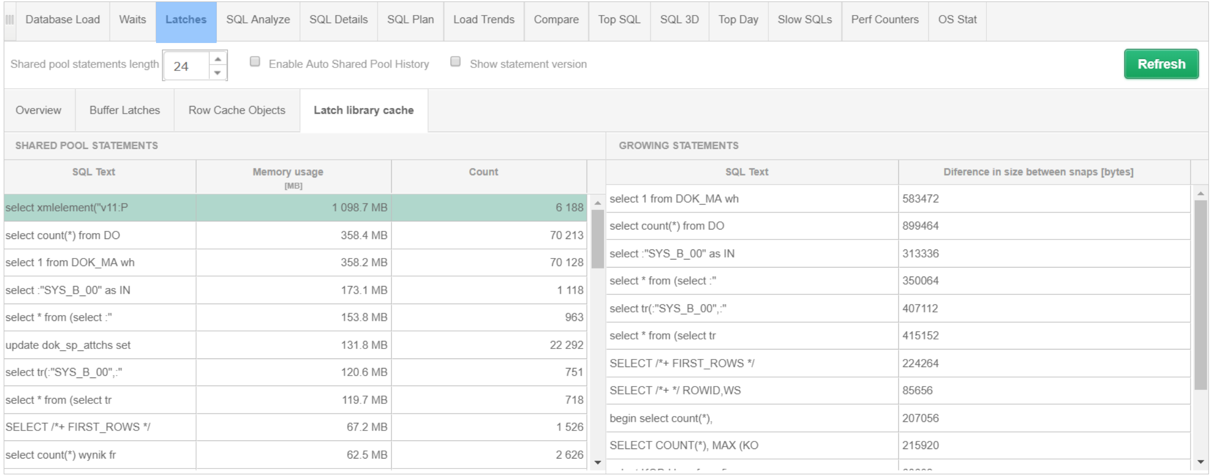
Search wait by any value from below table column

Name	Class	Total wait time in period [Seconds]	Load [%]
TCP Socket (KGAS)	Network	4 638 058.360	38.1
db file sequential read	User I/O	3 965 318.170	32.6
log file sync	Commit	1 059 337.490	8.7
db file parallel write	System I/O	770 981.500	6.3
log file parallel write	System I/O	446 859.680	3.7
read by other session	User I/O	256 956.830	2.1
db file scattered read	User I/O	248 760.080	2.0
enq: TX - row lock contention	Application	212 758.470	1.7
library cache: mutex X	Concurrency	156 363.820	1.3
db file async I/O submit	System I/O	67 511.940	0.6
direct path read	User I/O	66 796.450	0.5
db file parallel read	User I/O	59 476.950	0.5

Latches

It is possible to present information about the "latches" appearing in a database:

- **Buffer Latches** - information on memory utilisation queries is available.
- **Row Cache Objects** - presents "latches" divided into SHARED_POOL operations.
- **Latch Library Cache** - how much space the query takes up in the SHARED_POOL memory buffer.



Shared pool statements length 24 □ Enable Auto Shared Pool History □ Show statement version

SHARED POOL STATEMENTS

SQL Text	Memory usage [MB]	Count
select xmlelement("v11:P	1 098.7 MB	6 188
select count(*) from DO	358.4 MB	70 213
select 1 from DOK_MA wh	358.2 MB	70 128
select :"SYS_B_00" as IN	173.1 MB	1 118
select * from (select :"	153.8 MB	963
update dok_sp_attchs set	131.8 MB	22 292
select tr(:"SYS_B_00",:"	120.6 MB	751
select * from (select tr	119.7 MB	718
SELECT /*+ FIRST_ROWS */	67.2 MB	1 526
select count(*) wynik fr	62.5 MB	2 626

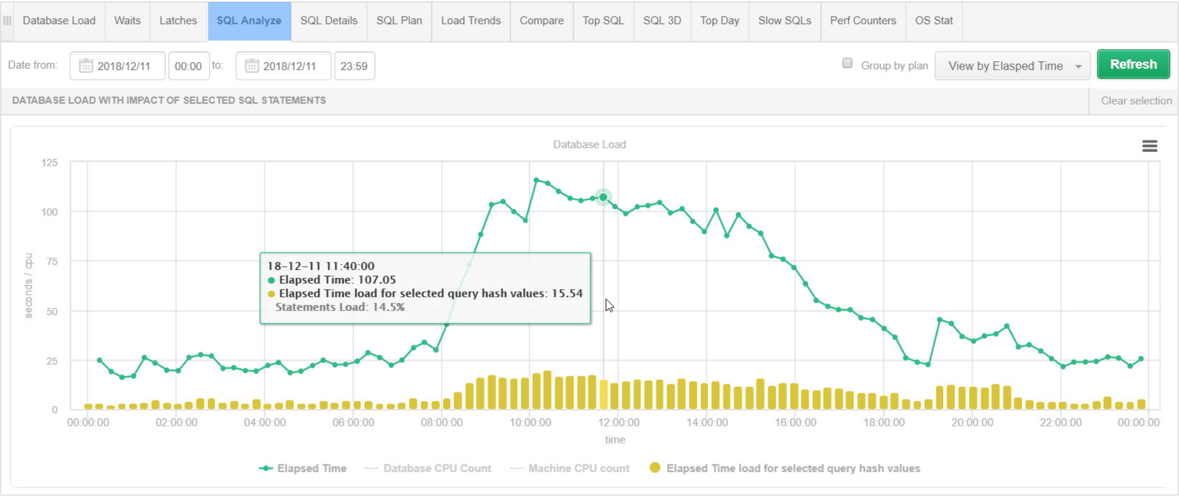
GROWING STATEMENTS

SQL Text	Diference in size between snaps [bytes]
select 1 from DOK_MA wh	583472
select count(*) from DO	899464
select :"SYS_B_00" as IN	313336
select * from (select :"	350064
select tr(:"SYS_B_00",:"	407112
select * from (select tr	415152
SELECT /*+ FIRST_ROWS */	224264
SELECT /*+ */ ROWID,WS	85656
begin select count(*),	207056
SELECT COUNT(*), MAX (KO	215920

SQL Analyze

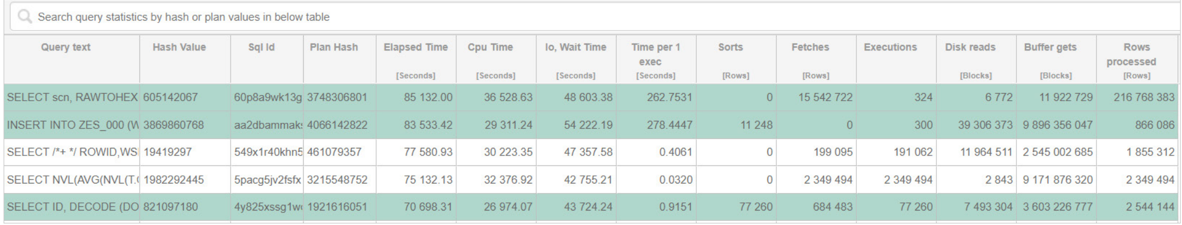
DBPLUS
better performance

Shows the Elapsed Time or CPU Time utilisation graph for a given period of time.



18-12-11 11:40:00
● Elapsed Time: 107.05
● Elapsed Time load for selected query hash values: 15.54
Statements Load: 14.5%

After selecting the queries under the graph, you get information about their share of the overall system utilisation for each parameter.



Search query statistics by hash or plan values in below table

Query text	Hash Value	Sql Id	Plan Hash	Elapsed Time	Cpu Time	Io, Wait Time	Time per 1 exec	Sorts	Fetches	Executions	Disk reads	Buffer gets	Rows processed
				[Seconds]	[Seconds]	[Seconds]	[Seconds]	[Rows]	[Rows]		[Blocks]	[Blocks]	[Rows]
SELECT scn, RAWTOHEX	605142067	60p8a9wk13g	3748306801	85 132.00	36 528.63	48 603.38	262.7531	0	15 542 722	324	6 772	11 922 729	216 768 383
INSERT INTO ZES_000 (W	3869860768	aa2dbammak	4066142822	83 533.42	29 311.24	54 222.19	278.4447	11 248	0	300	39 306 373	9 896 356 047	866 086
SELECT /*+ */ ROWID,WS	19419297	549x1r40khn5	461079357	77 580.93	30 223.35	47 357.58	0.4061	0	199 095	191 062	11 964 511	2 545 002 685	1 855 312
SELECT NVL(AVG(NVL(T.(	1982292445	5pacg5jv2fsfx	3215548752	75 132.13	32 376.92	42 755.21	0.0320	0	2 349 494	2 349 494	2 843	9 171 876 320	2 349 494
SELECT ID, DECODE (DO	821097180	4y825xssg1w	1921616051	70 698.31	26 974.07	43 724.24	0.9151	77 260	684 483	77 260	7 493 304	3 603 226 777	2 544 144

SQL Details

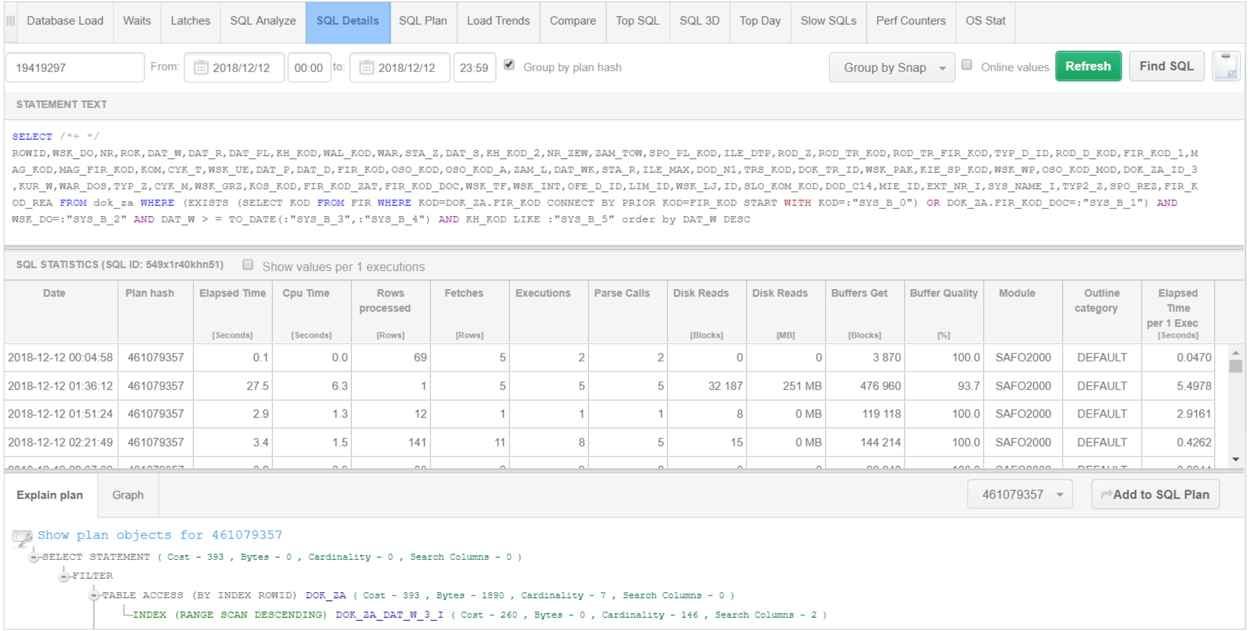
DBPLUS
better performance

Contains detailed information about queries.

Data are presented for the indicated period of time with the option to grouping by:

- *Snap (15 minutes)*
- *Hour*
- *Day*
- *Month*

The possibility to display Online data - downloading on a regular basis from the V$SQL system view.



19419297 From: 2018/12/12 00:00 to: 2018/12/12 23:59 ☑ Group by plan hash

Group by Snap □ Online values

STATEMENT TEXT

```
SELECT /*+ */
ROWID,WSK_DO,NR,ROK,DAT_W,DAT_R,DAT_PL,KH_KOD,WAL_KOD,WAR,STA_Z,DAT_S,KH_KOD_2,NR_ZEW,ZAM_TOW,SPO_PL_KOD,ILE_DTP,ROD_Z,ROD_TR_KOD,ROD_TR_FIR_KOD,TYP_D_ID,ROD_D_KOD,FIR_KOD_1,M
AG_KOD,MAG_FIR_KOD,KOM,CYK_T,WSK_UE,DAT_P,DAT_D,FIR_KOD,OSO_KOD,OSO_KOD_A,ZAM_L,DAT_WK,STA_R,ILE_MAX,DOD_N1,TRS_KOD,DOK_TR_ID,WSK_PAK,KIE_SP_KOD,WSK_WP,OSO_KOD_MOD,DOK_ZA_ID_3
,KUR_W,WAR_DOS,TYP_Z,CYK_M,WSK_GRZ,KOS_KOD,FIR_KOD_ZAT,FIR_KOD_DOC,WSK_TF,WSK_INT,OFE_D_ID,LIM_ID,WSK_LJ,ID,SLO_KOM_KOD,DOD_C14,MIE_ID,EXT_NR_I,SYS_NAME_I,TYP2_Z,SPO_REZ,FIR_K
OD_REA FROM dok_za WHERE (EXISTS (SELECT KOD FROM FIR WHERE KOD=DOK_ZA.FIR_KOD CONNECT BY PRIOR KOD=FIR_KOD START WITH KOD=:"SYS_B_0") OR DOK_ZA.FIR_KOD_DOC=:"SYS_B_1") AND
WSK_DO=:"SYS_B_2" AND DAT_W > = TO_DATE(:"SYS_B_3",:"SYS_B_4") AND KH_KOD LIKE :"SYS_B_5" order by DAT_W DESC
```

SQL STATISTICS (SQL ID: 549x1r40khn51) □ Show values per 1 executions

Date	Plan hash	Elapsed Time	Cpu Time	Rows processed	Fetches	Executions	Parse Calls	Disk Reads	Disk Reads	Buffers Get	Buffer Quality	Module	Outline category	Elapsed Time per 1 Exec
		[Seconds]	[Seconds]	[Rows]	[Rows]			[Blocks]	[MB]	[Blocks]	[%]			[Seconds]
2018-12-12 00:04:58	461079357	0.1	0.0	69	5	2	2	0	0	3 870	100.0	SAFO2000	DEFAULT	0.0470
2018-12-12 01:36:12	461079357	27.5	6.3	1	5	5	5	32 187	251 MB	476 960	93.7	SAFO2000	DEFAULT	5.4978
2018-12-12 01:51:24	461079357	2.9	1.3	12	1	1	1	8	0 MB	119 118	100.0	SAFO2000	DEFAULT	2.9161
2018-12-12 02:21:49	461079357	3.4	1.5	141	11	8	5	15	0 MB	144 214	100.0	SAFO2000	DEFAULT	0.4262

```
Show plan objects for 461079357
SELECT STATEMENT ( Cost - 393 , Bytes - 0 , Cardinality - 0 , Search Columns - 0 )
  FILTER
    TABLE ACCESS (BY INDEX ROWID) DOK_ZA ( Cost - 393 , Bytes - 1890 , Cardinality - 7 , Search Columns - 0 )
      INDEX (RANGE SCAN DESCENDING) DOK_ZA_DAT_W_3_I ( Cost - 260 , Bytes - 0 , Cardinality - 146 , Search Columns - 2 )
```

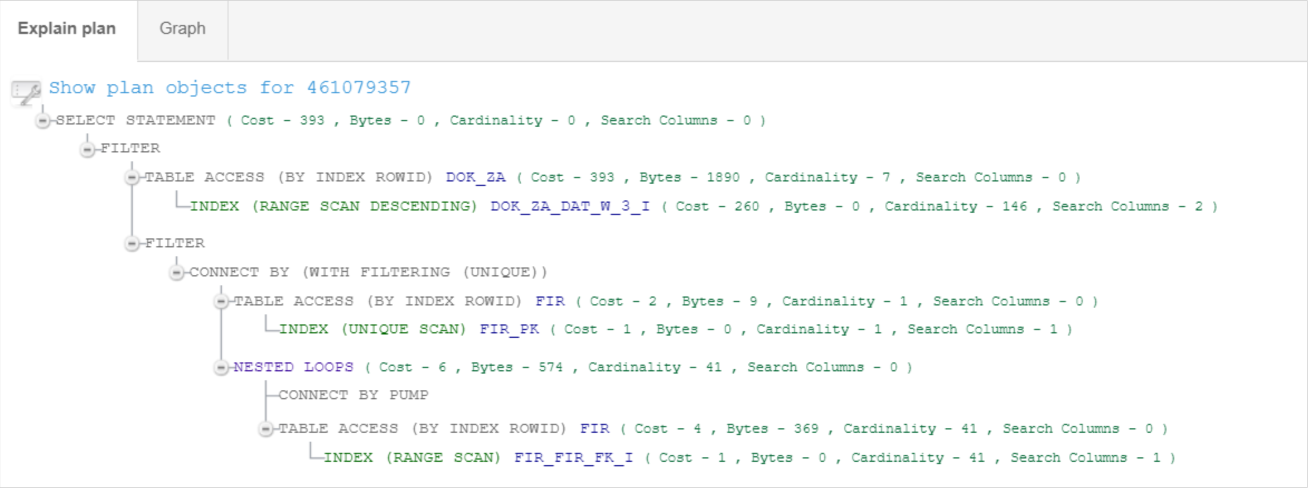
SQL Details

DBPLUS
better performance

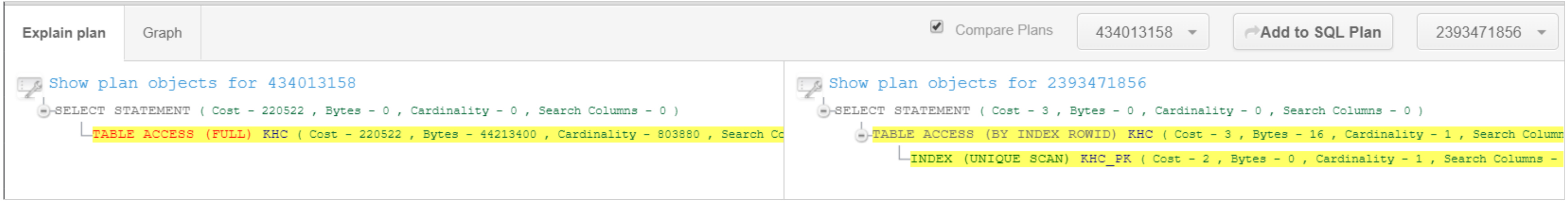
Easy access to the query plan (Explain plan).

The possibility to view sample parameters that the query is performed with.

The possibility to compare the plans used by a given query over a period of time.



```
Show plan objects for 461079357
 SELECT STATEMENT ( Cost - 393 , Bytes - 0 , Cardinality - 0 , Search Columns - 0 )
   FILTER
     TABLE ACCESS (BY INDEX ROWID) DOK_ZA ( Cost - 393 , Bytes - 1890 , Cardinality - 7 , Search Columns - 0 )
       INDEX (RANGE SCAN DESCENDING) DOK_ZA_DAT_W_3_I ( Cost - 260 , Bytes - 0 , Cardinality - 146 , Search Columns - 2 )
     FILTER
       CONNECT BY (WITH FILTERING (UNIQUE))
         TABLE ACCESS (BY INDEX ROWID) FIR ( Cost - 2 , Bytes - 9 , Cardinality - 1 , Search Columns - 0 )
           INDEX (UNIQUE SCAN) FIR_PK ( Cost - 1 , Bytes - 0 , Cardinality - 1 , Search Columns - 1 )
         NESTED LOOPS ( Cost - 6 , Bytes - 574 , Cardinality - 41 , Search Columns - 0 )
           CONNECT BY PUMP
           TABLE ACCESS (BY INDEX ROWID) FIR ( Cost - 4 , Bytes - 369 , Cardinality - 41 , Search Columns - 0 )
             INDEX (RANGE SCAN) FIR_FIR_FK_I ( Cost - 1 , Bytes - 0 , Cardinality - 41 , Search Columns - 1 )
```



```
Show plan objects for 434013158
 SELECT STATEMENT ( Cost - 220522 , Bytes - 0 , Cardinality - 0 , Search Columns - 0 )
   TABLE ACCESS (FULL) KHC ( Cost - 220522 , Bytes - 44213400 , Cardinality - 803880 , Search Co
```

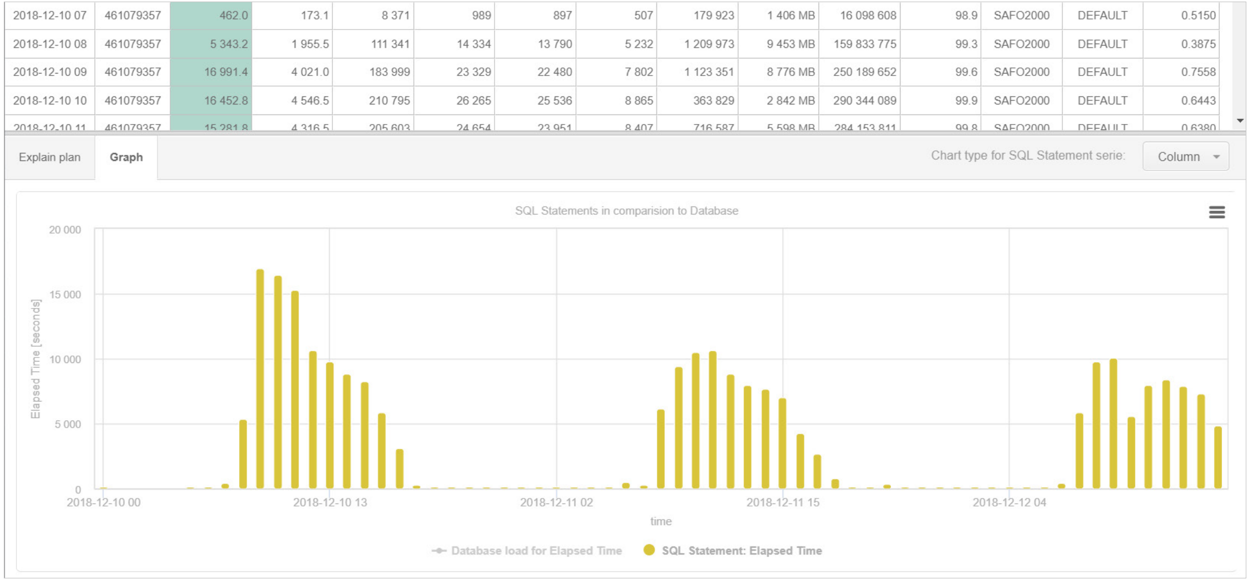
```
Show plan objects for 2393471856
 SELECT STATEMENT ( Cost - 3 , Bytes - 0 , Cardinality - 0 , Search Columns - 0 )
   TABLE ACCESS (BY INDEX ROWID) KHC ( Cost - 3 , Bytes - 16 , Cardinality - 1 , Search Column
     INDEX (UNIQUE SCAN) KHC_PK ( Cost - 2 , Bytes - 0 , Cardinality - 1 , Search Columns -
```

SQL Details

DBPLUS
better performance

The query statistics can be viewed in a graph by clicking on a given column in the table.

Database load for...
- the possibility to estimate the impact of a given query in relation to the statistics for the entire database.



2018-12-10 07	461079357	462.0	173.1	8 371	989	897	507	179 923	1 406 MB	16 098 608	98.9	SAFO2000	DEFAULT	0.5150
2018-12-10 08	461079357	5 343.2	1 955.5	111 341	14 334	13 790	5 232	1 209 973	9 453 MB	159 833 775	99.3	SAFO2000	DEFAULT	0.3875
2018-12-10 09	461079357	16 991.4	4 021.0	183 999	23 329	22 480	7 802	1 123 351	8 776 MB	250 189 652	99.6	SAFO2000	DEFAULT	0.7558
2018-12-10 10	461079357	16 452.8	4 546.5	210 795	26 265	25 536	8 865	363 829	2 842 MB	290 344 089	99.9	SAFO2000	DEFAULT	0.6443
2018-12-10 11	461079357	15 281.8	4 316.5	205 603	24 654	23 951	8 407	716 587	5 598 MB	284 153 811	99.8	SAFO2000	DEFAULT	0.6380

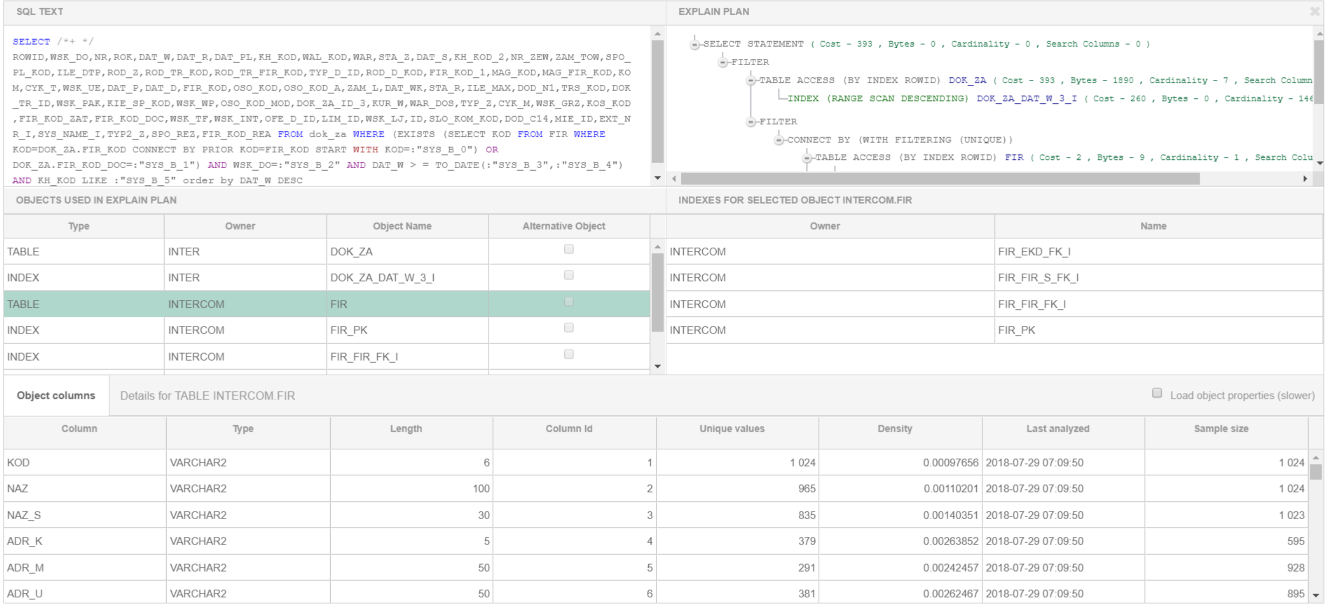
Chart type for SQL Statement serie: Column

Show Plan Objects

DBPLUS
better performance

Includes:

- Query content
- Query plan
- Query objects
- Indices
- Details of the object



SQL TEXT

```
SELECT /*+ */
ROWID,WSK_DO,NR,ROK,DAT_W,DAT_R,DAT_PL,KH_KOD,WAL_KOD,WAR,STA_Z,DAT_S,KH_KOD_2,NR_ZEW,ZAM_TOW,SPO_
PL_KOD,ILE_DTP,ROD_Z,ROD_TR_KOD,ROD_TR_FIR_KOD,TYP_D_ID,ROD_D_KOD,FIR_KOD_1,MAG_KOD,MAG_FIR_KOD,KO
M,CYK_T,WSK_UE,DAT_P,DAT_D,FIR_KOD,OSO_KOD,OSO_KOD_A,ZAM_L,DAT_WK,STA_R,ILE_MAX,DOD_N1,TRS_KOD,DOK
_TR_ID,WSK_PAK,KIE_SP_KOD,WSK_WP,OSO_KOD_MOD,DOK_ZA_ID_3,KUR_W,WAR_DOS,TYP_Z,CYK_M,WSK_GRZ,KOS_KOD
,FIR_KOD_ZAT,FIR_KOD_DOC,WSK_TF,WSK_INT,OFE_D_ID,LIM_ID,WSK_LJ,ID,SLO_KOM_KOD,DOD_C14,MIE_ID,EXT_N
R_I,SYS_NAME_I,TYP2_Z,SPO_REZ,FIR_KOD_REA FROM dok_za WHERE (EXISTS (SELECT KOD FROM FIR WHERE
KOD=DOK_ZA.FIR_KOD CONNECT BY PRIOR KOD=FIR_KOD START WITH KOD=:"SYS_B_0") OR
DOK_ZA.FIR_KOD_DOC=:"SYS_B_1") AND WSK_DO=:"SYS_B_2" AND DAT_W > = TO_DATE(:"SYS_B_3",:"SYS_B_4")
AND KH_KOD LIKE :"SYS_B_5" order by DAT_W DESC
```

EXPLAIN PLAN

```
⊖SELECT STATEMENT ( Cost - 393 , Bytes - 0 , Cardinality - 0 , Search Columns - 0 )
  ⊖FILTER
    ⊖TABLE ACCESS (BY INDEX ROWID) DOK_ZA ( Cost - 393 , Bytes - 1890 , Cardinality - 7 , Search Column
      └INDEX (RANGE SCAN DESCENDING) DOK_ZA_DAT_W_3_I ( Cost - 260 , Bytes - 0 , Cardinality - 146
    ⊖FILTER
      ⊖CONNECT BY (WITH FILTERING (UNIQUE))
        ⊖TABLE ACCESS (BY INDEX ROWID) FIR ( Cost - 2 , Bytes - 9 , Cardinality - 1 , Search Colu
```

OBJECTS USED IN EXPLAIN PLAN

Type	Owner	Object Name	Alternative Object
TABLE	INTER	DOK_ZA	□
INDEX	INTER	DOK_ZA_DAT_W_3_I	□
TABLE	INTERCOM	FIR	□
INDEX	INTERCOM	FIR_PK	□
INDEX	INTERCOM	FIR_FIR_FK_I	□

INDEXES FOR SELECTED OBJECT INTERCOM.FIR

Owner	Name
INTERCOM	FIR_EKD_FK_I
INTERCOM	FIR_FIR_S_FK_I
INTERCOM	FIR_FIR_FK_I
INTERCOM	FIR_PK

Object columns   Details for TABLE INTERCOM.FIR   □ Load object properties (slower)

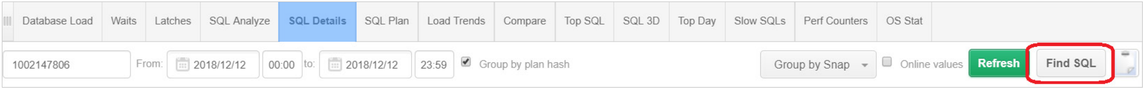
Column	Type	Length	Column Id	Unique values	Density	Last analyzed	Sample size
KOD	VARCHAR2	6	1	1 024	0.00097656	2018-07-29 07:09:50	1 024
NAZ	VARCHAR2	100	2	965	0.00110201	2018-07-29 07:09:50	1 024
NAZ_S	VARCHAR2	30	3	835	0.00140351	2018-07-29 07:09:50	1 023
ADR_K	VARCHAR2	5	4	379	0.00263852	2018-07-29 07:09:50	595
ADR_M	VARCHAR2	50	5	291	0.00242457	2018-07-29 07:09:50	928
ADR_U	VARCHAR2	50	6	381	0.00262467	2018-07-29 07:09:50	895

SQL Details

It is also possible to search queries using *Find SQL*.

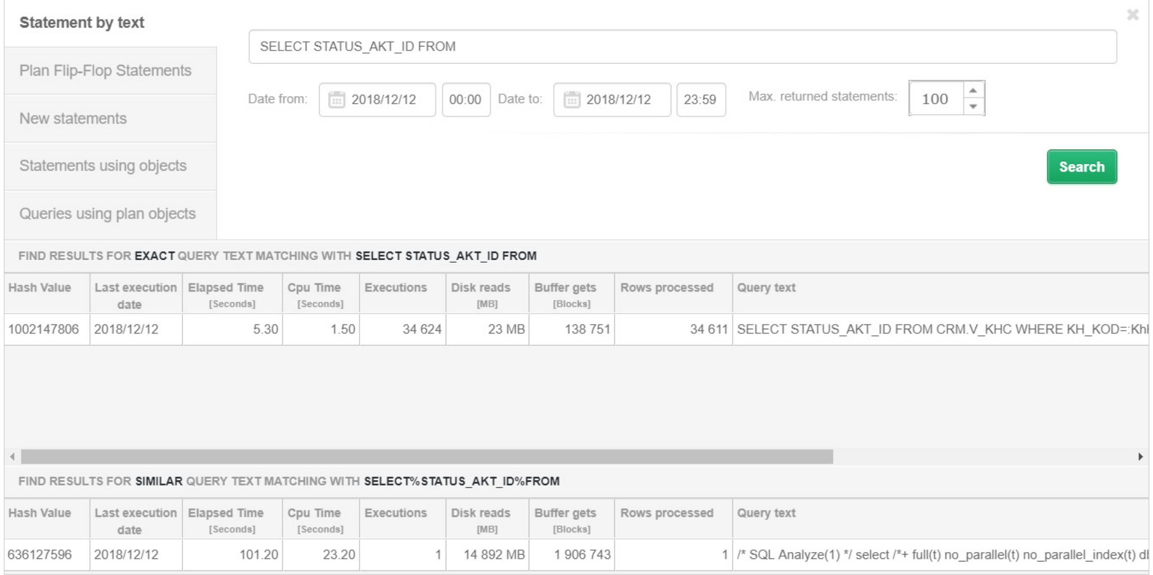
We can search via:

- Typing a text fragment
- Queries changing the plan
- New queries in a given period
- Queries using the object
- Query using a logical object (Outline, Baseline, Profile)



1002147806 From: 2018/12/12 00:00 to: 2018/12/12 23:59 ☑ Group by plan hash

Group by Snap □ Online values



SELECT STATUS_AKT_ID FROM

Date from: 2018/12/12 00:00 Date to: 2018/12/12 23:59 Max. returned statements: 100

FIND RESULTS FOR **EXACT** QUERY TEXT MATCHING WITH **SELECT STATUS_AKT_ID FROM**

Hash Value	Last execution date	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Disk reads [MB]	Buffer gets [Blocks]	Rows processed	Query text
1002147806	2018/12/12	5.30	1.50	34 624	23 MB	138 751	34 611	SELECT STATUS_AKT_ID FROM CRM.V_KHC WHERE KH_KOD=:Khl

FIND RESULTS FOR **SIMILAR** QUERY TEXT MATCHING WITH **SELECT%STATUS_AKT_ID%FROM**

Hash Value	Last execution date	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Disk reads [MB]	Buffer gets [Blocks]	Rows processed	Query text
636127596	2018/12/12	101.20	23.20	1	14 892 MB	1 906 743	1	/* SQL Analyze(1) */ select /*+ full(t) no_parallel(t) no_parallel_index(t) dl

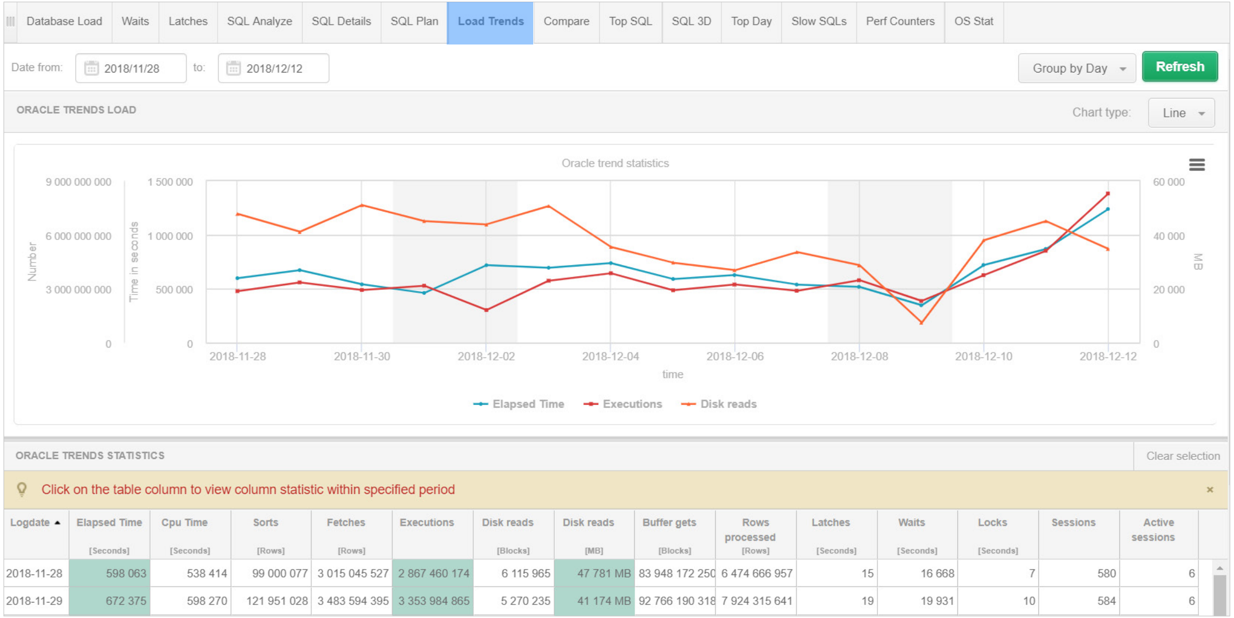
Load trends

DBPLUS
better performance

Allows you to get information about trends taking place in the database.

Data are presented for the indicated period of time with the possibility of grouping by:

- *Snap (15 minutes)*
- *Hour*
- *Day*
- *Month*



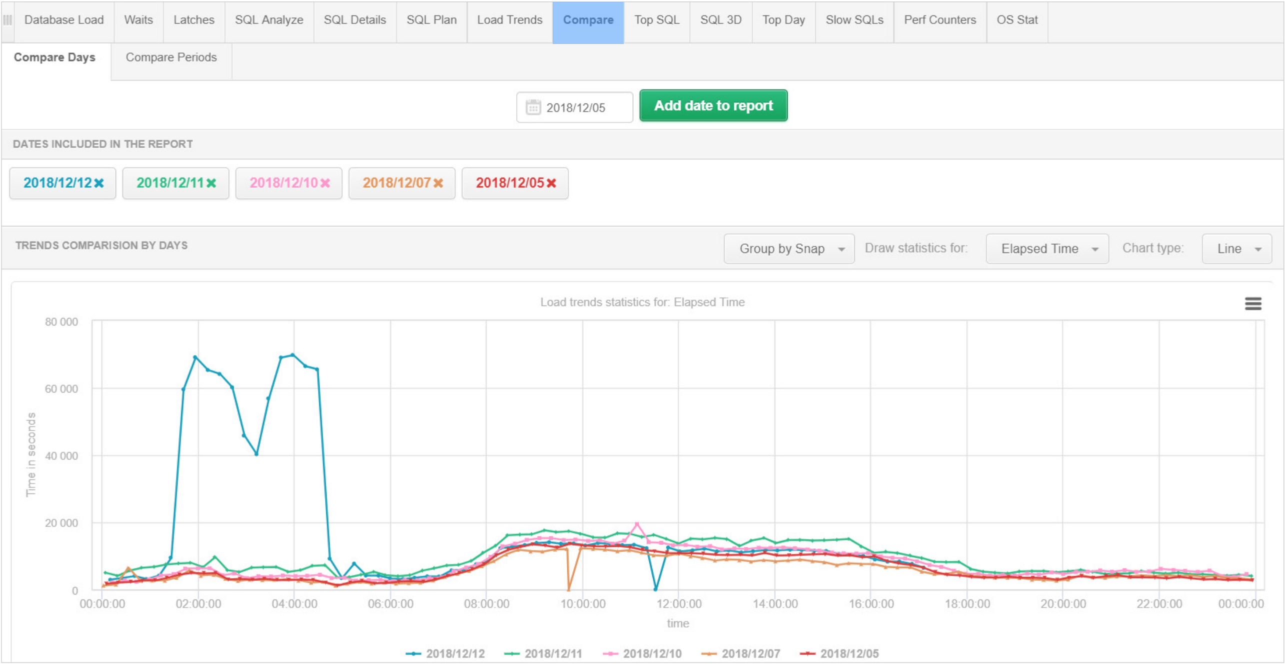
Click on the table column to view column statistic within specified period

Logdate ▲	Elapsed Time [Seconds]	Cpu Time [Seconds]	Sorts [Rows]	Fetches [Rows]	Executions	Disk reads [Blocks]	Disk reads [MB]	Buffer gets [Blocks]	Rows processed [Rows]	Latches [Seconds]	Waits [Seconds]	Locks [Seconds]	Sessions	Active sessions
2018-11-28	598 063	538 414	99 000 077	3 015 045 527	2 867 460 174	6 115 965	47 781 MB	83 948 172 250	6 474 666 957	15	16 668	7	580	6
2018-11-29	672 375	598 270	121 951 028	3 483 594 395	3 353 984 865	5 270 235	41 174 MB	92 766 190 318	7 924 315 641	19	19 931	10	584	6

Compare

Allows you to
compare statistics.

We can compare
a specific **day** as well
as a **period of time.**



2018/12/05

DATES INCLUDED IN THE REPORT

2018/12/12 ✖ 2018/12/11 ✖ 2018/12/10 ✖ 2018/12/07 ✖ 2018/12/05 ✖

TRENDS COMPARISION BY DAYS

Group by Snap | Draw statistics for: Elapsed Time | Chart type: Line

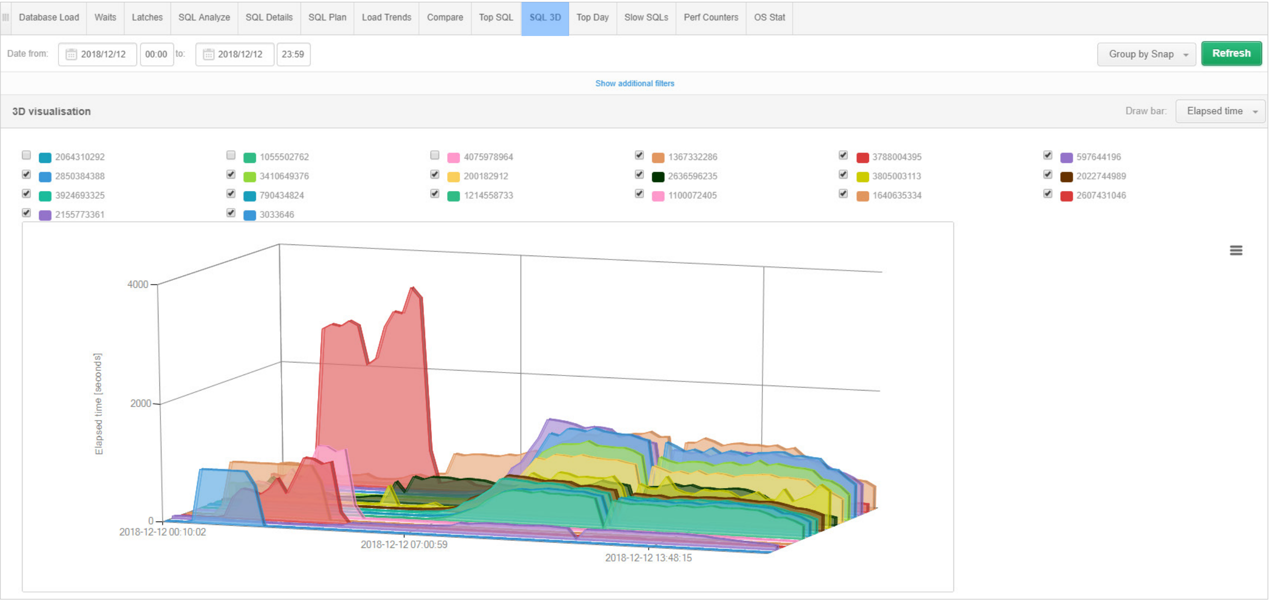
Top SQL/SQL 3D

DBPLUS
better performance

Presents information about the queries that have the largest share in a given statistic.

We can choose:

- Elapsed Time
- CPU Time
- Sorts
- Fetches
- Disk reads [block]
- Disk reads [MB]
- Rows processed
- Buffer Gets
- Execution



Date from: 2018/12/12 00:00 to: 2018/12/12 23:59

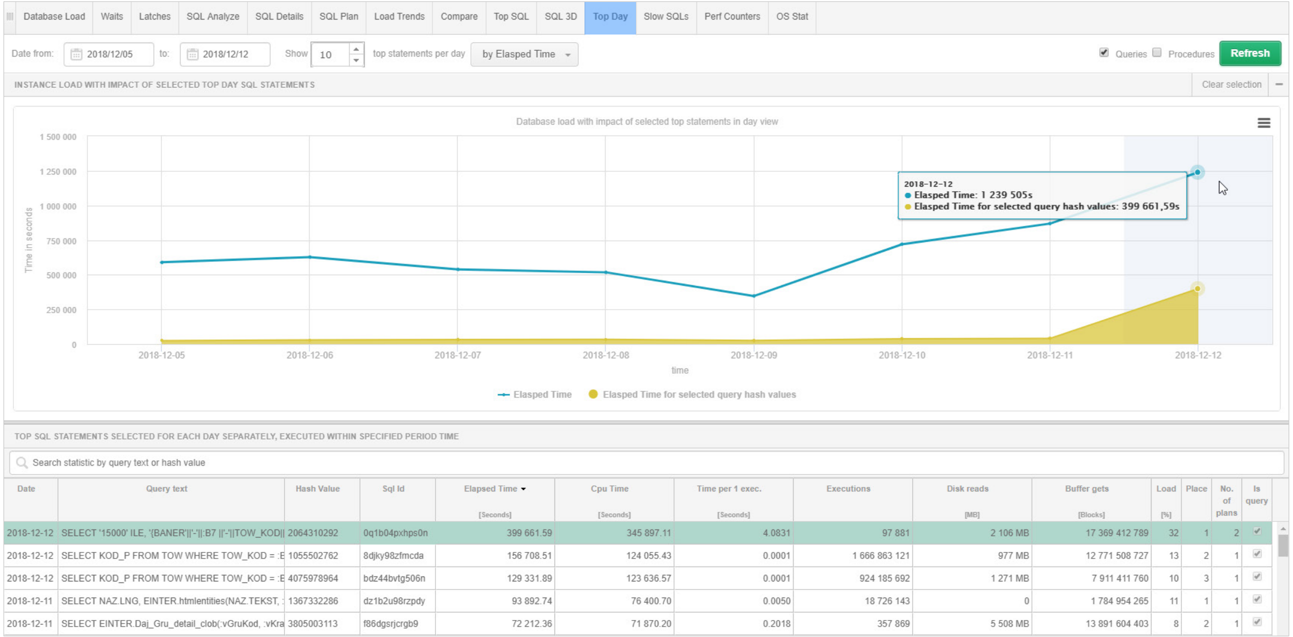
Group by Snap

3D visualisation

Draw bar: Elapsed time

Top SQL/SQL 3D

Allows you to display top queries for CPU Time or Elapsed Time and track changes in their behaviour.



Date from: 2018/12/05 to: 2018/12/12 Show 10 top statements per day by Elapsed Time ☑ Queries □ Procedures

INSTANCE LOAD WITH IMPACT OF SELECTED TOP DAY SQL STATEMENTS

TOP SQL STATEMENTS SELECTED FOR EACH DAY SEPARATELY, EXECUTED WITHIN SPECIFIED PERIOD TIME

Search statistic by query text or hash value

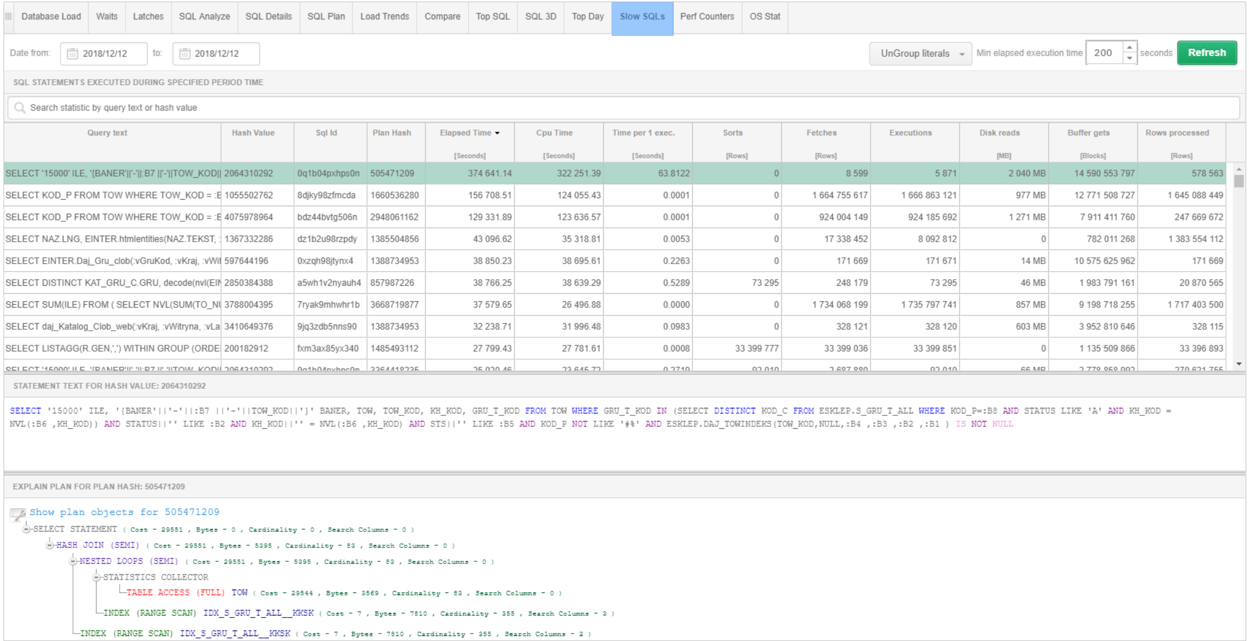
Date	Query text	Hash Value	Sql Id	Elapsed Time	Cpu Time	Time per 1 exec.	Executions	Disk reads	Buffer gets	Load	Place	No. of plans	Is query
				[Seconds]	[Seconds]	[Seconds]		[MB]	[Blocks]	[%]			
2018-12-12	SELECT '15000' ILE, '{BANER'||'-'||:B7 ||'-'||TOW_KOD||	2064310292	0q1b04pxhps0n	399 661.59	345 897.11	4.0831	97 881	2 106 MB	17 369 412 789	32	1	2	✓
2018-12-12	SELECT KOD_P FROM TOW WHERE TOW_KOD = :E	1055502762	8djky98zfmcda	156 708.51	124 055.43	0.0001	1 666 863 121	977 MB	12 771 508 727	13	2	1	✓
2018-12-12	SELECT KOD_P FROM TOW WHERE TOW_KOD = :E	4075978964	bdz44bvtg506n	129 331.89	123 636.57	0.0001	924 185 692	1 271 MB	7 911 411 760	10	3	1	✓
2018-12-11	SELECT NAZ.LNG, EINTER.htmlentities(NAZ.TEKST, :	1367332286	dz1b2u98rzpdy	93 892.74	76 400.70	0.0050	18 726 143	0	1 784 954 265	11	1	1	✓
2018-12-11	SELECT EINTER.Daj_Gru_detail_clob(:vGruKod, :vKra	3805003113	f86dgsrjcrgb9	72 212.36	71 870.20	0.2018	357 869	5 508 MB	13 891 604 403	8	2	1	✓

Slow SQL’s

DBPLUS
better performance

Presents queries that lasted for more than 200 seconds for a given period (default value).

It is possible to group queries by literals. To group the **literals** in place of the parameters, we insert the '#’ character, then we display the queries grouped by the query plan.



Date from: 2018/12/12 to: 2018/12/12

Min elapsed execution time 200 seconds

SQL STATEMENTS EXECUTED DURING SPECIFIED PERIOD TIME

Query text	Hash Value	Sql Id	Plan Hash	Elapsed Time	Cpu Time	Time per 1 exec.	Sorts	Fetches	Executions	Disk reads	Buffer gets	Rows processed
				[Seconds]	[Seconds]	[Seconds]	[Rows]	[Rows]		[MB]	[Blocks]	[Rows]
SELECT '15000' ILE, '{BANER'||'-'||:B7 ||'-'||TOW_KOD||	2064310292	0q1b04pxhps0n	505471209	374 641.14	322 251.39	63.8122	0	8 599	5 871	2 040 MB	14 590 553 797	578 563
SELECT KOD_P FROM TOW WHERE TOW_KOD = :E	1055502762	8djky98zfmcda	1660536280	156 708.51	124 055.43	0.0001	0	1 664 755 617	1 666 863 121	977 MB	12 771 508 727	1 645 088 449
SELECT KOD_P FROM TOW WHERE TOW_KOD = :E	4075978964	bdz44bvtg506n	2948061162	129 331.89	123 636.57	0.0001	0	924 004 149	924 185 692	1 271 MB	7 911 411 760	247 669 672
SELECT NAZ.LNG, EINTER.htmlentities(NAZ.TEKST, :	1367332286	dz1b2u98rzpdy	1385504856	43 096.62	35 318.81	0.0053	0	17 338 452	8 092 812	0	782 011 268	1 383 554 112
SELECT EINTER.Daj_Gru_clob(:vGruKod, :vKraj, :vWit	597644196	0xzqh98jtynx4	1388734953	38 850.23	38 695.61	0.2263	0	171 669	171 671	14 MB	10 575 625 962	171 669
SELECT DISTINCT KAT_GRU_C.GRU, decode(nvl(EIN	2850384388	a5wh1v2nyauh4	857987226	38 766.25	38 639.29	0.5289	73 295	248 179	73 295	46 MB	1 983 791 161	20 870 565
SELECT SUM(ILE) FROM (SELECT NVL(SUM(TO_NU	3788004395	7ryak9mhwhr1b	3668719877	37 579.65	26 496.88	0.0000	0	1 734 068 199	1 735 797 741	857 MB	9 198 718 255	1 717 403 500
SELECT daj_Katalog_Clob_web(:vKraj, :vWitryna, :vLa	3410649376	9jq3zdb5nns90	1388734953	32 238.71	31 996.48	0.0983	0	328 121	328 120	603 MB	3 952 810 646	328 115
SELECT LISTAGG(R.GEN,',') WITHIN GROUP (ORDE	200182912	fxm3ax85yx340	1485493112	27 799.43	27 781.61	0.0008	33 399 777	33 399 036	33 399 851	0	1 135 509 866	33 396 893
SELECT '15000' ILE, '{BANER'||'-'||:B7 ||'-'||TOW_KOD||	2064310292	0q1b04pxhps0n	3364418225	25 020.46	23 645.72	0.2719	92 010	2 687 880	92 010	66 MB	2 778 958 092	270 621 755

STATEMENT TEXT FOR HASH VALUE: 2064310292

```
SELECT '15000' ILE, '{BANER'||'-'||:B7 ||'-'||TOW_KOD||'}' BANER, TOW, TOW_KOD, KH_KOD, GRU_T_KOD FROM TOW WHERE GRU_T_KOD IN (SELECT DISTINCT KOD_C FROM ESKLEP.S_GRU_T_ALL WHERE KOD_P=:B8 AND STATUS LIKE 'A' AND KH_KOD =
NVL(:B6 ,KH_KOD)) AND STATUS||'' LIKE :B2 AND KH_KOD||'' = NVL(:B6 ,KH_KOD) AND STS||'' LIKE :B5 AND KOD_P NOT LIKE '#%' AND ESKLEP.DAJ_TOWINDEKS(TOW_KOD,NULL,:B4 ,:B3 ,:B2 ,:B1 ) IS NOT NULL
```

EXPLAIN PLAN FOR PLAN HASH: 505471209

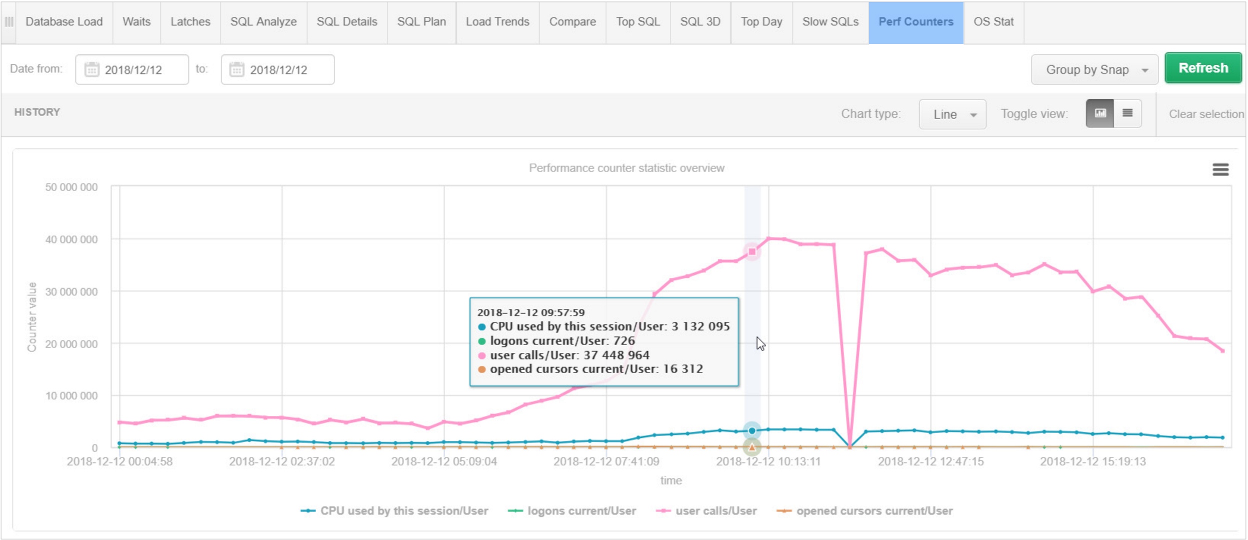
```
⊖-SELECT STATEMENT ( Cost - 29551 , Bytes - 0 , Cardinality - 0 , Search Columns - 0 )
   ⊖-HASH JOIN (SEMI) ( Cost - 29551 , Bytes - 5395 , Cardinality - 83 , Search Columns - 0 )
      ⊖-NESTED LOOPS (SEMI) ( Cost - 29551 , Bytes - 5395 , Cardinality - 83 , Search Columns - 0 )
      |  ⊖-STATISTICS COLLECTOR
      |  |  └TABLE ACCESS (FULL) TOW ( Cost - 29544 , Bytes - 3569 , Cardinality - 83 , Search Columns - 0 )
      |  └INDEX (RANGE SCAN) IDX_S_GRU_T_ALL__KKSK ( Cost - 7 , Bytes - 7810 , Cardinality - 355 , Search Columns - 3 )
      └INDEX (RANGE SCAN) IDX_S_GRU_T_ALL__KKSK ( Cost - 7 , Bytes - 7810 , Cardinality - 355 , Search Columns - 2 )
```

Perf Counters

All database statistics are displayed in the system view V$SYSTAT.

It is also possible to present information in tabular form.

Data is stored for a maximum of 30 days.



Date from: 2018/12/12 to: 2018/12/12

Group by Snap

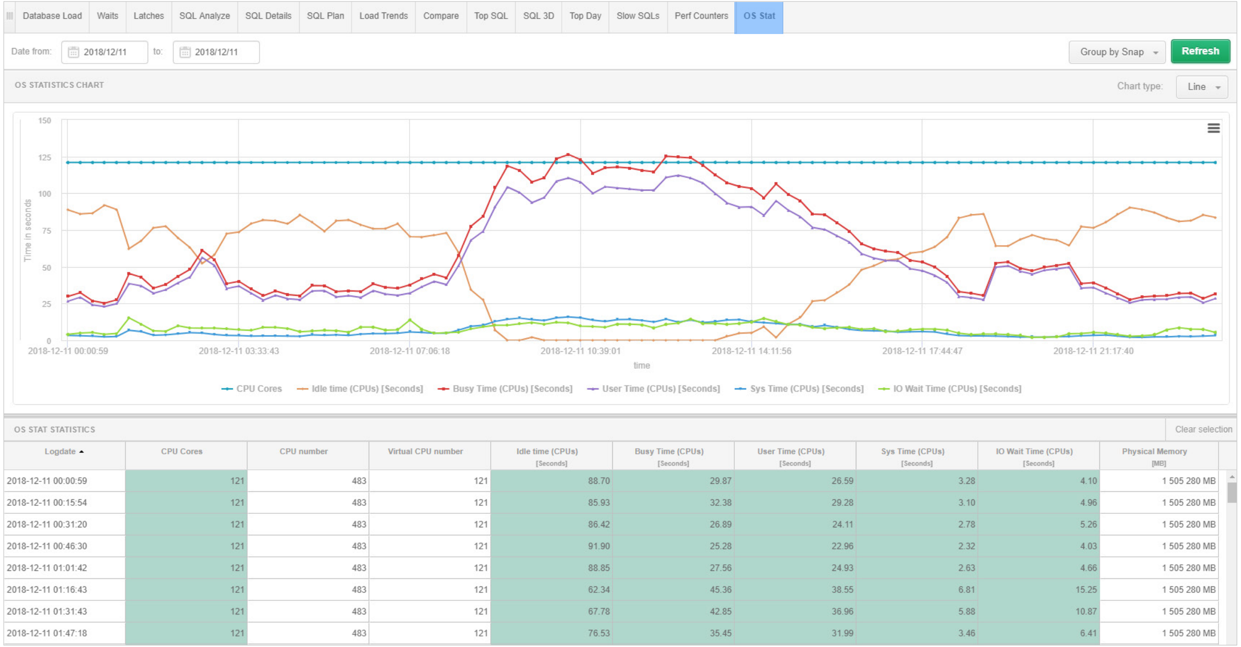
Chart type: Line

2018-12-12 09:57:59
- CPU used by this session/User: 3 132 095
- logons current/User: 726
- user calls/User: 37 448 964
- opened cursors current/User: 16 312

OS Stat

DBPLUS
better performance

Operational system statistics stored in the V$OSSTAT system view are presented.



Date from: 2018/12/11 to: 2018/12/11

Chart type: Line

Logdate ▲	CPU Cores	CPU number	Virtual CPU number	Idle time (CPUs) [Seconds]	Busy Time (CPUs) [Seconds]	User Time (CPUs) [Seconds]	Sys Time (CPUs) [Seconds]	IO Wait Time (CPUs) [Seconds]	Physical Memory [MB]
2018-12-11 00:00:59	121	483	121	88.70	29.87	26.59	3.28	4.10	1 505 280 MB
2018-12-11 00:15:54	121	483	121	85.93	32.38	29.28	3.10	4.96	1 505 280 MB
2018-12-11 00:31:20	121	483	121	86.42	26.89	24.11	2.78	5.26	1 505 280 MB
2018-12-11 00:46:30	121	483	121	91.90	25.28	22.96	2.32	4.03	1 505 280 MB
2018-12-11 01:01:42	121	483	121	88.85	27.56	24.93	2.63	4.66	1 505 280 MB
2018-12-11 01:16:43	121	483	121	62.34	45.36	38.55	6.81	15.25	1 505 280 MB
2018-12-11 01:31:43	121	483	121	67.78	42.85	36.96	5.88	10.87	1 505 280 MB
2018-12-11 01:47:18	121	483	121	76.53	35.45	31.99	3.46	6.41	1 505 280 MB

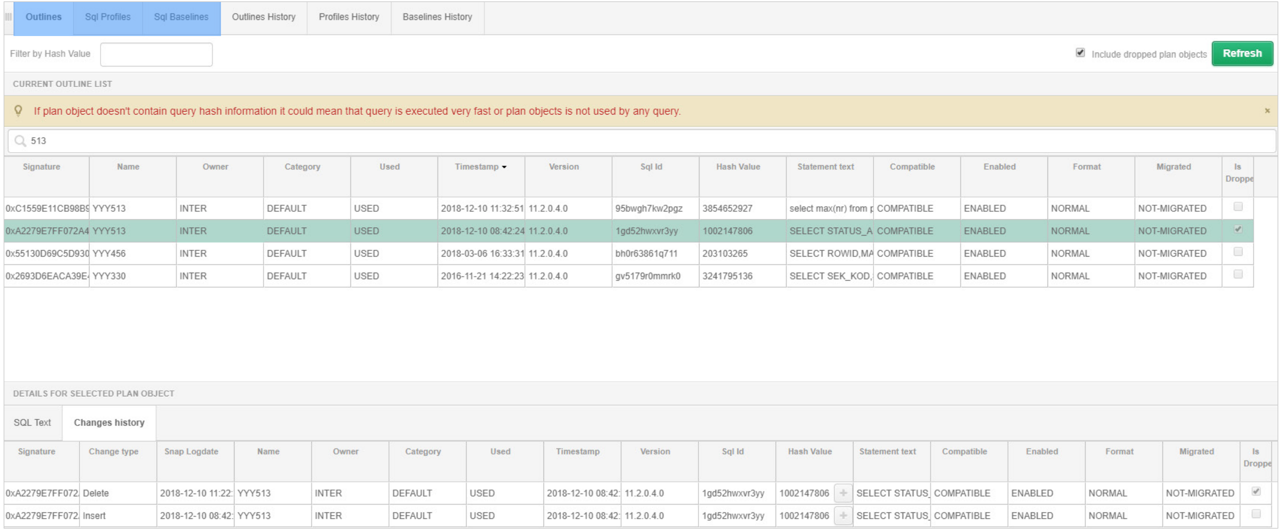
Plan Explorer

Information about objects is presented:

- Outlines
- Baselines
- Profiles

Current and historical information is available.

Information about what has changed and when, is stored.



CURRENT OUTLINE LIST

If plan object doesn't contain query hash information it could mean that query is executed very fast or plan objects is not used by any query.

513

Signature	Name	Owner	Category	Used	Timestamp	Version	Sql Id	Hash Value	Statement text	Compatible	Enabled	Format	Migrated	Is Droppe
0xC1559E11CB98B9	YYY513	INTER	DEFAULT	USED	2018-12-10 11:32:51	11.2.0.4.0	95bwgh7kw2pgz	3854652927	select max(nr) from p	COMPATIBLE	ENABLED	NORMAL	NOT-MIGRATED	□
0xA2279E7FF072A4	YYY513	INTER	DEFAULT	USED	2018-12-10 08:42:24	11.2.0.4.0	1gd52hwxvr3yy	1002147806	SELECT STATUS_A	COMPATIBLE	ENABLED	NORMAL	NOT-MIGRATED	☑
0x55130D69C5D930	YYY456	INTER	DEFAULT	USED	2018-03-06 16:33:31	11.2.0.4.0	bh0r63861q711	203103265	SELECT ROWID,MA	COMPATIBLE	ENABLED	NORMAL	NOT-MIGRATED	□
0x2693D6EACA39E	YYY330	INTER	DEFAULT	USED	2016-11-21 14:22:23	11.2.0.4.0	gv5179r0mmrk0	3241795136	SELECT SEK_KOD,	COMPATIBLE	ENABLED	NORMAL	NOT-MIGRATED	□

DETAILS FOR SELECTED PLAN OBJECT

Signature	Change type	Snap Logdate	Name	Owner	Category	Used	Timestamp	Version	Sql Id	Hash Value	Statement text	Compatible	Enabled	Format	Migrated	Is Droppe
0xA2279E7FF072	Delete	2018-12-10 11:22:	YYY513	INTER	DEFAULT	USED	2018-12-10 08:42:	11.2.0.4.0	1gd52hwxvr3yy	1002147806 +	SELECT STATUS_	COMPATIBLE	ENABLED	NORMAL	NOT-MIGRATED	☑
0xA2279E7FF072	Insert	2018-12-10 08:42:	YYY513	INTER	DEFAULT	USED	2018-12-10 08:42:	11.2.0.4.0	1gd52hwxvr3yy	1002147806 +	SELECT STATUS_	COMPATIBLE	ENABLED	NORMAL	NOT-MIGRATED	□

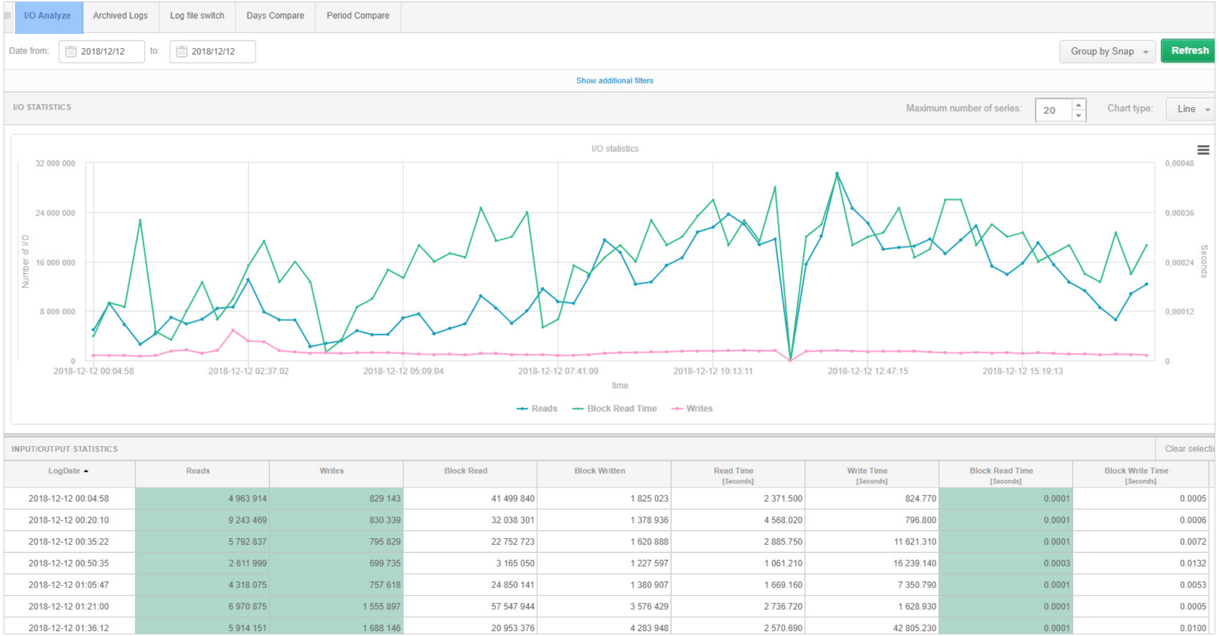
I/O Stats

DBPLUS
better performance

The module is used to analyze the performance of disk subassemblies.

Information is available on:

- Number of reads
- Number of writes
- Duration of the reads
- Duration of the writes



Date from: 2018/12/12 to: 2018/12/12

I/O STATISTICS

Maximum number of series: 20

Chart type: Line

INPUT/OUTPUT STATISTICS

LogDate ▲	Reads	Writes	Block Read	Block Written	Read Time [Seconds]	Write Time [Seconds]	Block Read Time [Seconds]	Block Write Time [Seconds]
2018-12-12 00:04:58	4 963 914	829 143	41 499 840	1 825 023	2 371.500	824.770	0.0001	0.0005
2018-12-12 00:20:10	9 243 469	830 339	32 038 301	1 378 936	4 568.020	796.800	0.0001	0.0006
2018-12-12 00:35:22	5 792 837	795 829	22 752 723	1 620 888	2 885.750	11 621.310	0.0001	0.0072
2018-12-12 00:50:35	2 611 999	699 735	3 165 050	1 227 597	1 061.210	16 239.140	0.0003	0.0132
2018-12-12 01:05:47	4 318 075	757 618	24 850 141	1 380 907	1 669.160	7 350.790	0.0001	0.0053
2018-12-12 01:21:00	6 970 875	1 555 897	57 547 944	3 576 429	2 736.720	1 628.930	0.0001	0.0005
2018-12-12 01:36:12	5 914 151	1 688 146	20 953 376	4 283 948	2 570.690	42 805.230	0.0001	0.0100

Space Monitor

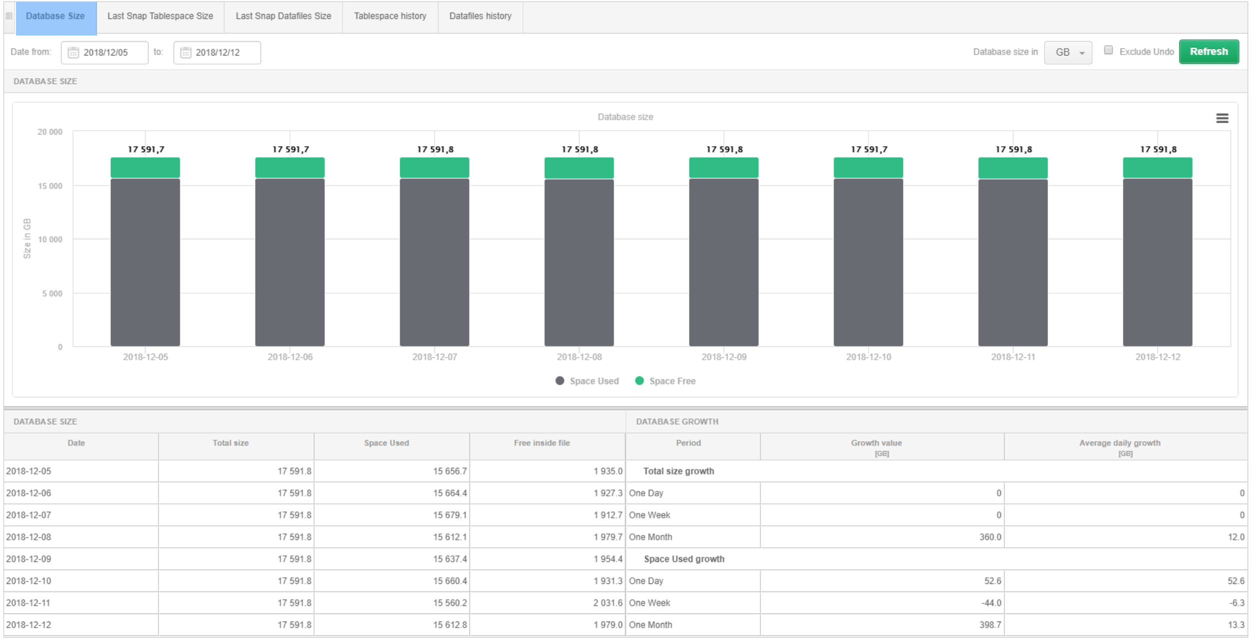
DBPLUS
better performance

Allows you to analyze the storage space occupancy by:

- database
- tablespaces
- datafiles

It is possible to verify historical data.

Information on average consumption for a given *day, week, month*.



Date from: 2018/12/05 to: 2018/12/12

Database size in GB

DATABASE SIZE

Date	Total size	Space Used	Free inside file
2018-12-05	17 591.8	15 656.7	1 935.0
2018-12-06	17 591.8	15 664.4	1 927.3
2018-12-07	17 591.8	15 679.1	1 912.7
2018-12-08	17 591.8	15 612.1	1 979.7
2018-12-09	17 591.8	15 637.4	1 954.4
2018-12-10	17 591.8	15 660.4	1 931.3
2018-12-11	17 591.8	15 560.2	2 031.6
2018-12-12	17 591.8	15 612.8	1 979.0

DATABASE GROWTH

Period	Growth value [GB]	Average daily growth [GB]
Total size growth		
One Day	0	0
One Week	0	0
One Month	360.0	12.0
Space Used growth		
One Day	52.6	52.6
One Week	-44.0	-6.3
One Month	398.7	13.3

Memory

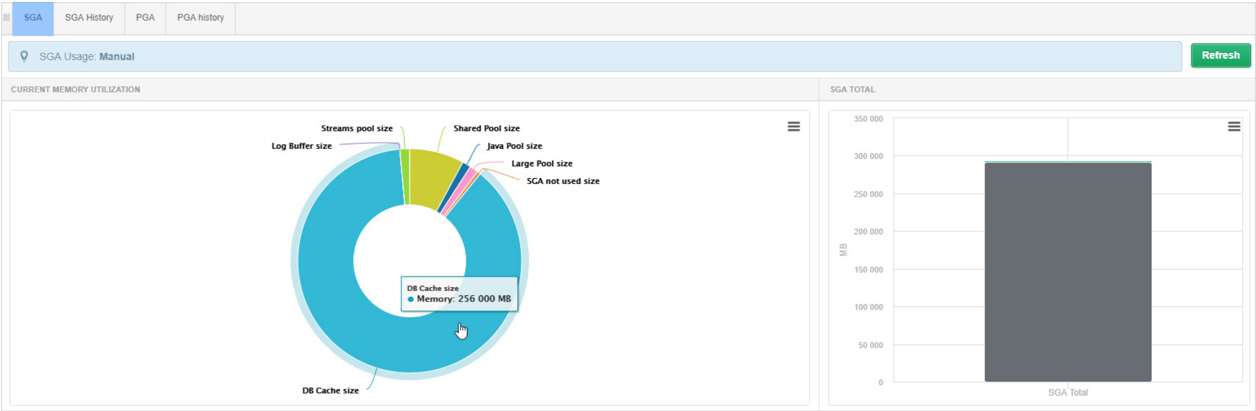
DBPLUS
better performance

Presents information about the memory used by a database.

Displays current memory usage.

History of memory usage over time.

Memory usage by user sessions.

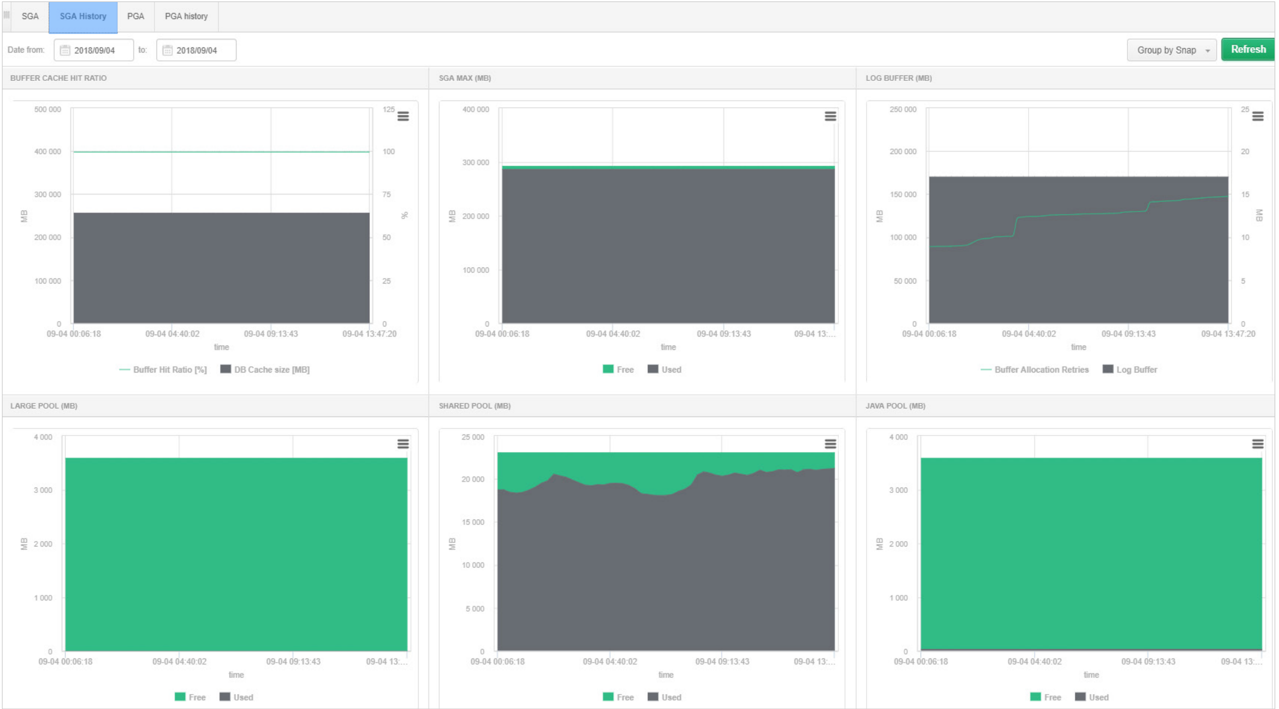


SGA Usage: **Manual**

DB Cache size
● Memory: 256 000 MB

Memory

SGA History presents the values of individual buffers and their utilisation in a given period of time.



Date from: 2018/09/04 to: 2018/09/04

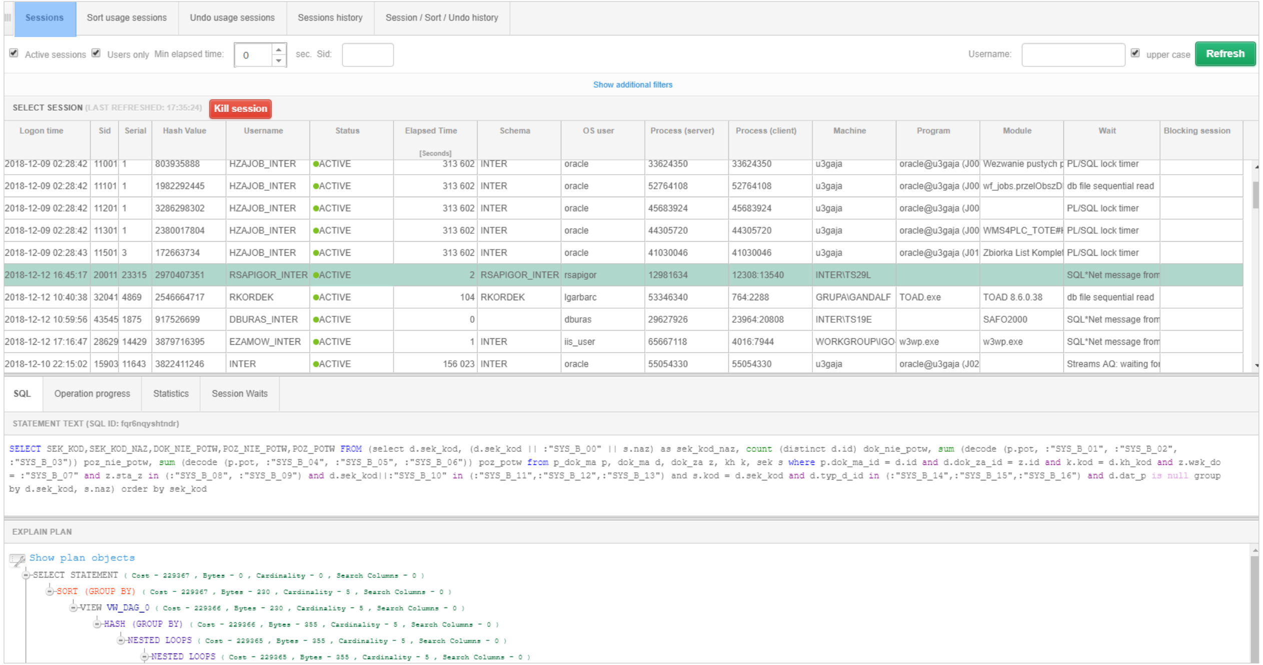
Sessions

DBPLUS
better performance

Stores information about sessions in a database, displayed according to the criteria in the filters.

Sort usage sessions
- a screen that allows for session analysis for the use of temporary space (e.g. a query session that sorts a large amount of data).

Undo usage session - functionality that allows for session analysis for the use of UNDO space (e.g. sessions holding a large portion of data in an uncommitted transaction).



☑ Active sessions ☑ Users only Min elapsed time: 0 sec. Sid: Username: ☑ upper case

SELECT SESSION (LAST REFRESHED: 17:35:24)

Logon time	Sid	Serial	Hash Value	Username	Status	Elapsed Time [Seconds]	Schema	OS user	Process (server)	Process (client)	Machine	Program	Module	Wait	Blocking session
2018-12-09 02:28:42	11001	1	803935888	HZAJOB_INTER	●ACTIVE	313 602	INTER	oracle	33624350	33624350	u3gaja	oracle@u3gaja (J00	Wezwanie pustych p	PL/SQL lock timer	
2018-12-09 02:28:42	11101	1	1982292445	HZAJOB_INTER	●ACTIVE	313 602	INTER	oracle	52764108	52764108	u3gaja	oracle@u3gaja (J00	wf_jobs.przelObszD	db file sequential read	
2018-12-09 02:28:42	11201	1	3286298302	HZAJOB_INTER	●ACTIVE	313 602	INTER	oracle	45683924	45683924	u3gaja	oracle@u3gaja (J00		PL/SQL lock timer	
2018-12-09 02:28:42	11301	1	2380017804	HZAJOB_INTER	●ACTIVE	313 602	INTER	oracle	44305720	44305720	u3gaja	oracle@u3gaja (J00	WMS4PLC_TOTE#I	PL/SQL lock timer	
2018-12-09 02:28:43	11501	3	172663734	HZAJOB_INTER	●ACTIVE	313 602	INTER	oracle	41030046	41030046	u3gaja	oracle@u3gaja (J01	Zbiorka List Komplet	PL/SQL lock timer	
2018-12-12 16:45:17	20011	23315	2970407351	RSAPIGOR_INTER	●ACTIVE	2	RSAPIGOR_INTER	rsapigor	12981634	12308:13540	INTER\TS29L			SQL*Net message from	
2018-12-12 10:40:38	32041	4869	2546664717	RKORDEK	●ACTIVE	104	RKORDEK	lgarbarc	53346340	764:2288	GRUPA\GANDALF	TOAD.exe	TOAD 8.6.0.38	db file sequential read	
2018-12-12 10:59:56	43545	1875	917526699	DBURAS_INTER	●ACTIVE	0		dburas	29627926	23964:20808	INTER\TS19E		SAFO2000	SQL*Net message from	
2018-12-12 17:16:47	28629	14429	3879716395	EZAMOW_INTER	●ACTIVE	1	INTER	iis_user	65667118	4016:7944	WORKGROUP\IGO	w3wp.exe	w3wp.exe	SQL*Net message from	
2018-12-10 22:15:02	15903	11643	3822411246	INTER	●ACTIVE	156 023	INTER	oracle	55054330	55054330	u3gaja	oracle@u3gaja (J02		Streams AQ: waiting for	

STATEMENT TEXT (SQL ID: fqr6nqyshtndr)

```
SELECT SEK_KOD,SEK_KOD_NAZ,DOK_NIE_POTW,POZ_NIE_POTW,POZ_POTW FROM (select d.sek_kod, (d.sek_kod || :"SYS_B_00" || s.naz) as sek_kod_naz, count (distinct d.id) dok_nie_potw, sum (decode (p.pot, :"SYS_B_01", :"SYS_B_02",
:"SYS_B_03")) poz_nie_potw, sum (decode (p.pot, :"SYS_B_04", :"SYS_B_05", :"SYS_B_06")) poz_potw from p_dok_ma p, dok_ma d, dok_za z, kh k, sek s where p.dok_ma_id = d.id and d.dok_za_id = z.id and k.kod = d.kh_kod and z.wsk_do
= :"SYS_B_07" and z.sta_z in (:"SYS_B_08", :"SYS_B_09") and d.sek_kod||:"SYS_B_10" in (:"SYS_B_11",:"SYS_B_12",:"SYS_B_13") and s.kod = d.sek_kod and d.typ_d_id in (:"SYS_B_14",:"SYS_B_15",:"SYS_B_16") and d.dat_p is null group
by d.sek_kod, s.naz) order by sek_kod
```

EXPLAIN PLAN

```
Show plan objects
⊖-SELECT STATEMENT ( Cost - 229367 , Bytes - 0 , Cardinality - 0 , Search Columns - 0 )
  ⊖-SORT (GROUP BY) ( Cost - 229367 , Bytes - 230 , Cardinality - 5 , Search Columns - 0 )
    ⊖-VIEW VW_DAG_0 ( Cost - 229366 , Bytes - 230 , Cardinality - 5 , Search Columns - 0 )
      ⊖-HASH (GROUP BY) ( Cost - 229366 , Bytes - 355 , Cardinality - 5 , Search Columns - 0 )
        ⊖-NESTED LOOPS ( Cost - 229365 , Bytes - 355 , Cardinality - 5 , Search Columns - 0 )
          ⊖-NESTED LOOPS ( Cost - 229365 , Bytes - 355 , Cardinality - 5 , Search Columns - 0 )
```

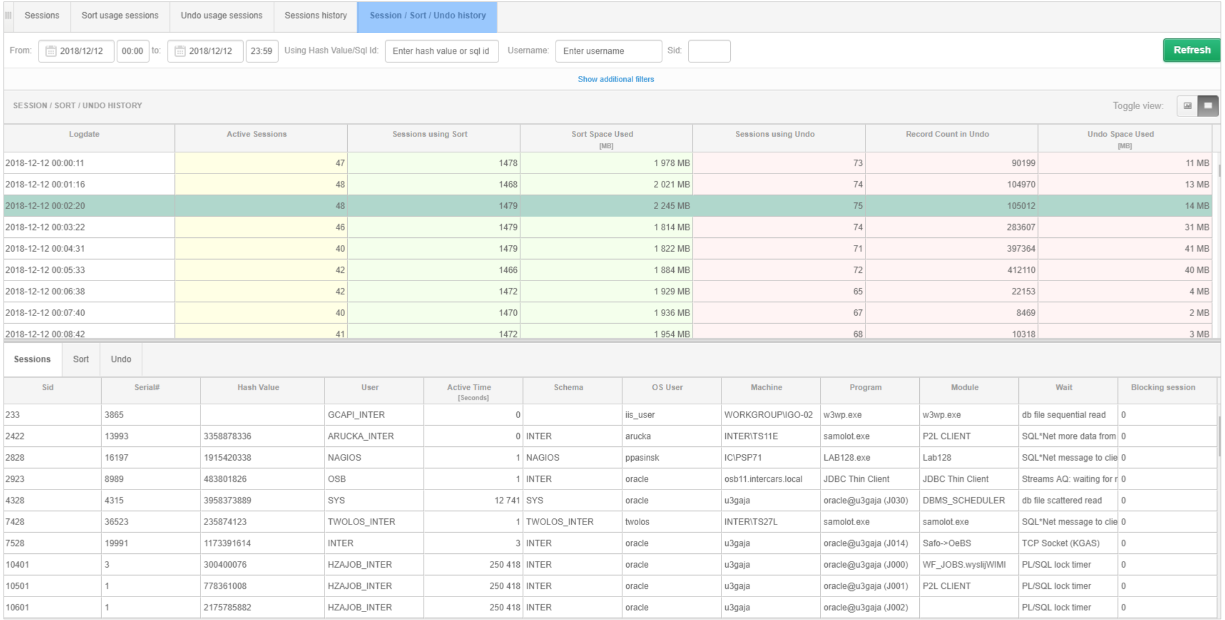
Sessions history

The table is divided into three groups:

the yellow group shows information about active sessions,

the green group shows information about sessions using sorting,

the red group shows information about sessions using the Undo.



From: 2018/12/12 00:00 to: 2018/12/12 23:59 Using Hash Value/Sql Id: Enter hash value or sql id Username: Enter username Sid:

SESSION / SORT / UNDO HISTORY

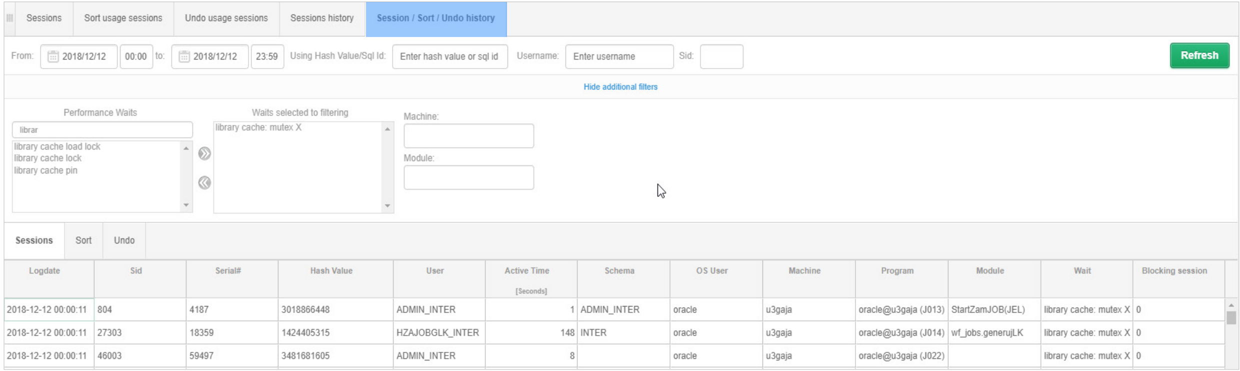
Logdate	Active Sessions	Sessions using Sort	Sort Space Used [MB]	Sessions using Undo	Record Count in Undo	Undo Space Used [MB]
2018-12-12 00:00:11	47	1478	1 978 MB	73	90199	11 MB
2018-12-12 00:01:16	48	1468	2 021 MB	74	104970	13 MB
2018-12-12 00:02:20	48	1479	2 245 MB	75	105012	14 MB
2018-12-12 00:03:22	46	1479	1 814 MB	74	283607	31 MB
2018-12-12 00:04:31	40	1479	1 822 MB	71	397364	41 MB
2018-12-12 00:05:33	42	1466	1 884 MB	72	412110	40 MB
2018-12-12 00:06:38	42	1472	1 929 MB	65	22153	4 MB
2018-12-12 00:07:40	40	1470	1 936 MB	67	8469	2 MB
2018-12-12 00:08:42	41	1472	1 954 MB	68	10318	3 MB

Sid	Serial#	Hash Value	User	Active Time [Seconds]	Schema	OS User	Machine	Program	Module	Wait	Blocking session
233	3865		GCAPI_INTER	0		iis_user	WORKGROUP\IGO-02	w3wp.exe	w3wp.exe	db file sequential read	0
2422	13993	3358878336	ARUCKA_INTER	0	INTER	arucka	INTER\TS11E	samolot.exe	P2L CLIENT	SQL*Net more data from	0
2828	16197	1915420338	NAGIOS	1	NAGIOS	ppasinsk	IC\PSP71	LAB128.exe	Lab128	SQL*Net message to clie	0
2923	8989	483801826	OSB	1	INTER	oracle	osb11.intercars.local	JDBC Thin Client	JDBC Thin Client	Streams AQ: waiting for r	0
4328	4315	3958373889	SYS	12 741	SYS	oracle	u3gaja	oracle@u3gaja (J030)	DBMS_SCHEDULER	db file scattered read	0
7428	36523	235874123	TWOLOS_INTER	1	TWOLOS_INTER	twolos	INTER\TS27L	samolot.exe	samolot.exe	SQL*Net message to clie	0
7528	19991	1173391614	INTER	3	INTER	oracle	u3gaja	oracle@u3gaja (J014)	Safo->OeBS	TCP Socket (KGAS)	0
10401	3	300400076	HZAJOB_INTER	250 418	INTER	oracle	u3gaja	oracle@u3gaja (J000)	WF_JOBS.wyslijWIMI	PL/SQL lock timer	0
10501	1	778361008	HZAJOB_INTER	250 418	INTER	oracle	u3gaja	oracle@u3gaja (J001)	P2L CLIENT	PL/SQL lock timer	0
10601	1	2175785882	HZAJOB_INTER	250 418	INTER	oracle	u3gaja	oracle@u3gaja (J002)		PL/SQL lock timer	0

Sessions history

Sessions can be sorted using:

- Hash value
- Username
- Sid
- Wait type
- Machine
- Module



From: 2018/12/12 00:00 to: 2018/12/12 23:59 Using Hash Value/Sql Id: Enter hash value or sql id Username: Enter username Sid:

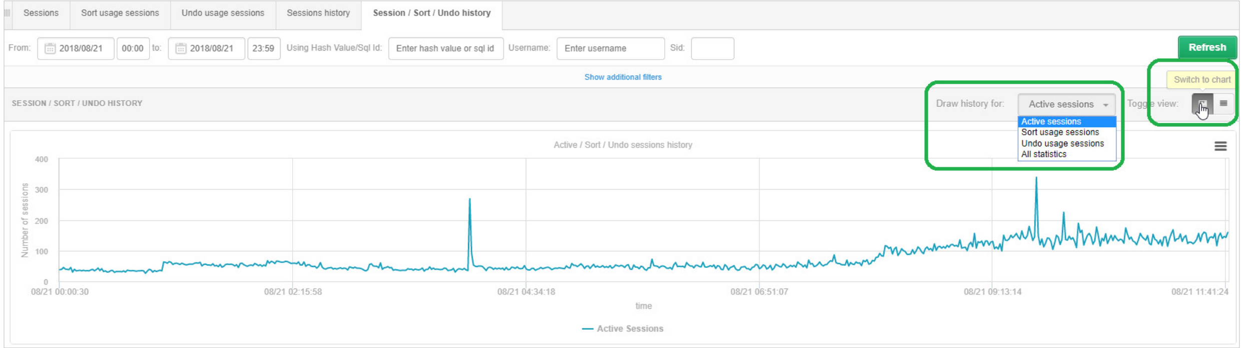
Performance Waits: librar
library cache load lock
library cache lock
library cache pin

Waits selected to filtering: library cache: mutex X

Machine:

Module:

Logdate	Sid	Serial#	Hash Value	User	Active Time [Seconds]	Schema	OS User	Machine	Program	Module	Wait	Blocking session
2018-12-12 00:00:11	804	4187	3018866448	ADMIN_INTER	1	ADMIN_INTER	oracle	u3gaja	oracle@u3gaja (J013)	StartZamJOB(JEL)	library cache: mutex X	0
2018-12-12 00:00:11	27303	18359	1424405315	HZAJOBGLK_INTER	148	INTER	oracle	u3gaja	oracle@u3gaja (J014)	wf_jobs.generujLK	library cache: mutex X	0
2018-12-12 00:00:11	46003	59497	3481681605	ADMIN_INTER	8		oracle	u3gaja	oracle@u3gaja (J022)		library cache: mutex X	0



From: 2018/08/21 00:00 to: 2018/08/21 23:59 Using Hash Value/Sql Id: Enter hash value or sql id Username: Enter username Sid:

Draw history for: Active sessions

In addition, information can be viewed in the form of a graph.

Locks

DBPLUS
better performance

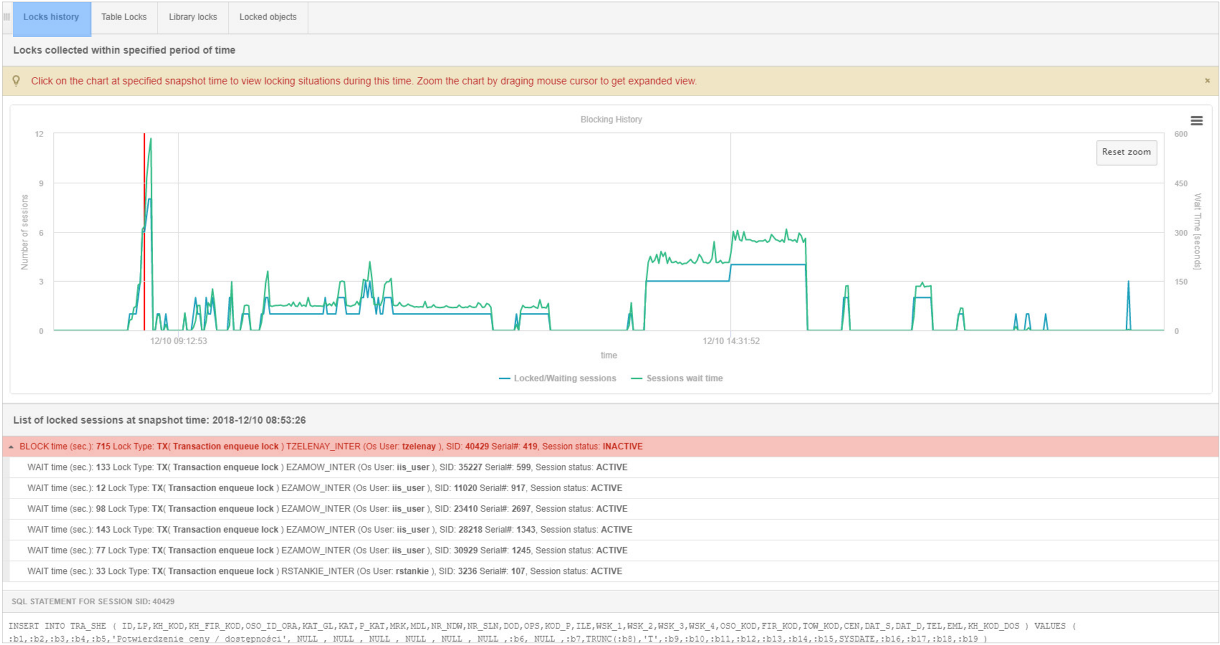
Contains information about the locks occurring in a given database.

Locks history
- allows you to track locks in time.

Table Locks
- allows for the current analysis of locks in a database.

Library Locks
- allows for the analysis of Library cache locks.

Locked Objects
- shows a list of objects on which locks are currently installed.



Locks collected within specified period of time

Click on the chart at specified snapshot time to view locking situations during this time. Zoom the chart by draging mouse cursor to get expanded view.

List of locked sessions at snapshot time: 2018-12/10 08:53:26

BLOCK time (sec.): **715** Lock Type: **TX**(**Transaction enqueue lock**) TZELENAY_INTER (Os User: **tzelenay**), SID: **40429** Serial#: **419**, Session status: **INACTIVE**

WAIT time (sec.): **133** Lock Type: **TX**(**Transaction enqueue lock**) EZAMOW_INTER (Os User: **iis_user**), SID: **35227** Serial#: **599**, Session status: **ACTIVE**

WAIT time (sec.): **12** Lock Type: **TX**(**Transaction enqueue lock**) EZAMOW_INTER (Os User: **iis_user**), SID: **11020** Serial#: **917**, Session status: **ACTIVE**

WAIT time (sec.): **98** Lock Type: **TX**(**Transaction enqueue lock**) EZAMOW_INTER (Os User: **iis_user**), SID: **23410** Serial#: **2697**, Session status: **ACTIVE**

WAIT time (sec.): **143** Lock Type: **TX**(**Transaction enqueue lock**) EZAMOW_INTER (Os User: **iis_user**), SID: **28218** Serial#: **1343**, Session status: **ACTIVE**

WAIT time (sec.): **77** Lock Type: **TX**(**Transaction enqueue lock**) EZAMOW_INTER (Os User: **iis_user**), SID: **30929** Serial#: **1245**, Session status: **ACTIVE**

WAIT time (sec.): **33** Lock Type: **TX**(**Transaction enqueue lock**) RSTANKIE_INTER (Os User: **rstankie**), SID: **3236** Serial#: **107**, Session status: **ACTIVE**

SQL STATEMENT FOR SESSION SID: 40429

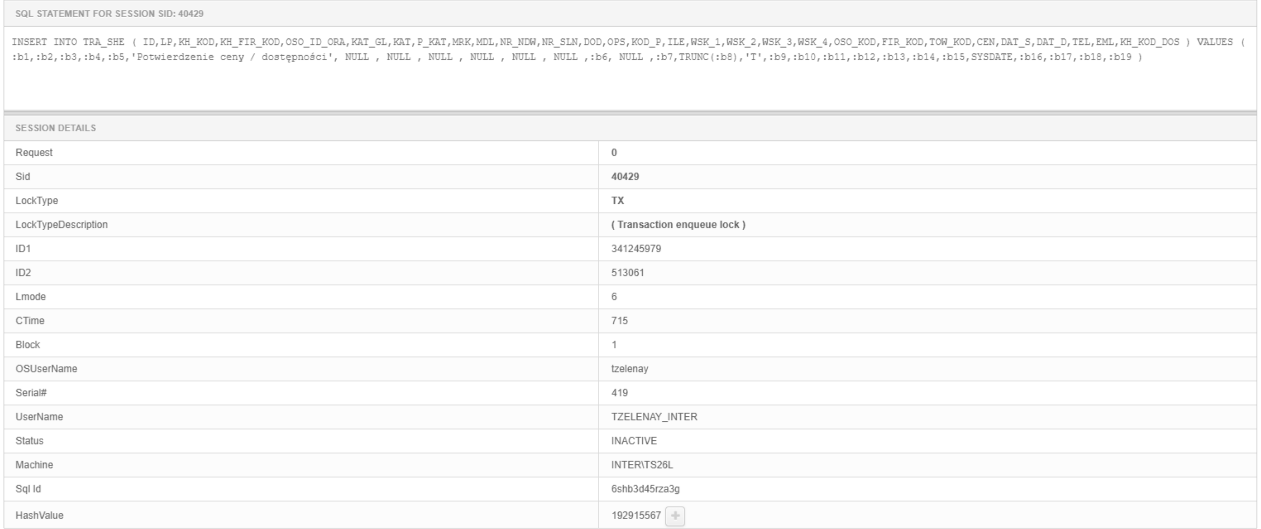
```
INSERT INTO TRA_SHE ( ID,LP,KH_KOD,KH_FIR_KOD,OSO_ID_ORA,KAT_GL,KAT,P_KAT,MRK,MDL,NR_NDW,NR_SLN,DOD,OPS,KOD_P,ILE,WSK_1,WSK_2,WSK_3,WSK_4,OSO_KOD,FIR_KOD,TOW_KOD,CEN,DAT_S,DAT_D,TEL,EML,KH_KOD_DOS ) VALUES (
:b1,:b2,:b3,:b4,:b5,'Potwierdzenie ceny / dostępności', NULL , NULL , NULL , NULL , NULL , NULL ,:b6, NULL ,:b7,TRUNC(:b8),'T',:b9,:b10,:b11,:b12,:b13,:b14,:b15,SYSDATE,:b16,:b17,:b18,:b19 )
```

Locks

DBPLUS
better performance

In addition, after selecting the session, we receive information such as:

- Text of the query
- Session parameters
- Query identifier
- Status
- Lock type



SQL STATEMENT FOR SESSION SID: 40429

```
INSERT INTO TRA_SHE ( ID,LP,KH_KOD,KH_FIR_KOD,OSO_ID_ORA,KAT_GL,KAT,P_KAT,MRK,MDL,NR_NDW,NR_SLN,DOD,OPS,KOD_P,ILE,WSK_1,WSK_2,WSK_3,WSK_4,OSO_KOD,FIR_KOD,TOW_KOD,CEN,DAT_S,DAT_D,TEL,EML,KH_KOD_DOS ) VALUES (
:b1,:b2,:b3,:b4,:b5,'Potwierdzenie ceny / dostępności', NULL , NULL , NULL , NULL , NULL , NULL ,:b6, NULL ,:b7,TRUNC(:b8),'T',:b9,:b10,:b11,:b12,:b13,:b14,:b15,SYSDATE,:b16,:b17,:b18,:b19 )
```

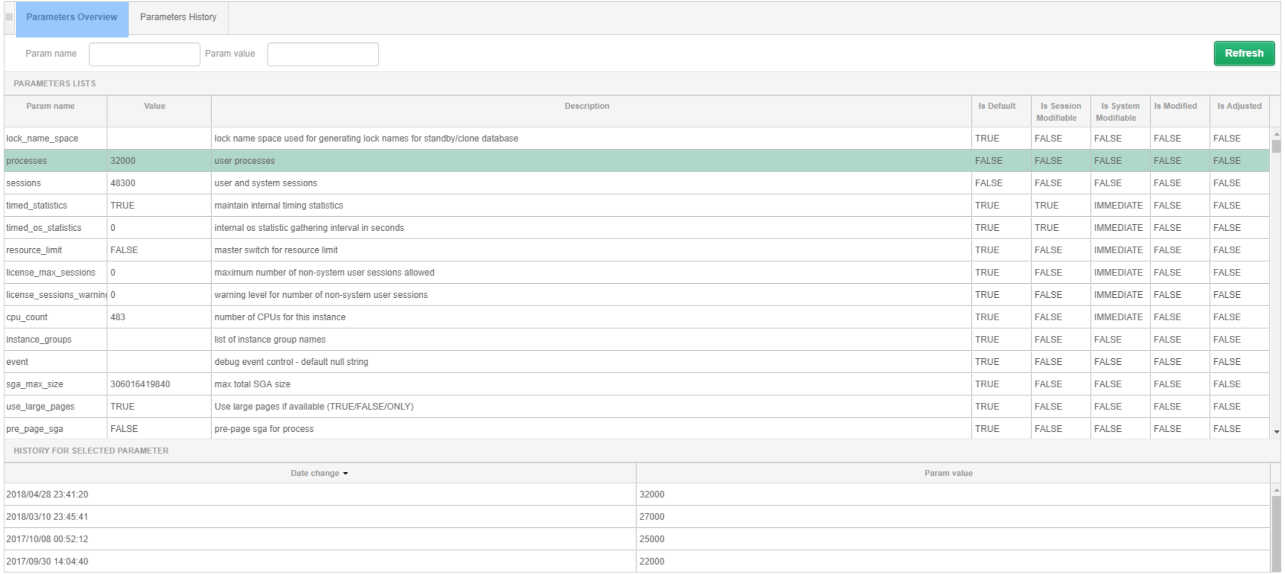
SESSION DETAILS

Request	**0**
Sid	**40429**
LockType	**TX**
LockTypeDescription	**(Transaction enqueue lock)**
ID1	341245979
ID2	513061
Lmode	6
CTime	715
Block	1
OSUserName	tzelenay
Serial#	419
UserName	TZELENAY_INTER
Status	INACTIVE
Machine	INTER\TS26L
Sql Id	6shb3d45rza3g
HashValue	192915567 +

Parameters

Allows you to view and report changes in database parameters over time.

The window presents the current status of parameters and their changes over time.



Param name	Value	Description	Is Default	Is Session Modifiable	Is System Modifiable	Is Modified	Is Adjusted
lock_name_space		lock name space used for generating lock names for standby/clone database	TRUE	FALSE	FALSE	FALSE	FALSE
processes	32000	user processes	FALSE	FALSE	FALSE	FALSE	FALSE
sessions	48300	user and system sessions	FALSE	FALSE	FALSE	FALSE	FALSE
timed_statistics	TRUE	maintain internal timing statistics	TRUE	TRUE	IMMEDIATE	FALSE	FALSE
timed_os_statistics	0	internal os statistic gathering interval in seconds	TRUE	TRUE	IMMEDIATE	FALSE	FALSE
resource_limit	FALSE	master switch for resource limit	TRUE	FALSE	IMMEDIATE	FALSE	FALSE
license_max_sessions	0	maximum number of non-system user sessions allowed	TRUE	FALSE	IMMEDIATE	FALSE	FALSE
license_sessions_warnin	0	warning level for number of non-system user sessions	TRUE	FALSE	IMMEDIATE	FALSE	FALSE
cpu_count	483	number of CPUs for this instance	TRUE	FALSE	IMMEDIATE	FALSE	FALSE
instance_groups		list of instance group names	TRUE	FALSE	FALSE	FALSE	FALSE
event		debug event control - default null string	TRUE	FALSE	FALSE	FALSE	FALSE
sga_max_size	306016419840	max total SGA size	TRUE	FALSE	FALSE	FALSE	FALSE
use_large_pages	TRUE	Use large pages if available (TRUE/FALSE/ONLY)	TRUE	FALSE	FALSE	FALSE	FALSE
pre_page_sga	FALSE	pre-page sga for process	TRUE	FALSE	FALSE	FALSE	FALSE

Date change	Param value
2018/04/28 23:41:20	32000
2018/03/10 23:45:41	27000
2017/10/08 00:52:12	25000
2017/09/30 14:04:40	22000

Anomaly Monitor

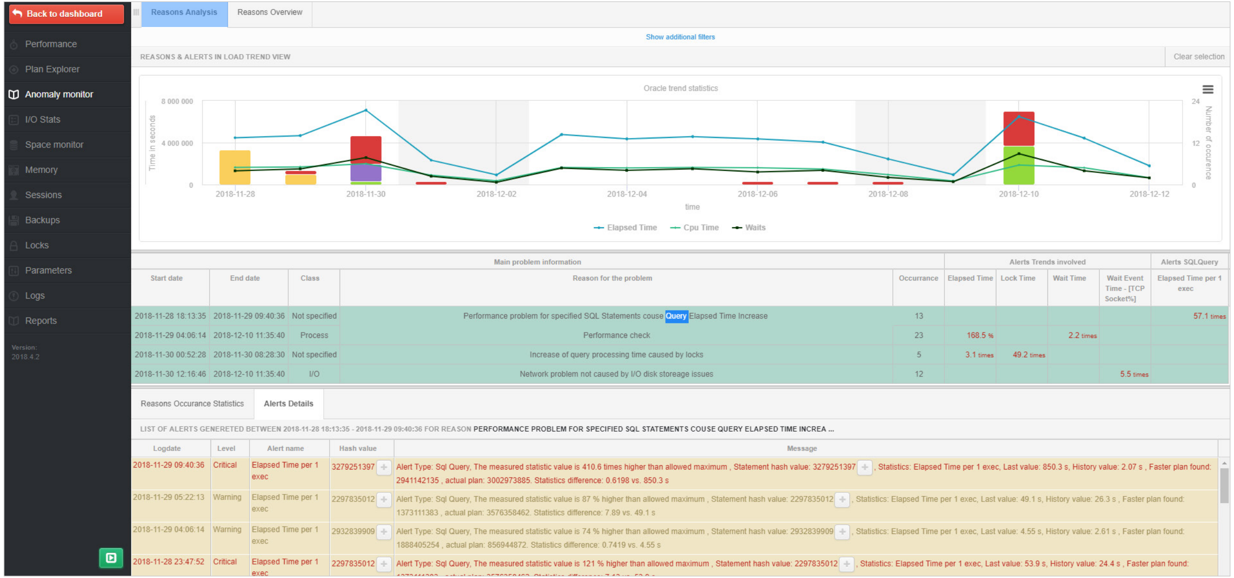
DBPLUS
better performance

A module containing information about problems affecting database performance.

Information is available from the level of the monitored database.

Two types of Alerting:

- Online
- Trends



Main problem information					Alerts Trends involved				Alerts SQLQuery
Start date	End date	Class	Reason for the problem	Occurrance	Elapsed Time	Lock Time	Wait Time	Wait Event Time - [TCP Socket%]	Elapsed Time per 1 exec
2018-11-28 18:13:35	2018-11-29 09:40:36	Not specified	Performance problem for specified SQL Statements couse Query Elapsed Time Increase	13					57.1 times
2018-11-29 04:06:14	2018-12-10 11:35:40	Process	Performance check	23	168.5 %		2.2 times		
2018-11-30 00:52:28	2018-11-30 08:28:30	Not specified	Increase of query processing time caused by locks	5	3.1 times	49.2 times			
2018-11-30 12:16:46	2018-12-10 11:35:40	I/O	Network problem not caused by I/O disk storeage issues	12				5.5 times	

Reasons Occurance Statistics | Alerts Details

LIST OF ALERTS GENERETED BETWEEN 2018-11-28 18:13:35 - 2018-11-29 09:40:36 FOR REASON **PERFORMANCE PROBLEM FOR SPECIFIED SQL STATEMENTS COUSE QUERY ELAPSED TIME INCREA ...**

Logdate	Level	Alert name	Hash value	Message
2018-11-29 09:40:36	Critical	Elapsed Time per 1 exec	3279251397 +	Alert Type: Sql Query, The measured statistic value is 410.6 times higher than allowed maximum , Statement hash value: 3279251397 + , Statistics: Elapsed Time per 1 exec, Last value: 850.3 s, History value: 2.07 s , Faster plan found: 2941142135 , actual plan: 3002973885. Statistics difference: 0.6198 vs. 850.3 s
2018-11-29 05:22:13	Warning	Elapsed Time per 1 exec	2297835012 +	Alert Type: Sql Query, The measured statistic value is 87 % higher than allowed maximum , Statement hash value: 2297835012 + , Statistics: Elapsed Time per 1 exec, Last value: 49.1 s, History value: 26.3 s , Faster plan found: 1373111383 , actual plan: 3576358462. Statistics difference: 7.89 vs. 49.1 s
2018-11-29 04:06:14	Warning	Elapsed Time per 1 exec	2932839909 +	Alert Type: Sql Query, The measured statistic value is 74 % higher than allowed maximum , Statement hash value: 2932839909 + , Statistics: Elapsed Time per 1 exec, Last value: 4.55 s, History value: 2.61 s , Faster plan found: 1888405254 , actual plan: 856944872. Statistics difference: 0.7419 vs. 4.55 s
2018-11-28 23:47:52	Critical	Elapsed Time per 1 exec	2297835012 +	Alert Type: Sql Query, The measured statistic value is 121 % higher than allowed maximum , Statement hash value: 2297835012 + , Statistics: Elapsed Time per 1 exec, Last value: 53.9 s, History value: 24.4 s , Faster plan found:

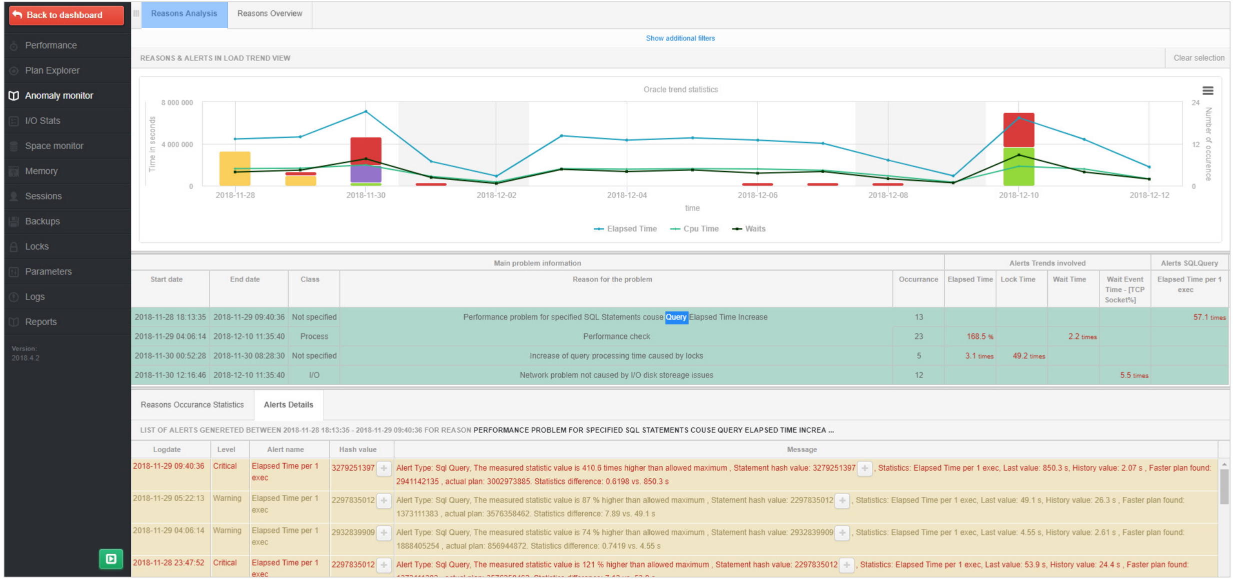
Anomaly Monitor

DBPLUS
better performance

Information about Alerts is visible in the form of a graph.

Grouped by the reasons for their creation and their impact on the given statistics in a database.

Presented in detail for a given period of time.



Main problem information					Alerts Trends involved				Alerts SQLQuery
Start date	End date	Class	Reason for the problem	Occurrance	Elapsed Time	Lock Time	Wait Time	Wait Event Time - [TCP Socket%]	Elapsed Time per 1 exec
2018-11-28 18:13:35	2018-11-29 09:40:36	Not specified	Performance problem for specified SQL Statements couse Query Elapsed Time Increase	13					57.1 times
2018-11-29 04:06:14	2018-12-10 11:35:40	Process	Performance check	23	168.5 %		2.2 times		
2018-11-30 00:52:28	2018-11-30 08:28:30	Not specified	Increase of query processing time caused by locks	5	3.1 times	49.2 times			
2018-11-30 12:16:46	2018-12-10 11:35:40	I/O	Network problem not caused by I/O disk storeage issues	12				5.5 times	

Reasons Occurance Statistics | Alerts Details

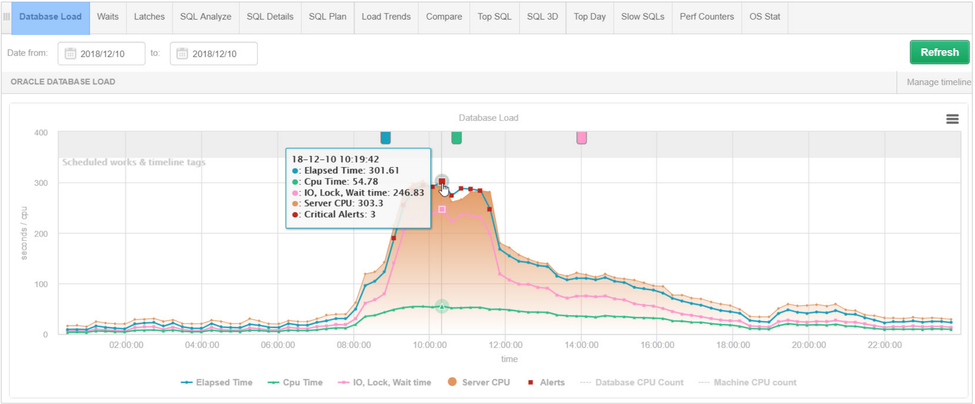
LIST OF ALERTS GENERETED BETWEEN 2018-11-28 18:13:35 - 2018-11-29 09:40:36 FOR REASON PERFORMANCE PROBLEM FOR SPECIFIED SQL STATEMENTS COUSE QUERY ELAPSED TIME INCREA ...

Logdate	Level	Alert name	Hash value	Message
2018-11-29 09:40:36	Critical	Elapsed Time per 1 exec	3279251397 +	Alert Type: Sql Query, The measured statistic value is 410.6 times higher than allowed maximum , Statement hash value: 3279251397 + , Statistics: Elapsed Time per 1 exec, Last value: 850.3 s, History value: 2.07 s , Faster plan found: 2941142135 , actual plan: 3002973885. Statistics difference: 0.6198 vs. 850.3 s
2018-11-29 05:22:13	Warning	Elapsed Time per 1 exec	2297835012 +	Alert Type: Sql Query, The measured statistic value is 87 % higher than allowed maximum , Statement hash value: 2297835012 + , Statistics: Elapsed Time per 1 exec, Last value: 49.1 s, History value: 26.3 s , Faster plan found: 1373111383 , actual plan: 3576358462. Statistics difference: 7.89 vs. 49.1 s
2018-11-29 04:06:14	Warning	Elapsed Time per 1 exec	2932839909 +	Alert Type: Sql Query, The measured statistic value is 74 % higher than allowed maximum , Statement hash value: 2932839909 + , Statistics: Elapsed Time per 1 exec, Last value: 4.55 s, History value: 2.61 s , Faster plan found: 1888405254 , actual plan: 856944872. Statistics difference: 0.7419 vs. 4.55 s
2018-11-28 23:47:52	Critical	Elapsed Time per 1 exec	2297835012 +	Alert Type: Sql Query, The measured statistic value is 121 % higher than allowed maximum , Statement hash value: 2297835012 + , Statistics: Elapsed Time per 1 exec, Last value: 53.9 s, History value: 24.4 s , Faster plan found:

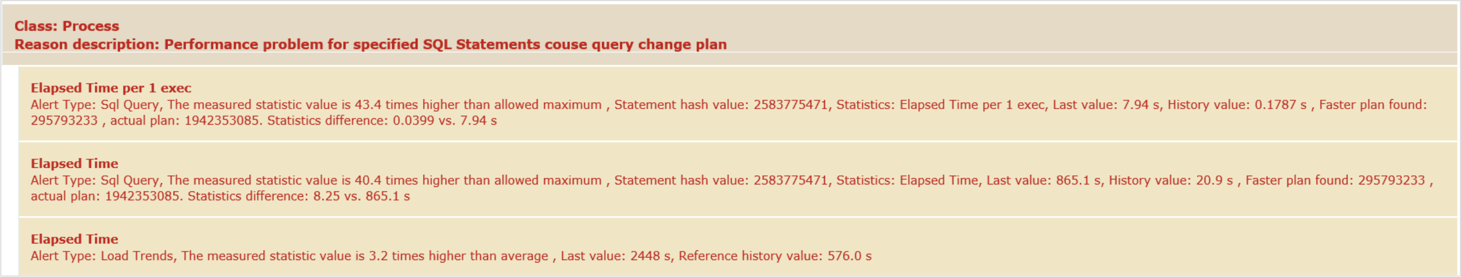
Anomaly Monitor - DatabaseLoad

DBPLUS
better performance

Information about Alerts is also visible on the chart in the Database Load tab.



Sample Alert informing about the change of the execution plan:



Class: Process
Reason description: Performance problem for specified SQL Statements couse query change plan

Elapsed Time per 1 exec
Alert Type: Sql Query, The measured statistic value is 43.4 times higher than allowed maximum , Statement hash value: 2583775471, Statistics: Elapsed Time per 1 exec, Last value: 7.94 s, History value: 0.1787 s , Faster plan found: 295793233 , actual plan: 1942353085. Statistics difference: 0.0399 vs. 7.94 s

Elapsed Time
Alert Type: Sql Query, The measured statistic value is 40.4 times higher than allowed maximum , Statement hash value: 2583775471, Statistics: Elapsed Time, Last value: 865.1 s, History value: 20.9 s , Faster plan found: 295793233 , actual plan: 1942353085. Statistics difference: 8.25 vs. 865.1 s

Elapsed Time
Alert Type: Load Trends, The measured statistic value is 3.2 times higher than average , Last value: 2448 s, Reference history value: 576.0 s

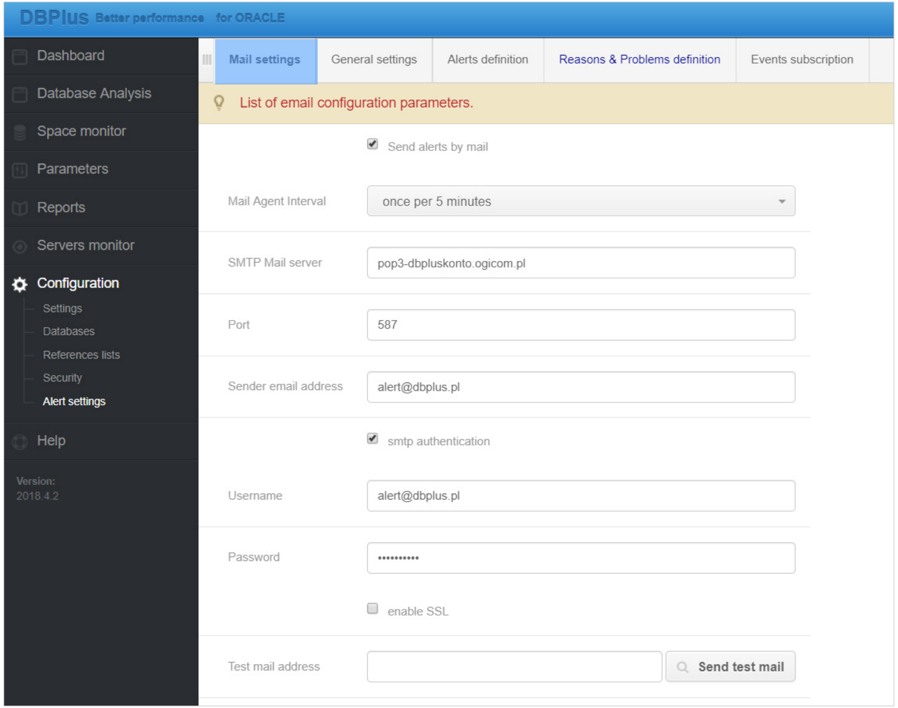
Anomaly Monitor - Configuration

DBPLUS
better performance

Configuration and alert definitions can be found in the menu:

Configuration > Alert settings

- Setting the mailbox



List of email configuration parameters.

☑ Send alerts by mail

Mail Agent Interval: once per 5 minutes

SMTP Mail server: pop3-dbpluskonto.ogicom.pl

Port: 587

Sender email address: alert@dbplus.pl

☑ smtp authentication

Username: alert@dbplus.pl

Password: ••••••••••

☐ enable SSL

Test mail address:

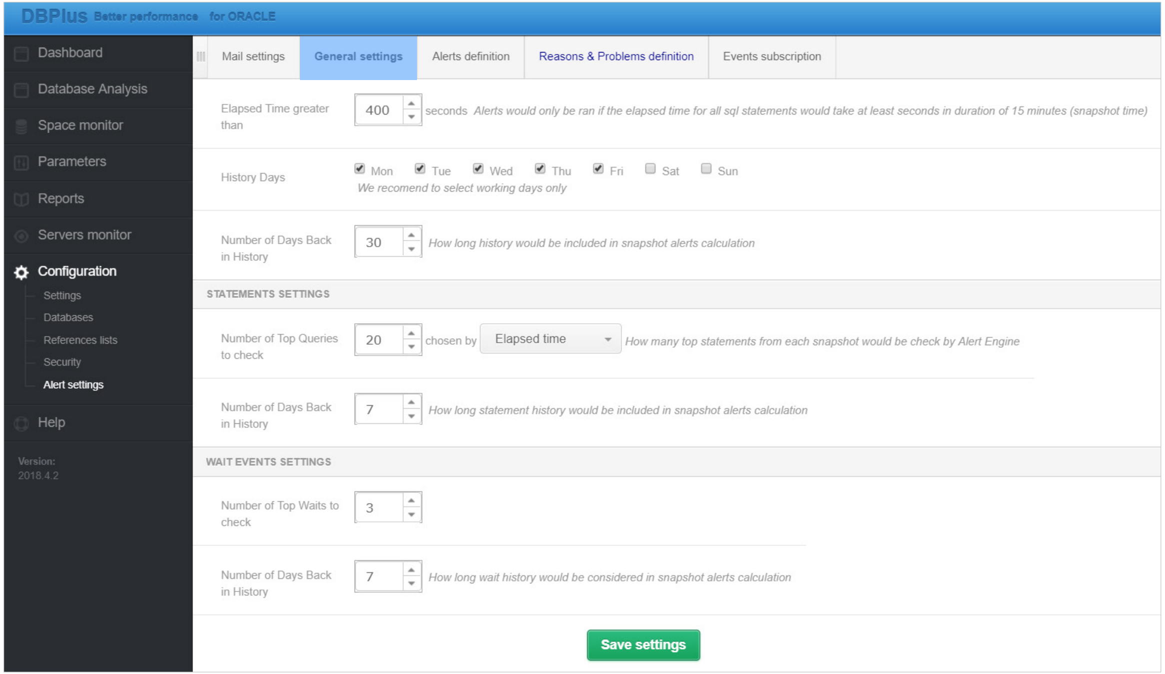
Anomaly Monitor - Configuration

DBPLUS
better performance

Configuration and alert definitions can be found in the menu:

Configuration > Alert settings

- General settings



Elapsed Time greater than: 400 seconds *Alerts would only be ran if the elapsed time for all sql statements would take at least seconds in duration of 15 minutes (snapshot time)*

History Days: ☑ Mon ☑ Tue ☑ Wed ☑ Thu ☑ Fri □ Sat □ Sun
We recomend to select working days only

Number of Days Back in History: 30 *How long history would be included in snapshot alerts calculation*

STATEMENTS SETTINGS

Number of Top Queries to check: 20 chosen by Elapsed time *How many top statements from each snapshot would be check by Alert Engine*

Number of Days Back in History: 7 *How long statement history would be included in snapshot alerts calculation*

WAIT EVENTS SETTINGS

Number of Top Waits to check: 3

Number of Days Back in History: 7 *How long wait history would be considered in snapshot alerts calculation*

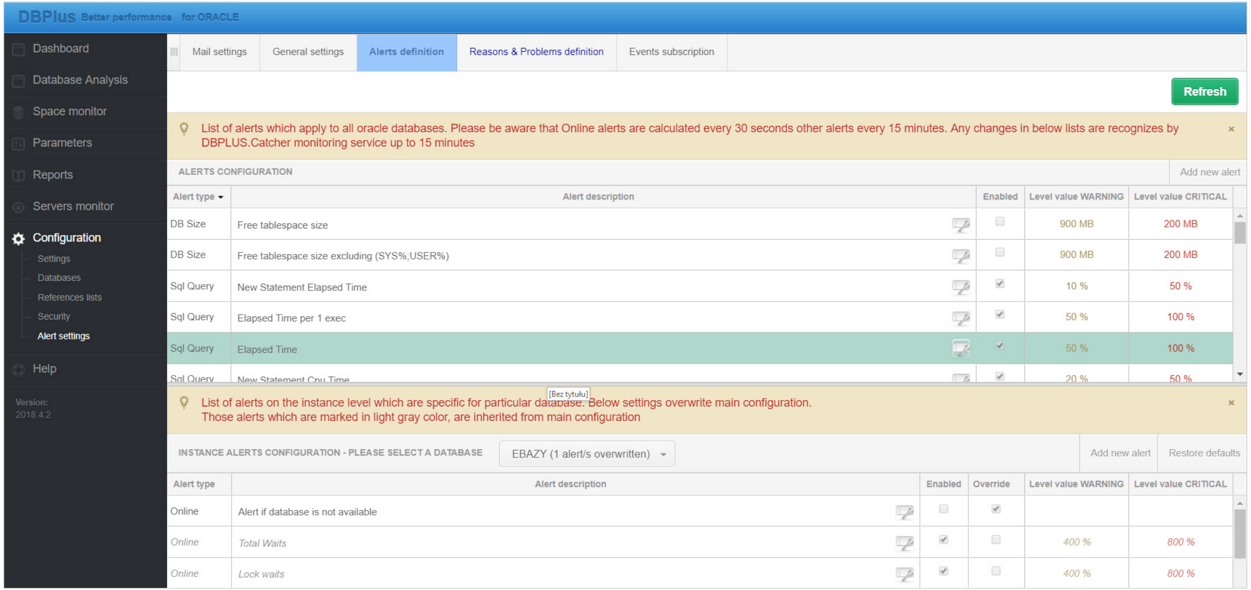
Anomaly Monitor - How does it work?

The anomaly monitor is based on gathering information about the statistics available in a database.

Alert Definitions
 - an alarm value is specified for each statistic.

Problem Definition
- a set of rules based on predefined Alerts.

Based on historical information, threshold exceeding events are generated.



List of alerts which apply to all oracle databases. Please be aware that Online alerts are calculated every 30 seconds other alerts every 15 minutes. Any changes in below lists are recognizes by DBPLUS.Catcher monitoring service up to 15 minutes

ALERTS CONFIGURATION

Alert type	Alert description	Enabled	Level value WARNING	Level value CRITICAL
DB Size	Free tablespace size	□	900 MB	200 MB
DB Size	Free tablespace size excluding (SYS%;USER%)	□	900 MB	200 MB
Sql Query	New Statement Elapsed Time	☑	10 %	50 %
Sql Query	Elapsed Time per 1 exec	☑	50 %	100 %
Sql Query	Elapsed Time	☑	50 %	100 %
Sql Query	New Statement Cpu Time	☑	20 %	50 %

List of alerts on the instance level which are specific for particular database. Below settings overwrite main configuration.
Those alerts which are marked in light gray color, are inherited from main configuration

INSTANCE ALERTS CONFIGURATION - PLEASE SELECT A DATABASE: EBAZY (1 alert/s overwritten)

Alert type	Alert description	Enabled	Override	Level value WARNING	Level value CRITICAL
Online	Alert if database is not available	□	☑		
Online	*Total Waits*	☑	□	*400 %*	*800 %*
Online	*Lock waits*	☑	□	*400 %*	*800 %*

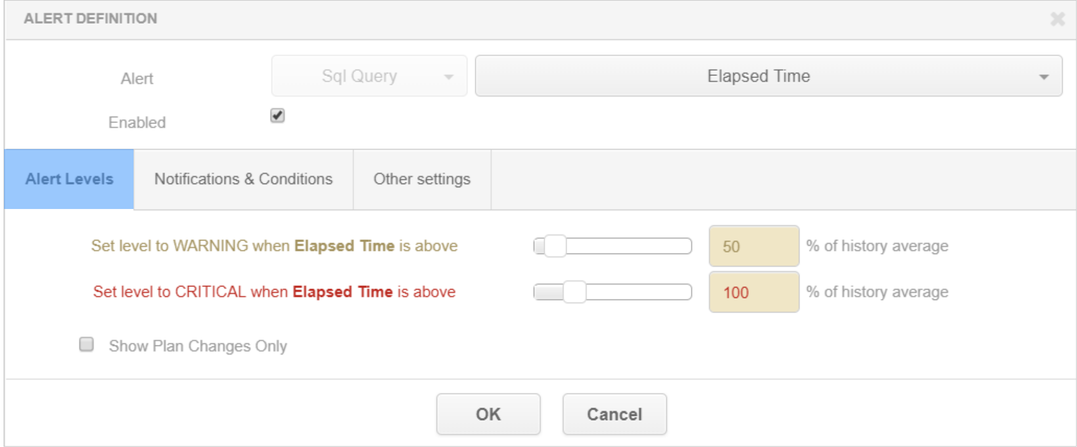
Anomaly Monitor - How does it work?

DBPLUS
better performance

Alert definition consists of:

Selecting the Alert Type:

- Online
- I/O Stats
- Load Trends
- SQL Query
- DB Size



ALERT DEFINITION

Alert: Sql Query | Elapsed Time

Enabled ☑

Set level to WARNING when **Elapsed Time** is above 50 % of history average

Set level to CRITICAL when **Elapsed Time** is above 100 % of history average

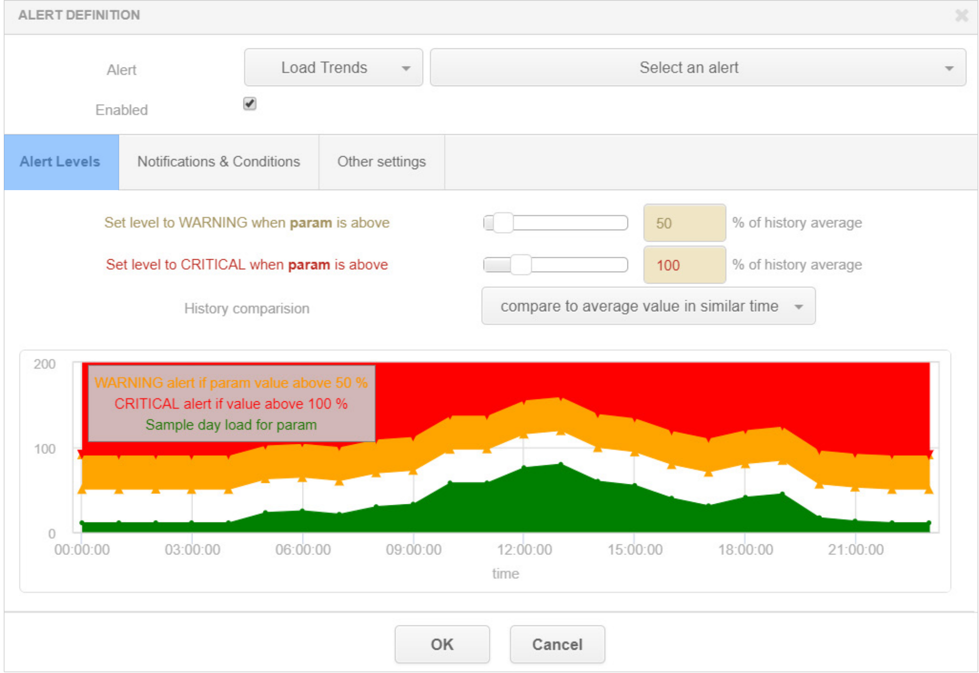
□ Show Plan Changes Only

Anomaly Monitor - How does it work?

DBPLUS
better performance

Determining the threshold value

- **WARNING**/**CRITICAL**



ALERT DEFINITION

Alert: Load Trends | Select an alert

Enabled ☑

Set level to WARNING when **param** is above 50 % of history average

Set level to CRITICAL when **param** is above 100 % of history average

History comparision: compare to average value in similar time

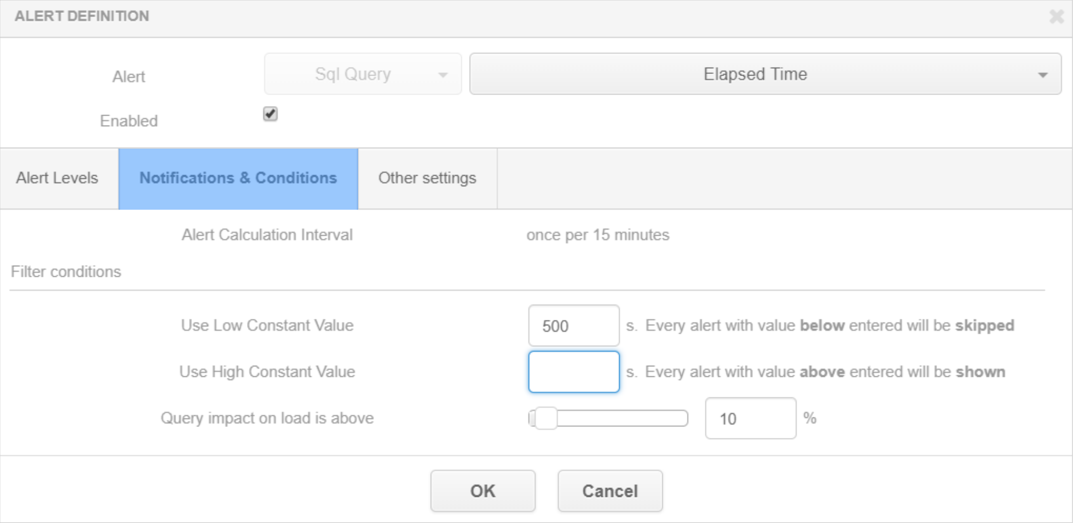
WARNING alert if param value above 50 %
CRITICAL alert if value above 100 %
Sample day load for param

Anomaly Monitor - How does it work?

DBPLUS
better performance

Setting additional conditions:

- Value below which the alert does not appear.
- Value above which the alert will always occur.
- What impact the query generates (only SQL Query).



ALERT DEFINITION

Alert: Sql Query | Elapsed Time

Enabled ☑

Alert Calculation Interval once per 15 minutes

Filter conditions

Use Low Constant Value 500 s. Every alert with value **below** entered will be **skipped**

Use High Constant Value s. Every alert with value **above** entered will be **shown**

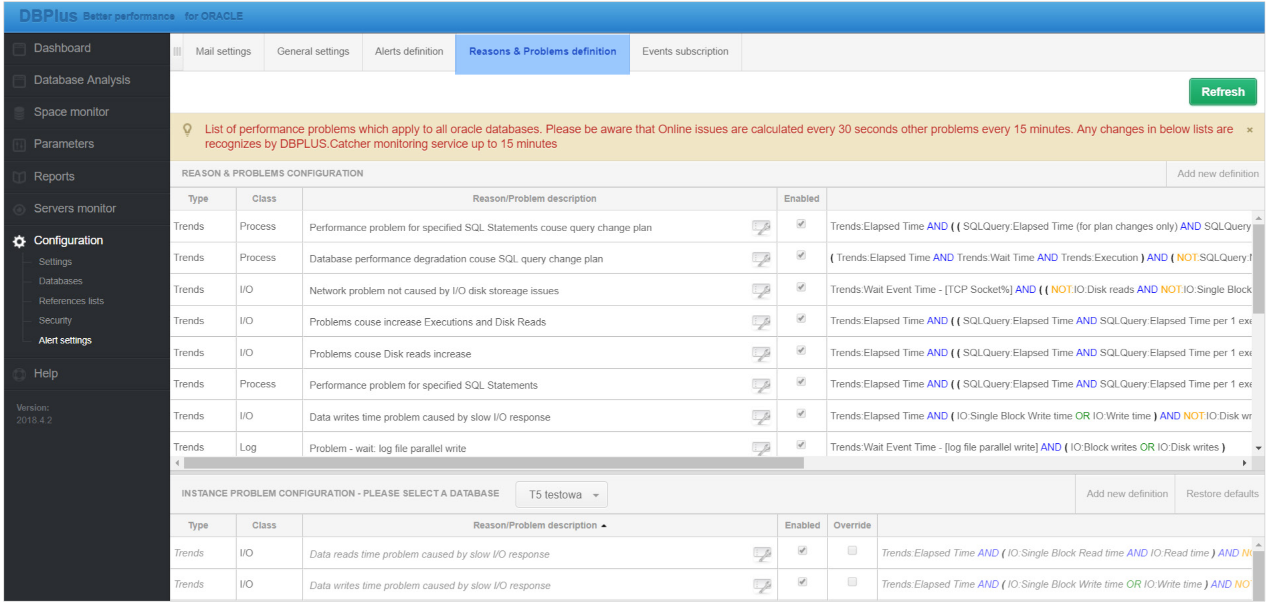
Query impact on load is above 10 %

Anomaly Monitor – Problem definitions

DBPLUS
better performance

Consist of:

- Giving the name of the problem
- Determining the class of the problem



List of performance problems which apply to all oracle databases. Please be aware that Online issues are calculated every 30 seconds other problems every 15 minutes. Any changes in below lists are recognizes by DBPLUS.Catcher monitoring service up to 15 minutes

REASON & PROBLEMS CONFIGURATION

Type	Class	Reason/Problem description	Enabled	
Trends	Process	Performance problem for specified SQL Statements couse query change plan	☑	Trends:Elapsed Time AND ((SQLQuery:Elapsed Time (for plan changes only) AND SQLQuery
Trends	Process	Database performance degradation couse SQL query change plan	☑	(Trends:Elapsed Time AND Trends:Wait Time AND Trends:Execution) AND (NOT:SQLQuery:
Trends	I/O	Network problem not caused by I/O disk storeage issues	☑	Trends:Wait Event Time - [TCP Socket%] AND ((NOT:IO:Disk reads AND NOT:IO:Single Block
Trends	I/O	Problems couse increase Executions and Disk Reads	☑	Trends:Elapsed Time AND ((SQLQuery:Elapsed Time AND SQLQuery:Elapsed Time per 1 exe
Trends	I/O	Problems couse Disk reads increase	☑	Trends:Elapsed Time AND ((SQLQuery:Elapsed Time AND SQLQuery:Elapsed Time per 1 exe
Trends	Process	Performance problem for specified SQL Statements	☑	Trends:Elapsed Time AND ((SQLQuery:Elapsed Time AND SQLQuery:Elapsed Time per 1 exe
Trends	I/O	Data writes time problem caused by slow I/O response	☑	Trends:Elapsed Time AND (IO:Single Block Write time OR IO:Write time) AND NOT:IO:Disk wr
Trends	Log	Problem - wait: log file parallel write	☑	Trends:Wait Event Time - [log file parallel write] AND (IO:Block writes OR IO:Disk writes)

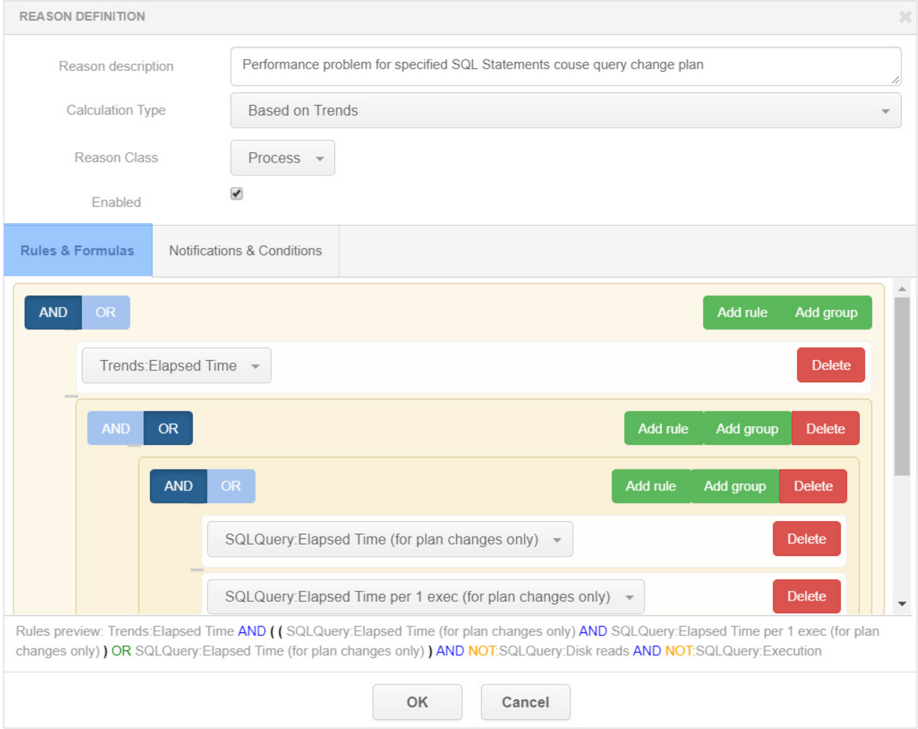
INSTANCE PROBLEM CONFIGURATION - PLEASE SELECT A DATABASE

Type	Class	Reason/Problem description	Enabled	Override	
Trends	I/O	*Data reads time problem caused by slow I/O response*	☑	□	*Trends:Elapsed Time AND (IO:Single Block Read time AND IO:Read time) AND N*
Trends	I/O	*Data writes time problem caused by slow I/O response*	☑	□	*Trends:Elapsed Time AND (IO:Single Block Write time OR IO:Write time) AND NO*

Anomaly Monitor – Problem definitions

Consist of:

- Setting up a set of rules based on Alerts.



REASON DEFINITION

Reason description: Performance problem for specified SQL Statements couse query change plan

Calculation Type: Based on Trends

Reason Class: Process

Enabled: ☑

Trends:Elapsed Time

SQLQuery:Elapsed Time (for plan changes only)

SQLQuery:Elapsed Time per 1 exec (for plan changes only)

Rules preview: Trends:Elapsed Time AND ((SQLQuery:Elapsed Time (for plan changes only) AND SQLQuery:Elapsed Time per 1 exec (for plan changes only)) OR SQLQuery:Elapsed Time (for plan changes only)) AND NOT:SQLQuery:Disk reads AND NOT:SQLQuery:Execution

Access management

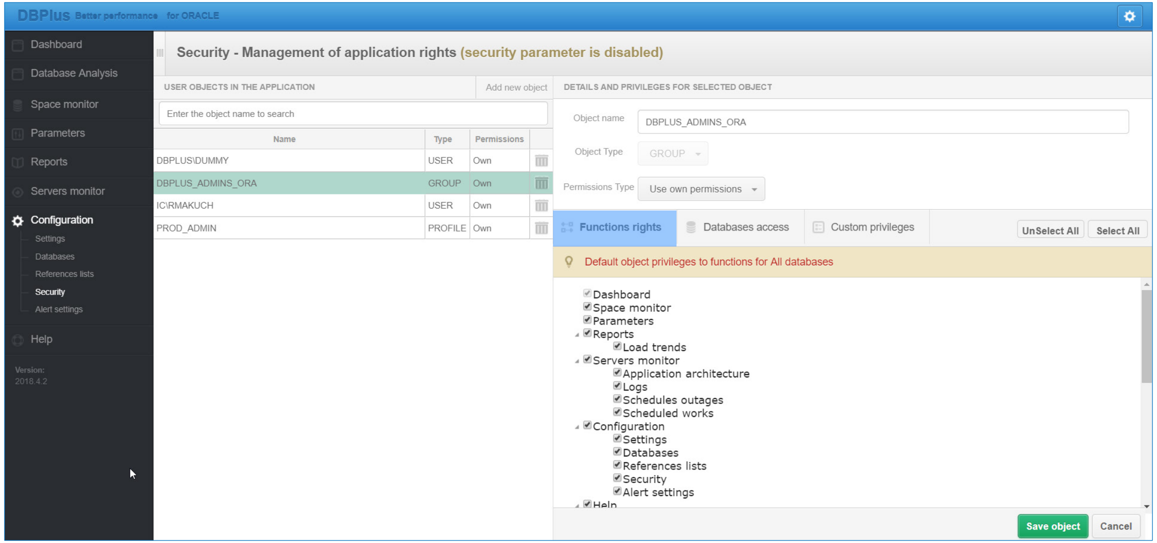
It is possible to grant access to individual databases and the screens in the application.

Setting access for:

- ***USER***
  (Object name: DOMAIN\USER)

GROUP:

- ***Local***
  (Object name: GROUP NAME)
- ***Domain***
  (Object name: DOMAIN\USER)
- ***PROFILE***
  (Object name: PROFILE NAME).



Security - Management of application rights (security parameter is disabled)

USER OBJECTS IN THE APPLICATION

Name	Type	Permissions
DBPLUS\DUMMY	USER	Own
DBPLUS_ADMINS_ORA	GROUP	Own
IC\RMAKUCH	USER	Own
PROD_ADMIN	PROFILE	Own

DETAILS AND PRIVILEGES FOR SELECTED OBJECT

Object name: DBPLUS_ADMINS_ORA

Object Type: GROUP

Permissions Type: Use own permissions

Default object privileges to functions for All databases

- ☑ Dashboard
- ☑ Space monitor
- ☑ Parameters
- ☑ Reports
  - ☑ Load trends
- ☑ Servers monitor
  - ☑ Application architecture
  - ☑ Logs
  - ☑ Schedules outages
  - ☑ Scheduled works
- ☑ Configuration
  - ☑ Settings
  - ☑ Databases
  - ☑ References lists
  - ☑ Security
  - ☑ Alert settings
- ☑ Help

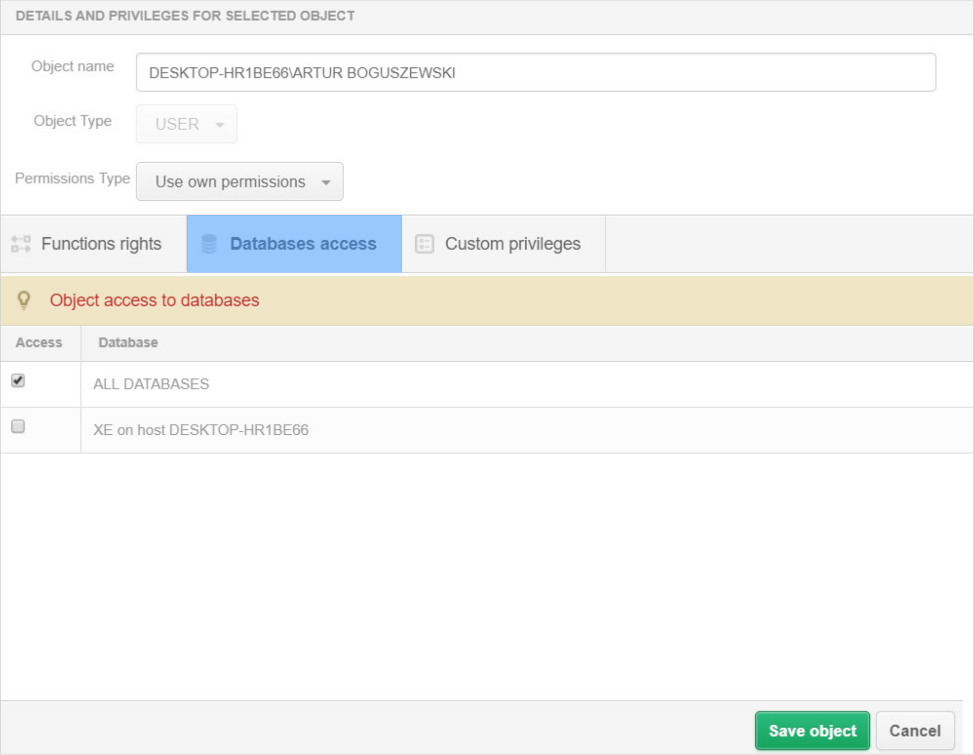
It is possible to configure rights:

- own permissions (use Own permissions)
- inherited permissions (Inherited permissions).

Access management

DBPLUS
better performance

- Own permissions (Use own permissions).
- This type of permission can be granted for each of the three objects (USER, GROUP, PROFILE).
- We assign permissions to individual functionalities (Function rights)
- Permissions for individual databases (Database access)
- Local privillages (Local privillages).



DETAILS AND PRIVILEGES FOR SELECTED OBJECT

Object name: DESKTOP-HR1BE66\ARTUR BOGUSZEWSKI

Object Type: USER

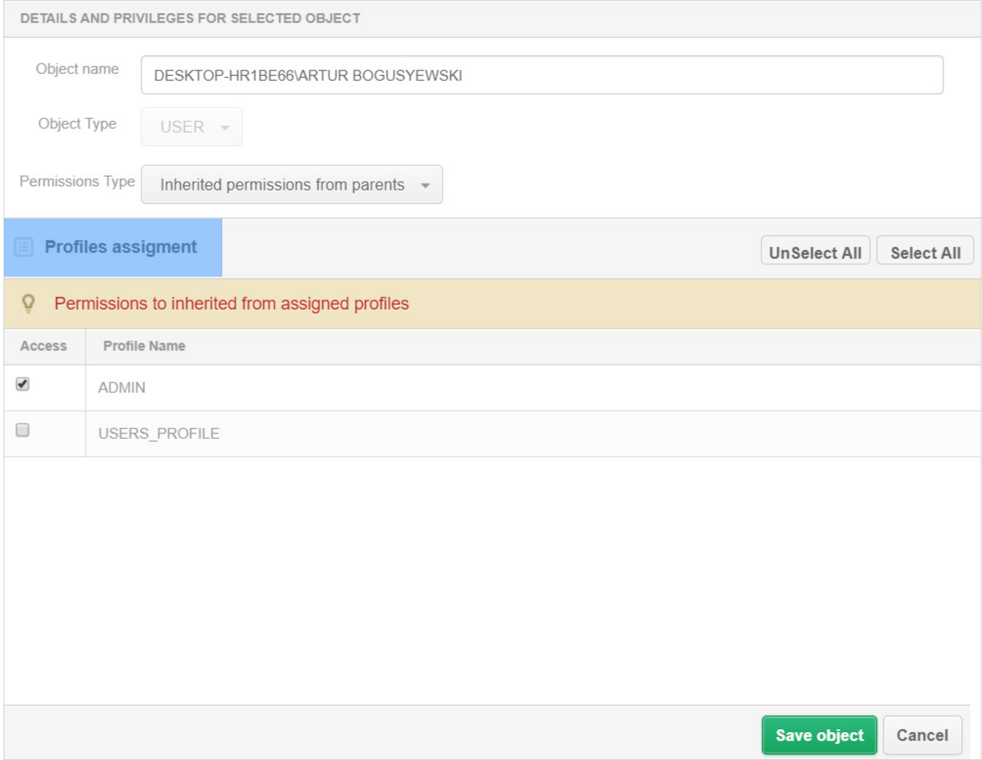
Permissions Type: Use own permissions

Object access to databases

Access	Database
☑	ALL DATABASES
☐	XE on host DESKTOP-HR1BE66

Access management

- Inherited permissions from parents
- This type of permission can be granted for each of the three objects (USER, GROUP, PROFILE).
- When assigning permissions, we always point to the PROFILE for which we have previously defined the permissions.



DETAILS AND PRIVILEGES FOR SELECTED OBJECT

Object name: DESKTOP-HR1BE66\ARTUR BOGUSYEWSKI

Object Type: USER

Permissions Type: Inherited permissions from parents

Profiles assigment

Permissions to inherited from assigned profiles

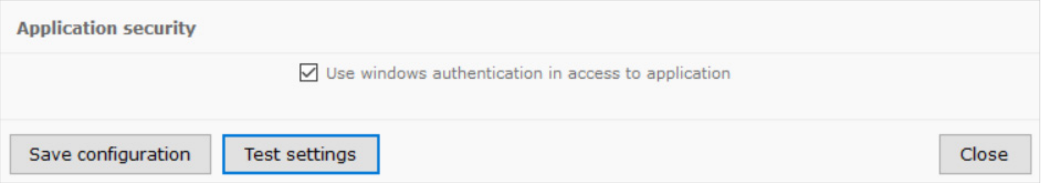
Access	Profile Name
☑	ADMIN
□	USERS_PROFILE

Access management

Access management is set on two levels:

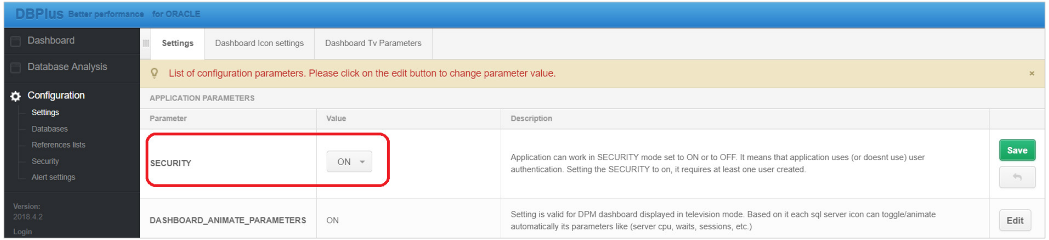
- DBPLUS Configuration Wizard: Applications settings > Application Options > Configure

- DBPLUS Performance Monitor: Configuration > Settings > SECURITY parameter



Application security

☑ Use windows authentication in access to application



List of configuration parameters. Please click on the edit button to change parameter value.

APPLICATION PARAMETERS

Parameter	Value	Description	
SECURITY	ON	Application can work in SECURITY mode set to ON or to OFF. It means that application uses (or doesnt use) user authentication. Setting the SECURITY to on, it requires at least one user created.	Save
DASHBOARD_ANIMATE_PARAMETERS	ON	Setting is valid for DPM dashboard displayed in television mode. Based on it each sql server icon can toggle/animate automatically its parameters like (server cpu, waits, sessions, etc.)	Edit

DBPLUS

better performance

Thank you

www.dbplus.tech