

DBPLUS™
better performance

DBPLUS Performance Monitor™ for Oracle®
User's Manual

March 2020

Table of contents

1	Introduction	4
1.1	DBPLUS Technical Support	5
1.2	System architecture	5
1.3	System requirements	6
1.4	Installation of DBPLUS Performance Monitor	7
2	System Configuration	11
2.1	Main configurator screen.....	11
2.2	Setting up DBPLUSORACLECATCHER monitoring service	14
2.3	System Repository configuration.....	15
2.3.1	The database name for the database repository	15
2.3.2	Establishing a connection using the TNS file	16
2.3.3	Database repository parameters	16
2.4	IIS service configuration.....	18
2.4.1	Configuration of SSL in the IIS environment	19
2.5	User application configuration	23
2.6	Configuration summary	25
3	Adding a database for monitoring	28
4	System Upgrade	33
4.1	Setting up for the latest version.....	33
5	License	36
6	Working with program.....	37
6.1	Menu „Dashboard“	37
6.1.1	Information Bar	37
6.1.2	The summary area	38
6.1.3	Servers and databases area	39
6.1.4	Details of ORACLE database performance.....	41
6.1.5	Dashboard – various forms of presentation	46
6.2	Menu Database Analysis	48
6.2.1	Menu Performance – Database Analysis	48
6.2.2	Menu Plan explorer	97
6.2.3	Menu Anomaly Monitor	99
6.2.4	Menu I/O Stats - Database Analysis	103
6.2.5	Menu Space Monitor - Database Analysis	109
6.2.6	Menu Memory - Database Analysis	113
6.2.7	Menu Sessions – Database Analysis.....	120
6.2.8	Menu Backups - Database Analysis	127
6.2.9	Menu Locks - Database Analysis	129
6.2.10	Menu Parameters - Database Analysis.....	131
6.2.11	Menu Logs - Database Analysis	131
6.2.12	Menu Reports - Database Analysis	135
6.3	Menu Space Monitor.....	136
6.4	Menu Parameters	136
6.5	Menu Reports – Load Trends.....	137
6.6	Menu Servers Monitor.....	138
6.6.1	Application architecture	138
6.6.2	Logs.....	139
6.7	Menu Configuration	140
6.7.1	Settings	140
6.7.2	Databases	141
6.7.3	Reference lists	143

6.7.4	Security.....	143
6.7.5	Alert settings	147
6.7.6	Outages setting.....	163
6.7.7	Time line settings	166
6.8	Additional functionalities	168
6.8.1	RAC support	168
6.8.2	Standby DataGuard support.....	170
6.8.3	Export / Import statistics of monitored databases.....	170

1 Introduction

What is DBPLUS Performance Monitor?

DBPLUS Performance Monitor™ tool is the software used for monitoring and analyzing the Oracle database performance

Using DBPLUS Performance Monitor, you can:

- observe the current database performance,
- track trends of database server load and the individual components: CPU, I / O buffers
- identify performance issues of Oracle databases
- track performance trends of individual SQL queries
- analyze data and present them in graphical form
- watch in real time active user sessions
- observe the status of full and incremental databases backups
- troubleshoot a non-optimal SQL queries
- legibly report database problems

and many, many more

Question:

"Why do database work too slow in any specified period of time?"

will never be left without an answer!

1.1 DBPLUS Technical Support

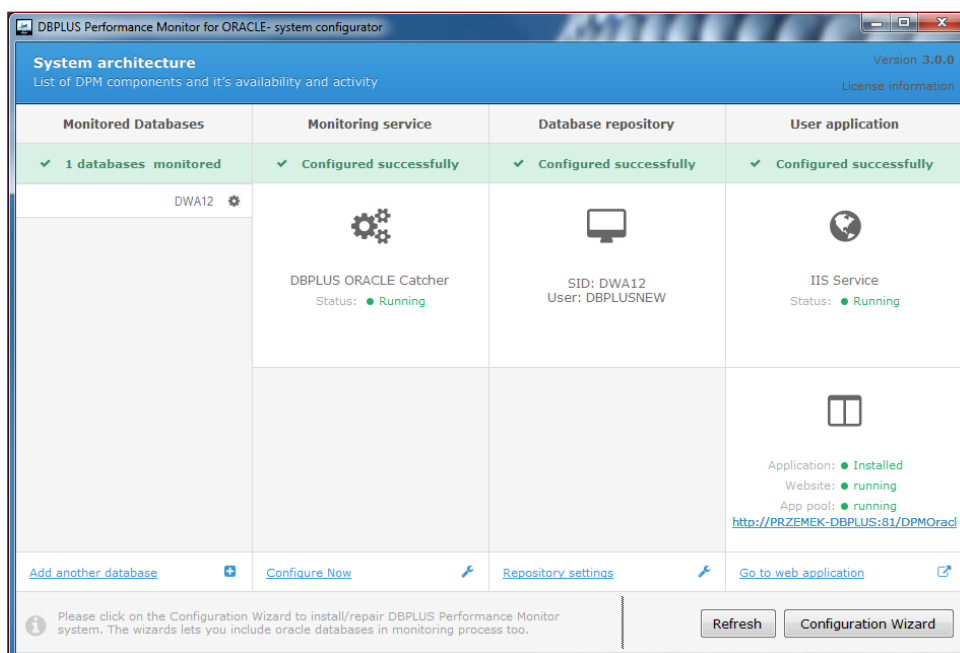
Technical support provides the access to new software updates published 4 times a year as well as to engineers' - help in DBPLUS Oracle database diagnosis (by using **DBPLUS Performance Monitor** software).

1.2 System architecture

The system is designed in client-server architecture and in the presented solution we can distinguish the following components:

- **Databases** - a list of Oracle databases covered by the monitoring,
- **Server program** - an application running as a windows service, which consists of a set of procedures performed on individual databases. The aim of the program is to run periodically procedures, which are responsible for collecting basic data about Oracle servers' performance. According to the DBPLUS nomenclature, program is called **DBPLUSORACLECATCHER** and one-up cycle within the service **DBPLUSORACLECATCHER** is called "a snap".
- **Repository** - selected database that stores performance statistics of monitored databases. Collected statistics are the result of the work of **DBPLUSORACLECATCHER** service.
- **Application** - this is a client of the system, which implements user interface which allows to implement functionality of the system, i.e. monitoring review, performance analysis, query execution statistics reports, the current sessions of database, chart of server load, etc. The application is made in web technology using IIS application server and it is accessible from a web browser.

DBPLUS Performance Monitor requires the installation and configuration of each of the elements to ensure full functionality of the solution. Below we present a general model of the system:



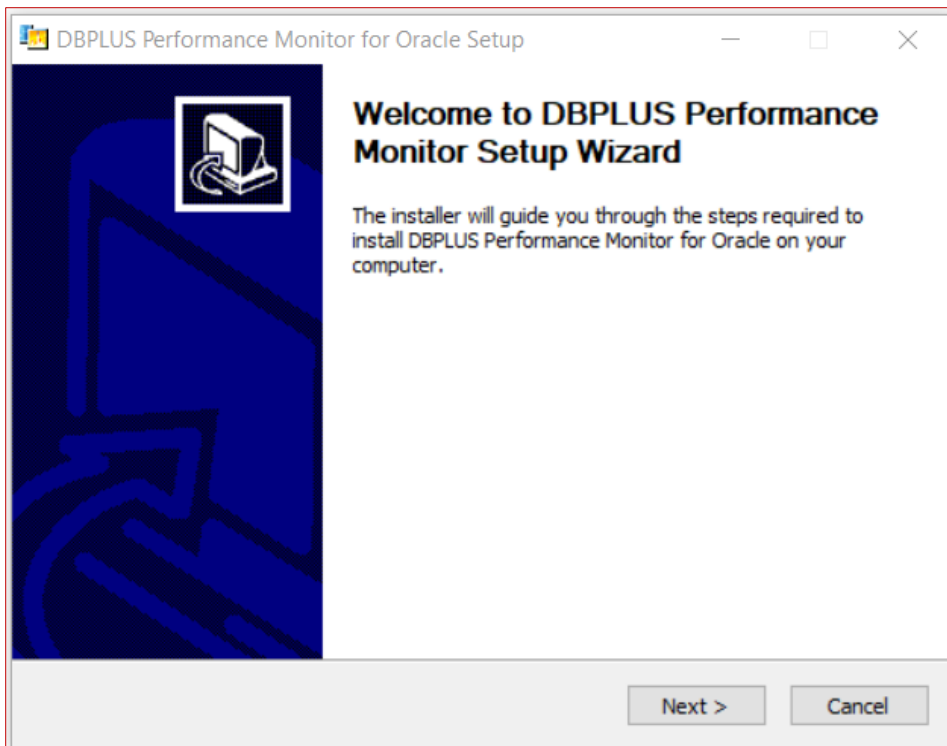
IMPORTANT: DBPLUS System Performance Monitor requires the installation and configuration on any given server / computer in the company. During normal use of application, system does not require any installation on the user's local computers.

1.3 System requirements

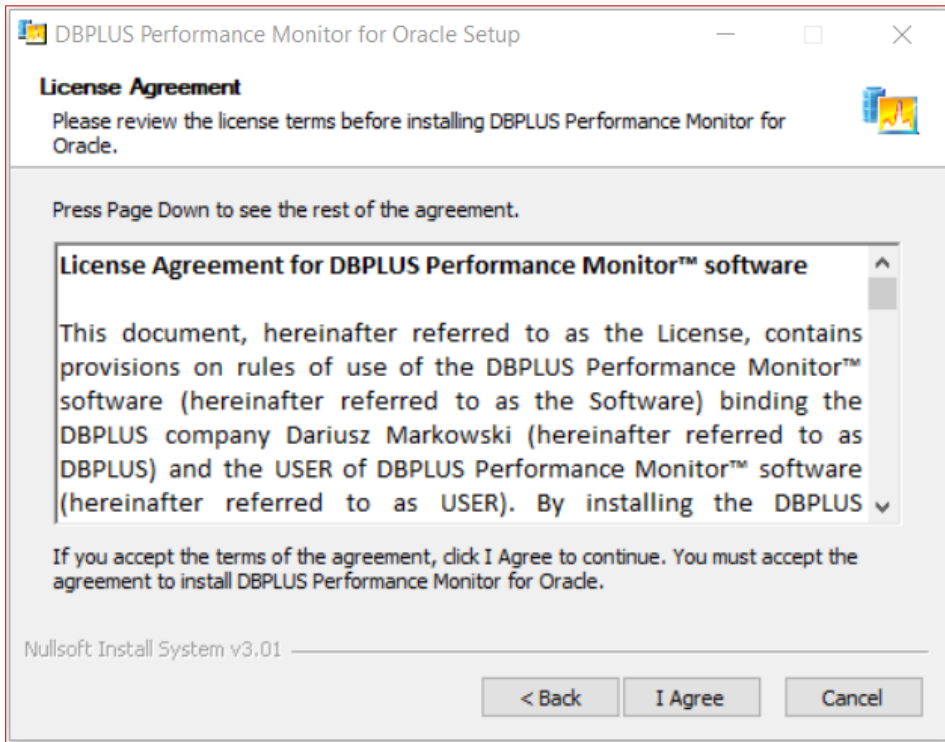
Parameter	Description
Monitored Oracle database	Supported types of monitored SQL database: Monitored all versions of Oracle database from version 10.2 and above
Server operating system with installed DBPLUS PERFORMANCE MONITOR software	Servers: <ul style="list-style-type: none"> • Windows Server 2008 and above Also: <ul style="list-style-type: none"> • Windows 7 and above Additional requirements: <ul style="list-style-type: none"> • .NET Framework 4.0 (for DBPLUSORACLECATCHER service) • .NET Framework 4.0 (for the client application). Scale and layout: Screen resolution: 800x600 or higher Text size 100% On the server / computer with DBPLUS Performance Monitor software is not required to install Oracle components.
Server's hardware requirements with installed DBPLUS PERFORMANCE MONITOR software	<ul style="list-style-type: none"> • 4 CPU • 8 GB of RAM • HD – no requirements When monitoring 20 instances: <ul style="list-style-type: none"> • DBPLUSCATCHER Monitoring Service consumes at a level 2 GB, IIS to 500 MB of RAM • Assign 4 CPU due to the multithreading services, monitoring a number of instances, plus user applications. • DBPLUS Software is 30 MB, so no special requirements as to the size of the drives on the machine • Repository database size for one database is 10GB.
The impact of the system to Oracle servers	The system generates an average load of less than 1% dependent on generally accepted "quality" of databases Repository database: As a result of the installation of repository on a selected database, the system sets up: <ul style="list-style-type: none"> • DBPLUS scheme objects - tables <ul style="list-style-type: none"> ○ User functions with permissions that allow to read the views of systems. Warning! DBPLUS database user does not have permission to read data from the database scheme of other users. Monitored database: As a result of inclusion in the monitoring process a specific database it is set up the user used only to connect and data collection with a given database.
User interface	The user application is accessible from a web browser. Supported browsers include: <ul style="list-style-type: none"> • Internet Explorer (ver. 9 and above) • Google Chrome • Mozilla Firefox • Opera

1.4 Installation of DBPLUS Performance Monitor

DBPLUS Performance Monitor is available on DBPLUS server through the provided link. User can install DBPLUS Performance Monitor by double-clicking downloaded EXE file:

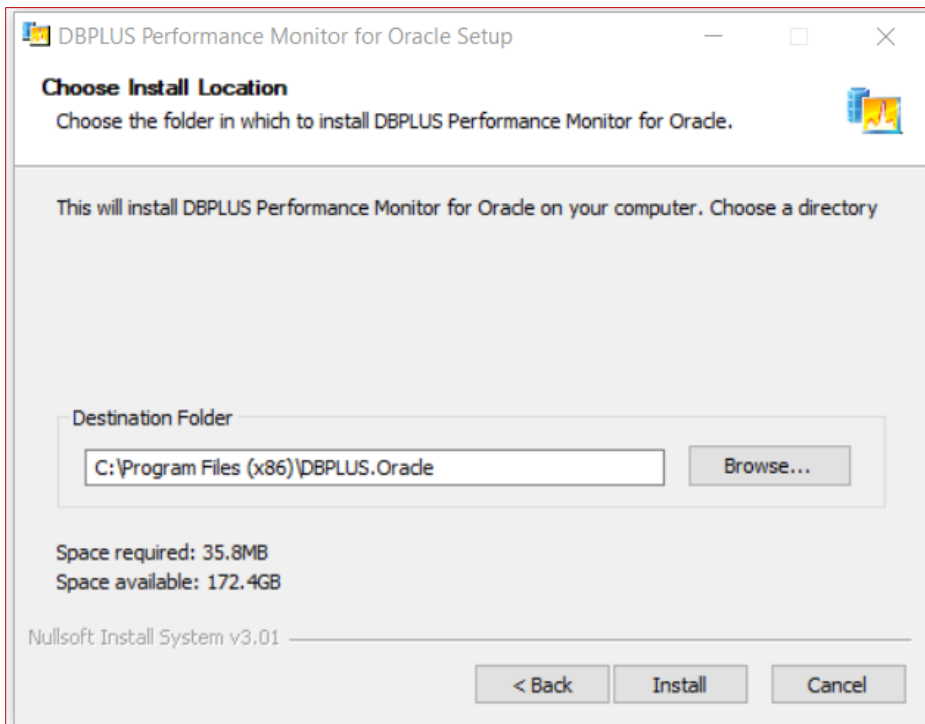


By clicking [Next] button User get information about the license agreement:

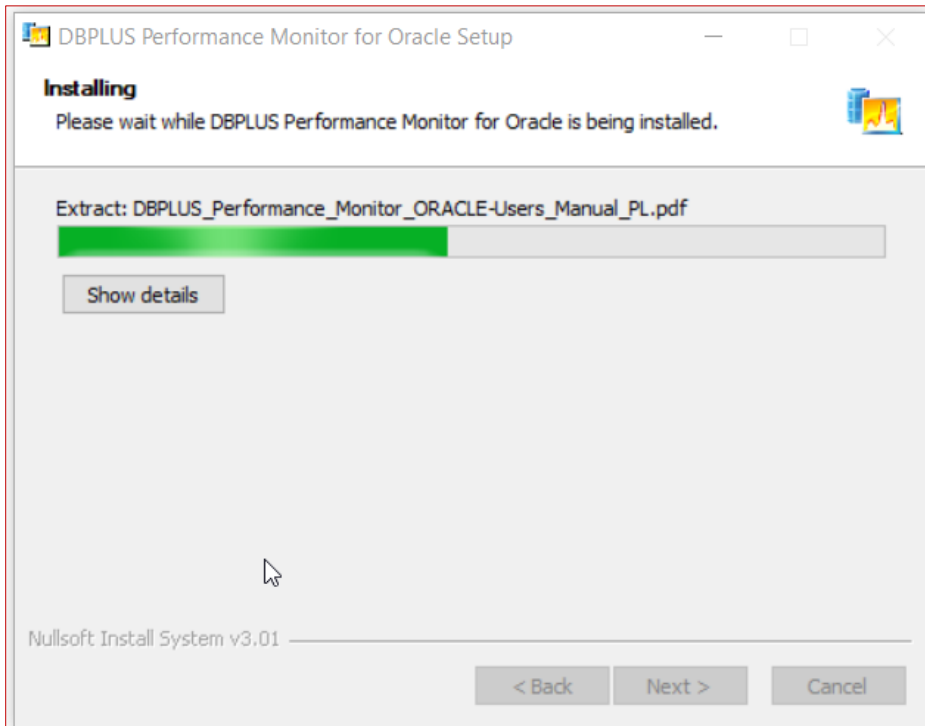


In order to continue installation, you should read and accept the terms of the license. The next step is to select the directory, where DBPLUS Performance Monitor will be installed

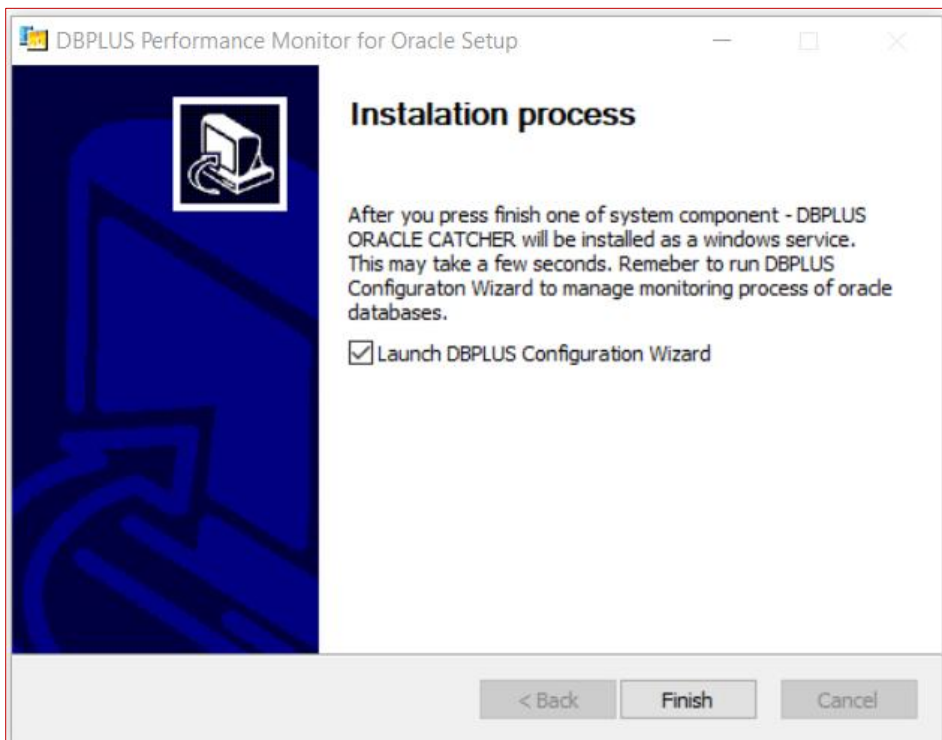
Default directory is C:\Program Files (x86)\DBPLUS.Oracle:



Visible progress of the installation process:



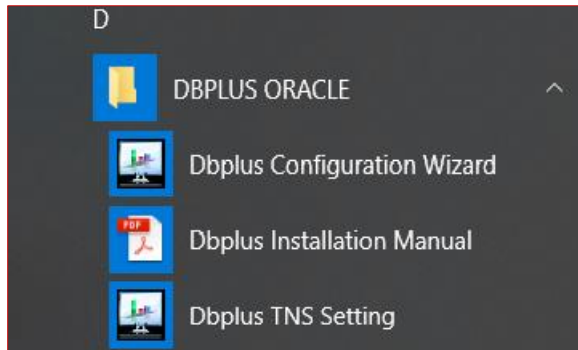
Finally after the installation User will receive the following information:



The installation process is completed by pressing "Finish" button. By default, we start system configurator, which will carry out the further process of installing individual components of the system or configuration.

Installed DBPLUS Performance Monitor is in the menu:

„Start” -> „Programs”->”DBPLUS ORACLE”



The following tools are available after the correct installation

1. DBPLUS Configuration Wizard
2. DBPLUS TNS Setting

2 System Configuration

In the first stage you must set up a system on the server with **DBPLUS Performance Monitor™** installed, in order to:

- Create a DBPLUS database repository in the selected Oracle database, which will store all the information about Oracle databases performance,
- Inclusion Oracle servers in the monitoring process,
- Configuration monitoring service DBPLUSORACLECATCHER responsible for gathering information about individual servers' performance,
- User Application Configuration

For performing these tasks, system requires permissions:

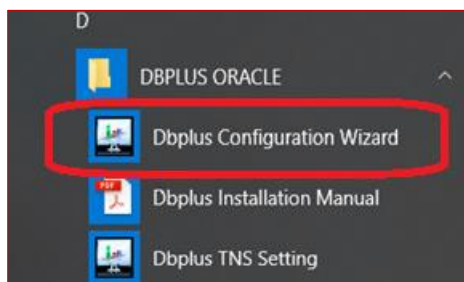
SYS or SYSTEM user or user with DBA privileges. This is required in the first step in which the Repository database is configured. User indicate one of the Oracle databases, which will be created a new DBPLUS database user and created technical tables in the same database schema.

As part of adding a database for monitoring on a monitored basis, a database user is created. This user is responsible for collecting statistics on the monitored database.

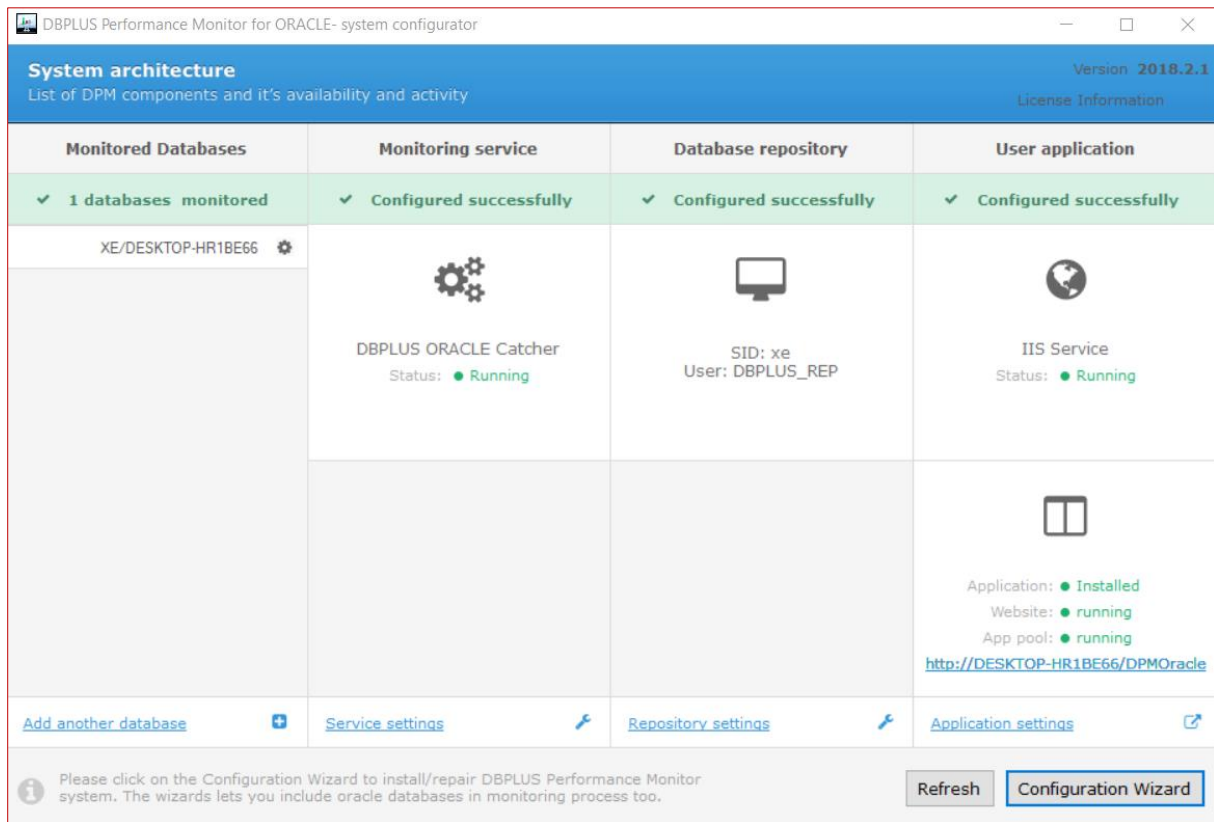
After completing the above steps, the application will be available to the user from the level of the web browser.

2.1 Main configurator screen

On the server, where software has been installed, by clicking „Start”→”Programs”→ DBPLUS ORACLE→„DBPLUS Configuration Wizard”



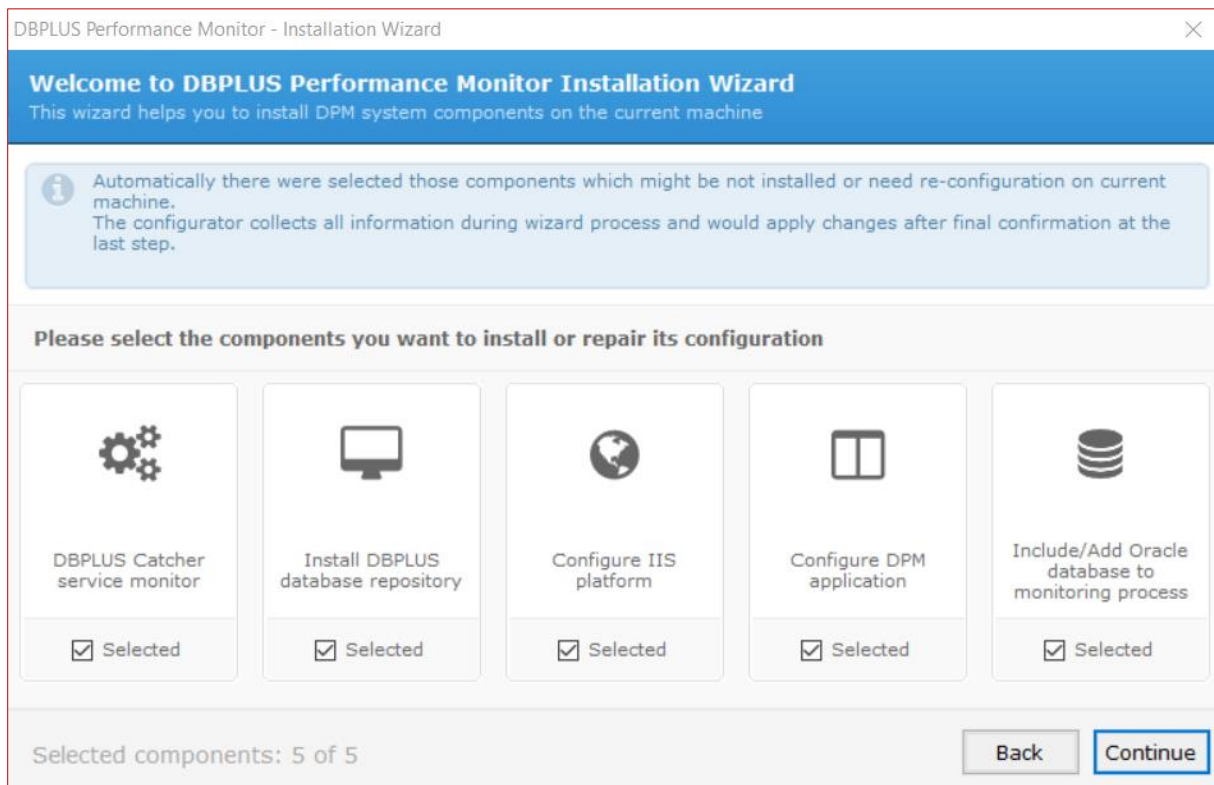
User open a window with system management tool:



The main window shows the system architecture and informs among others about:

- number of monitored databases
- place in which monitoring data is stored (Database Repository),
- installation / configuration of individual components of system, for example:
 - lack of monitored Oracle databases,
 - user application installed or not, application services (IIS website, application pool) running or not,
 - if the monitoring service is enabled.

In order to perform basic system configuration, click [**Configuration Wizard**] button and - as a result - we get this screen to configure individual components.



By default, system selects components that require configuration. You can always reconfigure e.g. a monitoring service or add another (not included so far) SQL database to monitoring.

In the initial stage:

- We create the DBPLUS database repository
- We include / IIS role/service on the current machine
- We configure DBPLUSORACLECATCHER monitoring
- We configure user application

2.2 Setting up DBPLUSORACLECATCHER monitoring service

DBPLUSORACLECATCHER is a program that runs as a Windows service. In the current version this service can operate using a local account.

Service configuration screen is presented below:

DBPLUS Performance Monitor - Installation Wizard

DBPLUS ORACLE Catcher - windows service responsible for oracle databases monitoring
Specify if service should be ran in context of windows/domain account or using local system account

Catcher	Repository	IIS	App	Finish

i For DBPLUSORACLECATCHER service it can be used:
 - Local system account
 - Windows/Domain account.
 On the database level system uses only internal oracle users.
 Remarks:
 - Please do not use account with administrator privileges, it's not required

Set an user account which will be used by the DBPLUSORACLECATCHER service

Login type: Local System Account

Username:

Password:

Step 1 from 6

Back Continue

Click on the **[Continue]** button to advance to the next configuration item.

IMPORTANT NOTICE: All settings - made in the various components of system - are ultimately confirmed in the final step of the creator.

2.3 System Repository configuration

DBPLUS Performance Monitor System Repository is a database that must be created on the selected Oracle database. To do this, you have to enter among others.

- Database name
- User name (from which the installation will be carried out) - it is mostly a system or sys - the name shall be given only once during the installation process and it is not subsequently used or stored anywhere.
- Login/user name that will be used to connect to a given database – for this user will occur subsequent connection to database repository. This user does not have permission to read data from schemes other than system dictionaries (owner: SYS or SYSTEM)

2.3.1 The database name for the database repository

In the first step, system asks for the basic information:

- The name of Oracle database - you can login using TNS or by entering the full server data without using TNS (connection type: Basic)
- Determining a user account with sufficient permissions to create a new user that will be used to complete the installation database repository.

DBPLUS Performance Monitor - Installation Wizard
✕

DBPLUS database repository
Specify database where repository user can be installed

Catcher	Repository	IIS	App	Finish

i You need to specify the database server where dbplus repository would be located.
Database details like name, files and any specific features you can select in the next following steps

Connection Type TNS

SID

Set an user account with administrator rights.
It will be used to perform DPM monitoring objects instalation on selected database

Authentication Oracle Authentication Role Default

Username

Password

Test credentials

Step 2 from 6
Back
Continue

2.3.2 *Establishing a connection using the TNS file*

If you choose to authenticate via TNS, the system also verifies how many Oracle clients are installed on the machine with the DBPLUS Performance Monitor software and gives the user the option to select the appropriate tnsnames.ora file. This setting is made from the Program-> DBPLUS ORACLE-> Dbplus TNS Setting program.

IMPORTANT NOTICE: After each change of the path from which the tnsnames.ora file is read:

- restart the DBPLUS Configuration Wizard - restart the DBPLUS Catcher monitoring service.
- verify whether there is an error in monitoring specific databases, i.a. due to the lack of connectivity.

2.3.3 *Database repository parameters*

The next step is to provide information about database, such as:

- Name of the tablespace in which data will be kept - failure to provide the name will cause that data will be kept in the default tablespace of created / selected user
- The path to data file
- Initial file sizes

Then specify the account details that will be used to connect to the repository database. You can indicate an existing user or create a new one by entering login and password.

DBPLUS Performance Monitor - Installation Wizard
✕

DBPLUS database repository
Specify an account which will be used by DBPLUSORACLECATCHER service and user application to connect to database

Catcher	Repository	IIS	App	Finish

i You need to specify the user which will be used for connection purposes by DBPLUSORACLECATCHER service and DBPLUS Performance Monitor application
We strongly advise to use separate tablespace and user for Dbplus objects, but you can also use existing one(s).
For new tablespace/user, please specify its name(s), file location, password

Create new user and select tablespace

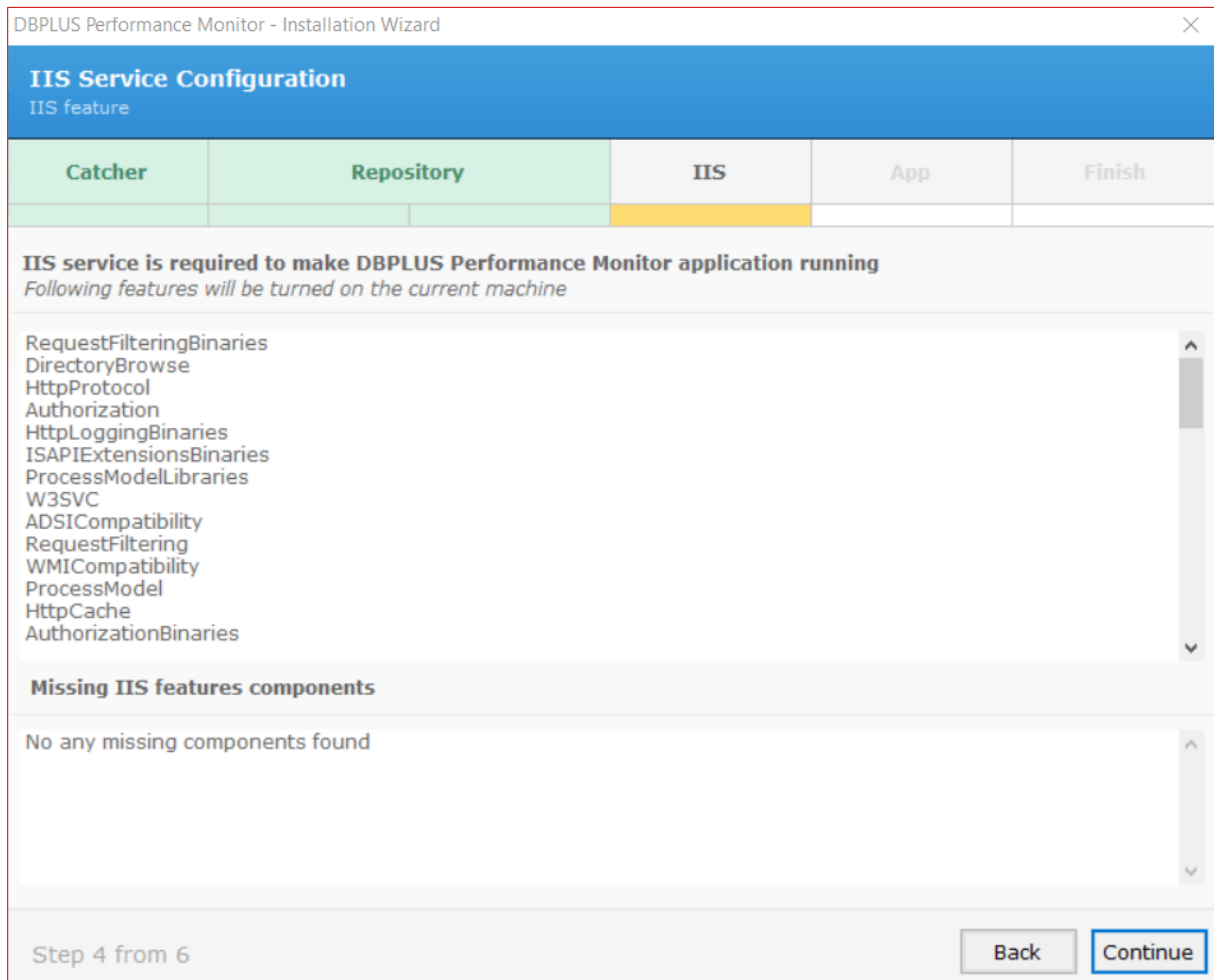
Authentication	Oracle authentication	Tablespace name	DBPLUS_R
User name	DBPLUS_REP	Data file	C:\ORACLEXE\APP\ORACLE\ORADATA\XE\DBPLUS_R
Password	••••••••	Initial size for data file	1000 MB
Profile	DEFAULT	Temp Tablespace	TEMP

<p>Use existing user</p> <p><input type="checkbox"/> Use existing user</p> <p>User name: DBPLUS</p> <p>Password: <input type="password"/></p> <p><input type="button" value="Test credentials"/></p>	<p>Use existing tablespace</p> <p><input type="checkbox"/> Do NOT create new Tablespace</p> <p>Existing Tablespace: DBPLUS</p>
---	---

Step 3 from 6

2.4 IIS service configuration

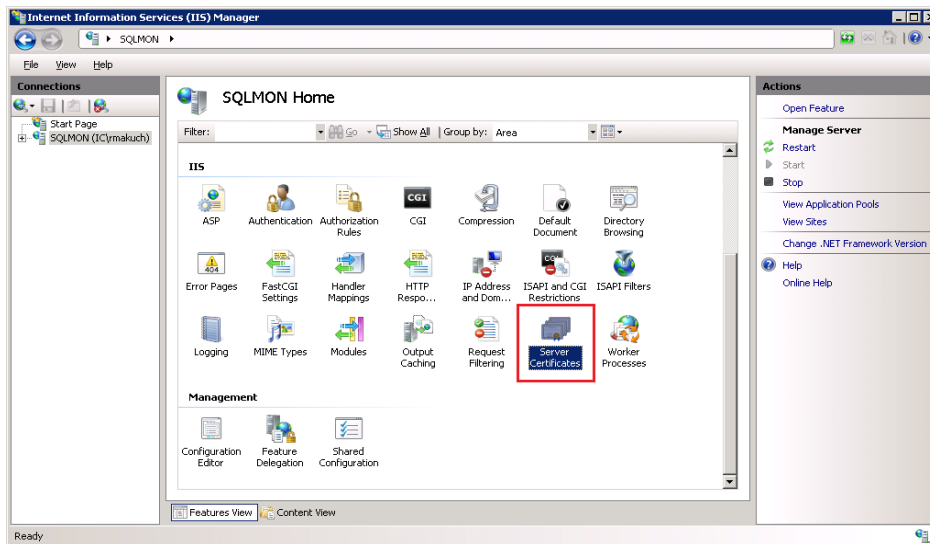
Launching of the IIS role / function on the server is required to run the user interface. The creator window announces additions / services of IIS application server that will be installed. If the “**Missing IIS features components**” box is empty, no configuration is required.



2.4.1 Configuration of SSL in the IIS environment

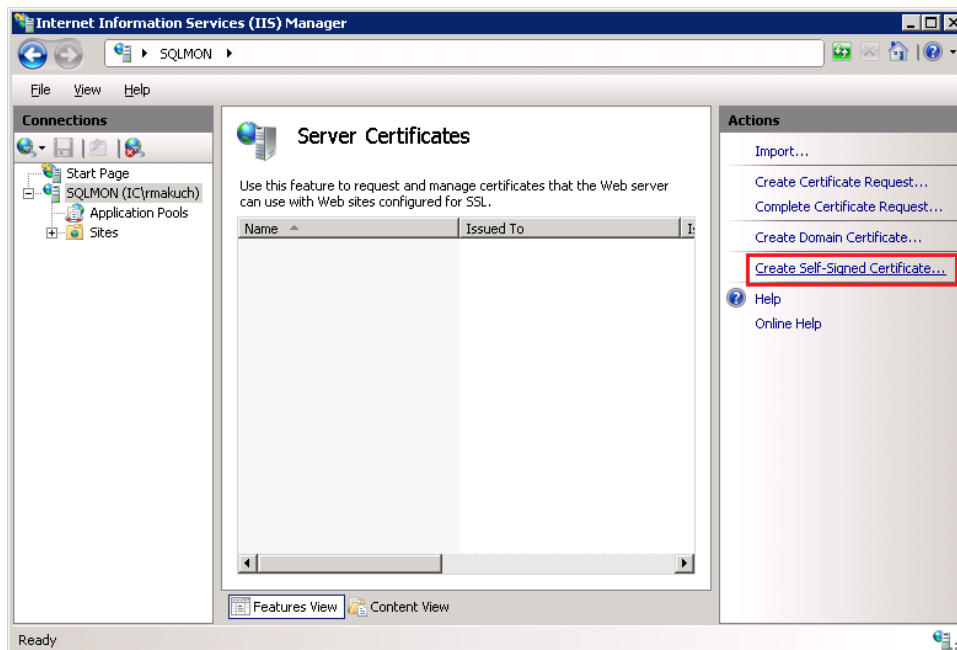
In case you want to enable the SSL functions in the DBPLUS Performance Monitor application, you need to perform the steps on the server with the installed DBPLUS software:

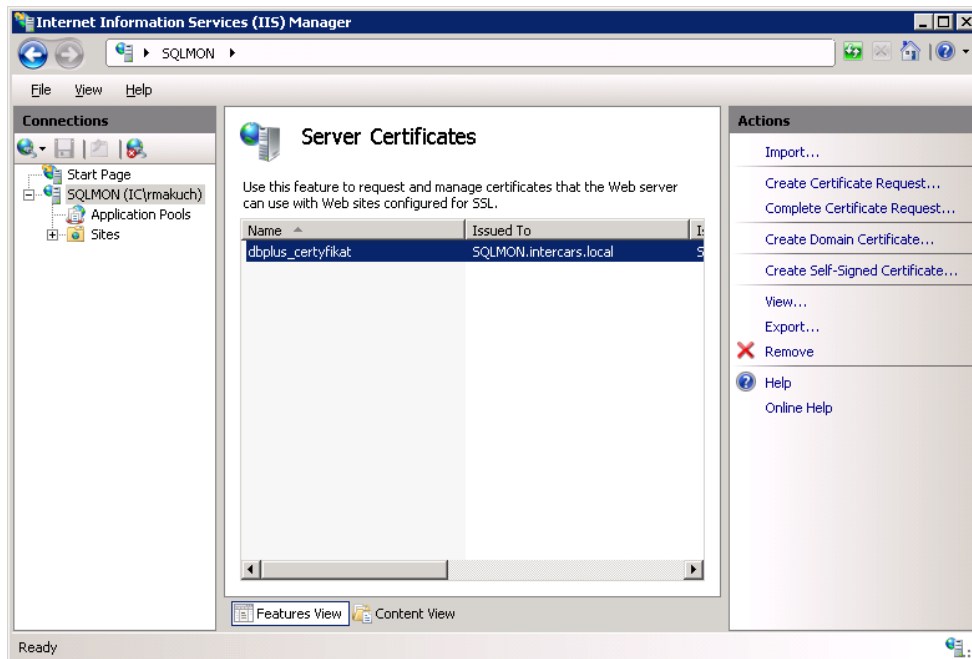
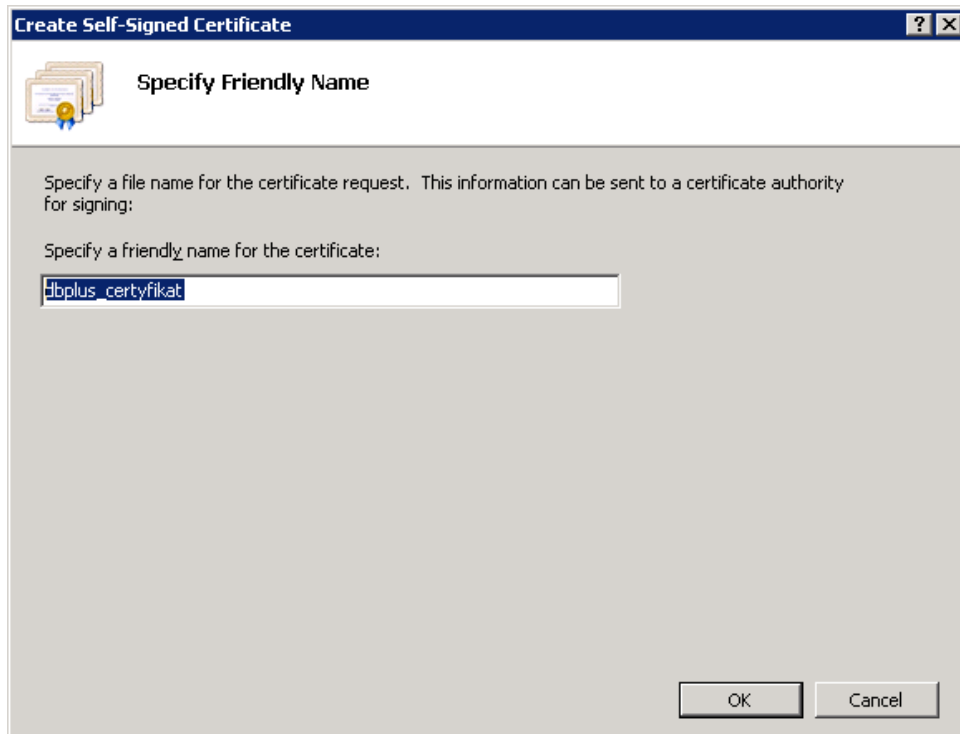
1. Run the IIS Manager (Internet Information Manager) from the command line with the **inetmgr** command
2. For the selected server, find the Server Certificates icon and enter to generate or import a certificate



3. Generation of the certificate on the IIS server (in case we do not have it)

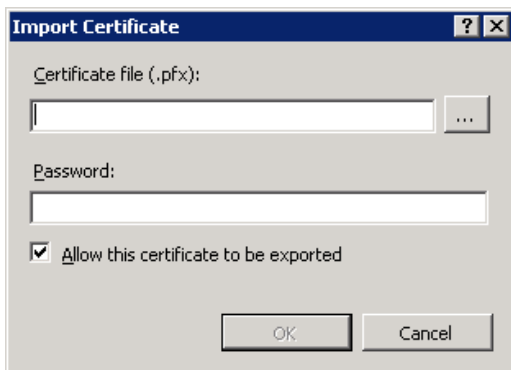
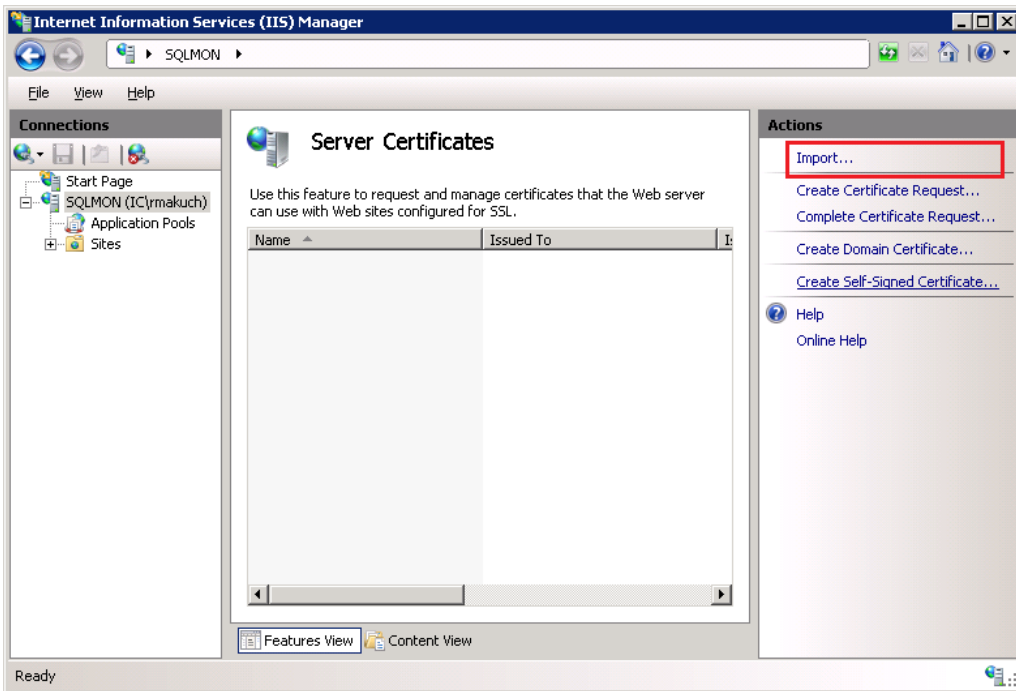
We run options according to the below screenshots





4. Certificates import (in case the certificate was not generated directly on the IIS server)

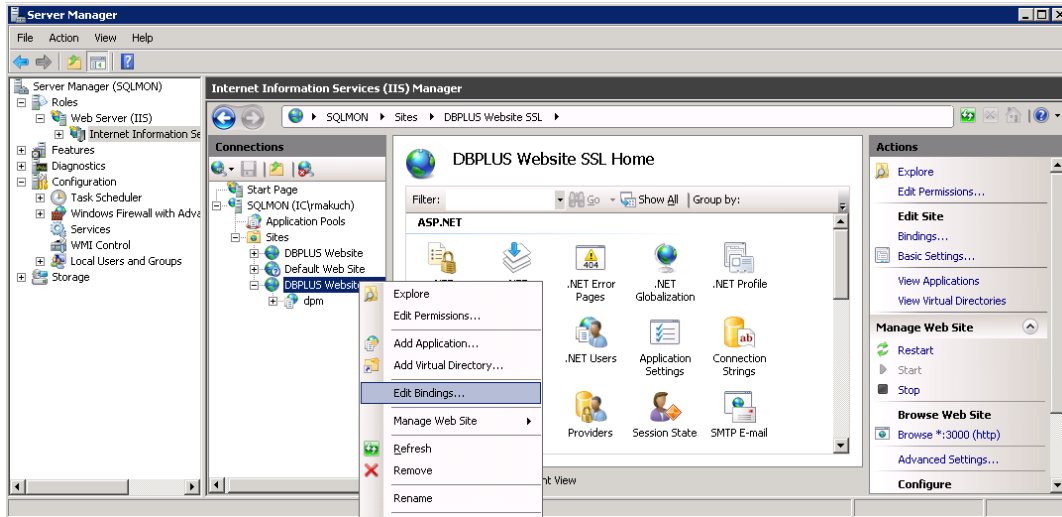
We run according to the following screens:



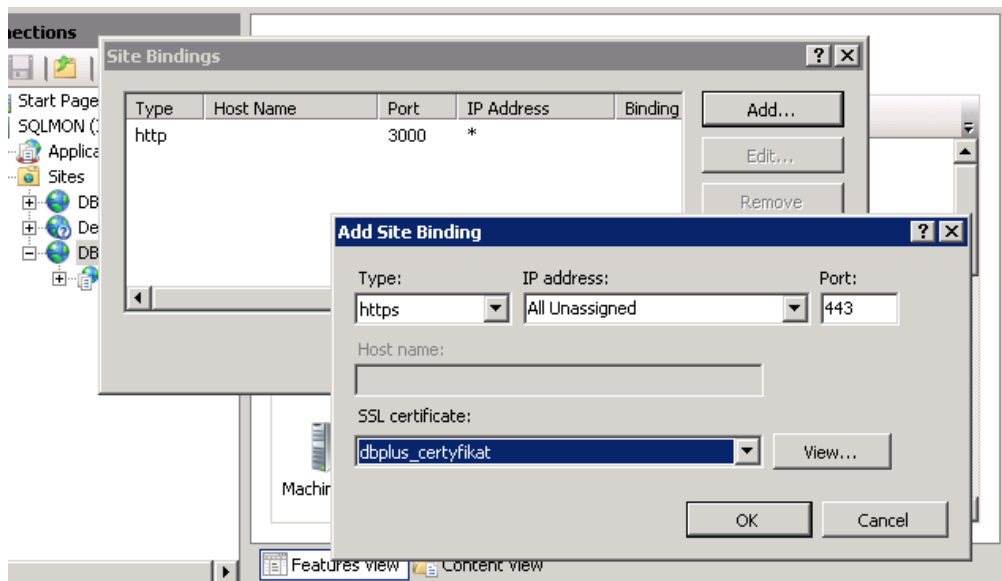
Pass the password if the certificate was exported with a password

5. Addition of the ssl protocol (binding update)

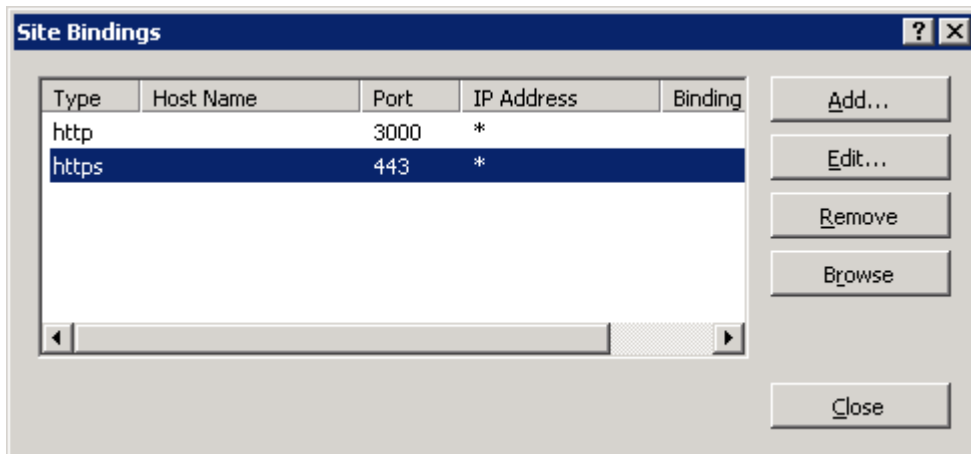
We are updating the link for the DBPLUS Website. Clicking on the site, then Edit Bindings.



In the Site Bindings window, add a new link specifying the SSL protocol and select the certificate previously created or imported as below:



As a result, we receive:



We are remove relationship with the http type.

On the configured DBPLUS Website, we click the restart (Refresh button).

2.5 User application configuration

Another element is the creation of user interface objects. Belong to them:

- DBPLUS Website
- Authorization:
 - LocalSystem,
 - LocalService,
 - NetworkService,
 - Windows Domain Account,
 - ApplicationPoolIdentity

When User choose a username type = LocalService – there is no need to give a user name and a password, the service will operate on default user for Windows (LocalService)

- Parameters:
 - Port number (default 80)
 - Binding property /Host Name
 - The way to access the application - whether users at the login to the site will be asked to authenticate (login and password) or not

DBPLUS Performance Monitor - Installation Wizard

DBPLUS Performance Monitor - user application

Specify settings for web user application

Catcher	Repository	IIS	App	Finish

i Specifying the user account used for connection purposes by application client, is available during DBPLUSORACLECATCHER configuration or installation database repository

It's required to use the same user type/account for mentioned components (DBPLUSORACLECATCHER service, IIS application).

You can specify the port for http protocol and turn on/off windows authentication for users who would access the application. Any access and user privileges to the application you can manage directly in the application

Login type	LocalSystem	▼	
Username	<input type="text"/>		
Password	<input type="password"/>		<input type="button" value="Test user"/>
Port	80		<input style="border: 2px solid blue;" type="button" value="Test port"/>
Binding property	Default	▼	
Host name	<input type="text"/>		
Access to application	Windows authorization	▼	
Application path	C:\Program Files (x86)\DBPLUS.Or		<input type="button" value="Select application"/>

Step 5 from 6

As a result of the entire setup process completion, your application will be available at the following url:

http://servername:port_number/dpmoracle

If the system will be running on port 80, link will be as follows:

<http://servername/dpmoracle>

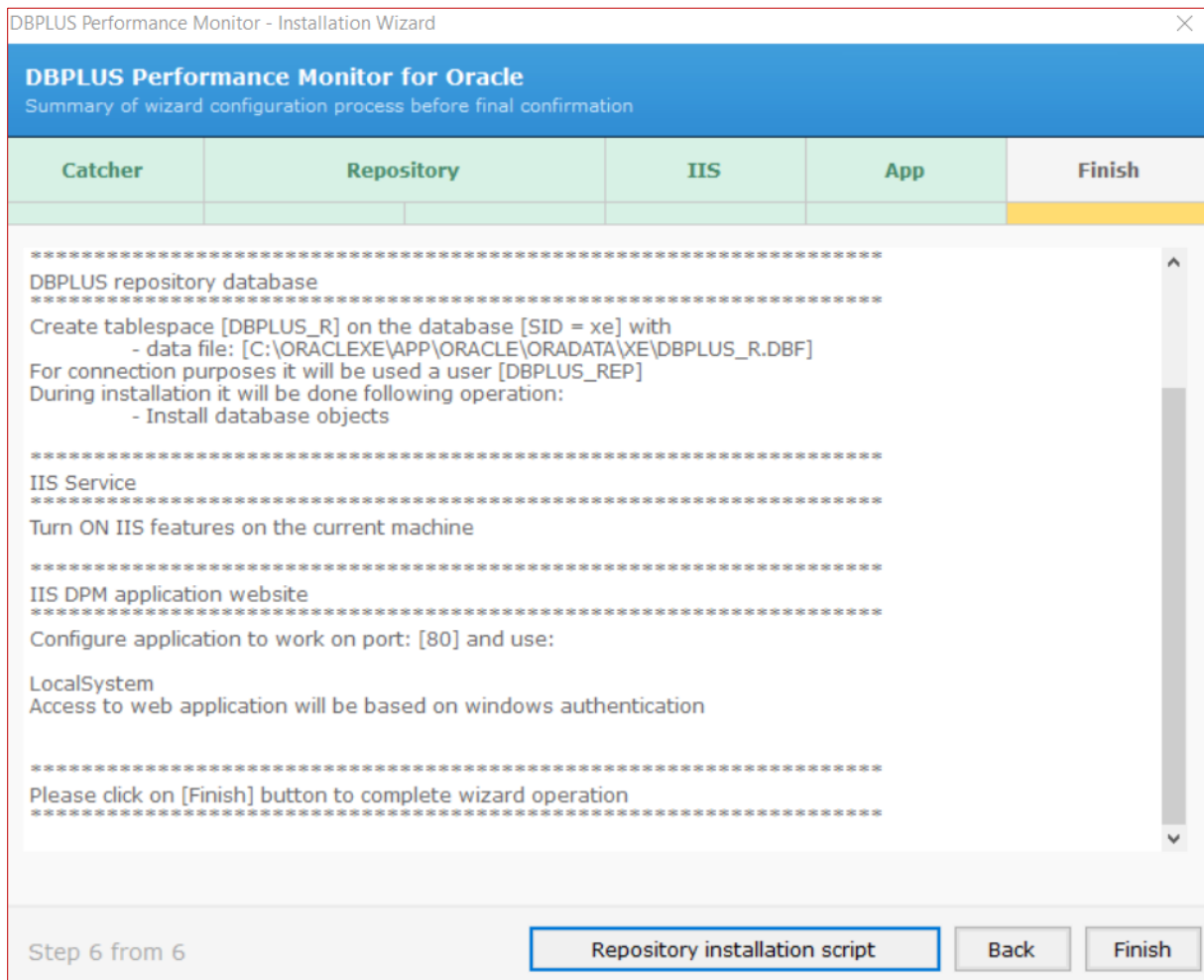
Click on the **[Continue]** button to proceed the next step

2.6 Configuration summary

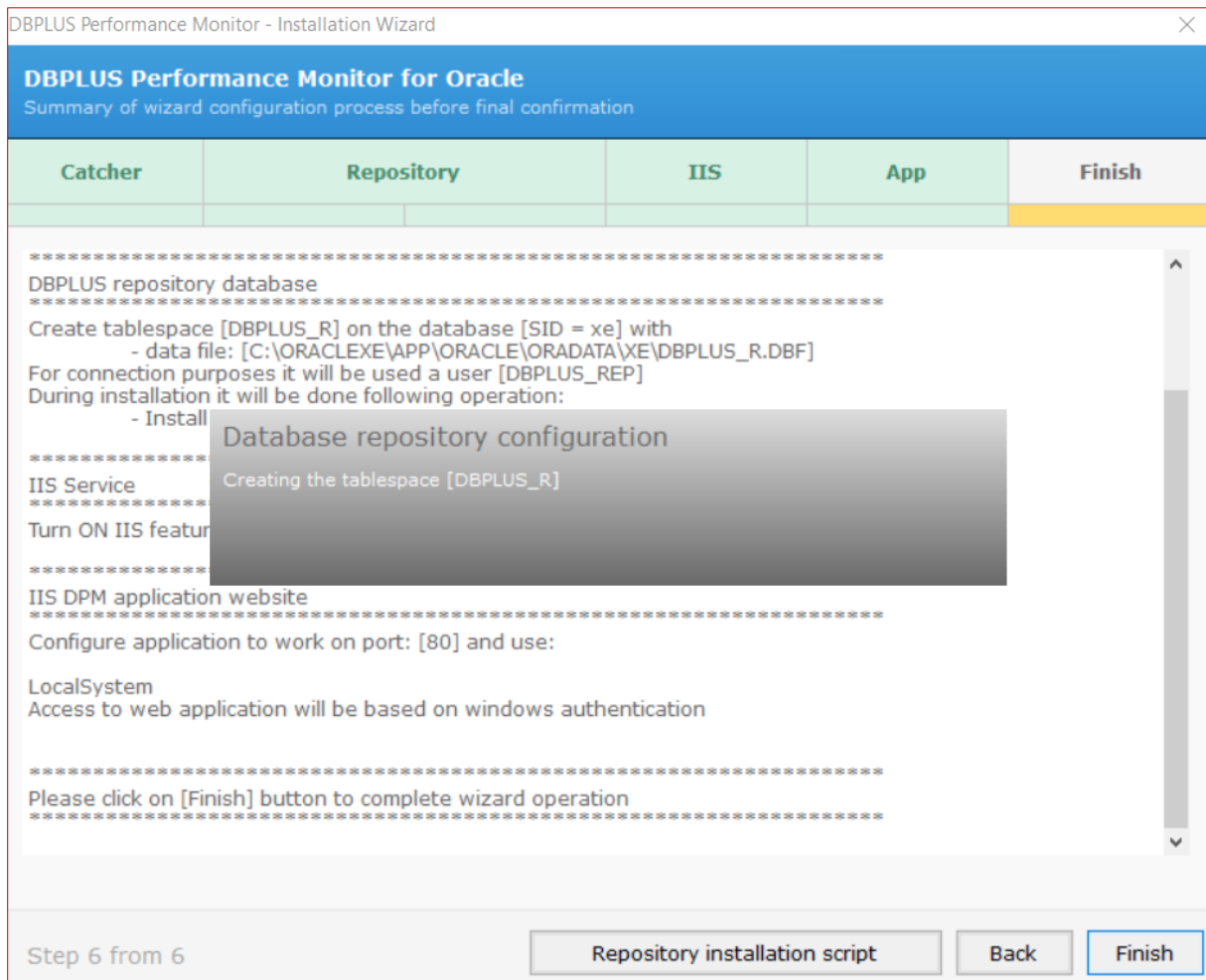
The last configuration step is to confirm all settings according to steps defined in the configurator. The final screen shows a summary.

IMPORTANT! In addition, there is a script available here that will be run on the repository database, by clicking on **[Repository installation script]** we have the option of saving it to disk.

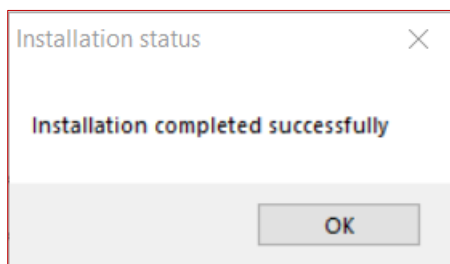
To confirm changes, click on **[Finish]** button.



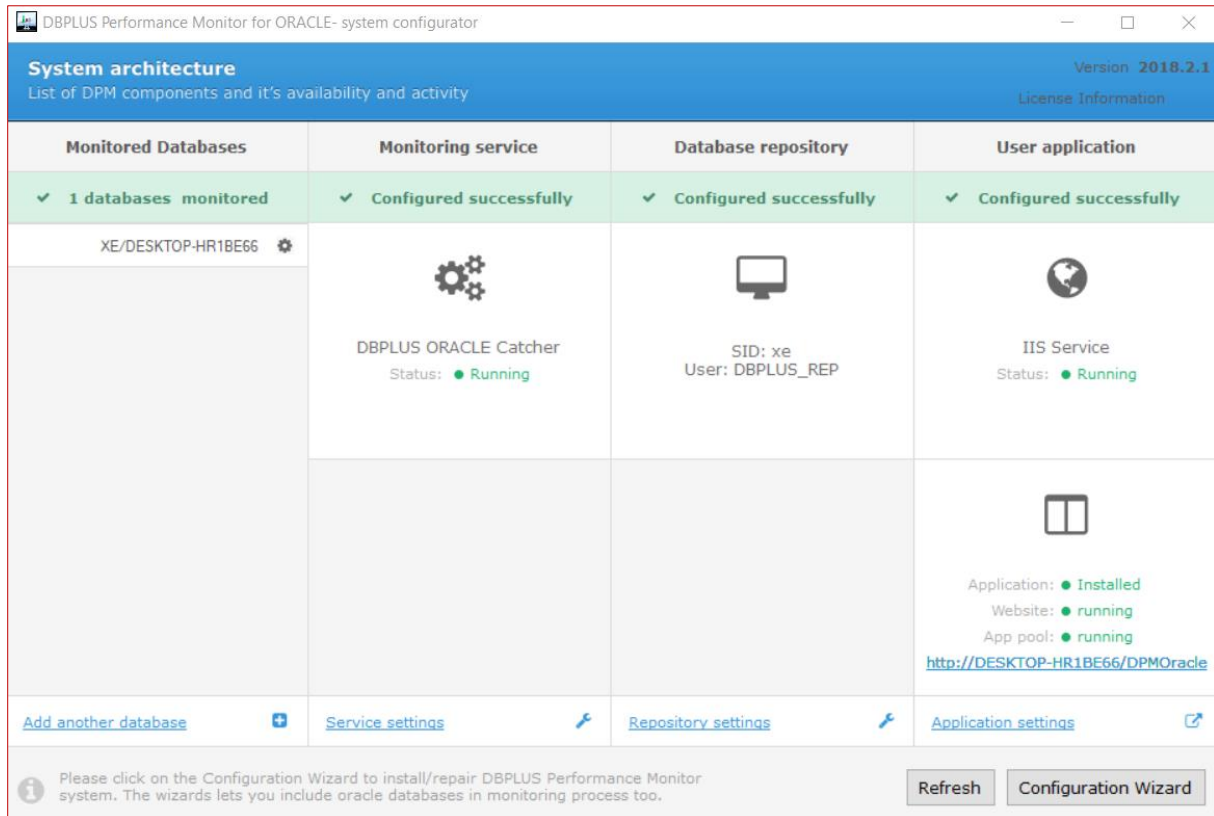
In the meantime, information about the progress of work is displayed:



At the end of the completion status of the installation/configuration



As a result, system configuration main window looks like this:



From the above sample screen, we can read that DBPLUS system PERFORMANCE MONITOR is:

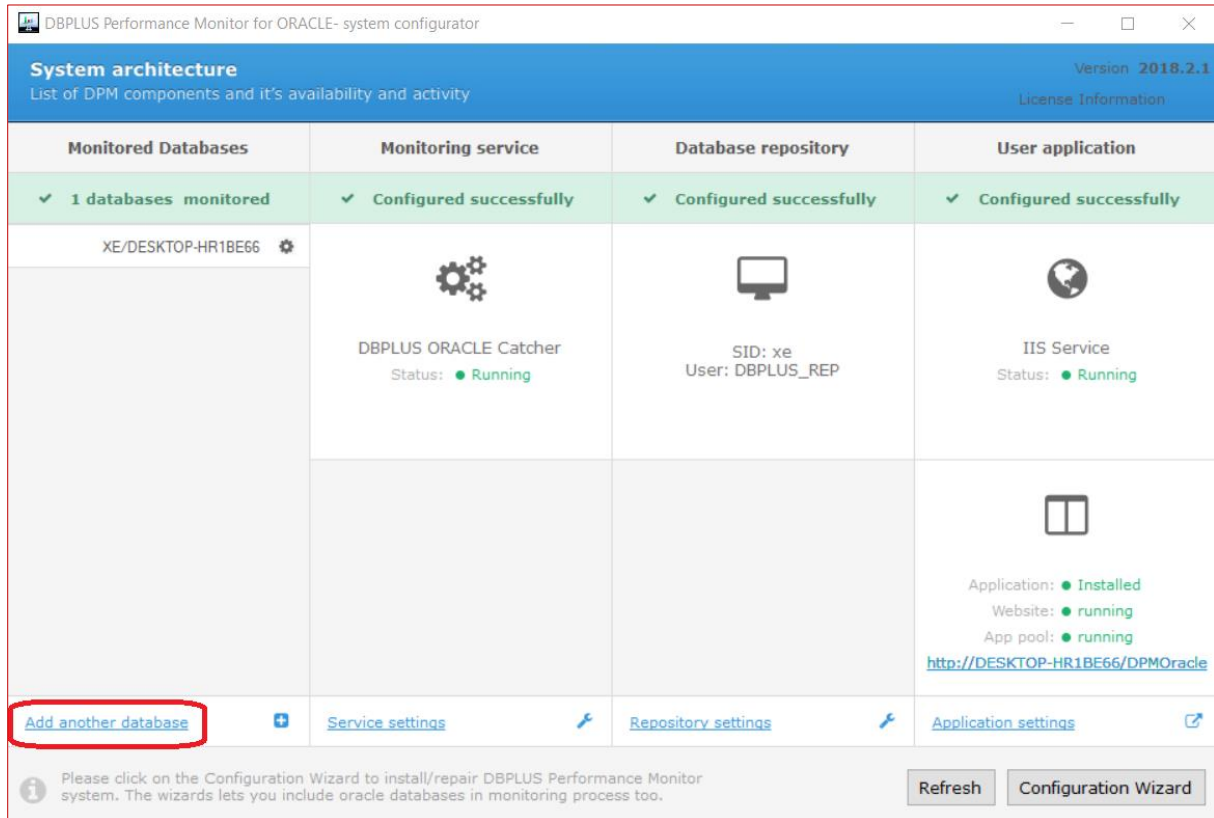
- installed on the server DESKTOP-HR1BE66 (link to the application in the lower right corner)
- all components are properly configured (the bar with information “Configured successfully”)
- appropriate services are running:
 - DBPLUSORACLECATCHER - a service responsible for database monitoring
 - IIS, Website, App pool - which means that the application is available to the user
- We have 1 monitored Oracle database
- Information from the monitoring of all instances (currently one) are stored in the database „XE/DESKTOP-HR1BE66” (XE database on the server DESKTOP-HR1BE66),
-

Interface / User application is available at:

<http://DESKTOP-HR1BE66/DPMORACLE>

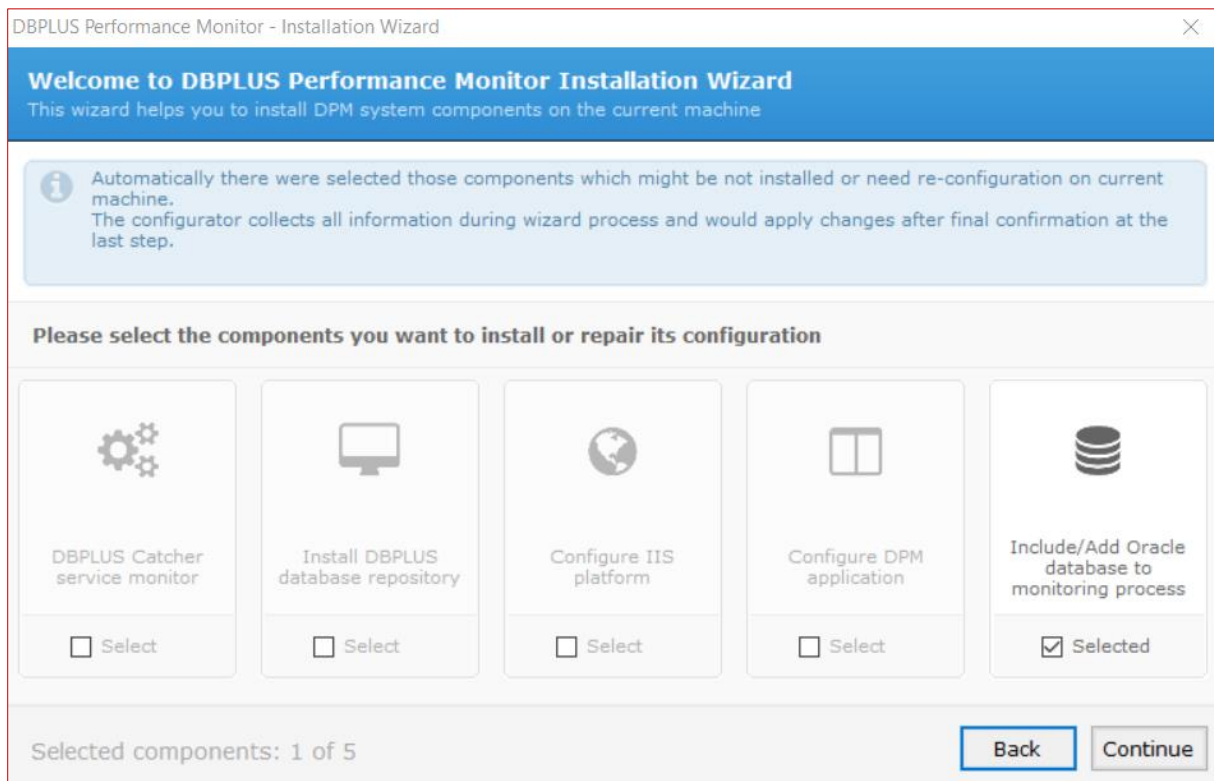
3 Adding a database for monitoring

After the initial configuration, you can proceed to add more databases for monitoring. For this purpose, in the main System Configurator Window we click **[Add Another database]** button.



IMPORTANT: If the **[Add another database]** button is not available it is the result of a license for a specified number of databases.

The second option to add an instance is clicking **[Configuration Wizard]** button and select the component **[Include / Add Oracle database to monitoring process]**



As a result, we go to the wizard to add a new database. In the first place we provide basic information:

- Database details,
- Monitoring database user account:
 - Create new user
 - Use existing user

Create of a new user requires one-time login and password for the user with DBA permissions, through which a monitoring account will be created.

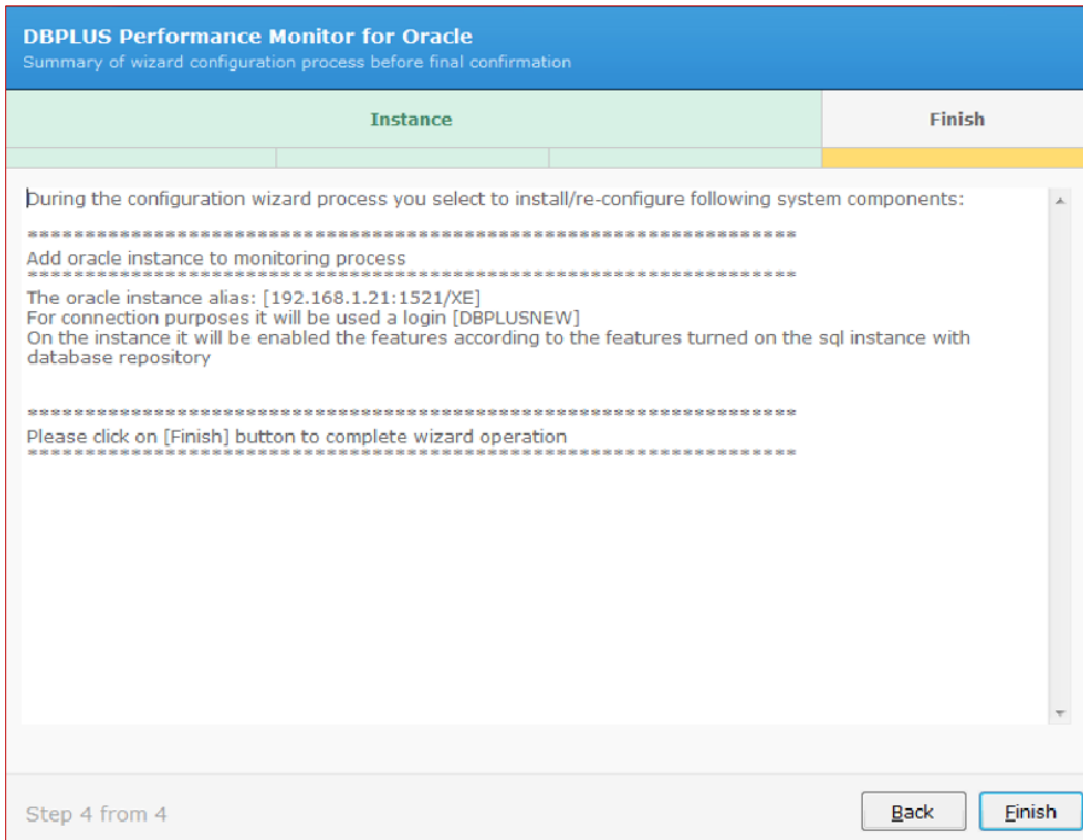
If an existing user option is selected, the monitoring need to have following roles:

- SELECT_CATALOG_ROLE,
- CONNECT.

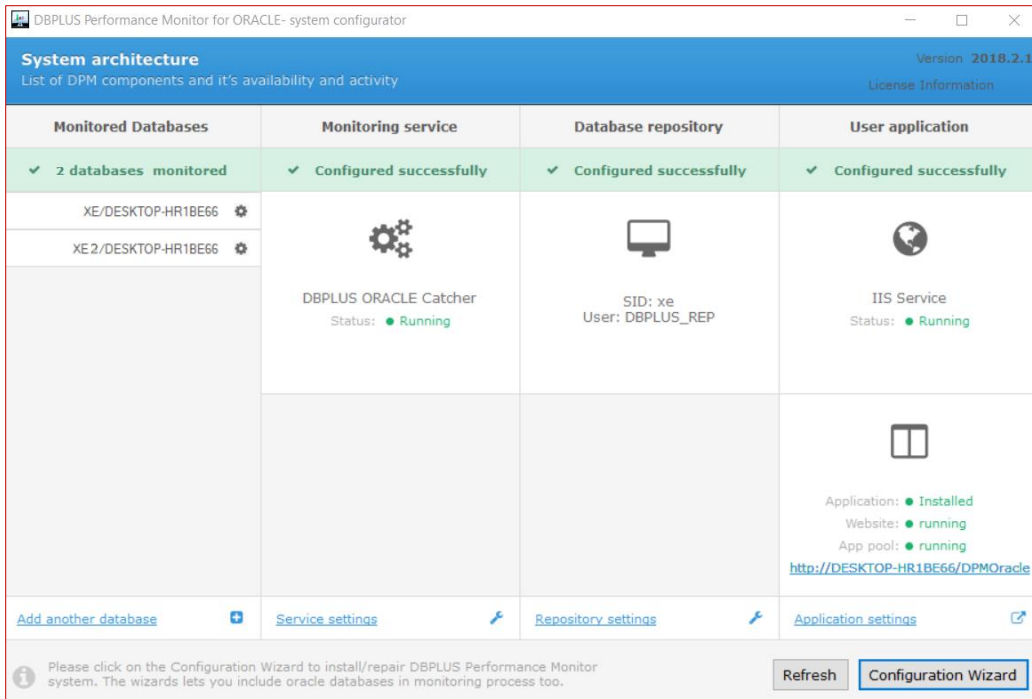
When entering a database name, you can define a connection without using TNS.

In order to connect a new database, we can create a new database usage or indicate an existing one. This user will be used to download statistics from the monitored database (DBPLUSORACLECATCHER will log in to this user).

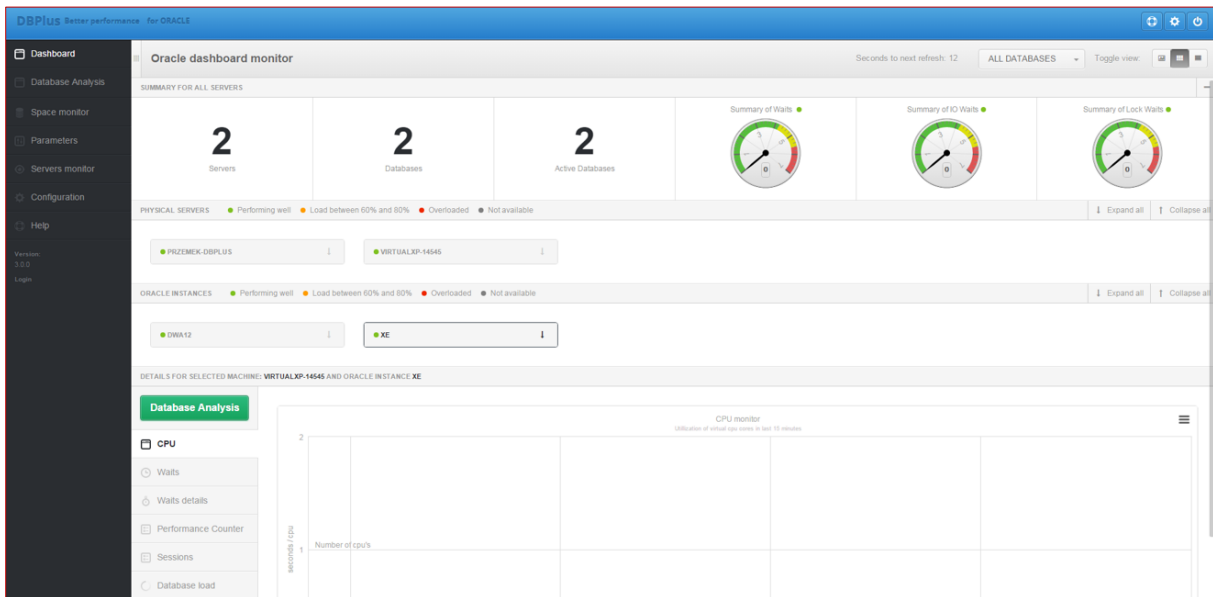
After clicking the [Continue] button, the final screen appears which summarizes the previous steps.



Click **[Finish]** button to add an instance to monitoring. As a result, changes are visible in the system configuration main window - DBPLUS Performance Monitor supports two ORACLE databases.



When User click the link to the application (in this case <http://desktop-hr1be66/DPMOracle>), we will start the application:



4 System Upgrade

Technical support provides the access to new software updates that are published 4 times a year, as well as to DBPLUS engineers' help in Oracle database servers diagnosis, using **DBPLUS Performance Monitor** software.

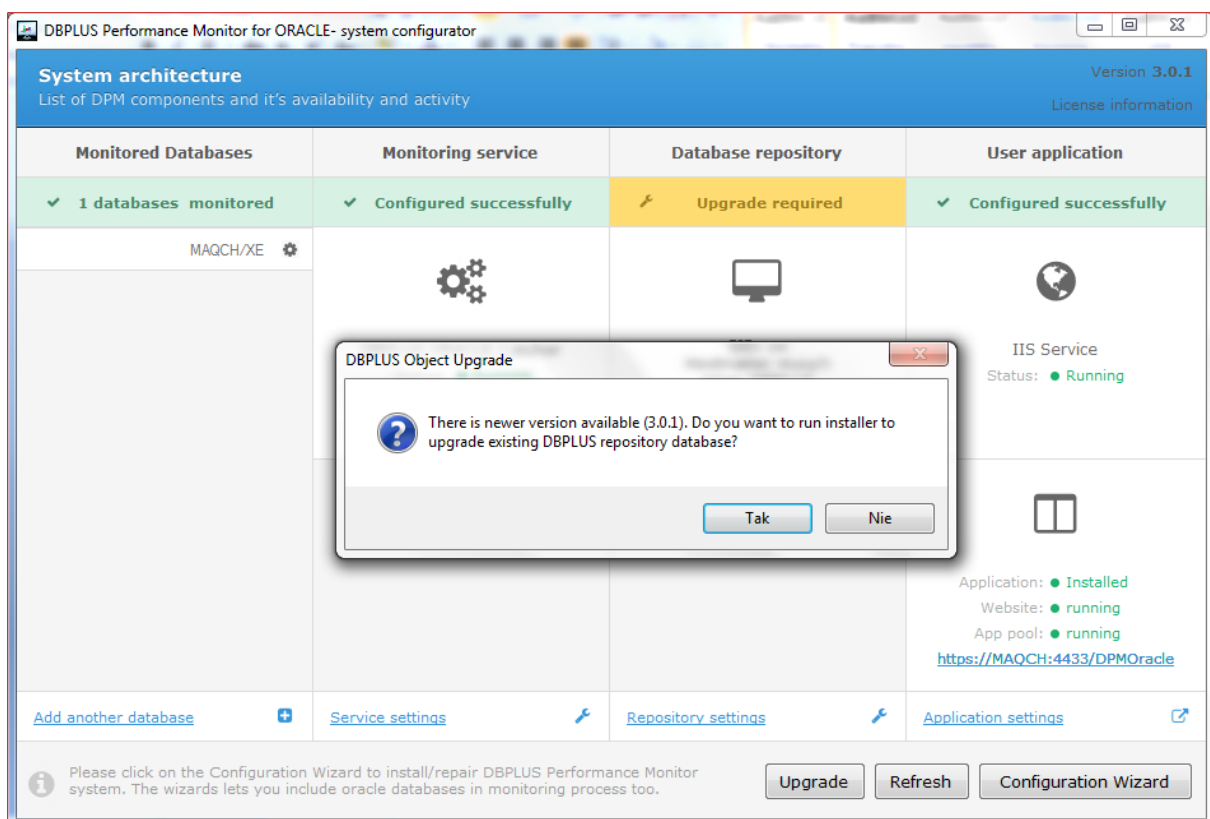
Upgrading system combines with two steps:

- Run the installation file (which goes the same as the first installation)
- Upgrade of database objects repository on DBPLUS user to the latest version

IMPORTANT: The upgrade process involves running the dpmOracleInstaller.exe file containing the new version of the application. Remember to choose exactly the same folder you used during the first installation during installation.

4.1 Setting up for the latest version

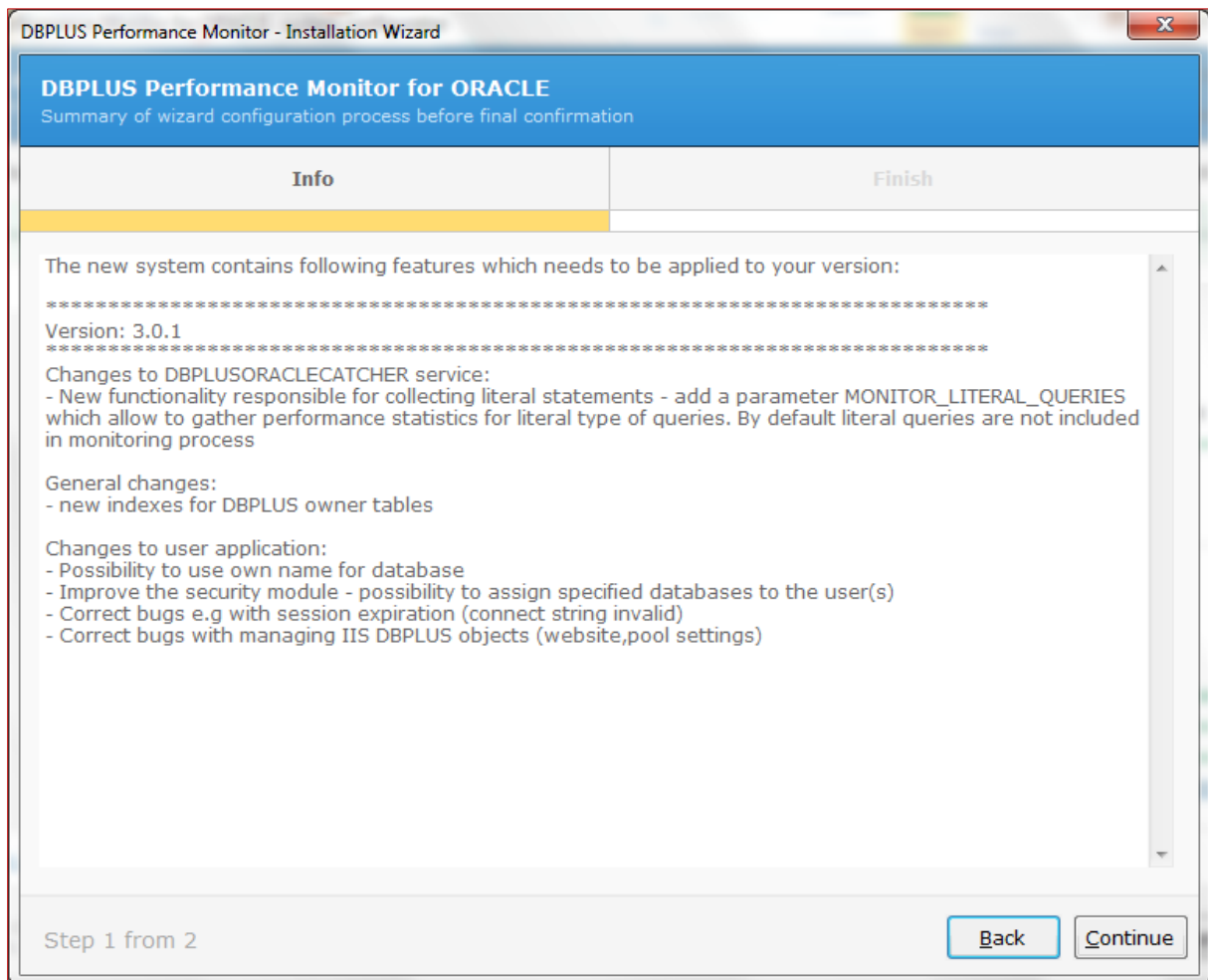
In order to go through the upgrade process, you have to run DBPLUS Configuration Wizard, which also runs automatically after installation. In the result:



System automatically detects the need to update to the latest version. We accept the dialog box and we run the wizard that will guide us through the upgrade process.

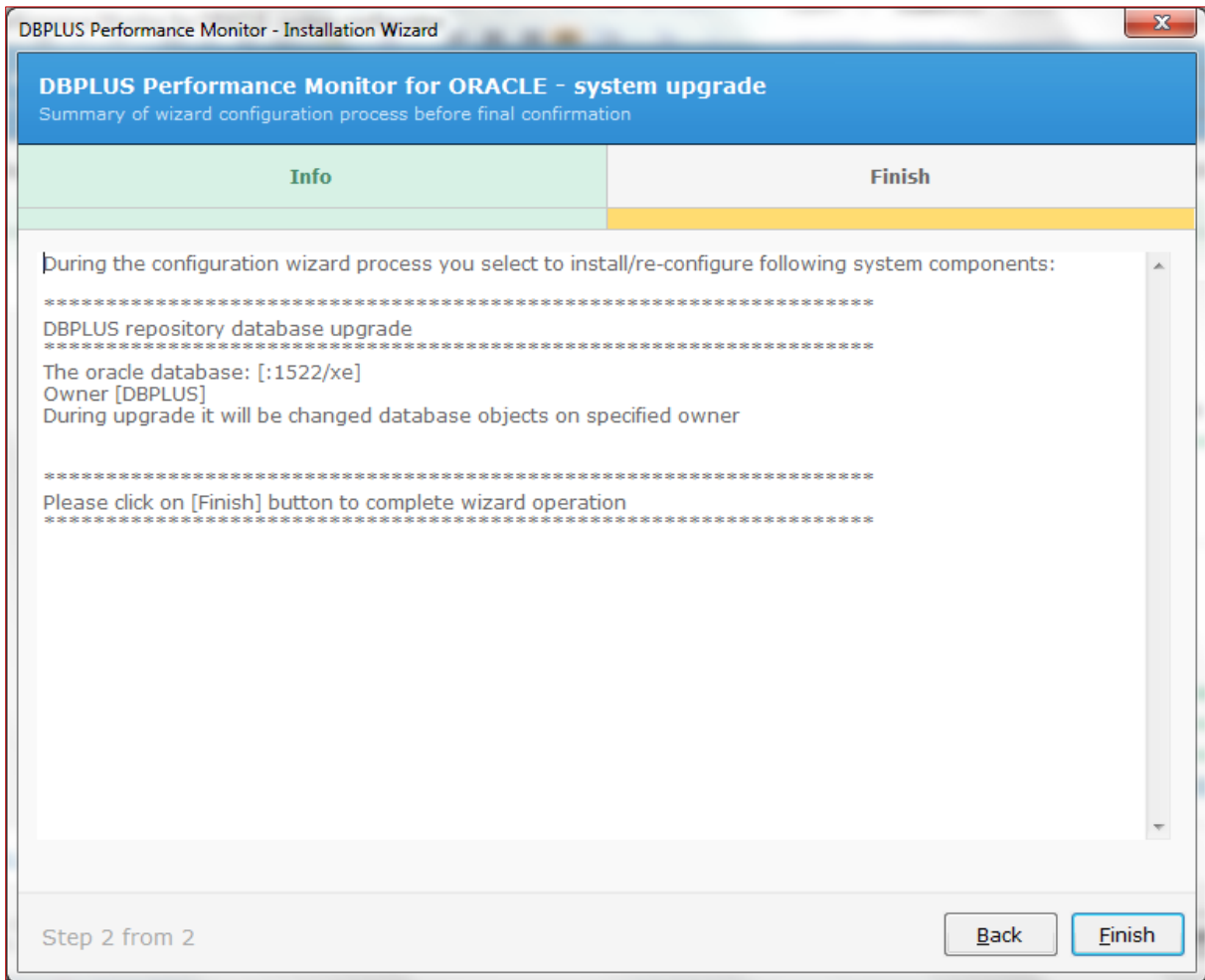
In case of withdrawal from the operation we can always return to it by clicking [**Upgrade**] in the main configurator's window.

As the first screen we have information about system version, to which application will be updated (with system functionality description):



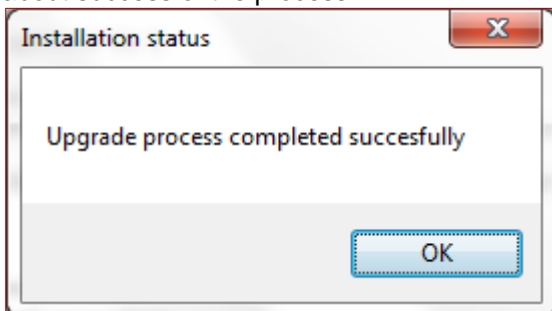
The upgrade procedure applies to objects only in database where there is a DBPLUS repository.

We accept by clicking **[Continue]** button.



System informs about operations that will perform in database on DBPLUS owner. We accept it by clicking **[Finish]**.

Depending on version of the update process can take 1 to 3 minutes. At the end we get the information about success of the process.



Next, close the configurator's window.

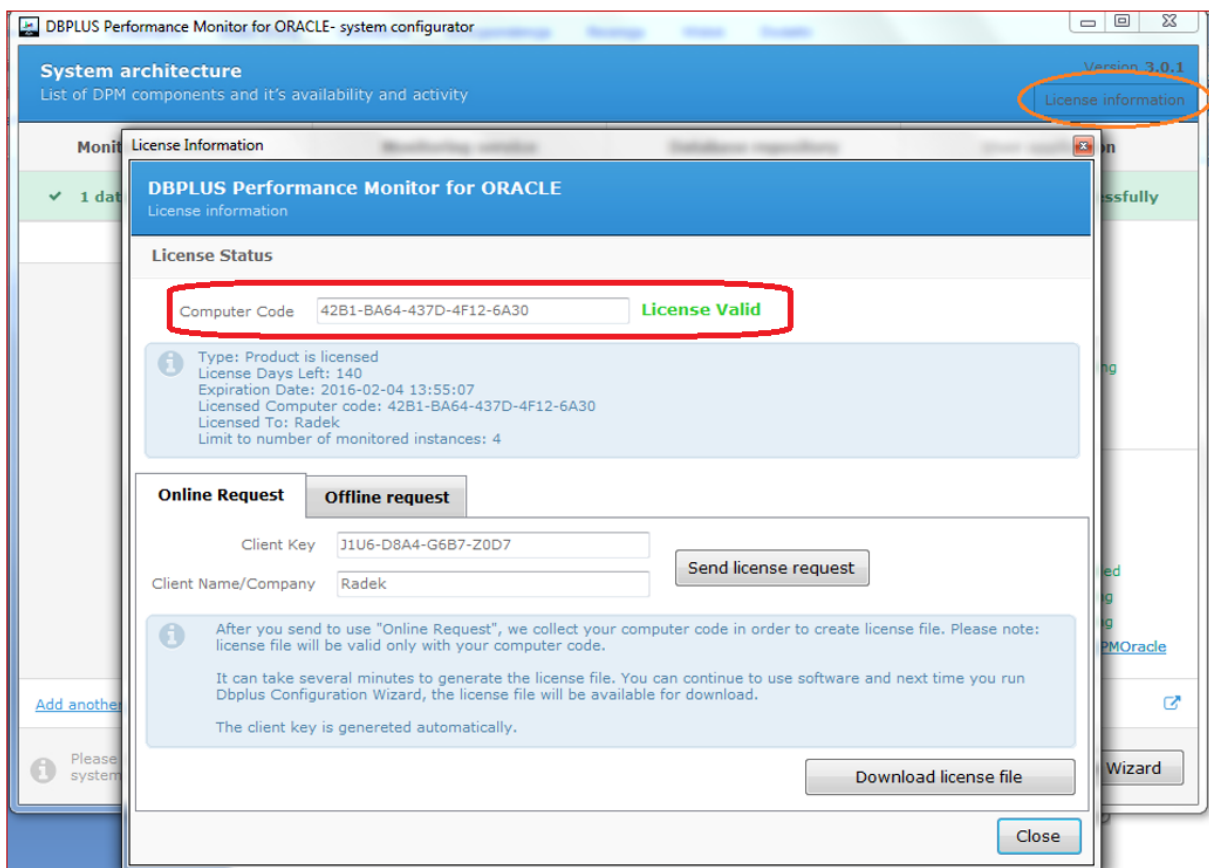
5 License

The license is generated always for the server with **DBPLUS Performance Monitor™** software and not for the database.

Standard system license includes:

- System uptime
- The number of monitored Oracle databases

Information about licensing is available from the configurator, i.e. **DBPLUS Configuration Wizard**



After the first installation, system operates in trial version. This period lasts 14 days and system is available in its full functionality. By the end of the period it's necessary to register the system. It can be done in two ways:

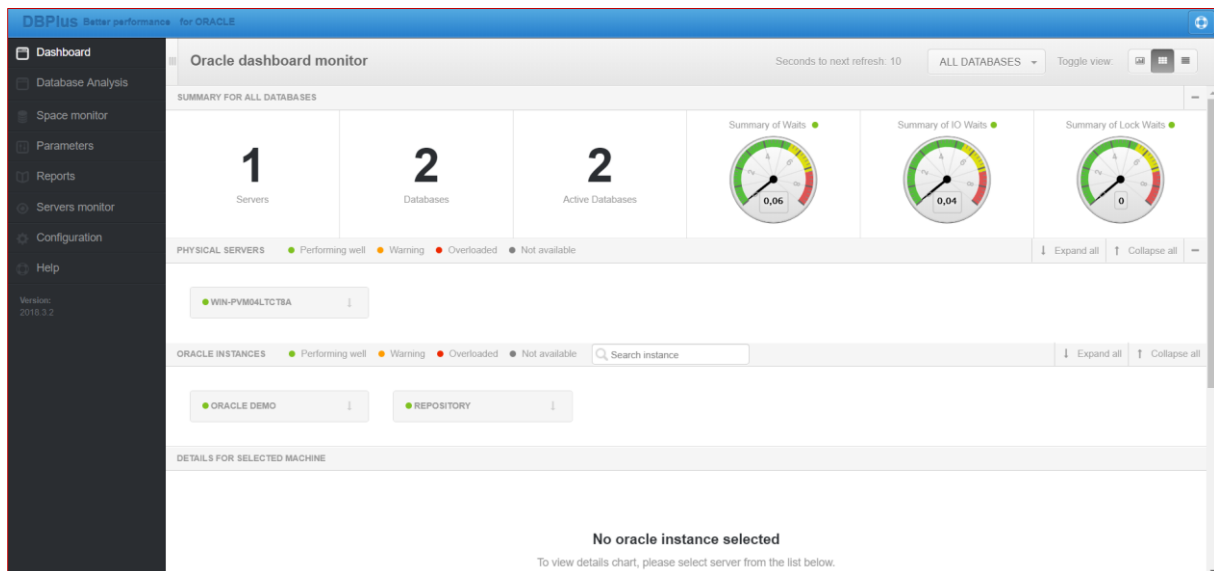
- Sending requests for licenses from the form by clicking button **[Send license request]** (Internet access required on the machine)
- Sending computer code via e-mail

6 Working with program

The user interface is available through the browser at the previously configured address. Default starting page is a dashboard showing the currently monitored databases and how they perform.

6.1 Menu „Dashboard”

When **DBPLUS Performance Monitor** has start, dashboard showing all currently monitored Oracle databases appears.



Dashboard is divided into the following sections:

- Information Bar
- Servers Summary
- Number of Physical Servers
- Number of Oracle Instances
- Further Details about Selected Database

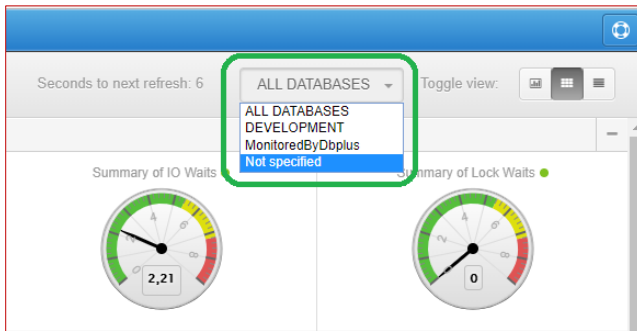
6.1.1 Information Bar

User can switch between dashboard and different mode using the information section. List of available modes:

- **Icon View** - displays monitored servers / databases as icons (default)
- **Grid View** – databases are displayed in a grid/ table view
- **Television Mode** - shows instances of Oracle in the form of developed icons with automatically switching performance indicators.

Additionally, User is informed how much time is left until next dashboard refresh with the most recent data about the current performance of all monitored instances.

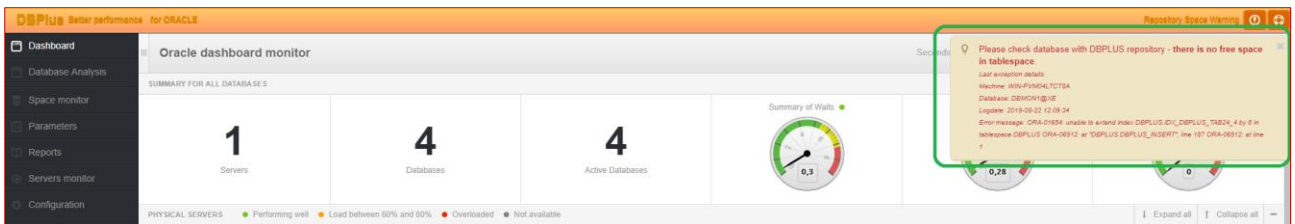
Users can change display information's about database instances via the drop-down menu in the information bar. Various types of databases can be defined and attributed in the 'Configuration' menu, which is described in the next chapter.



In case when information bar changes colour from blue to orange – this implicates insufficient space in the DBPLUS repository base or that DBPLUSORACLECATHER service doesn't work.

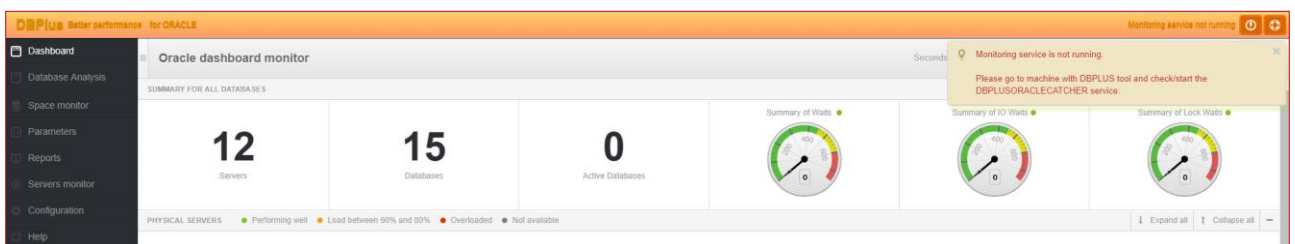
Insufficient space at repository database

In the instance of space in the pattern database, which repository is intended for DBPLUS to collect data, a warning message will show. The toolbar on Dashboard page turns orange and a lack of space message „Repository Space Warning” about lack of space will be shown.



DBPLUSORACLECATHER service is not running

In the event of monitoring service problem, the toolbar on Dashboard page turns orange and information „Monitoring service not running” will appear. To fix the problem you need to check for any issues with the server on which the client DBPLUS is installed, after which restart the DBPLUSORACLECATHER service.

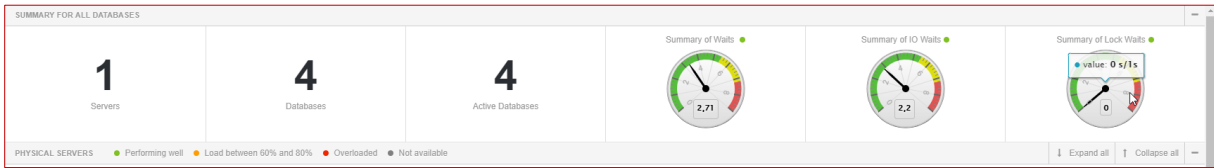


6.1.2 The summary area

The main area provides a general summary of:

- number of monitored servers and databases
- number of active databases
- total of current waits, databases spend their time on:
 - Summary of Waits
 - Summary of IO Waits

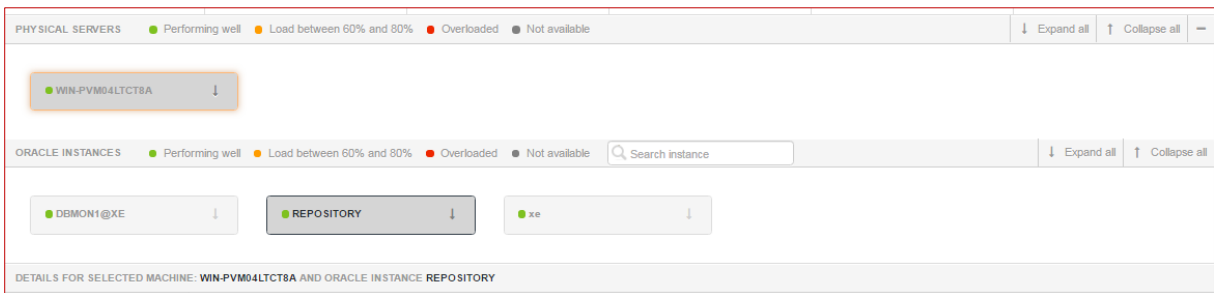
- Summary of Lock Waits



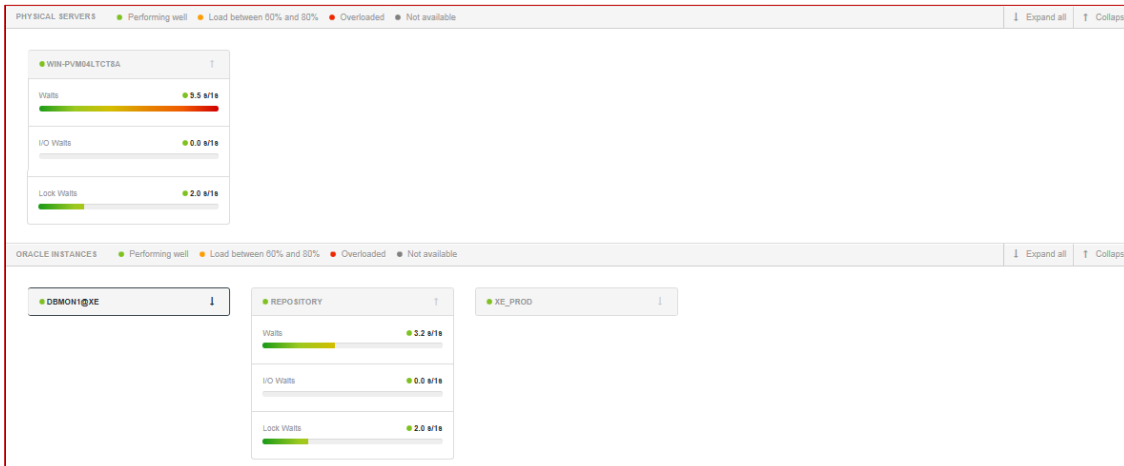
From the main area you can tell whether the level of waits is high, or should you investigate further to resolve any issues.

6.1.3 Servers and databases area

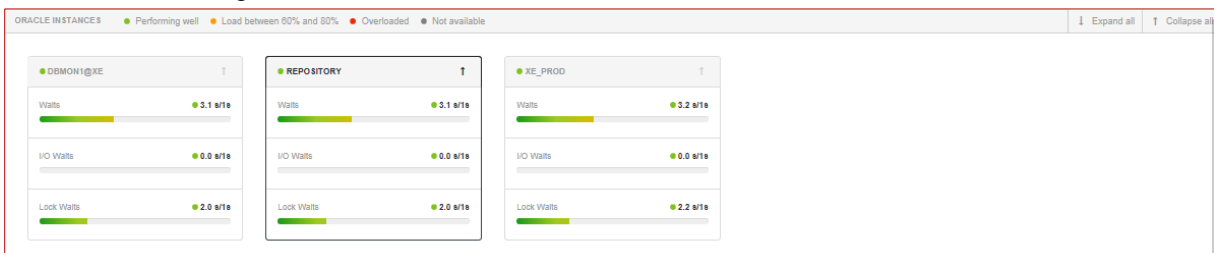
Inside the 'Physical Servers' section, we can see the number of servers currently running ORACLE databases. Clicking on the server in the area below, will illuminate any Oracle databases operating on the machine.



More information about each server or database can be accessed by clicking on the expansion arrow or by clicking the **[Expand All]** button on the top right of each section.

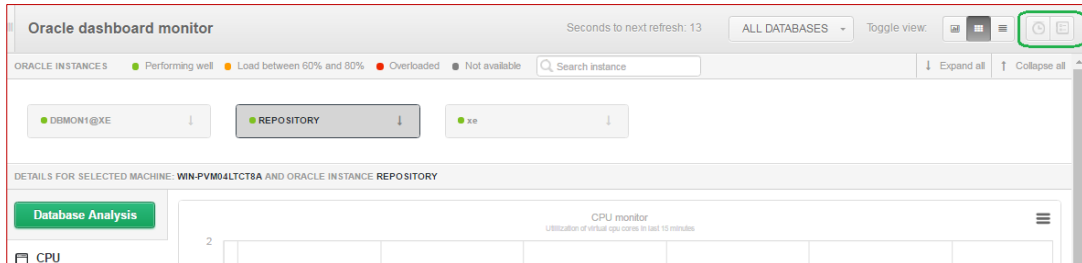


When **[Expand All]** button is clicked, for the database section, you will see exactly which Oracle database has the highest level of waits.



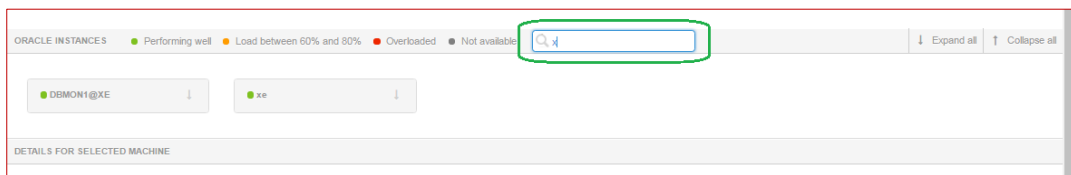
Additional options on dashboard screen:

- Option to collapse the summary bar and the physical server area



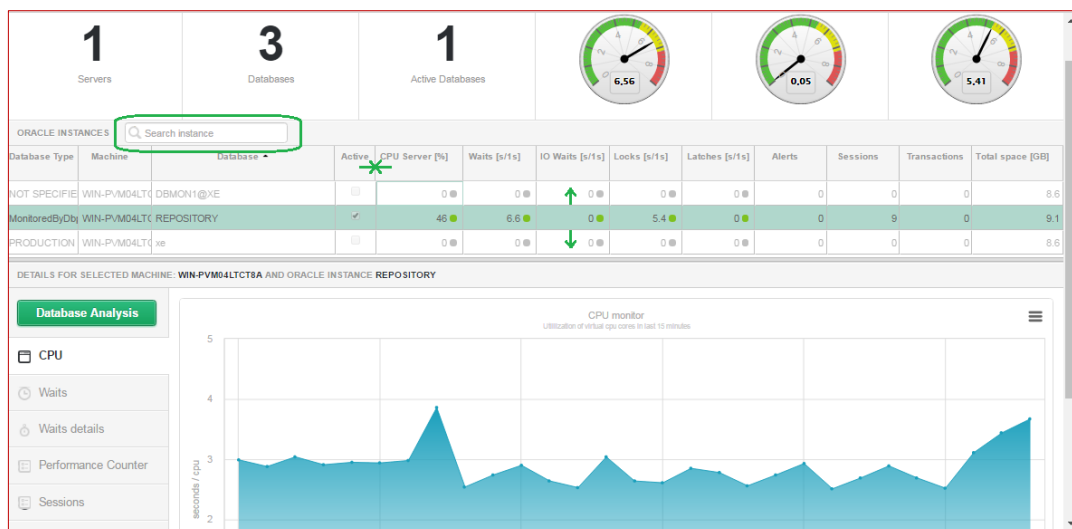
Buttons situated at the top right-hand corner of the screen allow Users to re-open any collapsed areas, additionally system remembers user's settings on used browser.

- Search databases – the search option is available throughout the dashboard. This is especially useful for monitoring high number of databases.



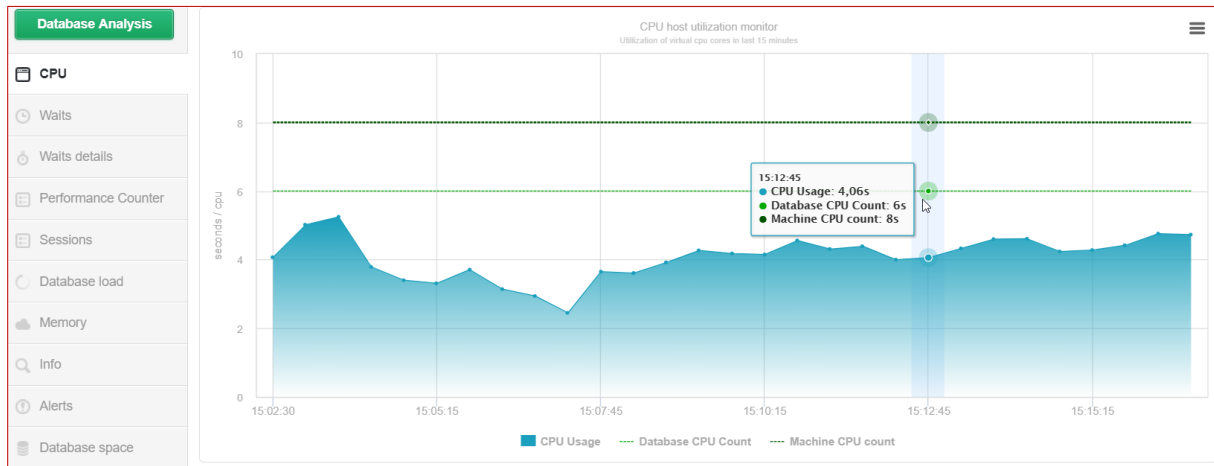
Database Type	Machine	Database	Active	CPU Server [%]	Waits [s/f/s]	IO Waits [s/f/s]	Locks [s/f/s]	Latches [s/f/s]	Alerts	Sessions	Transactions	Total space [GB]
NOT SPECIFIC	WIN-PVM04LTCT	DBMON1@XE	☑	12	1.4	0.3	1.1	0	0	2	0	9.1
MonitoredByDb	WIN-PVM04LTCT	REPOSITORY	☑	12	1.9	0.9	1	0	0	3	0	9.1
PRODUCTION	WIN-PVM04LTCT	xe	☑	12	1.4	0.3	1.1	0	0	2	0	9.1

- *Grind View* showing the current database performance comes with following abilities:
 - Changing the width of columns
 - In case of larger number of records, scrolling does not hide table headlines
 - Use of arrow buttons to navigate between databases



6.1.4 Details of ORACLE database performance

To analyze the current workload, click the specified database icon. As a result, the lower part of the dashboard reloads itself and presents details of the selected instance.



Dashboard allows here:

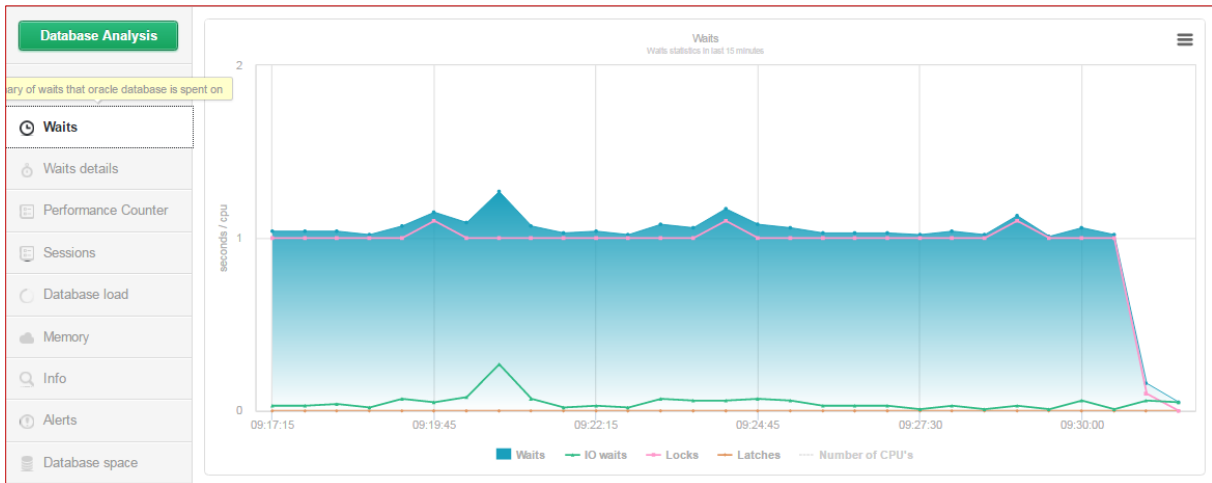
- To observe the current CPU usage - **this is a load of all processors running on the machine,**
- determine which waits the database currently spends time on – **Waits bookmark, Waits details bookmark,**
- analyze performance indicators from the last 24 hours - **Performance Counters,**
- check the level of the session / locks and active transactions – **Sessions,**
- display the load of the database from the last 24 hours - **Database Load,**
- verify memory utilization – **Memory,**
- display basic information about database – **Info,**
- check alerts – **Alerts,**
- check database size by tablespaces – **Database space**

Information about the CPU load, Waits, and sessions are presented here horizontally for the last 15 minutes, and are refreshed every 30 seconds. For example, the waits chart below provides following information:

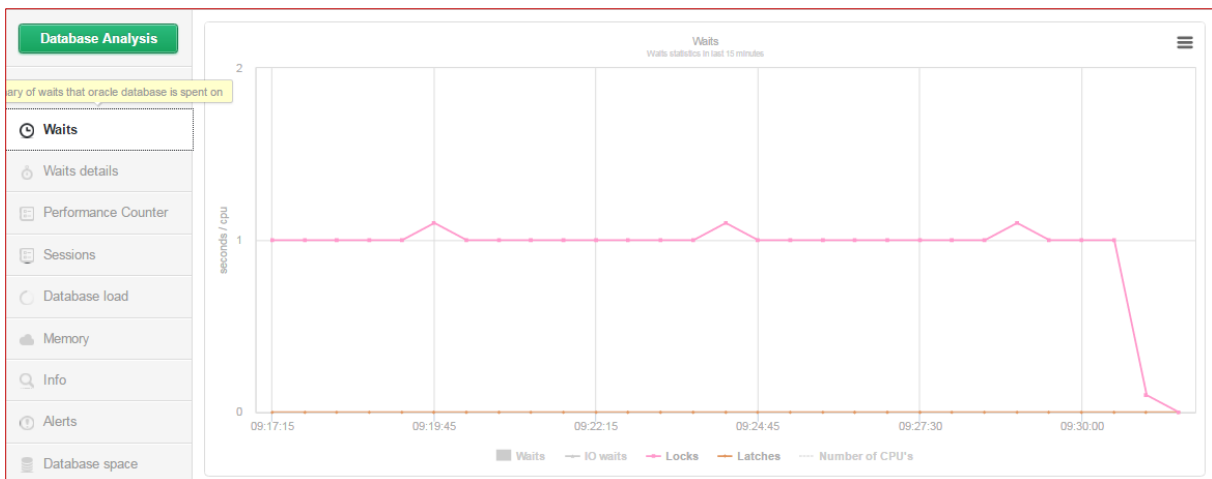
- I/O waits - disk readings
- Locks – sessions locked records
- Latches - waits before accessing the database buffers

The graph shows that at a given moment in time (the time read from the X-axis), all users (active sessions) are waiting for the query result, indicated number of seconds (the result read from the Y-axis). Categories I/O, Locks, Latches help to state why sessions are staying idle.

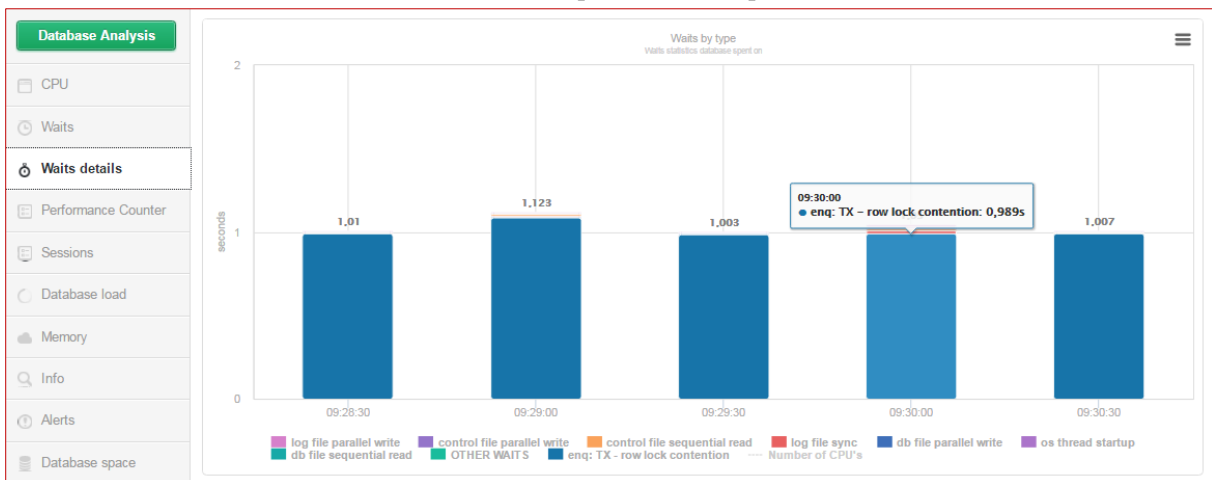
By default, all series are visible:



After clicking on the **[Waits, IO Waits, Latches]** series, only the **[Lock]** waits series remains visible. A second click on the legend bar will display the selected series again.

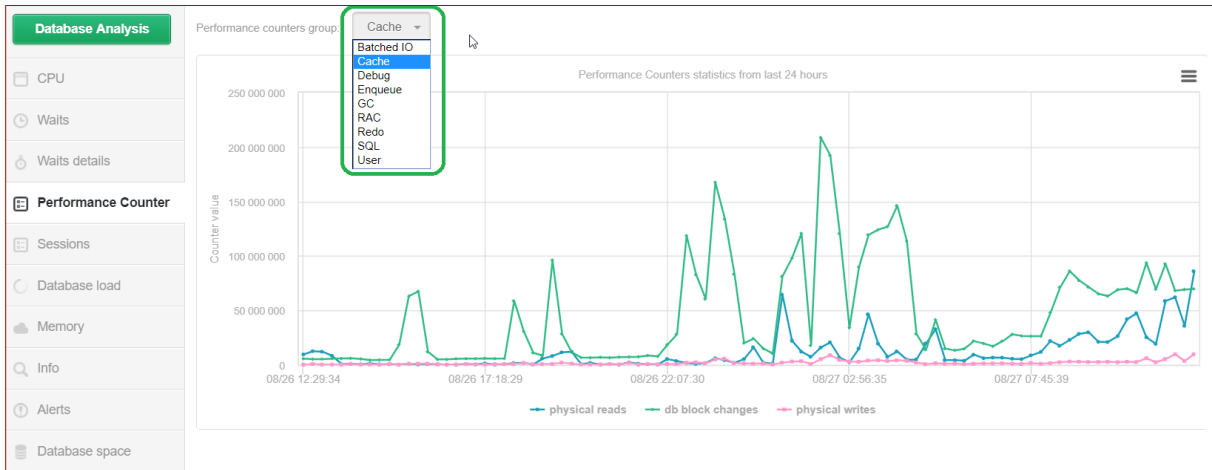


IMPORTANT: the level of waits is re-calculated to one second.
Details about Waits can be found in the next tab **[Waits details]**.

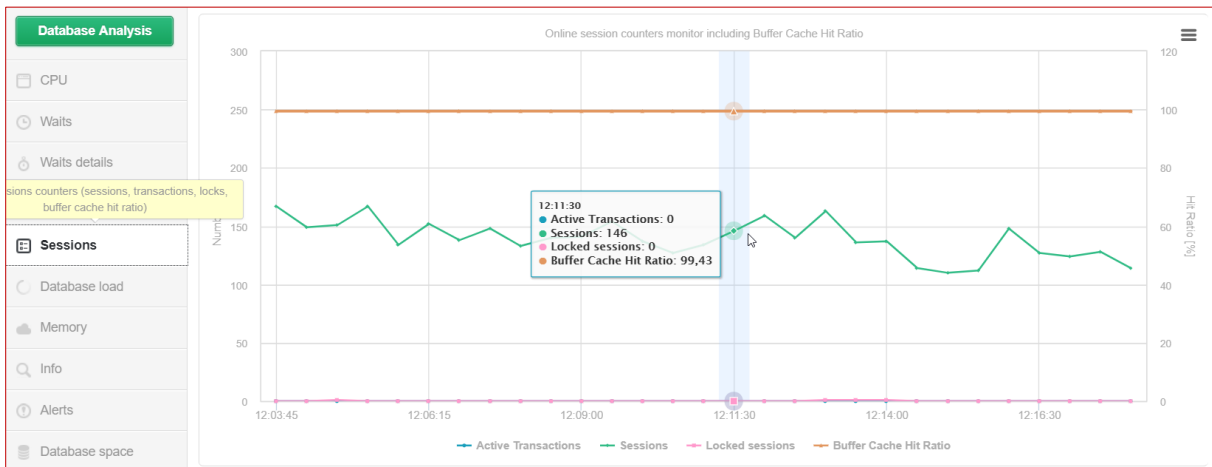


On the dashboard screen for the selected database, you can also check the load, system indicators - so called performance counters and memory utilization of the last 24 hours.

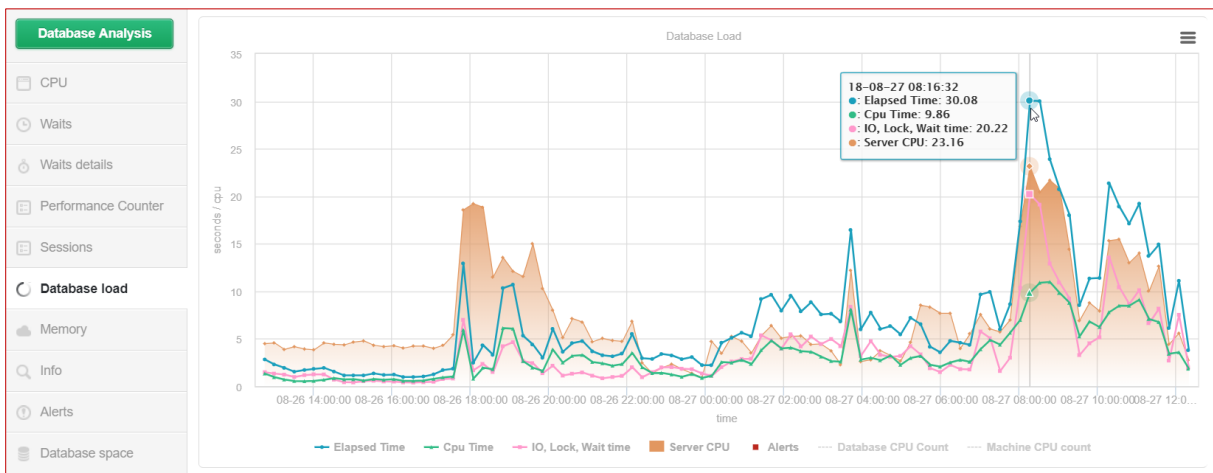
To do this, click on the appropriate on menu on the left side. Be aware of the additional selection filters for the **Performance Counter** dropdown menu.



The next tab shows **Sessions** view where you can find basic information about number of session and hit ratio of the Buffer Cache.



Information about the load for last 24h can be view after clicking on **Database Load**.

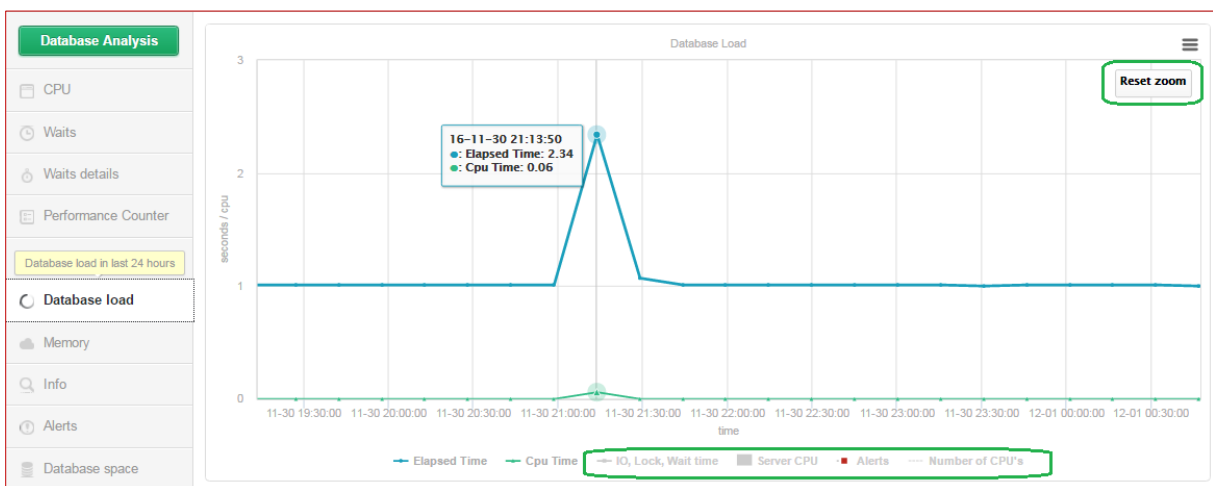


Database Load is one of the core modules used by DBPLUS engineers to analyze performance. Graph consists of the following series:

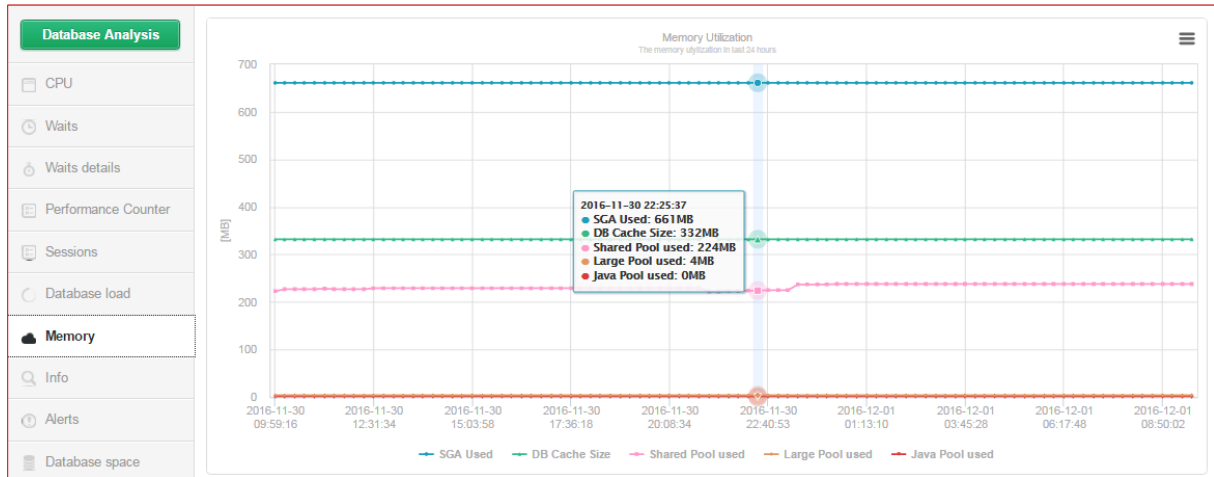
- **Elapsed Time** - shows the waiting time for all users on the query result in a given second of time. The graph for the displayed Elapsed Time point is 2.93 seconds that can be interpreted as follows:
 - three Users have execute different queries
 - two Users waited for one second,
 - third User waited 0.93 second.
- **CPU Time** – the utilization of server processors by all queries in a given second in time.
- **IO Lock, Wait Time** - the result of the difference between the Elapsed Time and CPU time of database
- **Server CPU** - CPU load on the server
- **Database CPU Count** – number of processors which are assigned to the database instance
- **Machine CPU Count** – number of processors on the server (series of the graph is hidden by default)
- **Alerts** – number of alerts that has occurred during this time
- **Missing snapshots** – the series that exists in case when monitoring service didn't do efficiency measurements (e.g. database is unavailable).

For better readability of the graph: you can disable (or enable) given series of the chart – you do it in the chart legend. You can also zoom in or out in the chart.

Here is an example showing series of Elapsed Time and CPU Time in the narrower time frame:



Clicking the “**Memory**” tab, information about memory utilization in the database is displayed. Users can also see the memory levels assigned to different areas. (DB Cache size, Shared, Large, Java Pools):



Dashboard also allows you to view basic information about database such as:

- database version
- the number of available processors
- date of last reboot
- recent changes of database parameters

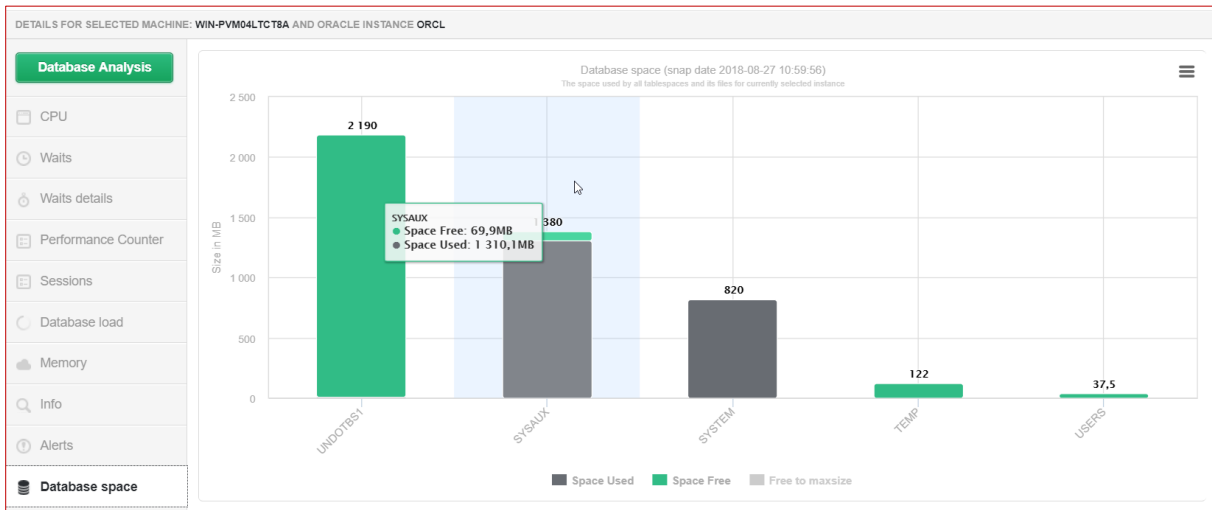
Clicking the **Info** tab displays following window:

Database Analysis	Database information	Last changes																																		
	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Server type</td> <td>TEMPORARY</td> </tr> <tr> <td>service_names</td> <td></td> </tr> <tr> <td>compatible</td> <td>12.1.0.0.0</td> </tr> <tr> <td>instance_type</td> <td>RDBMS</td> </tr> <tr> <td>nls_language</td> <td>AMERICAN</td> </tr> <tr> <td>undo_tablespace</td> <td>UNDOTBS1</td> </tr> <tr> <td>optimizer_mode</td> <td>ALL_ROWS</td> </tr> <tr> <td>memory_target</td> <td>0</td> </tr> <tr> <td>Startup_Time</td> <td>2018/05/08 13:06:13</td> </tr> <tr> <td>Version</td> <td>12.1.0.1.0</td> </tr> <tr> <td>Tablespaces count</td> <td>818</td> </tr> <tr> <td>Virtual CPU count for database</td> <td>26</td> </tr> <tr> <td>Virtual CPU count on the machine</td> <td>26</td> </tr> </tbody> </table>	Parameter	Value	Server type	TEMPORARY	service_names		compatible	12.1.0.0.0	instance_type	RDBMS	nls_language	AMERICAN	undo_tablespace	UNDOTBS1	optimizer_mode	ALL_ROWS	memory_target	0	Startup_Time	2018/05/08 13:06:13	Version	12.1.0.1.0	Tablespaces count	818	Virtual CPU count for database	26	Virtual CPU count on the machine	26	<table border="1"> <thead> <tr> <th>Date change</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>2018-08-26 16:02:40</td> <td>Server parameter Parameter DEFAULT_PLAN changed to resource_manager_plan</td> </tr> <tr> <td>2018-05-08 14:19:14</td> <td>Last tablespace created BIIC_DOSTF_2016</td> </tr> </tbody> </table>	Date change	Description	2018-08-26 16:02:40	Server parameter Parameter DEFAULT_PLAN changed to resource_manager_plan	2018-05-08 14:19:14	Last tablespace created BIIC_DOSTF_2016
Parameter	Value																																			
Server type	TEMPORARY																																			
service_names																																				
compatible	12.1.0.0.0																																			
instance_type	RDBMS																																			
nls_language	AMERICAN																																			
undo_tablespace	UNDOTBS1																																			
optimizer_mode	ALL_ROWS																																			
memory_target	0																																			
Startup_Time	2018/05/08 13:06:13																																			
Version	12.1.0.1.0																																			
Tablespaces count	818																																			
Virtual CPU count for database	26																																			
Virtual CPU count on the machine	26																																			
Date change	Description																																			
2018-08-26 16:02:40	Server parameter Parameter DEFAULT_PLAN changed to resource_manager_plan																																			
2018-05-08 14:19:14	Last tablespace created BIIC_DOSTF_2016																																			

An additional feature of the dashboard is **alerting** about various performance events on the server, such as:

- Increased CPU utilization on server
- Increased waits
- Locks
- Decrease in efficiency levels of request/requests
- Increase in the number of sessions or open transactions, etc.

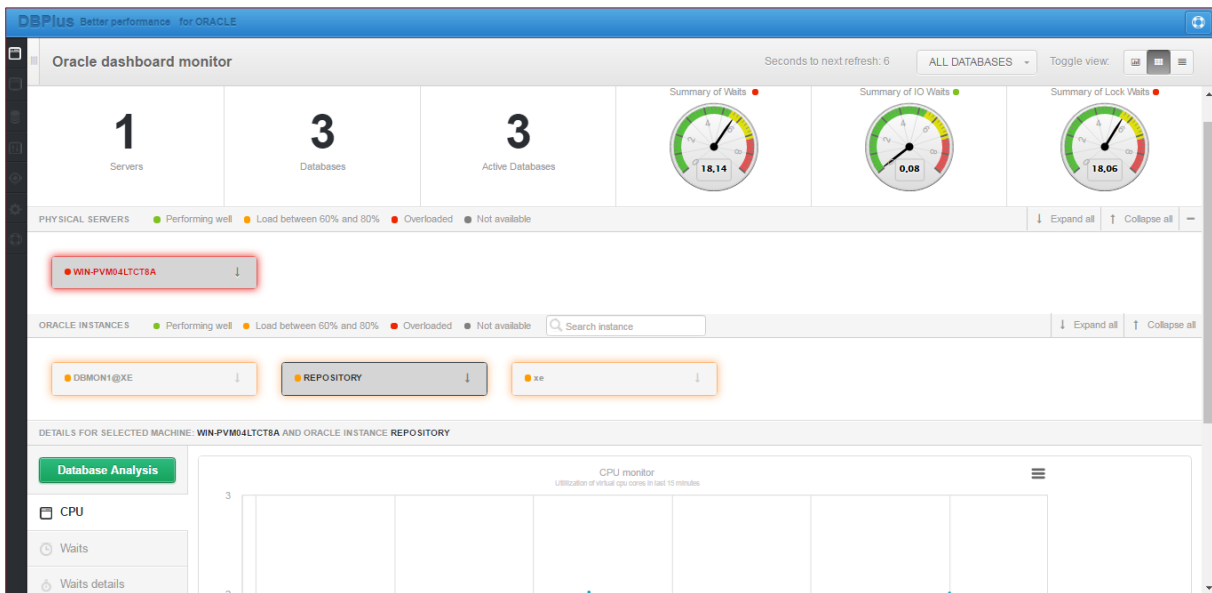
Clicking the [**Database space**] tab, information about current database size is divided by schemas (size in Megabytes). The data in the graph presents information about occupied space (Space Used), free space (Space Free) and about maximum file size the database can extend to (Note that there is no check if such space is available on the disk array).



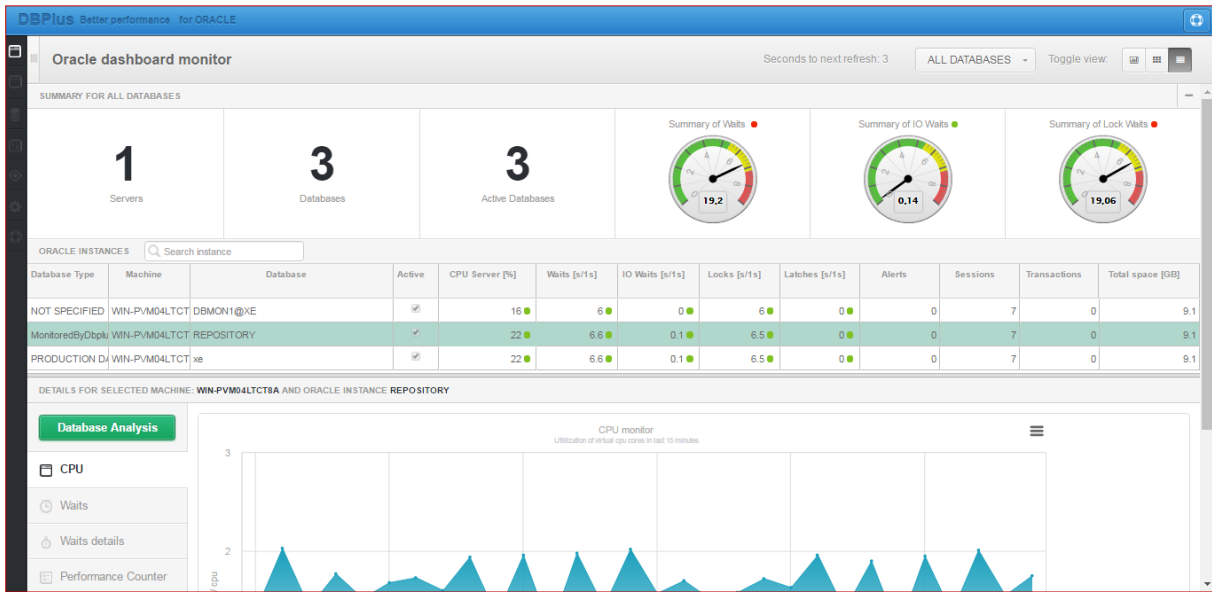
6.1.5 Dashboard – various forms of presentation

Dashboard is available in 3 modes, which are switched by clicking [Toggle View] icon in the top right corner. Available modes are:

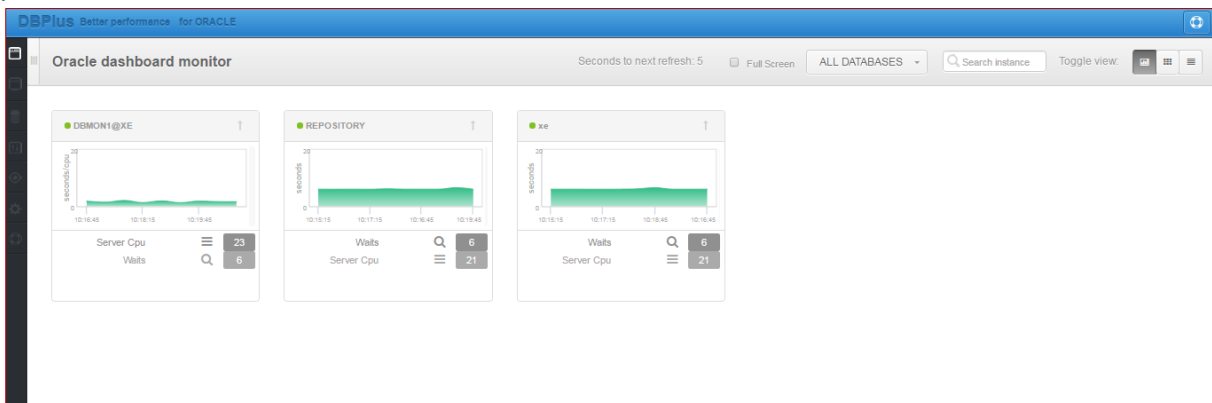
Icon View - showing monitored servers / databases in a form of icons (default)



Grid View - showing instances in a table



Television Mode - showing instances of Oracle in a form of icons with automatically switching performance indicators



Changes to the alerting and indicators appearing in the **[Television Mode]** can be made at the option Configuration -> Settings.

6.2 Menu Database Analysis

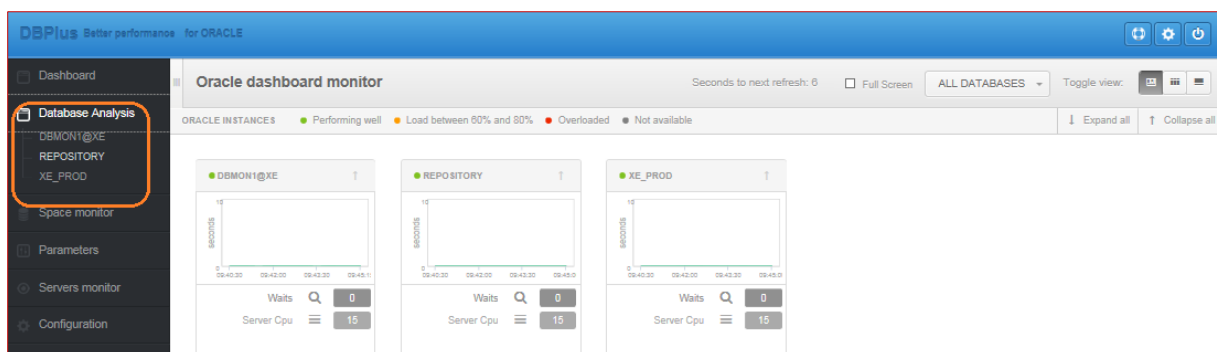
6.2.1 Menu Performance – Database Analysis

Dashboard of the DBPLUS System Performance Monitor allows you to track the performance of Oracle databases and show how it looked over the course of last 15 minutes or the last 24 hours. For a detailed analysis of the load at any given moment in time, and seek answers to questions like:

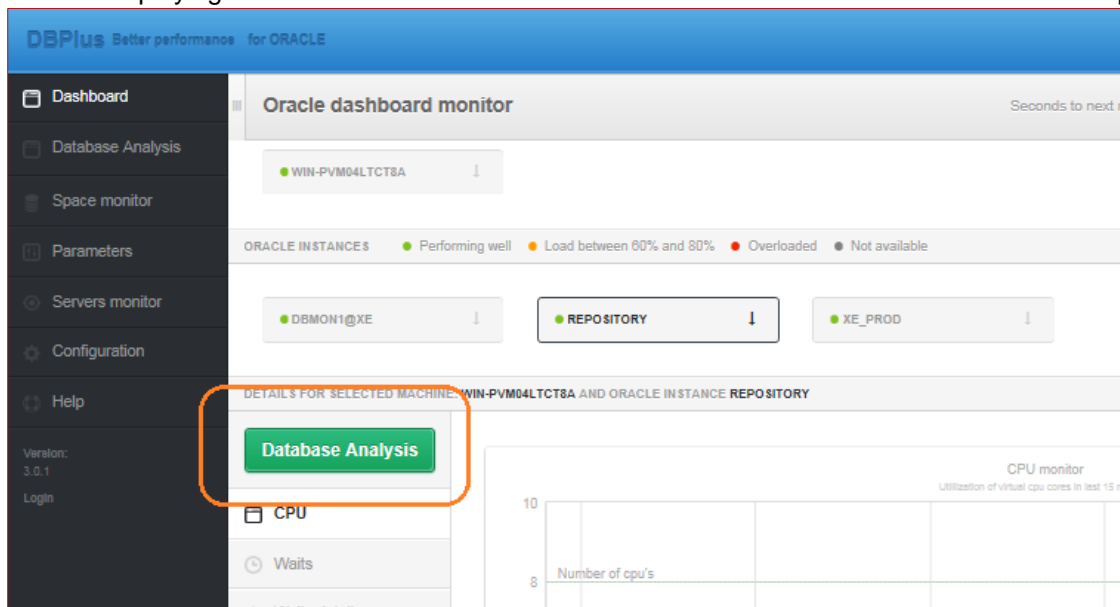
- why database is running slow?
- why User had problems in the application 3 days ago at 15:48?
- why my report took 15 minutes to perform?
- etc.

Users enter the module **[Database Analysis]** and have two possibilities:

- On the left side of the menu, clicking on the **[Database Analysis]** tab, a list of ORACLE databases is displayed



- Displaying the details of database after earlier database selection from the dashboard page



Activating performance module of a system, information about chosen database, its version and the time of the last restart are displayed on the bar:



- Format SQL text queries

On each page where the query is presented, a [Format SQL] button has been added. After pressing the button, the query text will be formatted.

The screenshot shows a table with columns: 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, and Rows processed. A 'Format SQL' button is highlighted in red in the bottom right corner of the interface.

- Grid manager on Load Trends/SQL details pages

We have introduced the function of change the settings for tables visible on application pages. The ability to change settings will be introduced in stages, each release will be followed by additional tables. This was first introduced on the Load Trends and SQL Details pages. The User for these tables on the pages can change for each of the columns:

- Order of displayed columns
- Visibility of columns
- Change the format
- Change of precision
- Change of width

Additionally, it is now possible to hide the Summary row on each page, using the settings available after press the "cog" icon. As before, the data contained in the grid can be freely exported to a file.

The order of displayed columns

To change the order of columns, click on the header of the column, hold down the mouse button, drag the columns and drop them to the desired place on the table.

Visibility of columns

To hide a column, right-click on the column header to be hidden. A popup menu will open where the Hide column button should be selected. The indicated column is hidden.

The screenshot shows a table titled 'POSTGRESQL TRENDS STATISTICS' with columns: Logdate, Elapsed Time, Rows, Executions, Blks hit, Blks read, Blks dirtied, Blks written, Temp blks read, Temp blks written, and Wait time. A popup menu titled 'Blks dirtied column properties' is open over the 'Blks dirtied' column header, with the 'Hide column' option highlighted in red.

To reveal a column, click the [cog] button in the upper right corner of the table. After the popup menu open, select the [Show hidden columns] option, then indicate the column you want to rediscover in the table. The uncovered column will appear last on the right side of the table.

Logdate	Elapsed Time	Rows	Blks hit	Blks dirtied	Temp blks read	Temp blks written	IO time	Active sessions	Sessions	Connectio...	Commits	Rollbacks	Tuples returned	Tuples fetched	Tuples inserted	Tuples updated	Tuples deleted	Conflicts
2020-03-09	77.11	279 140	1 466 907	21 084	20 747	20 720	0	0	10	221	34 014	2 574	15 619 ...	1 696 148	23 612	0	0	0

Change of data format / precision

To change the data format settings, precision, right-click on the column heading where you want to change the data. After making changes, save the changes by click [Apply] button.

Logdate	Elapsed Time	Rows	Blks hit	Blks dirtied	Temp blks read	Temp blks written	IO time	Active sessions	Sessions	Connecti...	Commits	Rollbacks	Tuples returned	Tuples fetched	Tuples inserted	Tuples updated	Tuples deleted	Conflicts
2020-03-02	4	16 120	16 510	0	0	0	0	0	0	1	87	10 974	47	893 425	46 707	0	0	0

Change of width

To change the column width, click the column edge, hold and move it to the right or left to change the width. The current solution used in the DBPLUS application adjusts the width of the columns to the width of the screen. Therefore, with many columns in the table, the width of the columns will always be converted in proportion to the width of the screen.

Logdate	Elapsed Time	Rows	Blks hit	Blks dirtied	Temp blks read	Temp blks written	IO time	Sessions	Tuples inserted	Tuples updated	Tuples deleted	Conflicts	No of temp files	Data written to temp	Deadlocks	Blk read time	Blk write time	Executions	Blks written	
2020-03-02	49.93	16 120	16 510	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	8 109	0

Storage of table configurations

The configuration for each of the tables is saved in two ways: at the browser cache level on the user's computer or in the repository database.

In order to permanently save the settings to the repository database, Windows authorization must be enabled in the DBPLUS Performance Monitor application (enabled at the Configuration Wizard level),

and the Security module (Menu Configuration> Settings: Security "ON") must be started. The settings are saved for all monitored instances for each user separately.

Restore default settings

If User need to return to the default settings, they can do this by click the cog button and select [Restore grid defaults].

Date	Plan Id	Elapsed Time [seconds]	Blocks read time [seconds]	Blocks write time [seconds]	Executions	Blocks hit [Blocks]	Blocks read [Blocks]	Blocks dirtied [Blocks]	Blocks written [Blocks]	Rows per 1 Exec	Blocks hit per 1 Exec [Blocks]
2020-03-16 10:09:01	2626426938	2.9	0	0	60	60	0	0	0	1.00	1.00
2020-03-16 10:39:21	2626426938	2.8	0	0	60	60	0	0	0	1.00	1.00
2020-03-16 10:24:11	2626426938	2.6	0	0	60	60	0	0	0	1.00	1.00
2020-03-16 13:10:11	2626426938	2.4	0	0	59	59	0	0	0	1.00	1.00

At any time, the User can restore the default setting for a given column by click on the [Restore defaults] button for a given column.

6.2.1.1 „Database Load” Tab

Database Load is a screen showing database load time, which was partially described in the previous section about the **Dashboard**.

In the **Performance** module the graph functionality is superior as it gives us options to:

- check the load of the database on the wider scale e.g. today, yesterday, a month or even 3.5 years ago.
- look at the SQL queries / commands, which generated the load
- asses what database was currently doing. If it performed any disk operations or whether there were locks, etc.

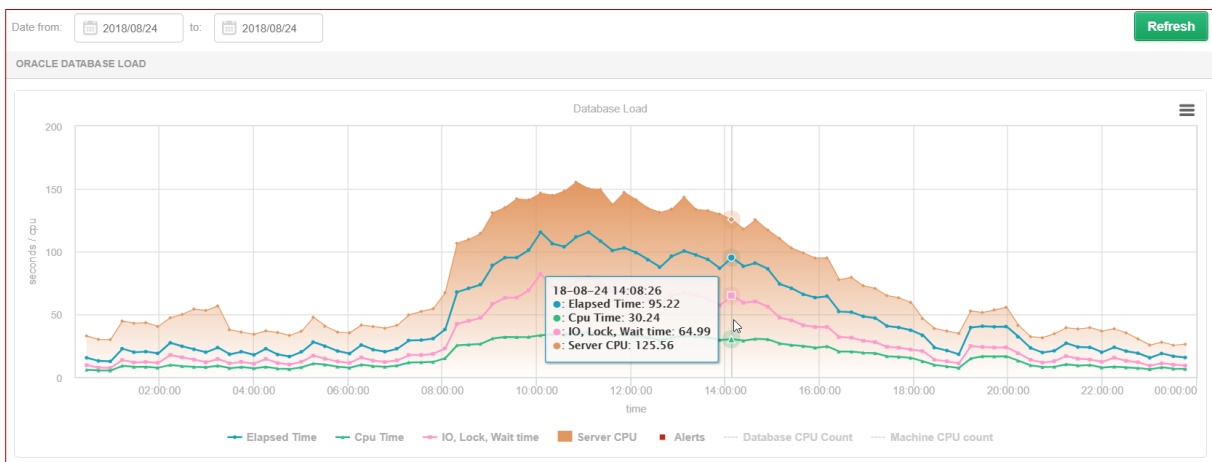
Database Load screen consists of following areas:

- filtration fields - fields of dates by which we define the period in which we want to check- the load
- the graph presenting the load
- the load information at a given moment of time:
 - list of queries with execution statistics
 - Waits - what database was doing at the time to perform queries

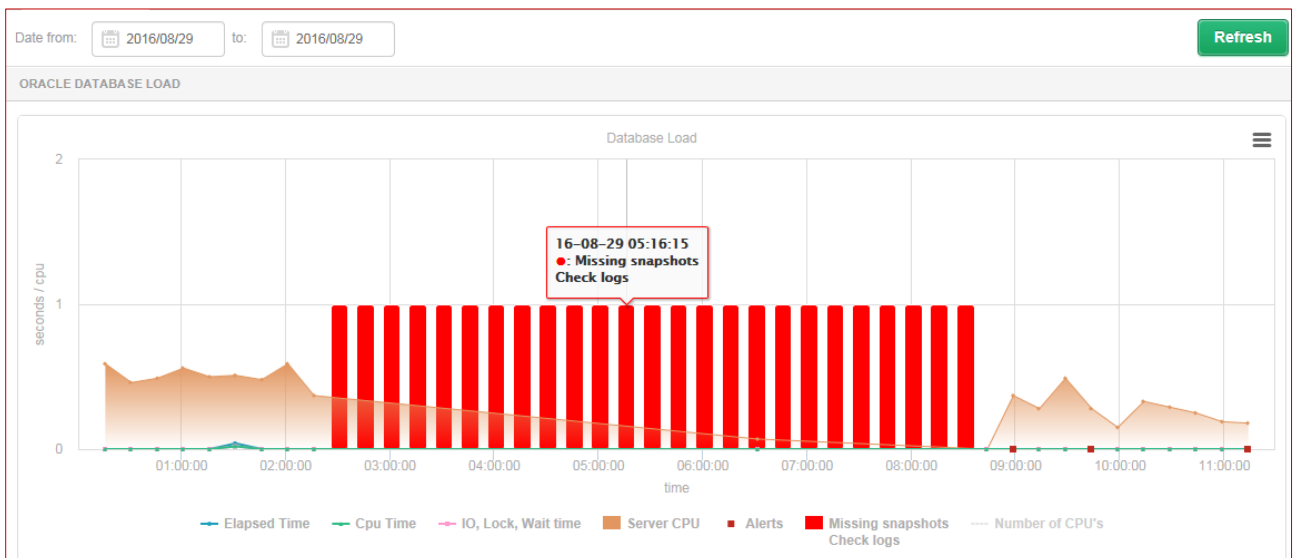
Chart consists of following series:

- **Elapsed Time** - shows the waiting time of all users on the query result in a given second of time
- **CPU Time** – the utilization of server processors by all queries in a given second in time.
- **IO Lock, Wait Time** - the result of the difference between the Elapsed Time and CPU time of database
- **Server CPU** - load of the server processors
- **Database CPU Count** – number of processors allocated to databases instances
- **Machine CPU count** – number of processors on the server (graph is hidden by default)
- **Alerts** – number of alerts at a given moment in time
- **Missing snapshots** – series occurring in case of monitoring service hasn't done its efficiency measurement (e.g. database is unavailable)

Example chart showing load for the selected day:



An example screen with no snapshots (2:30am to 8:30am)



The chart is "clickable" - click on the selected part / section will refresh the bottom of the screen with information about requests and waits that generated the data load.

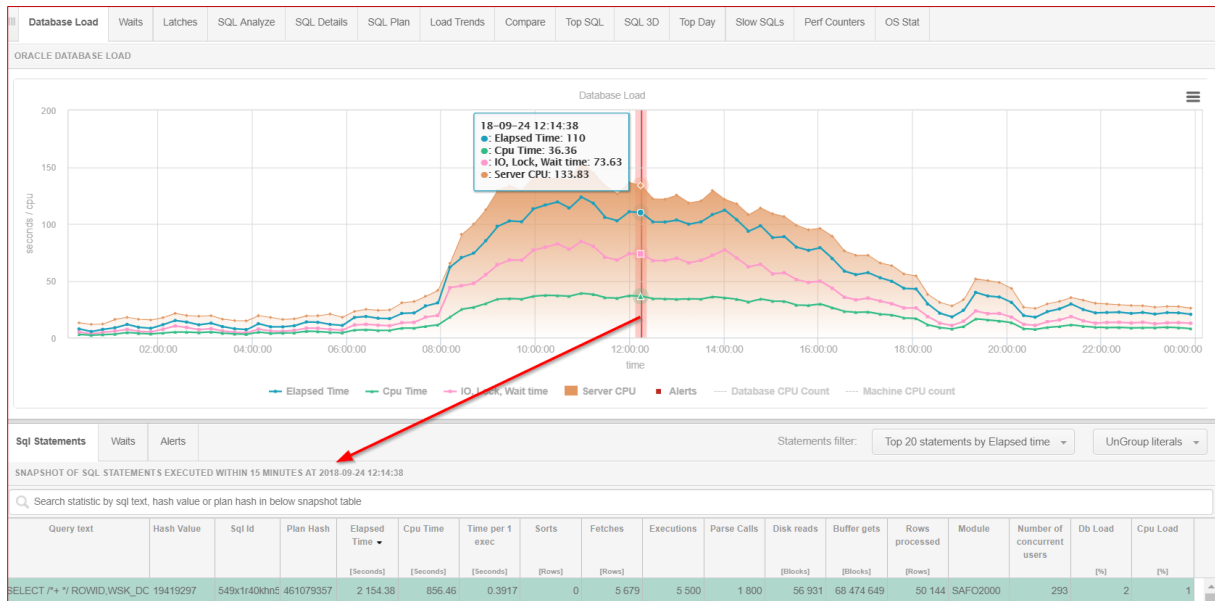
IMPORTANT: data for chart of the database load is calculated by monitoring service DBPLUSORACLECATCHER - a component of the DBPLUS Performance Monitor. Monitoring Service performs several procedures examining database performance. The result of the operation of these procedures is a snapshot (snap) that is created every 15 minutes.

If one session for 15 minutes performed 3 queries:

- The first query 5 minutes
- The second query 09 minutes and 59 seconds
- The third query 1 second

... the graph for a given situation will present utilization of one CPU with Elapsed Time close to 1 second.

After clicking on the selected point in time, the lower part of the screen is refreshed, with information's about queries and waits



If you scroll down the screen:

SQL Statements Waits Alerts

Statements filter: Top 20 statements by Elapsed time UnGroup literals

SNAPSHOT OF SQL STATEMENTS EXECUTED WITHIN 15 MINUTES AT 2018-08-27 13:50:32

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]	[Rows]	[Rows]	[Blocks]	[Blocks]	[Rows]			[%]	[%]
INSERT INTO ZES_000 (W50)	3869860768	aa2dbammak	4066142822	3 700.44	1 381.93	528 6338	221	0	7	7	472 376	500 498 02	42 147	SAFO200	3	4	2
SELECT ID, DECODE (DOK_Z	870226685	3rs2w5sbox6rx	1732560816	3 456.77	1 272.37	12 0866	286	30 777	286	286	232 219	171 819 46	91 738	w3wp.exe	3	4	1
SELECT DECODE (K FIR_KOI	2435395181	d20tc5f8kkbm	1818372388	1 938.37	355.61	0 6794	0	2 852	2 853	2 853	65 334	24 830 87	2 852	w3wp.exe	3	2	0
SELECT /*+ */ ROWID, WSK_D	19419297	549xtr40khn5	461079357	1 790.44	743.41	0 3730	0	5 001	4 800	1 646	2 537	61 748 70	49 334	SAFO200	290	2	1
SELECT ID, DECODE (DOK_Z	870226685	3rs2w5sbox6rx	1732560816	1 475.22	531.58	0 9278	1 590	22 633	1 590	1 590	154 892	71 043 06	64 712	w3wp.exe	2	2	1
-- WIDOK TOWARY (TJ, ML) --	1159534948	7rkh8bd2ju6b	2354673971	1 405.90	595.88	2 0767	0	3 494	677	323	255	25 703 66	114 802	ifrun60.exe	5	2	1
SELECT SUM (WAR) FROM (S	1059949311	fg7gbzczkv2rz	3490454522	1 361.31	521.37	0 0084	0	162 974	162 971	270	4 718	68 046 45	162 969	w3wp.exe	0	2	1

Three additional tabs appear:

- SQL Statements
- Waits
- Alerts

SQL Statements is the query statistics presented in the form of a table. By default, system displays top statements for the duration of Elapsed Time or CPU utilization. The display method can be changed after clicking on the field [Statement filter] - you can also view a complete list of queries that participated in the load.

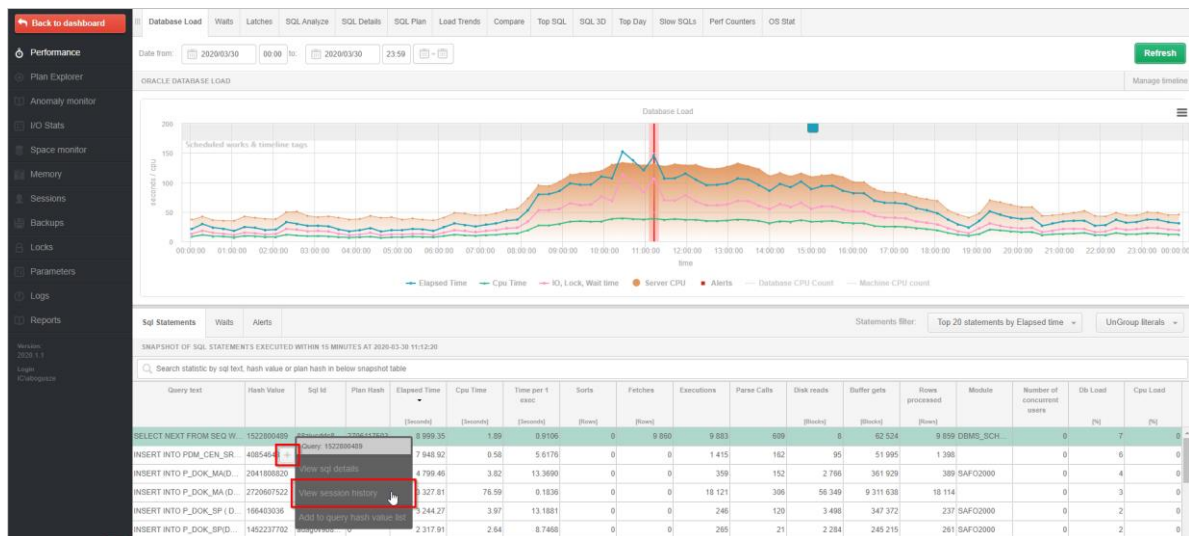
The table with queries:

- Can be sorted by any column
- Can be search by a specific text snippet
- Change size or width of individual columns
- Change the precision of the presented data in the table.

The table contains information collected from the database as part of the last snap (15 minutes). Information can be displayed in two modes: UNGroup literals and Group literals. Depending on the chosen variant, the table will present the following columns:

- Query Text - SQL command
- Hash value/sql_id - query identifier
- Plan Hash - execution plan identifier
- Elapsed Time [Seconds] - the duration in seconds for all query executions within last 15 minutes.
- CPU time (seconds) - CPU utilization time in seconds by the query during last 15 minutes.
- Time per 1 Exec [Seconds] – duration in seconds for one execution
- Sorts [Rows] - number of sorts performed by the query during last 15 minutes
- Fetches [Rows] – number of downloads from query cursor
- Executions – number of queries for the last 15 minutes
- Parse Calls – number of query plan calculation for a given query in the last 15 minutes
- Disk Reads [Blocks] - number of disk reads of the query for the last 15 minutes,
- Buffer Gets [Blocks] - number of buffers utilized by the query in last 15 minutes
- Rows processed [Rows] - number of rows returned by the query in last 15 minutes
- Module – the name of app/module activated by query
- Number of concurrent users – how many unique users activated a query
- Database Load (%) - the percentage of total database load caused by the query during for last 15 minutes.
- CPU Load (%) - the percentage in which given queries are loaded by the database server processors data in the last 15 minutes,

Hash value column (by each line presenting statistics) additionally shows **[+]** ("plus") button.



Clicking the **[+]** ("plus") button, an additional context menu is shown, which enables for detailed analysis of the selected query, which will be described in the section **"Performance SQL Details"**.

After pressing the **View session history** button, a session history window dedicated to the given query is opened. The window always opens in the context of a given day (sysdate). The User also has the option of applying a number of filters available so far on the session history screen.

A quick view session history screen significantly speeds up the analysis of the performance of a given query, enabling e.g. identification of the user performing the analyzed query.

For example, if you select **"Add to query hash list"** option, we move a query identifier to the clipboard with a list of queries for later analysis of specific queries.

Slide below presents functionality for adding queries to the analysis in **SQL Details**.

		Database Load	Waits	Latches	SQL Analyze	SQL Details	SQL Plan	Load Trend																																													
<p>Click on hash value to analyze Query Performance Details</p> <p>Query Hashes list</p> <ul style="list-style-type: none"> 3715327 77461126 3952919183 3532405831 3432012016 1982292445 2276889026 		<p>Search statistic by sql text, hash value or plan hash in below snapshot table</p> <table border="1"> <thead> <tr> <th>Query text</th> <th>Hash Value</th> <th>Sql Id</th> <th>Plan Hash</th> <th>Elapsed Time</th> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td>[Seconds]</td> </tr> </thead> <tbody> <tr> <td>INSERT INTO ZES_000 (W50,W1,W2,W3, W4</td> <td>3869860768</td> <td>aa2dbammak</td> <td>4066142822</td> <td>3 700.44</td> </tr> <tr> <td>SELECT ID, DECODE (DOK_ZA_ID_3, NULL,</td> <td>870226685</td> <td>3rs2w5stbx6rx</td> <td>1732560816</td> <td>3 456.77</td> </tr> <tr> <td>SELECT DECODE (K.FIR_KOD, 'CZY', 'HZA',</td> <td>2435395181</td> <td>d20tc5f8kkbm</td> <td>1818372388</td> <td>1 938.37</td> </tr> <tr> <td>SELECT /*+ */ ROWID,WSK_DO,NR,ROK,DA</td> <td>19419297</td> <td></td> <td></td> <td>1 790.44</td> </tr> <tr> <td>SELECT ID, DECODE (DOK_ZA_ID_3, NULL,</td> <td>870226685</td> <td></td> <td></td> <td>1 475.22</td> </tr> <tr> <td>-- WIDOK TOWARY (TJ, ML) -- IC_KARTOTE</td> <td>1159534948</td> <td></td> <td></td> <td>405.90</td> </tr> <tr> <td>SELECT SUM (WAR) FROM (SELECT WAR, Y</td> <td>105994931</td> <td></td> <td></td> <td>1 361.31</td> </tr> </tbody> </table>							Query text	Hash Value	Sql Id	Plan Hash	Elapsed Time					[Seconds]	INSERT INTO ZES_000 (W50,W1,W2,W3, W4	3869860768	aa2dbammak	4066142822	3 700.44	SELECT ID, DECODE (DOK_ZA_ID_3, NULL,	870226685	3rs2w5stbx6rx	1732560816	3 456.77	SELECT DECODE (K.FIR_KOD, 'CZY', 'HZA',	2435395181	d20tc5f8kkbm	1818372388	1 938.37	SELECT /*+ */ ROWID,WSK_DO,NR,ROK,DA	19419297			1 790.44	SELECT ID, DECODE (DOK_ZA_ID_3, NULL,	870226685			1 475.22	-- WIDOK TOWARY (TJ, ML) -- IC_KARTOTE	1159534948			405.90	SELECT SUM (WAR) FROM (SELECT WAR, Y	105994931			1 361.31
Query text	Hash Value	Sql Id	Plan Hash	Elapsed Time																																																	
				[Seconds]																																																	
INSERT INTO ZES_000 (W50,W1,W2,W3, W4	3869860768	aa2dbammak	4066142822	3 700.44																																																	
SELECT ID, DECODE (DOK_ZA_ID_3, NULL,	870226685	3rs2w5stbx6rx	1732560816	3 456.77																																																	
SELECT DECODE (K.FIR_KOD, 'CZY', 'HZA',	2435395181	d20tc5f8kkbm	1818372388	1 938.37																																																	
SELECT /*+ */ ROWID,WSK_DO,NR,ROK,DA	19419297			1 790.44																																																	
SELECT ID, DECODE (DOK_ZA_ID_3, NULL,	870226685			1 475.22																																																	
-- WIDOK TOWARY (TJ, ML) -- IC_KARTOTE	1159534948			405.90																																																	
SELECT SUM (WAR) FROM (SELECT WAR, Y	105994931			1 361.31																																																	

Below the table with a list of queries a full text block is presented for the query with the execution plan. Clicking on query will refresh these fields.

<p>STATEMENT TEXT FOR QUERY HASH: 232111811</p> <pre>select * from test where id = 1000 for update</pre>
<p>EXPLAIN PLAN FOR PLAN HASH: 2283087408</p> <p>Show plan objects</p> <pre> └SELECT STATEMENT (Cost = 1026 , Bytes = 0 , Cardinality = 0 , Search Columns = 0) └FOR UPDATE └BUFFER (SORT) └TABLE ACCESS (FULL) TEST (Cost = 1026 , Bytes = 2301 , Cardinality = 177 , Search Columns = 0) </pre>

Under the Explain plan – a link is available for functionality which enables its analysis, and analysis of objects involved in the query, such as:

- what tables, indexes participated in the execution of the query
- how the engine called the given objects
 - data search (seek)
 - read complete data (index or table scan)
- whether the query was carried out in multi-threaded mode
- what mechanism was used to download and connect data from objects:
 - Nested Loop
 - Hash/Merge Join

Clicking the **[Show Plan Object]** link, presents User with a form of used queries.

SQL TEXT		EXPLAIN PLAN	
<pre>select ofe_d_id from ofe_d where ofe_d_gru_k_kod in (select gru_k_kod from kh_gru_k where kh_kod=ipR8Rod) and ofe_d_fir_kod=FiR8d and ofe_d_dat_d_id <= :!dat1 and ofe_d_wat_id = '0' and sp_spl_ofe_sta('990', ofe_d_ofe_sta_kod) = 1 and (ofe_d_ofe_d_id is null) or ofe_d_ofe_d_id >= :!dat2)</pre>	<pre>SELECT STATEMENT (Cost = 15, Bytes = 0, Cardinality = 0, Search Columns = 0) NESTED LOOPS (SDM) (Cost = 0, Bytes = 33, Cardinality = 1, Search Columns = 0) TABLE ACCESS (BY INDEX ROWID) OFE_D (Cost = 7, Bytes = 23, Cardinality = 1, Search Columns = 0) INDEX (FULL SCAN) OFE_D_GRU_K_FK_I (Cost = 4, Bytes = 0, Cardinality = 817, Search Columns = 0) INDEX (UNIQUE SCAN) KH_GRU_K_FK_I (Cost = 2, Bytes = 0, Cardinality = 1, Search Columns = 2)</pre>		
OBJECTS USED IN EXPLAIN PLAN		INDEXES FOR SELECTED OBJECT INTER.OFE_D	
Type	Owner	Object Name	Name
TABLE	INTER	OFE_D	OFE_D_FIR_DB
INDEX	INTER	OFE_D_GRU_K_FK_I	OFE_D_GRU_T_FK_I
INDEX	INTER	KH_GRU_K_FK_I	OFE_D_TYP_D_FK_I
TABLE	INTER	KH_GRU_K	OFE_D_GRU_R_FK_I
			OFE_D_LIM_FK_I
			OFE_D_R2B_FK_I
			OFE_D_IM_FK_I
			OFE_D_GRU_R_FK_I

Object columns									
Details for TABLE INTER OFE_D									
Columns	Type	Length	Column Id	Unique values	Density	Last analyzed	Sample size		
ID	NUMBER	22	1	263 367	0.00000389	2019/07/08 07:06:36	263 367		
OFE_STA_KOD	VARCHAR2	6	2	6	0.00000168	2019/07/08 07:06:36	6 091		
FIR_KOD	VARCHAR2	6	3	29	0.00000188	2019/07/08 07:06:36	6 093		
ISO_KOD	VARCHAR2	6	4	2 178	0.00217391	2019/07/08 07:06:36	6 091		
DAT_W	DATE	7	5	4 309	0.00007166	2019/07/08 07:06:36	6 091		
DAT_D_ID	DATE	7	6	261 504	0.00000041	2019/07/08 07:06:36	6 091		
NOK	NUMBER	22	7	19	0.00000109	2019/07/08 07:06:36	6 092		
NR	NUMBER	22	8	8 138	0.00151676	2019/07/08 07:06:36	6 091		

In **[Show Plan Objects]** we have information about the query text and the explain plan. Below we see areas such as:

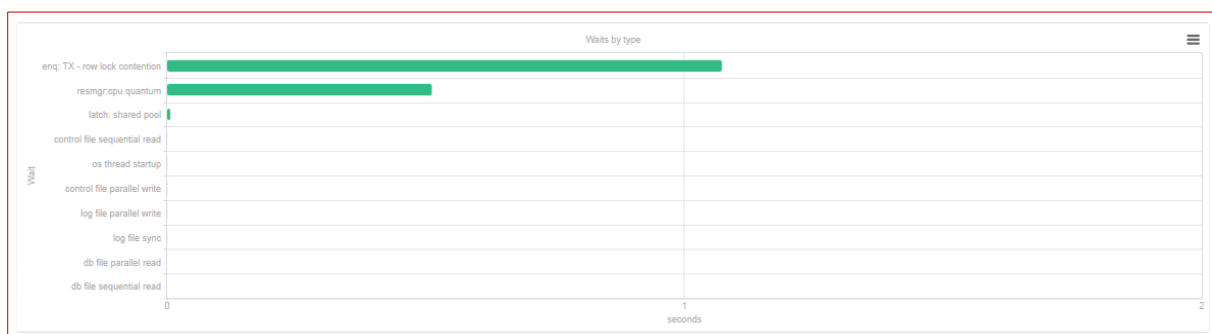
- **Objects Used in Explain Plan** – a list of all objects used by the query in the explain plan
- **Indexes for selected object** – list of indexes for selected table - row selected in the "Objects Used in Explain Plan" consists of 3 bookmarks:
 - a. **Object Columns** – a list of the individual columns of the selected object, along with information such as: column name, data type, column id, density (the lower density, the higher selectivity of the column)
 - b. **Info** – basic information about selected object (DDL info)
 - c. **Properties** – additional properties of selected object

Info tab and Properties Tab are visible when checkbox "Load object properties (slower)" is selected.

When analyzing the explain plan, we pay attention to:

- **Limiting the choice of data, or of the data with the where clause and table joins**
- **Whether the request is with parameters or literals**
- **The operation the database engine chose to retrieve/download data**
- **Whether the table has appropriate indexes**
- **Way of reading the data - Nested Loops vs. Hash Join**

Another tab (next to **[SQL Statements]**) is **[Waits]**. Waits presented here are shown in graphical and table form. The graph shows the duration for each second of the selected snapshot (of 15 minutes) for each type of wait that occurred at the time of the instance.



A table is located below the graph with following columns:

- Name - the name of wait
- Wait time - per 1 second (sec.) - Duration of wait type in seconds
- Total wait time for snapshot (sec.) - the total duration of wait type in the snapshot (15 minutes)

Name	Wait time per 1 second (Seconds)	Total wait time for snapshot (Seconds)
db file sequential read	10.3793	9 341.410
TCP Socket (KGAS)	8.8111	7 930.000
log file sync	2.7333	2 459.980
db file parallel write	1.5906	1 431.500
log file parallel write	0.7917	712.540
latch free	0.5559	500.300
enq: TX - row lock contention	0.4630	416.710
library cache: mutex X	0.3036	273.270
db file scattered read	0.3029	272.640

Next tab is **Alerts**. A list of alerts that occurred during this snapshot.

Logdate	Alert name	Message
2016/11/16 14:07:18	Elapsed Time	Alert Type: Load Trends, Alert level: CRITICAL, The measured statistic value is 30,6 times higher than average , Last value: 5462 s, Reference history value: 172,8 s
2016/11/16 14:07:18	Free tablespace size	Alert Type: Tablespace size, Alert level: CRITICAL, The free space for tablespace TB_ERP is 0 MB, Current used space: 1000 MB, Reference total max size: 999 MB
2016/11/16 14:07:18	Free tablespace size	Alert Type: Tablespace size, Alert level: CRITICAL, The free space for tablespace DBPLUS is 5 % of total size, Current used space: 5,1 GB, Reference total size: 5,4 GB
2016/11/16 14:07:18	Free tablespace size	Alert Type: Tablespace size, Alert level: CRITICAL, The free space for tablespace SYSTEM is 1 % of total size, Current used space: 394 MB, Reference total size: 400 MB
2016/11/16 14:07:18	Free tablespace size	Alert Type: Tablespace size, Alert level: CRITICAL, The free space for tablespace TB... current used space: 1000 MB, Reference total size: 1000 MB
2016/11/16 14:07:18	New Statement Elaps	Alert Type: Sql Query, Alert level: CRITICAL, Statement hash value: 23211811... asped Time, Last value: 5443 , The measured statistic value has 99,7 % of databas

If alert is about performance problem for specified SQL Statements, we can analyze it on **SQL Details** screen.

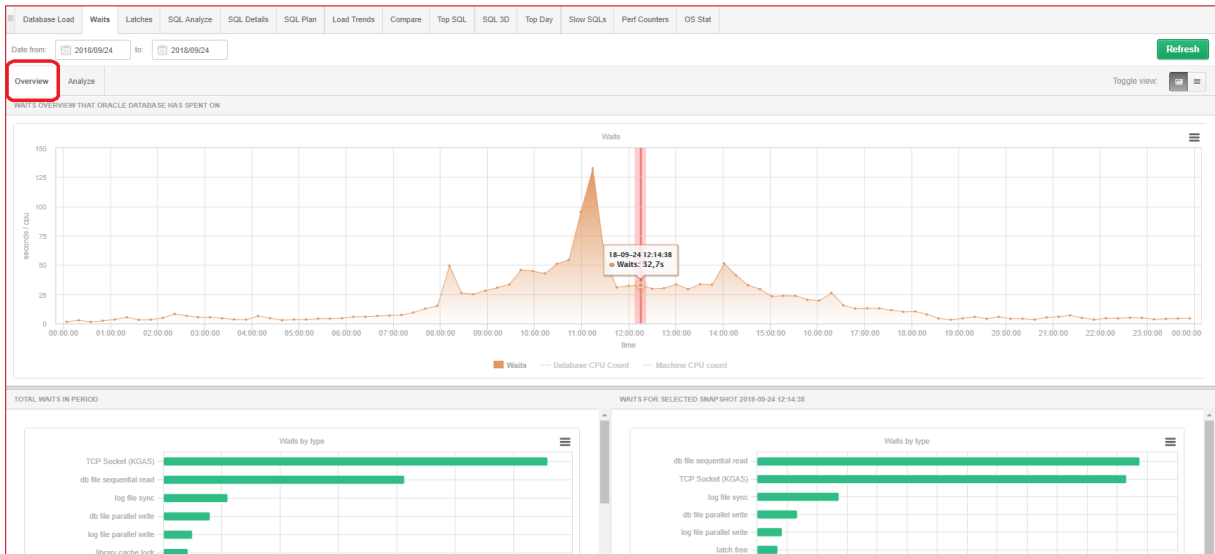
6.2.1.2 Waits Tab

Waits tab shows the duration of waits, which occurred at a time for all sessions on ORACLE database. Depending on the selection of **[Toggle View]** option, data can be shown graphically or in table form.

Waits screen in a similar way to **[Database Load]** screen, consists of the following fields:

- filtration fields - fields of dates by which we define the period in which we see database' waits
- graph presenting the level of waits for specific time (on the left) and for snaps (on the right).
- detailed information about waits in a given moment of time

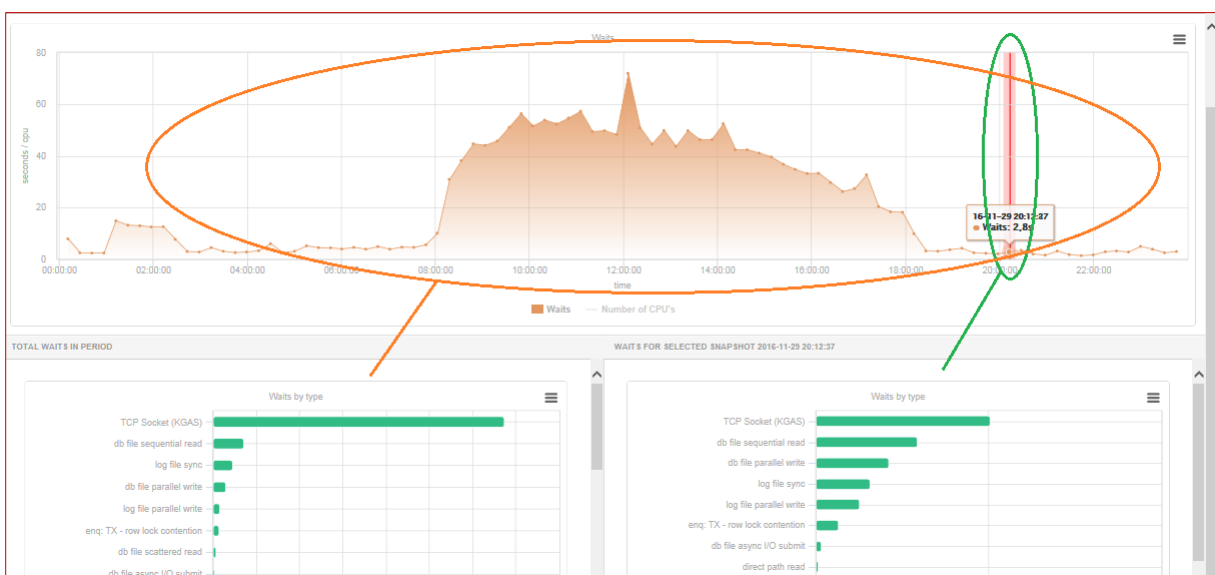
The Y axis of the graph illustrates (in seconds for a given second) time of all waits that occurred during the period shown on the X-axis. X-axis of the graph shows the period in which waits occur.



After switching the view with **[Toggle view]** button we get detailed information about the duration of the type of wait.

TOTAL WAITS IN PERIOD			WAITS FOR SELECTED SNAPSHOT 2018-09-24 12:14:38		
Wait	Total wait time in period [seconds]		Wait	Total wait time in period [seconds]	
db file sequential read	370 474.920		TCP Socket (KGAS)	15.522	
TCP Socket (KGAS)	347 564.720		db file sequential read	14.564	
log file sync	116 584.950		log file sync	11.643	
db file parallel write	73 077.010		db file parallel write	1.547	
log file parallel write	33 076.150		read by other session	1.499	
latch free	19 525.550		log file parallel write	0.824	
read by other session	18 036.710		enq: TX - row lock contention	0.754	
library cache: mutex X	17 202.720		latch free	0.468	
enq: TX - row lock contention	15 654.810		db file scattered read	0.329	
db file scattered read	8 542.730		library cache: mutex X	0.312	
direct path read	7 014.140		direct path read	0.281	

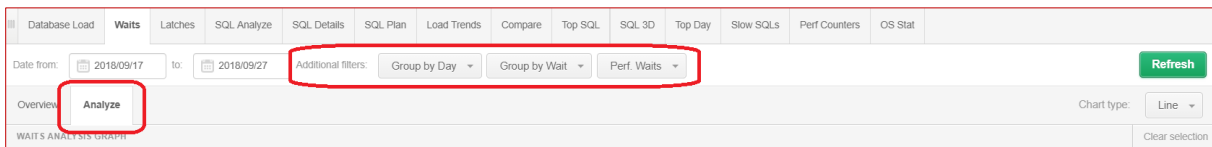
Just like in the screen **[Database Load]**, **[Waits]** chart is "clickable". Clicking on the part of the graph (its point) will show us waits summary, appropriate for a snapshot in time.



From the above chart we can see:

- what the database doing during the day (the default) or a selected period limited by dates in the filter
- what database doing during the last snap

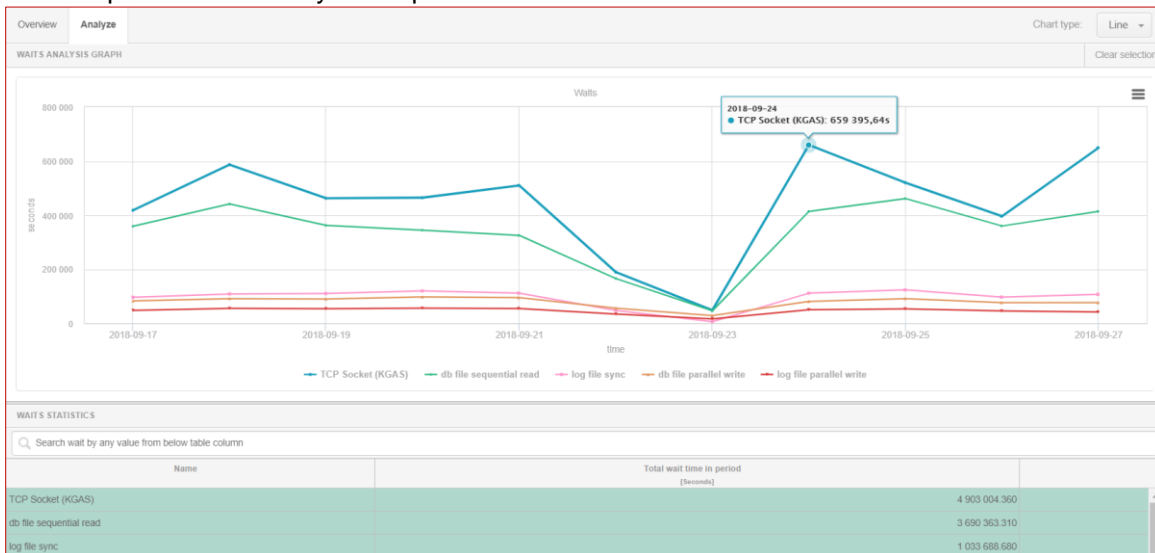
The system also allows to analyze individual Waits - the frequency, length and time of occurrence. To do this, click on the subtab **[Analyze]**:



On this page User can analyze performance waits or all of waits grouped by:

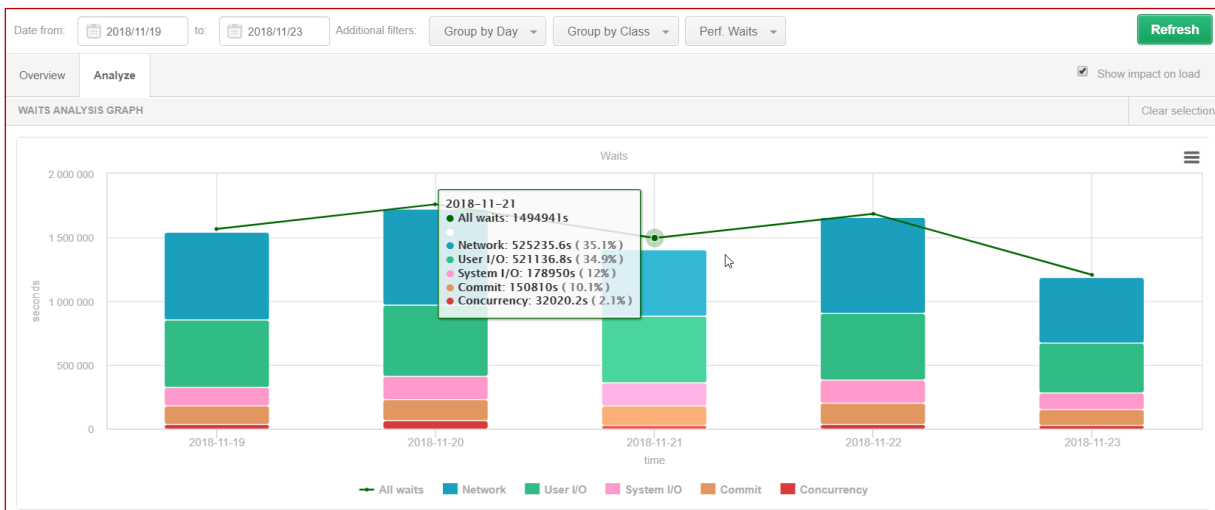
- Perf. Waits or All Waits,
- day, hour, snapshot.
- Group by Wait Class

An example results of analysis we present below:



Data is transferred on the chart area by clicking on the selected wait's row or dragging it to the empty field below.

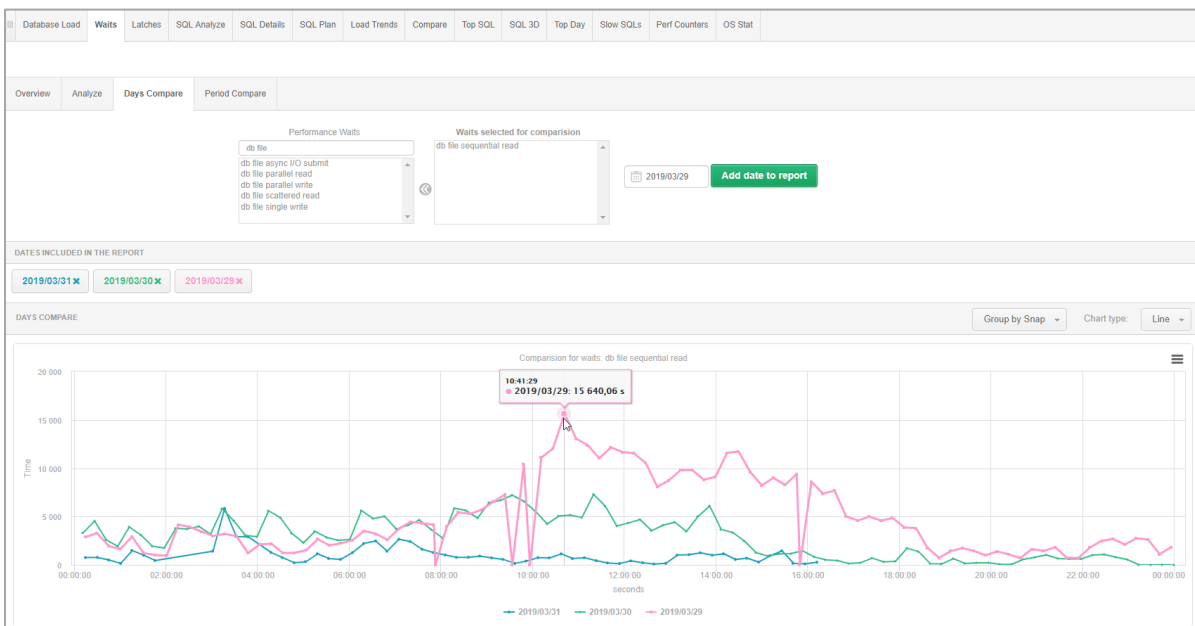
A view showing the class of top waits divided into days:



In the next two tabs, it is possible to make a waits comparison. There are two comparison modes to choose from:

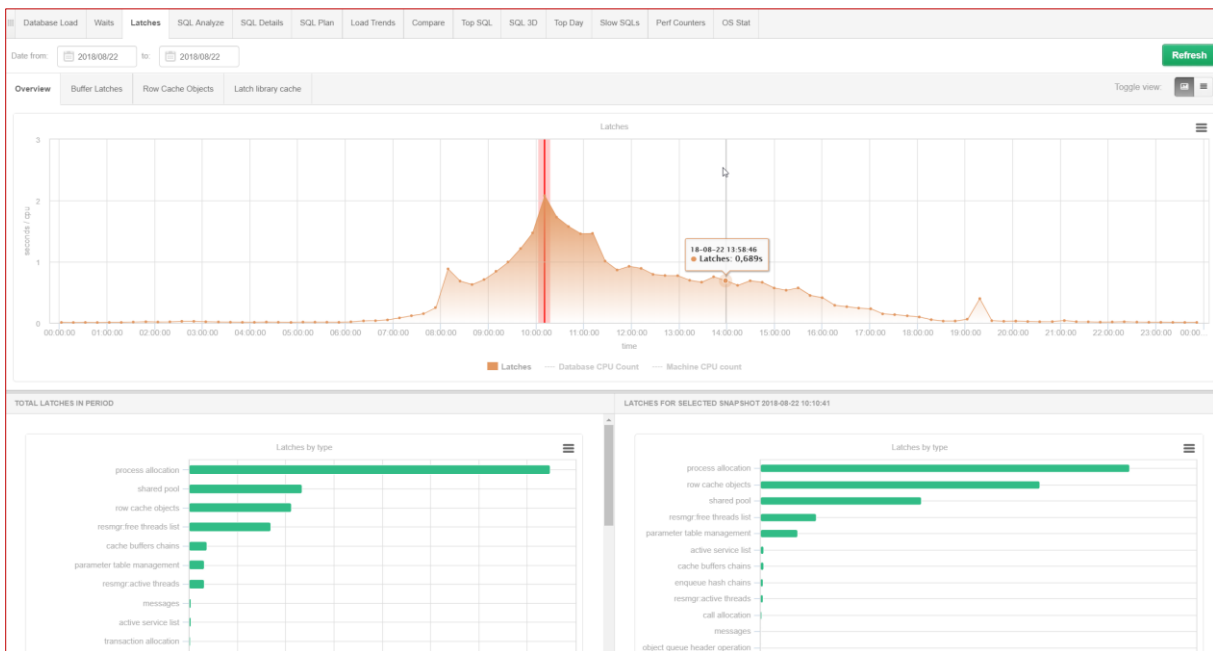
- Days Compare
- Period Compare

To compare wait level, select the type of wait to be compared first (one or more types), then select specific days to compare (Days Compare) or whole period of days (Period Compare).



6.2.1.3 Latches Tab

The "Latch" tab shows the duration of all LATCHES in seconds, which occurred at a given time for all user' sessions of ORACLE database.



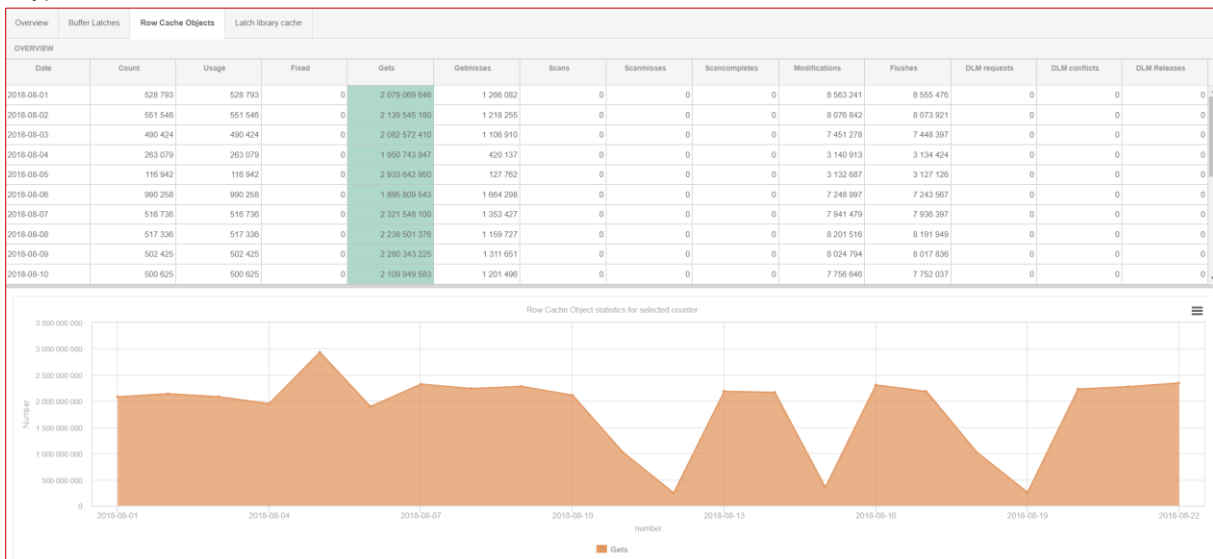
Latch screen maintains almost the same functionality as the Waits screen (click-through charts, all latches presentation and latches at a specific point in time).

The Y axis graph shows the time in seconds of all LATCHES that occurred at the time shown on the X-axis. The X-axis of the graph shows the time in which LATCHES occur. The chart is active and refreshes every 15 minutes by clicking the **[Refresh]** button.

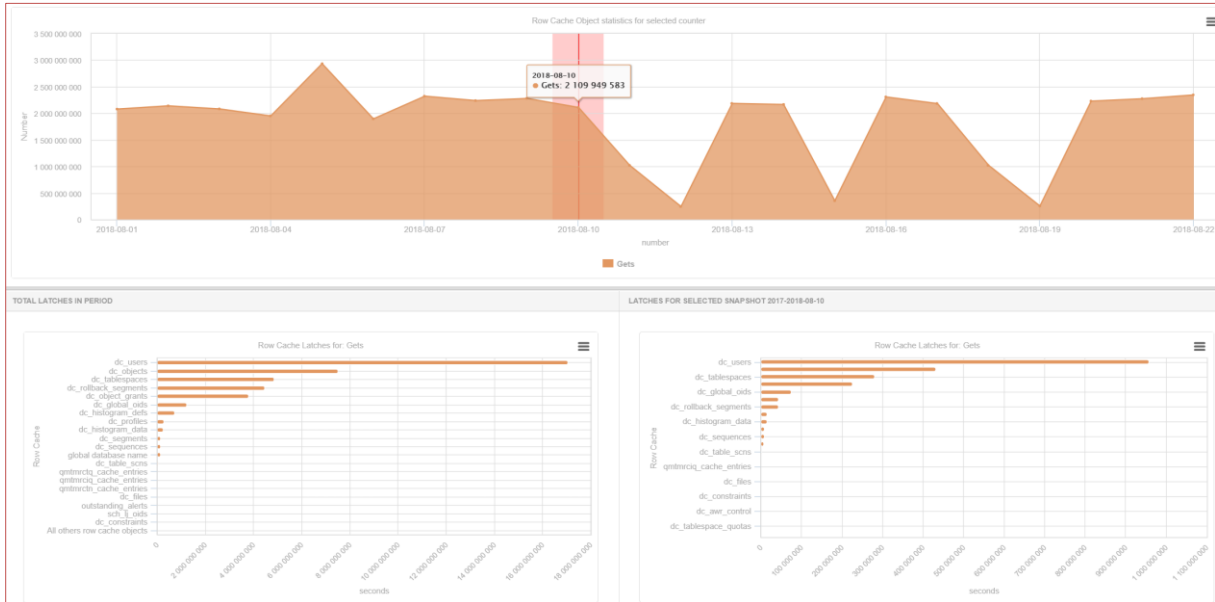
Additionally, the [Latch] screen provides following functionality:

- Buffer Latches - analysis of queries that most utilize memory area
- Row Cache Objects - latches broken down by operations related to SHARED_POOL
- Latch Library Cache - allows you to check how much space is occupied by queries in the SHARED_POOL memory buffer

For example, the Row Cache object analysis in a weekly horizon (period of 8 days in grouping after day):



Clicking on a specific statistic: [Count], [Usage], [Gets] refreshes the graph. As with the charts of Database Load, Waits, Latch, this chart is also "clickable" - clicking on the point will refresh the following charts below.



The "Latch library cache" tab allows you to check how much space queries occupy in the memory buffer SHARED_POOL.

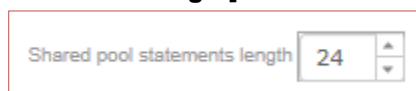
Shared pool statements length Enable Auto Shared Pool History Show statement version Refresh

Overview | Buffer Latches | Row Cache Objects | **Latch library cache**

SHARED POOL STATEMENTS			STATEMENTS	
SQL Text	Memory usage [bytes]	Count	SQL Text	
SELECT /*+rule*/ SYS_XML	17 613 872	2	/* SQL Analyze(1) /* select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	<div style="border: 1px solid gray; height: 100px; width: 100%;"></div>
MERGE /*+ dynamic_sampli	10 470 736	8	/* SQL Analyze(1) /* select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
/* SQL Analyze(1) /* sel	15 476 987	323	/* SQL Analyze(1) /* select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
			/* SQL Analyze(1) /* select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
			/* SQL Analyze(1) /* select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
			/* SQL Analyze(1) /* select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
			/* SQL Analyze(1) /* select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
			/* SQL Analyze(1) /* select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
			/* SQL Analyze(1) /* select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
			/* SQL Analyze(1) /* select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	

STATEMENTS HISTORY			HISTORY CHART	
Snap date	Memory usage [bytes]	Count		
2016-11-30 16:49:55	15 476 987	323	<div style="border: 1px solid gray; height: 100px; width: 100%;"></div>	

Option [Shared Pool Statements length] – allows to set the length of the query.



Setting the query length to check how many bytes in the SHARED_POOL buffer occupies the same query with the accuracy of the query length set. Queries will be grouped if they are the same as for the given length. This is especially useful for checking how much space takes up queries in SHARED_POOL containing LITERALS instead of variables. All you must do is set the length of a query to a value which ensures that the queries are the same for a given length. Then the following statistics will be presented for such queries for the requested length:

SHARED POOL STATEMENTS		
SQL Text	Memory usage [bytes]	Count
SELECT /*+rule*/ SYS_XML	17 613 872	2
MERGE /*+ dynamic_sampli	10 470 736	8
/* SQL Analyze(1) */ sel	15 476 987	323

The table contains following information:

- Partial SQL text - Text of the query to the desired length (in the above example, 24 characters)
- Memory usage (bytes) - the amount of memory used in bytes in the SHARED_POOL buffer
- Count - the number of existing versions of the query in the SHARED_POOL buffer

„Show Statement Version” checkbox enables or disables the view of all versions of the queries searched, for a given query length located in SHARED_POOL.

Show statement version

Depending on whether the checkbox "Show Statement Version" is selected or not - two types of data are shown:

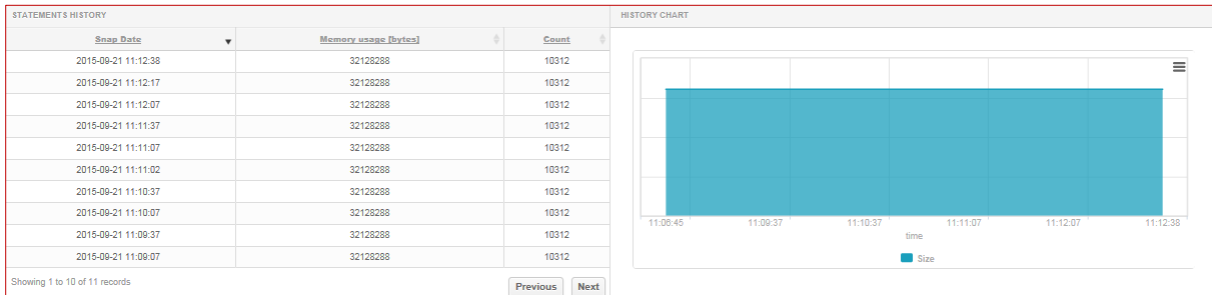
- Statements (checkbox "Show Statement Version" is selected) - the full content of the queries that are in the set length is shown

STATEMENTS	
SQL Text	
/* SQL Analyze(1) */ select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	▲
/* SQL Analyze(1) */ select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
/* SQL Analyze(1) */ select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
/* SQL Analyze(1) */ select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
/* SQL Analyze(1) */ select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
/* SQL Analyze(1) */ select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
/* SQL Analyze(1) */ select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
/* SQL Analyze(1) */ select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
/* SQL Analyze(1) */ select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	
/* SQL Analyze(1) */ select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_stats cursor_sharing_e...	▼

- Growing Statement (checkbox "Show Statement Version" is not selected) – following is show:
 - Statement – query text up to the set length
 - Difference in size between snaps (bytes) - The difference in the size occupied by the SHARED_POOL buffer for all queries that have changed the occupancy between refreshes of the screen.

GROWING STATEMENTS	
SQL Text	Diference in size between snaps [bytes] ▼
select xmlelement("v11:PublishDocumentRequestEBM",(select xmlagg(xml	13195588
select nvl(sum(nvl(decod	4524440

Statement History - area is divided into two parts



The first part shows a list of changes in the size of used memory in bytes in the SHARED_POOL and the number of versions of the queries existing in the buffer.

The second part is a graph which displays the size of used memory, where:
 The X-axis represents the time at which the query caused SHARED POOL cache utilization. The Y-axis represents the size of the SHARED POOL buffer used in bytes by the query.

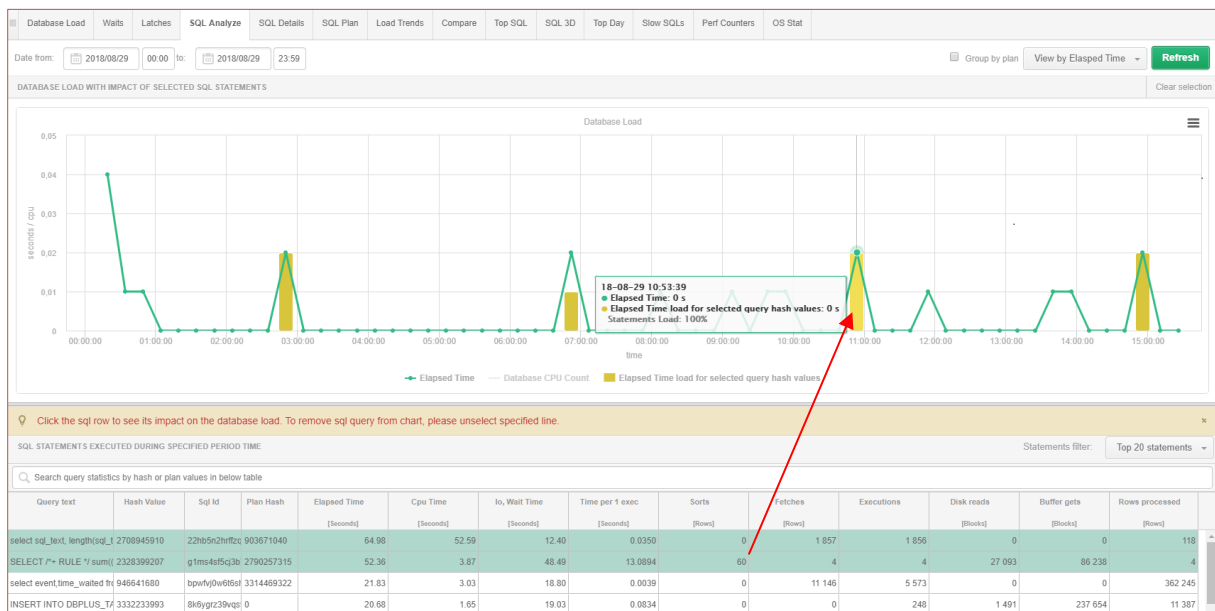
After selecting the checkbox "Enable Auto Shared Pool History", area "Statement History" refreshes automatically every 30 seconds.

6.2.1.4 SQL Analyze Tab

SQL Analyze functionality presents an additional view of database load. As with the Database Load chart, graph here also shows the utilization of the base.

The screen consists of the following areas:

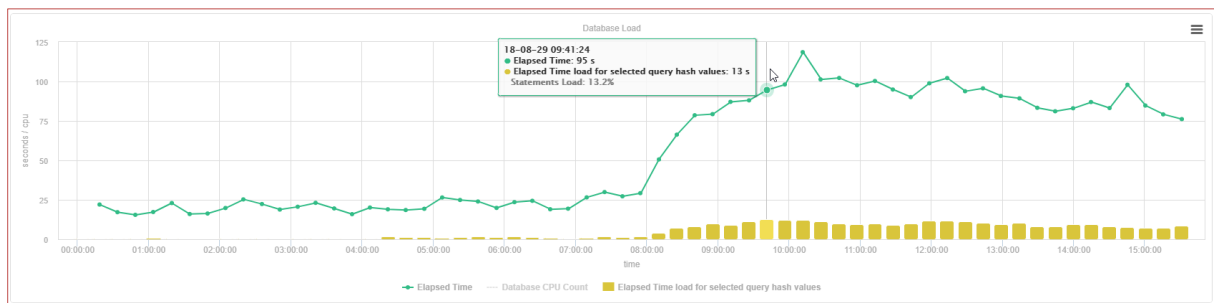
- filtration fields:
 - date and time fields by which the time is determine in which users can to familiarize themselves with the instance load
 - way of the presentation of the load - by CPU Time or Elapsed Time
- the graph shows the CPU load or Elapsed Time (depending on the choice in the filter)
- cumulative statistics:
 - with Group by plan option unchecked – broken down into queries that generated a specific load in a given period
 - with Group by plan option checked – broken down into performance plans that generated a specific load in a given period



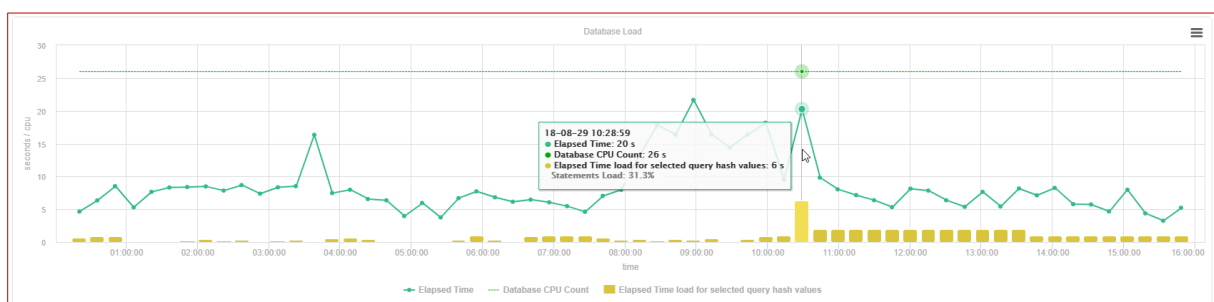
Graphs Y-axis shows the number of seconds for each second of duration of the query in ORACLE database.

The X-axis represents the time at which the query caused the utilization of database server. Differences that can show up between the load shown in the Database Load graph, and utilization statistics of the database server from the operating system side, arise due to including in the chart all kinds of waits, which is not shown in the operating system. The graph shows a full picture or performances, not just time.

After clicking on certain number of queries, Users can see their share of database load and when it took place with an accuracy to 15 minutes:



When selecting the Database CPU Count on chart – additional information about the extent to which the machine this database operates is utilized:



The table in the SQL Statements tab shows statistics for each query:

- Query text – content of the query
- HashValue – the query ID
- Sql ID – the query ID
- Plan Hash – execution plan ID
- Elapsed Time (sec) – summary of time in seconds of query duration for all queries executions taking place in the selected time. If the filter is marked "Group by Plan," then the Time of duration (sec) indicates the duration of all searches for the PLAN HASH VALUE for a selected period.
- CPU time (sec) – summary of time in seconds of CPU utilization for all executions of the query taking place in the selected time interval. If the filter is marked "Group by Plan," time of duration (sec) indicates the duration of all searches for the HASH VALUE PLAN for a selected period.
- I/O Wait Time [Seconds] – total time in seconds of waiting for I/O for all query executions taking place in the selected time interval. If the "Group by Plan" filter is selected, then Time of duration (sec) shows the duration of all queries for a given PLAN HASH in the selected time interval
- Sorts [Rows] - The number of sorts performed by the query in the selected time
- Fetches [Rows] - The number of returned records
- Executions - The number of query executions in the selected time
- Disk Reads - The number of disk reads for a query in the selected time
- Buffet Gets - The number of buffers utilized for a query in the selected time
- Rows processed - Number of rows returned by the query in the selected time

IMPORTANT: SQL Analyze screen maintains similar functionality to the Database Load:

- Clicking on a query row (in addition to showing the load on the graph) will display the full text of the query and its execution plan
- Next to the query identifier the [+] [Plus] button is located, which adds a query to the clipboard with a list of queries
- If the Group by plan option is checked - the [Plus] button adds the hash plan identifier to the SQL Plan screen

Database Load Waits Latches SQL Analyze SQL Details SQL Plan Load Trends Compare Top SQL SQL 3D Top Day Slow SQLs Perf Counters OS Stat

SQL STATEMENTS EXECUTED DURING SPECIFIED PERIOD TIME Statements filter: Top 20 statements

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]
INSERT /*+ append parallel 686750865	97xxthkmbz6	1948097753		8 048.14	5 308.56	2 739.58	8 048.1365	0	0	0	4 375.274	155 093 874	0
INSERT INTO F_ALGO_M_394574216	330bpmbs9c	3773095541		6 267.91	1 337.89	4 938.02	6 267.9146	0	0	1	2 353.729	138 075 563	21 766 690
call ETL_DEK_CODZIENN 409619430	Query 394574216			5 814.32	3 307.37	2 506.95	5 814.3168	0	0	1	7 728.587	384 816 345	0
INSERT INTO F_ALGO_M_394574216				5 770.05	1 443.74	1 032.74	5 770.0504	0	0	0	4 674.354	45 528 843	54 026 805

STATEMENT TEXT FOR LAST SELECTED HASH: 531555986

```

select num(T157283_SPP2_HETID) as
num(T157283_SPP2_SAPFD) as c1,
T13231_GRP_HAZ as c3,
T13231_GRP_KOD as c4,
S14.c7 as c5,

```

EXECUTION PLAN FOR LAST SELECTED PLAN HASH: 531555986

Show plan objects for 531555986

```

SELECT STATEMENT ( Cost = 2854, Bytes = 0, Cardinality = 0, Search Columns = 0 )
  SORT (ORDER BY) ( Cost = 2854, Bytes = 240, Cardinality = 1, Search Columns = 0 )
    HASH JOIN ( Cost = 2854, Bytes = 240, Cardinality = 1, Search Columns = 0 )
      HASH JOIN ( Cost = 2854, Bytes = 240, Cardinality = 1, Search Columns = 0 )
        NESTED LOOPS

```

6.2.1.5 SQL Details Tab

SQL Details tab shows detailed information about the query such as:

- frequency,
- execution time,
- whether the request has changed execution plan,
- the number of returned records,
- the number of executions,
- role of query in load of databases,

This information provides the opportunity to decide whether it makes sense to optimize given query.

The screenshot displays the SQL Details tab interface. At the top, there's a navigation bar with various performance monitoring tabs. The SQL Details tab is selected, showing a query ID (1169289566) and a date range (2018/08/29) with a time range (00:00 to 23:59). Below this, the SQL statement text is shown. The main section is titled 'SQL STATISTICS (SQL ID: 159ptg12v3way)' and contains a table with columns: Date, Plan hash, Elapsed Time, Cpu Time, Rows processed, Fetches, Executions, Parse Calls, Disk Reads, Disk Reads, Buffers Get, Buffer Quality, and Elapsed Time per 1 Exec. The table shows data for the date 2018-08-29 with a plan hash of 824340315. Below the statistics is the 'Explain plan' section, which includes a tree view of the execution plan and a 'Show plan objects for 824340315' button.

[SQL Details] tab is divided into several areas:

Clipboard with a list of query IDs (expanded and collapsed by clicking the green button in the lower right corner of the screen) - the queries to the clipboard are added from screens

- Performance ->Database Load
- Performance ->SQL Analyze
- Performance ->Top SQL
- Performance ->SQL 3D
- Performance ->Top Day
- Performance ->Slow SQLs
- Sessions
- Locks

The screenshot shows the Query Hashes list interface. At the top right, there is a green clipboard icon. Below it is a yellow message box with a lightbulb icon that says 'Click on Hash Value to analyze Query Performance Details'. The main area is titled 'Query Hashes list' and contains a list of query IDs: 2916465158, 1044079983, 394574216, 3715327, and 1169289566. The last item, 1169289566, is highlighted in green and has a green checkmark next to it. At the bottom, there are buttons for 'Clear list', a document icon, a share icon, and a clipboard icon.

Important: List of queries is remembered under the database for specified user. That list can be saved to file or opened again.

Filters area and the way to display statistics for:

- specified Query Hash / Hash Value or SQL ID of the query
- selected date range
- a grouping of statistics by day, snapshot, etc.
- navigation buttons which allow to refresh the screen, to search another query or to show the statistics of queries in report

Enter hash value or sql: From: 2018/08/29 00:00 to: 2018/08/29 23:59 Group by plan hash Group by Day Online values Refresh Find SQL

Important: Selecting [checkbox Online Values] – some filters are hidden by default and clicking the [Refresh] button will present statistic of specified query according to information which are available in V\$SQL system view

- Area with query text– with a scroll ability – convenient for longer query content

```
STATEMENT TEXT
SELECT /*+ index(T4 DBPLUS_TAB4_NUM9) */ SNAP_ID , NUM1 , NUM2 , NUM3 , NUM4 , NUM5 , NUM6 , NUM7 , NUM8 , NUM10 , NUM23 , VAR5 , NUM26 FROM DBPLUS_TAB4 T4 WHERE T4.NUM9 = :b1
AND T4.NUM12 = :b2 AND T4.NUM11 = :b3 AND T4.NUM22 = :b4 AND T4.NUM10 = :b5 AND T4.SNAP_ID = (SELECT /*+ index(m4 DBPLUS_TAB4_NUM9) */ MAX (SNAP_ID) FROM DBPLUS_TAB4 M4 WHERE
M4.SNAP_ID > :b6 AND M4.NUM9 = :b1 AND M4.NUM12 = :b2 AND M4.NUM22 = :b4 AND M4.NUM10 = :b5 )
```

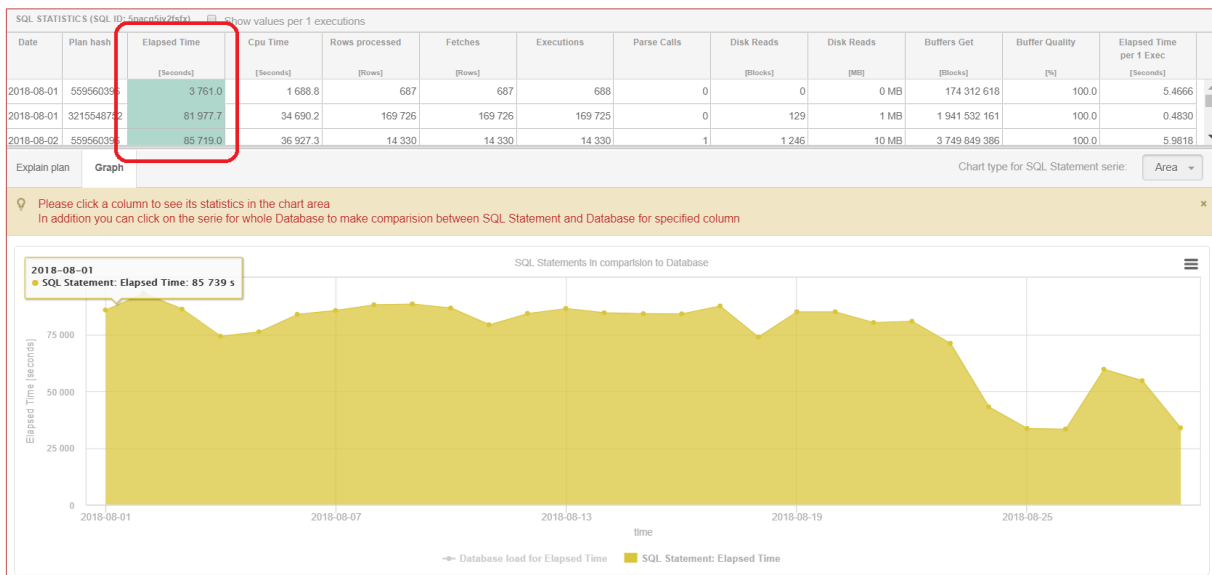
- Detailed execution statistics in form of the table

Date	Plan hash	Elapsed Time [Seconds]	Cpu Time [Seconds]	Rows processed [Rows]	Fetches [Rows]	Executions	Parse Calls	Disk Reads [Blocks]	Disk Reads [MB]	Buffers Get [Blocks]	Buffer Quality [N]	Elapsed Time per 1 Exec [Seconds]
2018-08-29	824340315	102.8	30.8	145 183	1 540 225	1 540 225	66	72 030	563 MB	5 235 466	98.6	0.0001

- Execution plan (with the [Explain Plan] tab selected)

```
Explain plan Graph Compare Plans 3995048130 
Show plan objects for 3995048130
SELECT STATEMENT ( Cost = 199949 , Bytes = 0 , Cardinality = 0 , Search Columns = 0 )
├── SORT (ORDER BY) ( Cost = 199949 , Bytes = 903500 , Cardinality = 3476 , Search Columns = 0 )
│   ├── HASH (GROUP BY) ( Cost = 199949 , Bytes = 903500 , Cardinality = 3476 , Search Columns = 0 )
│   │   ├── HASH JOIN ( Cost = 199947 , Bytes = 7464100 , Cardinality = 28786 , Search Columns = 0 )
│   │   │   ├── INDEX (FAST FULL SCAN) W_RODIAJ_SFR2_KSFR2_NAD_IDS_I ( Cost = 2 , Bytes = 3842 , Cardinality = 164 , Search Columns = 0 )
│   │   │   └── HASH JOIN ( Cost = 199945 , Bytes = 6922046 , Cardinality = 28786 , Search Columns = 0 )
```

- Graphical presentation (With the Graph Tab selected) of any indicators/column from the statistic table



Typing a query ID (hash value, SQL id) into the 'Enter Query Hash or SQL id field':

[Enter Query Hash] displays statistics for given query identifier according to set filters.



IMPORTANT: If you do not know and do not have any query ID and clipboard with a list of queries is empty, you can:

- go to one of the screens (Database Load, SQL Analyze, Top SQL, Top Day), where you can search suboptimal/long-lasting query
- click on the Find SQL to search for a specific query (search by its text)

Statistics show:

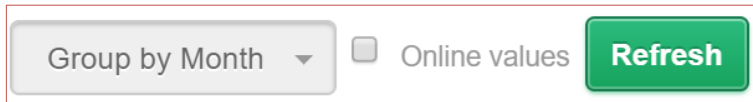
- Plan Hash- query plan identifier
- Elapsed Time (sec) - total time in seconds for the duration of the query for the selected grouping period.
- CPU time (sec) - total time in seconds for CPU utilization for the query for the selected grouping period.
- Rows processed [Rows] - Number of rows processed by query for the selected grouping period
- Fetches [Rows] – number of downloads from the query cursor
- Executions - The number of query executions for the selected grouping period
- Parse Calls – the number of query plan calculations for a given query over the last 15 minutes
- Disk Reads (MB) - Number of readings from the disk for a given query for the selected grouping period, broken down by the amount of MB and data blocks
- Buffet Gets - The number of utilized buffers for a given query for the selected grouping period
- Buffer quality [%] – percentage of data that has been downloaded from the memory for a given query
- Elapsed time per 1 exec - duration of a single query execution for the selected period grouping

Filter **[Group by period]** - shows statistics for a given query grouped according to the choice:

- **No group by period** - no grouping i.e. selection date ranges from 1 to 20 days of the month will show summary statistics for the selected period
- **Month** – shows statistic for a given query, broken down by months
- **Day** - shows statistics for a given query, broken down into periods of one day

- **Hour** - shows statistics for a given query, broken down into periods of one hour
- **Snap** - shows statistics for a given query, broken down by snapshots - periods of 15 minutes

Online values option allows Users to display current information about queries stored in the database buffer.



Statistics on queries in other tabs appear in 15 minutes and after activating the **[Show Online Values]** option, you can see them right away if only the query starts at that moment.

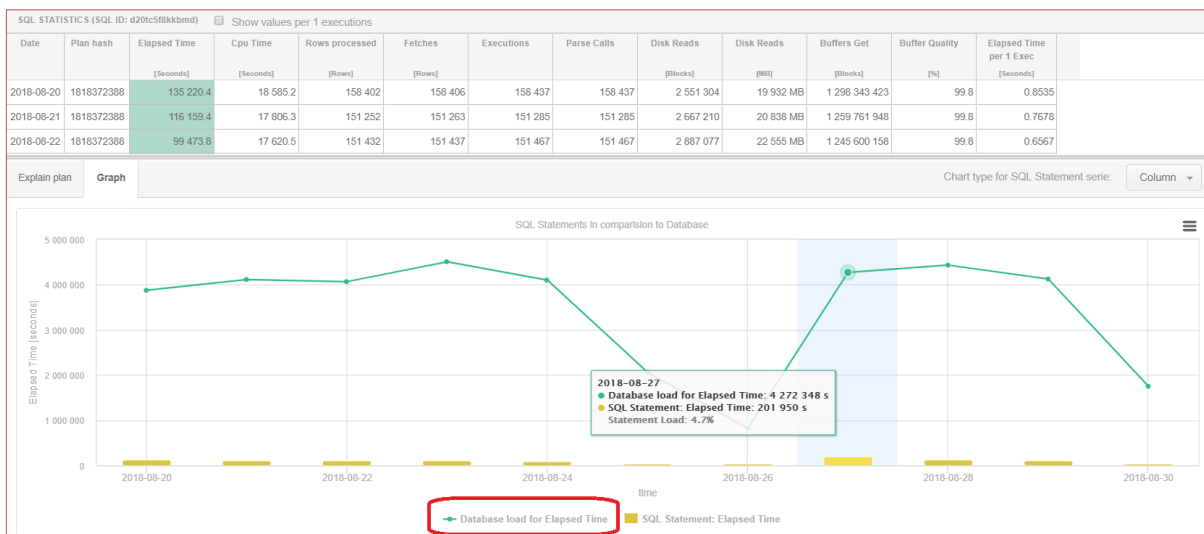
To check whether the query is performed or not at the moment, after supplying the Query Hash identifier, activate the Online Values checkbox and click the **[Refresh]** button to observe the value in the columns Execution, Elapsed Time, CPU Time. If the values change, it means that the query is still executing. If the values are fixed, the query has stopped running.

Additionally, with **Online values** option is checked apart from standard statistics, additional information is presented, such as:

- **Module** – name of the module / program that runs the given query, downloaded according to information available in the V\$SQL system view.
- **Outline category** – column supplemented with the Outline category name in case the given query is assigned an Outline.
- **Versions count** – how many versions of a given query (Query Hash) have been found for a given plan (Plan Hash) at a given moment in a shared pool.

Graph tab

By clicking on the **[Graph tab]** we can see the load generated by the given query (line / area yellow) against the background of the total base load. By default, the graph presents the data for the selected statistics, to compare the impact of the query on the entire database, additionally select the Database load for ... (column name) at the bottom of the chart.



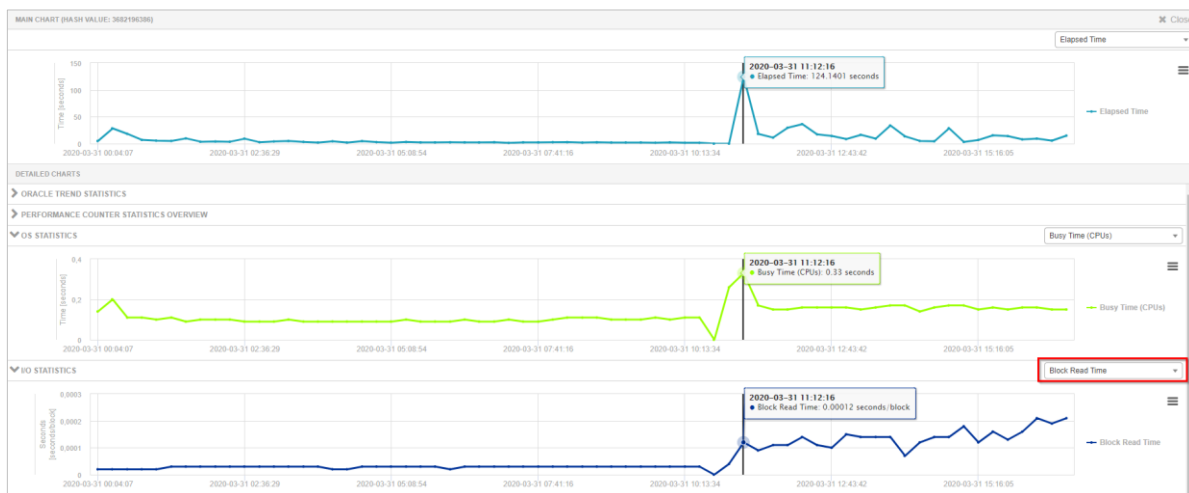
On the Graph tab we have provided the option of in-depth query analysis. The function is available after select **Show detailed charts**. This option allows User to compare any statistics of a given query in combination with performance statistics of the entire database at the same time.

After selecting this option, the screen shows a graph showing statistics for a given query and graphs showing general performance data of the entire database, such as:

- Oracle Trend Statistics (Load trends)
- Performance Counters
- OS Statistics
- I/O Statistics
- Memory Statistics

The statistics can be compared for a specific point in time and verified for all statistics on one screen. By compiling many statistics in one screen, the User can easily find the source of the problem affecting query performance.

In the example shown in the figure below, the change in query performance was caused by a decrease in the read performance of the data block on the disk array, which is visible in the I/O Statistics chart where the Block Read Time series is selected.

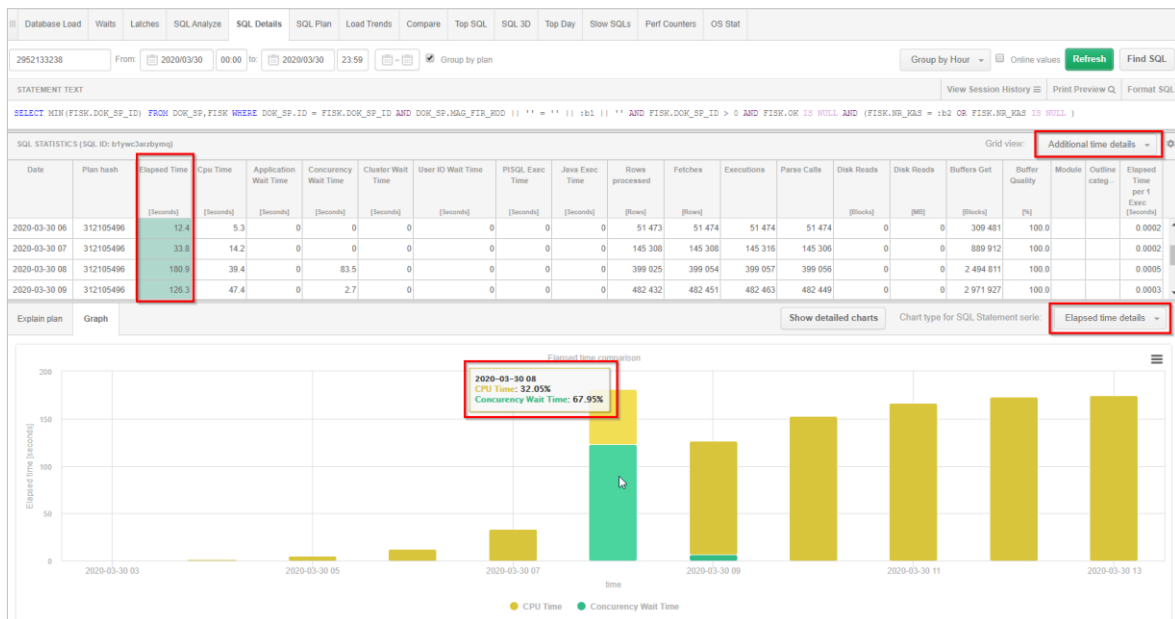


On the screen, the user can add any series available for a given chart by clicking the dropdown button and selecting the given statistic from the list.

Charts can be freely zoomed as well as saved to a file, according to the logic available so far in the application.

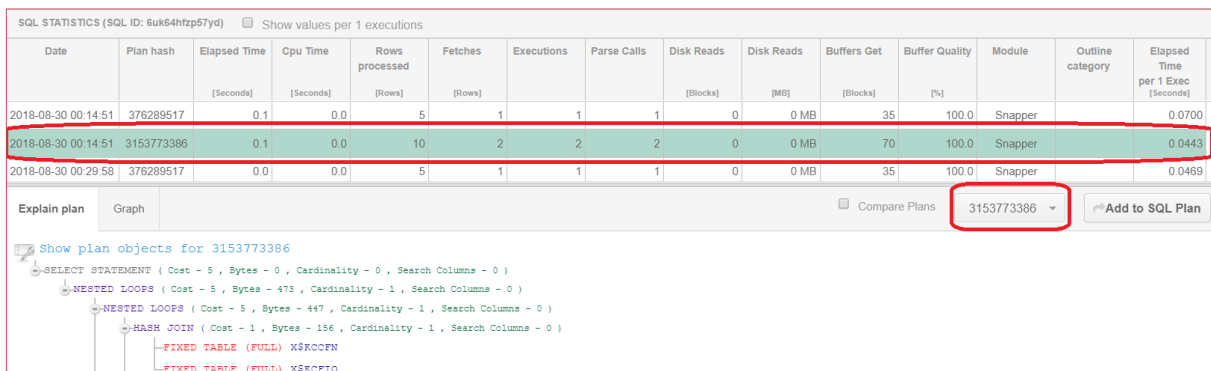
For the SQL Details page, statistics are available when User selects the: **Additional time details** view from the dropdown Grid view. In addition to the basic statistics, this view presents additional statistics that contain information on what expectations a given query spent time on. Data is downloaded from system views and is not further processed.

In addition to the information available in tabular form, this information is available in the form of a graph. In this case, a new chart type should be selected in the Graph tab: Elapsed time statistics. The chart presents the duration of the Elapsed time query broken down into individual components (CPU Time and newly added statistics). After indicate a given bar on the chart, the data in the tooltip are presented as a percentage.

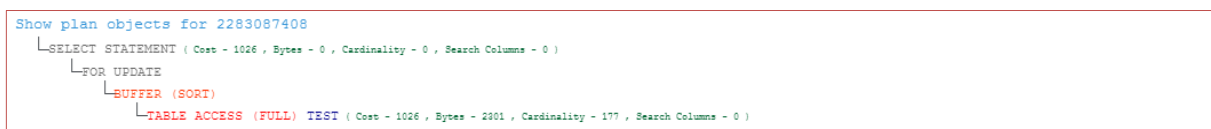


Explain Plan tab

Shows the execution plan for the query currently selected. The selection of the execution plan is made by clicking on the statistics table or through the control with the list of plans.



Example execution plan:



With the [Online Values] option enabled on the execution plan, you can additionally see which parameter values the query runs:

Show plan objects for 149800732

```

SELECT STATEMENT ( Cost - 8 , Bytes - 0 , Cardinality - 0 , Search Columns - 0 )
  SORT (GROUP BY) ( Cost - 8 , Bytes - 67 , Cardinality - 1 , Search Columns - 0 )
    FILTER
      NESTED LOOPS
        NESTED LOOPS ( Cost - 7 , Bytes - 67 , Cardinality - 1 , Search Columns - 0 )
          INDEX (RANGE SCAN) IDX_DBPLUS_SNAPS_SERVER_ID ( Cost - 3 , Bytes - 16 , Cardinality - 1 , Search Columns - 2 )
          INDEX (RANGE SCAN) DBPLUS_TAB4_NUM9 ( Cost - 2 , Bytes - 0 , Cardinality - 1 , Search Columns - 2 )
          TABLE ACCESS (BY INDEX ROWID) DBPLUS_TAB4 ( Cost - 4 , Bytes - 102 , Cardinality - 2 , Search Columns - 0 )
    Parameters list
      :SERVER_ID(NUMBER): 41
      :DAT1 (DATE): '2018/08/21 00:00:00'
      :DAT2 (DATE): '2018/08/30 00:00:00'
      :A (NUMBER): 3209863117
  
```

It is also possible to preview the contents of the query with the completed parameters, for this purpose click on the key icon and select **Show statement script with filled parameters**.

If there is more than one then you can click on the Compare plans checkbox, which will display two execution plans. This makes it easier to compare and find differences between them:

Compare Plans 1144377621 Add to SQL Plan 1089751689

Show plan objects for 1144377621

```

SELECT STATEMENT ( Cost - 38 , Bytes - 0 , Cardinality - 0 , Search Columns - 0 )
  SORT (GROUP BY) ( Cost - 38 , Bytes - 3840 , Cardinality - 60 , Search Columns - 0 )
    FILTER
      HASH JOIN ( Cost - 37 , Bytes - 3840 , Cardinality - 60 , Search Columns - 0 )
        TABLE ACCESS (BY INDEX ROWID) DBPLUS_TAB4 ( Cost - 33 , Bytes - 102 , Cardinality - 2 , Search Columns - 0 )
          INDEX (SKIP SCAN) DBPLUS_TAB4_SNAP_ID ( Cost - 18 , Bytes - 0 , Cardinality - 1 , Search Columns - 2 )
          INDEX (RANGE SCAN) IDX_DBPLUS_SNAPS_SERVER_ID ( Cost - 3 , Bytes - 16 , Cardinality - 1 , Search Columns - 2 )
        Parameters list
          :SERVER_ID(NUMBER): 1
          :DAT1 (DATE): 11/16/2016 00:00:00
          :DAT2 (DATE): 11/25/2016 00:00:00
          :A (NUMBER): 1210997395
      
```

Show plan objects for 1089751689

```

SELECT STATEMENT ( Cost - 5 , Bytes - 0 , Cardinality - 0 , Search Columns - 0 )
  SORT (GROUP BY) ( Cost - 5 , Bytes - 64 , Cardinality - 1 , Search Columns - 0 )
    FILTER
      NESTED LOOPS
        NESTED LOOPS ( Cost - 4 , Bytes - 64 , Cardinality - 1 , Search Columns - 0 )
          INDEX (RANGE SCAN) IDX_DBPLUS_SNAPS_SERVER_ID ( Cost - 3 , Bytes - 16 , Cardinality - 1 , Search Columns - 2 )
          INDEX (RANGE SCAN) DBPLUS_TAB4_SNAP_ID ( Cost - 1 , Bytes - 0 , Cardinality - 1 , Search Columns - 2 )
          TABLE ACCESS (BY INDEX ROWID) DBPLUS_TAB4 ( Cost - 2 , Bytes - 102 , Cardinality - 2 , Search Columns - 0 )
        Parameters list
          :SERVER_ID(NUMBER): 1
          :DAT1 (DATE): 11/12/2016 00:00:00
          :DAT2 (DATE): 11/21/2016 00:00:00
          :A (NUMBER): 84828483
      
```

The execution plan has an active Show Plan Objects link, which is use for detailed analysis of the execution plan. After clicking, a form presenting the objects used in the query plan appears:

SQL TEXT

```
select /*+ USE_NL(t4) LEADING(s t4) index(t4 DBPLUS_TAB4_NUM9) */ s.logdate,
t4.num5, sum(t4.num13), sum(t4.num14), sum(t4.num15), sum(t4.num16), sum(t4.num17), sum(t4.num18), sum(t4.n
um19), sum(t4.num20), t4.num10
from dbplus_tab4 t4, dbplus_snaps s
where s.server_id = :sserver_id
and s.logdate >= :dat1 and s.logdate <= :dat2
and s.snap_id = t4.snap_id
and t4.num9 = :a
and t4.num24 > 0 group by t4.num9, t4.num10, s.logdate order by s.logdate
```

EXPLAIN PLAN

```
SELECT STATEMENT ( Cost = 8, Bytes = 0, Cardinality = 0, Search Columns = 0 )
  SORT (GROUP BY) ( Cost = 8, Bytes = 67, Cardinality = 1, Search Columns = 0 )
    NESTED LOOPS
      NESTED LOOPS ( Cost = 7, Bytes = 67, Cardinality = 1, Search Columns = 0 )
        INDEX (RANGE SCAN) IDX_DBPLUS_SNAPS_SERVER_ID ( Cost = 3, Bytes = 16, Cardinal
        INDEX (RANGE SCAN) DBPLUS_TAB4_NUM9 ( Cost = 2, Bytes = 0, Cardinality = 1, Sea
```

OBJECTS USED IN EXPLAIN PLAN

Type	Owner	Object Name	Alternative Object
INDEX	DBPLUS	IDX_DBPLUS_SNAPS_SERVER	
INDEX	DBPLUS	DBPLUS_TAB4_NUM9	
TABLE	DBPLUS	DBPLUS_TAB4	
TABLE	DBPLUS	DBPLUS_SNAPS	
INDEX	DBPLUS_PIP	IDX_DBPLUS_SNAPS_SERVER	

INDEXES FOR SELECTED OBJECT DBPLUS.IDX_DBPLUS_SNAPS_SERVER_ID

Owner	Name
DBPLUS	DBPLUS_SNAPS_LOGADATE
DBPLUS	IDX_DBPLUS_SNAPS_SERVER_ID
DBPLUS	SNAP_ID_LOGDATE
DBPLUS	IDX_TMOBILE_DBPLUS_SNAP

Object columns Details for INDEX DBPLUS.IDX_DBPLUS_SNAPS_SERVER_ID Load object properties (slower)

Column	Position
SERVER_ID	1
LOGDATE	2
SNAP_ID	3

On the SQL Details screen, we've added special support for queries for which we have many execution plans. On the Graph tab, a new type of Separate plans chart has been added. This function allows you to view query statistics separately for each of the execution plans. By choosing this option, the data on the graph is presented separately for each of the plans.

Below is the Elapsed time per 1 exec chart - sorted with the No Group by period option. On this chart, the user can easily assess which of the implementation plans is the fastest for a given query in the analyzed period of time. Additionally, the columns are sorted based on the number of executions of a given query with a given plan (most often performed from the left).

In the Separate Plans mode, the rows in the grid are colored in accordance with the color assigned in the chart for the given query plan.

Instance Load: 0x038B015C7EDC8153

From: 2019/12/16 00:00 to: 2019/12/24 23:59

No group by period
Online values
Refresh
Find SQL

STATEMENT TEXT

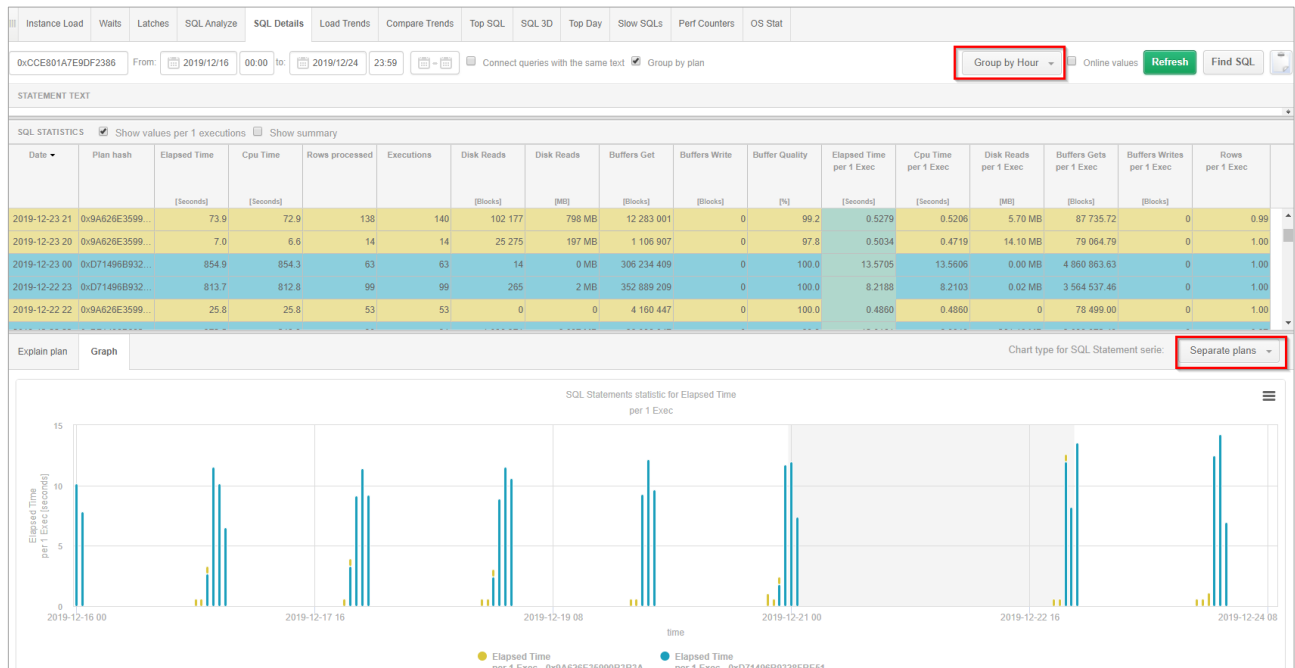
```
SQL STATISTICS Show values per 1 executions Show summary
```

Plan hash	Elapsed Time	Cpu Time	Rows processed	Executions	Disk Reads	Disk Reads	Buffers Get	Buffers Write	Buffer Quality	Elapsed Time per 1 Exec	Cpu Time per 1 Exec	Disk Reads per 1 Exec	Buffers Gets per 1 Exec	Buffers Writes per 1 Exec	Rows per 1 Exec
	[seconds]	[seconds]			[blocks]	[MB]	[blocks]	[blocks]	[%]	[seconds]	[seconds]	[MB]	[blocks]	[blocks]	
0x29E74EBC0	12.7	7.6	0	2	329 154	2 572 MB	6 003 982	0	94	6.3296	3.7815	1 285.76 MB	3 001 991.00	0	0
0x3C53842361	9.4	9.4	0	18	0	0	10 872 424	0	100	0.5218	0.5208	0	604 023.56	0	0
0x5A1ABBF4A	61.6	63.4	0	62	1 681 520	13 137 MB	36 118 627	0	95	0.9932	1.0219	211.89 MB	582 558.50	0	0
0x68DF2F7B7	46 874.4	21 093.1	898	56 294	62 778 373	490 456 MB	22 853 511 085	300	99	0.8327	0.3747	8.71 MB	405 967.09	0.01	0.02
0x8D96CF8F0	93.7	75.5	0	85	212 634	1 661 MB	101 002 141	0	99	1.1018	0.8880	19.54 MB	1 188 260.48	0	0
0xE22A43AC6	276 479.9	261 793.3	3 357	952 007	201 604 903	1 576 601 MB	285 774 142 342	0	99	0.2904	0.2750	1.66 MB	300 100.72	0	0.08

Explain plan Chart type for SQL Statement series: Separate plans

SQL Statements statistic for Elapsed Time per 1 Exec

In the case of sorting by hour or snap (15 minutes), the user will easily obtain information when and at what times the query uses a slower plan.



The **[Show Plan Objects]** consists of repeated information about the query text and the execution plan. Below the text and the execution plan, there are areas:

- **Objects Used in Explain Plan** - a list of all objects used by the query in given execution plan
- **Indexes for selected object** - list of indexes for selected table - row selected in the "Objects Used in Explain Plan"
- The area comprised of three tabs:
 - a. **Object Columns** - a list of individual columns of the selected object, along with information such as: column name, data type, id columns, density (the lower density - the higher selectivity of the column)
 - b. **Info** - basic information about selected object
 - c. **Properties** - additional properties of selected object

In the new version of the application for Oracle databases we have added a mechanism for formatting and parsing queries that run on the monitored database. The mechanism will be developed by the next releases. In the first phase, the mechanism is available only on the Show Plan Objects screen in the SQL Details tab.

In the current version, the Show Plan Objects view shows the content of the query in unformatted form.

SQL TEXT

```
SELECT P_DOK_MA.TOM_KOD, STM(DECODE(P_DOK_MA.ZNA,1,P_DOK_MA.ILE,0)) FRM, STM(DECODE(P_DOK_MA.ZNA,1,0,P_DOK_MA.ILE)) FRM,
STM(DECODE(P_DOK_MA.ZNA,1,P_DOK_MA.WAR,0)) WFRF, STM(DECODE(P_DOK_MA.ZNA,1,0,P_DOK_MA.WAR)) WFRF FROM DOK_MA, P_DOK_MA WHERE
DOK_MA.AND = 'B' AND DOK_MA.TYP_D_ID=0 IN (347,545,555,326,544,546) AND DOK_MA.DAT_M BETWEEN :B3 AND :B2 AND
DOK_MA.WAR_FIR_KOD(1) = :B1 AND DOK_MA.ID = P_DOK_MA.DOK_MA_ID GROUP BY P_DOK_MA.TOM_KOD
```

EXPLAIN PLAN

```
INSERT STATEMENT
LOAD TABLE CONVENTIONAL
SORT (GROUP BY)
NESTED LOOPS
TABLE ACCESS (BY INDEX ROWID) DOK_MA
INDEX (RANGE SCAN) DOK_MA_DAT_M
INDEX (RANGE SCAN) P_DOK_MA_DOK_MA_FK_I
TABLE ACCESS (BY INDEX ROWID) P_DOK_MA
```

Type	Owner	Object Name	Owner	Name
TABLE	INTER	DOK_MA		
INDEX	INTER	DOK_MA_DAT_W		DOK_MA_WSK_POT_2
INDEX	INTER	P_DOK_MA_DOK_MA_FK_I		DOK_MA_WSK_POT_3
TABLE	INTER	P_DOK_MA		DOK_MA_DATKPK_FK_I
			INTER	DOK_MA_DATWPK_FK_I
			INTER	DOK_MA_FIR_DOC_I
			INTER	DOK_MA_FIR_DOC_I

The new function is available in two modes:

- manual,
- automatic.

In the manual mode, after enter the Show Plan Objects and after press the [Parse SQL Query] button, the query is formatted and parsed. Formatting the query changes the presentation in the SQL TEXT window to a form that facilitates query analysis.

The parsing function in the current version gives the opportunity to highlight the columns that belong to the given object participating in the query. In the following case, the DOK_MA table was selected on the query, and all columns associated with the given table.

Depends on the object which user indicates; objects are marked in different colors:

- Table (green),
- Indexes (yellow).

The highlighting is performed in both the SQL TEXT and EXPLAIN_PLAN fields.

SQL TEXT

```
SELECT p_dok_ma.tom_kod, tom_kod,
STM(DECODE(p_dok_ma.zna,1,p_dok_ma.ile,0)) frm,
STM(DECODE(p_dok_ma.zna,1,0,p_dok_ma.ile)) frm,
STM(DECODE(p_dok_ma.zna,1,p_dok_ma.war,0)) wfrf,
STM(DECODE(p_dok_ma.zna,1,0,p_dok_ma.war)) wfrf
FROM
p_dok_ma
WHERE
dok_ma.and = 'B'
AND dok_ma.typ_d_id = 0 IN (347,545,555,326,544,546)
AND dok_ma.dat_m BETWEEN :B3 AND :B2
AND dok_ma.war_fir_kod(1) = :B1
AND dok_ma.id = p_dok_ma.dok_ma_id
GROUP BY p_dok_ma.tom_kod
```

EXPLAIN PLAN

```
INSERT STATEMENT
LOAD TABLE CONVENTIONAL
SORT (GROUP BY)
NESTED LOOPS
TABLE ACCESS (BY INDEX ROWID) DOK_MA
INDEX (RANGE SCAN) DOK_MA_DAT_M
INDEX (RANGE SCAN) P_DOK_MA_DOK_MA_FK_I
TABLE ACCESS (BY INDEX ROWID) P_DOK_MA
```

Type	Owner	Object Name	Owner	Name
TABLE	INTER	DOK_MA		
INDEX	INTER	DOK_MA_DAT_W		DOK_MA_WSK_POT_2
INDEX	INTER	P_DOK_MA_DOK_MA_FK_I		DOK_MA_WSK_POT_3
TABLE	INTER	P_DOK_MA		DOK_MA_DATKPK_FK_I
			INTER	DOK_MA_DATWPK_FK_I
			INTER	DOK_MA_FIR_DOC_I
			INTER	DOK_MA_PACZ_ITRD_FK_I

As part of the mechanism, the user can choose the range of highlighted objects on the query. To change the configuration, click the "cog" button on the Show plan Objects page.

Parse SQL Query

EXPLAIN PLAN

```
INSERT STATEMENT
LOAD TABLE CONVENTIONAL
SORT (GROUP BY)
NESTED LOOPS
TABLE ACCESS (BY INDEX ROWID) DOK_MA
INDEX (RANGE SCAN) DOK_MA_DAT_M
INDEX (RANGE SCAN) P_DOK_MA_DOK_MA_FK_I
TABLE ACCESS (BY INDEX ROWID) P_DOK_MA
```

As a result of the click, a window will open where user can choose:

- SQL Parser – On demand/Automatic (On demand/Automatic) – parsing mode.

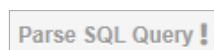
Depending on the mode, when user enter the Show Plan Objects screen, the query will be automatically formatted and parsed (Automatic mode).

- Highlight columns - depending on the selection, the columns in the query will be highlighted
- Highlight color – color selection for table / index highlight.

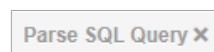
Each time after parsing the query, user will receive information about the status of the performed operation. If everything went well, the button on the right will be presented in this form:



In case the query was formatted correctly, however, there was a problem with reading all objects from the query:



If after the parsing the "X" sign is displayed, it means that the query could not be properly formatted as well as parsing could not be done. Support for such queries will be provided in next releases.



Within the Info tab, an option is made available to display the object definition after selecting the Load object properties filter and selecting the Info tab. To do this, enter the searched object and press the **[Show]** button.

The screenshot shows the SQL Developer interface. On the left, the SQL text is displayed, including a MERGE statement and a WHERE clause. The EXPLAIN PLAN on the right shows a MERGE STATEMENT with a NESTED LOOPS (OUTER) join. Below the explain plan, the 'OBJECTS USED IN EXPLAIN PLAN' and 'INDEXES FOR SELECTED OBJECT' are listed. The 'Info' tab is selected, showing the object's properties and the CREATE OR REPLACE FUNCTION 'DBPLUS_RC' definition.

Next [Properties] tab has basic information about chosen object, e.g. size [MB]

Object columns	Info	Properties	Details for TABLE DBPLUS.DBPLUS_TAB4
			Property Value
			SIZE (MB) 12970
			TABLESPACE_NAME SA_TEMP
			LOGGING YES
			PARTITIONED NO
			TEMPORARY N
			SAMPLE_SIZE 4021884
			LAST_ANALYZED 9/25/2018 8:32:01 AM

Important: When analyzing the execution plan, a particular attention is paid to:

- Limiting the choice of data, or of the data with the where clause and tables joins
- Whether the request is with parameters or literals
- Actions that SQL Engine chose to data download
- Whether appropriate indexes are in the table
- Way of reading the data - Nested Loops vs. Hash Join

In this example, we evidently see that after change of the execution plan, the query sped up more than 40 times. **Elapsed time of a single execution changed from 0.0174 to 0.0004 seconds.**

SQL STATISTICS														Export results			
Show values per 1 executions																	
Date	Plan hash	Time statistics [seconds]				Other statistics								Statistics per 1 execution			
		Elapsed Time	CPU Time	Rows	Fetches	Execs	Parse Calls	Disk Reads	Disk Reads (MB)	Buffers Get	Buffer Quality	Elapsed Time	CPU Time	Rows	Fetches	Disk Reads	Buffers Gets
2015-09-21 11:20:33	359490790	367.8	58.9	933324	0	933352	28	1529	11.9	3786793	99.96	0.0004	0.0001	1	0	0	4.04
2015-09-21 11:05:30	359490790	370.3	59.3	933324	0	933352	28	1518	11.9	3785846	99.96	0.0004	0.0001	1	0	0	4.03
2015-09-21 10:50:27	359490790	370	57.8	999990	0	1000020	30	1678	13.1	4035190	99.96	0.0004	0.0001	1	0	0	4.04
2015-09-21 10:20:23	2258898854	350.1	174.9	7141	0	7142	0	0	0	16011215	100	0.049	0.0245	1	0	0	2241.84
2015-09-21 10:05:23	2258898854	348.8	216.1	13487	0	13488	1	0	0	25025206	100	0.0257	0.016	1	0	0	1855.37
2015-09-21 09:50:22	2258898854	359.2	354.2	28787	0	28788	1	0	0	48578247	100	0.0125	0.0123	1	0	0	1888.82
2015-09-21 09:35:19	2258898854	359.2	278.8	25541	0	25541	0	0	0	34719162	100	0.0141	0.0109	1	0	0	1359.35
2015-09-21 09:20:14	2258898854	14.5	8.1	597	0	598	1	0	0	772031	100	0.0256	0.0143	1	0	0	1359.21
2015-09-21 09:05:16	2258898854	282	149.2	16772	0	16773	1	515	4	16408531	100	0.0168	0.0089	1	0	0.03	978.27
2015-09-21 08:42:07	2258898854	81.9	80.6	9384	0	9385	1	0	0	8810134	100	0.0096	0.0085	1	0	0	917.44

[Find SQL] button

In a situation where no Query ID is known, and a specific query needs to be found e.g.:

- Queries that refer to a specified table
- Queries that changed execution plan
- New Queries

- Queries that use specific database objects e.g. an index
- Queries using objects stabilizing execution plans

Clicking on **[Find SQL]** button – displays the search query window.

Searching for queries containing specific text - Statement by text

Queries are searched in the DBPLUS repository database or in Online mode directly on the monitored database. Queries found can be "moved" to the clipboard of the SQL Details screen by clicking the [Plus] button next to the query identifier. For each query information is presented on:

- Query ID
- Last execution date
- Elapsed Time
- CPU Time
- Number of execution
- Number of reading blocks:
 - From disk devices
 - From memory
- Number of contained records
- Text of the query

Entering several expressions in the search field, the result will be returned in two separate grids:

- FIND RESULT FOR **EXACT** QUERY TEXT MATCHING WITH
- FIND RESULT FOR **SIMILAR** QUERY TEXT MATCHING WITH

For example, the result presented for searching for queries after entering "select max". In the upper table queries that exactly agree with the searched content "SELECT MAX (SNAP_ID) ..." are returned.

In the bottom table, queries for "select% max" were found
 "SELECT NVL(MAX(P.LP), 0) + 1 FROM P_R_SAM P...".

Statement by text

Plan Flip-Flop Statements

New statements

Statements using objects

Date from:
Date to:
Max. returned statements:

FIND RESULTS FOR EXACT QUERY TEXT MATCHING WITH SELECT MAX

Hash Value	Last execution date	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Disk reads [MB]	Buffer gets [Blocks]	Rows processed	Query text
3715327	2018/09/18	5.40	2.20	16	7 MB	1 531 452	16	SELECT MAX (SNAP_ID) FROM DBPLUS_SNAPS WHERE LOGDA
8345023	2018/09/18	175.40	70.30	3 174	0 MB	41 822 477	351 808	select v.sek_kod as SECTOR, case when v.jest_foto = "SYS_B_00" t
24937523	2018/09/18	7.40	2.80	238 855	0 MB	1 057 889	238 855	select max(mod(nvl(option\$,0),2)), count(*) from objauth\$ where obj#
58585869	2018/09/18	4 298.40	1 500.10	16	130 537 MB	234 512 229	15	DECLARE job BINARY_INTEGER := :job; next_date DATE := :mydat

FIND RESULTS FOR SIMILAR QUERY TEXT MATCHING WITH SELECT%MAX

Hash Value	Last execution date	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Disk reads [MB]	Buffer gets [Blocks]	Rows processed	Query text
103703884	2018/09/18	13.70	5.80	120 587	0	1 163 220	120 587	SELECT NVL(MAX(P.LP), 0) + 1 FROM P_R_SAM P WHERE 1 = 1
103748849	2018/09/18	60.80	25.00	1 016	17 MB	3 072 537	13 486	SELECT /*+ */ TRS_KOD,TRS_NAZ,STA_Z,ILE_DOK_ZA,WAR_DC
112012398	2018/09/18	8.40	2.90	23 160	0	62 247	23 160	SELECT ROWID,KH_KOD_2,KH_KOD,FIR_KOD_REA,DAT_W,DAT
119227748	2018/09/18	166.60	62.80	179	17 MB	6 873 284	165	SELECT WAL_KOD,WAR_W,WAR_WARN_W,WARN,WARZ_W,WAI

Searching the queries that change the plan

With the Plan Flip-Flop Statements tab selected, a search for queries that have changed the plan of execution in a given period of time. Using the values of i.a. Elapsed Time, CPU Time, an ability to search for those queries whose share in the load is significant.

For queries changing the execution plan, additional information is grouped according to the following areas:

- Statistics with a summary for all performance plans on which the query worked,
- Slowest plan statistics summary,
- Fastest plan statistics summary,
- Comparison between Slowest and Fastest
- Possible time reductions for queries statistic.

Below is an example of the search results for those questions that will change the execution plan within two weeks:

View of the areas [Total statistics, Slowest plan statistics]

Statement by text		Date from:	2018/04/10	00:00	Date to:	2018/04/24	23:59				
Plan Flip-Flop Stateme...											Search
New statements											
CLICK ON [ADD TO SQL DETAILS] BUTTON (ICON WITH +) TO ADD QUERY IDENTIFIER TO QUERY HASHES TOOLBAR LIST											
		Total statistics				Slowest plan statistics					
Query Hash	Query text	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Number of plans	Plan Hash	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Elapsed Time Per 1 exec [Seconds]	
0x64C102F23329DC98	select top(@v)	486 925.70	394 926.91	426 821	2	0x31F605092B25	442 606.41	367 606.00	156 096	2.8355	
0xA86C8E58E207D6E8	select max(Err	70.20	24.68	43	2	0x397376A5E330	52.39	19.52	21	2.4946	
0x25B65C61193863C4	select * from Pr	11 726.99	10 221.35	1 176 774	3	0xD445611DDBA	420.77	138.82	1 773	0.2373	
0xE95D16F7F24BD1F3	SELECT DB_ID	68.70	60.64	6 695	2	0x2370E781E95E	25.13	22.26	1 339	0.0188	
0x248FF45573B477FD	select convert(i	98.46	88.74	1 343	2	0x89C31130AB10	26.72	24.42	343	0.0779	
0x89EB3EE49C2797CF	select ? as rec	16.09	15.89	20 742	2	0x43B435618BC8	7.77	7.68	6 612	0.0012	

View of the areas [Fastest plan statistics, Slowest vs. Fastest, Estimation statistics]

		Fastest plan statistics				Slowest vs Fastest		Estimation statistics		
ns	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]
5 096	2.8355	0xF02EB8B03876	44 319.29	27 320.91	270 725	0.1637	17	2.6718	417 052.5628	351 853.1681
21	2.4946	0xFE2C0C637B8	17.82	5.16	22	0.8098	3	1.6848	35.3808	14.5963
1 773	0.2373	0x90B998ECB7C	8 388.18	7 310.25	1 169 949	0.0072	33	0.2301	3 289.8811	2 868.4574
1 339	0.0188	0x43E66D931657	43.57	38.38	5 356	0.0081	2	0.0106	14.2387	12.6670
343	0.0779	0x2B459523C16C	71.73	64.33	1 000	0.0717	1	0.0062	2.1186	2.3535
5 612	0.0012	0xC99C4CF8765	8.32	8.21	14 130	0.0006	2	0.0006	3.8777	3.8421

An important area of the **Flip-Flop Statements** plan screen is the statistics estimation. The columns **Elapsed Time to reduce** and **CPU Time to reduce**, is a calculation about the possible reduction of time for the case when the query would work to be disabled on the fastest execution plan.

Helpful tip:

Sorting one of these columns will allow to find those questions whose optimization will bring the greatest improvement in performance.

Searching new queries – New Statements

It is also possible to search for new queries that started to run in a given time period, for which the total execution time is greater than the specified value.

In this case, queries that were performed on 30.08.2018 will be searched and not performed on 29.08.2018 for which the total duration for all queries was greater than 100 seconds.

Statement by text

Plan Flip-Flop Statements

New statements

Statements using objects

Statement executed in period

Date from: Date to: Min. elapsed time (sec):

And statement not executed in the period range

Date from: Date to:

[Search](#)

CLICK ON [ADD TO SQL DETAIL S] BUTTON (ICON WITH +) TO ADD QUERY IDENTIFIER TO HASH VALUES TOOLBAR LIST

Hash Value	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Disk reads [MB]	Buffer gets [Blocks]	Rows processed	Query text
3733539908	13 318.53	3 066.84	1	119 672 MB	1 348 035 454	47 086	INSERT INTO TROSZAK_INTER.ZES_J79858 (W50,W1,W2,W3, W4, W5, W6, ...
559175374	8 969.64	3 181.98	1	32 728 MB	1 434 189 250	73 804	INSERT INTO TROSZAK_INTER.ZES_J79859 (W50,W1,W2,W3, W4, W5, W6, ...
4186994132	8 544.33	3 597.58	1	2 084 686 MB	265 382 584	0	select sek, sum(lenpz) ilosc_pozycji_na_pz, sum(ilepzwroz) ilosc_pozycji_nie_p...
256669286	5 144.71	937.73	1	69 004 MB	189 288 654	182 684 758	create table synchron
3737631907	3 127.87	1 278.85	1	601 MB	121 684 939	0	SELECT DS.MAG_KOD, DS.NR_SAD, DS.MAG_FIR_KOD, DS.ID, DS.DOK_SF...
3607165638	2 404.29	1 030.99	0	163 MB	47 219 149	520	select tab, kod, kod2, naz, naz_lang, kon, sta, upr, id from (select ...SYS_B_00'...
3560144616	2 269.23	84.55	1	15 706 MB	2 375 460	33	SELECT ROWID,MAG_FIR_KOD_STR,MAG_FIR_KOD_STR,BIE,DOK_ZA_ID,
2823342106	2 140.42	894.07	1	37 935 MB	4 855 694	0	/* SQL Analyze(1) */ select /*+ full(t) no_parallel(t) no_parallel_index(t) dbms_sta...
4145457797	1 951.60	94.89	3	6 786 MB	911 044	0	SELECT /*+ */ ROWID,ID,KOR,MAG_KOD,ROD_D_KOD,NR,DAT,W,KH,KOD,I...

Statements using objects

It is also possible to search for a query after entering the name of the object. In the case below, queries using DBPLUS_SNAPS objects were searched in a given period of time.

Statement by text

Plan Flip-Flop Statements

New statements

Statements using objects

Date from: Date to: Max. returned statements:

[Search](#)

FIND RESULTS

Hash Value	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Disk reads [MB]	Buffer gets [Blocks]	Rows processed	Query text
3710599702	51.52	21.57	132	11 MB	3 710 599 702	132	SELECT MAX (SNAP_ID) FROM DBPLUS_SNAPS WHERE SNAP_ID < :b1 AND NUM...
965693090	14.53	5.72	1 650	7 MB	965 693 090	1 650	SELECT MIN (SNAP_ID) FROM DBPLUS_SNAPS WHERE LOGDATE >= :b1
276493203	6.72	2.51	66	26 MB	276 493 203	66	SELECT NVL(MAX (LOGDATE) , '2000-01-01 00:00:00') , MAX (SNAP_ID) FROM DBP...
3715327	5.41	2.20	16	7 MB	3 715 327	16	SELECT MAX (SNAP_ID) FROM DBPLUS_SNAPS WHERE LOGDATE <= SYSDATE -

Statements using plan objects

It is possible to search queries is associated with Outlines / Baselines or Profiles. The search engine works in two ways:

- searching for queries that used any object in a given period (empty search field),
- searching for queries with an indication of the object's name (field supplemented with the name of the object, e.g. outline)

Statement by text

Plan Flip-Flop Statements

New statements

Statements using objects

Search queries using outlines
 Search queries using profiles
 Search queries using baselines

Date from:
 Date to:
 Max. returned statements:

Search

Queries using plan obj...

FIND RESULTS

Hash Value	Outline name	Profile name	Baseline plan name	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Disk reads [MB]	Buffer gets [Blocks]	Rows processed	Query text
1792761674	XXX13			5 402.88	414.34	13 647 832	10 919 MB	1 792 761 674	13 647 345	SELECT SUM(DECODE(CTL

Another functionality available on the SQL Details page is the ability to report statistics for a given query. To do this, click the [Report] button.



This function allows you to display the statistics of the query in a new window from which among other things, an ability to copy information.

As a result, following is obtained:

DBPlus Performance - SQL Details for 3715327 - Google Chrome

about:blank

Hash value: **3715327** Sql Id: **0nhfkk403jc7z**

SQL Text

```
SELECT MAX (SNAP_ID) FROM DBPLUS_SNAPS WHERE LOGDATE < = SYSDATE - :b1
```

Date	Plan hash	Elapsed Time [Seconds]	Cpu Time [Seconds]	Rows processed [Rows]	Fetches [Rows]	Executions	Parse Calls	Disk Reads [Blocks]	Disk Reads [MB]	Buffers Get [Blocks]	Buffer Quality (%)	Elapsed Time per 1 Exec [Seconds]
2018-08-27	3145828564	5.1	2.2	16	16	16	16	924	7 MB	1 498 536	99.9	0.3205
2018-08-28	3145828564	5.2	2.2	16	16	16	16	926	7 MB	1 500 060	99.9	0.3251
2018-08-29	3145828564	5.1	2.2	16	16	16	16	927	7 MB	1 501 515	99.9	0.3195
2018-08-30	3145828564	0.1	0.0	0	0	0	0	0	0 MB	1 469	100.0	0.0776

PLAN Hash Value: **3145828564**

```
SELECT STATEMENT
SORT (AGGREGATE)
TABLE ACCESS (BY INDEX ROWID) DBPLUS_SNAPS
INDEX (RANGE SCAN) DBPLUS_SNAPS_LOGADATE
```

6.2.1.6 SQL Plan Tab

The "SQL Plan" tab allows for detailed information about a given query plan and queries using it. Visually the screen retains similar functionality to the SQL Details tab.

The screenshot shows the SQL Plan tab interface. At the top, there are navigation tabs: Database Load, Waits, Latches, SQL Analyze, SQL Details, **SQL Plan**, Load Trends, Compare, Top SQL, SQL 3D, Top Day, Slow SQLs, Perf Counters, and OS Stat. Below the tabs, there is a search bar with 'Plan hash: 3250272785' and filters for 'From: 2018/08/28 00:00' to '2018/08/30 23:59'. A 'Group by query' checkbox is checked and highlighted with a red box. A 'Refresh' button is also present.

The main area displays 'PLAN EXECUTION STATISTICS' as a table:

Date	Elapsed Time [Seconds]	Cpu Time [Seconds]	Rows processed [Rows]	Fetches [Rows]	Executions	Disk Reads [Blocks]	Buffers Get [Blocks]	Elapsed Time per 1 Exec [Seconds]
2018-08-30	85 848.1	32 872.0	696 624	62 171	41 475	843 830	3 096 983 319	2.069
2018-08-29	116 223.7	48 682.1	1 274 137	113 741	76 167	2 754 889	5 011 133 201	1.525
2018-08-28	118 389.2	49 348.5	1 318 365	117 717	78 934	206 867	5 063 634 594	1.499

Below the table, there are tabs for 'Explain plan', 'Graph', and 'Statements using plan'. The 'Explain plan' tab is active, showing a tree view of the query plan:

```

Show plan objects for 3250272785
SELECT STATEMENT ( Cost = 9924 , Bytes = 0 , Cardinality = 0 , Search Columns = 0 )
├── FILTER
│   ├── TABLE ACCESS (BY INDEX ROWID) DOR_MA ( Cost = 5 , Bytes = 188 , Cardinality = 1 , Search Columns = 0 )
│   │   └── INDEX (RANGE SCAN DESCENDING) DOR_MA_DAT_M ( Cost = 4 , Bytes = 0 , Cardinality = 1 , Search Columns = 3 )
│   └── TABLE ACCESS (BY INDEX ROWID) DOR_MA ( Cost = 4 , Bytes = 11 , Cardinality = 1 , Search Columns = 0 )
│       └── INDEX (UNIQUE SCAN) DOR_MA_PK ( Cost = 3 , Bytes = 0 , Cardinality = 1 , Search Columns = 1 )
└── TABLE ACCESS (BY INDEX ROWID) DOR_SA ( Cost = 4 , Bytes = 12 , Cardinality = 1 , Search Columns = 0 )
    └── INDEX (UNIQUE SCAN) DOR_SA_PK ( Cost = 3 , Bytes = 0 , Cardinality = 1 , Search Columns = 1 )
    
```

SQL plan window is divided into areas:

- Filter fields – containing the ID plan for the query and date field. This is used to determine the time period in which the search should focus on.

The area also includes the option of selecting grouping after the query identifier. Mark the 'Group by query' checkbox, information will be returned to be grouped by the query identifier for the plan.

This close-up shows the 'Group by query' checkbox, which is checked and highlighted with a red box. It is located next to the search filters and the 'Refresh' button.

It is also possible to display online statistics for queries that use a given query plan.

This close-up shows the 'Online values' checkbox, which is checked, and a green 'Refresh' button next to it.

- Clipboard with execution plan IDs

The screenshot shows a 'Clipboard with execution plan IDs' window. It has a green clipboard icon at the top right. Below it is a list titled 'Plan Hashes list' containing the following IDs: 779893416, 3908273366, 2813991789, 1404789149, 2271935929, and 139336083. At the bottom left is a 'Clear list' button, and at the bottom right is another green clipboard icon.

- **Plan Execution Statistics** presents the following information:

Date	Hash value	Elapsed Time [Seconds]	Cpu Time [Seconds]	Rows processed [Rows]	Fetches [Rows]	Executions	Disk Reads [Blocks]	Buffers Get [Blocks]	Elapsed Time per 1 Exec [Seconds]
2018-08-28	2165587507	735.1	303.4	1 076	372	371	1	32 860 535	1.9814
2018-08-28	177336216	595.5	256.1	39 911	2 584	1 380	34	29 454 396	0.4315
2018-08-28	507106686	486.5	198.6	393	271	271	1	21 230 192	1.7952
2018-08-28	722351551	1 445.8	592.3	1 185	561	559	19	63 302 669	2.5864
2018-08-28	839632487	375.0	158.4	699	75	58	0	17 844 916	6.4653
2018-08-28	1015965212	183.0	75.3	22	35	35	2	8 169 689	5.2285
2018-08-28	1320996319	259.8	103.9	1 498	165	123	34	9 698 189	2.0392

- Date - date of query execution.
- Hash Value – query ID ('Group by Query' option selected)
- Elapsed Time [Seconds] - total time in seconds of the duration of the query for the selected grouping period.
- CPU time [Seconds] - total time in seconds of CPU utilization for the query for the selected grouping period.
- Rows processed [Rows] – number of returned records
- Fetches [Rows] – number of download lines form query pointer
- Executions - The number of query executions for the selected grouping period
- Disk Reads [Blocks] - The number of disk reads for a query for the selected grouping period
- Buffet Gets [Blocks] - The number of buffers utilized for the query for the selected grouping period
- Elapsed time per 1 exec [Seconds] - duration of a single query execution for the selected grouping period

- **Explain Plan**

Explain plan Graph Statements using plan

Show plan objects for 4055344431

```

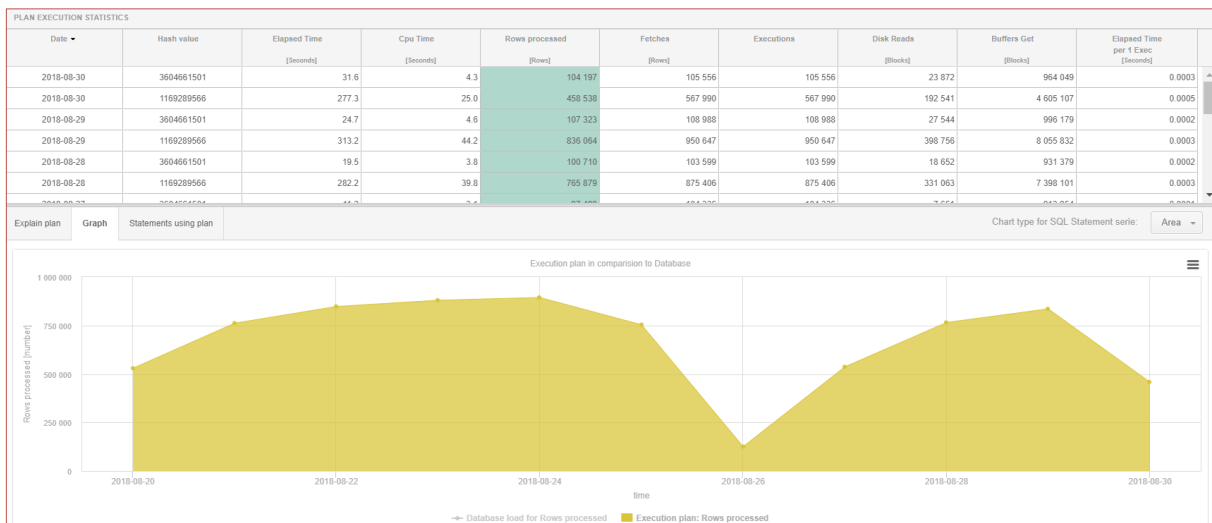
SELECT STATEMENT ( Cost = 5 , Bytes = 0 , Cardinality = 0 , Search Columns = 0 )
├─TABLE ACCESS (BY INDEX ROWID) DBPLUS_TAB4 ( Cost = 5 , Bytes = 103 , Cardinality = 1 , Search Columns = 0 )
│   └─INDEX (RANGE SCAN) DBPLUS_TAB4_NUM9 ( Cost = 4 , Bytes = 0 , Cardinality = 1 , Search Columns = 4 )
│       └─SORT (AGGREGATE)
│           └─FIRST ROW ( Cost = 4 , Bytes = 31 , Cardinality = 1 , Search Columns = 0 )
│               └─INDEX (RANGE SCAN (MIN/MAX)) DBPLUS_TAB4_NUM9 ( Cost = 4 , Bytes = 31 , Cardinality = 1 , Search Columns = 2 )

```

The screen presents a detailed execution plan for a given SQL query.

- **Graph**

The Graph tab, displays the graph for the selected column. To do this, indicate the selected column in the grid for the given query plan. The chart also provides information on how many percent of a given plan is in the context of the entire database (analogically to the SQL Analyze tab).



- Statements Using plan

The table shows queries that use the given execution plan, which allows for a more complete assessment of the situation, it often turns out that the same plan is used by a very similar query.

- Statement Text

Hash Value	Sql Id	Elapsed Time [Seconds]	Cpu Time [Seconds]	Rows processed [Rows]	Fetches [Rows]	Executions	Disk Reads [Blocks]	Buffers Get [Blocks]	Elapsed Time per 1 Exec [Seconds]
2708945910	22hb5n2hrffzq	62.0	48.1	142	1 946	1 946	0	0	0.0319
3488042108	a9ttjcm7yfm3w	2.0	2.0	6 834	130	65	0	0	0.0303

STATEMENT TEXT FOR HASH VALUE: 2708945910

```

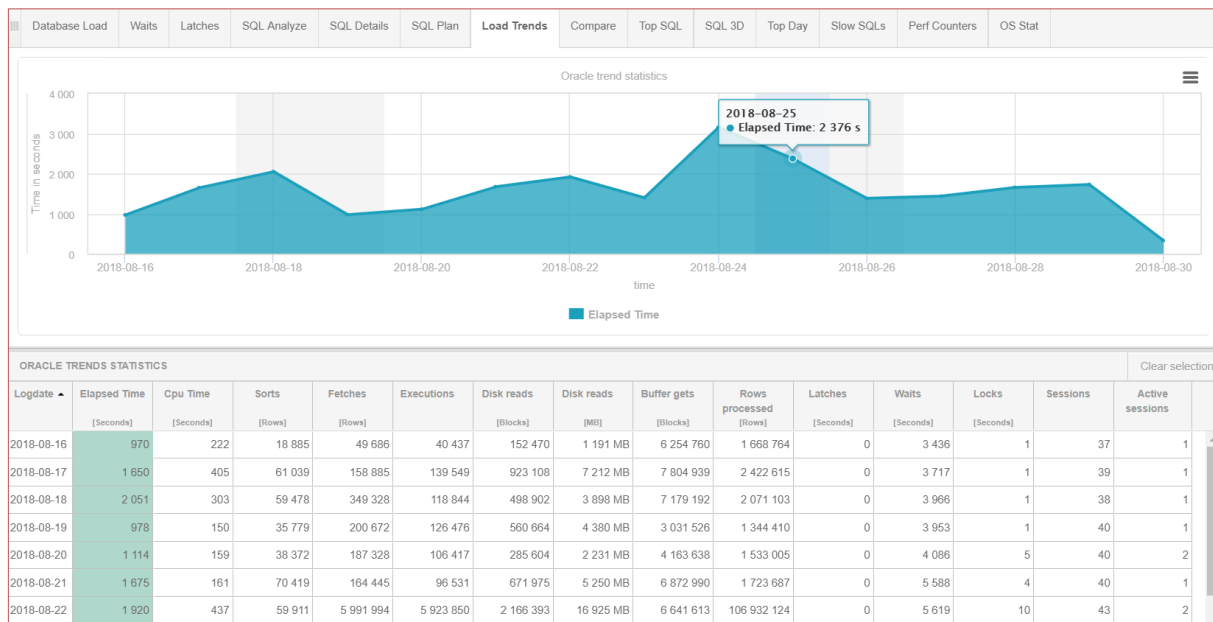
select sql_text, length(sql_text) as sql_length, sorts, loaded_versions, open_versions, users_opening, fetches, executions, users_executing, loads, first_load_time,
invalidations, parse_calls, disk_reads, buffer_gets, rows_processed, command_type, optimizer_mode, optimizer_cost, parsing_user_id,
parsing_schema_id, kept_versions, address, hash_value, plan_hash_value, child_number, module, module_hash, action, action_hash,
serializable_aborts, outline_category, cpu_time, elapsed_time, outline_sid, child_address, sqltype, remote, object_status,
literal_hash_value, last_load_time, is_obsolete, child_latch, sql_id FROM v$sql WHERE elapsed_time < :long_sql_min_ElapsedTime and executions < :minExec and last_load_time >
To_Char((sysdate - :daysAgo), 'YYYY-MM-DD/HH24:MI:SS')

```

Shows identifiers and the contents of all SQL queries that use the execution plan.

6.2.1.7 Load Trends Tab

Load Trends tab allows for detailed information on trends in ORACLE database.



The page consists of three components:

- Filter with the date range and grouping option
- Graph presenting certain indicators over time
- The table of statistics

Information displayed on the graph can be shown in groups of:

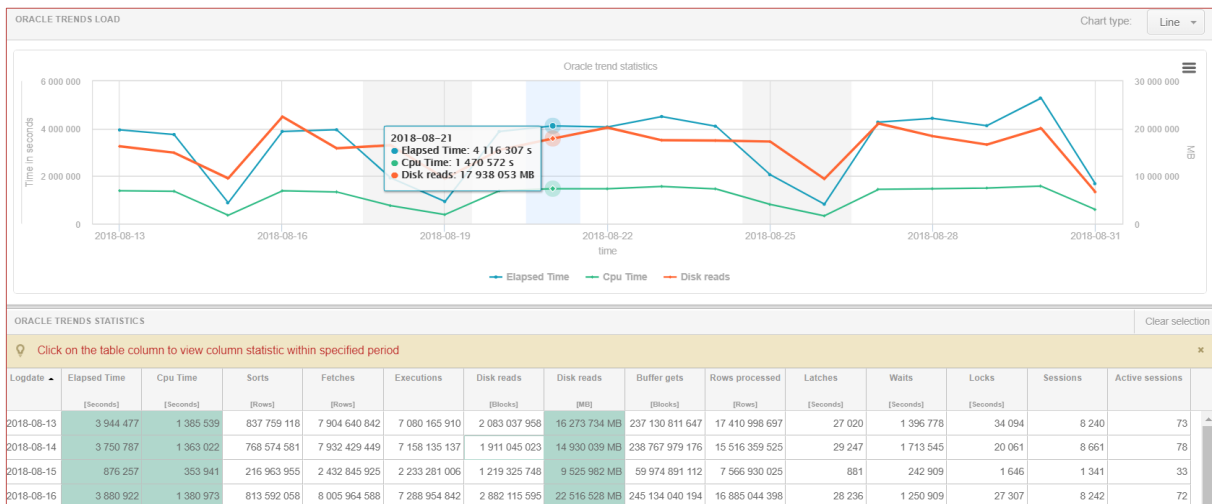
- **No group by period** – with no grouping; selection of date range
- **Month** – statistics broken for months
- **Day** – statistics broken by day
- **Hour** – statistics broken by one hour
- **Snap** – statistics broken by 15 minutes

Load Trends Statistics include the following information:

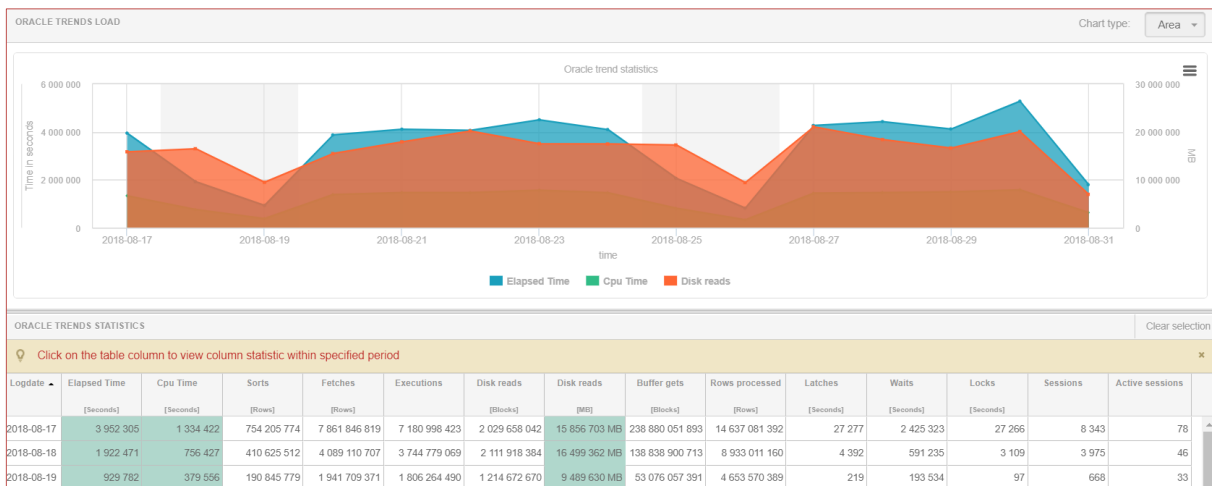
- Log date - represents the point in time for which the statistics are presented (i.e.: day, hour, minutes, for the entire period)
- Elapsed Time [Seconds] - total length of time in seconds of all queries for the selected grouping period
- CPU Time [Seconds] – total time of utilization CPU for query for the selected grouping period,
- Application wait Time,
- Concurrency wait Time,
- Cluster wait Time,
- User IO Wait Time,
- PLSQL Exec Time,
- Java Exec Time
- Fetches [Rows] – number of rows downloads from the query cursor,
- Executions - number of performances of all searches for the selected grouping period
- Disk Reads [Blocks] – number of block readings for the selected grouping period
- Disk Reads [MB] - The number of disk reads for all queries for the selected grouping period
- Buffer Gets [Blocks] - number of utilized buffers for all queries for the selected grouping period
- Rows processed [Rows] - number of rows processed by all queries for the selected grouping period

- Latches [Seconds]- total time in seconds the duration of all latches that occurred for the selected grouping period
- Waits [Seconds]- total time in seconds spent on all waits including latches that occurred for the selected grouping period
- Locks [Seconds]- total time in seconds of the duration of all locks that have occurred for the grouping period
- Sessions – average number of logged users
- Active sessions – average number of active sessions

Clicking selected columns presents their behavior as function of time:



Changing the graph type to 'Area' results in Graph changes to the example below:



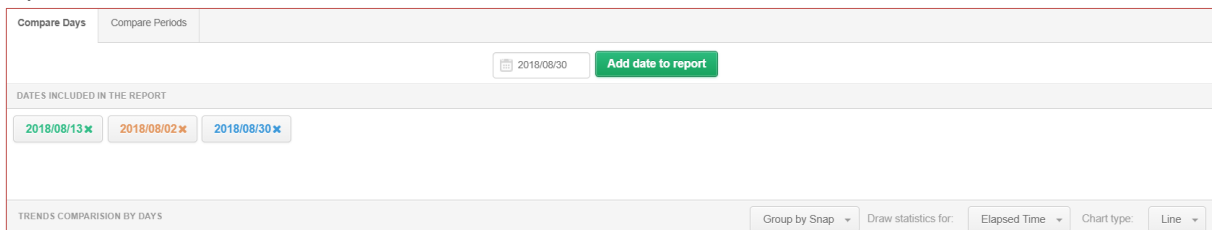
6.2.1.8 Compare Tab

Compare tab allows for statistics comparison. These can be compared by either days or periods.

Information are presented in three areas:

- Selection of comparing method – Compare Days or Compare Periods
- Date, range of date and statistic selection
- The graph presenting specific indicators over time

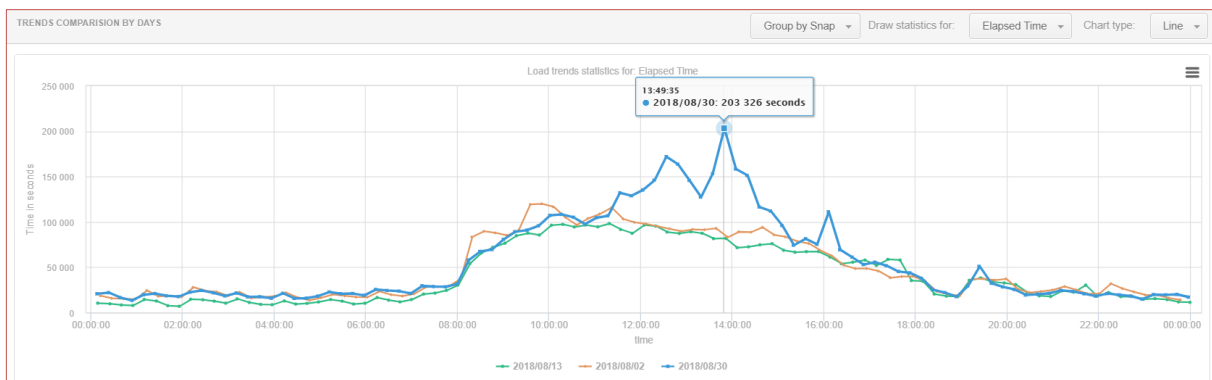
Comparing databases by day [**Compare Days**], any day can be picked from the calendar and added to the report.



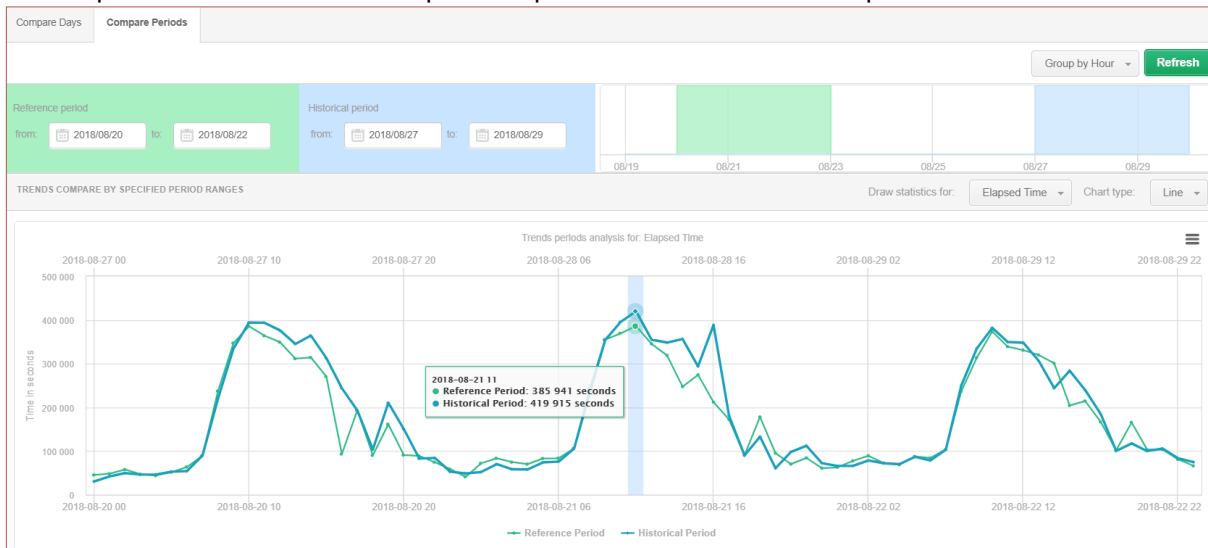
Two possibilities are available when comparing data, these are:

- Grouping divided into snapshots – periods of 15 minutes
- Grouping divided into one-hour periods

At the end, a choice of type of statistics for which the graph should be generated. Below a graph for three days indicated, grouped into snapshots for the [Elapsed Time] statistics is presented.



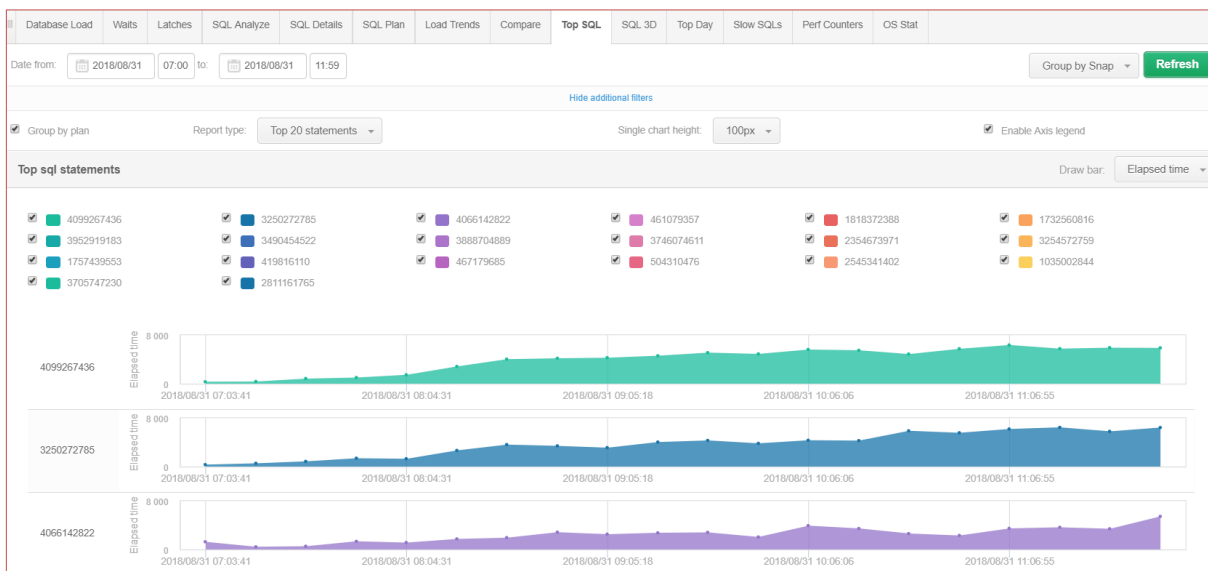
Comparing databases by period [Compare Periods] filtering options available are as for comparison after day. The chart below shows a comparison between two periods from 20.08 to 22.08 and 27.08 to 29.08, the chart is presented for one-hour samples and presents the statistics of Elapsed Time.



6.2.1.9 Top SQL Tab

The data presented on the Top SQL tab presents the most demanding queries depending on whether the user is interested in the query execution time, the number of read data, the number of processed blocks from memory (Buffer Gets), or the number of readings from disk (Disk Reads), etc.

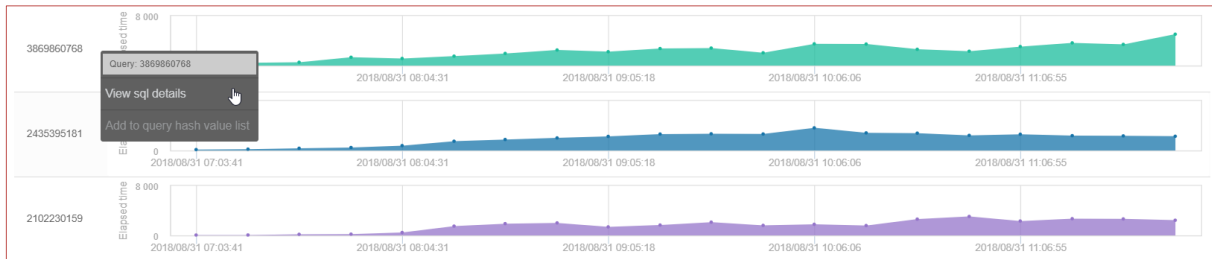
Queries are presented in the form of graphs in descending order according to the duration of the query in the selected time period for Elapsed time or other selected indicator.



From the [Top SQL] screen, any query can be easily added to the [SQL Details] by clicking the [Plus] button next to the query identifier and clicking the options:

- SQL View details - to move to the SQL Details screen and analyze specific query
- Add to query hash list - to add the query to the clipboard with a list of questions for further analysis

If the query is grouped by the query plan (selected checkbox [Group by plan]), clicking the [Plus] button, adds the query plan identifier which will be available in the [SQL Plan] tab).



Deleting individual charts from the [Top SQL] view can be done using the checkboxes in the legend.

Top sql statements Draw bar: Elapsed time

<input checked="" type="checkbox"/> 232111811	<input checked="" type="checkbox"/> 2348013182	<input checked="" type="checkbox"/> 3995904050	<input checked="" type="checkbox"/> 2072701542	<input checked="" type="checkbox"/> 1099793455	<input checked="" type="checkbox"/> 2085529658
<input checked="" type="checkbox"/> 1783198728	<input checked="" type="checkbox"/> 2800190848	<input checked="" type="checkbox"/> 1834802753	<input checked="" type="checkbox"/> 557193341	<input checked="" type="checkbox"/> 2137181839	<input checked="" type="checkbox"/> 2835489409
<input checked="" type="checkbox"/> 3209883117	<input checked="" type="checkbox"/> 2345318872	<input checked="" type="checkbox"/> 2568388571	<input checked="" type="checkbox"/> 781475369	<input checked="" type="checkbox"/> 3272558210	<input checked="" type="checkbox"/> 1045122158
<input checked="" type="checkbox"/> 1928966091	<input checked="" type="checkbox"/> 4253554421				

The right side presents a choice to show top queries by selected filter:

- Elapsed time
- CPU Time
- Sorts
- Fetches
- Disk Reads
- Buffer Gets
- Rows Process
- Executions

After clicking on the Show Additional filters link additional possibilities are presented including:

- Display statistics by execution plans
- Change the report type to:
 - o Top 20 queries
 - o Top 20 Procedures
 - o Queries generating majority of Log files
- Change the size of the charts for the presented queries
- Adding the name of the presented statistic to the Y axis in the chart.

Hide additional filters

Group by plan Report type: Top 20 statements Single chart height: 100px Enable Axis legend

Top sql statements Draw bar: Elapsed time

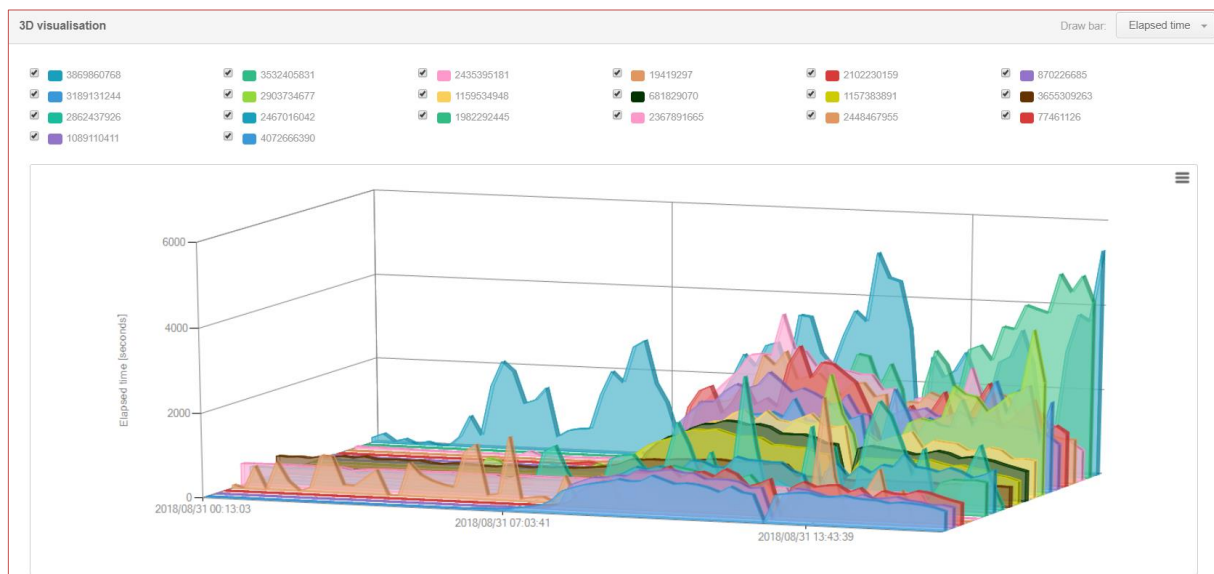
<input checked="" type="checkbox"/> 2283087408	<input checked="" type="checkbox"/> 4239951450	<input checked="" type="checkbox"/> 3053748652	<input checked="" type="checkbox"/> 448096503	<input checked="" type="checkbox"/> 359489790	<input checked="" type="checkbox"/> 2258898954
<input checked="" type="checkbox"/> 3540075994	<input checked="" type="checkbox"/> 3551382599	<input checked="" type="checkbox"/> 2399208389	<input checked="" type="checkbox"/> 1022258389	<input checked="" type="checkbox"/> 1950795881	<input checked="" type="checkbox"/> 2939867596
<input checked="" type="checkbox"/> 1935744842	<input checked="" type="checkbox"/> 0	<input checked="" type="checkbox"/> 378289517	<input checked="" type="checkbox"/> 903871040	<input checked="" type="checkbox"/> 2432278962	<input checked="" type="checkbox"/> 3680778372
<input checked="" type="checkbox"/> 1575148752	<input checked="" type="checkbox"/> 599100469				

6.2.1.10 SQL 3D Tab

Data presented on the SQL 3D tab presents the most demanding queries: execution time, number of read data, number of blocks processed from memory (Buffer Gets), number of reads from disk (Disk Reads), etc.

The data presented on this page is analogous to those presented in Top SQL. They differ in the way of presentation. In this case, an opportunity to look at the queries in one view is presented. It is easier to indicate which query at the time has the most impact on the indicator.

Each query can be freely unchecked by clicking on the checkbox for a given query. This will remove the query from the graph.



On the website an option to display queries for a given date range is available. It is possible to present data in samples for day, time and snap.

Similarly, as for the Top SQL, the graph can present data for indicators:

- Elapsed time
- CPU Time
- Application wait Time,
- Concurrency wait Time,
- Cluster wait Time,
- User IO Wait Time,
- PLSQL Exec Time,
- Java Exec Time
- Fetches
- Disk Reads [Blocks]
- Disk read [MB]
- Buffer Gets
- Rows Processed
- Executions

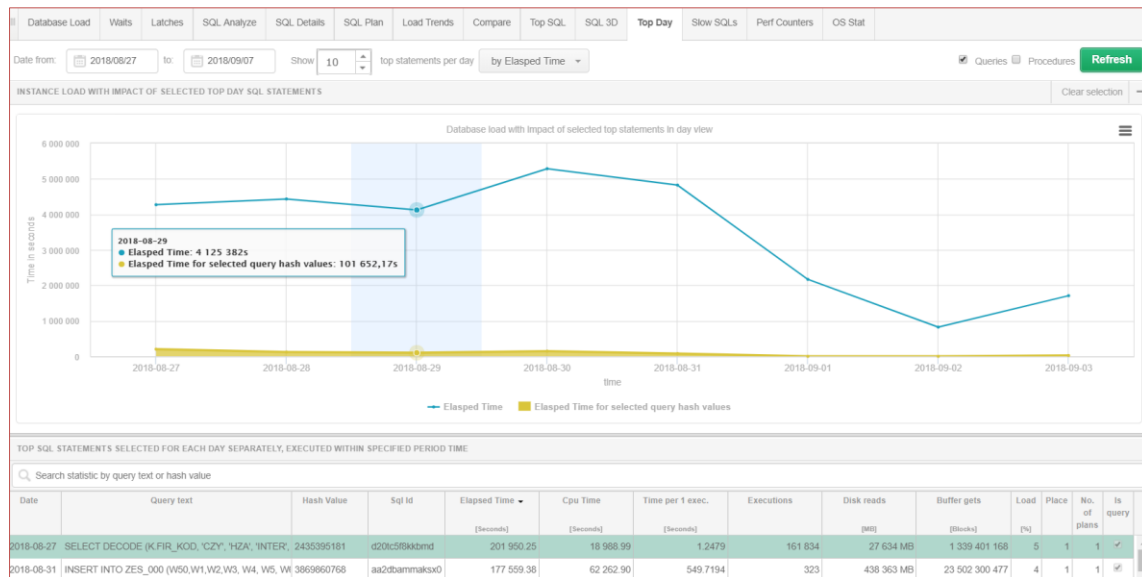
Selecting additional filters will allow for each question to be grouped by the plan as well as chart for top procedures or queries to the log.

After indicating the appropriate query on the graph, user can add them for further analysis by adding to the clipboard or going straight to the details (SQL Details tab).

Attention: In the case of selected group by plan [Group By Plan] clicking details, user is taken to the details of the given query plan (SQL Plan tab).

6.2.1.11 Top Day Tab

[Top Day] window allows to view top queries for CPU Time or Elapsed Time and track their behavior changes.



On the above slide, 10 top queries are presented in the period 27.08.2018 to 07/09/2018 the share of the first query against the background of the entire base load (the yellow area is the level of the selected query).

Conclusion: optimizing the selected queries database load can be reduced by 100%!

Table with top queries contains

- Date – the date the request was made,
- Query text – the content of the query,
- Hash Value – SQL query ID,
- SQLID – SQL query ID,
- Elapsed Time [Seconds] – the total execution time of the SQL query on a given day,
- CPU Time [Seconds]– total processor usage time,
- Time per 1 exec [Seconds] – the time of a single query execution
- Executions – number of executions on a given day for a given query,
- Disk reads [MB] – the amount of read data from the disk
- Buffer gets [Blocks] – the number of utilized buffers for all queries,
- Load [%] – percentage of database load,
- Place – the place where the given query affects the database on a given day
- No. of plans – number of execution plans for a given query,
- Is Query – whether a given record is a query or a procedure.

Below the table a **Statement Text** is located– text of the selected query. By checking the query in the table, user can drag query to chart **[Database Load]** and observe changes of its influence on the overall load of the database.

Remember about the possibility of detailed analysis of a particular query by clicking on the button [+] ["plus"] next to the query.

6.2.1.12 Slow SQLs Tab

On the tab system presents queries depending on the duration time. Queries which exceeded 200 seconds are presented by default (all queries for a given Hash Value).

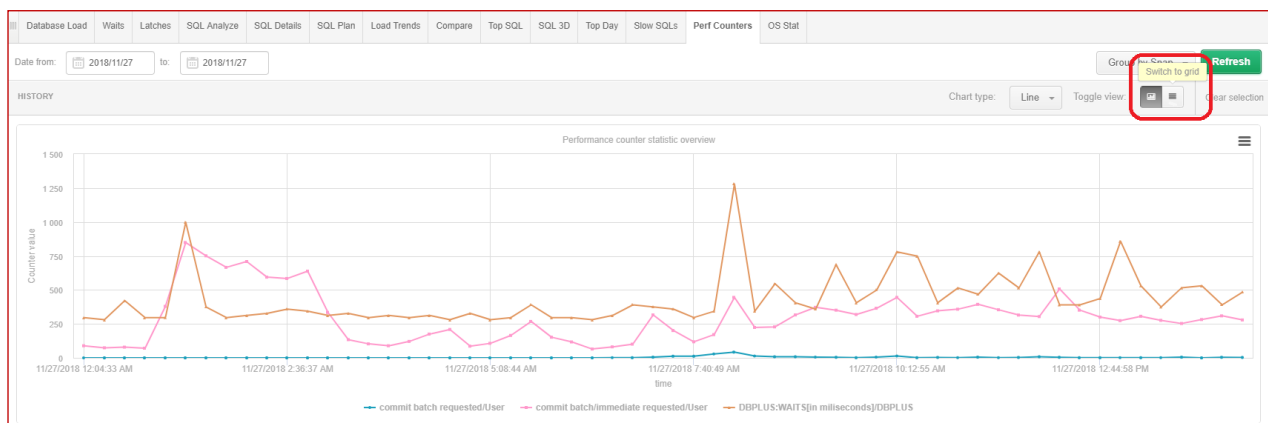
The screenshot shows the 'Slow SQLs' tab in the DBPLUS application. At the top, there are navigation tabs: Database Load, Waits, Latches, SQL Analyze, SQL Details, SQL Plan, Load Trends, Compare, Top SQL, SQL 3D, Top Day, Slow SQLs (selected), Perf Counters, and OS Stat. Below these, there are date filters (2018/05/03 to 2018/09/03) and a 'Refresh' button. The main area displays a table of SQL statements. The first row is highlighted in green. Below the table, the SQL statement text and the explain plan for the highlighted query are shown.

To find queries on the Slow SQL Tab user has the ability to change filter settings like duration time of queries, and has the option of filtering queries grouped by literals. The application, grouping the literals in place of parameters inserted in the query body inserts the character '#', then queries groups are displayed by query plan

6.2.1.13 Perf Counters Tab

This tab presents all database statistics available in the system view V\$SYSTAT.

Indicators can be selected for a given time range, grouped by day, hour and snap. User can present many different statistics on the chart at the same time. The application, the ability to export performance statistics. Export is possible by changing the chart preview to the tabular form [Switch to grid].



Export is performed for statistics previously selected from the table.

Date from: 2018/11/27 to: 2018/11/27 Group by Snap Refresh

HISTORY Toggle view: Clear selection

PERFORMANCE COUNTERS STATISTICS DURING SPECIFIED PERIOD TIME

Search performance counter by any value in below table

Name	Class
ADG parselock X get attempts	User
ADG parselock X get successes	User
Batched IO (bound) vector count	Batched IO
Batched IO (full) vector count	Batched IO
Batched IO (space) vector count	Batched IO
Batched IO block miss count	Batched IO
Batched IO buffer defrag count	Batched IO
Batched IO double miss count	Batched IO
Batched IO same unit count	Batched IO
Batched IO single block count	Batched IO

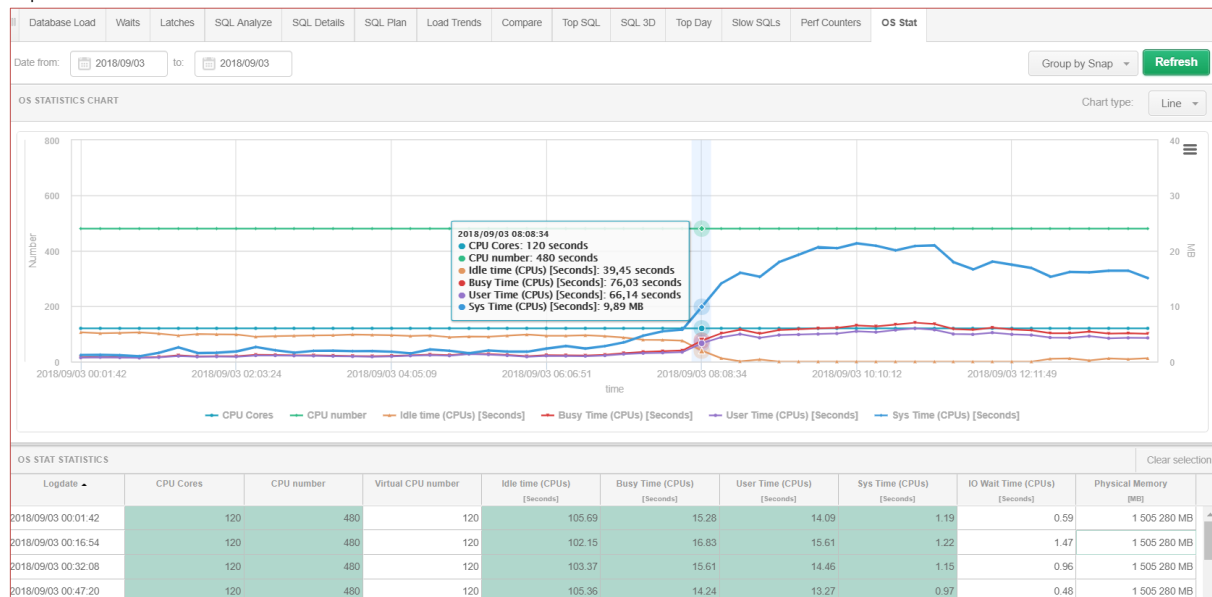
SELECTED COUNTERS DETAILS WITHIN SPECIFIED PERIOD

Logdate	ADG parselock X get attempts/User	Batched IO (space) vector count/Batched IO	Batched IO same unit count/Batched IO	Batched IO buffer defrag count/Batched IO	Batched IO double miss count/Batched IO
2018-11-27 00:03:37	0	0	205 614	3 718	3 209
2018-11-27 00:18:50	0	0	159 869	3 273	993
2018-11-27 00:34:01	0	0	519 359	5 021	3 005
2018-11-27 00:49:14	0	0	19 402	599	1 075
2018-11-27 01:04:26	0	12 830		2 112	24 473
2018-11-27 01:19:38	0	0		1 211	5 344
2018-11-27 01:34:51	0	0		631	7 863
2018-11-27 01:50:03	0	0	322 606	2 322	13 419

Grid option
Export grid
Export grid with formatted data

6.2.1.14 OS Stat Tab

The tab presents information about the operating system statistics presented in the system view of V\$OSSTAT.



The statistics available in the table:

- CPU Cores – number of available processor cores,
- CPU number – number of available processor sockets
- Virtual CPU number – the number of virtual processors,

- Idle time (CPUs) [Seconds] – the number of processor inactivity seconds, relative to all processors
- Busy Time (CPUs) [Seconds] - number of seconds in which the processor was busy executing the user or kernel code, including all processors (sum of User Time, Sys Time),
- User Time (CPUs) [Seconds] - the number of seconds the processor was busy executing the user code, including all processors,
- Sys Time (CPUs) [Seconds] - the number of seconds that the processor was busy executing the kernel code, summarized for all processors,
- IO Wait Time (CPUs) [Seconds] the number of seconds the processor waited for the I / O to complete, for all processors in total,
- Physical Memory [MB] - total amount of physical memory.

6.2.2 Menu Plan explorer

Application allows you to manage objects such as:

- Outlines
- Baselines
- Profiles

Plan Explorer menu, available from Database Analysis for each database. The screen provides information about all Outlines / Baselines / Profiles objects established in a given database. Available current information as well as historical data.

Browsing information about Outlines, such information's are presented:

- Signature - a unique SQL text identifier
- Name - name of the Outline created,
- Owner - the name of the user creating the Outline,
- Category - the category assigned when the Outline was created,
- Used - contains information on the use of Outline by the query, [UNUSED] - Outline has never been used by the query,
- Timestamp - date of creation of Outline,
- Version - Oracle version for which Outline was created,
- Sql id - query identifier,
- Hash Values - query identifier,
- Statement text - query text (SQL text),
- Compatible - whether outline hints for use are compatible during migration,
- Enabled - information whether Outline is enabled,
- Format - tooltip format [NORMAL / LOCAL]
- Migrated - whether Outline has been migrated to SQL Baselines plan.

Attention! Not all Outlines will have assigned the SQL id / Hash Value. This will refer specifically to those Outlines that have been created in the past and for which queries are not currently performed.

The screenshot shows the 'Plan Explorer' interface with a table of Outline objects. The table has columns for Signature, Name, Owner, Category, Used, Timestamp, Version, Sql id, Hash Value, Statement text, Compatible, Enabled, Format, and Migrated. Below the table, there is a 'DETAILS FOR SELECTED PLAN OBJECT' section showing the SQL Text and a 'STATEMENT TEXT' section with a complex SQL query.

Signature	Name	Owner	Category	Used	Timestamp	Version	Sql id	Hash Value	Statement text	Compatible	Enabled	Format	Migrated
6f034E8D5A44DEE:XXX13		APPS	DEFAULT	USED	2019-08-21 14:59:36	11.2.0.4.0	a5awcdpdrjua	1792761674	SELECT SUM(DECO)	COMPATIBLE	ENABLED	NORMAL	NOT-MIGRATED
6f16494E208C2A1C:XXX08		APPS	DEFAULT	USED	2016-04-11 07:39:19	11.2.0.4.0	9u02u62y0yb	3425257387	SELECT unique_id	COMPATIBLE	ENABLED	NORMAL	NOT-MIGRATED
6f4F118AA28CEFE9:XXX05		APPS	DEFAULT	UNUSED	2015-12-18 12:28:02	11.2.0.4.0			SELECT SUM(account)	COMPATIBLE	ENABLED	NORMAL	NOT-MIGRATED
6f61B767FAA51BED:XXX09		APPS	DEFAULT	USED	2016-04-20 11:51:07	11.2.0.4.0			UPDATE /*+ use_nl	COMPATIBLE	ENABLED	NORMAL	NOT-MIGRATED
6f6EDD01E4308C72F:XXX11		APPS	DEFAULT	USED	2017-11-07 09:14:09	11.2.0.4.0	6366tuydm0gt	2084629369	UPDATE XLA_ACCT	COMPATIBLE	ENABLED	NORMAL	NOT-MIGRATED
6f72D367E12DDA36:XXX08		APPS	DEFAULT	USED	2016-04-11 11:33:01	11.2.0.4.0			SELECT SUM(CTLE)	COMPATIBLE	ENABLED	NORMAL	NOT-MIGRATED
6f95CC8BA7C5E202:XXX12		APPS	DEFAULT	USED	2018-04-11 11:45:24	11.2.0.4.0	922kpuab0s	1421126440	select data , data	COMPATIBLE	ENABLED	NORMAL	NOT-MIGRATED
6f9AA024B5DE4E4C:XXX10		APPS	DEFAULT	USED	2017-03-22 10:17:46	11.2.0.4.0			INSERT INTO XX_NF	COMPATIBLE	ENABLED	NORMAL	NOT-MIGRATED

STATEMENT TEXT

```

SELECT unique_to_char(OUT_MENU_ID) FROM fnd_grants OUT, FND_IRFP_FUNCTION_FLAVORS OUTTAB WHERE OUT.object_id = 4207 AND (OUT.grantee_key in (select role_name from wf_user_roles var, ( select 'SOATEST' name from dual union all (select incrn.name from wf_local_roles incrn, fnd_user f where 'S1_PARTY' = incrn.orig_system and f.user_name = 'SOATEST' and f.person_party_id = incrn.orig_system_id and incrn.partition_id = 9 ) ) incrn2 where
  
```

In addition, information about all Outline is stored in the database in the Outlines History tab. To search for a historical Outline, select the appropriate date range.

Another object that is made available for viewing is SQL Plan Baseline. The information available in the SQL Plan Baseline application includes:

- Signature - unique identifier for the SQL text
- Statement text - query text (SQL text),
- Sql id / Hash Value - query identifier,
- Baseline name plan - unique identifier Baseline plan
- Creator - user creating Baseline
- Origin - how the Baseline Plan was created:
 - MANUAL-LOAD
 - AUTO-CAPTURE
 - MANUAL-SQLTUNE,
 - AUTO-SQLTUNE
- Parsing schema - the name of the schema,
- Description - additional description,
- Version - database version at the time Baseline was created,
- Created - data when Baseline was created,
- Last modified - the date when Baseline was last modified,
- Last executed - the date when Baseline was last executed,
- Last verified - the date when Baseline was last verified,
- Enabled - [YES / NO] - information whether Baseline is available,
- Accepted - [YES / NO] - information whether Baseline is accepted,
- Fixed - [YES / NO] - information whether Baseline is repaired,
- Reproduced - [YES / NO] - indicates whether the optimizer could recreate the plan,
- Auto-purge - [YES / NO] - information whether Baseline is automatically cleaned,
- Optimizer cost - cost optimizer when Baseline was created
- Module - the name of the application module
- Action - action in the application.

For SQL Baseline, as well as for Outline, information about historical data is also available.

Signature	Statement text	Sql Id	Hash Value	Plan baseline name	Creator	Origin	Parsing schema	Description	Version	Created	Last modified	Last executed	Last verified	Enabled	Accepted	Fixed	Reproduced	Autopurge	Optimizer cost	Module	Action	Is Dropped
76154399481	SELECT dec			SQL_PLAN_APPS	APPS	MANUAL-LO	APPS		11.2.0.4.0	2018-10-16 1	2018-10-16 1	2018-10-16 1		YES	YES	NO	YES	YES	94778	XXZ0BR12_	Concurrent R	<input type="checkbox"/>
18086767021	SELECT dec			SQL_PLAN_APPS	APPS	MANUAL-LO	APPS		11.2.0.4.0	2018-10-16 0	2018-10-16 0			NO	YES	NO	YES	YES	195448	XXZ0BR12_	Concurrent R	<input type="checkbox"/>

The Plan Explorer tab also provides information on Sql Profiles. Information is available from the table DBA_SQL_PROFILES.

For each Outlines / Baselines / Profiles object, a function that also displays deleted objects (Drop) is available - functions can be started by selecting "Include dropped plan objects".

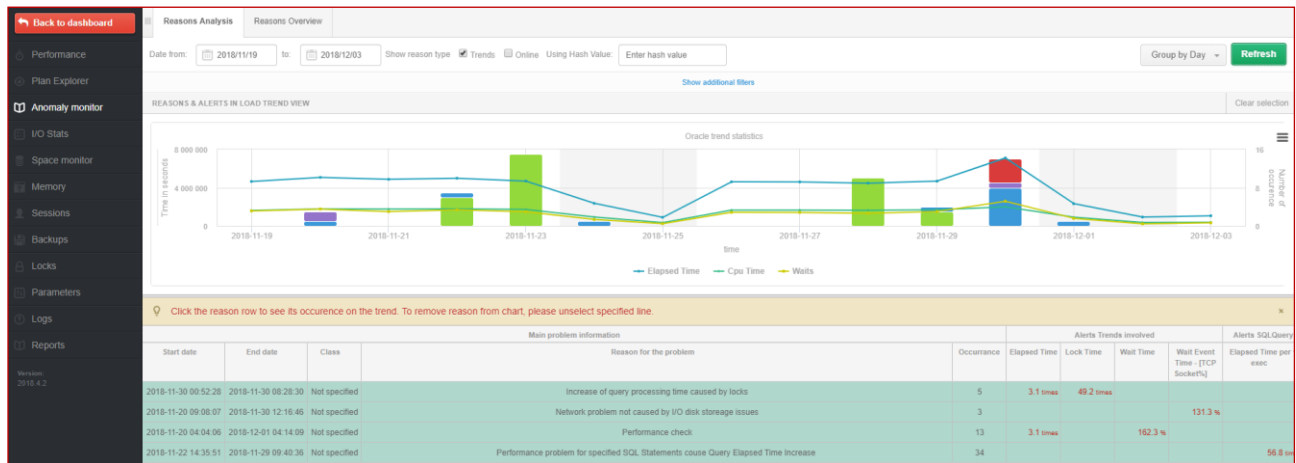
6.2.3 Menu Anomaly Monitor

The Anomaly Monitor allows to viewing anomalies (alerts) generated on the database. The browser is available from the Database Analysis> Anomaly Monitor database details.

6.2.3.1 Alerts viewer in the database

On the page user can choose between two tabs; Reasons Analysis and Reasons Overview.

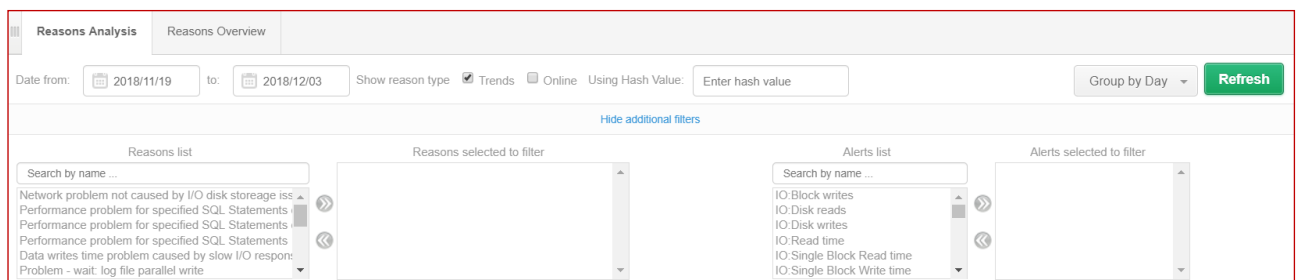
Reasons Analysis



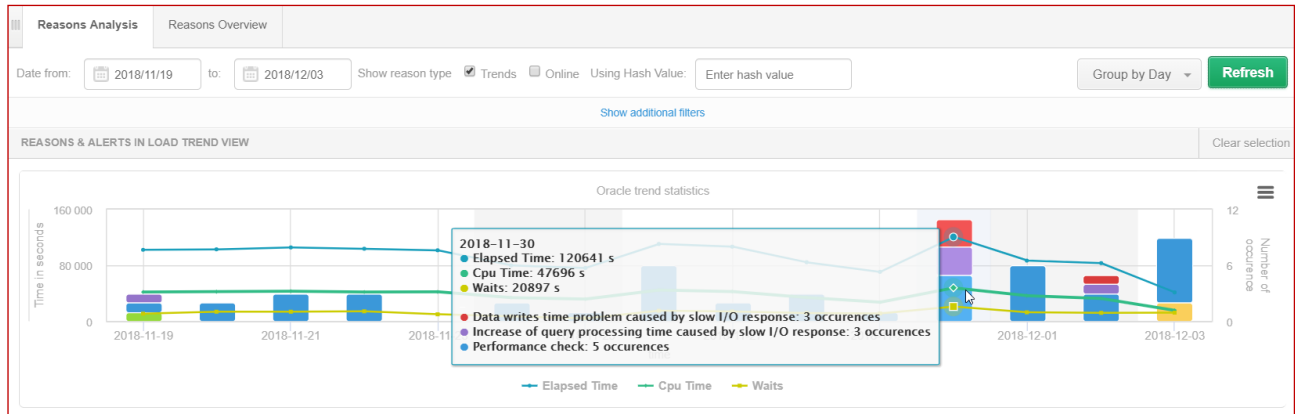
On the page you can choose several filters to help you find the problem you are looking for. As part of the filtering these options are available:

- choosing a date or range of dates,
- sorting after month, day, hour, snap,
- (Trends or Online) - the ability to indicate which type of alerts we want to view,
- Hash value - selection of alerts in which the indicated query identifier occurred,
- Reason list - the opportunity to indicate the dedicated causes of the problem,
- Alert lists - the ability to indicate dedicated alerts.

Screen of available filters on the Anomaly Monitor page:



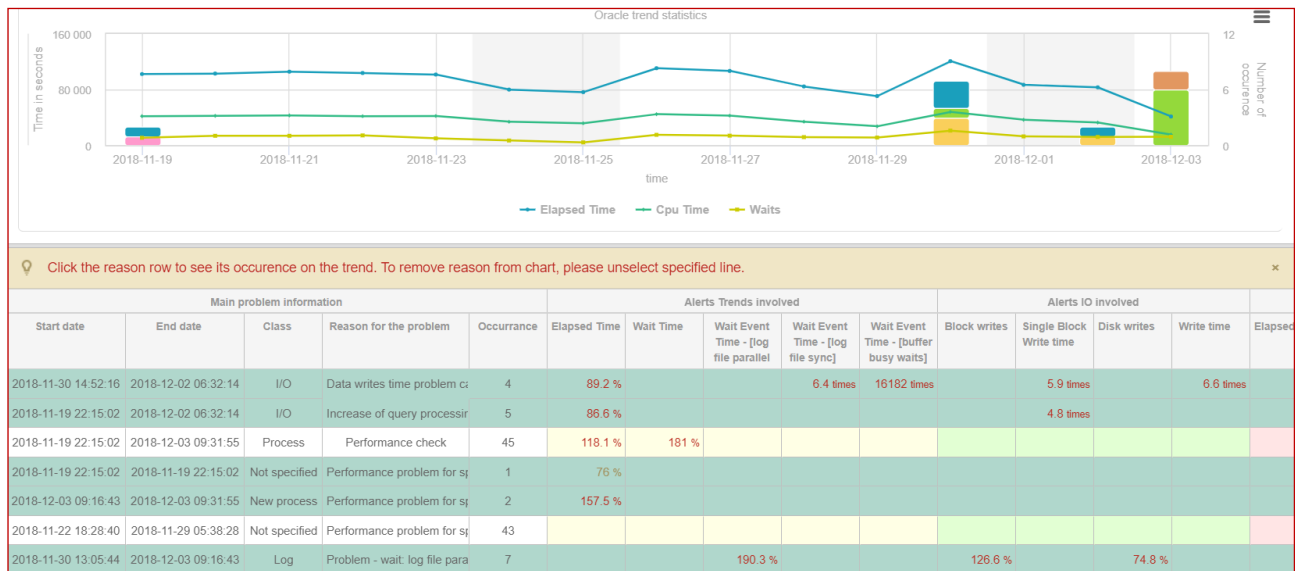
After configuring the appropriate filters, click the [Refresh] button. As a result, a graph will be presented in which, apart from the basic data bases such as Elapsed Time, CPU Time o Waits, the number of occurrences of a given problem per unit of time will be presented in the form of bars in the graph. By indicating a given bar on the chart, a tooltip will be presented with information containing data on basic measures as well as the number of instances of a given problem per unit of time.



Below the graph is a table containing additional information about the causes of alerts presented in the graph. The table contains:

- **Start date / End date** - date range in which the given reason occurred,
- **Class** - the class / area to which the given reason was assigned,
- **Reason for the problem** - the cause of the problem,
- **Occurrence** - the number of instances of a given cause in the selected date range
- **Alerts trends involved / IO involved / SQL Query involved** - sets of alerts included in the definition of a given problem cause.

The user can indicate the reasons for the problem in the table. Each selection / uncheck will convert the data in the chart and present only the selected rows.



The data contained in the table are average data for all occurrences of a given reason. For a more in-depth analysis of a given problem, after selecting a row, the table will display additional detailed information in the **Alerts Details** tab. This view contains information on alerts that have exceeded the thresholds defined for the given cause of the problem.

There is also a view grouping alerts of the same type that occurred after each other. This allows you to verify how long the problem lasted. This information is available in the Reasons Occurrence Statistics tab.

Reasons Occurrence Statistics		Alerts Details	
REASONS CHARACTERISTIC BETWEEN 2018-12-04 05:38:02 - 2018-12-14 17:44:14 FOR PERFORMANCE CHECK			
Start date	End date	Snapshots occurrence	Problem duration rounded to snap intervals [HH:MM:SS]
2018-12-04 05:38:02	2018-12-04 06:08:29	3	00:45:27
2018-12-04 11:12:55	2018-12-04 11:12:55	1	00:15:00
2018-12-05 05:59:20	2018-12-05 05:59:20	1	00:15:00
2018-12-06 05:34:58	2018-12-06 05:34:58	1	00:15:00
2018-12-07 15:56:14	2018-12-07 15:56:14	1	00:15:00

In the case below for one (Occurrence = 1) occurrence of the problem Data reads time problem caused by slow I / O response. Values for each of the alert defined for this problem which exceeded the threshold values were presented.

Alerts Details				
LIST OF ALERTS GENERATED IN 2018-11-16 22:19:43 FOR REASON DATA READS TIME PROBLEM CAUSED BY SLOW I/O RESPONSE				
Logdate	Level	Alert name	Hash value	Message
2018-11-16 22:19:43	Critical	Read time		Alert Type: I/O Stat, The measured statistic value is 110 % higher than average , Last value: 1814 s, Reference history value: 862.4 s
2018-11-16 22:19:43	Critical	Single Block Read time		Alert Type: I/O Stat, The measured statistic value is 135 % higher than average , Last value: 0.0035 s, Reference history value: 0.0015 s
2018-11-16 22:19:43	Warning	Elapsed Time		Alert Type: Load Trends, The measured statistic value is 81 % higher than average , Last value: 7557 s, Reference history value: 4164 s

Note: information in the Alerts Details tab is only available for the last selected cause of the problem.

Reasons Overview

As part of this tab, the application allows you to view problems in one set. We can choose the same filters as for the Reasons Analysis tab and additionally the option of marking / deselecting grouping after the Cause.

The screenshot shows the 'Reasons Overview' tab with the following elements:

- Filters:** Date from: 2018/11/01 to: 2018/12/03. Show reason type: Trends. Online. Using Hash Value: Enter hash value. A checkbox for 'Group by reason' is highlighted with a red box, and a 'Refresh' button is next to it.
- Reasons list:** Search by name... Network problem not caused by I/O disk storage iss, Performance problem for specified SQL Statements, Performance problem for specified SQL Statements, Performance problem for specified SQL Statements, Data writes time problem caused by slow I/O respon, Problem - wait. log file parallel write.
- Reasons selected to filter:** An empty list.
- Alerts list:** Search by name... IO:Block writes, IO:Disk reads, IO:Disk writes, IO:Read time, IO:Single Block Read time, IO:Single Block Write time.
- Alerts selected to filter:** An empty list.

Depending on the checkbox **[Group by reason]**, alert data will be displayed in various lists:

- selected

REASONS & ALERTS OVERVIEW	
Logdate	Reason name
2018-12-02 06:32:14	I/O>Data writes time problem caused by slow I/O response
	Single Block Write time Alert Type: I/O Stat, The measured statistic value is 10.5 times higher than allowed maximum , Last value: 1.87 s, Reference history value: 0.1623 s
	Write time Alert Type: I/O Stat, The measured statistic value is 2.6 times higher than allowed maximum , Last value: 10137 s, Reference history value: 2849 s
	Wait Event Time Alert Type: Load Trends, The measured statistic value is 119 % higher than average , Wait. log file sync, Last value: 60.6 s, Reference history value: 27.6 s
	Elapsed Time Alert Type: Load Trends, The measured statistic value is 66 % higher than average , Last value: 1769 s, Reference history value: 1067 s

- unselected

REASONS & ALERTS OVERVIEW					
Logdate	Reason	Level	Alert name	Hash value	Message
2018-12-02 06:32:14	I/O/Data writes time problem caused by slow I/O response	Critical	Single Block Write time		Alert Type: I/O Stat, The measured statistic value is 10.5 times higher than allowed maximum , Last value: 1.87 s, Reference history value: 0.1623 s
2018-12-02 06:32:14	I/O/Data writes time problem caused by slow I/O response	Critical	Write time		Alert Type: I/O Stat, The measured statistic value is 2.6 times higher than allowed maximum , Last value: 10137 s, Reference history value: 2849 s
2018-12-02 06:32:14	I/O/Data writes time problem caused by slow I/O response	Critical	Wait Event Time		Alert Type: Load Trends, The measured statistic value is 119 % higher than average , Wait: log file sync, Last value: 60.6 s, Reference history value: 27.6 s
2018-12-02 06:32:14	I/O/Data writes time problem caused by slow I/O response	Warning	Elapsed Time		Alert Type: Load Trends, The measured statistic value is 66 % higher than average , Last value: 1769 s, Reference history value: 1067 s
2018-12-02 06:32:14	I/O/increase of query processing time caused by slow I/O response	Critical	Single Block Write time		Alert Type: I/O Stat, The measured statistic value is 10.5 times higher than allowed maximum , Last value: 1.87 s, Reference history value: 0.1623 s

6.2.4 Menu I/O Stats - Database Analysis

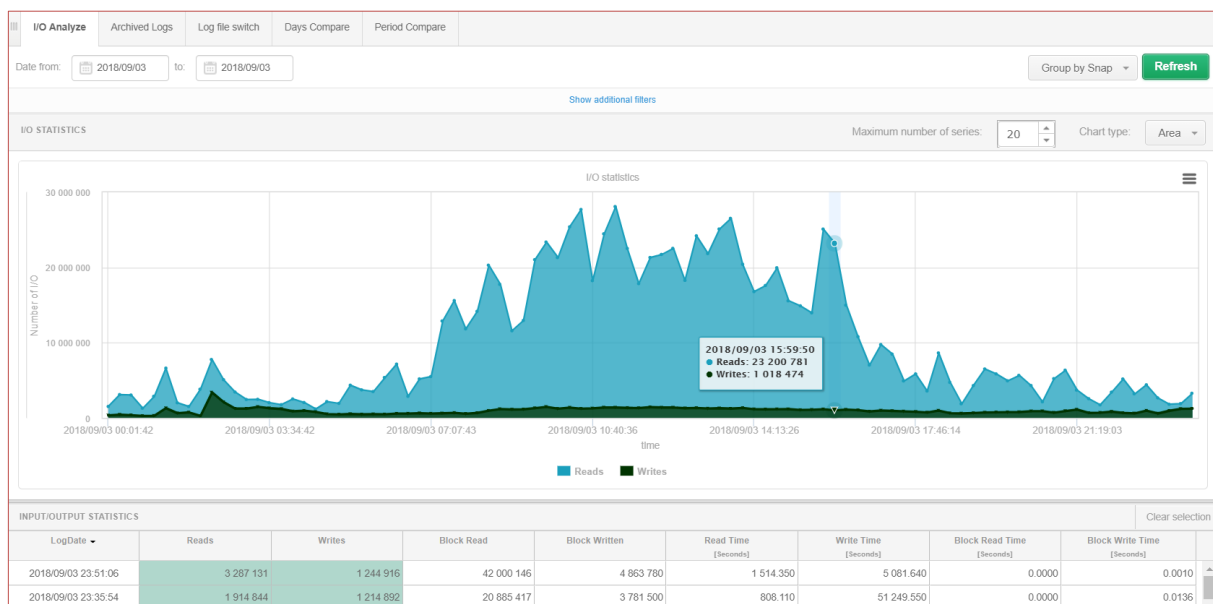
The screen is accessed from the menu on the left side and is used to analyze the performance of disk components. **[I/O Analyze]** functionality allows you to see any performance problems on disk devices, among others comparing the performance of writing and reading for specific days, hours, data files, tablespaces as well as collectively for the entire database

[I/O Stats] consists of the following features:

- I/O Analyze
- Archived Log
- Log file switch
- Days Compare, Period Compare - functionality that allows to compare IO operations at specified days or periods of time.

6.2.4.1 I/O Analyze Tab

The **[I/O Analyze]** tab allows to see any performance problems on disk devices including comparison of the efficiency of records and readings for particular days, hours, data files, table space as well as in a collective manner for the whole database.



The window is divided into the following sections:

- Filters area with the range of dates and additional filters
- Chart for presentation of specific indicators
- Table showing statistics of:
 - Reads - the number of reads
 - Writes - the number of data writes by DBWR
 - Block Reads - the number of read blocks
 - Block Writes - number of wrote blocks
 - Read Time - time to read blocks
 - Write Time - time to write blocks
 - Block Read Time - time to read a single block
 - Block Write Time - time to write a single block

Group by period - allows you to choose the period by which the grouping will occur

- **Day** - result is grouped by day
- **Hour** - result is grouped by hour
- **Snap** - result is grouped by periods of 15 minutes
- **None** - grouping is switched off, will show the total sum for a selected period for database, data files or table space depending on which filter is used

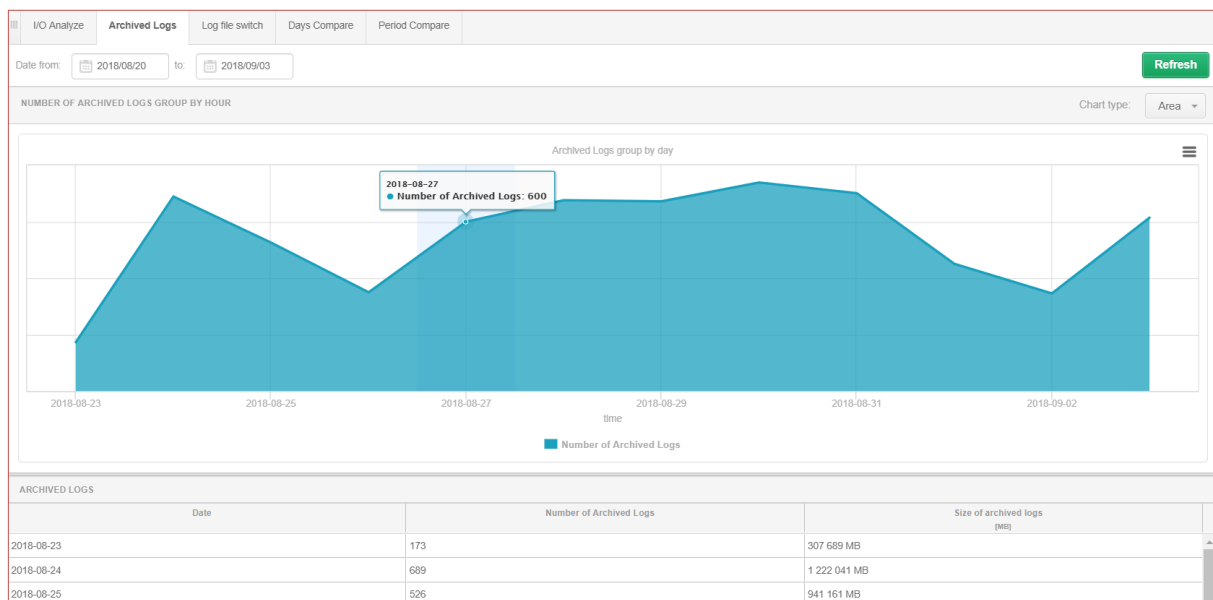
Additional filters are available by clicking on **[Show additional filters]** link, among others:

- Grouping by space or data file
- The filter on the data file
- The filter on the tablespace

It allows for specific I/O in a file or data space analysis.

6.2.4.2 Archive Logs Tab

On the page shows the number of archive files broken down by days



The window is divided into two parts:

- **Statistics Table:**
 - Date - the date for which statistics are shown
 - Number of Archive Logs - the number of archive files, which was created on that day
 - Size of Archived Logs (bytes) - the sum in bytes of all archive files that were created on that day
- Graph "Number of Archived Logs Group by Day"

Hovering the mouse over a graph's bar allows for extra details showing how many and what size archive files were generated.

The Y axis graph shows the number of created archive files.

The X axis shows the time of creation of the files with an accuracy of one day

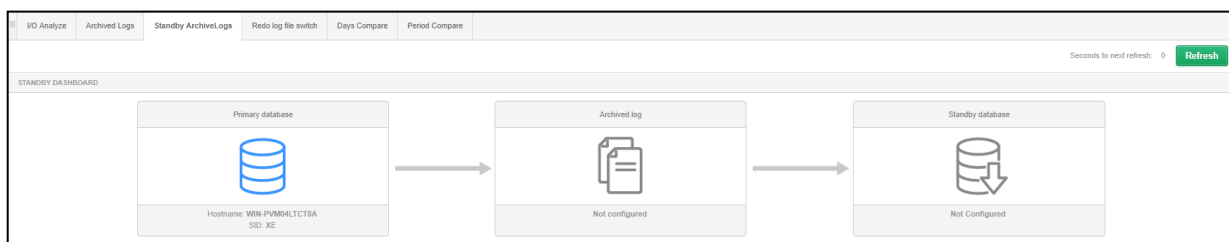
6.2.4.3 Standby Archivelogs Tab

The page contains the state verification functionality of the Standby database synchronization status. Management of standby databases consists of check whether a given database has a dedicated standby database/databases.

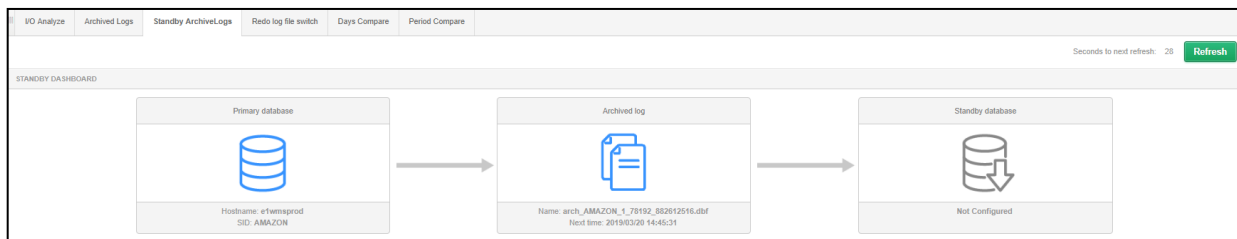
The tab is divided into three parts, which show from the left:

- The database icon with the host name and database SID,
- The name of the last archivedlog file (if the database is archived)
- A list of Standby databases where databases are refreshed based on ArchivedLog files.

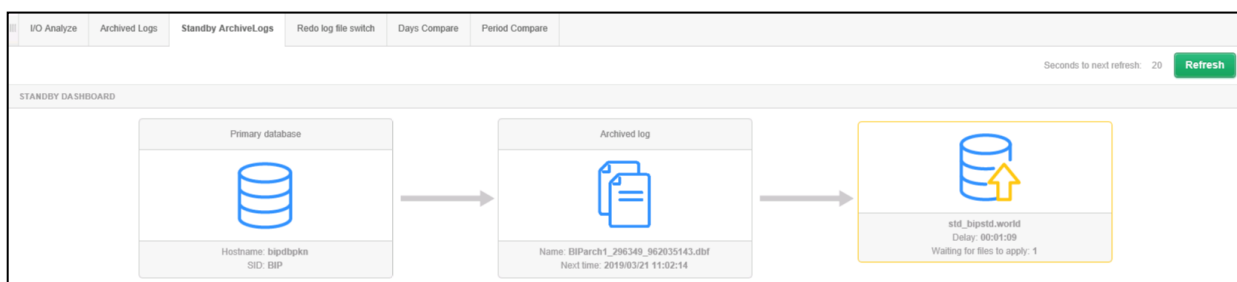
View when archive is not set in the monitored main database:



View when archive is set but there is no Standby base configured:



A view when the configuration of the Standby database is detected:



Under the Standby icon there is information about:

- Standby database name,
- Delays in the applied files between databases,
- The number of files transported to the Standby database, waiting to be loaded (APPLIED = NO) from the production database to the Standby database.

When archivedlog files are sent to more Standby databases, an icon with additional information will be presented for each database.

Information about Standby databases are read from the dictionary view each time after enter the tab. Then, the archiving status is verified based on the archivedlog files.

The pictograms that present the archivedlog file and the standby database are clickable. Click the ArchivedLog file icon displays a view that show the basic information about archived logs for the indicated period of time (by default the current day) in a given database.

SHOW ARCHIVED LOGS DATA										
Date from: <input type="text" value="2019/03/20"/>		to: <input type="text" value="2019/03/20"/>								<input type="button" value="Refresh"/>
Standby name	First scn	Next scn	First time	Next time	Completion time	Archived	Deleted	Status	Applied	
/amazon/arch/a	134797075058	134797075274	2019-03-19 23:40:00	2019-03-20 00:00:00	2019-03-20 00:00:00	YES	NO	Available	NO	
/amazon/arch/a	134797075274	134797075426	2019-03-20 00:00:00	2019-03-20 01:00:00	2019-03-20 01:00:00	YES	NO	Available	NO	
/amazon/arch/a	134797075426	134797075509	2019-03-20 01:00:00	2019-03-20 01:00:00	2019-03-20 01:00:00	YES	NO	Available	NO	
/amazon/arch/a	134797075509	134797075647	2019-03-20 01:00:00	2019-03-20 01:00:00	2019-03-20 01:00:00	YES	NO	Available	NO	
/amazon/arch/a	134797075647	134797075839	2019-03-20 01:00:00	2019-03-20 02:00:00	2019-03-20 02:00:00	YES	NO	Available	NO	
/amazon/arch/a	134797075839	134797076007	2019-03-20 02:00:00	2019-03-20 03:00:00	2019-03-20 03:00:00	YES	NO	Available	NO	
/amazon/arch/a	134797076007	134797076259	2019-03-20 03:00:00	2019-03-20 04:00:00	2019-03-20 04:00:00	YES	NO	Available	NO	
/amazon/arch/a	134797076259	134797076485	2019-03-20 04:00:00	2019-03-20 04:00:00	2019-03-20 04:00:00	YES	NO	Available	NO	
/amazon/arch/a	134797076485	134797076671	2019-03-20 04:00:00	2019-03-20 05:00:00	2019-03-20 05:00:00	YES	NO	Available	NO	
/amazon/arch/a	134797076671	134797077018	2019-03-20 05:00:00	2019-03-20 06:00:00	2019-03-20 06:00:00	YES	NO	Available	NO	

Click on the dedicated Standby database icon displays the dedicated information about transfer status archivedlog file for the indicated period of time (by default the current day) in the Standby database. Information available on the Standby database is below:

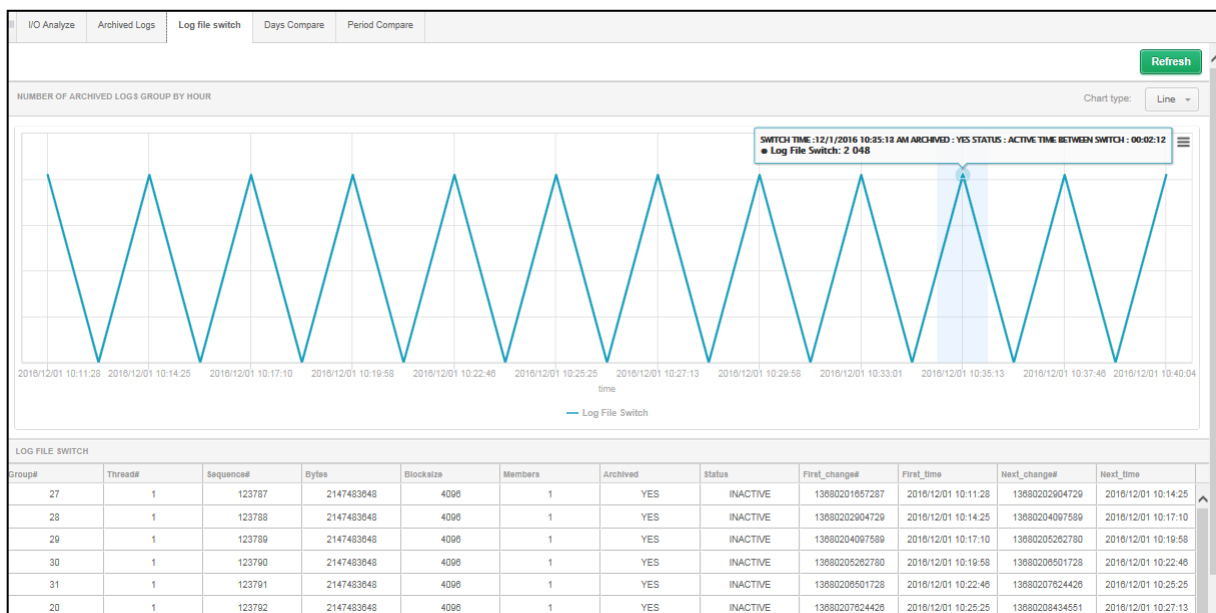
SHOW LOGS FOR STANDBY DATABASE: STD_BIPSTD.WORLD

Date from: 2019/03/21 to: 2019/03/21 Refresh

Standby name	First scn	Next scn	First time	Next time	Completion time	Archived	Deleted	Status	Applied
std_bipstd.w...	80016923822	80016942114	2019:03:21...	2019:03:21...	2019:03:21...	YES	NO	Available	NO
std_bipstd.w...	80016897729	80016923822	2019:03:21...	2019:03:21...	2019:03:21...	YES	NO	Available	YES
std_bipstd.w...	80016892638	80016897729	2019:03:21...	2019:03:21...	2019:03:21...	YES	NO	Available	YES
std_bipstd.w...	80016883074	80016892638	2019:03:21...	2019:03:21...	2019:03:21...	YES	NO	Available	YES
std_bipstd.w...	80016857699	80016883074	2019:03:21...	2019:03:21...	2019:03:21...	YES	NO	Available	YES
std_bipstd.w...	80016825775	80016857699	2019:03:21...	2019:03:21...	2019:03:21...	YES	NO	Available	YES
std_bipstd.w...	80016799069	80016825775	2019:03:21...	2019:03:21...	2019:03:21...	YES	NO	Available	YES
std_bipstd.w...	80016766678	80016799069	2019:03:21...	2019:03:21...	2019:03:21...	YES	NO	Available	YES
std_bipstd.w...	80016736574	80016766678	2019:03:21...	2019:03:21...	2019:03:21...	YES	NO	Available	YES
std_bipstd.w...	80016713200	80016736574	2019:03:21...	2019:03:21...	2019:03:21...	YES	NO	Available	YES

6.2.4.4 Redo Log file switch Tab

On the page presents in a graphical form switching of redo files.



The X axis represents the time in which the files were switched. The shorter the time between the "dots" in the chart, the faster it was to switch the redo files.

The Y-axis shows the file size in MB

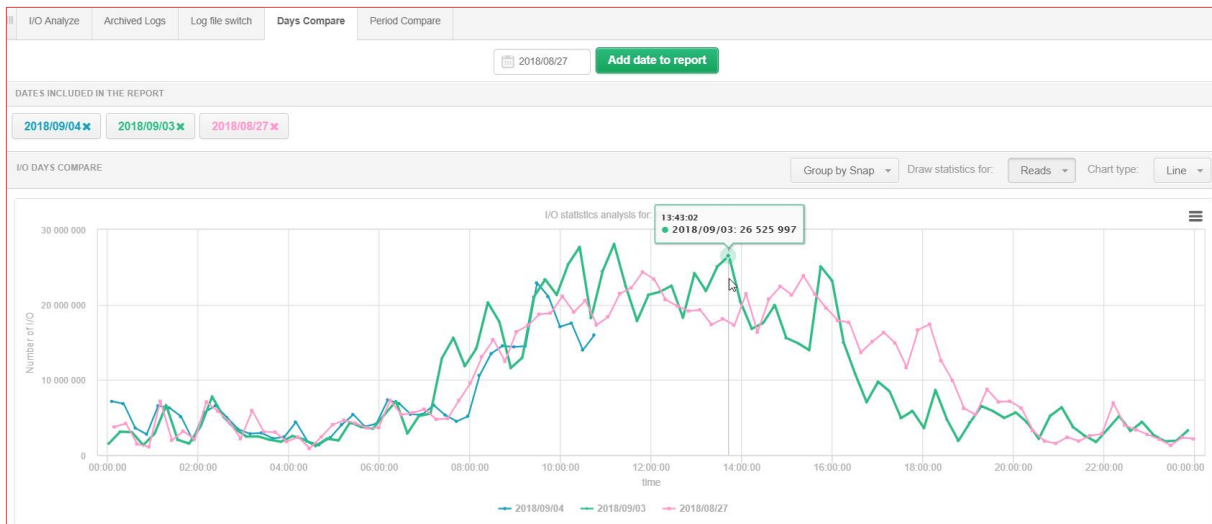
6.2.4.5 „Days Compare/Period Compare” Tab

On the website a possibility of a comparative analysis for I / O statistics is available. The comparison is available for statistics such as:

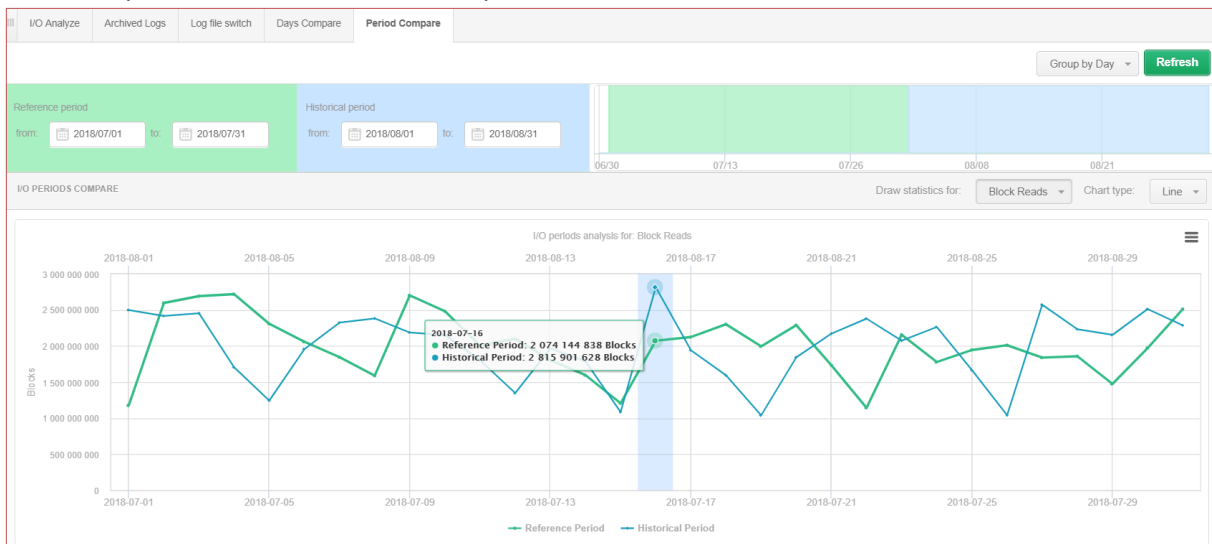
- Reads - number of readings,
- Writes - number of data records by DBWR,
- Block Reads - number of block reads,

- Block Writes - number of block writes,
- Read Time [Seconds] - block read time,
- Write Time [Seconds] - block write time,
- Block Read Time [Seconds] - reading time of a single block,
- Block Write Time [Seconds] - writing time of a single block.

Days Compare page consists of two options to compare data: grouped for the whole day or by snap. Comparing consists of adding specific days from the calendar and adding them to the graph.



The Period Compare page provides opportunity to compare the same statistics grouped for the whole day and in hourly collections. To generate a comparison, a selection of reference period and the historical period should be picked, to which calendar comparison will be made for.



6.2.5 Menu Space Monitor - Database Analysis

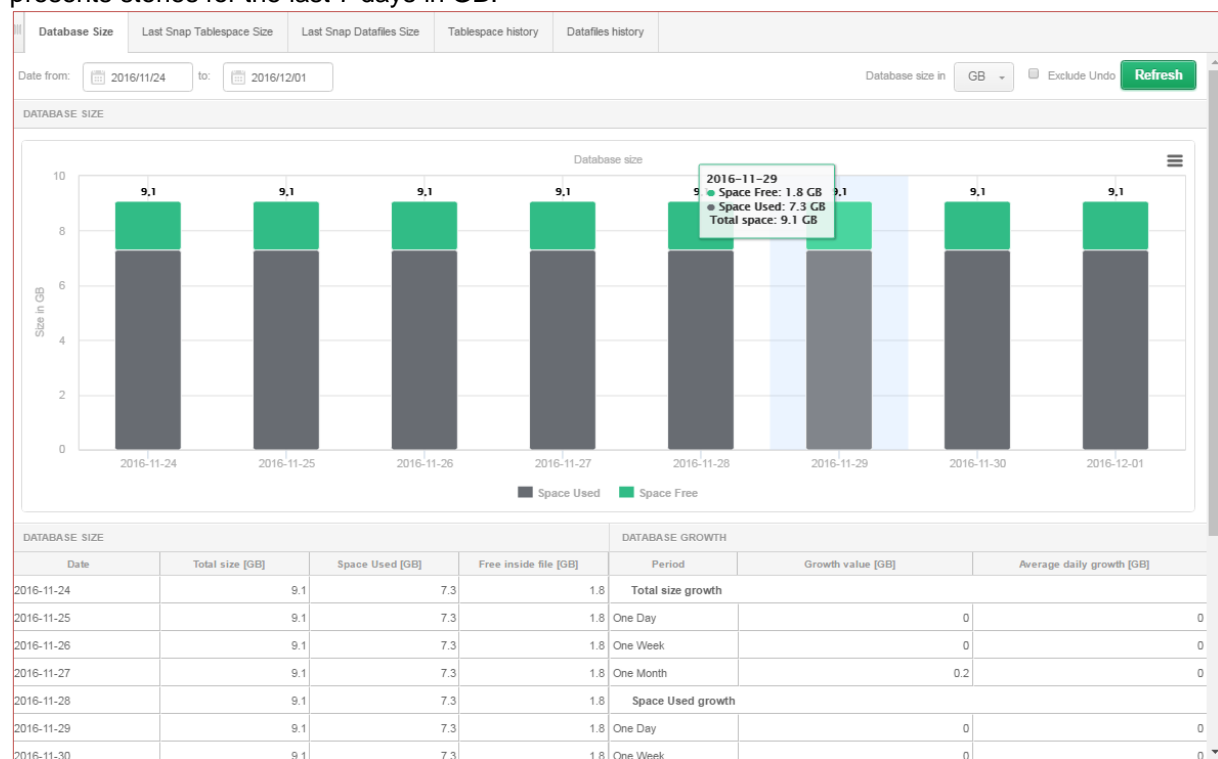
The **Space Monitor** module allows for database storage analysis. Three options are provided which are:

- Display the current size of database
- Detailed information on database occupancy (by database objects)
- The history of database size change in table and graphical form

IMPORTANT: [Space Monitor] module is also accessible from the main page (click [Back to dashboard] button) this allows to analyze the space used by all databases.

6.2.5.1 Database Size Tab

The Database Size tab shows the current size of the database and its size over time. By default, the system presents stories for the last 7 days in GB.



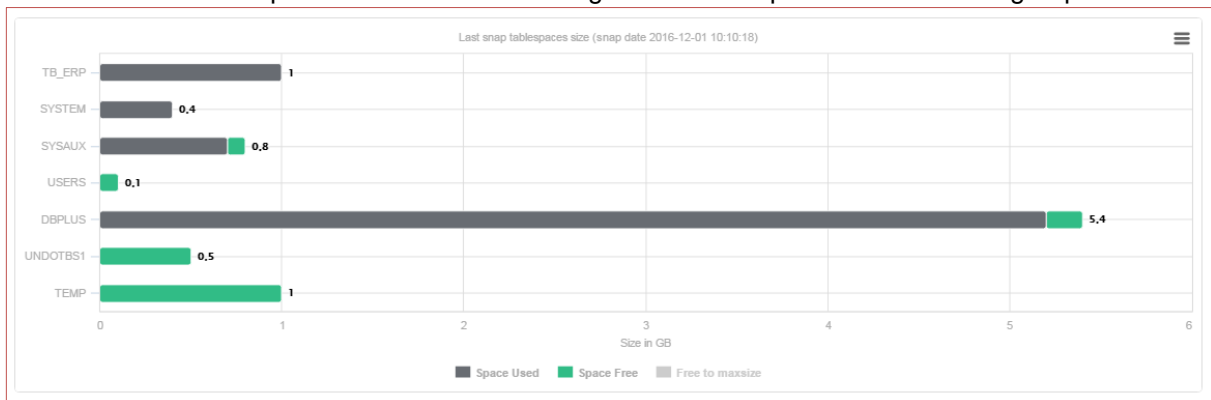
Tables below the graph represent:

- Database size increase history is broken down into:
 - Total Size,
 - Space Used,
 - Free inside file
 - Statistics on base size increase for the last day, week, month

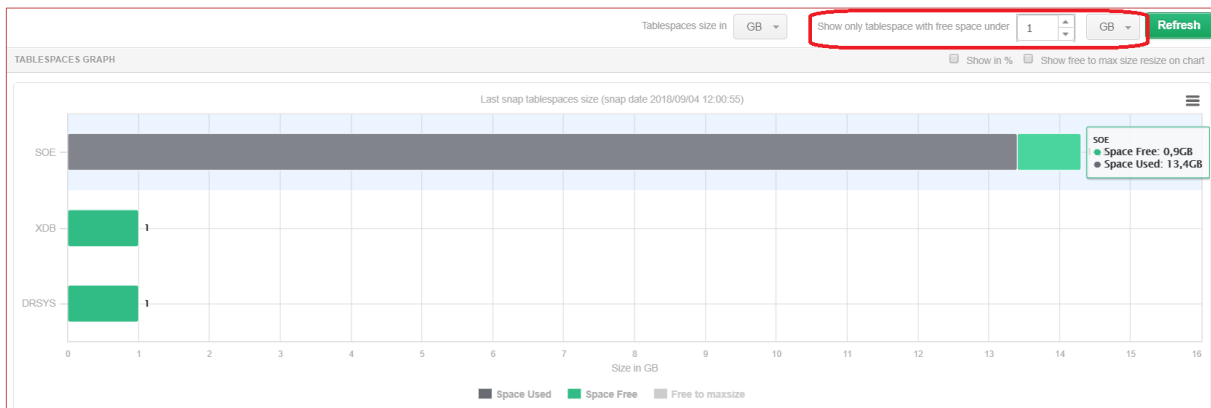
DATABASE SIZE				DATABASE GROWTH		
Date	Total size	Space Used	Free inside file	Period	Growth value [GB]	Average daily growth [GB]
2018-08-28	16 726.8	14 598.2	2 128.6	Total size growth		
2018-08-29	16 726.8	14 620.5	2 106.2	One Day	0	0
2018-08-30	16 726.8	14 703.9	2 022.9	One Week	5.0	0.7
2018-08-31	16 726.8	14 742.5	1 984.3	One Month	165.0	5.3
2018-09-01	16 730.8	14 699.2	2 031.6	Space Used growth		
2018-09-02	16 731.8	14 719.5	2 012.3	One Day	-17.0	-17.0
2018-09-03	16 731.8	14 651.0	2 080.8	One Week	35.8	5.1
2018-09-04	16 731.8	14 633.9	2 097.8	One Month	362.7	11.7

6.2.5.2 Last Snap Tablespace Size” And Last Snap Datafiles Size Tabs

In the Last Snap Tablespace Size screens, Last Snap Datafiles Size system presents the size of the database divided into spaces or data files according to the last snapshot from monitoring a specific database.



In the case of the Last Snap Tablespace Size tab, option to filter the result by the amount of free space is available. For this purpose, the field below needs to be completed, e.g. the value of 1 GB and a result shows the space for which amount of free space does not exceed 1 GB.



Database available for each space:

- Tablespace name
- Total size
- Space Used
- Free inside file
- Free [%] – free space [%],
- Free to max file resize
- Daily Growth
- Weekly Growth

- Monthly Growth

Tablespace name	Total size	Space Used	Free inside file	Free [%]	Free to max file resize	Daily Growth	Weekly Growth	Monthly Growth
SYSTEM	20.0	15.2	4.8	23.8	12.0	0.0	0.1	0.4
TEMPORARY_DATA	180.0	21.7	158.3	87.9	0	1.3	1.7	13.6
SA_TEMP	2 728.0	2 524.4	203.6	7.5	30 040.0	-104.1	12.8	89.3

Information returned for data files:

- File Name – the name of the data file,
- Tablespace name - name of the space
- Maxbytes – maximum available space,
- Free to max file resize - information about the value to which the space can expand,
- Total space – total file space,
- Free inside file – free / unused space,
- Free [%] – percentage of free space
- Autoextend – information on the possibility of extending the space.

File name	Tablespace name	Maxbytes	Free to max file resize	Total space	Free inside file	Free [%]	Autoextend
/cars/data/system01.dbf	SYSTEM	32.0	12.0	20.0	4.8	23.8	YES
TEMPORARY_DATA (All files)	TEMPORARY_DATA	180.0	0	180.0	163.8	91.0	NO
/cars/data/sa_temp.dbf	SA_TEMP	32 768.0	30 040.0	2 728.0	203.6	7.5	YES

IMPORTANT: in the column [Free to max file resize], the data presents information about the parameter set on the database for a given space. they do not verify if space is available on the disk array.

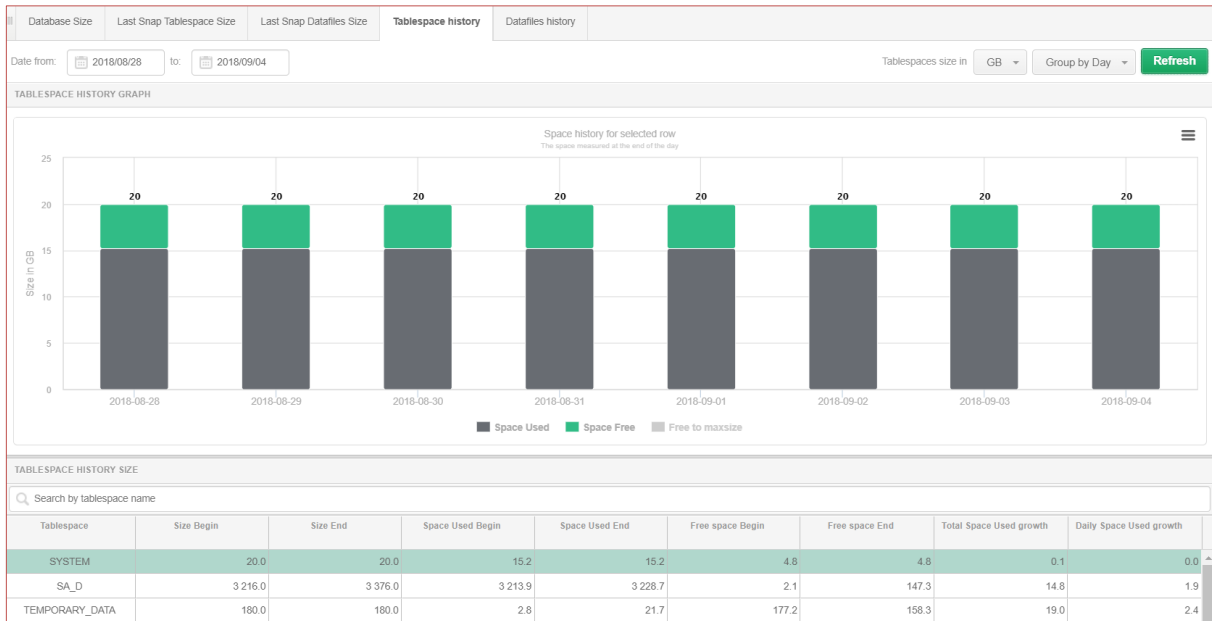
For both tabs below the graph, a search field that allows for quickly information searching about the selected space or data file.

LAST SNAP TABLESPACE SIZE (SNAP DATE 2016-12-01 10:10:18)									
Search by tablespace name									
Tablespace name	Total size [GB]	Space Used [GB]	Free inside file [GB]	Free [%]	Free to max file resize	Daily Growth [GB]	Weekly Growth [GB]	Monthly Growth [GB]	
TB_ERP	1	1	0	0	0	0	0	0	
SYSTEM	0.4	0.4	0	1.42	0.2	0	0	0	
SYS_AUX	0.8	0.7	0.1	5.08	31.2	0	0	0	
USERS	0.1	0	0.1	97.44	10.9	0	0	0	
DBPLUS	5.4	5.2	0.2	4.82	26.6	0	0	0.2	

6.2.5.3 Tablespace History And Datafiles History Tabs

Both tabs show the size in GB for the selected table space or data file for each day of the selected date range.

Graphs can be presented for a given period, grouped by day or time.



The chart area is refreshed after clicking on the indicated lines from the table below.

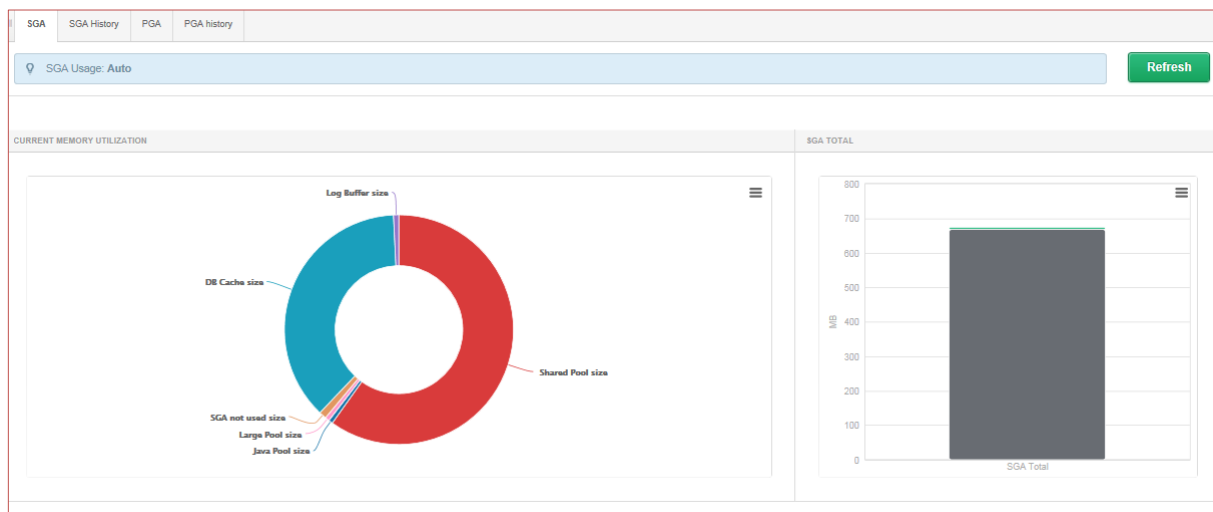
6.2.6 Menu Memory - Database Analysis

The Memory module allows to analyze the memory utilization in the database. The tabs provide options to:

- Display the current memory usage,
- Memory usage history over time,
- The use of memory by user sessions.

6.2.6.1 SGA Tab

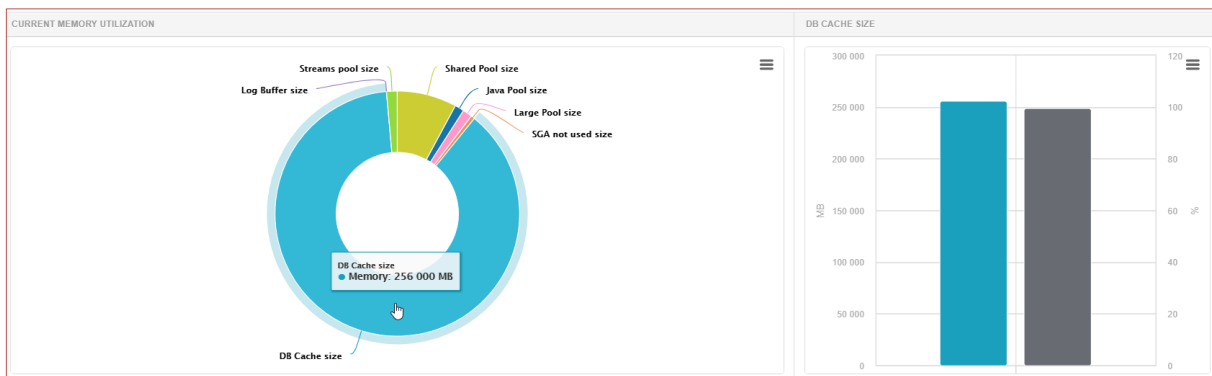
The "SGA" tab presents the settings of individual Oracle database parameters included in the SGA database. Parameter settings are shown as components of the wheel. The sizes of individual parameters are expressed in MB. Clicking on the interesting fragment of the wheel showing the given parameter will show the statistics of this parameter using the graph on the right side of the SGA wheel. The statistics graph has a different characteristic for each parameter.



The "SGA" window is divided into two parts:

- A pie chart that presents proportionally:
 - Shared Pool Size,
 - Streams pool size,
 - Shared Pool size,
 - Java Pool Size,
 - Large Pool size,
 - SGA Not Used Size,
 - DB Cache Size,
 - InMemory (for database versions 12 and higher),
- Bar chart showing individual components where:
 - The color used for the selected area is the amount of buffer used
 - The amount of the free part of the buffer is marked in green

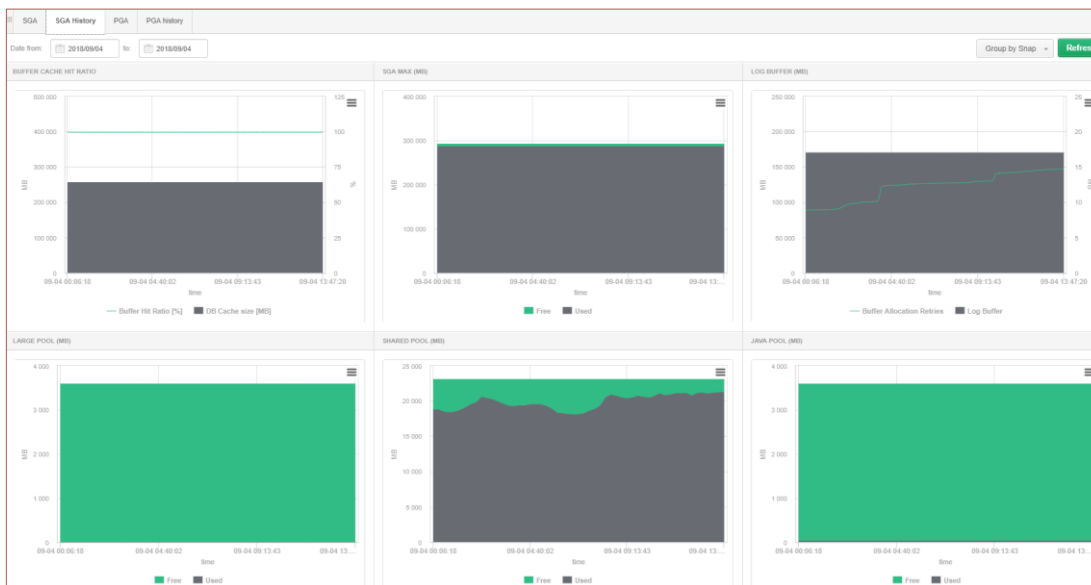
The exception is "DB Cache" where the bar graph shows the percentage of hits in this buffer as a percentage.



6.2.6.2 SGA History Tab

The "SGA History" tab is divided into six parts and shows the values of individual buffers. It shows their utilization in the given time. Using the "Group by" pull-down menu, the graph shows the data grouped according to the choice:

- Snap – divided into periods of 15 minutes,
- Hour – divided into one-hour periods,
- Day - divided into one-day periods.



Buffer Cache Hit Ratio

The graph shows the coefficient of hits in the DB_CACHE_SIZE buffer as well as the size of the DB_CACHE_SIZE buffer for the period selected in the Group function. The Y-axis on the right side of the graph shows the hit coefficient in the DB_CACHE_SIZE buffer. The Y-axis on the left side of the graph shows the size expressed in the MB buffer DB_CACHE_SIZE. The X-axis defines the time for which the data on the graph is shown.



SGA MAX

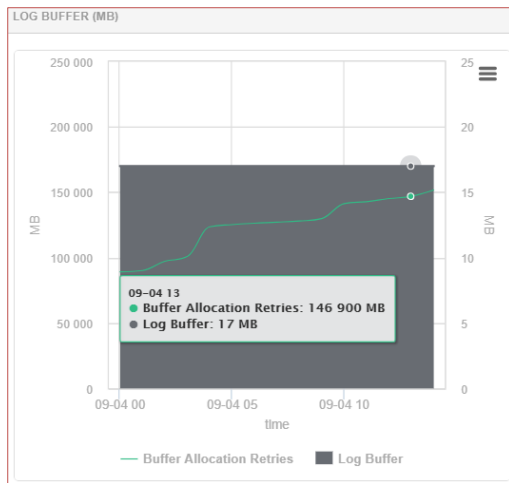
The graph shows the value of the parameter "sga_max_size" in the for selected time. The Y axis represents the value expressed in MB. The X-axis defines the time for which the data on the graph is shown. with the accuracy selected by the administrator: day, hour, snap (every 15 minutes).



Log Buffer (MB)

The graph shows the size of the log_buffer buffer as well as the largest number of attempts to write to the buffer due to its occupancy.

The Y-axis on the left side of the graph shows the statistics. Buffer Allocation Retries shows how many times the user's process waited for the space in the REDO buffer. The Y axis on the right side of the graph shows the size of log_buffer. The X-axis defines the time for which the data on the graph is shown.

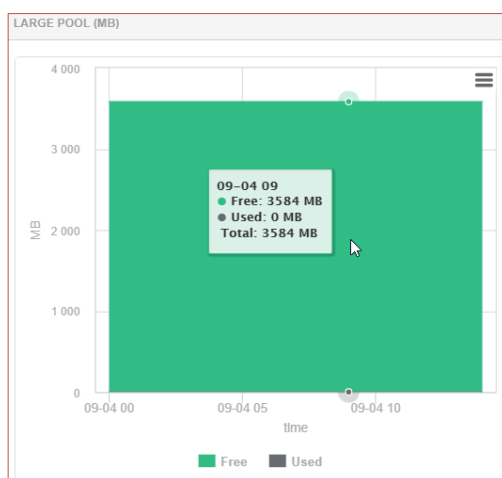


Large Pool

The graph shows the utilization of the LARGE POOL buffer expressed in MB.

The Y-axis of the graph shows the utilization value expressed in MB. The X-axis of the graph shows the time in which the given disposal took place with the accuracy chosen by the user, ie: day, hour, snap (every 15 minutes).

The gray color represents the occupied space in the LARGE POOL buffer. The green color represents the free space in the LARGE POOL buffer.

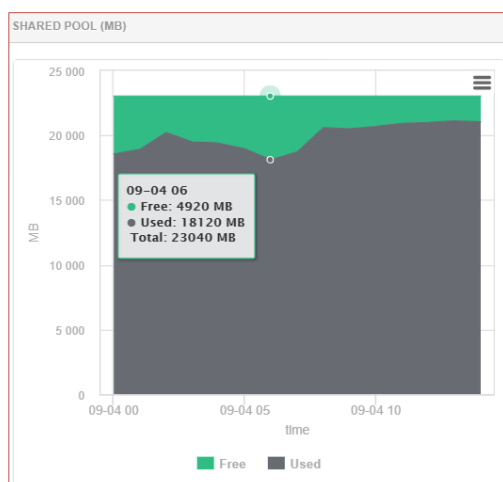


Shared Pool

The graph shows the utilization of the SHARED POOL buffer expressed in MB.

The Y axis represents the utilization value expressed in MB. The X-axis of the graph shows the time in which the given disposal took place with the accuracy selected by the administrator, i.e.: day, hour, snap (every 15 minutes).

The gray color represents the occupied space in the SHARED POOL buffer. The green color represents the free space in the SHARED POOL buffer.



Java Pool

The graph shows the utilization of the JAVA POOL buffer expressed in MB.

The Y-axis of the graph shows the utilization value expressed in MB. The X-axis of the graph shows the time in which the given disposal took place with the accuracy selected by the administrator, i.e.: day, hour, snap (every 15 minutes).

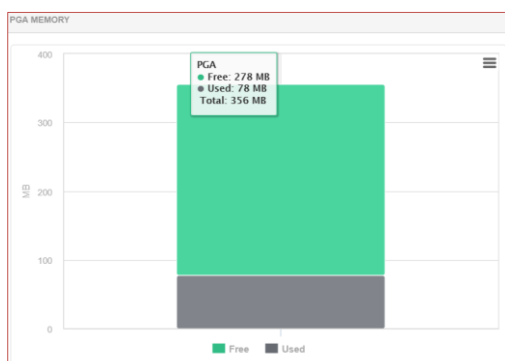
The gray color represents the occupied space in the JAVA POOL buffer. The green color represents the free space in the JAVA buffer.



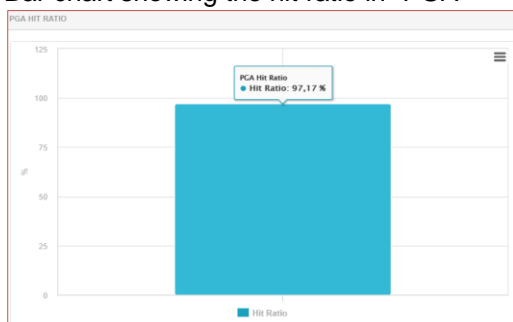
6.2.6.3 PGA Tab

The information on the "PGA" website is divided into three areas:

- Bar chart showing the amount of memory of "PGA"



- Bar chart showing the hit ratio in "PGA"



- The table shows the utilization of the PGA buffer for each session, where:
 - Session Sid – user's session ID,
 - Memory Usage [kB] – number of occupied bytes in the buffer,
 - OS user name – the username of the operating system logged in to the Oracle database,
 - Username – Oracle database user name,
 - Machine – the name of the machine from which you logged into the Oracle database
 - Status – user session status [ACTIVE, INACTIVE, KILLED]
 - Program – the name of the program you logged in to the Oracle database

PGA usage by users session						
Session Sid	Memory Usage [kB]	OS user name	Username	Machine	Status	Program
1	508.8 kB	SYSTEM		WIN-PVM04LTCT8A	ACTIVE	ORACLE.EXE (DIAG)
2	1 229.4 kB	SYSTEM		WIN-PVM04LTCT8A	ACTIVE	ORACLE.EXE (CKPT)
3	564.6 kB	SYSTEM		WIN-PVM04LTCT8A	ACTIVE	ORACLE.EXE (QMNC)
4	1 340.8 kB	SYSTEM		WIN-PVM04LTCT8A	ACTIVE	ORACLE.EXE (CJQ0)
7	1 012.6 kB	SYSTEM	DBMON2	ADDBPLUSWIN-PVM04LTCT8A	INACTIVE	DBPLUS.Oracle.CatcherService.e
31	764.8 kB	SYSTEM		WIN-PVM04LTCT8A	ACTIVE	ORACLE.EXE (DBRM)

6.2.6.4 PGA History Tab

The information on the "PGA History" tab is divided into two areas:

- A graph showing the utilization of the PGA buffer [MB]

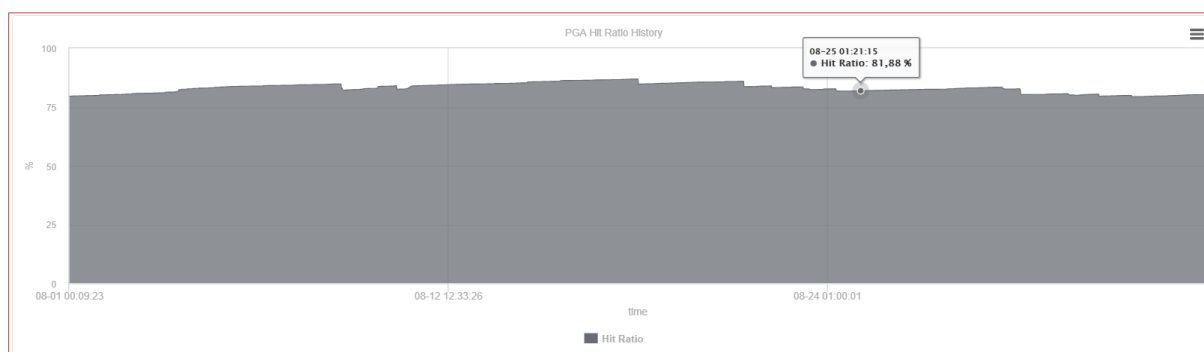
The Y-axis of the graph shows the utilization value [MB]. The X-axis of the graph shows the time at which the utilization was selected by the administrator, ie: day, hour, snap (every 15 minutes).

The gray color represents the occupied space in the PGA buffer. The green color represents the free space in the PGA buffer.



- The graph shows the hit ratio in the PGA buffer [%]

The Y-axis of the graph shows the value in% of hits in the buffer by Oracle database sessions. The X-axis of the graph shows the time with the accuracy selected by the administrator, i.e.: day, hour, snap (every 15 minutes).



In the 'Group by' drop-down menu, the corresponding charts show the data grouped according to a choice:

- Snap – divided into 15 minute periods
- Hour – divided into one-hour periods
- Day - divided into one-day periods

6.2.7 Menu Sessions – Database Analysis

Sessions functionality presents information about sessions in the database. From the level of upper tabs, access is provided to:

- **Sessions** – sessions in the 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 (eg a session that performs a query that sorts a large amount of data),
- **Undo usage session** - functionality that allows for session analysis for the use of UNDO (eg sessions holding a large portion of data in an uncommitted transaction)
- **Session history** – historical information about the session,
- **Session/Sort/Undo history** – Field allowing for following searches:
 - What queries the program / user runs
 - Which users the specified query hash are run

6.2.7.1 Sessions Tab

In the Session tab, information about sessions is displayed. This data can be filtered appropriately:

- only active sessions (status ACTIVE),
- sessions performed by logged in users
- performed by the user with the given name.

Additional filtering, allows for filtering the list presented by the type of wait that affects performance. The available list has been limited by DBPLUS analysts that have impact on performance.

The screenshot shows the DBPLUS Sessions Tab interface. At the top, there are filter options: 'Active sessions' (checked), 'Users only', 'Min elapsed time' (0 sec), 'Sid', 'Username', and 'upper case' (checked). A 'Refresh' button is on the right. Below these is a 'Hide additional filters' link. The main area is divided into two sections: 'Performance Waits' and 'Waits selected for filtering'. The 'Performance Waits' section has a search box and a list of wait types: ADR block file read, ADR block file write, ADR file lock, ARCH wait for archivelog lock, ASM file metadata operation, and buffer busy waits. The 'Waits selected for filtering' section is currently empty.

After selecting the appropriate filters, session information will be presented:

- Logon Time – time of user logging into the Oracle database,
- Sid – user session ID,
- Serial - user session ID
- Hash value – identifier of the currently executed command (it means that the command is currently being executed with accuracy provided by the Oracle database),
- User name – Oracle's database user name,
- Status – status of the session [ACTIVE, INACTIVE, KILLED],
- Elapsed Time [Seconds] – duration of the query or PL / SQL block in seconds,
- Schema – the name of the schema on which the query is performed,
- OS user - user name in the operating system on which the Oracle database was logged in
- Process (server) process number in the operating system that supports the session,
- Process (client) – process number on the user's side,
- Machine – the name of the machine the Oracle database was logged in,
- Program – the name of the program you logged in to the Oracle database,
- Wait – name of the session wait type,
- Blocking session – the number of the parent session that blocked the current session (when the value is greater than zero),
- ClientInfo (presented only value is not null).

Sessions																
Sort usage sessions		Undo usage sessions		Sessions history		Session / Undo history										
<input checked="" type="checkbox"/> Active sessions <input checked="" type="checkbox"/> Users only Min elapsed time: <input type="text" value="0"/> sec. Sid: <input type="text"/> Username: <input type="text"/> <input checked="" type="checkbox"/> upper case <input type="button" value="Refresh"/>																
SELECT SESSION (LAST REFRESHED: 11:00:31) <input type="button" value="Kill session"/>																
Logon time	Sid	Serial	Query Hash	Username	Status	Elapsed Time	Schema	OS user	Process (server)	Process (client)	Machine	Program	Module	Blocking session	Wait	
2016-12-01 09:43:12	52	13757	232111811	SYSTEM	ACTIVE	4 854	SYSTEM	ADDBPLUSIA	26360	22572:22820	ADDBPLUSIW	sqlplus.exe	SQL*Plus	75	enq: TX - row	
2016-12-01 09:43:12	114	59653	232111811	SYSTEM	ACTIVE	4 854	SYSTEM	ADDBPLUSIA	20956	11928:16772	ADDBPLUSIW	sqlplus.exe	SQL*Plus	75	enq: TX - row	
2016-12-01 09:43:12	27	48561	232111811	SYSTEM	ACTIVE	4 854	SYSTEM	ADDBPLUSIA	20488	12952:22956	ADDBPLUSIW	sqlplus.exe	SQL*Plus	75	enq: TX - row	
2016-12-01 09:43:12	50	59321	232111811	SYSTEM	ACTIVE	4 854	SYSTEM	ADDBPLUSIA	7324	23540:15908	ADDBPLUSIW	sqlplus.exe	SQL*Plus	75	enq: TX - row	
2016-12-01 09:43:12	74	64575	232111811	SYSTEM	ACTIVE	4 854	SYSTEM	ADDBPLUSIA	8908	11884:16704	ADDBPLUSIW	sqlplus.exe	SQL*Plus	75	enq: TX - row	
2016-12-01 09:43:12	120	5007	232111811	SYSTEM	ACTIVE	4 854	SYSTEM	ADDBPLUSIA	13272	20112:11664	ADDBPLUSIW	sqlplus.exe	SQL*Plus	75	enq: TX - row	
2016-12-01 11:04:06	137	10239	3994527576	DBPLUS	ACTIVE	0	DBPLUS	USLUGA LOK	11468	21528:2	ZARZADZANI	w3wp.exe			SQL*Net mes	

Selected rows provide following information below:

- SQL Sub-tab

It contains the text of the SQL query. The information is displayed only for queries for which Hash value is stored.

- Operation Progress Sub-tab

Presents information about the status of the current operation being performed by the session in the Oracle database.

SQL	Operation progress	Statistics	Session Waits
Progress	Operation name	Target	Time remaining
1307329 of 1576332 (Blocks)	Table Scan	REDANCOM.KH_FIR	79
			383
			Table Scan: REDANCOM.KH_FIR

- Statistics Sub-tab

Displays information about the statistics of the selected session.

SQL	Operation progress	Statistics	Session Waits
Name	Value		
Requests to/from client	195		
logons cumulative	1		
logons current	1		
opened cursors cumulative	1128		
opened cursors current	3		
user commits	56		
user calls	356		
recursive calls	1956		

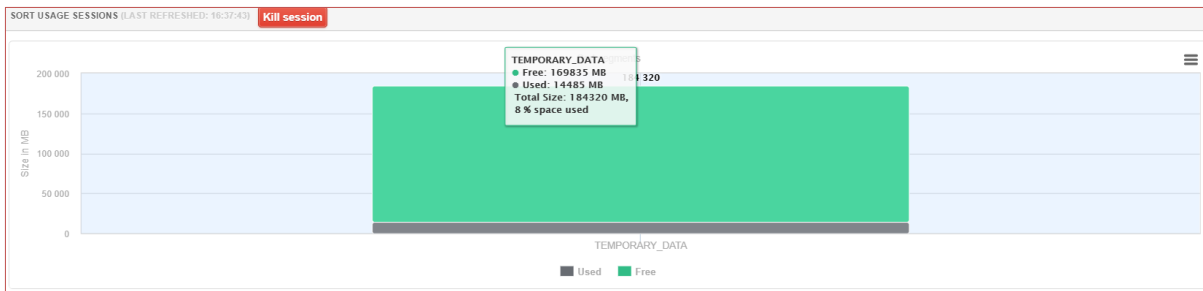
- Session Waits Sub-tab

Shows detailed information about the given wait for the selected query.

SELECT SESSION (LAST REFRESHED: 11:00:31) <input type="button" value="Kill session"/>																
Logon time	Sid	Serial	Query Hash	Username	Status	Elapsed Time	Schema	OS user	Process (server)	Process (client)	Machine	Program	Module	Blocking session	Wait	
2016-12-01 09:43:12	52	13757	232111811	SYSTEM	ACTIVE	4 854	SYSTEM	ADDBPLUSIA	26360	22572:22820	ADDBPLUSIW	sqlplus.exe	SQL*Plus	75	enq: TX - row	
2016-12-01 09:43:12	114	59653	232111811	SYSTEM	ACTIVE	4 854	SYSTEM	ADDBPLUSIA	20956	11928:16772	ADDBPLUSIW	sqlplus.exe	SQL*Plus	75	enq: TX - row	
2016-12-01 09:43:12	27	48561	232111811	SYSTEM	ACTIVE	4 854	SYSTEM	ADDBPLUSIA	20488	12952:22956	ADDBPLUSIW	sqlplus.exe	SQL*Plus	75	enq: TX - row	
2016-12-01 09:43:12	50	59321	232111811	SYSTEM	ACTIVE	4 854	SYSTEM	ADDBPLUSIA	7324	23540:15908	ADDBPLUSIW	sqlplus.exe	SQL*Plus	75	enq: TX - row	
SQL	Operation progress	Statistics	Session Waits													
Seq#	Event	P1text	P1	P2text	P2	P3text	P3	Wait time	Seconds in wait	State						
50	enq: TX - row lock co	name mode	1415053318	usn<<16 slot	131078	sequence	59908	0	84	WAITING						

6.2.7.2 Sort usage sessions Tab

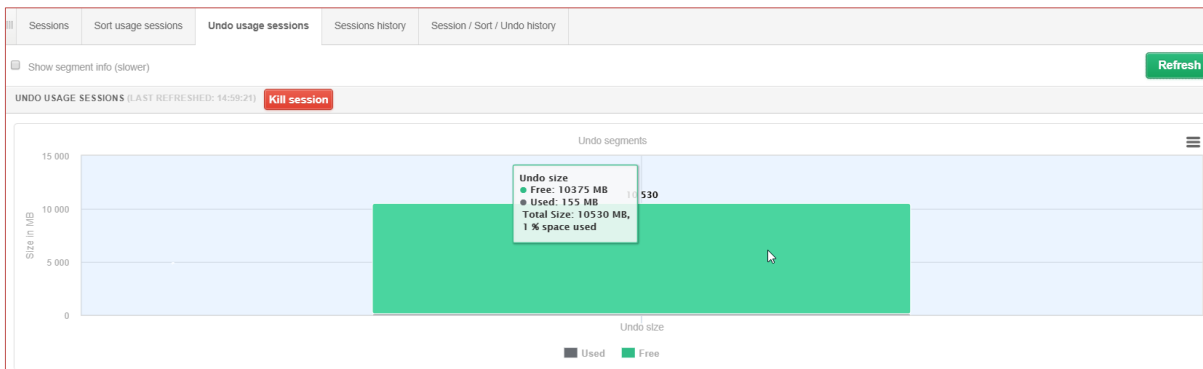
In the next [Sort usage sessions] tab, sessions are presented for the use of temporary space:



Below the chart is a list of sessions with the same functionality as in the Sessions tab - including clicking on the sessions shows in the sub-tabs the content of the query, information about the wait and the state of the session.

6.2.7.3 Undo usage sessions Tab

In the [Undo usage sessions] tab, sessions presented for the use of the UNDO transaction space are located. The functionality allows to track those sessions that take up a large UNDO space (e.g. a large portion of updated data) or long period of time sessions.



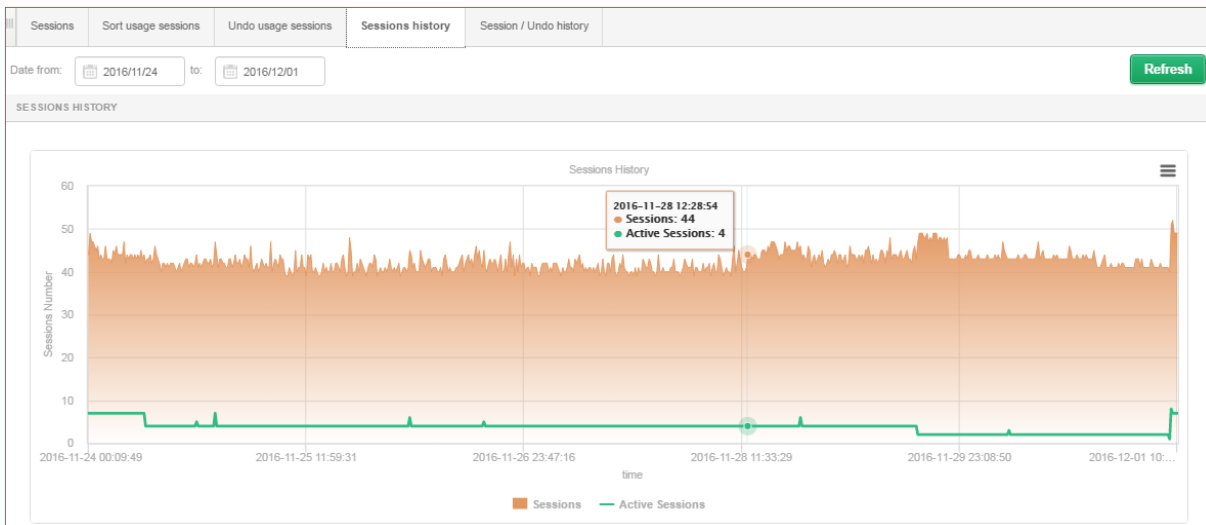
As in the previous tabs (i.e. Sessions, Sort usage sessions), below the chart is a list of sessions.

Additionally, after selecting the checkbox "Show segment Info", information about segment name and size will be displayed for each session.

Logon time	Sid	Serial	Hash Value	Username	Status	Elapsed Time [seconds]	Schema	OS user	Machine	Program	Wait	Used records	Used blocks	Segment size [KB]	Segment name
2018/09/10 14:59:17	17960	24741		ICEDOC	INACTIVE	2	ICEDOC	edoc	edoc-app02n	JDBC Thin Client	SQL*Net message from	16		2 176.0 KB	_SYSSMUJ2_119
2018/09/10 14:23:06	1978	13469		ICEDOC	INACTIVE	2	ICEDOC	edoc	edoc-app02n	JDBC Thin Client	SQL*Net message from	28		3 200.0 KB	_SYSSMUS_125
2018/09/10 14:49:13	3510	21981		ICEDOC	INACTIVE	2	ICEDOC	edoc	edoc-app02n	JDBC Thin Client	SQL*Net message from	76		2 176.0 KB	_SYSSMU10_31

6.2.7.4 Session history Tab

On this page displays in form of a graph the number of active and inactive sessions in the selected time period.



6.2.7.5 Session/ Sort/ Undo history Tab

The session / sort / undo tab shows detailed information about open sessions at a given time:

The data in the table are divided into three groups:

- yellow color presents information about active sessions,
- green color shows information about sessions using sorting,
- red color shows information about sessions using Undo.

The screenshot shows the 'Session / Sort / Undo history' tab. The table has columns: Logdate, Active Sessions, Sessions using Sort, Sort Space Used [MB], Sessions using Undo, Record Count in Undo, and Undo Space Used [MB]. A 'Toggle view' button is highlighted in red. Below the table, there are three tabs: Sessions, Sort, and Undo, with 'Sessions' selected.

Logdate	Active Sessions	Sessions using Sort	Sort Space Used [MB]	Sessions using Undo	Record Count in Undo	Undo Space Used [MB]
2018/09/10 00:00:54	6	4	4 MB	0	0	0
2018/09/10 00:01:55	3	4	4 MB	0	0	0
2018/09/10 00:02:56	5	5	5 MB	0	0	0
2018/09/10 00:03:57	4	4	4 MB	0	0	0
2018/09/10 00:04:58	4	4	4 MB	0	0	0
2018/09/10 00:06:00	5	4	4 MB	0	0	0
2018/09/10 00:07:01	3	4	4 MB	0	0	0
2018/09/10 00:08:02	5	4	4 MB	1	57	0
2018/09/10 00:09:03	4	4	4 MB	0	0	0

Sid	Serial#	Hash Value	User	Active Time [seconds]	Schema	OS User	Machine	Program	Module	Wait	Blocking session
6136	7971	2322825964	SOA_SOAINFRA	2	SOA_SOAINFRA	oracle	osb12.intercars.local	JDBC Thin Client	JDBC Thin Client	Streams AQ: waiting f	0
7889	10515	2322825964	SOA_SOAINFRA	0	SOA_SOAINFRA	oracle	osb12.intercars.local	JDBC Thin Client	JDBC Thin Client	Streams AQ: waiting f	0

Clicking on the table record presents details for the selected snapshot in the Sessions, Sort and Undo tabs. In both cases, following information are presented:

- Sid – user session id along with Serial #,

- Serial# - user session identifier with SID,
- Hash Value – the identifier of the command being executed
- User – Oracle database user name,
- Active Time – duration of the query or PL / SQL block in seconds,
- Schema – the name of the schema where the given SQL statement is executed,
- OS user – user name in the operating system where the Oracle database was logged in,
- Machine – the name of the machine where the Oracle database was logged in,
- Program – the name of the system / program the session was launched,
- Module - the name of the application / module that launched the query,
- Wait – specific type of wait,
- Blocking session – the number of the parent session that blocked the current session (when the value is greater than zero).

In the Session> Session / Sort / Undo history tab, the function to search session history to find blocking sessions has been added. If there were blockades in each snap, the blocking sessions column which contains the identifier of the blocking session is supplemented in such cases. In the latest version, a blocking session search mechanism has been added.

When a blockage occurs, click the "loupe" button that appears in the Blocking sessions column. This will cause the table to be automatically scrolled and the row with the session which is the blocker will be highlighted.

Sid	Serial#	Hash Value	User	Active Time [seconds]	Schema	OS User	Machine	Program	Module	Wait	Blocking session
35345	4	2104331062	OFA2BI	61	OFA2BI	oracle	e1orabi	oracle@e1orabi (TNS ...	oracle@e1orabi (TNS ...	direct path read temp	
14383	207	1955834744	HZAJOBGLK_INTER	92	INTER	oracle	e3kronos	oracle@e3kronos (J016)	wf_jobs.generuLK	library cache: mutex X	14383
14947	183	1496488952	HZAJOBGLK_INTER	121	INTER	oracle	e3kronos	oracle@e3kronos (J022)	wf_jobs.generuLK	enq: TX - row lock con	
14477	289	2088278979	SYNCRON	124	SYNCRON	oracle	e3kronos	oracle@e3kronos (J023)		db file sequential read	
13067	445	1780980331	SYS	128	SYS	oracle	e3kronos	oracle@e3kronos (J002)	DBMS_SCHEDULER	Streams AQ: waiting f...	
14665	43	4289129004	ALUCZYNS	128	ALUCZYNS	oracle	e3kronos	oracle@e3kronos (J019)		db file sequential read	

The application has ability to search information about the user's session using a given type of validity. We start the search by pressing the "Hide additional filters" button and then from the list of available waits we add the ones we want to view.

After pressing the Refresh button, only those sessions that were waiting, for a wait selected by the user from the list will be presented in the given period. At the same time, you can also select other filters, e.g. such as SID session ID or Hash Value.

Sessions | Sort usage sessions | Undo usage sessions | Sessions history | Session / Sort / Undo history

From: to: Using Hash Value/Sql Id: Username: Sid: Refresh

Hide additional filters

Performance Waits

Search by name ...

- cursor: pin S
- cursor: pin S wait on X
- cursor: pin X
- db file async I/O submit
- db file parallel write
- db file scattered read

Waits selected to filtering

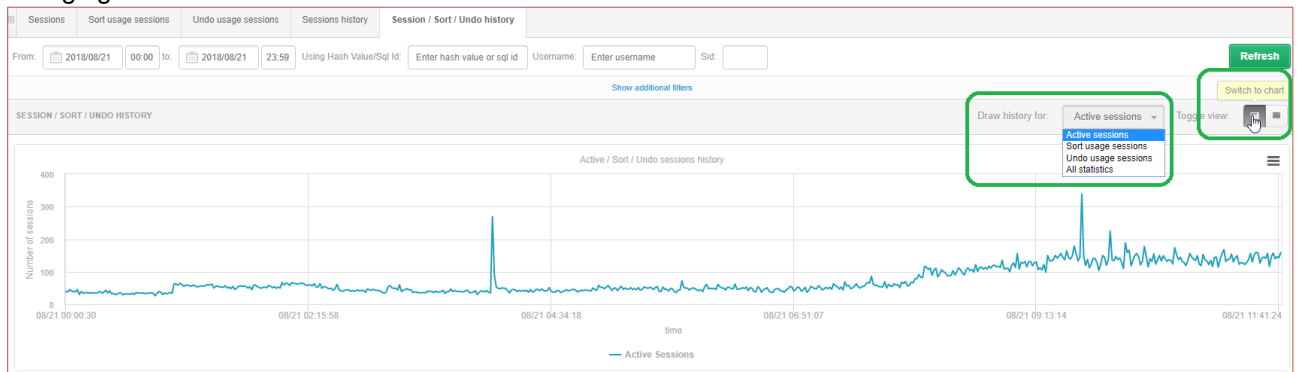
- buffer busy waits
- buffer deadlock
- control file heartbeat
- control file sequential read
- db file parallel read

Machine:

Module:

Logdate	Sid	Serial#	Hash Value	User	Active Time [seconds]	Schema	OS User	Machine	Program	Module	Wait	Blocking session
2018-11-23 04:59:15	21410	47505	234068006	FKRYNICK_INTER	1	FKRYNICK_INTER	rkwie	INTERITS15E	samolot.exe	samolot.exe	buffer busy waits	0
2018-11-23 05:00:20	2830	15843	2104281773	INTER	3	INTER	oracle	u3gaja	oracle@u3gaja (J033)		buffer busy waits	0
2018-11-23 05:00:20	3319	22803	1216626633	CRM	101	CRM	crm	INTERICRMSQL31	sqbsevr.exe	sqbsevr.exe	db file parallel read	0
2018-11-23 05:01:24	1026	48407	2717635498	SYNCRON	0	SYNCRON	mliuka	DEVEL-RUNTIME	dbfexp.exe	dbfexp.exe	buffer busy waits	0
2018-11-23 05:03:31	3319	22803	1216626633	CRM	290	CRM	crm	INTERICRMSQL31	sqbsevr.exe	sqbsevr.exe	db file parallel read	0
2018-11-23 05:04:33	4961	11065	1180465985	DMACHURA_INTER	2	DMACHURA_INTER	dmachur	INTERITS48			buffer busy waits	0
2018-11-23 05:04:33	31941	14713	1586261291	MLATKA_INTER	1	MLATKA_INTER	mlatk-	INTERITS42			buffer busy waits	0

This screen also provides functionality for generating the graph, which can be cycled through by changing the view "Switch to chart".



Four charts are available, these are:

- Active sessions
- Sessions using sorting
- Sessions using the Undo
- A summary graph containing all statistics

After hovering over a point on the graph, depending on the type of the graph, detailed information about the session will be presented.

6.2.7.6 Menu Session – Session Resources

As part of the application, the ability to view statistics on sessions performed on a given SQL instance on an ongoing basis is available. To do this, select menu under the "Sessions" menu "Sessions Resource". The website presents session statistics, downloaded directly from the database.

Sid	Serial	Statistic name	Statistic value	Global value	Hash Value	Username	Status	Elapsed time	Schema	Os user	Machine
Statistic: CPU used by this session (3 items)											
10956	61871	CPU used by this sessi	181	297		NAGIOS	INACTIVE	0	NAGIOS	ppasinsk	ICPSPF71
16426	12009	CPU used by this sessi	14	297		ICEDOC	INACTIVE	8	ICEDOC	edoc	edoc-app02n
16864	44065	CPU used by this sessi	29	297		ICEDOC	INACTIVE	8	ICEDOC	edoc	edoc-app02n
Statistic: physical read total bytes (3 items)											
5038	1	physical read total bytes	131072	5857280			ACTIVE	8010106	SYS	oracle	urjajpet
7447	1	physical read total bytes	114688	5857280			ACTIVE	8010100	SYS	oracle	urjajpet
10956	61871	physical read total bytes	5111808	5857280		NAGIOS	INACTIVE	0	NAGIOS	ppasinsk	ICPSPF71
Statistic: user commits (3 items)											
884	16537	user commits	2	109		ICEDOC	INACTIVE	3	ICEDOC	edoc	edoc-app01n

Three basic statistics on the use of the CPU, physical readings or information about the triggered "commit", are present upon accessing the website.

The user can change the list of statistics presented on the website by clicking the [Settings] button and selecting any statistics from the list available.

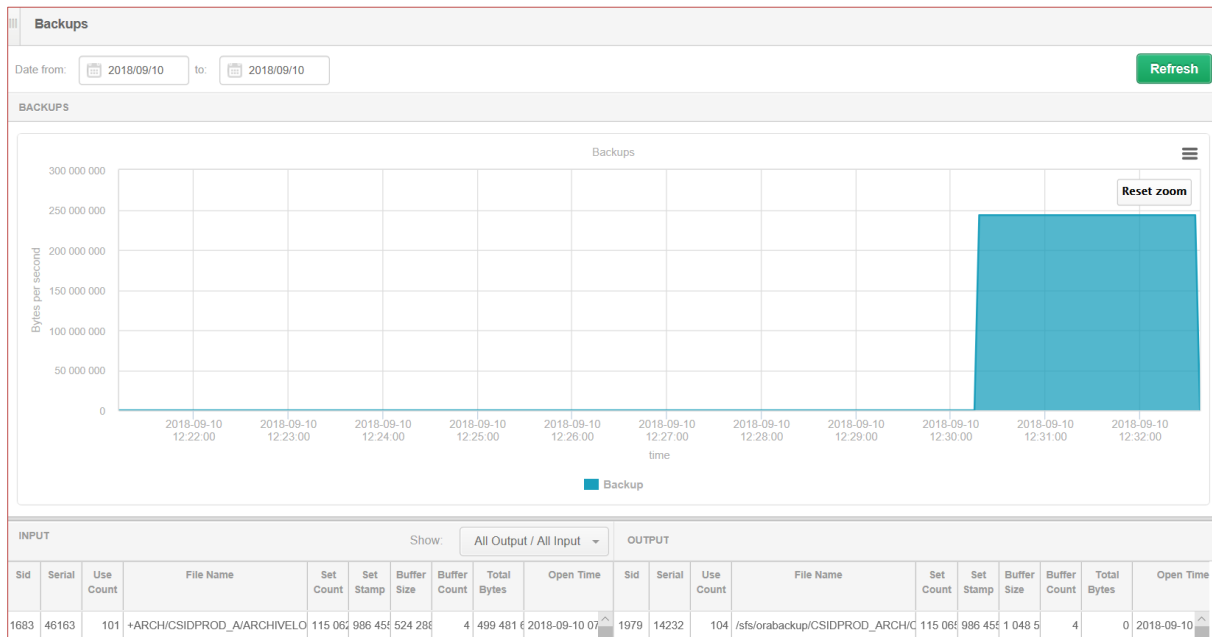
In the case of viewing online statistics, we recommend mark the checkbox "Enable Auto Refresh", this will mean that the data presented on the website will be enlisted and refreshed on the screen in ten-second cycles.

In addition to the standard information available for a given session, the table presents values for a given statistic (Statistic value). This is the delta calculated for the last 10 seconds for a given session in a given statistic. The table also presents the global value (Global value). It is also a delta calculated for the last 10 seconds but containing information from all sessions for a given statistic.

After clicking on a row for a given session, detailed information for a given session is available in the table below, for example, such as: query content or query plan.

6.2.8 Menu Backups - Database Analysis

The data in the [Backups] tab allow for backup performance checks, i.e. the execution time and backup histories. It is divided into three parts:



A graph shows the database security copies made (data files, control files, archive files, init files) that were made in the selected date range.

- The Y-axis of the graph shows the performance of the backup copy expressed in bytes.
- The X-axis of the graph shows the time in which the backup took place.

The "Output" table contains information about the output files created on the medium to which the backup copy using the RMAN tool saved the data:

- SID – System identifier,
- Serial – user session ID,
- Use Count – a counter used to identify rows from different sets of backups,
- Filename – the name of the output file,
- Set Count – number of read or written backups,
- Set Stamp – a set of backups that is read or written,
- Buffer Size – size of the buffers used to read / write the file, in bytes
- Buffer Count – the number of buffers used to read / write the file,
- Total Bytes - the total number of bytes that will be read or written to the file, if known. If it is not known, this column will be empty,
- Open Time – time to start recording the media to a given output file
- Close Time – the time of stopping the execution of writing to the medium to a given output file,
- Elapsed time – the duration of the write to the medium for the given output file
- Max Open Files number of simultaneously opened DISK files. This value only appears in rows where TYPE = 'AGGREGATE'.
- Bytes – size expressed in bytes that has been saved to a given output file,
- Effective Bytes per Second – performance expressed in bytes per second of the speed of writing to a given output file,
- Device Type – the type of device to which the recording was made to the given output file (tape, disk),
- Type – type of operation. Output means a record,
- Status - the status of the operation being performed.

The "Input" table contains information about files that have been subjected to the backup process using the RMAN tool. It contains the same columns as for the OUTPUT table, containing information about data readout.

The [All Output / All Input] option and a mouse click on the security copies of interest on the chart will show in two tables all files that have been subjected to the backup process, as well as all output files, i.e. the names of files stored on the carrier.

6.2.9 Menu Locks - Database Analysis

The page contains information about blockades occurring in a given data base. The entire lock module consists of the following tabs:

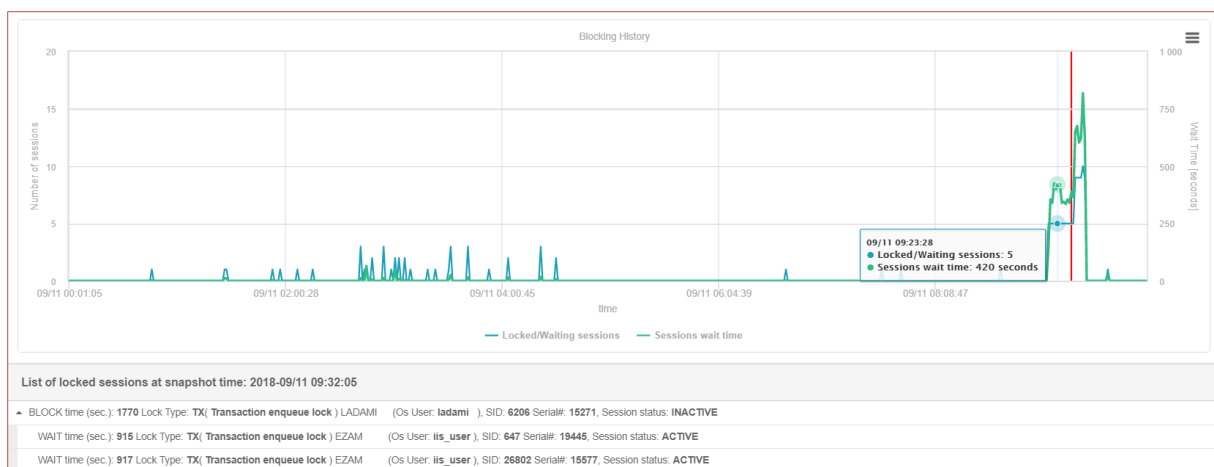
- Locks history – allow to track blockades in time
- Table Locks – allow the current blocking analysis on the database,
- Library Locks – allow the analysis of Library cache locks,
- Locked Objects - show a list of objects on which locks are currently locked.

6.2.9.1 Locks history Tab

The page contains information about the history of blockades occurring in the database. The screen consists of the following areas:

- The filter bar over the date range
- A graph showing the locks in time
- Tree of blocked sessions refreshed after clicking on the fragment / given point of the chart
 - at the top of the tree, blocking sessions are shown
 - in nodes below, waiting sessions blocked by sessions in the parent node
- Details for the selected session
 - Text of the query
 - Session parameters, including transaction opening time, transaction type, etc.

It's possible to search for information for any period by selecting an interesting date range (by default, the page opens with the current date set).



The chart presents information on the number of blocked sessions in a given snap and the duration of blocking, also for a given snap.

Indicating a point on the chart, will get additional information about the session, including Information about:

- blocking and blocking sessions,
- duration of the session (all, not only for a given snap),
- type of blockade,
- user database login for a given session,
- session status.

In addition, after selecting a specific session, information about the content of the query is obtained, as well as detailed information about the session. Analysis of a given query is provided by clicking the [Plus] button next to the Hash Value query ID.

6.2.9.2 Table Locks Tab

The data on the page contains information about blocked sessions and their objects, i.e. those for which one session is blocked by another session.

The Table Locks tab consists of the following areas:

- Filter bar
- Tree of blocked sessions:
 - at the top of the tree show blocking sessions
 - in nodes below waiting sessions blocked by sessions in the parent node
- Details for the selected session
 - Text of the query
 - Session parameters

An example lock screen is presented below:

IMPORTANT: For the selected session, the content of the query and its identifier can also be seen in the blocking tree. At the [Hash Value], the [Plus] button is available allowing users to enter the “SQL Details” screen.

6.2.9.3 Library locks Tab

The page contains information about locks for the Library cache locks type.

6.2.9.4 Locked objects Tab

The site presents information on objects blocked by currently ongoing sessions. This does not mean, however, that the session is blocked by another session. Below is an example of the presentation of objects blocked by ongoing sessions.

Object Name	Owner	Object Type	Lock Mode	Session Id	Serial#	User name	OS User Name	Process	Instance Number
SP_ZA_GRA_TMP	INTER	TABLE	ROW EXCLUSIVE	13667	2433	MISTAK	tmistak	17192.17188	1
DOK_ZA	INTER	TABLE	ROW EXCLUSIVE	40315	12963	MISTAK	tmistak	35496.24496	1
SP_ZA_GRA_TMP	INTER	TABLE	ROW EXCLUSIVE	36285	3887	MISTAK	tmistak	29128.29664	1
SP_ZA_GRA_TMP	INTER	TABLE	ROW EXCLUSIVE	40345	5713	MISTAK	tmistak	35496.24496	1
PH_CLIPBOARD_SPACE	INTER	TABLE	ROW EXCLUSIVE	10427	22191	MISTAK	tmistak	6528.23180	1
P_DOK_ZA_FIL	INTER	TABLE	ROW EXCLUSIVE	40315	12963	MISTAK	tmistak	35496.24496	1
PH_CLIPBOARD_SPACE	INTER	TABLE	ROW EXCLUSIVE	40315	12963	MISTAK	tmistak	35496.24496	1
SP_ZA_GRA_TMP	INTER	TABLE	ROW EXCLUSIVE	40315	12963	MISTAK	tmistak	35496.24496	1

6.2.10 Menu Parameters - Database Analysis

The page allows for viewing and report changes in database parameters over time. The window presents the current status of parameters and their changes over time. Below are the example screens: Status of parameters containing the word “cpu”.

Param name	Value	Description	Is Default	Is Session Modifiable	Is System Modifiable	Is Modified	Is Adjusted
cpu_count	400	number of CPUs for this instance	TRUE	FALSE	IMMEDIATE	FALSE	FALSE
resource_manager_cpu_	400	Resource Manager CPU allocation	TRUE	FALSE	IMMEDIATE	FALSE	FALSE
parallel_threads_per_cpu	2	number of parallel execution threads per CPU	TRUE	FALSE	IMMEDIATE	FALSE	FALSE

Date change	Param value
2018/06/06 10:42:42	480
2018/06/06 10:26:46	470
2018/06/06 10:11:35	424
2018/06/06 09:58:22	349
2018/06/04 22:26:16	336

IMPORTANT: The parameter module is also available from the main menu level after exiting the Database Analysis performance module (go through clicking [Back to dashboard]). Then the system allows analyzing parameters for all monitored databases simultaneously.

6.2.11 Menu Logs - Database Analysis

The Logs module allows the user to check logs from the operation of the database monitoring procedure.

"DBPLUS procedure statistics" tab

In the tab, the user can check if any errors occurred while monitoring the specified database. Additionally, the duration of the monitoring procedure is shown - the number of seconds for 15 minutes.

In the latest version of the application, the presentation of information on the times of collecting data from monitored databases by the CATCHER Windows service has been more detailed. This information concerns the procedure for monitoring the database at 15-minute intervals(snaps).

The data, as before, is available in the Logs tab at the detail level of the given database. From this version, by clicking on a row in the Snaps grid runtime procedure, we get detailed information on the next steps that make up the monitoring procedure.

The screenshot displays the 'Snap procedure run time' grid on the left, with a red box highlighting the snapshot '2019-12-23 15:38:09'. The main area shows 'Snap details at 2019-12-23 15:38:09'. The 'INTERNAL PROCEDURES RUN TIME' table lists 11 steps. Step 2, 'Waits events statistics', is highlighted with a red box. Below it, the 'DETAILS FOR INTERNAL PROCEDURE: WAITS EVENTS STATISTICS' table shows metrics for 'Read data', 'Write data', and 'Rows processed'.

Step	Procedure	Start	End	Duration [Seconds]	Status
1	Check last database restart	2019-12-23 15:38:09	2019-12-23 15:38:09	0	●
2	Waits events statistics	2019-12-23 15:38:09	2019-12-23 15:38:09	0.452	●
3	Latches statistics	2019-12-23 15:38:09	2019-12-23 15:38:10	0.140	●
4	Operating system information	2019-12-23 15:38:10	2019-12-23 15:38:10	0.016	●
5	Query statistics (queries, procedures) including sql text and plans	2019-12-23 15:38:10	2019-12-23 15:38:14	4.727	●
6	Database size (total, used, free space)	2019-12-23 15:38:14	2019-12-23 15:38:14	0	●
7	I/O operation statistics	2019-12-23 15:38:14	2019-12-23 15:38:14	0.140	●
8	Memory informations (SGA including shared pool, db cache size)	2019-12-23 15:38:14	2019-12-23 15:38:15	0.328	●
9	Merge Query statistics to day view	2019-12-23 15:38:15	2019-12-23 15:38:16	1.279	●
10	Merge I/O operations to day view	2019-12-23 15:38:16	2019-12-23 15:38:16	0.078	●
11	Parameters informations	2019-12-23 15:38:16	2019-12-23 15:38:16	0.094	●

Statistics	Type	Counter value	Start	End	Timer Duration [Seconds]
Read data	Timer		2019-12-23 15:38:09	2019-12-23 15:38:09	0.437
Write data	Timer		2019-12-23 15:38:09	2019-12-23 15:38:09	0.016
Rows processed	Counter	58			

Then, by pointing to the step (in the Snap details grid), the user receives information on the duration of the procedure and the number of rows processed (information available only for certain steps).

This is a zoomed-in view of the 'Snap details' section. The 'INTERNAL PROCEDURES RUN TIME' table is visible, with the 'Waits events statistics' row highlighted in red. Below it, the 'DETAILS FOR INTERNAL PROCEDURE: WAITS EVENTS STATISTICS' table is also visible, with the 'Read data', 'Write data', and 'Rows processed' rows highlighted in red.

Information on the status of a given snap is contained in the Status column. If the monitoring process is running correctly, the green dot is displayed in the column.

If one of the monitoring procedure steps has not been performed or has been interrupted and the step concerned is not critical, the user receives information about the reason for the interruption of the step and the status of the entire snap is presented in orange.

Snap procedure run time			Snap details at 2019-12-23 14:06:23					
Date	Work time [seconds]	Status	Step	Procedure	Start	End	Duration [seconds]	Status
2019-12-23 16:08:03	0	● running						
2019-12-23 15:52:49	1	●	1	Check last database restart	2019-12-23 14:06:23	2019-12-23 14:06:23	0	●
2019-12-23 15:37:36	1	●	2	Waits events statistics	2019-12-23 14:06:23	2019-12-23 14:06:23	0.047	●
2019-12-23 15:22:23	4	●	3	Latches statistics	2019-12-23 14:06:23	2019-12-23 14:06:23	0.031	●
2019-12-23 15:07:09	2	●	4	Operating system information	2019-12-23 14:06:23	2019-12-23 14:06:23	0.016	●
2019-12-23 14:51:56	1	●	5	Query statistics (queries.procedures) including sql text and plans	2019-12-23 14:06:23	2019-12-23 14:06:23	0.671	●
2019-12-23 14:36:43	1	●	6	Database size (total, used, free space)	2019-12-23 14:06:23	2019-12-23 14:14:45	501.122	●
2019-12-23 14:21:29	4	●	7	I/O operation statistics	2019-12-23 14:14:45	2019-12-23 14:14:45	0.281	●
2019-12-23 14:06:23	503	●	8	Memory informations (SGA including shared pool, db cache size)	2019-12-23 14:14:45	2019-12-23 14:14:45	0.125	●
2019-12-23 13:51:10	1	●	9	Merge Query statistics to day view	2019-12-23 14:14:45	2019-12-23 14:14:45	0.078	●
2019-12-23 13:35:56	1	●	10	Merge I/O operations to day view	2019-12-23 14:14:45	2019-12-23 14:14:45	0.031	●
2019-12-23 13:20:43	4	●	11	Parameters informations	2019-12-23 14:14:45	2019-12-23 14:14:45	0.062	●
2019-12-23 13:05:30	1	●	DETAILS FOR INTERNAL PROCEDURE: DATABASE SIZE (TOTAL, USED, FREE SPACE)					
2019-12-23 12:50:16	1	●	Statistics	Type	Counter value	Start	End	Timer Duration [seconds]
2019-12-23 12:35:03	1	●	Read data	Timer		2019-12-23 14:06:23		0
2019-12-23 12:19:49	4	●	Write data	Timer				0
2019-12-23 12:04:42	404	●	Rows processed	Counter	0			
2019-12-23 11:49:29	1	●	ERROR LOGS FOR SELECTED STEP: DATABASE SIZE (TOTAL, USED, FREE SPACE)					
2019-12-23 11:34:16	1	●	Date	Log message				
2019-12-23 11:19:02	4	●	2019-12-23 14:14:45	Error reported in following program: StandardSnap_CatchODBSize. Execution for query SELECT /*+ALL_ROWS*/ file_id, nvl(Sum(bytes),0) bytes FROM DBA_free_space GROUP BY file_id timeout-ed at DBPLU...				
Average time	55	-						
Min time	0	-						

If there was a connection problem at the time of the monitoring procedure or the problem related to a critical step for a given procedure, the status information is presented in red.

Snap procedure run time			Snap details at 2019-12-23 16:15:00					
Date	Work time [seconds]	Status	Step	Procedure	Start	End	Duration [seconds]	Status
2019-12-23 16:15:00		●						
2019-12-23 16:00:00		●	1	No any steps executed for specified snapshot			0	●
2019-12-23 15:45:00		●						
2019-12-23 15:30:00		●						
2019-12-23 15:15:00		●						
2019-12-23 15:00:00		●						
2019-12-23 14:45:00		●						
2019-12-23 14:30:00		●						
2019-12-23 14:15:00		●						
2019-12-23 14:00:00		●						
2019-12-23 13:45:00		●						
2019-12-23 13:30:00		●						
2019-12-23 13:15:00		●						
2019-12-23 13:00:00		●						
2019-12-23 12:45:00		●						
2019-12-23 12:30:00		●						
2019-12-23 12:15:00		●						
2019-12-23 12:00:00		●						
2019-12-23 11:45:00		●						
2019-12-23 11:30:00		●						
Average time	0	-						
Min time	0	-						
Max time	0	-						
Count snaps	65	-						

If the monitoring procedure is in progress, this information is visible in the status (running) field, as well as the Online steps refresh button is displayed, after which the information on the monitoring procedure progress is refreshed.

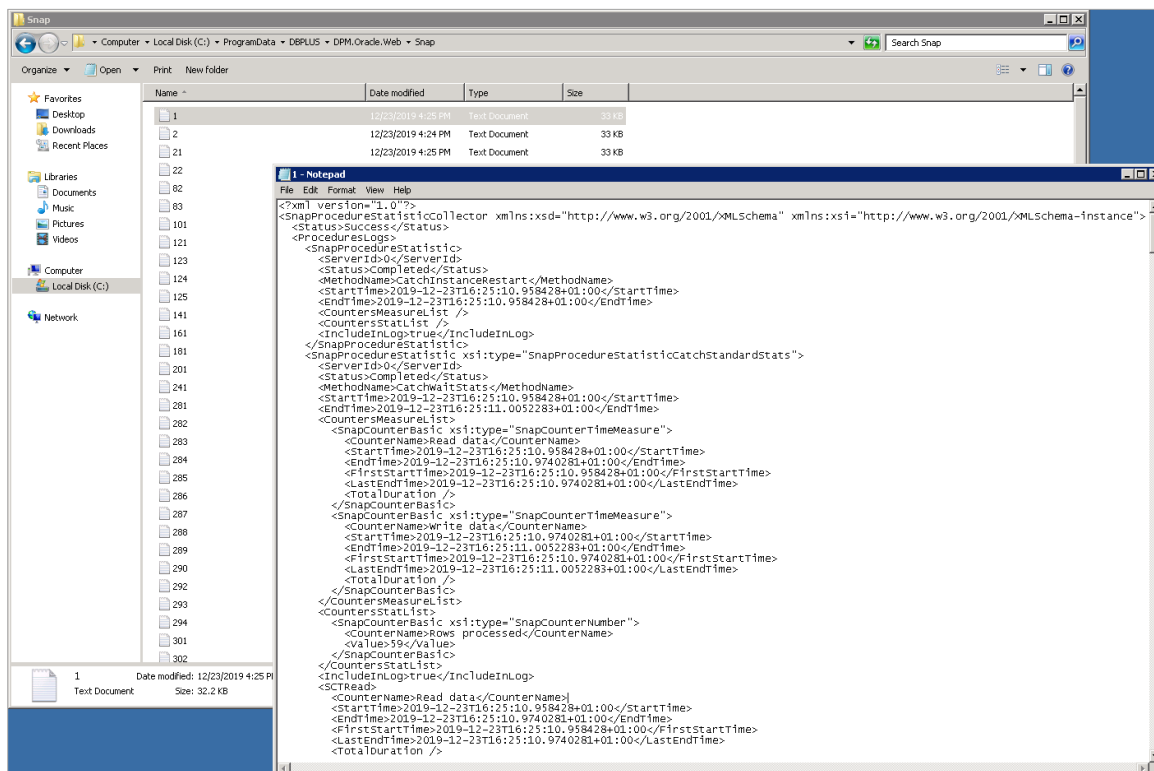
Snap procedure run time			Snap details at 2019-12-24 10:28:11 with selected currently executed step					
Date	Work time (seconds)	Status	Step	Procedure	Start	End	Duration (seconds)	Status
2019-12-24 10:28:11	7	● running	1	Check last database restart	2019-12-24 10:28:11	2019-12-24 10:28:11	0	●
2019-12-24 10:12:59	36	●	2	Wait events statistics	2019-12-24 10:28:11	2019-12-24 10:28:16	5.444	●
2019-12-24 09:57:46	72	●	3	Latches statistics	2019-12-24 10:28:16	2019-12-24 10:28:17	0.234	●
2019-12-24 09:42:34	35	●	4	Operating system information	2019-12-24 10:28:17	2019-12-24 10:28:17	0.047	●
2019-12-24 09:27:21	34	●	5	Query statistics (queries, procedures) including sql text and plans	2019-12-24 10:28:17		0.858	●

DETAILS FOR INTERNAL PROCEDURE: WAIT'S EVENTS STATISTICS					
Statistics	Type	Counter value	Start	End	Timer Duration (seconds)
Read data	Timer		2019-12-24 10:28:11	2019-12-24 10:28:16	0
Write data	Timer		2019-12-24 10:28:16	2019-12-24 10:28:16	0
Rows processed	Counter	62			

In addition, all problems with the monitoring procedure are available in the form of a list on the Procedure Errors tab.

Information on the monitoring procedure is also included in the form of a file on the application server. The file contains information about the last snap performed on a given database. The file is located in the folder: C:\ProgramData\DBPLUS\DPM.Oracle.Web\Snap

Each file is marked with a digit assigned to the database when it is included in the monitoring (dbplus_central_servers table in the DBPLUS schema in the repository database).



6.2.12 Menu Reports - Database Analysis

The following reports are available in the Reports module:

- Performance Report,
- Not Used indexes.

6.2.12.1 Performance Report

The report presents the performance of the database in the selected time period. The report contains information about:

- Top queries operating in the database for:
 - Duration: Elapsed Time
 - Utilization Processors: CPU Time
 - Readings from disk devices
 - Block reads from memory
 - Number of queries
- The duration of blockades in an hourly manner
- Top wait lists
- Top latches

6.2.12.2 Not used indexes

The report can be run for selected objects (specified index, tablespace). The system checks if the indexes were used by queries running in a given period of time.

Report of not used indexes

Index name Indexes on table Index owner Table owner Tablespace

Show indexes that were NOT used by any sql query executed in the period from: to: Run Report

[Show additional filters](#)

! Index usage is calculated based on checking the execution statistics of sql queries that spent on database at least 5 seconds
 For each filter fields you can use % character to run report with like condition

Index owner	Index name	Index type	Table owner	Table name	Uniqueness	Compression	Tablespace
HR	LOC_COUNTRY_IX	NORMAL	HR	LOCATIONS	NONUNIQUE	DISABLED	USERS
HR	EMP_JOB_IX	NORMAL	HR	EMPLOYEES	NONUNIQUE	DISABLED	USERS
HR	EMP_DEPARTMENT_IX	NORMAL	HR	EMPLOYEES	NONUNIQUE	DISABLED	USERS
HR	LOC_STATE_PROVINCE	NORMAL	HR	LOCATIONS	NONUNIQUE	DISABLED	USERS
HR	JHIST_DEPARTMENT_IX	NORMAL	HR	JOB_HISTORY	NONUNIQUE	DISABLED	USERS
HR	JHIST_EMPLOYEE_IX	NORMAL	HR	JOB_HISTORY	NONUNIQUE	DISABLED	USERS
HR	JHIST_JOB_IX	NORMAL	HR	JOB_HISTORY	NONUNIQUE	DISABLED	USERS
HR	DEPT_LOCATION_IX	NORMAL	HR	DEPARTMENTS	NONUNIQUE	DISABLED	USERS
HR	EMP_NAME_IX	NORMAL	HR	EMPLOYEES	NONUNIQUE	DISABLED	USERS
HR	EMP_MANAGER_IX	NORMAL	HR	EMPLOYEES	NONUNIQUE	DISABLED	USERS
HR	COUNTRY_C_ID_PK	IOT - TOP	HR	COUNTRIES	UNIQUE	DISABLED	USERS

IMPORTANT: Please note that the report is calculated on the basis of queries that lasted at least 5 seconds in the database !!!

6.3 Menu Space Monitor

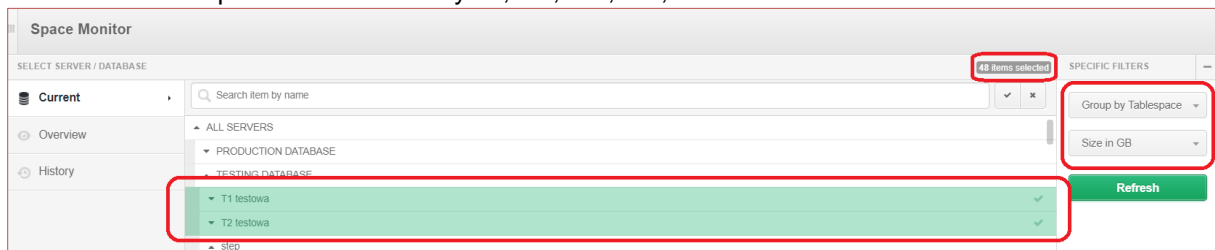
On the website ability view the occupancy of all monitored databases is provided. As part of the preview, three options follow:

- Current - verification of the current status of occupancy,
- Overview - presents the occupation of databases for a given period of time in tabular form,
- History - presents the occupation of databases for a given day.

For each variant, the result can be verified for the selected filter:

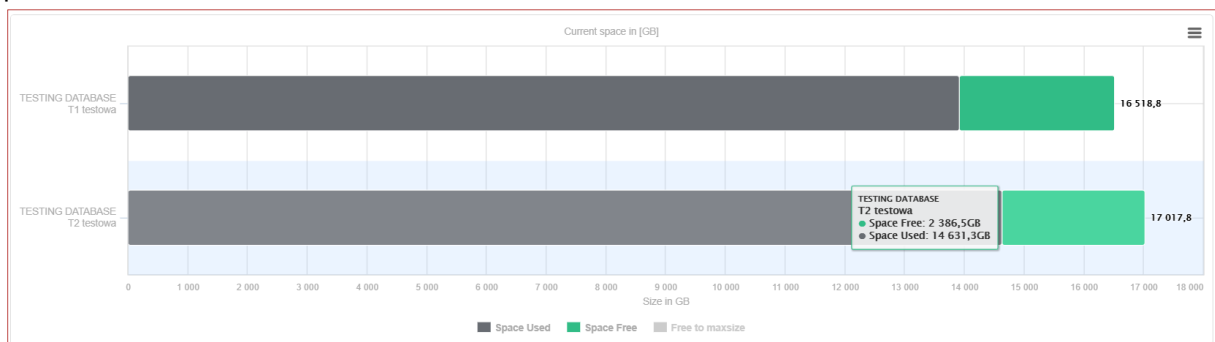
- all databases,
- grouping by the type of database,
- grouping by database,
- grouping of the database after Tablespace,
- grouping after the database file.

The result can be presented in units Bytes, KB, MB, GB, TB.



As part of a given option, each time selected databases are indicated, by marking in the list (highlighted in green). Each time after indicating the database, the selection is counted and presented on the page (the number shows all Tablespace for a given database).

The graph below presents the Current occupancy of databases grouped after the database name, presented in units of GB.



6.4 Menu Parameters

The website allows for the parameters of monitored databases to be verified. Two options are available:

- Overview - shows the current values of database parameters,
- History - presents information on the change of the parameter value for the selected period.

As in the case of **Space Monitor**, option to indicate which database user wants to verify (by marking - highlighting in green).

Available parameters can be modified by selecting a specific parameter by entering the name (or part

of it) or by entering the parameter value user is looking for.

In the presented example, two databases were selected and the name of the searched parameter having the expression %cpu% in the name, with the value equal to 32.

Parameters overview

SELECT SERVER: 2 items selected

SEARCH: Search item by name

SPECIFIC FILTERS: Parameter name: cpu, Parameter value: 32

Database selection: T1 testowa, T2 testowa

Database type	Database	Param name	Param value
TESTING DATABASE	T2 testowa	cpu_count	32
TESTING DATABASE	T2 testowa	resource_manager_cpu_allocation	32

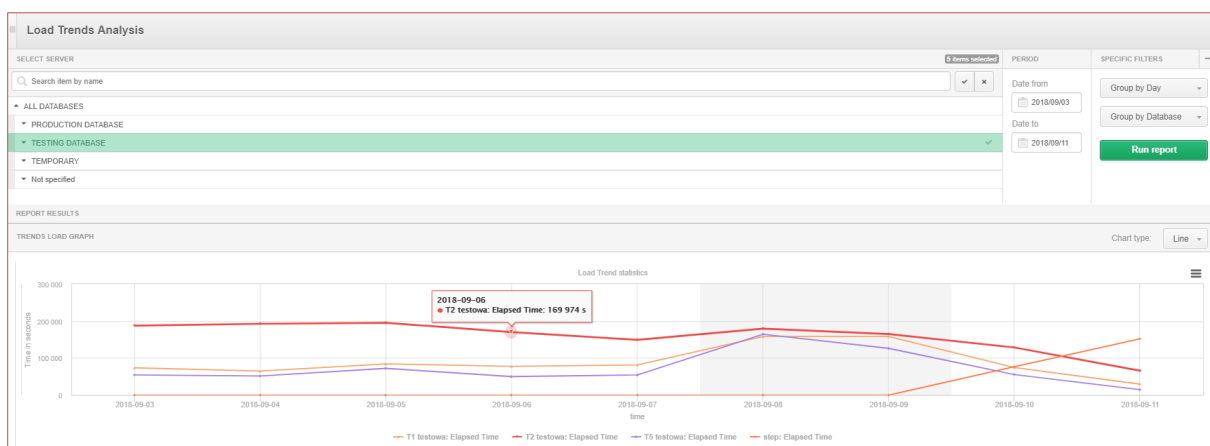
Date change	Param value
2018/05/11 11:16:22	32
2017/08/29 10:21:53	24

In addition, information when this parameter was last time modified is provided and what value has been set.

6.5 Menu Reports – Load Trends

Website provides the option of comparing database statistics in the context of selected monitored databases. For this purpose, user should firstly select the databases that they want to extract, then select the period for which a comparison should be run and set the appropriate filter if necessary. As a result, user obtains a graph for a given statistic.

The graph shows a view for all databases with the type "TESTING DATABASE" (5 Items), and for the selected Elapsed Time statistics, for the period from 03/09/2018 to 11/09/2018.



The selection of any statistics is made by selecting the desired column in the table below the graph. The selection can be made for one or more statistics at the same time.

TRENDS STATISTICS																Clear select	
Logdate	DB type	Database	Elapsed Time [seconds]	Cpu Time [seconds]	Sorts [rows]	Fetches [rows]	Executions	Disk reads [rows]	Buffer gets [blocks]	Rows processed [rows]	Latches [seconds]	Waits [seconds]	Locks [seconds]	Sessions	Active sessions	Warning alerts	Critical alerts
2018-09-03	TESTING DATABASE	T2 testowa	187 550	22 106	2 836 605	70 284 960	46 635 426	1 930 020 MB	5 810 241 257	186 490 270	665	202 129	0	130	7	0	72
2018-09-03	TESTING DATABASE	T1 testowa	73 574	18 515	3 441 626	93 239 702	79 277 423	977 304 MB	5 129 515 732	106 604 444	10 843	44 000	219	201	12	0	0
2018-09-03	TESTING DATABASE	T5 testowa	54 504	7 181	290 233	21 202 535	10 936 830	393 313 MB	698 831 319	28 455 898	108	31 668	15	132	1	0	0
2018-09-04	TESTING DATABASE	T1 testowa	64 734	14 742	3 430 619	93 902 557	79 968 801	902 947 MB	3 039 789 883	103 290 937	11 267	44 920	267	191	12	0	0
2018-09-04	TESTING DATABASE	T2 testowa	192 429	39 942	868 036	59 844 388	44 029 496	1 841 322 MB	4 006 104 527	117 352 790	24	196 434	162	149	7	0	102
2018-09-04	TESTING DATABASE	T5 testowa	51 439	7 136	281 643	21 389 309	10 967 353	459 456 MB	638 908 956	29 921 560	114	30 044	15	121	1	0	0
2018-09-05	TESTING DATABASE	T2 testowa	195 127	25 252	1 248 376	67 246 119	53 789 413	2 098 765 MB	5 090 707 883	141 721 982	2 401	200 383	0	183	7	2	99
2018-09-05	TESTING DATABASE	T1 testowa	84 966	18 933	3 423 710	90 719 386	77 206 322	1 355 295 MB	5 248 034 755	100 353 341	13 052	51 665	308	182	12	0	0
2018-09-05	TESTING DATABASE	T5 testowa	72 133	9 368	284 646	21 154 589	10 887 288	866 205 MB	691 574 886	28 806 185	115	49 901	14	136	2	0	0

6.6 Menu Servers Monitor

Information on the performance of the DBPLUS Performance Monitor is available on the website. Two submenus available from there:

- Application architecture
- Logs

6.6.1 Application architecture

The site contains information on the status of monitored databases. For each of the monitored databases, information about the date of the last collected snapshot (download information about the database statistics) and the last action (operations from the level of the DBPLUS Application towards the database) is available.

In the middle part of the Monitoring service page, information about the status of the DBPLUSCATCHERSERVICE monitoring service is also available. Information is also collected on the amount of memory used and the CPU utilization of the server on which the DBPLUS Performance Monitor application is installed.

The screenshot displays the 'Monitoring service' page. On the left, under 'DBPLUS Oracle Catcher', the service status is 'Running' (indicated by a green dot). Below this, a table shows service activity and resource usage:

Service status	● Running
Last service activity	2018/09/11 16:51:15
Machine Total Memory	4095 MB
Machine Memory Usage	3474 MB
DBPLUSCATCHER Memory Usage	846 MB
DBPLUSCATCHER CPU Usage	0.7 %

At the bottom of this section are 'Refresh' and 'View service activity' buttons. On the right, under 'Repository Information', the database service name is 'RMANCAT' and the host is '10.234.105.56'.

On the right side of DBPLUS Performance Monitor, data about the database on which the repository is installed is located. The service name or SID is indicated (depending on the choice during the installation process) and the host name.

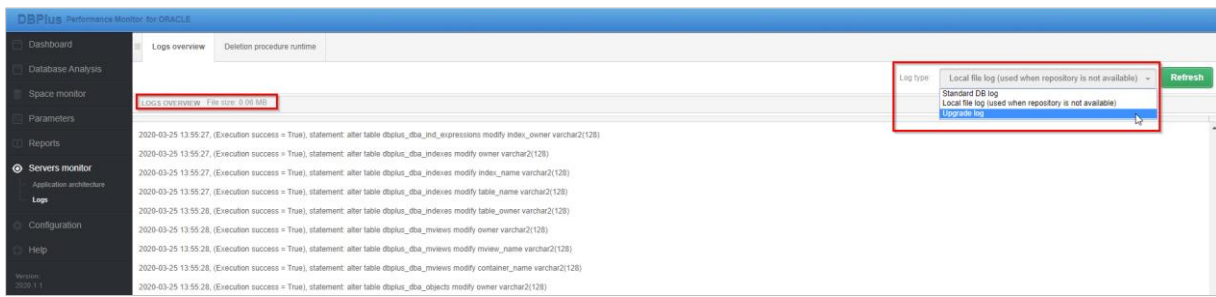
6.6.2 Logs

The website contains information about any irregularities or errors related to the monitoring activity. Information is available about the name of the database server on which the problem was detected as well as the date and content of the reported error.

After entering the Logs overview tab, the User will be presented with default logs saved in the Repository database (Standard DB Log), that contain information about problems with possible monitoring.

The User will also have the option of displaying information available in the logs available locally on the application server (Local file log). Information about problems is saved there when it is not possible to save this information in the repository database.

The next log concerns information related to the application update process. This file is created during the application upgrade process (downloading the new version). We save information about changes made to the data model as well as the update process. In addition, information about the size of the file is displayed for each file.



6.7 Menu Configuration

In this menu the application provides the possibility to modify the configuration regarding the performance of the DBPLUS Performance Monitor. Several submenus are available:

- Settings,
- Databases,
- Reference lists,
- Security,
- Alert settings.

6.7.1 Settings

This submenu allows to modify the parameters controlling the operation of the application. The Settings tab presents the basic configuration parameters. Some parameters are set globally for all monitored data bases.

Parameter	Value	Description	
MONITOR_LITERAL_QUERIES	OFF	DBPLUSCATCHER service can monitor literal queries executed on your databases. Set to [ON] to run feature on all databases or make such change for specified database.	Edit
KEEP_SNAPSHOT_HISTORY_DAYS	31	Number of days how long to keep detail statistics for sql statement executions, waits, latches, performance counters.	Edit
SECURITY	OFF	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.	Edit
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, sql instance cpu, waits, sessions, etc.)	Edit
LOCKING_SNAPSHOT_FREQUENCY	60	The interval time in seconds between each snapshot of locks made by DBPLUS CATCHER service. The parameter can be setup separately for each instance. In a case of frequent locks, please consider lower value for LOCKING_SNAPSHOT_FREQUENCY. In a case of rarely occurred locks, please use bigger value for it.	Edit
STATEMENT_LENGTH_LIMIT	4000	Setting used during collecting statistics data for running queries. It's the maximum sql statement length that will be stored in system repository. Statements with length greater than STATEMENT_LENGTH_LIMIT will be truncated. To switch off this setting please use value 0.	Edit
STORE_ONETIME_STATEMENT_TEXT	ON	Setting used during collecting statistics data for running queries. It lets to store (or not) the statement text for the queries running only once.	Edit

Depending on the quality of queries and the type of problems in the system, following options can be enabled:

- **MONITOR_LITERAL_QUERIES** - collecting queries with literals,
- **LOCKING_SNAPSHOT_FREQUENCY** - changing the frequency of collecting block history,
- **KEEP_SNAPSHOT_HISTORY_DAYS** - number of storage days for the retail history of database performance.

To change the configuration for a dedicated database, select the database instances at the bottom of the page and make changes by clicking the [Edit] button.

Parameter	Value	Description	
LOCKING_SNAPSHOT_FREQUENCY	60	The interval time in seconds between each snapshot of locks made by DBPLUS CATCHER service. The parameter can be setup separately for each instance. In a case of frequent locks, please consider lower value for LOCKING_SNAPSHOT_FREQUENCY. In a case of rarely occurred locks, please use bigger value for it.	Edit
MONITOR_LITERAL_QUERIES	OFF	DBPLUSCATCHER service can monitor literal queries executed on your databases. Set to [ON] to run feature on all databases or make such change for specified database.	Edit
STATEMENT_LENGTH_LIMIT	4000	Setting used during collecting statistics data for running queries. It's the maximum sql statement length that will be stored in system repository. Statements with length greater than STATEMENT_LENGTH_LIMIT will be truncated. To switch off this setting please use value 0.	Edit
STORE_ONETIME_STATEMENT_TEXT	ON	Setting used during collecting statistics data for running queries. It lets to store (or not) the statement text for the queries running only once.	Edit

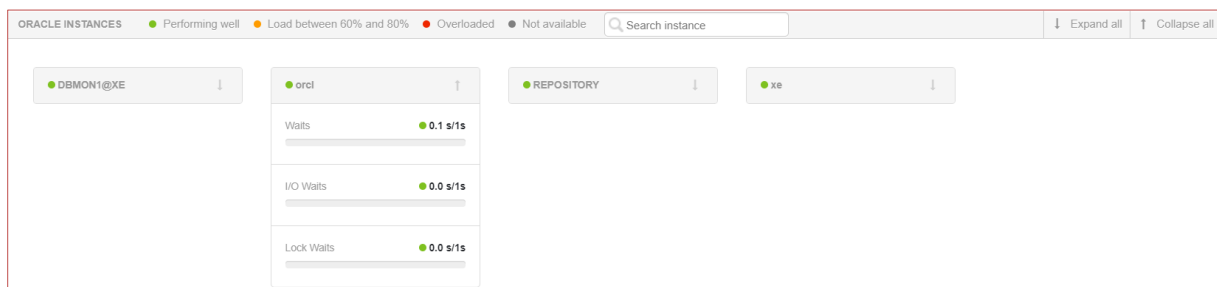
IMPORTANT: Parameters can be set at the general level or for specific / selected databases. This applies to the parameters: LOCKING_SNAPSHOT_FREQUENCY, MONITOR_LITERAL_QUERIES.

Dashboard icon settings

At this point, the user has the option of changing the levels of alarm presentation on mainly the Dashboard page.

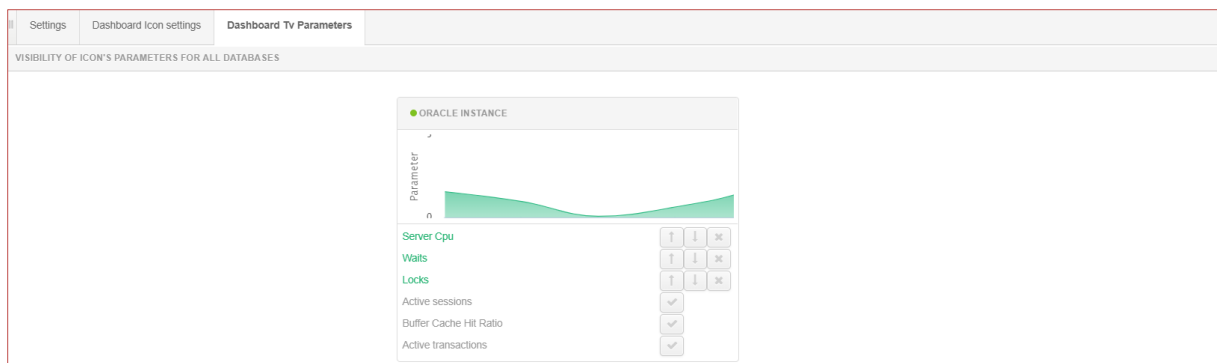
Settings		Dashboard Icon settings		Dashboard TV Parameters		
ALERTS CONFIGURATION						
Name	Alert Enabled	Value for Alert	Warning Enabled	Value for Warning	Description	
INSTANCE_CPU	YES	80	YES	50	Alert if sql instance process utilization is equal or greater than specified in the parameter. <i>Alert is calculated every 15 seconds.</i>	
SERVER_CPU	YES	80	YES	50	Alert if server cpu utilization is equal or greater than specified in the parameter. <i>Alert is calculated every 15 seconds.</i>	
INSTANCE ALERTS CONFIGURATION - PLEASE SELECT A SERVER					T1 testowa	Restore defaults
Name	Alert Enabled	Value for Alert	Warning Enabled	Value for Warning	Description	
INSTANCE_CPU	YES	80	YES	50	Alert if sql instance process utilization is equal or greater than specified in the parameter. <i>Alert is calculated every 15 seconds.</i>	
SERVER_CPU	YES	80	YES	50	Alert if server cpu utilization is equal or greater than specified in the parameter. <i>Alert is calculated every 15 seconds.</i>	

Depending on this configuration, information on the alarm level will be presented on the Dashboard. If the warning value is exceeded (Warning level), the icon will be displayed in orange. When the alarm threshold (Alert level) is exceeded, the icon next to the Oracle database instance name will turn red.



Dashboard Tv Parameters

On the page it is possible to change the information presentation settings as it will be displayed in the Television mode on the main Dashboard screen. The changes can be used for all monitored instances as well as for each database separately.



6.7.2 Databases

This page allows to configure which databases should be monitored and set the database type. The correct type setting for each database allows the user to use this grouping in various functions of the

DBPLUS Performance Monitor application, such as Space Monitor, when present the size of databases assigned to a given group.

On the website it is also possible to set, among others:

- assignment of categories for the database
- settings of the database name display format - the user has the option of displaying after:
 - instance name
 - SID-base
 - in the user @ SID format
 - determine users own name

The format assignment and / or change of the database category occurs after selecting it in advance:

View databases and its connections

DATABASES SETTINGS

Default Database Name Format: User@SID

ORACLE DATABASES LIST

Host name	Instance name	Database SID	Used database name	Type	Enabled
WIN-PVM04LTCT8A	xe	XE	DBMON1@XE	Not Specified	<input checked="" type="checkbox"/>
WIN-PVM04LTCT8A	xe	XE	dbmon2@XE	DEVELOPMENT	<input checked="" type="checkbox"/>
WIN-PVM04LTCT8A	orcl	ORCL	dbplus@ORCL	Not Specified	<input checked="" type="checkbox"/>
[HOST]	[SID]	[SID]	dbplus@[SID]	Not Specified	<input type="checkbox"/>
WIN-PVM04LTCT8A	xe	XE	REPOSITORY	MonitoredByDbplus	<input checked="" type="checkbox"/>

DETAILS FOR SELECTED DATABASE

Basic | Connection properties

Connection Type: Basic

Hostname: localhost

Connect By: SID

SID: XE

TCP Port: 1521

Type: DEVELOPMENT

Database Name format: Default

As a result, the appropriate name is presented on the following screens:

- Dashboard screen - database icons bar:

bibi@XE

REPOSITORY

- In the main menu - after clicking Database Analysis

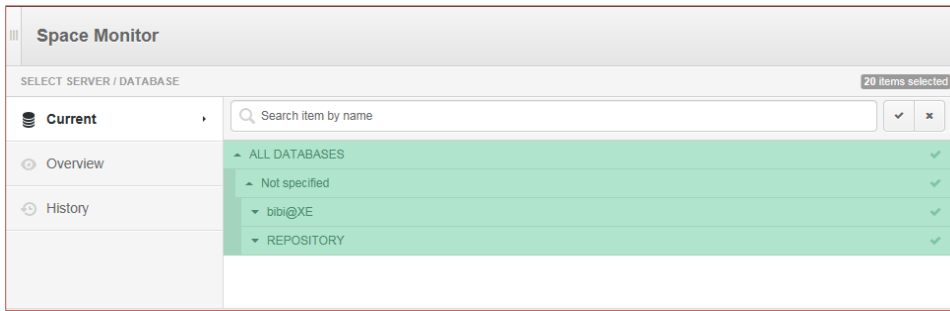
Dashboard

Database Analysis

- bibi@XE
- REPOSITORY

Space monitor

In the options available to all databases - Space Monitor / Parameters in the filter form:



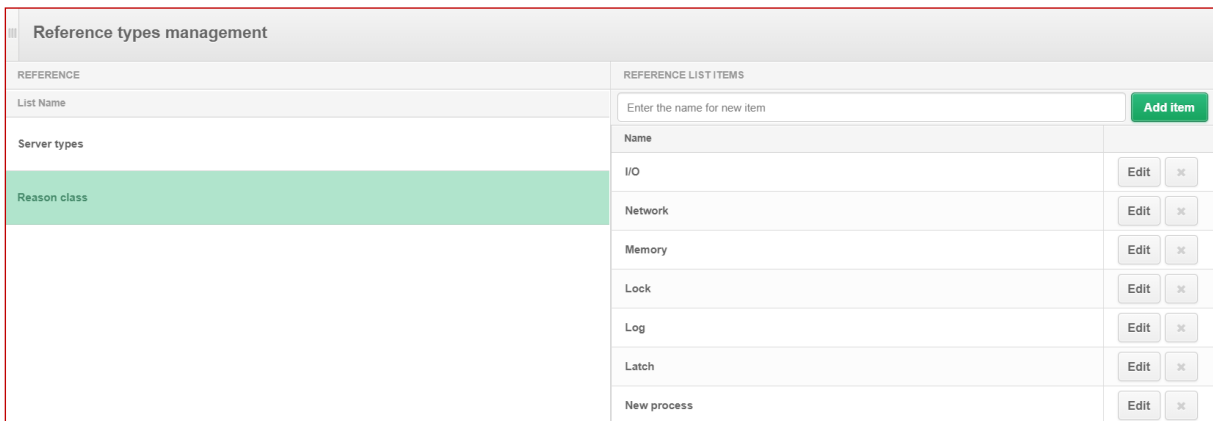
In addition, in the Connection properties tab, a preview of the database connection configuration with the DBPLUS Performance Monitor application is available.

6.7.3 Reference lists

This tab contains the system dictionaries used in the application. Existing dictionary data can be freely modified.

Two collections are currently available:

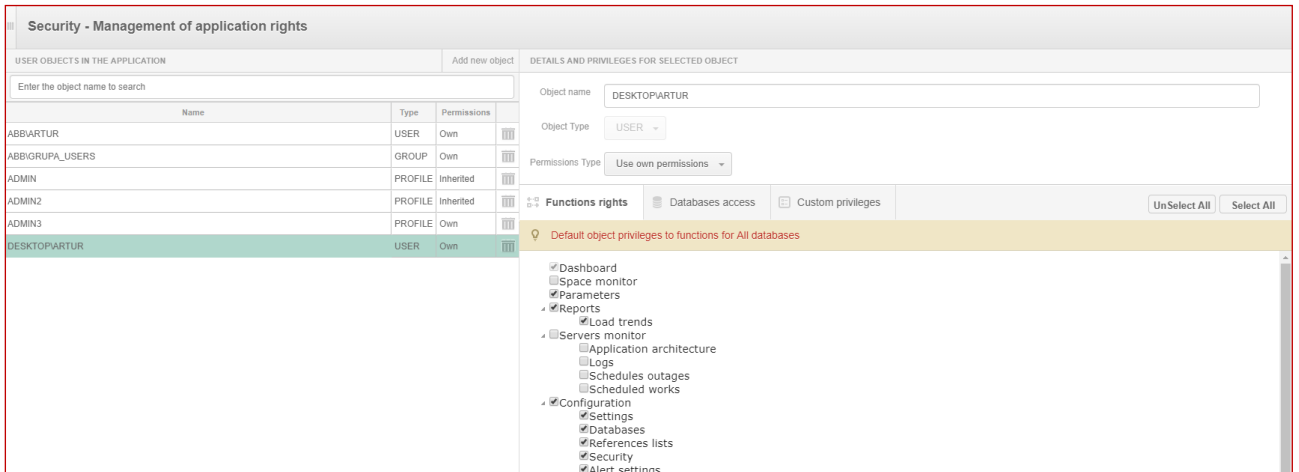
- Server types (server type that can be assigned to the indicated database),
- Reasons class (class Causes, additional information assigned to the cause of the problem in the database).



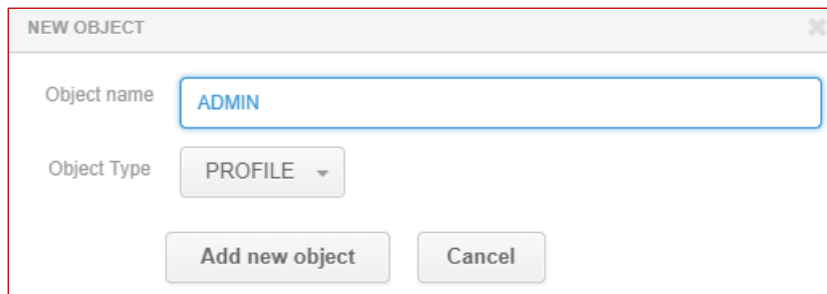
6.7.4 Security

This tab provides the option of setting access for a user ,group of users or profiles. Access is granted at the database level and at the level of available pages in the menu.

The PROFILE access object allows assigning appropriate access to the profile and then granting rights by assigning the profile to the user or groups.



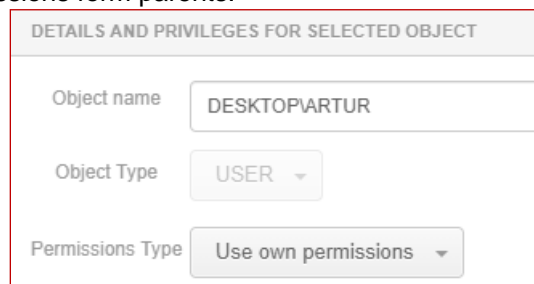
In order to create a new object, eg a profile (PROFILE), click on [Add new object], then select the object type "PROFILES" and give the name of the object.



To assign permissions to a given object, select it from the list on the left side of the screen. After clicking on the object on the right side, the page with the access configuration will be displayed.

First you need to choose whether the permissions will be:

- own (Use own permissions).
- inherited permissions form parents.

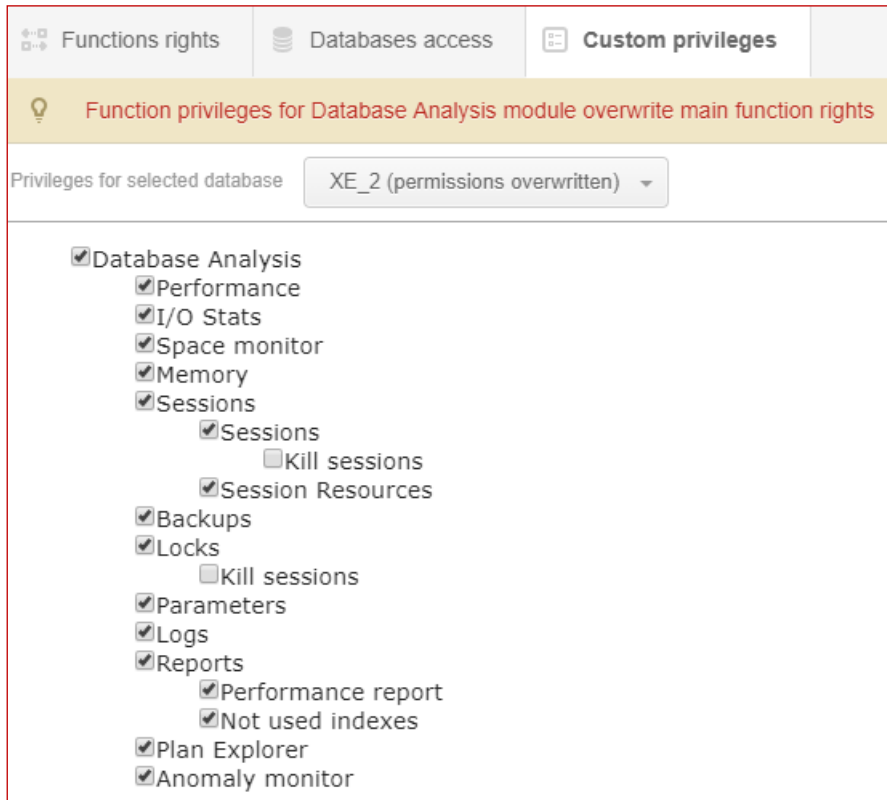


6.7.4.1 Own permissions

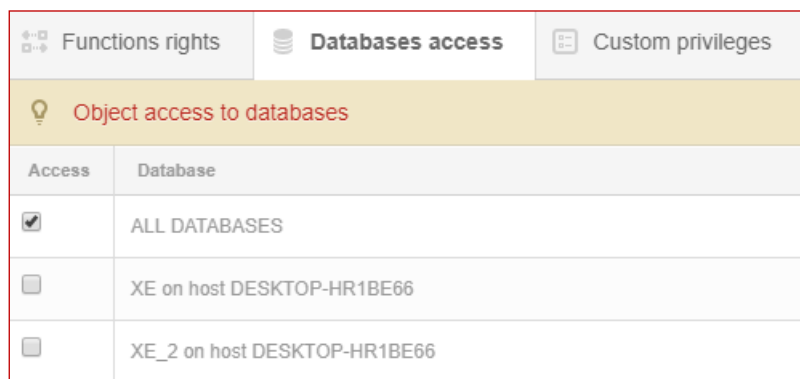
If you choose (own permissions), you have three tabs to configure permissions:

- Function rights,
- Databases access,
- Custom privileges

Functional settings allow you to give rights to pages or functionality in the application at the global level for a given user / group or profile for all databases. You can override these rights by granting custom permissions for a specific database. Custom permissions can only be changed for the Database Analysis module. Custom permission is superordinate to a given database in relation to functional rights. If you assign custom permissions, the (permissions overwritten) message will be displayed next to the database name.



In addition, you can restrict access to specific databases. To do this, in the Database access tab, select the appropriate check boxes for a given database or select ALL_DATABASES. If certain bases are restricted, this will also limit the Custom privileges tab.



6.7.4.2 Inherited permissions form parents

If you choose inherited rights, you can specify which profile or profiles to use for a given user or user group. Each profile contains a list of objects and access to which. Granting permissions to multiple

profiles for the user will result in the entitlement for a given user being the sum of rights for selected profiles.

Profiles assignment	
Permissions to inherited from assigned profiles	
Access	Profile Name
<input type="checkbox"/>	ADMIN
<input type="checkbox"/>	ADMIN2
<input type="checkbox"/>	ADMIN3

Attention! In order to enable the functionality of limited access to the application, you must change the settings at the level of the DBPLUS Configuration Wizard> Applications settings> Applications Options> Configure. As well as change the status of the SECURITY parameter to ON.

Parameter	Value	Description	
SECURITY	ON	Application can work in SECURITY mode set to ON or to OFF. It means that application uses (or doesn't 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, walls, sessions, etc.)	Edit
LOCKING_SNAPSHOT_FREQUENCY	300	The interval time in seconds between each snapshot of locks made by DBPLUS CATCHER service. The parameter can be setup separately for each instance. In a case of frequent locks, please consider lower value for LOCKING_SNAPSHOT_FREQUENCY. In a case of rarely occurred locks, please use bigger value for it.	Edit

Below screen with the DBPLUS Configuration Wizard:

It's recommended to use the same user type/account for DBPLUSORACLECATCHER service, IIS application and oracle instances monitoring purposes. Please do not use account with administrator privileges.

In application security tab please specify if application should be available in anonymous mode (for every user who enter the application url) or in secure mode (for users who authenticate)

If you want to change the protocol, you have to do it directly in IIS manager.

Application pool settings (AppPoolDPM)

Login type: LocalSystem
 Username:
 Password:

Website settings (DBPLUS Website)

Protocol: http Binding property: Default
 Port: 80 Host name:
 Application path: C:\Program Files (x86)\DBPLUS.Or

Application security

Use windows authentication in access to application

After saving the settings, the next steps to manage the settings are made from the level of the web system application in the option **Configuration-> Security**.

6.7.5 Alert settings



The alert module is available from the main menu, i.e. **Configuration-> Alert settings**. From this tab users have the ability to:

- Parameter settings related to mailing - i.a. data of the mail server and account from which alert messages will be sent,
- Making general module settings,
- Define alerts,
- Specify the list of alert recipients.

6.7.5.1 Mail settings Tab

For the information about an alert to be sent via email, user must configure the SMTP server settings.

As part of the configuration, users have the option to set the frequency of sending information about the event, depending on the configuration it is from 1 minute to 1 hour.

Mail settings	General settings	Alerts definition	Alerts subscription
<p> List of email configuration parameters.</p>			
<p><input checked="" type="checkbox"/> Send alerts by mail</p>			
Mail Agent Interval	<p>once per 5 minutes</p>		
SMTP Mail server	<p>pop3-dbpluskonto.ogicom.pl</p>		
Port	<p>587</p>		
Sender email address	<p>alert@dbplus.pl</p>		
<p><input checked="" type="checkbox"/> smtp authentication</p>			
Username	<p>alert@dbplus.pl</p>		
Password	<p>.....</p>		
<p><input type="checkbox"/> enable SSL</p>			
Test mail address	<p></p>		<p> Send test mail</p>
<p>Save mail settings</p>			

IMPORTANT: Email alerts for all databases are sent from one email account.

6.7.5.2 General settings Tab

In this tab, users can make general settings of the alert module. User has the option to configure parameters related to the alert mechanism.

Mail settings	General settings	Alerts definition	Reasons & Problems definition	Events subscription
Elapsed Time greater than <input type="text" value="400"/> seconds <small>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)</small>				
History Days <input checked="" type="checkbox"/> Mon <input checked="" type="checkbox"/> Tue <input checked="" type="checkbox"/> Wed <input checked="" type="checkbox"/> Thu <input checked="" type="checkbox"/> Fri <input type="checkbox"/> Sat <input type="checkbox"/> Sun <small>We recomend to select working days only</small>				
Number of Days Back in History <input type="text" value="30"/> <small>How long history would be included in snapshot alerts calculation</small>				
STATEMENTS SETTINGS				
Number of Top Queries to check <input type="text" value="20"/> chosen by <input type="text" value="Elapsed time"/> <small>How many top statements from each snapshot would be check by Alert Engine</small>				
Number of Days Back in History <input type="text" value="7"/> <small>How long statement history would be included in snapshot alerts calculation</small>				
WAIT EVENTS SETTINGS				
Number of Top Waits to check <input type="text" value="3"/>				
Number of Days Back in History <input type="text" value="7"/> <small>How long wait history would be considered in snapshot alerts calculation</small>				
Save settings				

General parameters:

- **Elapsed Time greater than** - alerts will be calculated when in a given snap-time the duration for all queries exceeds 200 seconds.
- **History Days** - defining the days of the week that will be considered when examining performance problems.
- **Number of Days Back in History** - The number of historical days on the basis of which the system will test the performance of the current day.
- **Minimal History Days** - Specifies the minimum time after which trend-based alerts will be calculated

Statements Settings:

- **Number of Top Queries to check** - the number of top queries in individual snaps to be tested for performance problems, **Chosen by Elapsed Time / Cpu Time** - the choice according to which the statistics will be selected Elapsed Time queries or Cpu Time processor utilization time.
- **Number of Days Back in History** - The number of historical days based on which the system will analyze the performance of top queries on the current day.

Wait Events Settings:

- **Number of Top Waits to check** - used to handle waits calculated on the basis of the trend. The number of top waits depending on this parameter is taken into account for the calculation.
- **Number of Days Back in History** - how many days back, are taken into account for the calculation of history.

6.7.5.3 Alerts definition Tab

Defining alerts in the application has been divided into two stages:

- selection and configuration of appropriate CRITICAL / WARNING thresholds for a given type of alert,
- a rule definition based on configured alerts, and the attribution of the cause of the problem.

Website displays the information in columns:

- type of alert,
- description of the alert,
- availability,
- warning level,
- critical level.

The website presents only alerts that have been added to the configuration. If the alert has not been configured, please add it using the **[Add new alert]** button.

Mail settings | General settings | **Alerts definition** | Reasons & Problems definition | Events subscription

Refresh

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 Add new alert

Alert type	Alert description	Enabled	Level value WARNING	Level value CRITICAL
Online	Alert if database is not available	<input checked="" type="checkbox"/>		
Online	Total Waits	<input checked="" type="checkbox"/>	200 %	400 %
Online	Lock waits	<input checked="" type="checkbox"/>	200 %	400 %
Online	Latches	<input checked="" type="checkbox"/>	100 %	200 %
Online	Server CPU utilization	<input checked="" type="checkbox"/>	300 %	500 %
Load Trends	Elapsed Time	<input checked="" type="checkbox"/>	50 %	100 %
Load Trends	Wait Time	<input checked="" type="checkbox"/>	30 %	80 %
Load Trends	Lock Time	<input checked="" type="checkbox"/>	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: T14 Add new alert | Restore defaults

Alert type	Alert description	Enabled	Override	Level value WARNING	Level value CRITICAL
Online	Alert if database is not available	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Online	Total Waits	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200 %	400 %

Alerts can be configured for all databases or for a dedicated database. At any time, user can delete the previously configured alert by using the [Key] button and selecting an option "Delete", this will delete the given alert from the configured list.

The second option is to disable the alert by unmark the "Enabled" checkbox. This can also be done by pressing the [Key] button and selecting the Edit option.

Mail settings | General settings | **Alerts definition** | Reasons & Problems definition | Events subscription

Refresh

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 Add new alert

Alert type	Alert description	Enabled	Level value WARNING	Level value CRITICAL
Online	Alert if database is not available	<input checked="" type="checkbox"/>		
Online	Total Waits	<input checked="" type="checkbox"/>	200 %	400 %
Online	Lock waits	<input checked="" type="checkbox"/>	200 %	400 %

As part of the alert definition, user does not make the alert dependent on other alerts. Depending on the type of alert, threshold values are set in various ways.

Collecting data about problems in the application has been divided into 5 alert categories:

- **Online alerts** - calculated every 30 seconds,
- **Load Trends alerts** - calculated every 15 minutes based on general performance statistics,
- **Alerts type IO Stats** - calculated every 15 minutes on the basis of read / write statistics from / to disk devices,
- **Sql Query alerts** - calculated every 15 minutes based on statistics of top queries,
- **DB Size alerts** - calculated every 15 minutes based on space occupancy.

Alerts can be defined at the general level (for all bases) and at the level of individual databases. Two alarm thresholds can be defined for each alert:

- **WARNING** event - warning alert level
- **CRITICAL** event - high alert level - critical alert

For example: setting for the Load Trends category for the CPU Time alert.

Load Trends	Cpu Time		<input checked="" type="checkbox"/>	50 %	100 %
-------------	----------	--	-------------------------------------	------	-------

- If the CPU utilization of the server exceeds 50%
 - generate an alert at the warning level,
- If the CPU utilization of the server exceeds 100%
 - generate a critical alert
- In other cases, there is no alert.

Window below presents the main list of alerts:

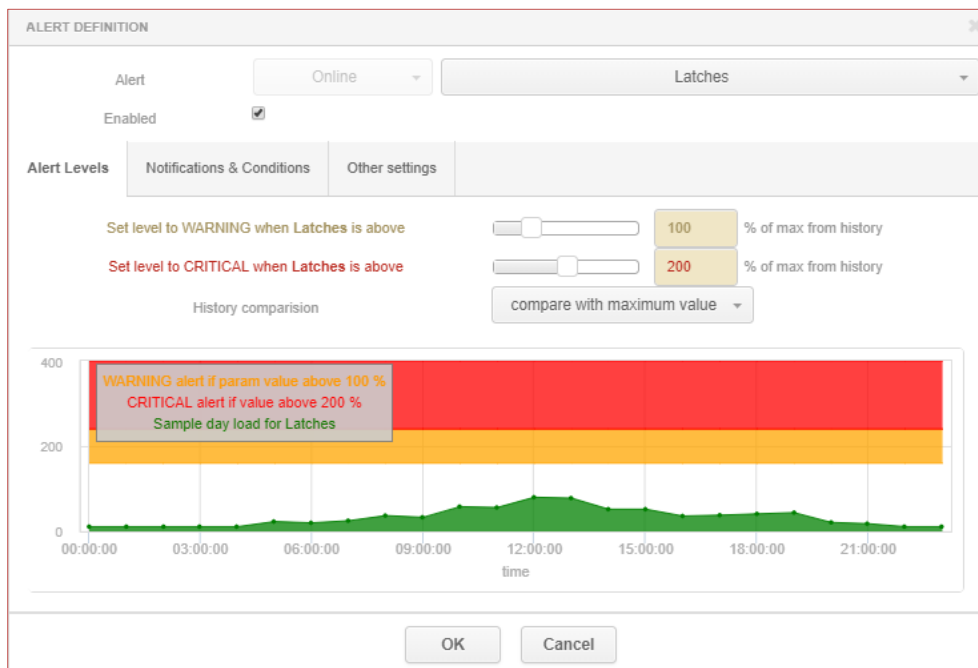
6.7.5.3.1 Online Alerts

The Online list includes the following alerts:

- **Active Sessions** - number of active sessions,
- **Number of Active Sessions with Elapsed Time longer than** - the number of active sessions with a duration longer than seconds
- **Lock waits** - lock type expectations,

- **Total Waits** - all expectations together,
- **Specific Wait** - an alert for a specific expectation,
- **Latches**
- **Server CPU utilization** - utilizing server CPU,
- **Custom alert calculated based on sql statement** - an alert calculated based on a freely arranged query,
- **Alert if database is not available.**

The example alert tab of the alert looks like this:



Please note that the field specifying the type of alert (Online, Load Trends, IO Stats, Sql Query) is changeable only when creating a new definition. When re-editing the alert, the field is in read-only mode. Depending on the rule chosen, the list of available and required fields to be completed is changed. For the alert: **Specific Wait** should be completed - the name of the wait for which the alert should react.

The screenshot shows the 'ALERT DEFINITION' dialog box for an alert named 'Specific wait'. The alert is set to 'Online' and is enabled. The 'Wait name' field is highlighted with a green box and contains the text '%read%'. Below this, the 'Alert Levels' tab is active, showing two thresholds: 'Set level to WARNING when Specific wait is above' at 4 s, and 'Set level to CRITICAL when Specific wait is above' at 10 s. The '4' and '10' values are also highlighted with a green box.

The following example will appear in the presented example:

an alert warning when the sum of expectations with a name containing reads exceeds at least 4 seconds / 1 second (a valid alert is not calculated here in percent).

critical alert when the sum of expectations with the name containing reads exceeds at least 10 seconds / 1 second (a valid alert is not calculated here in percent).

For the alert: **Custom alert calculated based on sql statement**, enter the query text.

IMPORTANT: the query must return a single-column record. The alert will occur when the value returned by the query exceeds the thresholds according to the given definition.

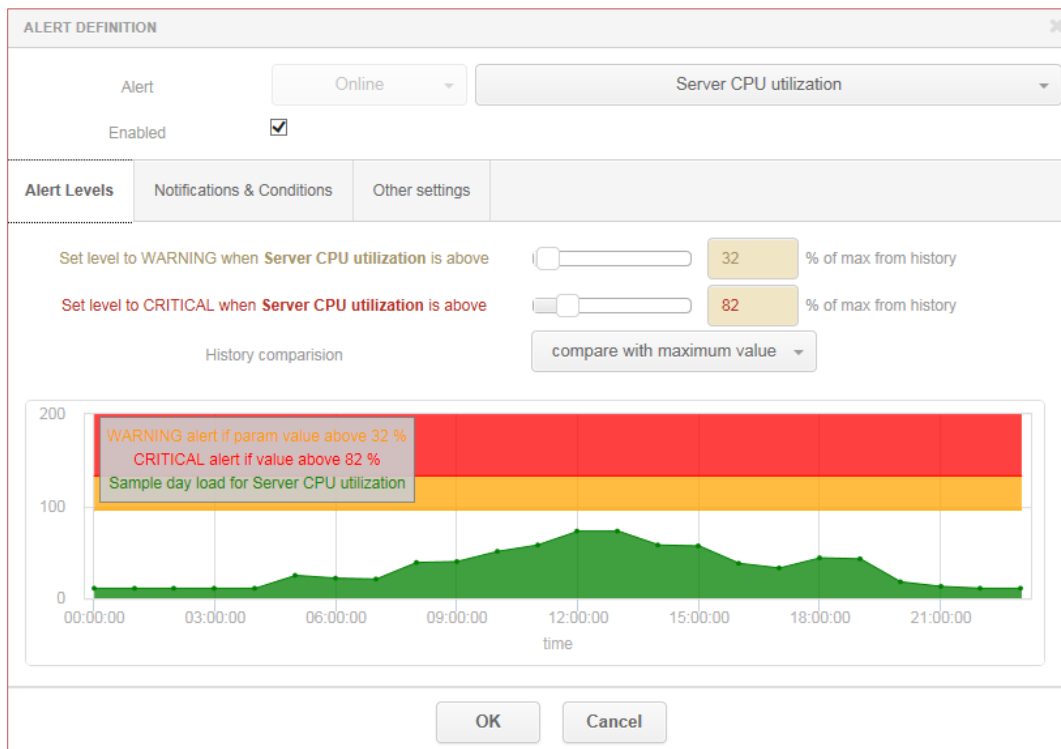
The following example will appear in the presented example:

- alert warning when the number of inactive sessions with an open transaction in the database exceeds at least 10 sessions
- critical alert when the number of inactive sessions with an open transaction in the database exceeds at least 40 sessions

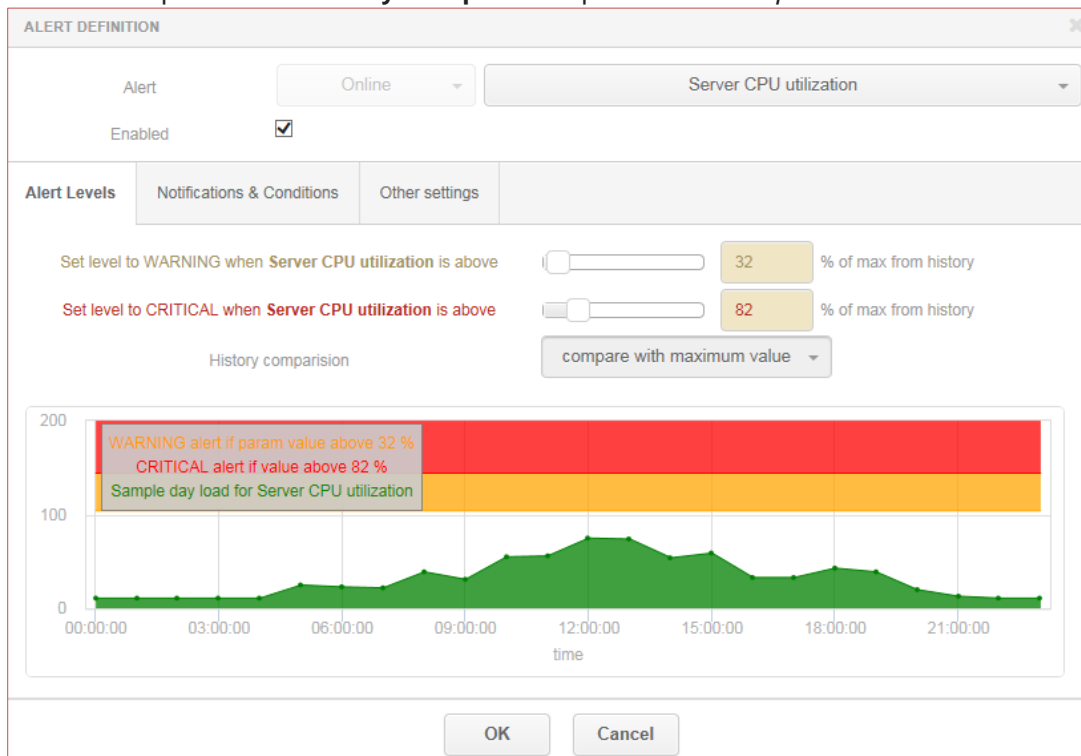
For the alert: Server CPU utilization, defines standard parameters, i.e.

- Alert thresholds **WARNING**, **CRITICAL**
- The way of calculating and reaction of the Alert on the History Comparison event (comparison of the performance of a given parameter with the history)
 - Compare to average value in similar time - the performance of a parameter is compared to the statistics history at similar times
 - Compare with maximum value - the performance of a parameter is compared with the maximum values that were present for a given statistic.

The following screen with the option of **History Comparison** set to *Compare to average value in similar time*:



And an example with the **History Comparison** option set to *Compare with maximum value*:



This slide will showcase:

Alert warning when the disposal of server processors will be 32% greater than the maximum historical value.

Critical alert when the utilization of server processors will be 82% greater than the maximum historical value.

In the alert edit tab, additional settings can be found in the Notification & Condition tab:

- **Mail Notification Interval** - how often to generate an email notification when an alert occurs
- **Number of snapshots to check** - the number of 30 seconds of snapshots in which there must be a "problem" for a given parameter. If a given statistic, e.g. Total Waits - stays at a high level and exceeds the alert threshold by X snapshots, then the system will generate an alert
- **Use Low Constant Value** - the minimum value that must be met first. According to the example screen below - within the dashboard snapshot (started in a 30-second cycle) the value of all wait-time must be at least 30 seconds.
- **Use High Constant Value** - the value at which the alert will always be generated, even if the WARNING, CRITICAL alert thresholds are not met.

Alert Levels	Notifications & Conditions	Other settings
	Alert Calculation Interval	once per 30 seconds
	Mailing Notification Interval	once per 5 minutes
Filter conditions		
	Use Low Constant Value	30 s. Every alert with value below entered will be skipped
	Use High Constant Value	60 s. Every alert with value above entered will be shown
Snapshot conditions		
	Number of snapshot to check	5 in which property must exceed alert level value

6.7.5.3.2 Load Trends, I/O Stats Alerts

The Load Trends, I / O Stats rules refer to performance indicators available on website (functionalities) with the same names.

In the case of Load Trends, the system allows users to alter the following indicators:

- Elapsed Time
- Cpu Time
- Sorts
- Fetches
- Executions
- Disk reads
- Buffer gets
- Rows Process
- Latches
- Waits
- Locks
- Sessions
- Active sessions
- Wait time
- Wait Event Time

For IO Stats, users have the following indicators:

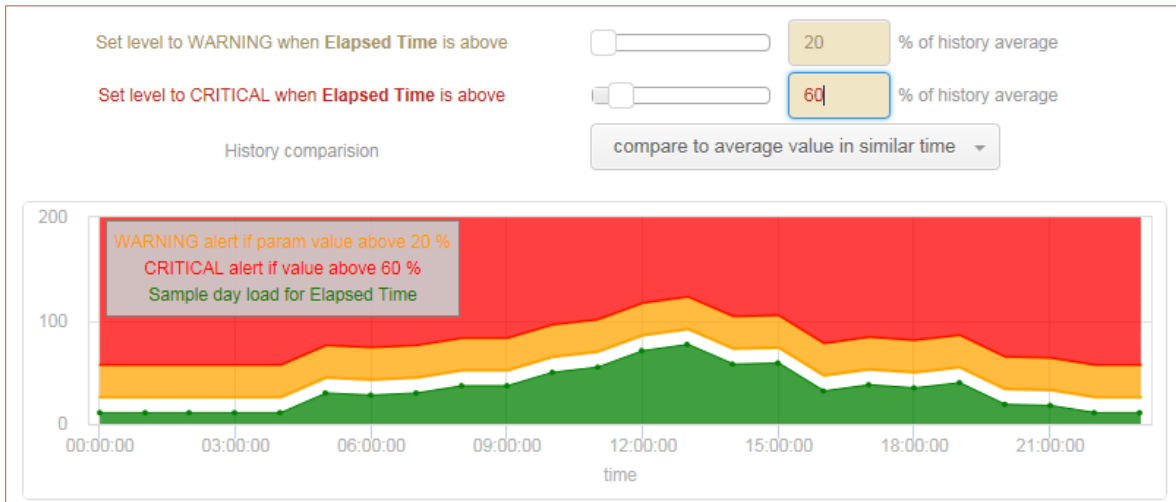
- Reads
- Writes
- Block Reads
- Block Writes
- Read Time
- Write Time
- Single Block Read Time
- Single Block Write Time

The edit tab of such alert looks like the below:

In the form, the user specifies the following:

- Type of alert (according to the indicators given above)
- Is enabled
- Own name - **Other settings** tab
- Message format - **Other settings** tab
- E-mail settings - spam protection in case of an ongoing alert - Notification & Conditions tab
- **When and with what threshold an alert will occur:**
 - The rule is calculated as a percentage.
 - The alert will occur when the given alert threshold is exceeded by X% in relation to the average over the past period.
 - In the **Filter condition** section we have additional filter settings, i.e:
 - **Use Low Constant Value** - eg, alert when Elapsed Time will deteriorate from X% in relation to the average, but in a situation where Elapsed Time is greater than 500 seconds.
 - **Use High Constant Value** - as above

Below are some examples of definitions for the **Elapsed Time** parameter - with the option of **History Comparison** set to Compare to average value in similar time:



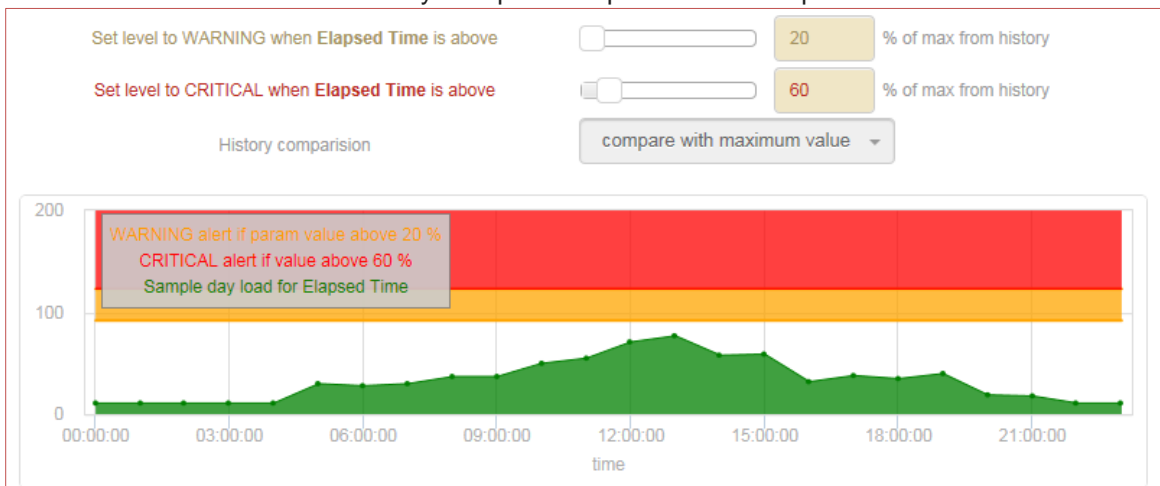
According to the above, the other load occurs during business hours and outside business hours. For example:

The duration of all queries, i.e. Elapsed Time at 08:00, is historically 1000 s in a 15-minute snapshots.

The duration of all queries, i.e. Elapsed Time at 12:00, is historically 5000 seconds in a 15-minute snapshots.

Alert warning type WARNING for a defined threshold $\geq 20\%$ will occur at 08:00, when the duration of all queries exceeds 1200 seconds, while around 12:00, when Elapsed time exceeds 6000 seconds.

For the second case with the History Comparison option set to Compare with maximum value:



In this example:

WARNING for the defined threshold $\geq 20\%$ will occur only if the duration of all queries exceeds 6000 seconds (reference to the maximum value of the day) regardless of the time of day.

6.7.5.3.3 Sql Query Alerts

SQL Query rules apply to performance indicators available for SQL queries and contain a similar list as for Load Trends.

For SQL queries, the system allows users to alter the following indicators:

- Buffer gets
- CPU Time
- Disk reads
- Elapsed Time
- Elapsed Time Per 1 Exec
- Execution
- Fetches
- Rows Process
- Sorts
- Wait Time

In addition, the list of rules also includes:

- New statement Elapsed Time
- New statement CPU Time

The SQL Query Alert Definition tab looks like this:

In the form, the user specifies similar parameters as in the alert definition for Load Trends statistics, IO Stats. In addition, user can indicate whether the alert reacts only when the execution plan is changed - the **Show Plan Changes Only flag (assuming that the indicator has deteriorated in relation to the history)**.

For alerts with the **New Statement** prefix, the thresholds are determined at the level of the share in the database load.

The application allows the dependence of an alert instance on the general trend (for the entire database) for a given statistic in the snap. This option is only available for SQL Query type alerts. For the configuration shown in the picture below, this means for the SQL Query Rows processed type alarm:

- the alarm will be skipped if the value of Rows processed for a given snap for a specific Query Hash is below 10 and if the number of returned rows processed for a given query is less than 15% of all returned rows for queries (the number depends on the Number of Top Queries to check). Additionally, the condition of exceeding the WARNING / CRITICAL alarm threshold must be met.
- the alarm will occur if the value of Rows processed for the given snap in the query is above 25%. The alarm will occur even if the alarm threshold has not been exceeded (then WARNING will occur with the Above max constant comment ...).

ALERT DEFINITION

Alert: Sql Query, Rows processed

Enabled:

Alert Calculation Interval: once per 15 minutes

Filter conditions:

Use Low Constant Value: 10 (Every alert with value below entered will be skipped)

Use High Constant Value: 25 (Every alert with value above entered will be shown)

Query impact on load is above: 15 %

OK Cancel

6.7.5.3.4 DB Size Alerts

The DB Size list has an alert for the lack of free space. The user has the option to specify:

- For which / what space the alert should be calculated (**Tablespace** setting),
- What spaces are an exception and are not considered (**Excluded tablespaces** setting),
- The method of calculation: percentage or units,
- Calculations in relation to the current space size or possible maximum value.

The DB Size Alert Definition tab looks like this:

ALERT DEFINITION

Alert: DB Size, Free tablespace size

Enabled:

Tablespace name: A_TAB%;DBPLUS
Use % character to run alert with like condition, use empty to run for each tablespace

Excluded tablespaces:
Use % character to run alert with like condition, use ; to separate names

Alert Levels: Notifications & Conditions, Other settings

Use percentage:

Check free space to MAXSIZE:

Set level to WARNING when Free tablespace size is below: 10 % space

Set level to CRITICAL when Free tablespace size is below: 2 % space

OK Cancel

According to the above definition, the alert applies to spaces with names beginning with A_TAB, and for DBPLUS spaces. If the free space drops below 10%, an alert will be generated.

Below is another example:

According to which the alert will occur when the free space is below 1000 MB for any space, omitting the spaces beginning with SYS, UNDO and USER spaces.

Example with conditional alerts:

DB Size	Free tablespace size - [PROD%] excluding (SYS%;USERS%.)		<input checked="" type="checkbox"/>	10 %	5 %
DB Size	Free tablespace size - [PROD%.] excluding (SYS%;USERS%.)		<input checked="" type="checkbox"/>	1000 MB	400 MB

Alert selected in green has been added as a conditional alert - it means that the alert about the lack of free space will be generated only when the free space size:

- drops below 10% and at the same time below 1000 MB,
- for spaces starting with the PROD name.

6.7.5.3.5 Alert settings at the database level

The list of alerts can be set for each base independently. By default, alerts are inherited from general settings. If any alert parameter is changed then the information appears in the Override column about overriding this rule.

As the example below:

ALERTS CONFIGURATION					Add new alert
Alert type	Alert description		Enabled	Level value WARNING	Level value CRITICAL
Online	Alert if database is not available		<input checked="" type="checkbox"/>		
Online	Number of active sessions with Elapsed time longer than 0,03 seconds		<input checked="" type="checkbox"/>	2	5

Lightbulb icon: 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		XE (1 alert/s overwritten)		Add new alert	Restore defaults
Alert type	Alert description	Enabled	Override	Level value WARNING	Level value CRITICAL
Online	Alert if database is not available	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
Online	Number of active sessions with Elapsed time longer than 0,03 seconds	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2	5

The system will generate an unavailability alert for all databases except the XE database. At the XE Alert level, *Alert if database is not available*, has been disabled (Enabled = false).

6.7.5.4 Reasons and Problems definition Tab

The next stage of alarm configuration consists of, assigning rules and defining the dedicated cause of the problem. Screen below shows an example of a list of alarms defined by default by DBPLUS analysts. Definitions can be assigned at a general level to all databases or create dedicated definitions for selected databases.

Mail settings | General settings | Alerts definition | **Reasons & Problems definition** | Events subscription

Refresh

Lightbulb icon: 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	Reason/Problem description	Enabled	Override	Rule preview
Trends	Performance problem for specified SQL Statements cause increase Executions and Disk Reads	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Trends Elapsed Time AND ((SQLQuery.Elapsed Time AND SQLQuery.Elapsed Time per 1 exec AND SQLQuery.Execution AND SQLQuery.Disk reads) OR (SQLQuery.Elapsed Time AND SQLQuery.Execution AND SQLQuery.E...
Trends	Performance problem for specified SQL Statements cause Disk reads increase	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Trends Elapsed Time AND ((SQLQuery.Elapsed Time AND SQLQuery.Elapsed Time per 1 exec AND SQLQuery.Disk reads AND NOT SQLQuery.Execution) OR (SQLQuery.Elapsed Time AND SQLQuery.Disk reads AND NOT S...
Trends	Performance problem for specified SQL Statements	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Trends Elapsed Time AND ((SQLQuery.Elapsed Time AND SQLQuery.Elapsed Time per 1 exec) OR SQLQuery.Elapsed Time) AND NOT SQLQuery.Execution
Trends	Data writes time problem caused by slow I/O response	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Trends Elapsed Time AND (IO.Single Block Write time OR IO Write time) AND NOT IO.Disk writes AND (Trends.Wait Event Time - [log file sync] OR Trends.Wait Event Time - [job file parallel write] OR Trends.Wait Event Time - [bu...
Trends	Problem - wait. log file parallel write	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Trends.Wait Event Time - [log file parallel write] AND (IO.Block writes OR IO.Disk writes)
Trends	Data reads time problem caused by slow I/O response	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Trends Elapsed Time AND (IO.Single Block Read time AND IO Read time) AND NOT IO.Disk reads AND NOT SQLQuery.Disk reads
Trends	Performance problem for specified SQL Statements cause increase Executions	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Trends Elapsed Time AND ((SQLQuery.Elapsed Time AND SQLQuery.Elapsed Time per 1 exec AND SQLQuery.Execution AND NOT SQLQuery.Disk reads) OR (SQLQuery.Elapsed Time AND SQLQuery.Execution AND NOT S...
Trends	Performance problem for specified SQL Statements cause new query	<input checked="" type="checkbox"/>	<input type="checkbox"/>	(Trends.Elapsed Time AND SQLQuery.New Statement Elapsed Time) OR (Trends.Cpu Time AND SQLQuery.New Statement Cpu Time)
Trends	Database performance degradation	<input checked="" type="checkbox"/>	<input type="checkbox"/>	(Trends.Elapsed Time AND Trends.Wait Time AND Trends.Execution) AND (NOT SQLQuery.New Statement Cpu Time OR NOT SQLQuery.New Statement Elapsed Time) AND NOT IO.Single Block Read time AND SQLQuery.El...
Trends	Increase of query processing time caused by slow I/O response	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Trends Elapsed Time AND (IO.Single Block Read time OR IO.Single Block Write time)
Trends	Increase of query processing time caused by locks	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Trends Elapsed Time AND Trends.Lock Time

add new definition

Lightbulb icon: List of performance problems on the instance level which are specific for particular database. Below settings overwrite main configuration. Those lines which are marked in light gray color, are inherited from main configuration

INSTANCE PROBLEM CONFIGURATION - PLEASE SELECT A DATABASE

T5 testovna

Type	Reason/Problem description	Enabled	Override	Rule preview
Trends	Performance problem for specified SQL Statements cause increase Executions and Disk Reads	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Trends Elapsed Time AND ((SQLQuery.Elapsed Time AND SQLQuery.Elapsed Time per 1 exec AND SQLQuery.Execution AND SQLQuery.Disk reads) OR (SQLQuery.Elapsed Time AND SQLQuery.Execution AND S...
Trends	Performance problem for specified SQL Statements cause Disk reads increase	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Trends Elapsed Time AND ((SQLQuery.Elapsed Time AND SQLQuery.Elapsed Time per 1 exec AND SQLQuery.Disk reads AND NOT SQLQuery.Execution) OR (SQLQuery.Elapsed Time AND SQLQuery.Disk reads A...

To add a new rule, first define the reason for the problem (Reason description) for which the rule will be defined. Next, choose the type of calculation (Calculation type) - based on the trend or online and Reason class.

REASON DEFINITION

Reason description: Network problem not caused by I/O disk storage issues

Calculation Type: Based on Trends

Reason Class: I/O

Enabled:

Rules & Formulas | Notifications & Conditions

AND OR Add rule Add group

Trends:Wait Event Time - [TCP Socket%] Delete

AND OR Add rule Add group Delete

AND OR Add rule Add group Delete

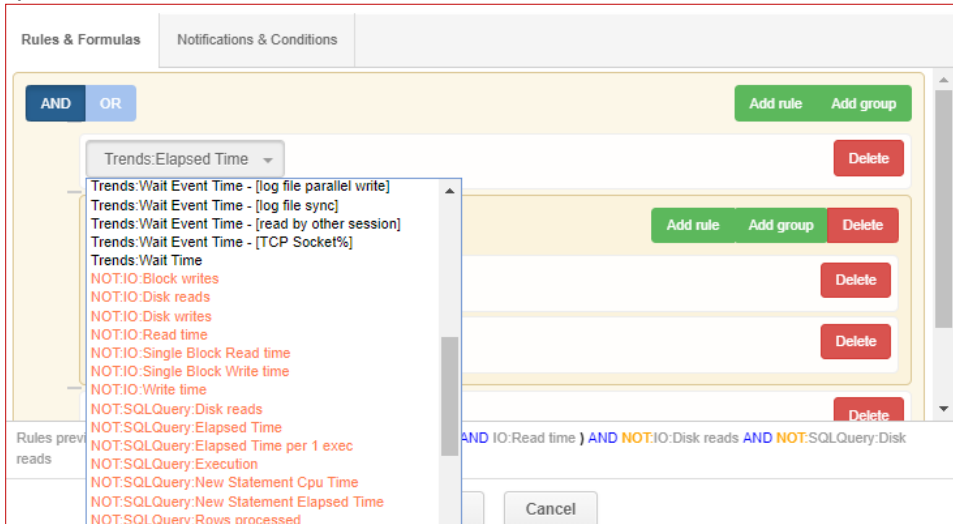
NOT IO:Disk reads Delete

NOT IO:Single Block Read time Delete

Rules preview: Trends:Wait Event Time - [TCP Socket%] AND ((NOT IO:Disk reads AND NOT IO:Single Block Read time) OR (NOT IO:Disk writes AND NOT IO:Single Block Write time))

OK Cancel

The most important element of the configuration is to create the cause of the problem and then define the appropriate rules based on alerts. To add a configuration, from the previously defined alerts (Alerts definition tab), create a rule using groups (Add group), AND, OR operators. In some cases, it is necessary to use negation, they are presented in the list of alerts marked in red and start with the NOT operator.



After defining the rule, correctly selecting the operators and completing all added alarms, the rule will be displayed below.

Rules preview: (Trends:Elapsed Time AND Trends:Wait Time AND Trends:Execution) AND (NOT:SQLQuery:New Statement Cpu Time OR NOT:SQLQuery:New Statement Elapsed Time) AND NOT:IO:Single Block Read time AND SQLQuery:Elapsed Time

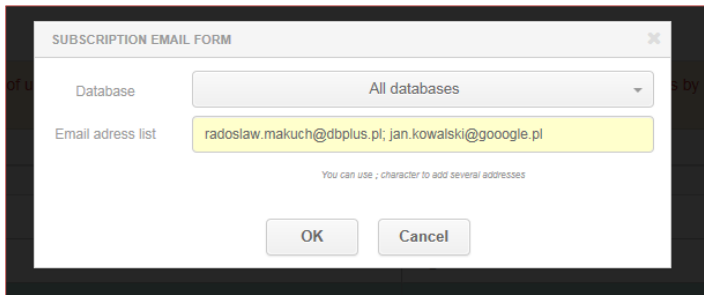
6.7.5.5 Events subscription Tab

In the last tab of the module user has the ability to manage the list of recipients, i.e. people who will receive alert messages.

EMAIL SUBSCRIPTION LIST			Add new email address	
Database	Email address		Edit	Delete
RS	mariusz.zacharewicz@dbplus.pl, dariusz.markowski@dbplus.pl, artur.boguszewski@dbplus.pl,		Edit	Delete
BAZY	mariusz.zacharewicz@dbplus.pl, radoslaw.makuch@dbplus.pl, artur.boguszewski@dbplus.pl		Edit	Delete
FK	mariusz.zacharewicz@dbplus.pl, dariusz.markowski@dbplus.pl, artur.boguszewski@dbplus.pl		Edit	Delete
CAT	mariusz.zacharewicz@dbplus.pl, radoslaw.makuch@dbplus.pl, artur.boguszewski@dbplus.pl		Edit	Delete
All databases	artur.boguszewski@dbplus.pl, radoslaw.makuch@dbplus.pl		Edit	Delete

The list of subscribers can be:

- a single email address or multiple addresses separated by a separator;
- assigned recipient's email address to all or selected databases.



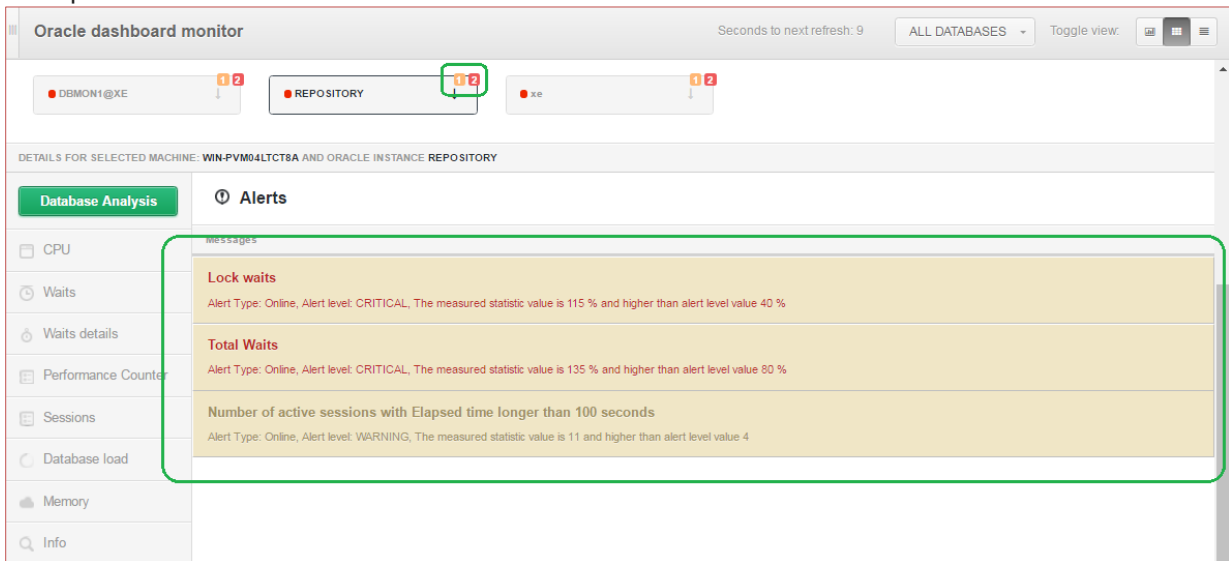
6.7.5.6 Visibility of alerts

Alerts are visible from the Anomaly Monitor menu and also from the:

- **Dashboard Level:**
 - the base icon contains information about the number of alert and critical alerts
 - after selecting a given database in the **Alerts** and **Database Load** tab
- after clicking **[Database Analysis]** on the **Database Load** graph
 - if any Alert have occurred on the Elapsed Time line, relevant information is displayed about their number
 - after clicking on a given time point (snapshot) - a list of alerts is displayed

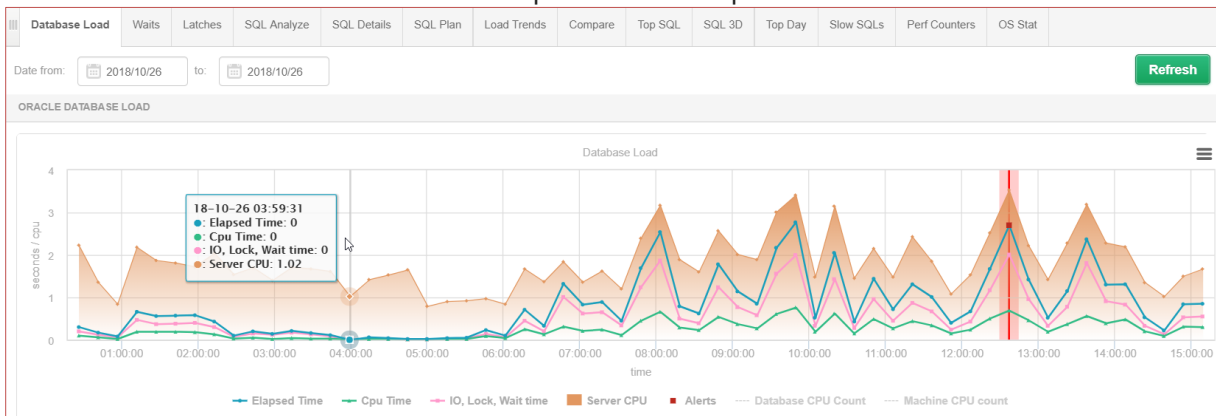
6.7.5.6.1 Dashboard

Example screen from dashboard with selected base with alerts:



6.7.5.6.2 Database load

Base load chart - information about alerts as points on the Elapsed Time chart line:



After selecting the snap for which the alert occurred, user can view information about the reason for the alert. For example, the application showed the cause of the problem related to query hash: **782769954**, caused by an increase in the number of queries performed and an increase in reading data from the disk. As a result, it affected the overall performance level of the entire database.

Sql Statements	Waits	Alerts
SNAPSHOT OF ALERTS GENERATED WITHIN 15 MINUTES AT 2018-10-26 12:36:58		
Logdate	Reason name	
2018/10/26 12:37:00	Performance problem for specified SQL Statements cause increase Executions and Disk Reads.	
	Execution	Alert Type: Sql Query, The measured statistic value is 163 % higher than allowed maximum , Statement hash value: 782769954 + , Statistics: Execution, Last value: 51 , History value: 19.4
	Disk reads	Alert Type: Sql Query, The measured statistic value is 161 % higher than allowed maximum , Statement hash value: 782769954 + , Statistics: Disk reads, Last value: 1693449 , History value: 649503
	Elapsed Time	Alert Type: Sql Query, The measured statistic value is 2.6 times higher than allowed maximum , Statement hash value: 782769954 + , Statistics: Elapsed Time, Last value: 597.7 s, History value: 164.2 s
	Elapsed Time	Alert Type: Load Trends, The measured statistic value is 70 % higher than average , Last value: 2458 s, Reference history value: 1445 s
2018/10/26 12:37:00	Performance problem for specified SQL Statements	
	Elapsed Time	Alert Type: Sql Query, The measured statistic value is 111 % higher than allowed maximum , Statement hash value: 1592221868 + , Statistics: Elapsed Time, Last value: 534.3 s, History value: 253.6 s
	Elapsed Time	Alert Type: Load Trends, The measured statistic value is 70 % higher than average , Last value: 2458 s, Reference history value: 1445 s

6.7.6 Outages setting

After entering the tab, we can view information about scheduled monitoring shutdowns. On the website, only the exclusions for the current day as well as those scheduled in the future are visible by default. The information can be viewed for all databases as well as for a specific database.

To add a new entry, click the [Add new outage] button.

Scheduled outages						
Date from:	2018/11/26	to:		Filter by database:	All databases	Refresh
DATABASES OUTAGES SCHEDULE						
Outages information and its schedules are refreshed within 15 minutes.						
Database	Enabled	Period	Outage days	Outage hours	Reason	
FK08T	<input checked="" type="checkbox"/>	From 2018-11-24 to 2018-11-28	Every Sat, Sun	between 17:00 - 17:20	Outage module testing	
FK08T	<input checked="" type="checkbox"/>	From 2018-11-26 to 2018-11-26	Every Mon	between 14:40 - 15:00	testowe wyłączenie monitoringu	

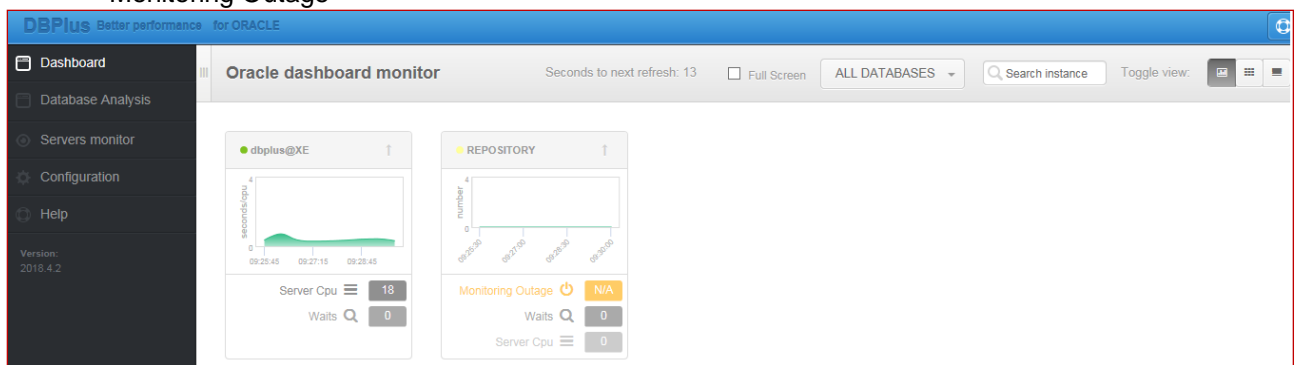
After clicking, we choose which database should be turned off, and then choose whether the shutdown should be:

- single or recurring,
- last one or many days,
- is expected to occur on a specific day of the week.

After selection, we add information about the reason for the exclusion and accept configurations. After the correctly entered configuration, the new entry will be visible in the table. It must be remembered that the information about the shutdown will appear on the chart when the new / next snapshot is generated.

Information about monitoring service being shut down is visible on the Dashboard screen:

- in the case of Television mode - a yellow mark next to the database and a description of "Monitoring Outage"



- in the case of Icons view

In this view, the base is also marked in yellow, which means a break in monitoring. As well as the base in which monitoring has been disabled, it is not included in the number of active databases.

Oracle dashboard monitor

SUMMARY FOR ALL DATABASES

1 Servers 2 Databases 1 Active Databases

PHYSICAL SERVERS ● Performing well ● Warning ● Overloaded ● Not available

● DESKTOP-HR1BE66 ↓

ORACLE INSTANCES ● Performing well ● Warning ● Overloaded ● Not available ● Monitoring in outage Search instance

● XE ↓ ● XE_2 ↓

➤ in case of Grid view

Oracle dashboard monitor Seconds to next refresh: 7 ALL DATABASES Toggle view

SUMMARY FOR ALL DATABASES

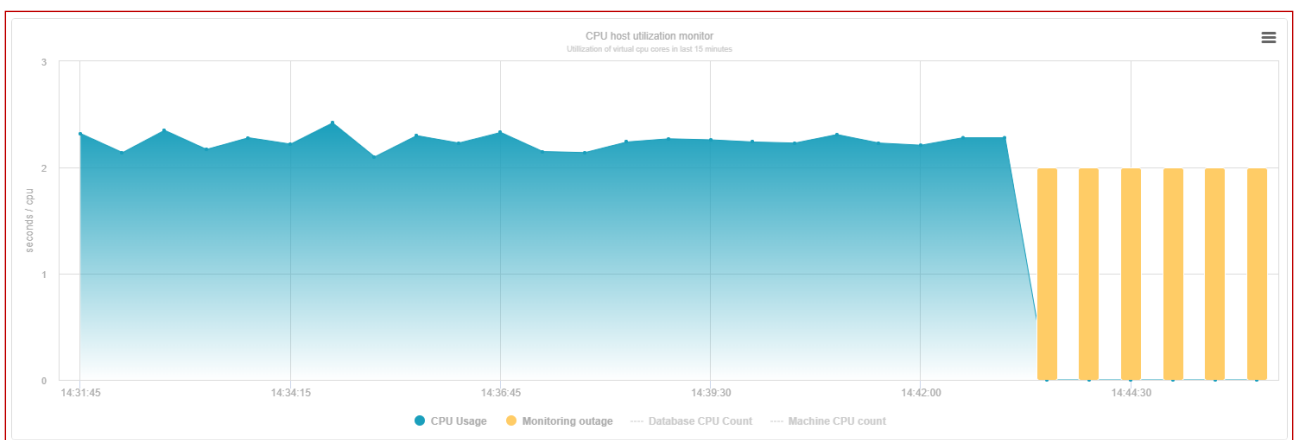
1 Servers 2 Databases 1 Active Databases

Summary of Waits ● Summary of IO Waits ● Summary of Lock Waits ●

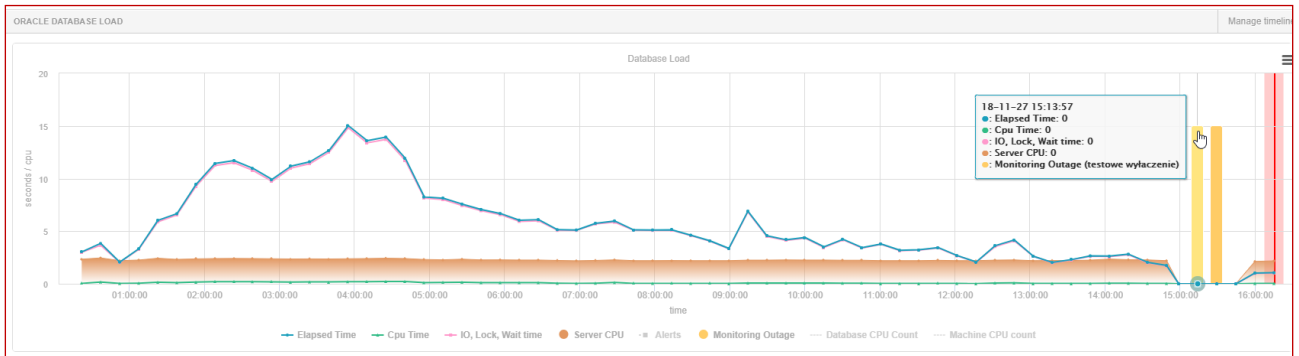
ORACLE INSTANCES Search instance

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]
NOT SPECIFIED	DESKTOP-HR1BE66	XE_2	☑	6 ●	0.00 ●	0.00 ●	0.00 ●	0.00 ●	0	1	0	5.9
NOT SPECIFIED	DESKTOP-HR1BE66	XE	☐	0 ●	0.00 ●	0.00 ●	0.00 ●	0.00 ●	0	0	0	5.9

Information about turning off the monitoring is visible in the graph of current CPU utilization for a given database in the form of yellow vertical bars.



Information about disabling is also visible on the Database Load chart. In case the database is excluded from monitoring, yellow vertical bars appear in the graph. At the moment of disabling, information on statistics is not collected.



6.7.7 Time line settings

After entering the tab, we can view information about the upcoming scheduled work. On the website, only works for the current day as well as those scheduled in the future are visible by default. The information can be viewed for all databases as well as for a specific database. The functionality is created to present information about scheduled work that may affect the performance of the database. To add a new entry, click the [Add new work or tag] button.

Scheduled works & timeline tags				
Date from: 2018/11/26 to: Filter by database: All databases Refresh				
PLANNED WORKS & TIMELINE TAGS SCHEDULE Add new work or tag				
Planned works, timeline tags are visible on Database load, Load Trends charts for specified databases				
Database	Timeline	Work title	Details & Description	
FK08T	2018-11-26 10:56	Wgranie poprawek	Praca testowa	
FK08T	2018-11-26 13:20	wgranie poprawek 2	Praca testowa	

After clicking, we choose for which database the planned work should be registered, and then we choose whether the shutdown should be:

- single or long period

After selecting the range, we add information in the "tag title" field (visible later in the chart), and add detailed information about the planned work, then we accept the configuration. After the correctly entered configuration, the new entry will be visible in the table. It must be remembered that the information about the planned work will appear on the chart when the new / next snapshot is generated.

WORK / TIMELINE TAG DEFINITION

Database: No database selected

Timeline setting

Use period range:

Date: 2018/11/26 15:14

Work / tag description

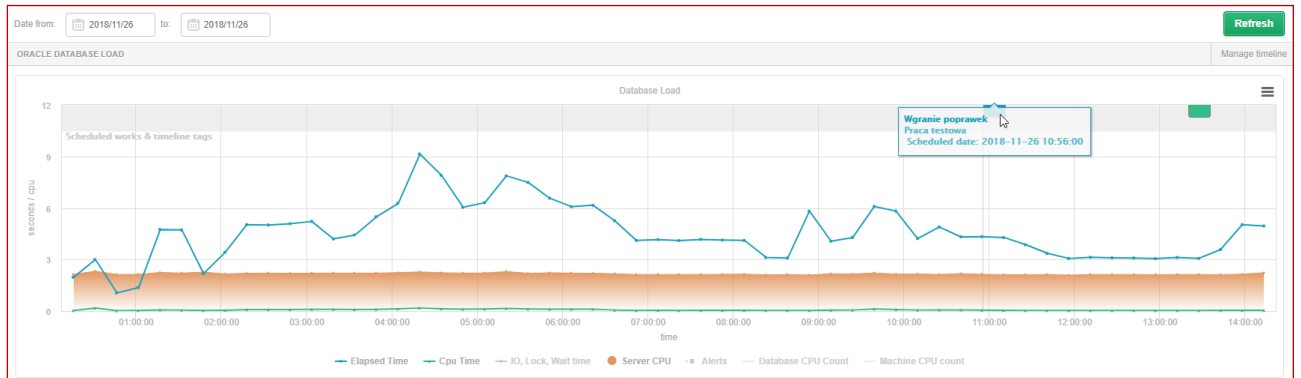
Enter tag title

Enter work/tag detail information

OK Cancel

Information about scheduled work is shown in the Database Load chart in the form of points (single events) or bars in the case of long-term work. After hovering over the point / bar, the information about the scope and the topic of the planned work will be displayed. If work is planned in the future, information about the work will be visible as a point on the right side of the chart.

In addition, from the Database Load level, we can manage deployments by clicking on the [Manage timeline] button.



6.8 Additional functionalities

6.8.1 RAC support

The DBPLUS Performance Monitor application supports database in the RAC architecture. In the case of databases in the RAC architecture, the system automatically detects them and enriches the most important screens with the option of switching between databases. The list where such an option has been added :

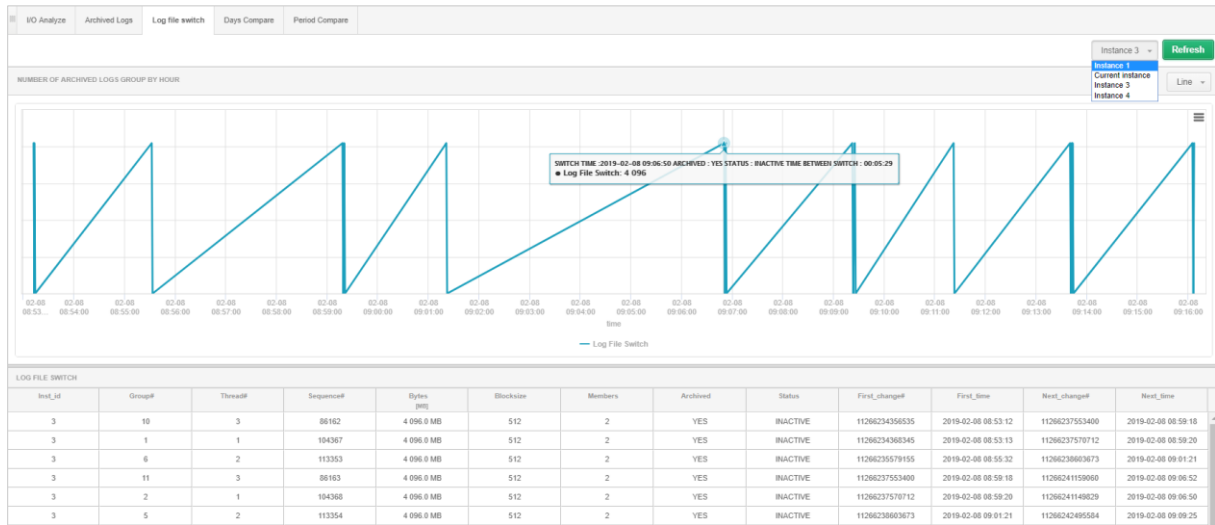
- Online Session,

The screenshot shows the 'Sessions' tab in the DBPLUS Performance Monitor. At the top, there are tabs for 'Sessions', 'Sort usage sessions', 'Undo usage sessions', 'Sessions history', and 'Session / Sort / Undo history'. Below these are search filters for 'Active sessions', 'Users only', and 'Min elapsed time'. A table lists active sessions with columns for 'Inst Id', 'Logon time', 'Sid', 'Serial', 'Hash Value', 'Username', 'Status', 'Elapsed Time', 'Schema', 'OS user', 'Process (server)', 'Process (client)', 'Machine', 'Module', 'Wait', and 'Blocking session'. Below the table, there is a 'SQL' section with 'Operation progress', 'Statistics', and 'Session Waits'. The 'SQL' section shows a complex SQL query for inserting data into a table. Below the SQL, there is an 'EXPLAIN PLAN' section showing the execution plan for the query.

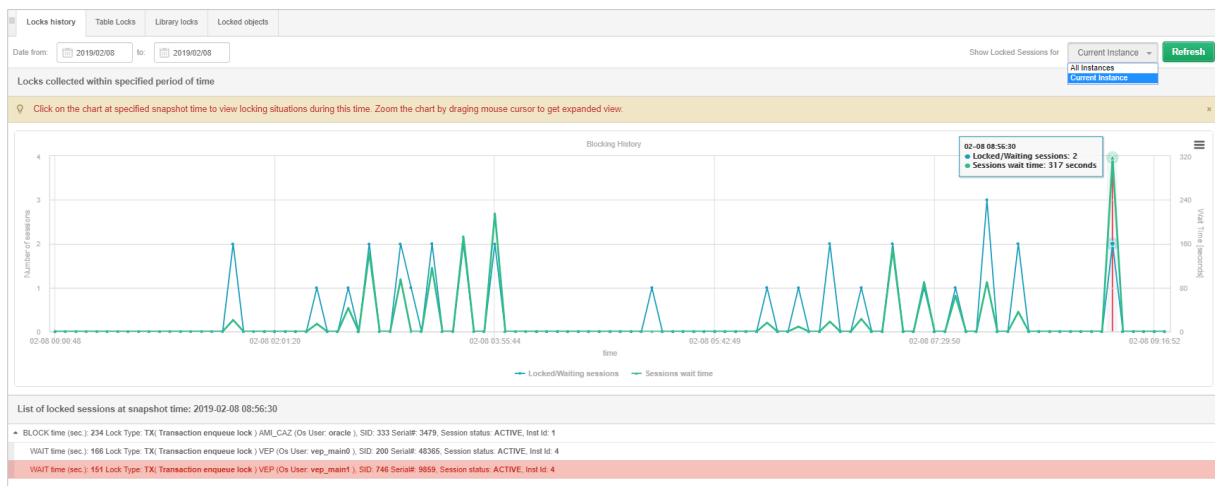
- Sort Usage Sessions,
- Undo Usage sessions,
- Sessions history,

The screenshot shows the 'Performance Waits' tab in the DBPLUS Performance Monitor. At the top, there are tabs for 'Sessions', 'Sort usage sessions', 'Undo usage sessions', 'Sessions history', and 'Session / Sort / Undo history'. Below these are search filters for 'From', 'To', 'Using Hash Value/Sql Id', 'Instance 1', 'Username', and 'Sid'. A 'Performance Waits' section shows a list of wait events with a search bar and a 'Waits selected to filtering' list. Below this is a table of sessions with columns for 'Logdate', 'Inst Id', 'Sid', 'Serial#', 'Hash Value', 'User', 'Status', 'Elapsed Time', 'Schema', 'OS User', 'Machine', 'Program', 'Tablespace', 'Wait', and 'Space Used'. A dropdown menu is open, showing 'All instances' and a list of instance IDs (Instance 1, Instance 2, Instance 3, Instance 4).

- Log file switch

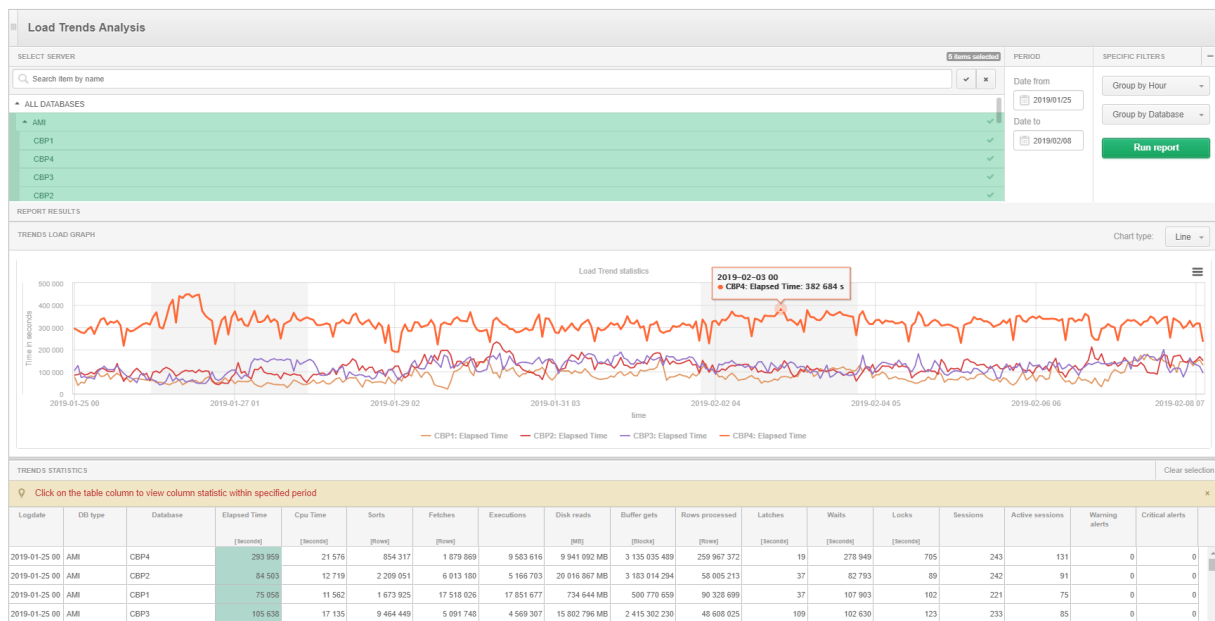


- Locks history
- Table Locks/Locked objects



Load trends analysis (main menu Reports>Load trends).

On this screen, the user can select and compare all databases included in the RAC for all performance statistics.



6.8.2 Standby DataGuard support

In case of using Standby DataGuard, we have listed two possible scenarios:

Scenario 1. Monitoring only the active database.

When used to connect databases using the TNS settings, the overvoltage between PROD <> STANDBY requires from the Configuration Wizard level change of the settings to the database and manual restart of the Dbpluscatcher service.

Scenario 2. Monitoring of all databases.

In this scenario, all active and standby databases are included in the monitoring and visible from the Performance Monitor application. In the case of the Standby database, from the level of the application, we will receive information about the lack of possibility to connect to the database. When switching to Active Standby the application will establish the connection automatically and will collect data from the database without the need for any additional action on the part of the administrator.

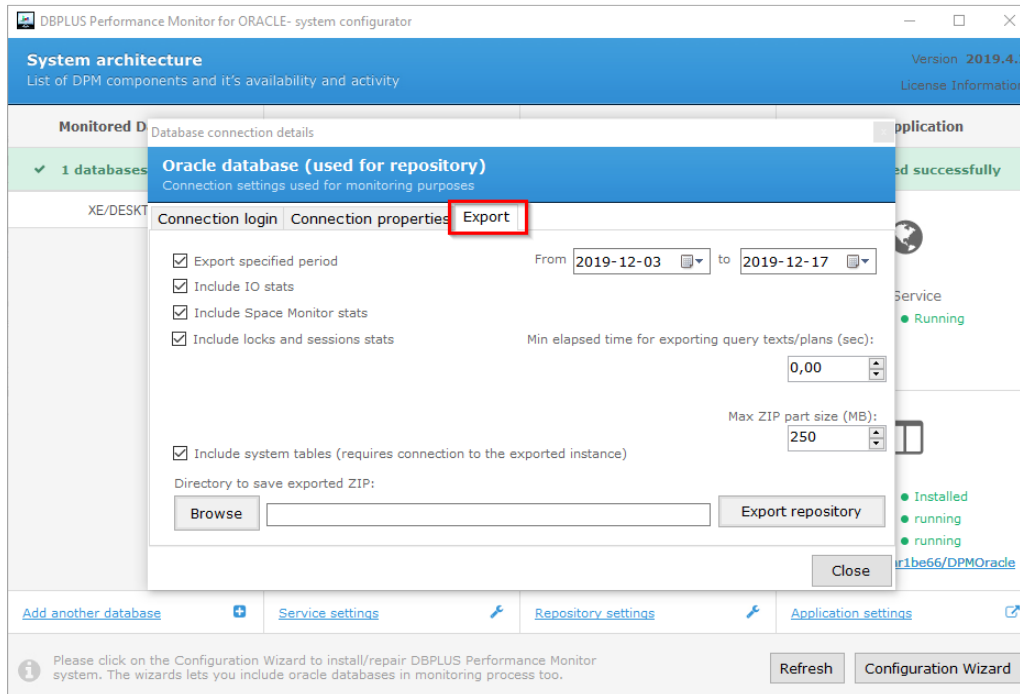
6.8.3 Export / Import statistics of monitored databases

In the latest version we have added the ability to export / import data collected by DBPLUS Performance Monitor. The user makes both export and import within the same platform, in this case Oracle. In the current version, the mechanism allows to export the entire database (without date range) as well as selected periods and monitoring modules. Import requires a previously configured repository database, and the result is the addition imported database as a new Oracle database to the current repository.

The imported database is added to the monitoring by default. Collecting data through the "Scheduled Outages" mechanism is blocked (performance statistics are not collected). Enabling statistics collection of an imported database is possible by changing the settings in the Configuration> Scheduled Outages menu for the appropriate Oracle database.

Oracle database export

The export can be made from the level of DBPLUS Configuration Wizard. After starting the program, the user goes to the details screen of the instance whose data user wants to export by clicking the button ("cog") next to the given database. Then it goes to the Export tab, like below:



The next step is to select the export option. The user can configure:

- date range for which statistics going to be exported (if selected),
- minimum duration of queries (queries below this value will not be exported),
- maximum export file size (another file will be created above this size).

Additionally, the user can choose which data will be exported by select options:

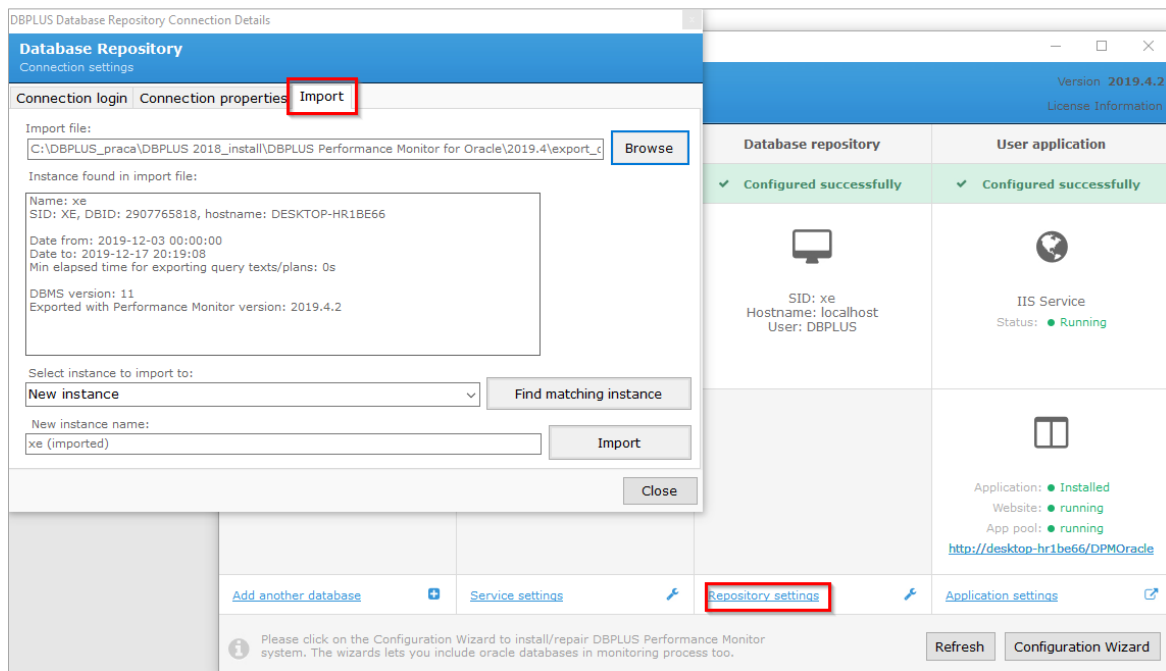
- Include IO stats
- Include Space Monitor Stats
- Include locks and sessions stats
- Include system tables.

If the "Include system tables" option is selected, online access to the exported database is required to download the data.

After selecting the data for export, indicate the directory where the data file will be saved. A log file will be created in the export directory with information on what data has been exported, and saved data with the exported file or files (depending on the size of the export).

Oracle database import

The import process can also be performed from the Dbplus Configuration Wizard. The imported database can be added as a new instance as well as part of an instance previously configured for monitoring. The import is started by clicking the "Repository settings" link from the Configuration Wizard and then user goes to the Import tab as shown below:



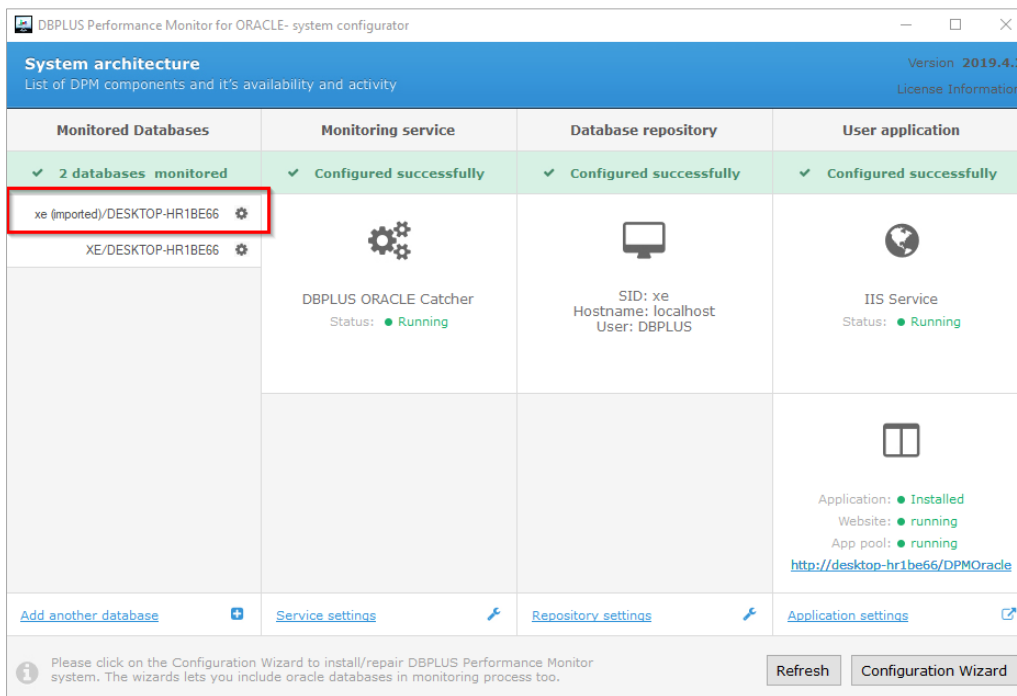
The first step in importing an Oracle database is to point to the file with the previously exported database. In case of many export files (the export may contain several files), as the file to be imported, the user indicates file with the extension *.zip. After selecting file, information about the content of files will be presented.

Then indicate whether the user wants to create a new Oracle database after import or add statistics to an existing database in monitoring.

Important! If you select an existing instance, remember to monitor the continuity of monitored data. If the date ranges overlap, some data may be overwritten and lost without being able to be restored.

After selecting the appropriate option, the Oracle database import from the file begins by clicking the [Import] button. After the import is completed, a message about successful import will be presented or error information will be presented, and the details will be saved to the log file created in the import directory.

After successful import, the instance will be added to the repository and visible in the Configuration Wizard, as shown below:



The instance will also be visible from the Dashboard screen. All imported statistics will be available on the Dbplus Performance Monitor screens, similarly to a running instance. Some screens present data directly by downloading it from the monitored database, in such cases information will only be visible if such a connection is possible.

DBPLUS Bruna 9/215 str; 02-594 Warsaw, Poland
Phone: (+48) 22 389-73-24; e-mail: info@dbplus.pl
<http://www.dbplus.tech>

© 2019 DBPLUS. All rights reserved. DBPLUS, DBPLUS logo and products mentioned in this document are trademarks of DBPLUS. All other trademarks and registered trademarks are property of their respective owners.