# Enhanced Oracle Database Monitor Plugin

Contents

[Enhanced Oracle Database Monitor Plugin 1](#_Toc384587002)

[1. Overview 3](#_Toc384587003)

[2. Configuration of the plugin 4](#_Toc384587004)

[3. Configuration of the Web UI application 6](#_Toc384587007)

[3.1 Execution of the create\_top\_sqls\_ddl.sql script 6](#_Toc384587008)

[3.2 Deploying SqlMVC.war file 6](#_Toc384587009)

[4. Example of the Enhanced Oracle Database Monitor Plugin dashboard 7](#_Toc384587010)

[4.1 Navigation Menu 8](#_Toc384587011)

[4.2 The “Slow SQLs” Web Page dashlet 9](#_Toc384587012)

[4.3 The “Oracle Locks” Web Page dashlet 10](#_Toc384587013)

[4.4 The “Analysis of Slow SQL” Web Page dashlet 11](#_Toc384587014)

[4.5 The “Analysis of Locks” Web Page dashlet 12](#_Toc384587015)

[4.6 The “Explain Plan” Web Page dashlet 13](#_Toc384587016)

[4.7 Dynamic measures captured for every slow SQL statement 16](#_Toc384587017)

[4.8 Dynamic measures captured for every database lock 17](#_Toc384587018)

[4.9 Example of a dashlet for a specific slow SID 18](#_Toc384587019)

[Figure 1 Configuration parameters 5](#_Toc384587020)

[Figure 2 Different types of slowness 6](#_Toc384587021)

[Figure 3 Example of a database.properties file 7](#_Toc384587022)

[Figure 4 Example of the Enhanced Oracle Database Monitor Dashboard 8](#_Toc384587023)

[Figure 5 Navigation Menu 9](#_Toc384587024)

[Figure 6 Example of the “Slow SQLs” dashlet 9](#_Toc384587025)

[Figure 7 Content URL parameter of the Slow SQLs dashlet 10](#_Toc384587026)

[Figure 8 Example of the Oracle Locks dashlet 10](#_Toc384587027)

[Figure 9 Content URL parameter of the Oracle Locks dashlet 11](#_Toc384587028)

[Figure 10 Example of the “Analysis of Slow SQL” dashlet 11](#_Toc384587029)

[Figure 11 Content URL parameter of the Analysis of Slow SQL dashlet 12](#_Toc384587030)

[Figure 12 Example of the Analysis of Locks dashlet 13](#_Toc384587031)

[Figure 13 Content URL parameter of the Analysis of Locks dashlet 13](#_Toc384587032)

[Figure 14 Example of the Explain Plan dashlet with selected SID 14](#_Toc384587033)

[Figure 15 Content URL parameter of the Explain Plan dashlet 15](#_Toc384587034)

[Figure 16 List of dynamic measures captured by every slow SQL statement 16](#_Toc384587035)

[Figure 17 List of dynamic measures captured by every database lock 17](#_Toc384587036)

[Figure 18 Chart which contains Elapsed Time, CPU Time, and Parse Calls stats over time for query with SID 2721813848 18](#_Toc384587037)

## Overview

Enhanced Oracle Database Monitor Plugin adds to the existing Oracle Monitor Plugin ability to gather statistics on the SQL level. It provides customers with the following statistics for every SQL statement:

* SQL SID
* SQL Full Text
* Child Number
* Number of Executions
* Elapsed Time
* Average Elapsed Time
* CPU Time
* Average CPU Time
* Disk Reads
* Direct Writes
* Buffer Gets
* Rows Processed
* Parse Calls
* First Load Time
* Last Load Time

For in-depth SQL analysis there is SQL explain plan which was captured at the time when this SQL statement was executed.

Besides detailed SQL level statistics there are stats about database locks, tablespaces (coming) etc. which give user additional information about state of the database. For database locks the following information is captured:

* Concatenation of the session\_id from the gv$locked\_object view and serial# from the v$session view
* Oracle User
* Object Name
* Object Type
* Lock Mode
* Status
* Last DDL Time

Plugin keeps information in the performance warehouse (or in any external relational database) and hence allows going back in history to compare performance of the SQL in question over time. The Web UI piece of the plugin handles getting historical data and allows performing analysis of the slow SQL statements, explaining plans, locks, etc. historically.

## Configuration of the plugin

Configuration of the plugin follows standard requirements for dynaTrace plugin. It has the following configuration parameters (see Figure 1):

1. hostname   
   Host name of the monitored database
2. dbName  
   Database name of monitored database
3. dbPort  
   Port number of the monitored database. Default value is 1521.
4. dbUsername  
   Name of the user to connect to the database
5. dbPassword  
   User’s password
6. Top Slow SQLs  
   Number of top slow SQL statements. Default value 10.
7. isExplainPlan  
   Indicator which shows if explain plan will be populated for each SQL statement. Values are “true” or “false”. Default value is “false”.
8. isDynamicMeasure  
   Indicator which shows if plugin will populate dynamic measures for the slow top N SQL statements and database locks. Values are “true” or “false”. To prevent measure explosion, set this indicator to “false”. In this case dynamic measures (see sections 4.7 and 4.8 below) will not be populated. Default value is “false”.
9. htmlFileSqls  
   path to the file which contains list of top slow SQL statements. This path depends on used web server. For embedded dynaTrace web server see article [KB-381](https://community.compuwareapm.com/community/display/KB/KB-381+How+To+add+custom+content+to+the+dynaTrace+internal+web+server).
10. htmlFileLocks  
    path to the file which contains list of database locks. This path depends on used web server. For embedded dynaTrace web server see article [KB-381](https://community.compuwareapm.com/community/display/KB/KB-381+How+To+add+custom+content+to+the+dynaTrace+internal+web+server).
11. typeOfSlowness  
    There are following 8 types of slowness (see Figure 2) that top SQL statements can be selected from. They are:
    1. Elapsed Time
    2. Buffer Gets
    3. CPU Time
    4. Executions
    5. Parse Calls
    6. Disk Reads
    7. Direct Writes
    8. Rows Processed
12. hostNameHistory  
    Host name of the database where historical data will be stored. dynaTrace performance warehouse can be used for this.
13. dbNameHistory  
    Database name of monitored database . dynaTrace performance warehouse can be used for this.
14. dbPortHistory  
    Port number of the monitored database. Default value is 1521. dynaTrace performance warehouse can be used for this.
15. dbUsernameHistory  
    Name of the user to connect to the database. dynaTrace performance warehouse can be used for this.
16. dbPasswordHistory  
    User’s password

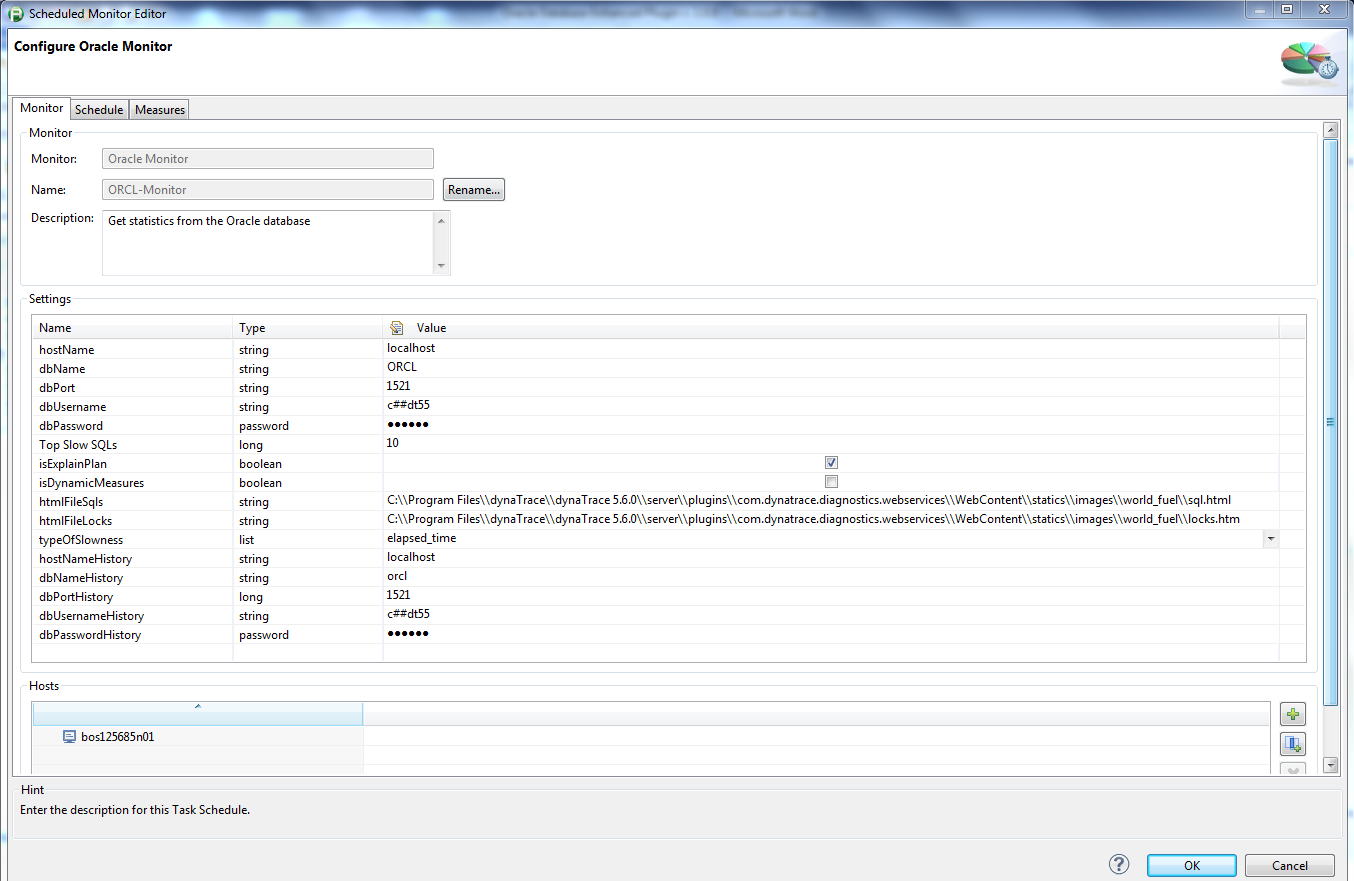


Figure 1 Configuration parameters

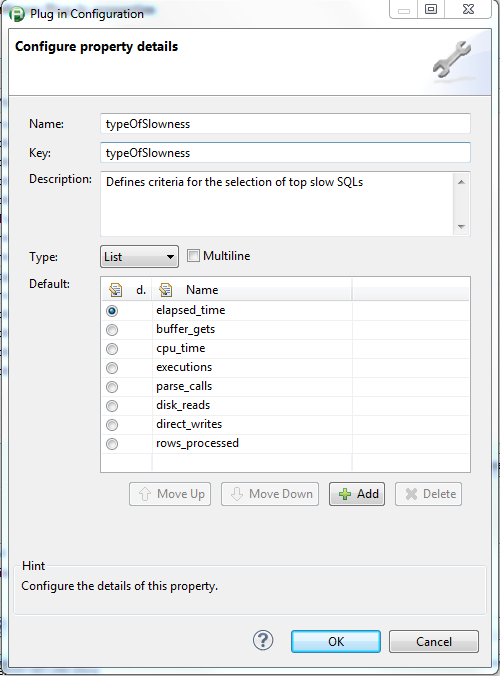


Figure 2 Different types of slowness



## Configuration of the Web UI application

There are two steps which need to be performed to configure and install the Web UI part:

1. Execute create\_top\_sqls\_ddl.sql script to create history tables and supporting indices
2. Deploy provided web application archive, i.e. war file, on the application server of choice (e.g. Tomcat, Jetty, etc.)

***Note***

* *Plugin needs to be deployed after the create\_top\_sqls\_ddl.sql script was executed.*
* *We are working to provide procedure to deploy SqlMVC application war file on the embedded Jetty server which is coming with the dynaTrace product.*

### Execution of the create\_top\_sqls\_ddl.sql script

Before execution of the create\_top\_sqls\_ddl.sql change owner name placeholder to the real name for tables, indices, and sequence.

### Deploying SqlMVC.war file

Before deploying SqlMVC.war file, set environmental variable “ext.prop.dir” to a directory where the database.properties file is located. For example, execute command   
 “set ext.prop.dir=C:\Users\dmaext0\”   
on Windows OS to set directory value to C:\Users\dmaext0\.   
*Note: Do not forget to add “\” (backward slash) at the end of the directory name. On Unix systems use “/”.*

Content of the database.properties file is depicted below. Make sure to change values of the ${db.url}, ${db.user}, and ${db.password} appropriately before using SqlMVC application.

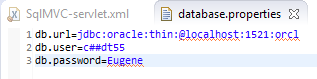


Figure Example of a database.properties file

Make sure that user which owns application server process (e.g. Tomcat, Jetty, JBoss, etc.) has read access to the database.properties file.

Use standard deployment procedure to deploy SqlMVC.war file on the application server of choice. For example, for Tomcat 7.0 the standard deployment procedure is described [here](https://tomcat.apache.org/tomcat-7.0-doc/appdev/deployment.html). One of the options to deploy war file (probably the easiest one) is to follow up steps from the following extract from the above article:

* Copy the web application archive file into directory *$CATALINA\_BASE/webapps/*. When Tomcat is started, it will automatically expand the web application archive file into its unpacked form, and execute the application that way. This approach would typically be used to install an additional application, provided by a third party vendor or by your internal development staff, into an existing Tomcat installation. **NOTE** - If you use this approach, and wish to update your application later, you must both replace the web application archive file **AND** delete the expanded directory that Tomcat created, and then restart Tomcat, in order to reflect your changes.

## Example of the Enhanced Oracle Database Monitor Plugin dashboard

Example of the dashboard is depicted on Figure 4. Highlighted in red are dashlets which display per SQL stats, database locks, and explain plans of slow SQLs. These dashlets are:

1. “Slow SQL” - contains the latest stats only (no history)
2. “Oracle Locks” - contains the latest stats about locks only (no history)
3. “Analysis of Slow SQL” - contains list of database slowest SQLs and ability to navigate through the historical data for the slow SQLs
4. “Analysis of Locks” - contains list of database locks and ability to navigate through the historical data for locks
5. “Explain Plans” – contain explain plan for every slow SQL statement. This dashlet has ability to navigate through the history data to see if SQL’s explain plan changed.
6. “SID XXXXXX” dashlet charts selected metrics captured by the plugin.

“Analysis of Slow SQL”, “Analysis of Locks”, and “Explain Plans” dashlets contain navigation menu which is described in next section. Configuration details for each of the above dashlets are depicted in the following sections.

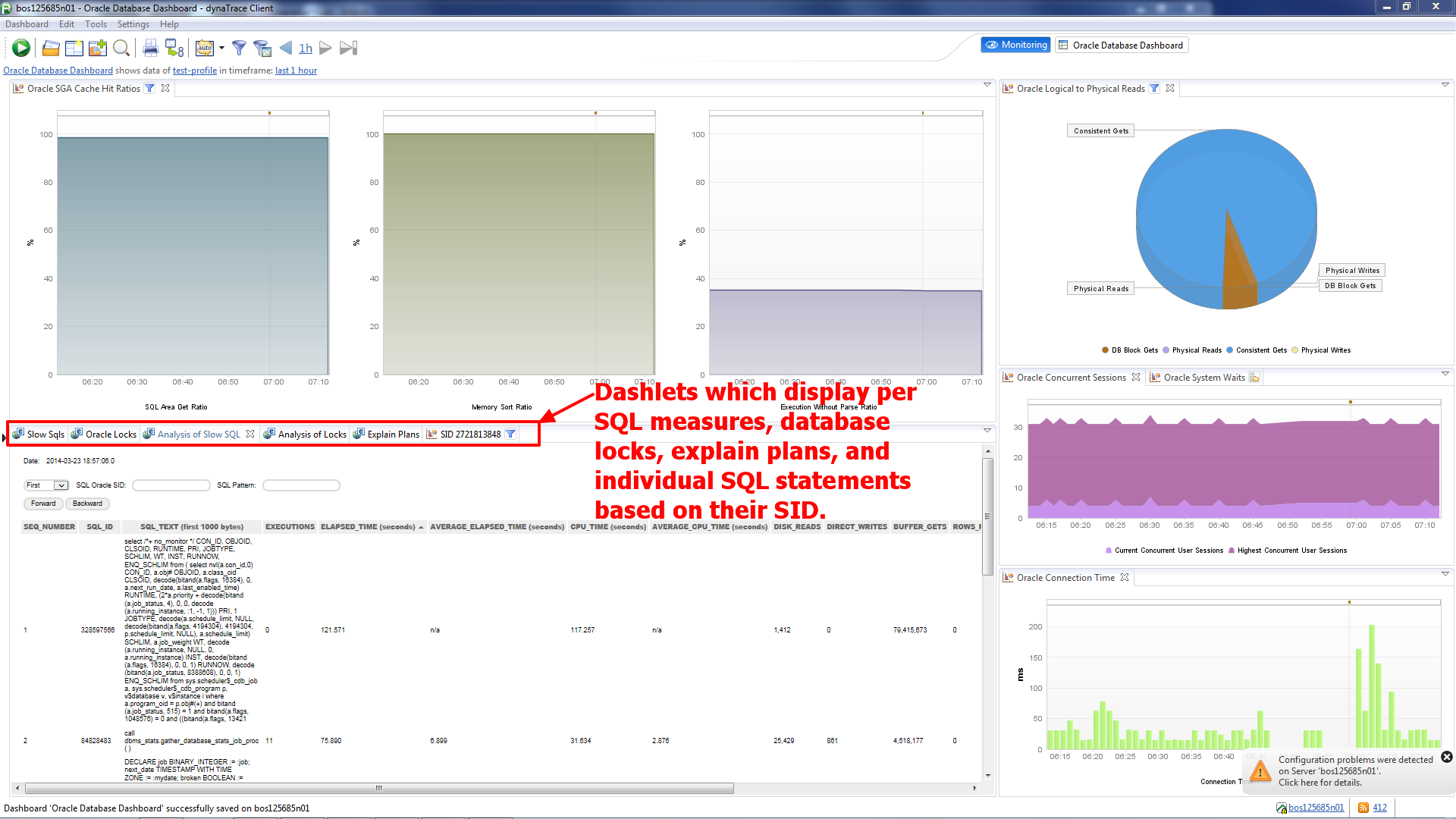


Figure 4 Example of the Enhanced Oracle Database Monitor Dashboard

### Navigation Menu

Analysis dashlets contain *navigation menu* (see Figure 5) which allows displaying stats historically. Forward and Backward buttons allow to navigate back and forth using time interval selected from the drop down list box. Date field shows timestamp of the data displayed by a dashlet. Time interval from the drop down list box can have one of the following values:

* First
* 1 min
* 5 min
* 10 min
* 15 min
* 30 min
* 1 hour
* 6 hours
* 12 hours
* 1 day
* 1 week
* Last

The First and the Last values allow to jump to the latest or to the earliest historical data respectively, while 1 min, 5 min, … allow to jump in history on 1 min, 5 min, … interval in direction defined by the Forward or Backward buttons.

The SQL Oracle SID field contains comma separated list of SID(s). If this field is populated, dashlet will display data related to the SQL statements with given SID(s).

The SQL Pattern field contains pattern that SQL statements will be searched for to find matched SQL(s). Only matched SQLs will be displayed.

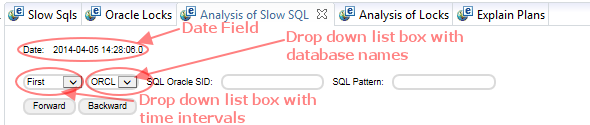


Figure 5 Navigation Menu

### The “Slow SQLs” Web Page dashlet

The “Slow SQLs” dashlet is a Web Page dashlet. Please see Figure 6 for an example of the dashlet. The only configuration parameter of the dashlet is the Content URL depicted in Figure 7. Dashlet displays about 15 metrics listed in the Overview section for each SQL statement. All columns on the html page are sortable.

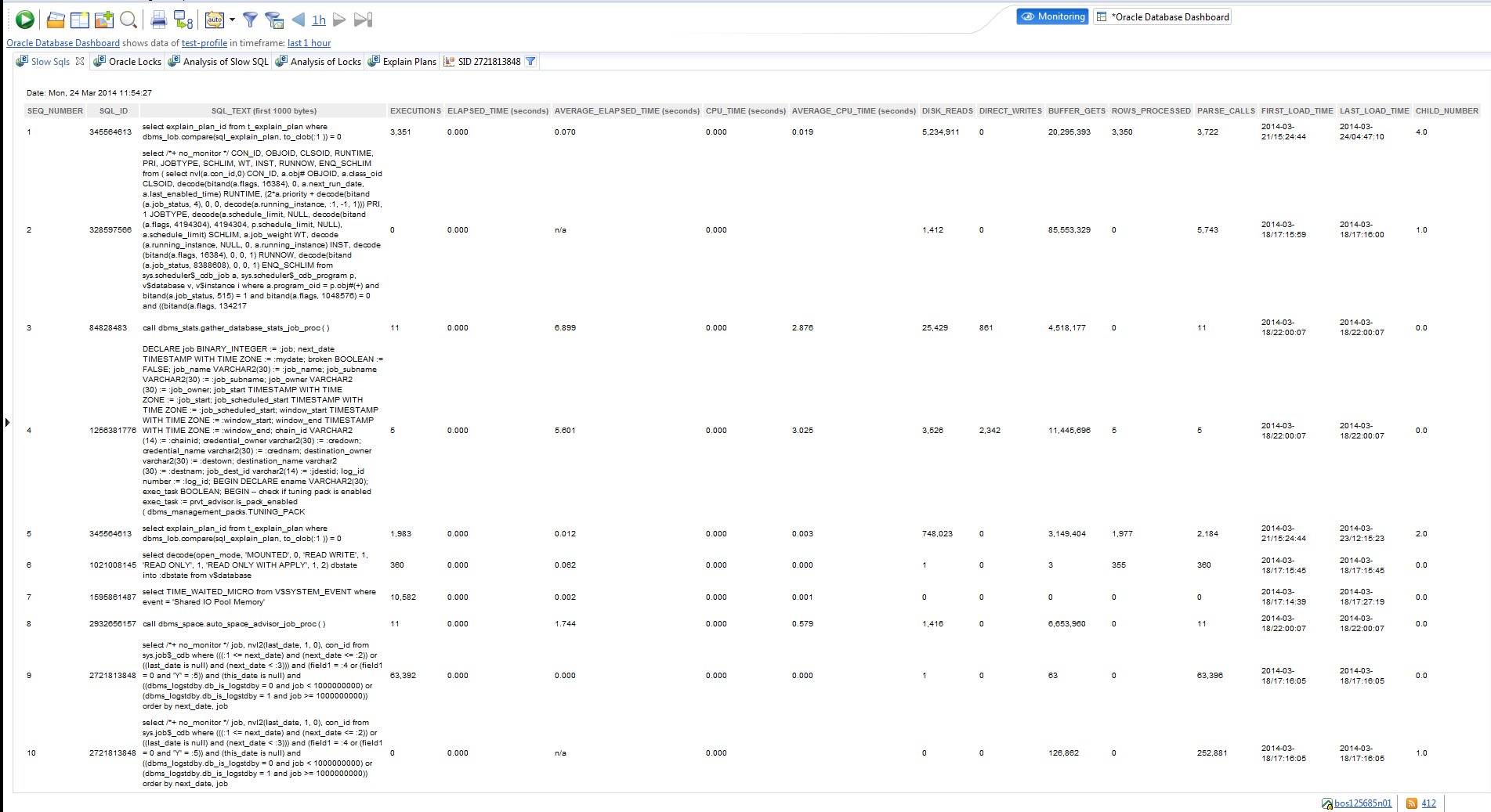


Figure 6 Example of the “Slow SQLs” dashlet

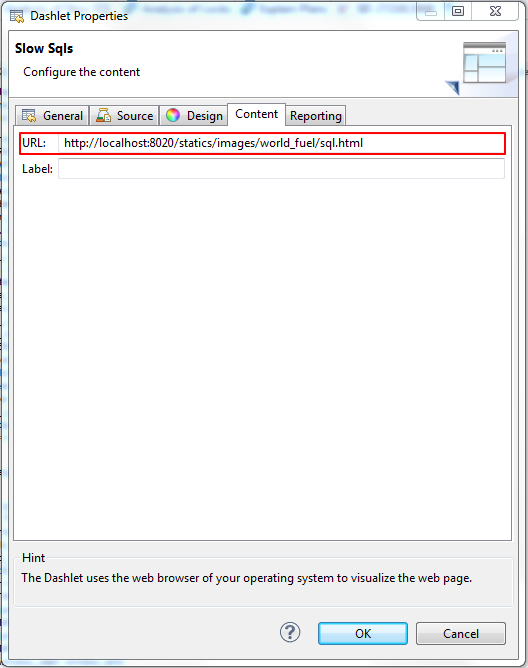


Figure 7 Content URL parameter of the Slow SQLs dashlet

### The “Oracle Locks” Web Page dashlet

The “Oracle Locks” dashlet is a Web Page dashlet. Please see Figure 8 for an example of the dashlet. The only configuration parameter of the dashlet is the Content URL depicted in Figure 9. Dashlet displays 7 metrics listed in the Overview section for each database lock. All columns on the html page are sortable.

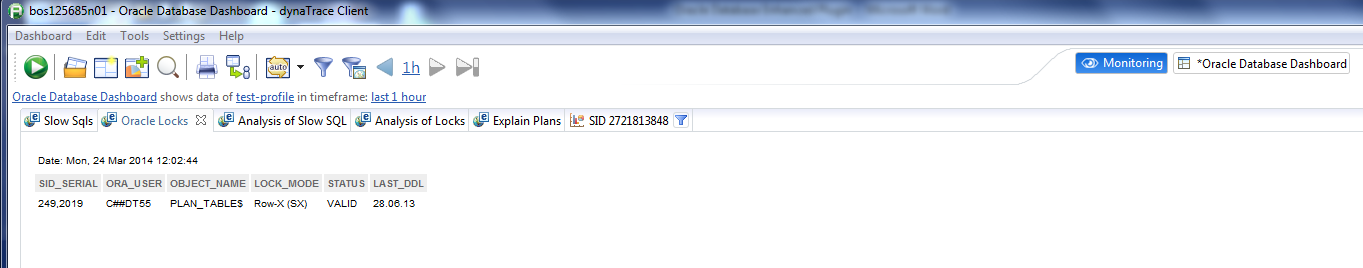


Figure 8 Example of the Oracle Locks dashlet

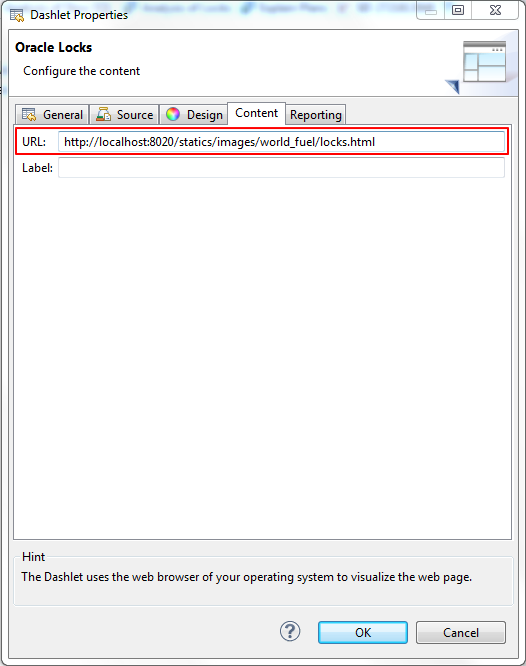


Figure 9 Content URL parameter of the Oracle Locks dashlet

### The “Analysis of Slow SQL” Web Page dashlet

The “Analysis of Slow SQL” dashlet is a Web Page dashlet. Please see Figure 10 for an example of the dashlet. The only configuration parameter of the dashlet is the Content URL depicted in Figure 11. Dashlet displays about 15 metrics listed in the Overview section for each SQL statement. All columns on the html page are sortable. Navigation Menu is described in section 4.1.

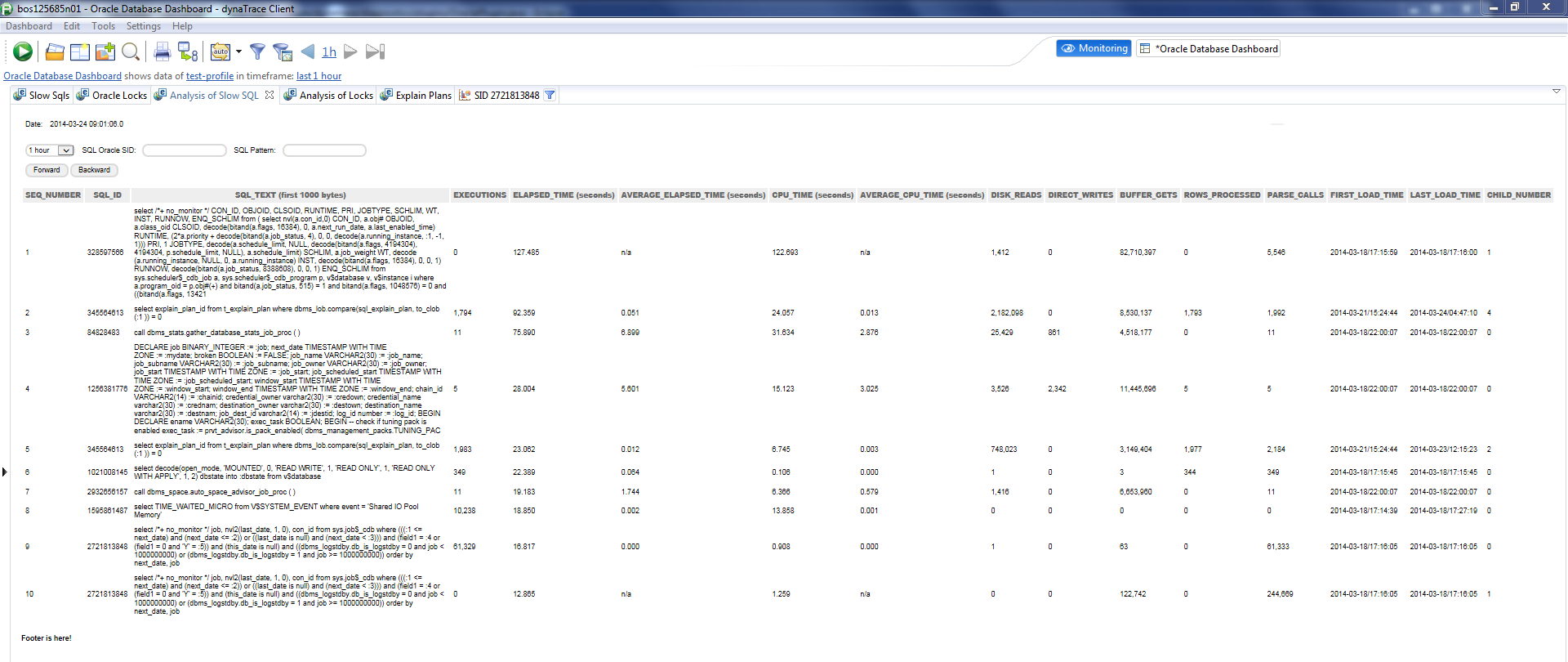


Figure 10 Example of the “Analysis of Slow SQL” dashlet

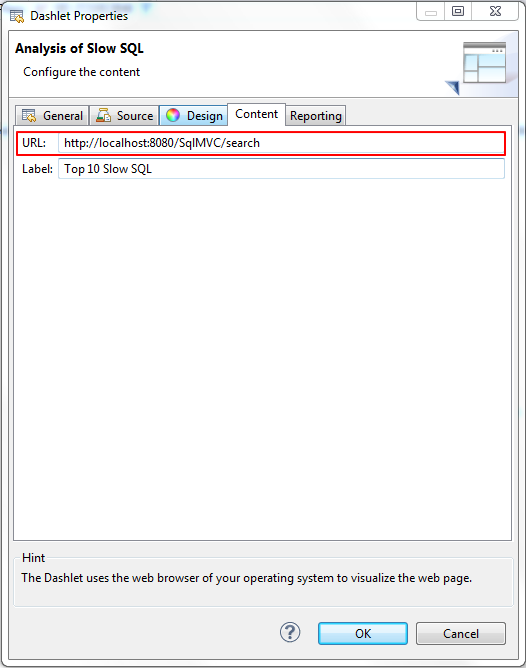


Figure 11 Content URL parameter of the Analysis of Slow SQL dashlet

### The “Analysis of Locks” Web Page dashlet

The “Analysis of Locks” dashlet is a Web Page dashlet. Please see Figure 12 for an example of the dashlet. The only configuration parameter of the dashlet is the Content URL depicted in Figure 13. Dashlet displays 7 metrics listed in the Overview section for each lock. All columns on the html page are sortable. Navigation Menu is described in section 4.1.

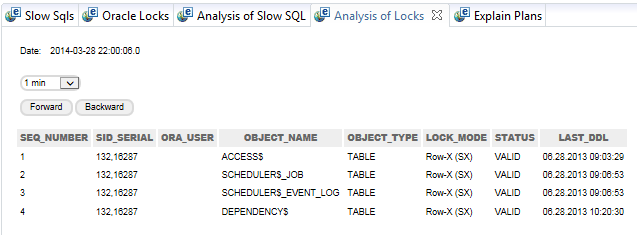


Figure 12 Example of the Analysis of Locks dashlet

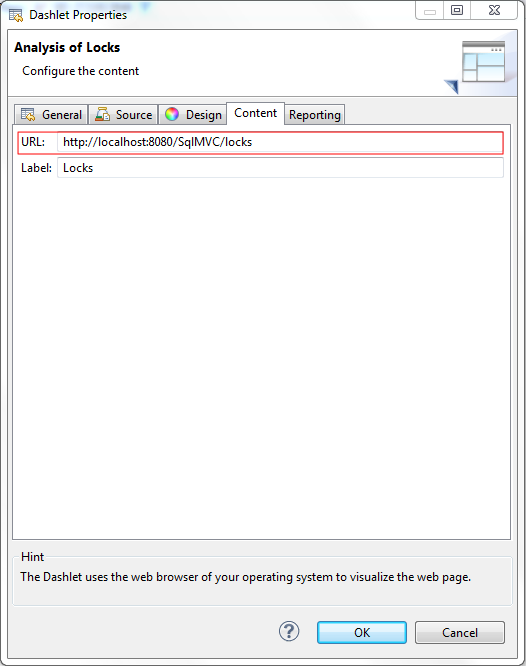


Figure 13 Content URL parameter of the Analysis of Locks dashlet

### The “Explain Plan” Web Page dashlet

The “Explain Plan” dashlet is a Web Page dashlet. Please see Figure 14 for an example of the dashlet. The only configuration parameter of the dashlet is the Content URL depicted in Figure 15. Dashlet displays Slow SQL explain plan. All columns on the html page are sortable. Navigation Menu is described in section 4.1.

Below is example of top 10 slow SQL statements and their explain plans. Please use zoom in/out to make image readable.



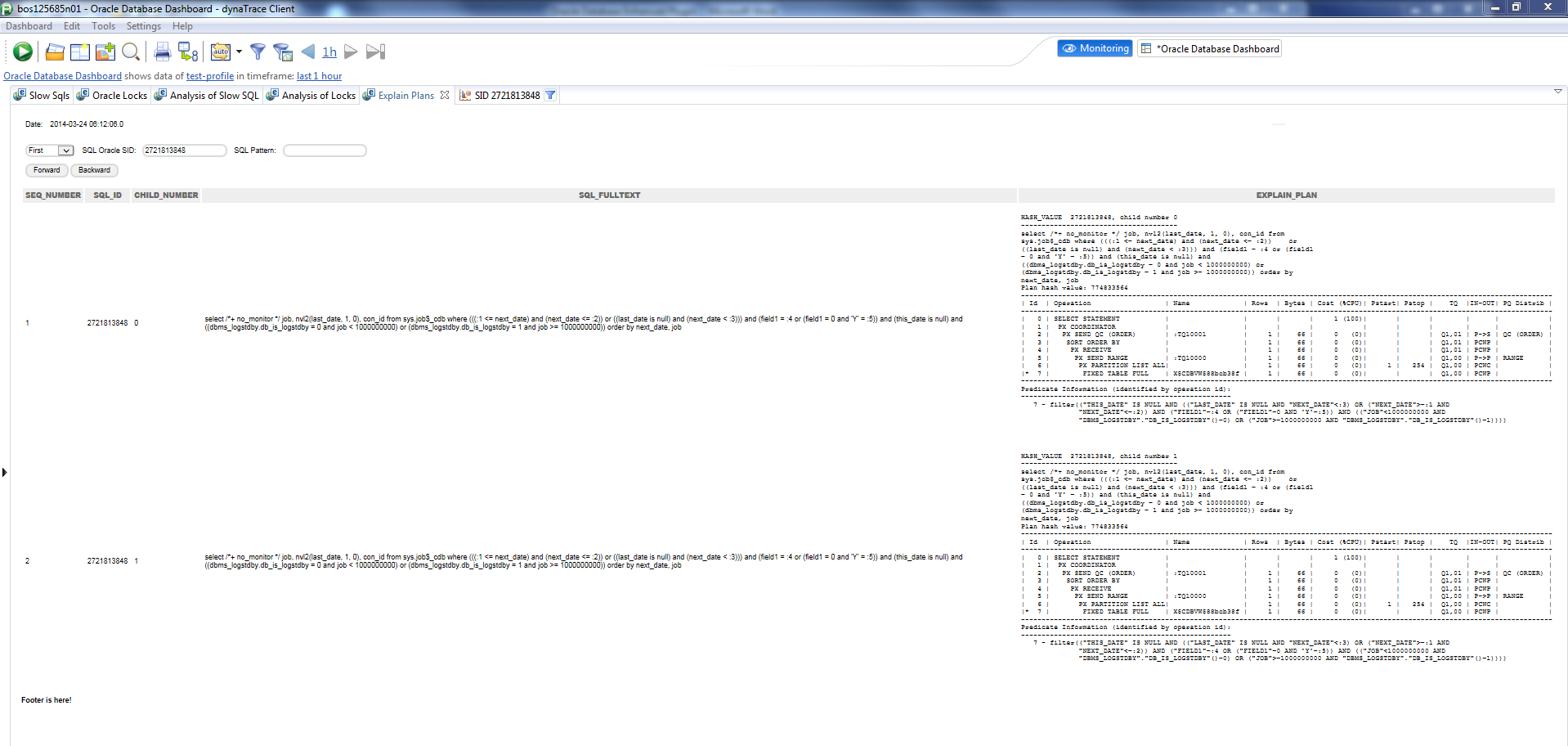


Figure 14 Example of the Explain Plan dashlet with selected SID

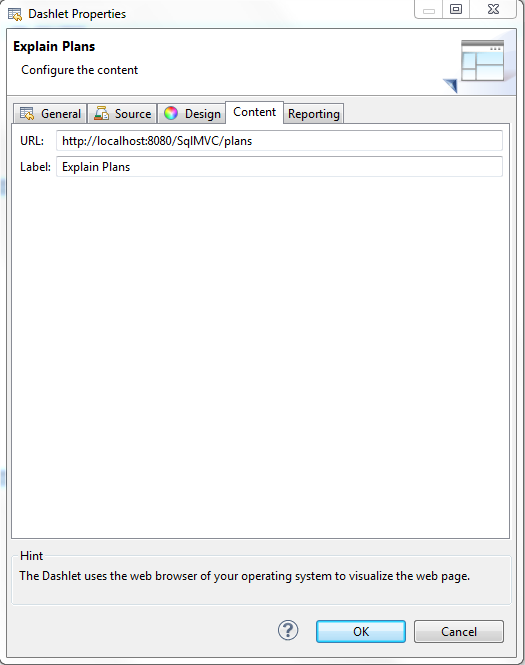


Figure 15 Content URL parameter of the Explain Plan dashlet

### Dynamic measures captured for every slow SQL statement

Figure 16 depicts list of dynamic measures captured for every slow SQL statement. Based measures of these dynamic measures can be used to setup incident rules to capture events of violated thresholds.

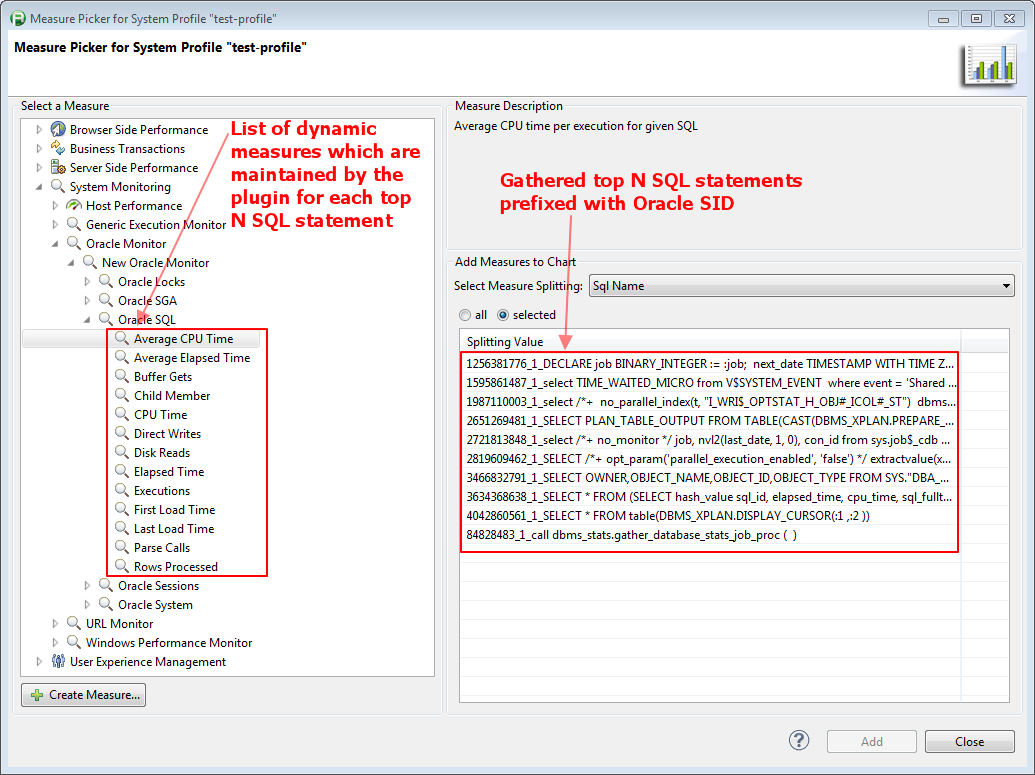


Figure 16 List of dynamic measures captured by every slow SQL statement

### Dynamic measures captured for every database lock

Figure 17 depicts list of dynamic measures captured for every database lock. Based measures of these dynamic measures can be used to setup incident rules to capture events of violated thresholds.

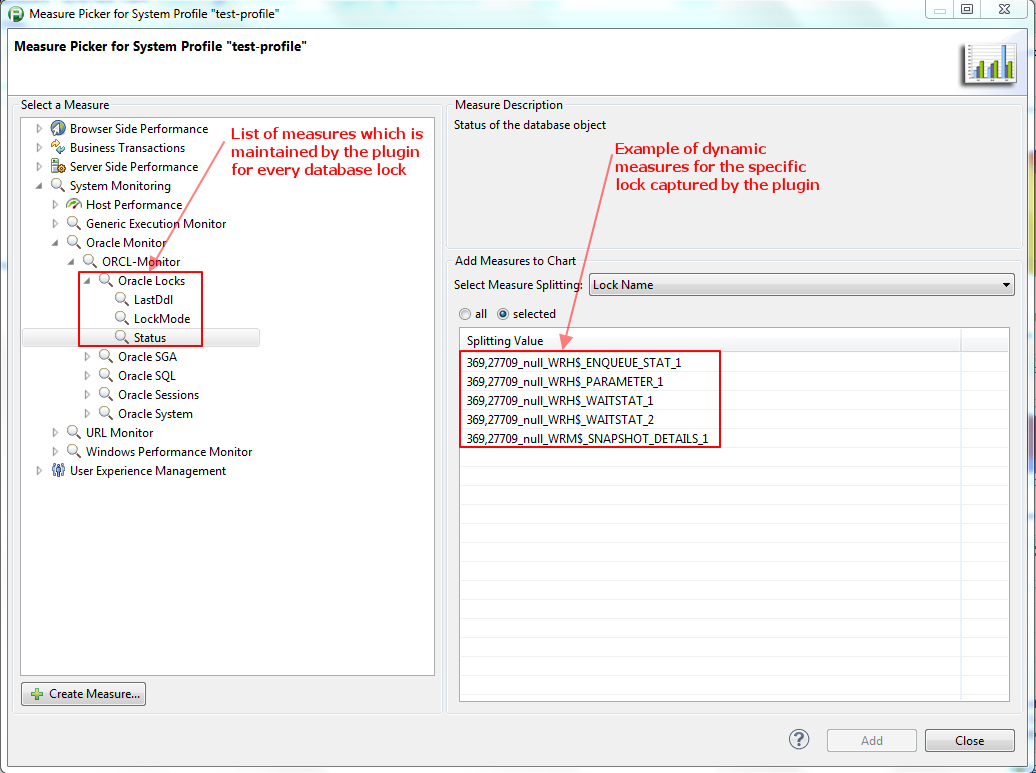


Figure 17 List of dynamic measures captured by every database lock

### Example of a dashlet for a specific slow SID

Dashlet on Figure 16 depicts Elapsed Time, CPU Time, and Parse Calls taken over time for Oracle query with SID 2721813848 which EXPLAIN Plan is presented on Figure 14. Please see that the above stats (i.e. Elapsed Time, CPU Time, and Parse Calls) are growing over time. Performance of this SQL statement is degrading.

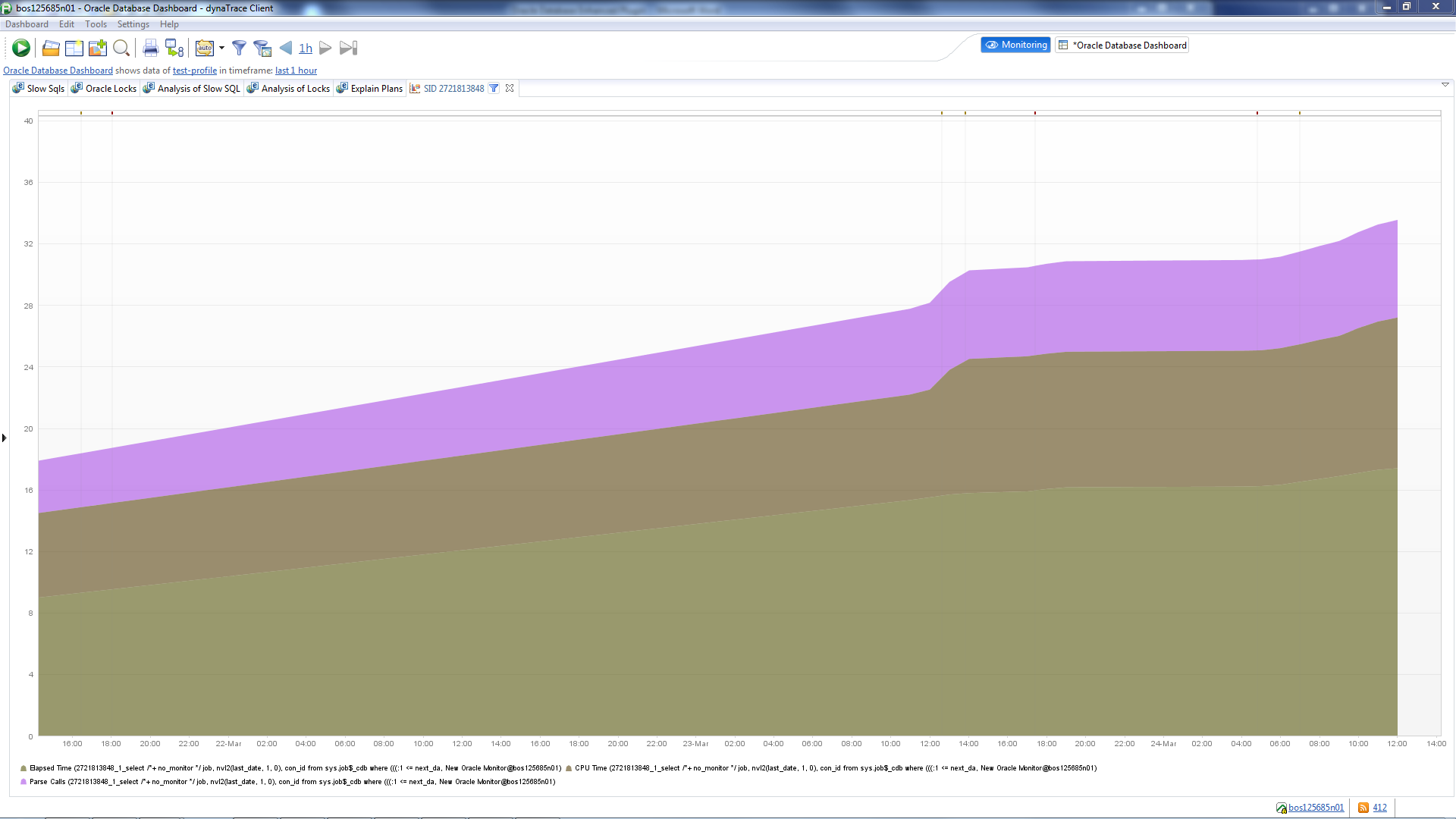


Figure 18 Chart which contains Elapsed Time, CPU Time, and Parse Calls stats over time for query with SID 2721813848