DBPLUS
better performance

DBPLUS
Performance Monitor for MS SQL
description of changes in version 2022.1

Date: 10,April 2022

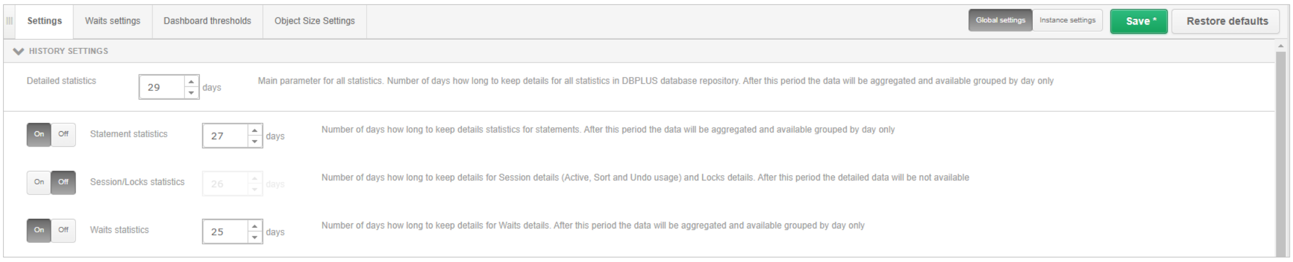
Table of Contents

Below is a list of changes to the DBPLUS Performance Monitor system for MS SQL instance monitoring.

New in 2022.1

1 Data management in the DBPLUS repository database

In the DBPLUS application, all data related to the monitoring of the instance are saved in one place in the repository database. Detailed data (e.g. query statistics collected at 15-minute intervals) are aggregated into daily data after a certain period of time. Until now, the user had only one main parameter that controlled the retention time of data in the repository.
In the latest version of the application, we have added parameters that enable individual configuration of the data retention time in the DBPLUS **repository** for the most important monitoring areas.
The configuration change is available in the settings menu from the main menu: **Configuration> Settings.** The parameters responsible for setting the time length can be found in the dedicated **HISTORY SETTINGS** section in the **Settings** tab.



HISTORY SETTINGS

Detailed statistics 29 days Main parameter for all statistics. Number of days how long to keep details for all statistics in DBPLUS database repository. After this period the data will be aggregated and available grouped by day only

On Off Statement statistics 27 days Number of days how long to keep details statistics for statements. After this period the data will be aggregated and available grouped by day only

On Off Session/Locks statistics 26 days Number of days how long to keep details for Session details (Active, Sort and Undo usage) and Locks details. After this period the detailed data will be not available

On Off Waits statistics 25 days Number of days how long to keep details for Waits details. After this period the data will be aggregated and available grouped by day only

The section includes the configuration for the main Detailed Statistics parameter, as well as the parameters of the main monitoring areas. Setting the configuration for any of the parameters overwrites the settings of the main parameter. Please note that the change applies to all monitored instances.

Parameters are available for the areas:

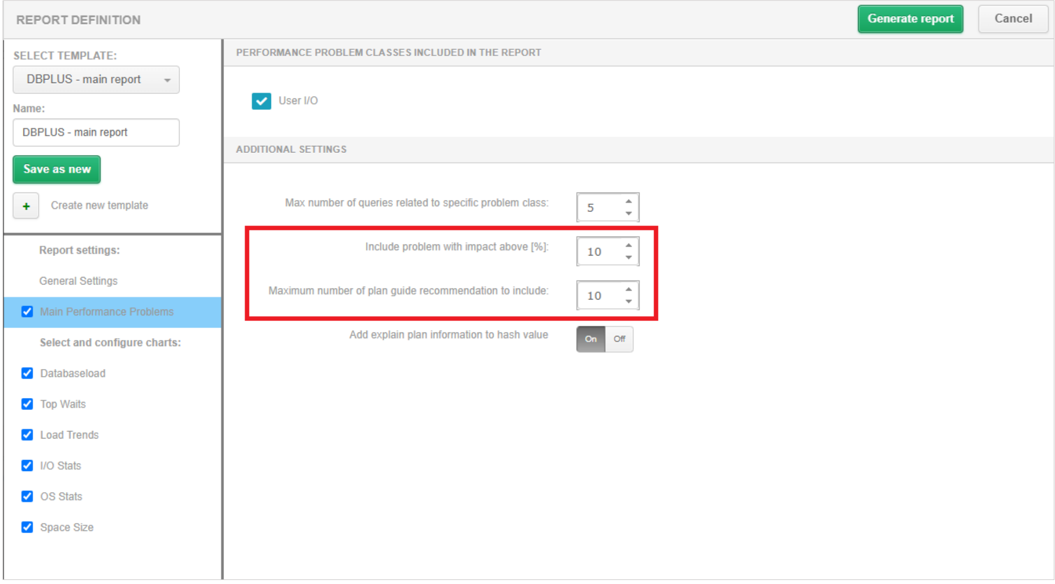
- Statements statistics - concerns the query area,
- Session \ Locks statistics - concerns the area of locks and sessions,
- Waits statistics - concerns the wait area,
- Latches statistics - applies to Latch statistics available on Latches: Overview and Buffer latches screens,
- Jobs - applies to Job statistics in the case of enabled monitoring user rights,
- Latch Spin Lock history - applies to information about Latch Spin Lock,
- Perf Counters statistics - performance statistics available in the Perf Counters tab,
- Session memory details - the parameter is responsible for storing detailed data on the memory usage by the session,
- Objects Space Size - a parameter responsible for the time of keeping information on the size of objects in the monitored database. Object occupancy monitoring is an additional module that we turn on in the settings menu in the Object Size Settings tab,
- IO Stats - the parameter is responsible for the storage time of IO Stats data in the DBPLUS repository,
- Backups - applies to the backup area.

Each change of parameters introduces changes from the next snap for the thread deleting the stories from the repository database. Information on data deletion can be followed in the logs available from the main menu **Servers monitor> Logs in the Deletion procedure runtime** tab.

2 Reports

In the DBPLUS Performance Monitor application, we have the ability to report the performance of the monitored instance. In the latest version of the application, we have added to the report information about the creation of an PlanGuide objects in the reporting period. If an PlanGuide has been created for a given query, such information will be included in the report. This information will be available in the Performance Monitor and Anomaly Monitor report.

During the report Anomaly monitor generation process, the user can configure how many newly created PlanGuide objects in the period for which the report is generated are to be included in the report.
Additionally, a parameter has been added thanks to which only significant problems affecting performance will be included in the report. The default configuration assumes that a given problem must account for over 10% of the impact in relation to other problems detected by DBPLUS Anomaly Monitor.



REPORT DEFINITION

Generate report | Cancel

SELECT TEMPLATE: DBPLUS - main report

Name: DBPLUS - main report

Save as new

+ Create new template

Report settings:

General Settings

☑ Main Performance Problems

Select and configure charts:

☑ Databaseload

☑ Top Waits

☑ Load Trends

☑ I/O Stats

☑ OS Stats

☑ Space Size

PERFORMANCE PROBLEM CLASSES INCLUDED IN THE REPORT

☑ User I/O

ADDITIONAL SETTINGS

Max number of queries related to specific problem class: 5

Include problem with impact above [%]: 10

Maximum number of plan guide recommendation to include: 10

Add explain plan information to hash value: On | Off

Below is an example of information, along with a space on the chart informing about adding an PlanGuide for a given query.

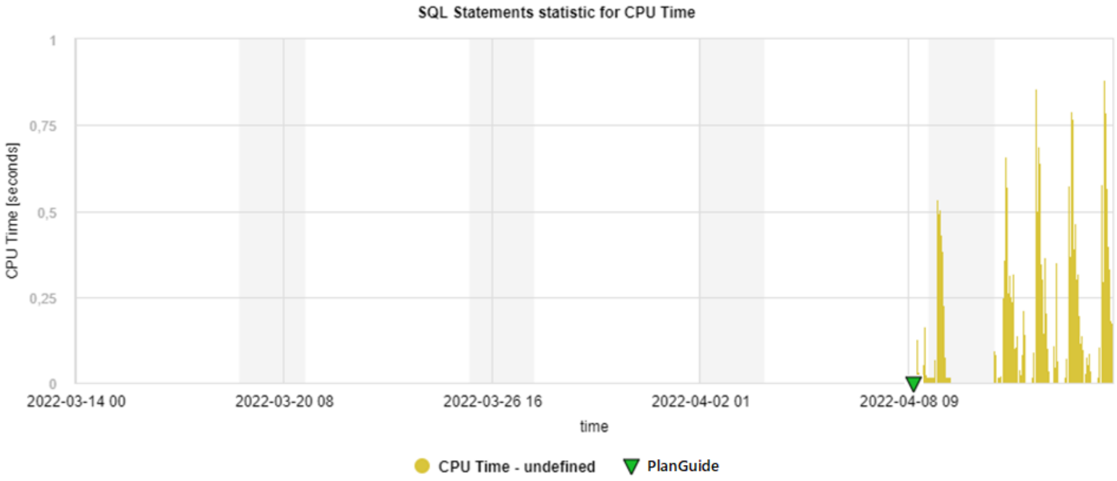
Notice! Please note that in the case of the MSSQL platform, after generating the PlanGuide for the query, a new query identifier is created.

The chapter includes Plan Guides created in the reported period.

8.1.1. Query hash: 0x8B597448B0BEA59D

The query was optimized via Plan Guide with name **DBPLUS_0x48CC450396BF8C79** , hints **USE PLAN HINTS** on **08.04.2022 14:06:47**.

Average elapsed time for a day is 4,79s in the reported period. Average elapsed time per 1 execution is 0,00038s in the reported period. The charts represents CPU time grouped by hour in the period between 31.03.2022 and 14.04.2022.



The problem with slow report generation has been corrected in the latest version. The problem was related to the change of the execution plan of one of the queries that collected data for the report. The query has been optimized.

Improvement in Anomaly Report

In the latest version, changes have been made to the Anomaly Monitor report related to:

- Improvement of the currently occurring data presentation errors in the report,
- acceleration of the report generation process
- improvement of the translation for the Polish version

3 Anomaly Monitor

The latest version introduces a number of changes to the functionality of detecting performance problems.

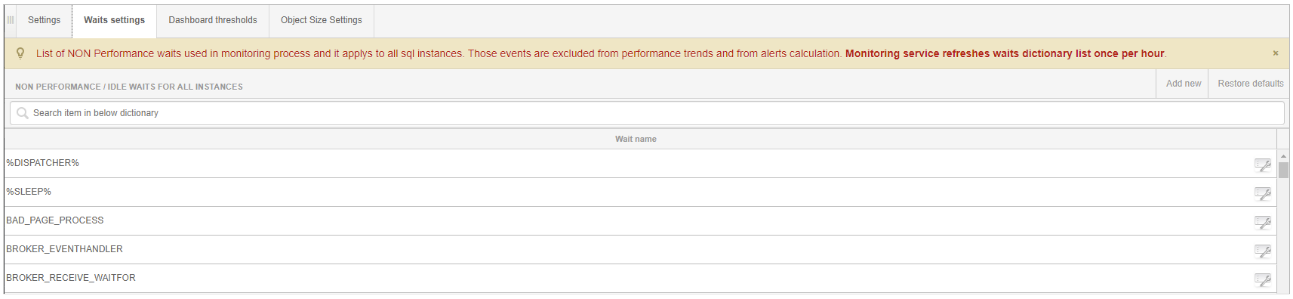
Improved detection calculation for queries that run at fixed times

In the latest version of the algorithm that verifies the occurrence of performance anomalies in the database, we have improved the mechanism that checks whether a given performance problem does not result from a repetitive, constant process that is run periodically on the monitored database. If a disturbance resulting from a cyclical process launched at fixed hours is detected, such an alert will not be generated.

4 Bug fixes and improvements

4.1. Improvements for Azure SQL Database version

In the latest version, POPULATE_LOCK_ORDINALS wait has been added to the non-performance wait dictionary. In the DBPLUS application when detecting a wait that does not affect performance, the user has the option to add it to the dictionary. This dictionary contains information about waits that do not affect the duration of queries and will not be counted by applications to the wait level.
The dictionary is available in the **Configuration> Settings** menu, in the **Waits settings** tab.



List of NON Performance waits used in monitoring process and it applies to all sql instances. Those events are excluded from performance trends and from alerts calculation. **Monitoring service refreshes waits dictionary list once per hour**.

Wait name
%DISPATCHER%
%SLEEP%
BAD_PAGE_PROCESS
BROKER_EVENTHANDLER
BROKER_RECEIVE_WAITFOR

Fixed a bug on the session page and blocks history

The latest version includes an amendment related to the presentation of data on the lock history screen and session. The fix was **for Microsoft Azure SQL.**

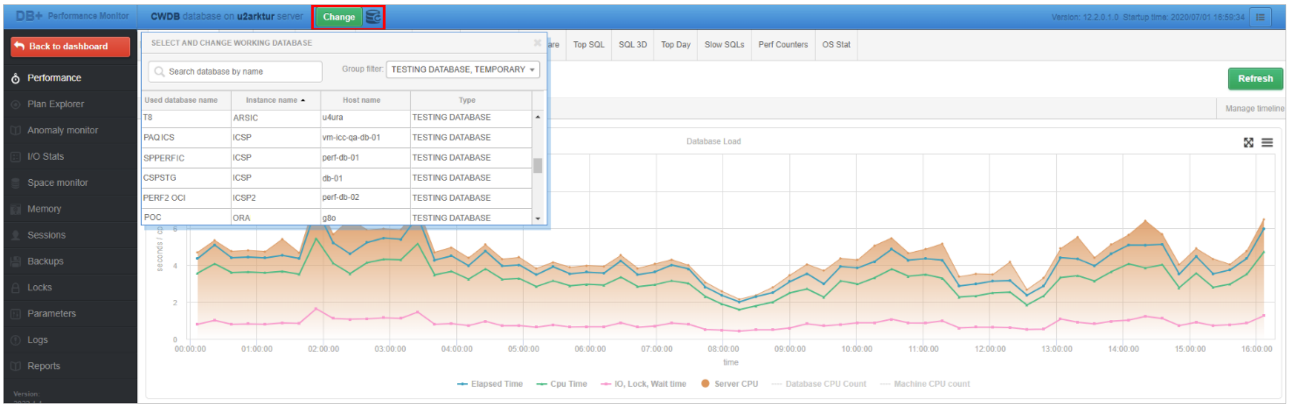
4.2. Fixes related to the operation of the SQL Parser functionality

In the latest version, we have introduced a number of fixes related to the query parsing mechanism available on the Show Plan Objects screen. The changes include improvements to the ergonomics of operation and improvement of the parser operation in the following areas:

- Support for new types of queries
- Parser support for non-existent objects

4.3. Database instance change

In the latest version of the application, we have improved the functionality of changing the monitored database instance. In the application, while viewing performance statistics, we have the option of changing the database context. It is possible to change it without having to go back to the Dashboard screen. In order to change the context, it is enough to click on the **[Change]** button available on the blue bar at the top of the screen. After clicking, the list of available monitored databases will be expanded with the possibility of searching by the selected filter or name.



SELECT AND CHANGE WORKING DATABASE

Search database by name

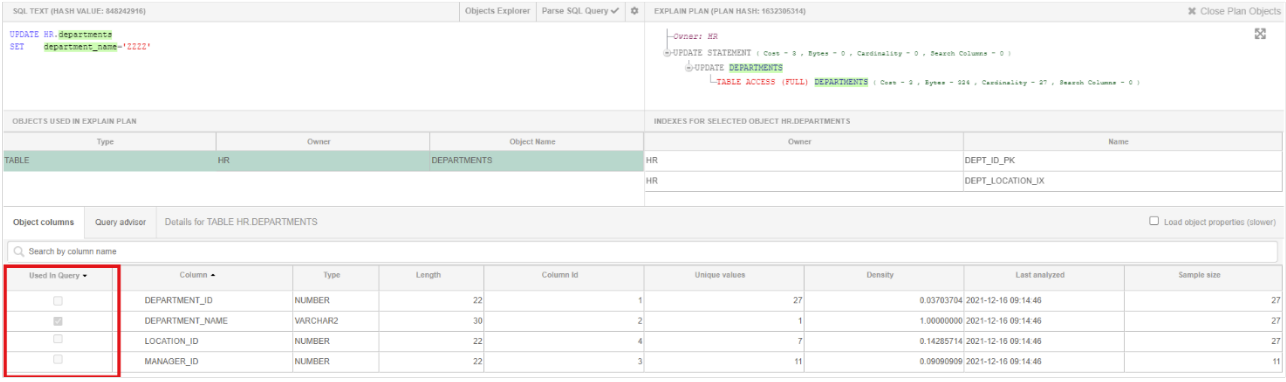
Group filter: TESTING DATABASE, TEMPORARY

Used database name	Instance name	Host name	Type
T8	ARSIC	u4ura	TESTING DATABASE
PAQICS	ICSP	vm-icc-qa-db-01	TESTING DATABASE
SPPERFIC	ICSP	perf-db-01	TESTING DATABASE
CSPSTG	ICSP	db-01	TESTING DATABASE
PERF2 OCI	ICSP2	perf-db-02	TESTING DATABASE
POC	ORA	g8o	TESTING DATABASE

4.4. Improvements on the Show Plan Objects page

Designation of the columns used in the query

In the latest version, we have added functionality that allows faster query performance analysis on the Show Plan Objects screen. The change consists in additional marking the columns used in a given query in the grid. The change is visible on the screen in the form of an additional column Used in query, which is displayed after the **Parse SQL Query** operation.



SQL TEXT (HASH VALUE: 848242916)

```
UPDATE HR.departments
SET    department_name='ZZZZ'
```

EXPLAIN PLAN (PLAN HASH: 1632305314)

```
 ├Owner: HR
⊖UPDATE STATEMENT ( Cost - 3 , Bytes - 0 , Cardinality - 0 , Search Columns - 0 )
   ⊖UPDATE DEPARTMENTS
      └TABLE ACCESS (FULL) DEPARTMENTS ( Cost - 3 , Bytes - 324 , Cardinality - 27 , Search Columns - 0 )
```

OBJECTS USED IN EXPLAIN PLAN

Type	Owner	Object Name
TABLE	HR	DEPARTMENTS

INDEXES FOR SELECTED OBJECT HR.DEPARTMENTS

Owner	Name
HR	DEPT_ID_PK
HR	DEPT_LOCATION_IX

Used In Query	Column	Type	Length	Column Id	Unique values	Density	Last analyzed	Sample size
□	DEPARTMENT_ID	NUMBER	22	1	27	0.03703704	2021-12-16 09:14:46	27
☑	DEPARTMENT_NAME	VARCHAR2	30	2	1	1.00000000	2021-12-16 09:14:46	27
□	LOCATION_ID	NUMBER	22	4	7	0.14285714	2021-12-16 09:14:46	27
□	MANAGER_ID	NUMBER	22	3	11	0.09090909	2021-12-16 09:14:46	11

Hypothetical indices

The Plan Objects screen shows detailed information about the objects included in the query (tables, indexes). The website also presented hypothetical indices. These indexes do not take up any physical disk space. This is a special type of object used only for generating an execution plan, without the possibility of using a given index when executing a command. Due to the lack of suitability for query optimization from the latest version, the indexes will no longer be presented to the Performance Monitor application.

4.5. Shared Dashboard with the Performance Monitor application for SAP Hana

In the latest version of the application, support for the new SAP Hana data platform has been added. Users with multi-platform monitoring can view online all four available platforms on one common Dashboard screen.

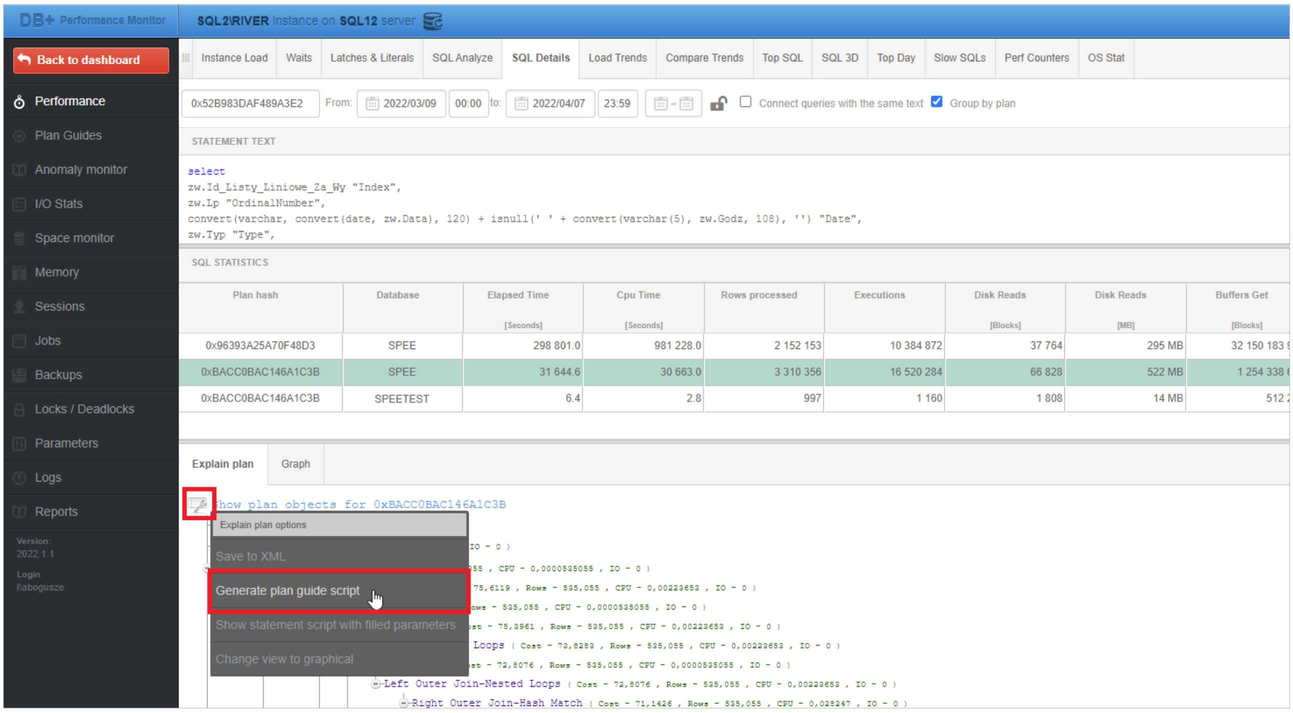
4.6. Fixed the problem with presenting detailed data on the lock screen

The problem with the presentation of lock session details on the Locks History page has been corrected. The problem was intermittent and related to a redundant character written to the repository, causing problems with displaying the data. The amendment only concerned the visual part, all data will be available retrospectively.

4.7. Create PlanGuide improvements

In the latest version of the application, we have improved the PlanGuide generation mechanism. In the Performance Monitor application, the user can easily and quickly assign an execution plan to the query in order to optimize the query. To do this, on the SQL Details page, select the plan for which you want to create a PlanGuide object, and then by clicking on the settings icon, select the **Generate Plan Guide script** option in the popup menu. The generated script should be copied and run from the level of a dedicated program on the indicated SQL instance.

In the latest version, we have improved the plan generation and the creation of PlanGuide for **Microsoft Azure databases** and **Express Edition** has been unlocked, as well as improved script generation when temporary objects are involved in the query.



0x52B983DAF489A3E2 From: 2022/03/09 00:00 to: 2022/04/07 23:59 ☐ Connect queries with the same text ☑ Group by plan

STATEMENT TEXT

```
select
zw.Id_Listy_Liniowe_Za_Wy "Index",
zw.Lp "OrdinalNumber",
convert(varchar, convert(date, zw.Data), 120) + isnull(' ' + convert(varchar(5), zw.Godz, 108), '') "Date",
zw.Typ "Type",
```

SQL STATISTICS

Plan hash	Database	Elapsed Time	Cpu Time	Rows processed	Executions	Disk Reads	Disk Reads	Buffers Get
		[Seconds]	[Seconds]			[Blocks]	[MB]	[Blocks]
0x96393A25A70F48D3	SPEE	298 801.0	981 228.0	2 152 153	10 384 872	37 764	295 MB	32 150 183
0xBACC0BAC146A1C3B	SPEE	31 644.6	30 663.0	3 310 356	16 520 284	66 828	522 MB	1 254 338
0xBACC0BAC146A1C3B	SPEETEST	6.4	2.8	997	1 160	1 808	14 MB	512

```
Show plan objects for 0xBACC0BAC146A1C3B
                                  IO - 0 )
                                  55 , CPU - 0,0000535055 , IO - 0 )
                                   75,6119 , Rows - 535,055 , CPU - 0,00223653 , IO - 0 )
                                  ows - 535,055 , CPU - 0,0000535055 , IO - 0 )
                                  st - 75,3961 , Rows - 535,055 , CPU - 0,00223653 , IO - 0 )
                                   Loops ( Cost - 73,8283 , Rows - 535,055 , CPU - 0,00223653 , IO - 0 )
                                  st - 72,8076 , Rows - 535,055 , CPU - 0,0000535055 , IO - 0 )
                      Left Outer Join-Nested Loops ( Cost - 72,8076 , Rows - 535,055 , CPU - 0,00223653 , IO - 0 )
                         Right Outer Join-Hash Match ( Cost - 71,1426 , Rows - 535,055 , CPU - 0,028247 , IO - 0 )
```