**Performance tuning**

**Inhalt**

[1 Database perf\_neartime 3](#_Toc138760037)

[**2.** **Tables and Information save in perf\_neartime** 4](#_Toc138760038)

[2.1 Server information 4](#_Toc138760039)

[2.2 Active sessions 5](#_Toc138760040)

[2.3 Active requests 8](#_Toc138760041)

[2.4 Active executing plans 10](#_Toc138760042)

[2.5 Databases and their sizes 11](#_Toc138760043)

[2.6 Blocking sessions 12](#_Toc138760044)

[2.7 Workload group statistics 13](#_Toc138760045)

[2.8 Active users to table 15](#_Toc138760046)

[2.10 All queries that have requested and are waiting for a memory grant or have been given memory grant 16](#_Toc138760047)

[2.11 Each request that is executing in SQL Server 17](#_Toc138760048)

[2.12 Current resource pool state, the current configuration of resource pools, and resource pool statistics 18](#_Toc138760049)

[2.13 Transactions for the instance of SQL Server 20](#_Toc138760050)

[2.14 Waits encountered by threads that executed 21](#_Toc138760051)

[2.15 Each scheduled job to be executed by SQL Server Agent 22](#_Toc138760052)

[2.16 Unique catalogue ID for collect data 22](#_Toc138760053)

[2.17 Current used memory 23](#_Toc138760054)

[2.18 Job history 24](#_Toc138760055)

[**3.** **Tables and Information not be used at the Moment, need Logic update** 25](#_Toc138760056)

[**4.** **Table and Fields** 25](#_Toc138760057)

[**5.** **Job which Triggers the Data load** 26](#_Toc138760058)

[**6.** **Check the Dataload by Job** 26](#_Toc138760059)

[**7.** **Procedure witch load the Data in the Tables trigger by Job** 26](#_Toc138760060)

[**8.** **Dataflow** 27](#_Toc138760061)

# **Database perf\_neartime**

The database collects real-time data by capturing system activities and storing them in various tables. This data is saved every 5 minutes to ensure an up-to-date record of system performance. The process of data saving is initiated by a job named "Performance\_Neartime" which executes a stored procedure called "StorePerformanceData" This job and stored procedure work together to retrieve the necessary information from the system and store it in the appropriate tables within the database.

The collected data provides valuable insights into the system's performance, allowing users to analyze and monitor various aspects of the system activities. By gathering data at regular intervals, it enables tracking changes and identifying trends over time. This information is crucial for system administrators and analysts to optimize system performance, troubleshoot issues, and make informed decisions regarding resource allocation and system configuration.

The data collected includes server information such as server name, version, login time, and session details. It also captures active requests, execution plans, database sizes, blocking sessions, workload group statistics, connected users, page allocation and deallocation activity, memory grants, executing queries, resource pool state, open transactions, wait status, scheduled jobs, and current memory usage. Each of these tables contains specific columns that hold relevant information related to the respective data category.

## **Tables and Information save in perf\_neartime**

Data collected from system tables of SQL Server. And write to tables of **perf\_neartime** database (see diagram). Collect and insert data produce using procedure **dbo.StorePerformancedata** which located in **master** database.

1. Server information

Source system table from where take data:

**sys.dm\_exec\_sessions** is a dynamic management view that provides information about all active user connections and internal tasks. This includes client information such as application and host names, security settings, configuration settings, and other session-specific data.

Procedure: **dbo.StorePerformancedata** uses internal logic to collect information from the system table **sys.dm\_exec\_sessions** and insert it in the **perf.perf\_exec\_serverinfos** table.

Destination table in **perf\_neartime** database — **perf.perf\_exec\_serverinfos**.

Fields description:

* **CatalogueID**: This column stores the unique identifier for the catalogue.
* **Servername**: This column holds the name of the server.
* **Version**: This column contains the version information of the server.
* **login\_time**: This column records the timestamp of when a session logs into the server.
* **SPID**: This column stores the Server Process ID of the session.
* **InfoLevel**: This column specifies the level of information being recorded.
* **ProfilerInfo**: This column holds any profiler information related to the session.
* **SpecialSet**: This column is used for any special settings for the session.

1. Active sessions

Source system table from where take data:

**sys.dm\_exec\_sessions** is a dynamic management view that provides information about all active user connections and internal tasks. This includes client information such as application and host names, security settings, configuration settings, and other session-specific data.

Procedure: **dbo.sp\_ins\_perf\_dm\_exec\_sessions** uses internal logic to collect information from the system table **sys.dm\_exec\_sessions** and insert it in the **perf.perf\_exec\_sessions** table.

Destination table in **perf\_neartime** database — **perf.perf\_exec\_sessions**.

Fields description:

* **CatalogueID**: This column stores the unique identifier for the catalogue.
* **date**: This column stores the date and time of the session.
* **month**\_id: This column holds the identifier for the month.
* **guid**: This column stores the globally unique identifier for the session.
* **session**\_id: This column stores the identifier for the session.
* **log**\_**time**: This column records the timestamp of when the session was logged.
* **host**\_**name**: This column holds the name of the host.
* **program**\_**name**: This column contains the name of the program being run in session.
* **host**\_**process**\_**id**: This column stores the Process ID of the host for the session.
* **client**\_**version**: This column contains the version of the client.
* **client**\_**interface**\_**name**: This column holds the name of the client interface.
* **security**\_**id**: This column stores the security identifier for the session.
* **login**\_**name**: This column contains the login name of the user.
* **nt\_domain**: This column stores the NT domain of the user.
* **nt\_user\_name**: This column holds the NT username of the user.
* **status:** This column specifies the status of the session.
* **context\_info**: This column contains any additional context information for the session**.**
* **cpu**\_**time**: This column records the CPU time consumed by the session.
* **memory**\_**usage**: This column indicates the memory usage of the session.
* **total\_schedulet\_time**: This column records the total scheduled time for the session.
* **total\_elapset\_time**: This column logs the total elapsed time of the session.
* **endpoint\_id**: This column stores the endpoint ID for the session.
* **last\_request\_start\_time**: This column records the start time of the last request.
* **last\_request\_end\_time**: This column records the end time of the last request.
* **reades**: This column contains the number of reads performed by the session.
* **writes**: This column contains the number of writes performed by the session.
* **logical\_reads**: This column contains the number of logical reads performed by the session.
* **is\_user\_process**: This column indicates whether the session is a user process.
* **text\_size**: This column stores the text size for the session.
* **language**: This column holds the language of the session.
* **date\_format**: This column contains the date format of the session.
* **date\_first**: This column specifies the first day of the week.
* **quoted\_identifier**: This column indicates whether the QUOTED\_IDENTIFIER setting is enabled.  
  **arithabort**: This column indicates whether the ARITHABORT setting is enabled.
* **ansi\_null\_dflt\_on**: This column specifies whether the ANSI\_NULL\_DFLT\_ON setting is enabled.
* **ansi\_defaults**: This column indicates whether the ANSI\_DEFAULTS setting is enabled.
* **ansi\_warnings**: This column specifies whether the ANSI\_WARNINGS setting is enabled.
* **ansi\_padding**: This column indicates whether the ANSI\_PADDING setting is enabled.
* **ansi\_nulls**: This column specifies whether the ANSI\_NULLS setting is enabled.
* **contact\_null\_yields\_null**: This column indicates whether the CONCAT\_NULL\_YIELDS\_NULL setting is enabled.
* **transaction\_isolation\_level**: This column stores the transaction isolation level for the session.
* **lock\_timeout**: This column records the lock timeout value for the session.
* **deadlock\_priority**: This column holds the deadlock priority for the session.
* **row\_count**: This column records the number of rows affected by the last statement in the session.
* **prev\_error**: This column stores the previous error number if there was one.
* **original\_security\_id**: This column stores the original security identifier for the session.
* **origonal\_login\_name**: This column contains the original login name for the session.
* **last\_successful\_logon**: This column records the last successful logon time for the session.
* **last\_unsuccessful\_logon**: This column records the last unsuccessful logon time for the session.
* **unsuccessful\_logons**: This column contains the number of unsuccessful logons for the session.
* **group\_id**: This column stores the group identifier for the session.
* **database\_id**: This column holds the database identifier for the session.
* **database\_name**: This column contains the name of the database for the session.
* **authenticating\_database\_id**: This column stores the ID of the database used for authentication.
* **open\_transation\_count**: This column records the number of open transactions in the session.

1. Active requests

Source system table from where take data:

**sys.dm\_exec\_requests** is a dynamic management view that provides information about each request that is executing within SQL Server. This includes requests from user sessions, system sessions and internal processes.

Procedure: **dbo.sp\_ins\_perf\_dm\_exec\_requests** uses internal logic to collect information from the system table **sys.dm\_exec\_requests** and insert it in the **perf.perf\_exec\_requests** table.

Destination table in **perf\_neartime** database — **perf.perf\_exec\_requests.**

Fields description:

* **CatalogueID**: This column stores the unique identifier for each catalogue entry.
* **date**: This column holds the timestamp when the entry is made.
* **month**\_**id**: This column represents the month associated with the date of the entry.
* **guid**: This column holds the globally unique identifier associated with the request.
* **session**\_**id**: This column stores the session identifier.
* **request**\_**id**: This column holds the unique identifier for the request.
* **start**\_**time**: This column records the timestamp when a request was initiated.
* **status**: This column indicates the current status of the request.
* **command**: This column stores the command associated with the request.
* **sql**\_**handle**: This column stores the SQL handle associated with the request.
* **statement**\_**start**\_**offset**: This column holds the start offset of the SQL statement in the batch.
* **statement**\_**end**\_**offset**: This column holds the end offset of the SQL statement in the batch.
* **plan**\_**handle**: This column stores the plan handle for the SQL statement.
* **database**\_**id**: This column holds the identifier of the database in which the request is made.
* **database**\_**name**: This column stores the name of the database in which the request is made.
* **user**\_**id**: This column holds the user identifier associated with the request.
* **connection**\_**id**: This column stores the identifier for the connection in which the request was made.
* **blocking**\_**session**\_**id**: This column holds the identifier of the blocking session, if any.
* **wait**\_**type**: This column indicates the type of wait associated with the request.
* **wait**\_**time**: This column records the amount of time the request has been waiting.
* **last**\_**wait**\_**type**: This column indicates the last type of wait that was recorded.
* **wait**\_**resource**: This column holds the resource for which the request is waiting.
* **open**\_**transaction**\_**count**: This column counts the number of open transactions associated with the request.
* **open**\_**result**\_**count**: This column counts the number of open results associated with the request.
* **transaction**\_**id**: This column holds the transaction identifier associated with the request.
* **contex**\_**info**: This column holds the context info of the session associated with the request.
* **percente**\_**complete**: This column holds the percentage of completion for the request.
* **estimated**\_**completion**\_**time**: This column holds the estimated time remaining for the request to complete.
* **cpu**\_**time**: This column records the amount of CPU time consumed by the request.
* **total**\_**elapsed**\_**time**: This column records the total time elapsed since the request was initiated.
* **scheduler**\_**id**: This column holds the identifier of the scheduler that the request is assigned to.
* **task**\_**address**: This column holds the address of the task associated with the request.
* **reads**: This column holds the number of reads by the request.
* **writes**: This column holds the number of writes by the request.
* **logical**\_**reades**: This column holds the number of logical reads by the request.
* **text**\_**size**: This column holds the size of the text data in the request.
* **language**: This column holds the language setting of the session associated with the request.
* **date\_format**: This column holds the date format setting of the session associated with the request.
* **date\_first**: This column holds the setting for the first day of the week for the session associated with the request.
* **quoted\_identifier**: This column indicates whether the QUOTED\_IDENTIFIER setting is ON or OFF for the session.
* **arithabort**: This column indicates whether the ARITHABORT setting is ON or OFF for the session.
* **ansi\_null\_dflt\_on**: This column indicates whether the ANSI\_NULL\_DFLT\_ON setting is ON or OFF for the session.
* **ansi\_defaults**: This column indicates whether the ANSI\_DEFAULTS setting is ON or OFF for the session.
* **ansi\_warnings**: This column indicates whether the ANSI\_WARNINGS setting is ON or OFF for the session.
* **ansi\_padding**: This column indicates whether the ANSI\_PADDING setting is ON or OFF for the session.
* **ansi\_nulls**: This column indicates whether the ANSI\_NULLS setting is ON or OFF for the session.
* **contact\_null\_yields\_null**: This column indicates whether the CONCAT\_NULL\_YIELDS\_NULL setting is ON or OFF for the session.
* **transaction\_isolation\_level**: This column holds the transaction isolation level for the session.
* **lock\_timeout**: This column holds the lock timeout setting for the session.
* **deadlock\_priority**: This column holds the deadlock priority for the session.
* **row\_count**: This column holds the number of rows affected by the request.

1. Active executing plans

Source system table from where take data:

**sys.dm\_exec\_query\_plan** is a dynamic management function that returns the query plan that is being executed by the specified handle. The plan includes the steps SQL Server takes to execute a query, such as which tables are scanned, which indexes are used, whether a parallel plan is applied, and more.

Procedure: **dbo.StorePerformancedata** uses internal logic to collect information from the system table **sys.dm\_exec\_query\_plan** and insert it in the **perf.executing\_plans** table.

Destination table in **perf\_neartime** database — **perf.executing\_plans.**

Fields description:

* **CatalogueID**: This column stores the unique identifier for each catalogue entry.
* **Plan ##SPID##**: This column stores the Server Process ID related to the executing plan.
* **ObjektName**: This column contains the name of the object associated with the executing plan.
* **query\_plan**: This column contains the XML representation of the query plan.
* **BatchStmt**: This column holds the batch statement related to the executing plan.
* **SubStmt**: This column stores the sub-statements, if any, associated with the batch statement.
* **SubStmtPlan**: This column holds the XML representation of the plan associated with the sub-statement.
* **SubStmtPlanText**: This column holds the textual representation of the plan associated with the sub-statement.
* **plan\_handle**: This column stores the handle to the plan associated with the executing plan.
* **Input\_Buffer**: This column stores the input buffer related to the executing plan.
* **Buffer\_Parameters**: This column represents the number of parameters in the input buffer.

1. Databases and their sizes

Source system table from where take data:

**sys.dm\_os\_performance\_counters** is a dynamic management view that provides access to the performance counter values for SQL Server processes. These performance counters include various system statistics such as SQL errors, batch requests, compilations and recompilations, user connections, transactions, lock requests, logins and logouts, and various other performance-related counters.

Procedure: **dbo.StorePerformancedata** uses internal logic to collect information from the system table **sys.dm\_os\_performance\_counters** and insert it in the **perf.db\_usage** table.

Destination table in **perf\_neartime** database — **perf.db\_usage.**

Fields description:

* **CatalogueID**: This column stores the unique identifier for each catalogue entry.
* **DB Size**: This column contains the size information of the database.
* **Data file (MB)**: This column holds the size of the data file associated with the database in megabytes.
* **Log file (MB)**: This column holds the size of the log file associated with the database in megabytes.
* **Log file space used (MB)**: This column holds the amount of space used in the log file in megabytes.
* **LogPercent**: This column represents the percentage of log file space used.

1. Blocking sessions

Source system table from where take data:

**sys.dm\_exec\_requests** is a dynamic management view that provides information about all active user connections and internal tasks. This includes client information such as application and host names, security settings, configuration settings, and other session-specific data.

Procedure: **dbo.sp\_ins\_perf\_blocking\_sessions** uses internal logic to collect information from the system table **sys.dm\_exec\_sessions** and insert it in the **perf.blocking\_sessions** table.

Destination table in **perf\_neartime** database — **perf.blocking\_sessions.**

Fields description:

* **CatalogueID**: This column stores the unique identifier for each catalogue entry.
* **date**: This column holds the timestamp when the entry is made.
* **month\_id**: This column represents the month associated with the date of the entry.
* **guid**: This column holds the globally unique identifier associated with the session.
* **session\_id**: This column stores the identifier of the blocking session.
* **login\_name**: This column stores the login name associated with the session.
* **blocking\_session\_id**: This column stores the identifier of the session that blocked.
* **chain\_id**: This column stores the chain identifier the blocking session.
* **status**: This column indicates the current status of the session.
* **host\_name**: This column stores the host name of the session.
* **database\_name**: This column stores the name of the database in the session.
* **command**: This column stores the command being executed by the session.
* **cpu\_time**: This column records the amount of CPU time consumed by the session.
* **disk\_to**: This column stores the disk usage of the session.
* **last\_batch**: This column stores the last batch executed by the session.
* **program\_name**: This column stores the program name associated with the session.
* **spid2**: This column stores the second SPID associated with the blocking session.
* **request\_id**: This column stores the unique identifier for the request in the session.
* **blocking\_guid**: This column holds the globally unique identifier blocking session.

1. Workload group statistics

Source system table from where take data:

**sys.dm\_resource\_governor\_workload\_groups** is a dynamic management view that provides detailed information about Resource Governor workload groups. These workload groups represent a collection of sessions that have similar classification criteria and resource requirements.

Procedure: **dbo.ins\_perf\_dm\_resource\_governor\_workload\_groups** uses internal logic to collect information from the system table **sys.dm\_resource\_governor\_workload\_groups** and insert it in the **perf.dm\_resource\_governor\_workload\_groups** table.

Destination table in **perf\_neartime** database — **perf.dm\_resource\_governor\_workload\_groups.**

Fields description:

* **CatalogueID**: This column stores the unique identifier for each catalogue entry.
* **date**: This column holds the timestamp when the entry is made.
* **month\_id**: This column represents the month associated with the date of the entry.
* **guid**: This column holds the globally unique identifier associated with the group.
* **group\_id**: This column stores the identifier of the workload group.
* **name**: This column stores the name of the workload group.
* **pool\_id**: This column stores the identifier of the resource pool associated with the workload group.
* **external\_pool\_id**: This column stores the identifier of the external resource pool associated with the workload group.
* **statistic\_start\_time**: This column holds the start time of the statistics collection.
* **total\_request\_count**: This column stores the total number of requests processed by the workload group.
* **total\_queued\_request\_count**: This column stores the total number of requests queued for the workload group.
* **active\_request\_count**: This column stores the number of currently active requests in the workload group.
* **queued\_request\_count**: This column stores the number of requests currently queued for execution in the workload group.
* **total\_cpu\_limit\_violation\_count**: This column stores the total number of CPU limit violations for the workload group.
* **total\_cpu\_usage\_ms**: This column stores the total CPU usage in milliseconds by the workload group.
* **max\_request\_cpu\_time\_ms**: This column stores the maximum CPU time in milliseconds for a single request in the workload group.
* **blocked\_task\_count**: This column stores the number of blocked tasks in the workload group.
* **total\_lock\_wait\_count**: This column stores the total number of lock waits for the workload group.
* **total\_lock\_wait\_time\_ms**: This column stores the total lock wait time in milliseconds for the workload group.
* **total\_query\_optimization\_count**: This column stores the total number of query optimization operations for the workload group.
* **total\_suboptimal\_plan\_generation\_count**: This column stores the total number of suboptimal plan generation for the workload group.
* **total\_reduced\_memgrant\_count**: This column stores the total number of reduced memory grants for the workload group.
* **max\_request\_grant\_memory\_kb**: This column stores the maximum memory grant in kilobytes for a single request in the workload group.
* **active\_parallel\_thread\_count**: This column stores the number of active parallel threads in the workload group.
* **importance**: This column indicates the importance level of the workload group.
* **request\_max\_memory\_grant\_percent**: This column stores the maximum memory grant percent for a request in the workload group.
* **request\_max\_cpu\_time\_sec**: This column stores the maximum CPU time in seconds for a request in the workload group.
* **request\_memory\_grant\_timeout\_sec**: This column stores the memory grant timeout in seconds for a request in the workload group.
* **group\_max\_requests**: This column stores the maximum number of requests allowed in the workload group.
* **max\_drop**: This column stores the maximum number of requests dropped from the workload group.
* **effective\_max\_drop**: This column stores the effective maximum number of requests dropped from the workload group.
* **total\_cpu\_usage\_preemptive\_ms**: This column stores the total CPU usage in milliseconds for preemptive tasks in the workload group.

1. Active users

Source system table from where take data:

**sys.dm\_exec\_sessions** is a dynamic management view that provides information about all active user connections and internal tasks. This includes client information such as application and host names, security settings, configuration settings, and other session-specific data.

Procedure: **dbo.sp\_ins\_perf\_connected\_users** uses internal logic to collect information from the system table **sys.dm\_exec\_sessions** and insert it in the **perf.connected\_users** table.

Destination table in **perf\_neartime** database — **perf.connected\_users.**

Fields description:

* **CatalogueID**: This column stores the unique identifier for each catalogue entry.
* **date**: This column holds the timestamp when the entry is made.
* **month\_id**: This column represents the month associated with the date of the entry.
* **guid**: This column holds the globally unique identifier associated with the user.
* **original\_login\_name**: This column stores the original login name of the user.
* **session\_count**: This column stores the number of active sessions users.

1. Page allocation and deallocation activity by task for the database

Source system table from where take data:

**sys.dm\_db\_task\_space\_usage** is a dynamic management view that provides information about the disk space usage, in tempdb, for both active and recently completed tasks. It is important when you are trying to track tempdb consumption at a task level and understand which tasks are causing tempdb to grow.

Procedure: **dbo.sp\_ins\_perf\_dm\_db\_task\_space\_usage** uses internal logic to collect information from the system table **sys.dm\_db\_task\_space\_usage** and insert it in the **perf.dm\_db\_task\_space\_usage** table.

Destination table in **perf\_neartime** database — **perf.dm\_db\_task\_space\_usage.**

Fields description:

* **CatalogueID**: This column stores the unique identifier for each catalogue entry.
* **date**: This column holds the timestamp when the entry is made.
* **month\_id**: This column represents the month associated with the date of the entry.
* **guid**: This column holds the globally unique identifier associated with the task.
* **session\_id**: This column stores the identifier of the session associated with the task.
* **request\_id**: This column stores the unique identifier for the request associated with the task.
* **temp\_usage\_gb**: This column stores the temporary usage in gigabytes for the task.

1. All queries that have requested and are waiting for a memory grant or have been given memory grant

Source system table from where take data:

**sys.dm\_exec\_query\_memory\_grants** is a dynamic management view that provides information about the memory granted to the executing queries. This information includes the ID of the session that submitted the request, the ID of the request, whether the request has been granted memory, and how much memory it has been granted or still requires.

Procedure: **dbo.ins\_perf\_dm\_db\_task\_space\_usage** uses internal logic to collect information from the system table **sys.dm\_db\_task\_space\_usage** and insert it in the **perf.dm\_exec\_query\_memory\_grants** table.

Destination table in **perf\_neartime** database — **perf.dm\_exec\_query\_memory\_grants.**

Fields description:

* **CatalogueID**: This column stores the unique identifier for each catalogue entry.
* **date**: This column holds the timestamp when the entry is made.
* **month\_id**: This column represents the month associated with the date entry.
* **guid**: This column holds the globally unique identifier associated with the query.
* **session\_id**: This column stores the identifier of the session associated with query.
* **request\_id**: This column stores the unique identifier for request associated query.
* **scheduler\_id**: This column stores the identifier of the scheduler associated query.
* **dop**: This column stores the degree of parallelism for the query.
* **request\_time**: This column holds the timestamp when the memory grant.
* **grant\_time**: This column holds the timestamp when the memory grant was granted.
* **requested\_memory\_kb**: This column stores the amount of memory requested.
* **granted\_memory\_kb**: This column stores the amount of memory granted.
* **required\_memory\_kb**: This column stores the amount of memory required.
* **used\_memory\_kb**: This column stores the amount of memory used in kilobytes.
* **max\_used\_memory\_kb**: This column stores the maximum amount of memory used
* **query\_cost**: This column stores the cost associated with the query.
* **timeout\_sec**: This column stores the timeout value in seconds.
* **resource\_semaphore\_id**: This column stores the identifier of the resource.
* **queue\_id**: This column stores the identifier of the queue associated with the query.
* **wait\_order**: This column stores the wait order of the query.
* **is\_next\_candidate**: This column indicates whether the query is the next candidate.
* **wait\_time\_ms**: This column stores the wait time in milliseconds.
* **plan\_handle**: This column stores the handle to the plan associated with the query.
* **sql\_handle**: This column stores the handle to the SQL text associated with query.
* **group\_id**: This column stores the identifier of the group associated with the query.
* **pool\_id**: This column stores the identifier of the pool associated with the query.
* **is\_small**: This column indicates whether the query is considered a small query.
* **ideal\_memory\_kb**: This column stores the ideal memory size in kilobytes for query.
* **reserved\_worker\_count**: This column stores the number of reserved workers.

1. Each request that is executing in SQL Server

Source system table from where take data:

**sys.dm\_exec\_query\_memory\_grants** is a dynamic management view that provides information about the memory granted to the executing queries. This information includes the ID of the session that submitted the request, the ID of the request, whether the request has been granted memory, and how much memory it has been granted or still requires.

Procedure: **dbo.ins\_perf\_dm\_db\_task\_space\_usage** uses internal logic to collect information from the system table **sys.dm\_db\_task\_space\_usage** and insert it in the **perf.dm\_exec\_requests** table.

Destination table in **perf\_neartime** database — **perf.dm\_exec\_requests.**

Fields description:

* + - **CatalogueID**: This column stores the unique identifier for each catalogue entry.
    - **date**: This column holds the timestamp when the request is made.
    - **month\_id**: This column represents the month associated with the date of request.
    - **guid**: This column holds the globally unique identifier associated with the request.
    - **session\_id**: This column stores the identifier of the session associated with request.
    - **request**\_id: This column stores the unique identifier for the request.
    - **start\_time**: This column records the timestamp when the request started.
    - **status**: This column indicates the current status of the request.
    - **command**: This column holds the command being executed by the request.
    - **sql\_handle**: This column stores the handle to the SQL text associated with request.
    - **plan\_handle**: This column stores the handle to the plan associated with the request.
    - **database\_id**: This column stores the identifier of database associated with request.
    - **database\_name**: This column stores the name of database associated with request.
    - **user\_id**: This column stores the identifier of the user associated with the request.
    - **connection\_id**: This column stores the unique identifier for the connection.
    - **blocking\_session\_id**: This column stores the identifier of the session.
    - **wait\_type**: This column stores the type of wait the request is currently experiencing.
    - **wait\_time**: This column stores the wait time in milliseconds for the request.
    - **last\_wait\_type**: This column stores the last wait type experienced by the request.
    - **wait\_resource**: This column stores the resource the request is waiting for.
    - **open\_transaction\_count**: This column stores the number of open transactions
    - **transaction\_id**: This column stores identifier of transaction associated with request.
    - **context\_info**: This column stores the context information associated with request.
    - **percent\_complete**: This column stores the percentage of completion for request.
    - **estimated\_completion\_time**: This column stores the estimated completion time.
    - **cpu\_time**: This column stores the CPU time in milliseconds consumed by request.
    - **total\_elapsed\_time**: This column stores the total elapsed time for the request.
    - **scheduler\_id**: This column stores identifier of scheduler associated with request.
    - **task\_address**: This column stores address of the task associated with request.
    - **reads**: This column stores the number of physical reads performed by the request.
    - **writes**: This column stores the number of physical writes performed by the request.

1. Current resource pool state, the current configuration of resource pools, and resource pool statistics

Source system table from where take data:

**sys.dm\_exec\_query\_memory\_grants** is a dynamic management view that provides information about the memory granted to the executing queries. This information includes the ID of the session that submitted the request, the ID of the request, whether the request has been granted memory, and how much memory it has been granted or still requires.

Procedure: **dbo.ins\_perf\_dm\_db\_task\_space\_usage** uses internal logic to collect information from the system table **sys.dm\_db\_task\_space\_usage** and insert it in the **perf.dm\_ressorce\_governor\_resource\_pools** table.

Destination table in **perf\_neartime** database — **perf.dm\_ressorce\_governor\_resource\_pools.**

Fields description:

* **CatalogueID** : This column stores the unique identifier for each catalogue entry.
* **date** : This column holds the timestamp when the entry is made.
* **month\_id** : This column represents the month associated with the date of the entry.
* **guid** : This column holds the globally unique identifier associated with resource pool.
* **pool\_id** : This column stores the identifier of the resource pool.
* **name** : This column stores the name of the resource pool.
* **statistic\_start\_time** : This column holds the timestamp when the statistics for the resource pool started.
* **total\_cpu\_usage\_ms** : This column stores the total CPU usage in milliseconds for the resource pool.
* **cache\_memory\_kb** : This column stores the amount of cache memory in kilobytes for the resource pool.
* **complete\_memory\_kb** : This column stores the complete memory in kilobytes for the resource pool.
* **used\_memgrant\_kb** : This column stores the amount of memory grant used in kilobytes for the resource pool.
* **total\_memgrant\_count** : This column stores the total number of memory grants for the resource pool.
* **total\_memgrant\_timeout\_count** : This column stores the total number of memory grant timeouts for the resource pool.
* **active\_memgrant\_count** : This column stores the number of active memory grants for the resource pool.
* **active\_memgrant\_kb** : This column stores the amount of active memory grant in kilobytes for the resource pool.
* **memgrant\_waiter\_count** : This column stores the number of memory grant waiters for the resource pool.
* **max\_memory\_kb** : This column stores the maximum memory in kilobytes for the resource pool.
* **used\_memory\_kb** : This column stores the used memory in kilobytes for the resource pool.
* **target\_memory\_kb** : This column stores the target memory in kilobytes for the resource pool.
* **out\_of\_memory\_count** : This column stores the total number of out-of-memory conditions for the resource pool.
* **min\_cpu\_percent** : This column stores the minimum CPU percentage for the resource pool.
* **max\_cpu\_percent** : This column stores the maximum CPU percentage for the resource pool.
* **cap\_cpu\_percent** : This column stores the capped CPU percentage for the resource pool.
* **min\_iops\_per\_volume** : This column stores the minimum IOPS per volume for the resource pool.
* **max\_iops\_per\_volume** : This column stores the maximum IOPS per volume for the resource pool.
* **read\_io\_queued\_total** : This column stores the total number of read I/Os queued for the resource pool.
* **read\_io\_issued\_total** : This column stores the total number of read I/Os issued for the resource pool.
* **read\_io\_completed\_total :** This column stores the total number of completed read I/Os for the resource pool.
* **read\_io\_throttled\_total** : This column stores the total number of throttled read I/Os for the resource pool.
* **read\_bytes\_total** : This column stores the total number of read bytes for the resource pool.
* **read\_io\_stall\_total\_ms** : This column stores the total I/O stall time in milliseconds for read operations in the resource pool.
* **read\_io\_stall\_queued\_ms** : This column stores the total I/O stall time in milliseconds for queued read operations in the resource pool.
* **write\_io\_queued\_total** : This column stores the total number of write I/Os queued for the resource pool.
* **write\_io\_issued\_total** : This column stores the total number of write I/Os issued for the resource pool.
* **write\_io\_completed\_total** : This column stores the total number of completed write I/Os for the resource pool.
* **write\_io\_throttled\_total** : This column stores the total number of throttled write I/Os for the resource pool.
* **write\_bytes\_total** : This column stores the total number of write bytes for the resource pool.
* **write\_io\_stall\_total\_ms** : This column stores the total I/O stall time in milliseconds for write operations in the resource pool.
* **write\_io\_stall\_queued\_ms** : This column stores the total I/O stall time in milliseconds for queued write operations in the resource pool.

1. Transactions for the instance of SQL Server

Source system table from where take data:

**sys.sysprocesses:** This is a system compatibility view provided for backward compatibility. It provides information about the processes that are running on an instance of SQL Server. This includes details like the session ID, status, user and system process indicator, the database in use by the process, and the command currently being executed.

**sys.dm\_tran\_active\_transactions:** This is a dynamic management view that provides information about the active transactions in the instance of SQL Server. The details include the transaction ID, the name of the transaction, transaction begin time, state, type, and whether it's a distributed transaction or not.

**sys.dm\_tran\_database\_transactions**: This dynamic management view provides information about the transactions at the database level. It includes details such as the database ID, transaction ID, type of transaction, state of transaction, and various other details related to the log, such as the log record count, and log bytes used and reserved.

Procedure: **dbo.sp\_ins\_perf\_open\_transactions** uses internal logic to collect information from the system table **sys.dm\_db\_task\_space\_usage** and insert it in the **perf\_neartime.perf.open\_transactions** table.

Destination table in **perf\_neartime** database — **perf\_neartime.perf.open\_transactions.**

Fields description:

* **CatalogueID** : This column stores the unique identifier for each catalogue entry.
* **date** : This column holds the timestamp when the entry is made.
* **month\_id** : This column represents the month associated with the date of the entry.
* **guid** : This column holds the globally unique identifier associated with transaction.
* **status** : This column stores the status of the transaction.
* **session\_id** : This column stores the identifier of session associated with transaction.
* **database\_name** : This column stores name database associated with transaction.
* **open\_transaction\_count** : This column stores number of open transactions.
* **login\_name** : This column stores the login name associated with the session.
* **login\_time** : This column stores the timestamp of when the session logged in.
* **last\_request\_start\_time** : This column stores the timestamp of the start time of the last request in the session.
* **last\_request\_end\_time** : This column stores the timestamp of the end time.
* **writes** : This column stores the number of writes performed in the transaction.
* **program\_name** : This column stores the name of program associated with session.
* **host\_name** : This column stores the name of the host associated with the session.
* **row\_count** : This column stores the number of rows affected by the transaction.
* **transaction\_isolation\_level** : This column stores the isolation level of transaction.
* **transaction\_begin\_time** : This column stores the timestamp.

1. Waits encountered by threads that executed

Source system table from where take data:

**sys.dm\_os\_wait\_stats** is a dynamic management view that provides information about all the waits encountered by threads that executed in your SQL Server instance. It summarizes the total wait time, the maximum wait time, and the wait count for each type of wait.

Procedure: **dbo.sp\_ins\_perf\_wait\_status\_log** uses internal logic to collect information from the system table **sys.dm\_os\_wait\_stats** and insert it in the **perf.wait\_status\_log** table.

Destination table in **perf\_neartime** database — **perf.wait\_status\_log.**

Fields description:

* **CatalogueID** : This column stores the unique identifier for each catalogue entry.
* **date** : This column holds the timestamp when the entry is made.
* **month\_id** : This column represents the month associated with the date of the entry.
* **wait\_type** : This column stores the type of wait encountered by the thread.
* **waiting\_task\_count** : This column stores number of tasks waiting for wait type.
* **wait\_time\_ms** : This column stores the total wait time for the specified wait type.
* **max\_wait\_time\_**ms : This column stores maximum wait time for specified wait type.
* **signal\_wait\_time\_ms** : This column stores signal wait time for specified wait type.

1. Each scheduled job to be executed by SQL Server Agent

Source system table from where take data:

**msdb.dbo.sysjobs** is a table in the MSDB database that provides information about the jobs that are managed by SQL Server Agent. These jobs are user-defined, and can perform various tasks such as database maintenance, backups, or running Transact-SQL scripts, SSIS packages, or command line applications.

Procedure: **dbo.sp\_mrg\_sched\_job** uses internal logic to collect information from the system table **msdb.dbo.sysjobs** and insert it in the **sched.job** table.

Destination table in **perf\_neartime** database — **sched.job.**

Fields description:

* **CatalogueID** : This column stores the unique identifier for each catalogue entry.
* **ident\_code** : This column stores the identifier code for the scheduled job.
* **name** : This column stores the name of the scheduled job.
* **description** : This column stores the description of the scheduled job.
* **enabled** : This column indicates whether the scheduled job is enabled or disabled.

1. Unique catalogue ID for collect data

Source system table from where take data:

Internal logic in procedure **dbo.StorePerformancedata** generating catalogue ID.

Procedure: **dbo.StorePerformancedata** uses internal logic to generating catalogue ID to table **perf.perf\_exec\_Catalogue**.

Destination table in **perf\_neartime** database — **perf.perf\_exec\_Catalogue.**

Fields description:

* **ID** : This column stores the unique identifier for each entry in the catalogue.
* **DATE**: This column stores the date associated with the entry.
* **TIME**: This column stores the time associated with the entry.

1. Current used memory

Source system table from where take data:

**sys.dm\_exec\_query\_memory\_grants** is a dynamic management view that provides information about the memory granted to the executing queries. This information includes the ID of the session that submitted the request, the ID of the request, whether the request has been granted memory, and how much memory it has been granted or still requires.

Procedure: **dbo.StorePerformancedata** uses internal logic to collect information from the system table **sys.dm\_exec\_query\_memory\_grants** and insert it in the **perf\_exec\_used\_memory** table.

Destination table in **perf\_neartime** database — **perf\_exec\_used\_memory.**

Fields description:

* **CatalogueID** : This column stores the unique identifier for each catalogue entry.
* **ExpressMEM ##SPID##** : This column stores the Express Memory value associated with the session (identified by SPID).
* **program\_name** : This column stores program name associated with session.
* **granted\_memory\_kb** : This column stores the granted memory in kilobytes.
* **subStmt** : This column stores the substatement associated with the session.
* **login\_name** : This column stores the login name associated with the session.
* **requested\_memory\_gb** : This column stores the requested memory.
* **granted\_memory\_gb** : This column stores the granted memory in gigabytes.
* **required\_memory\_gb** : This column stores the required memory in gigabytes.
* **used\_memory\_gb** : This column stores the used memory in gigabytes.
* **max\_used\_memory\_gb** : This column stores the maximum used memory.
* **query\_cost** : This column stores the query cost associated with the session.
* **resource\_semaphore\_id** : This column stores the ID of the resource semaphore associated with the session.
* **queue\_id** : This column stores the ID of the queue associated with the session.
* **wait\_order** : This column stores the wait order of the session.
* **is\_next\_candidate** : This column indicates whether session is next candidate.
* **wait\_time\_ms** : This column stores the wait time for the session.
* **dop** : This column stores the degree of parallelism for the session.

1. Job history

Source system table from where take data:

**msdb.dbo.sysjobhistory** is a table in the MSDB database used by SQL Server Agent to store the history of executed jobs. It includes information about each execution of a job, such as the job ID, step ID, the date and time it ran, its duration, and the outcome (success or failure), along with any message generated.

Procedure: **dbo.sp\_mrg\_sched\_job\_history** uses internal logic to collect information from the system table **msdb.dbo.sysjobhistory** and insert it in the **sched.systbl\_job\_history** table.

Destination table in **perf\_neartime** database — **sched.systbl\_job\_history.**

Fields description:

* **instance\_id**: A unique identifier for each instance of the job execution.
* **job\_id**: The unique identifier for the job. This is a GUID (globally unique identifier).
* **step\_id**: The ID of the step within the job. This ID is used to order the steps for execution.
* **step\_name**: The name of the step within the job.
* **sql\_message\_id**: The ID of any error message returned by the job. 0 if there was no error.
* **sql\_severity**: The severity level of any error message returned by the job. 0 if there was no error.
* **message**: The message text of any error or informational message returned by the job.
* **run\_status**: The status of the job execution (0 = Failed, 1 = Succeeded, 2 = Retry, 3 = Canceled).
* **run\_date**: The date the job execution started, stored as an integer in the format YYYYMMDD.
* **run\_time**: The time the job execution started, stored as an integer in the format HHMMSS.
* **run\_duration**: The duration of the job execution, stored as an integer in the format HHMMSS.
* **operator\_id\_emailed**: The ID of the operator to whom an email was sent when the job completed.
* **operator\_id\_netsent**: The ID of the operator to whom a net send was sent when the job completed.
* **operator\_id\_paged**: The ID of the operator who was paged when the job completed.
* **retries\_attempted**: The number of retry attempts made to run the job.
* **server**: The name of the server where the job executed.

## **Tables and Information not be used at the Moment, need Logic update**

Tables which present in database but logic of procedure not use it as destination is reserve tables for other logic:

* Resource Waits filtered to blacklist table — **perf\_neartime.wait\_type\_blacklist**
* Status of session table — **perf\_neartime.session\_status**
* Name version of server and server start time table — **perf\_neartime.server\_daten**
* Information about killed sessions table — **perf\_neartime.killed\_sessions**
* Information for killed sessions table — **perf\_neartime.tmp.kill\_config**
* Information for killed sessions users table — **perf\_neartime.tmp.kill\_whitelist**
* Mail on SQL Servertable — **perf\_neartime.tmp.mail**
* Mail notifications on SQL Servertable — **perf\_neartime.tmp.mail\_notification**
* Notifications on SQL Servertable — **perf\_neartime.tmp.tempdb\_notification**
* Time of warnings table — **perf\_neartime.tmp.warning\_timestamp**

## **Table and Fields**

Detailed description of tables and fields see in file:



## **Job which Triggers the Data load**

The execution of the stored procedure dbo.StorePerformancedata is managed by the job

"Performance\_Neartime" This job is scheduled to run every day, every 5 minutes, starting at 12:00:00 PM and ending at 11:59:59 AM. The purpose of this scheduled job is to ensure regular and timely execution of the stored procedure for collecting and storing performance data from the system. By running at frequent intervals throughout the day, it allows for a comprehensive and up-to-date record of system activities and performance metrics.

## **Check the Dataload by Job**

Another job named "Run\_Performance\_Neartime" controls the execution of the

"Performance\_Neartime " job to ensure that it is always running. This job periodically checks the status of the " Performance\_Neartime " job and takes necessary actions to restart it if it is not running or has encountered any issues. By monitoring the execution of the main job, the " Run\_Performance\_Neartime " job ensures the continuous and uninterrupted collection of performance data from the system.

## **Procedure witch load the Data in the Tables trigger by Job**

The dbo.StorePerformancedata procedure executes several other procedures to collect data:

* Executing procedure perf\_neartime.dbo.ins\_perf\_blocking\_sessions to collect data about blocking sessions and store it in the table perf\_neartime.perf.blocking\_sessions.
* Executing procedure perf\_neartime.dbo.ins\_perf\_dm\_exec\_sessions to collect data about active sessions and store it in the table perf\_neartime.perf.dm\_exec\_sessions.
* Executing procedure ins\_perf\_dm\_resource\_governor\_workload\_groups to collect data about workload group statistics and store it in the table perf\_neartime.perf.dm\_resource\_governor\_workload\_groups.
* Executing procedure perf\_neartime.dbo.ins\_perf\_connected\_users to collect data about active users and store it in the table perf\_neartime.perf.connected\_users.
* Executing procedure perf\_neartime.dbo.ins\_perf\_dm\_db\_task\_space\_usage to collect data about page allocation and deallocation activity by task for the database and store it in the table perf\_neartime.perf.dm\_db\_task\_space\_usage.
* Executing procedure perf\_neartime.dbo.ins\_perf\_dm\_exec\_query\_memory\_grants to collect data about queries that have requested and are waiting for a memory grant or have been granted memory grant and store it in the table perf\_neartime.dm\_exec\_query\_memory\_grants.
* Executing procedure perf\_neartime.dbo.ins\_perf\_dm\_exec\_requests to collect data about each executing request in SQL Server and store it in the table perf\_neartime.perf.dm\_exec\_requests.
* Executing procedure perf\_neartime.ins\_perf\_dm\_ressorce\_governor\_resource\_pools to collect data about the current resource pool state, the configuration of resource pools, and resource pool statistics and store it in the table perf\_neartime.perf.dm\_ressorce\_governor\_resource\_pools.
* Executing procedure perf\_neartime.dbo.ins\_perf\_open\_transactions to collect data about transactions for the instance of SQL Server and store it in the table perf\_neartime.perf.open\_transactions.
* Executing procedure perf\_neartime.dbo.ins\_perf\_wait\_status\_log to collect data about the waits encountered by threads and store it in the table perf\_neartime.perf.wait\_status\_log.
* Executing procedure perf\_neartime.dbo.sp\_ins\_perf\_wait\_type\_blacklist to collect data about wait types that are significant for performance analysis in SQL Server. It stores this information, alongside a provided catalogue ID, into the table perf\_neartime.perf.systbl\_perf\_wait\_type\_blacklist.
* Executing procedure perf\_neartime.dbo.sp\_ins\_perf\_sql\_infos to collect data about SQL execution details, including SQL and query plan handles, the corresponding text of the SQL commands, and their query plans. This information is tied to a provided catalogue ID and includes the current date and month. The gathered data is stored in the table perf\_neartime.perf.systbl\_perf\_sql\_infos.
* Executing procedure perf\_neartime.dbo.sp\_ins\_perf\_killed\_sessions to collect information about the session details from SQL Server's dynamic management view sys.dm\_exec\_sessions. A unique GUID is created for each session, and this information, along with the session ID, current date, month, and provided catalogue ID is stored in the table perf\_neartime.perf.systbl\_perf\_killed\_sessions.
* Executing procedure perf\_neartime.dbo.sp\_mrg\_sched\_job to collect data about each scheduled job to be executed by SQL Server Agent and store it in the table perf\_neartime.sched.systbl\_job.
* Procedure perf\_neartime.dbo.sp\_mrg\_sched\_job\_scheduler merges SQL Server Agent job scheduling details into the perf\_neartime.sched.systbl\_job\_scheduler table.
* Procedure perf\_neartime.dbo.sp\_mrg\_sched\_job\_step merges SQL Server Agent job step details into the perf\_neartime.sched.systbl\_job\_step table.
* Procedure dbo.sp\_mrg\_sched\_job\_history merges the history of job executions from msdb.dbo.sysjobhistory into the perf\_neartime.sched.systbl\_job\_history table.

These procedures are called within the dbo.StorePerformancedata procedure to gather specific data from different aspects of the SQL Server system and store it in the corresponding tables for further analysis and monitoring.

## **Dataflow**

Dataflow how information collected see below:

|  |  |  |
| --- | --- | --- |
| **Quelle** | **Quelle** | **Quelle** |
| **Quelle** | **Quelle** | **Quelle** |
| **Quelle** | **Quelle** | **Quelle** |
| **Quelle** | **Quelle** | **Quelle** |
| **Quelle** | **Quelle** | **Quelle** |
| **Quelle** | **Quelle** | **Quelle** |
| **Quelle** | **Quelle** | **Quelle** |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Quelle** | **Quelle** |  |

Detail information about source and destination see on Diagram:

