**Query store results description**

**Overview:**

1. [Table perf.perf\_exec\_Catalogue](#Catalog)
2. [Table perf.perf\_exec\_used\_memory](#Used_memory)
3. [Table perf.perf\_exec\_serverinfos](#Serverinfos)
4. [Table perf\_exec\_blocked\_sessions](#Blocked_sessions)
5. [Table perf\_exec\_sessions](#Sessions)
6. [Table perf\_exec\_requests](#Requests)
7. [Table perf.blocking\_sessions](#Blocking_sessions)
8. [Table perf.connected\_users](#Connected_users)
9. [Table perf.dm\_db\_task\_space\_usage](#Task_space_usage)
10. [Table perf.dm\_exec\_query\_memory\_grants](#Memory_grants)
11. [Table perf.dm\_exec\_requests](#DM_requests)
12. [Table perf.dm\_exec\_sessions](#DM_sessions)
13. [Table perf.dm\_resource\_governor\_workload\_groups](#Workload_groups)
14. [Table perf.dm\_ressorce\_governor\_resource\_pools](#Resource_pools)
15. [Table perf.killed\_sessions](#Killed_sessions)
16. [Table perf.open\_transactions](#Open_transactions)
17. [Table perf.sql\_infos](#Sql_infos)
18. [Table perf.wait\_status\_log](#Wait_status_log)
19. [Table perf.wait\_type\_blacklist](#Wait_type_blacklist)
20. [Table sched.job](#job)
21. [Table sched.job\_scheduler](#job_scheduler)
22. [Table sched.job\_step](#job_step)
23. **Table perf.perf\_exec\_Catalogue**

Returns information about date and timestamp. And generate ID for each execution.

1. **Table perf.perf\_exec\_used\_memory**

Returns information about all queries that have requested and are waiting for a memory grant or have been given a memory grant.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| ExpressMEM ##SPID## | ID (SPID) of the session where this query is running. |
| program\_name | Name of client program that initiated the session. The value is NULL for internal sessions. Is nullable. |
| granted\_memory\_kb | Total amount of memory actually granted in kilobytes. Can be NULL if the memory is not granted yet. |
| subStmt | Text of the SQL query. |
| login\_name | SQL Server login name under which the session is currently executing. For the original login name that created the session, see original\_login\_name. Can be a SQL Server authenticated login name or a Windows authenticated domain user name. Is not nullable. |
| requested\_memory\_gb | Total requested amount of memory in gigabytes. |
| granted\_memory\_gb | Total amount of memory actually granted in gigabytes. Can be NULL if the memory is not granted yet. |
| required\_memory\_gb | Minimum memory required to run this query in gigabytes. requested\_memory\_gb is the same or larger than this amount. |
| used\_memory\_gb | Physical memory used at this moment in gigabytes. |
| max\_used\_memory\_gb | Maximum physical memory used up to this moment in gigabytes. |
| query\_cost | Estimated query cost. |

|  |  |
| --- | --- |
| resource\_semaphore\_id | Non-unique ID of the resource semaphore on which this query is waiting.  Note: This ID is unique in versions of SQL Server that are earlier than SQL Server 2008. This change can affect troubleshooting query execution. For more information, see the "Remarks" section later in this article. |
| queue\_id | ID of waiting queue where this query waits for memory grants. NULL if the memory is already granted. |
| wait\_order | Sequential order of waiting queries within the specified queue\_id. This value can change for a given query if other queries get memory grants or time out. NULL if memory is already granted. |
| is\_next\_candidate | Candidate for next memory grant.  1 = Yes  0 = No  NULL = Memory is already granted. |
| wait\_time\_ms | Wait time in milliseconds. NULL if the memory is already granted. |
| dop | Applies to: SQL Server 2016 (13.x) and later.  The degree of parallelism of the query. |

1. **Table perf.perf\_exec\_serverinfos**

Returns information about server

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| Servername | Returns the name of the local server that is running SQL Server. |
| Version | Returns system and build information for the current installation of SQL Server. |
| login\_time | Time when session was established. |
| SPID | Returns text string depend of parameter 'All' - if SPID = 3 'relevante' - if SPID > 3 |
| InfoLevel | Returns text string depend of parameter 'Übersicht' if @InfoTiefe = 1 'Info' if @InfoTiefe = 2 'Deep Info' if @InfoTiefe = 9 |
| ProfilerInfo | Returns text string 'Profiler läuft unter User (SPID): ' and login\_name and session\_id |
| SpecialSet | Returns text string depend of parameter ' | nur Eigene Sessions' if @ShowNurEigeneSessions = 1 ' | mit Systemsessions' if @ShowSystemSessions = 1 ' | mit inkl. Sleeping/Dormant' if @ShowSleeping\_Dormant\_requests = 1 ' | inkl. IntelliSense/SQL Prompt' if @ShowIntelliSense = 1 |

1. **Table perf\_exec\_blocked\_sessions**

Returns information about blocked sessions on SQL Server

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| Blocker Session LV | ID of the session that is blocking the request. If this column is NULL or equal to 0, the request is not blocked, or the session information of the blocking session is not available (or cannot be identified). |
| SPID | Identifies the session associated with each active primary connection. |
| Status | If ID of the session to which this request is related is NULL and Number of transactions that are open for this request is not 0 |
| Returns string text |
| '!! Offene Session ohne Request mit offener Transaktion !!' |
| Otherwise returns Status of the session. Possible values: |
| Running - Currently running one or more requests |
| Sleeping - Currently running no requests |
| Dormant - Session has been reset because of connection pooling and is now in prelogin state. |
| Preconnect - Session is in the Resource Governor classifier. |
| login\_name | SQL Server login name under which the session is currently executing. For the original login name that created the session, see original\_login\_name. Can be a SQL Server authenticated login name or a Windows authenticated domain user name. Is not nullable. |
| BlockedBySPID | ID of the session that is blocking the request. If this column is NULL or equal to 0, the request is not blocked, or the session information of the blocking session is not available (or cannot be identified). |
| BlockedByUser | SQL Server login name under which the blocked session is currently executing. |

|  |  |
| --- | --- |
| command | Identifies the current type of command that is being processed. Common command types include the following values: |
| SELECT |
| INSERT |
| UPDATE |
| DELETE |
| BACKUP LOG |
| BACKUP DATABASE |
| DBCC |
| FOR |
| DB | Database name the request is executing against. |
| program\_name | Name of client program that initiated the session. |
| wait\_type | If the request is currently blocked, this column returns the type of wait. |
| wait\_seit | If the request is currently blocked, this column returns the duration in milliseconds, of the current wait. |
| StmtLfz | Time at which the last request on the session began. This includes the currently executing request. |
| wait\_resource | If the request is currently blocked, this column returns the resource for which the request is currently waiting. |
| Wait\_Resource\_Auflösung | If the request is currently blocked, this column returns the resource for which the request is currently waiting. With detailed description |

1. **Table perf\_exec\_sessions**

Returns one row per authenticated session on SQL Server. Is a server-scope view that shows information about all active user connections and internal tasks. This information includes client version, client program name, client login time, login user, current session setting, and more. Use table to first view the current system load and to identify a session of interest, and then learn more information about that session by using other dynamic management views or dynamic management functions.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| session ##SPID## | Identifies the session associated with each active primary connection. |
| Status | If ID of the session to which this request is related is NULL and Number of transactions that are open for this request is not 0 Returns string text '!! Offene Session ohne Request mit offener Transaktion !!' Otherwise returns Status of the session. Possible values: Running - Currently running one or more requests Sleeping - Currently running no requests Dormant - Session has been reset because of connection pooling and is now in prelogin state. Preconnect - Session is in the Resource Governor classifier. |
| BlockingSPID | ID of the session that is blocking the request. If this column is NULL or equal to 0, the request is not blocked, or the session information of the blocking session is not available (or cannot be identified). |
| S \_LifeTime | LastRequest | Period of time when session was established to current run time. And time period of last request on the session began and complete. |
| TempDB | Memory reserved or allocated for user objects by this session. |
| LoginName | Original | SQL Server login name under which the session is currently executing. For the original login name that created the session, see original\_login\_name. Can be a SQL Server authenticated login name or a Windows authenticated domain user name. |

|  |  |
| --- | --- |
| host\_name | Name of the client workstation that is specific to a session. The value is NULL for internal sessions. |
| Database | Database name where session was established |
| Command | Identifies the current type of command that is being processed. Common command types include the following values:  SELECT INSERT UPDATE DELETE BACKUP LOG BACKUP DATABASE DBCC FOR |
| ##SPID## | Identifies the session associated with each active primary connection. |
| cpu\_time | CPU time, in milliseconds, that was used by this session. |
| MB\_memory\_usage | Size in MB of memory used by this session. |
| total\_scheduled\_time | Total time, in milliseconds, for which the session (requests within) were scheduled for execution. |
| total\_elapsed\_time | Time, in milliseconds, since the session was established. |
| row\_count | Number of rows returned on the session up to this point. |
| reads | Number of reads performed, by requests in this session, during this session. |
| writes | Number of writes performed, by requests in this session, during this session. |
| logical\_reads | Number of logical reads that have been performed on the session. |

|  |  |
| --- | --- |
| transaction\_isolation\_level | Transaction isolation level of the session. 0 = Unspecified 1 = ReadUncommitted 2 = ReadCommitted 3 = RepeatableRead 4 = Serializable 5 = Snapshot |
| transaction\_Isolation\_Level\_value | Same as transaction isolation level of the session. Returns string with value according to  0 = Unspecified 1 = ReadUncommitted 2 = ReadCommitted 3 = RepeatableRead 4 = Serializable 5 = Snapshot |
| open\_transaction\_count | Applies to: SQL Server 2012 (11.x) and later. Number of open transactions per session. |
| ###SPID### | Identifies the session associated with each active primary connection. |
| request\_id | ID of the request. Unique in the context of the session. |

1. **Table perf\_exec\_requests**

Returns information about each request that is executing in SQL Server.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| Request ##SPID## | ID of the session to which this request is related. |
| Status | Status of the request. Can be one of the following values: - Background(The SPID is running a background task, such as deadlock detection.) - Running(The SPID is currently running on a scheduler.) - Runnable(Task wartet auf ein time quantum) - Sleeping(The SPID is not currently executing. This usually indicates that the SPID is awaitinq a command from the application.) - Suspended(Request wartet auf Ende von Event##wait\_type##) |
| BlockingSPlD | ID of the session that is blocking the request. If this column is NULL or equal to 0, the request is not blocked, or the session information of the blocking session is not available (or cannot be identified). |
| request\_id | ID of the request. Unique in the context of the session. |
| RequestLifeTime | Time period between Timestamp when the request arrived and time of current execution |
| Database | Name of the database the request is executing against. |
| dop | Applies to: SQL Server 2016 (13.x) and later. The degree of parallelism of the query. |
| command | Identifies the current type of command that is being processed. Common command types include the following values:  SELECT INSERT UPDATE DELETE BACKUP LOG BACKUP DATABASE DBCC FOR |

|  |  |
| --- | --- |
| MemGrant MB from Pages | Total amount of memory actually granted If number of pages allocated to the execution of a query on the request is 0 then returns string 'not needed' if executing\_managed\_code is 1 returns string 'managed code' |
| MemUsed | Physical memory used at this moment |
| MaxUsed | Maximum physical memory used up to this moment |
| cpu\_time | CPU time in milliseconds that is used by the request. |
| total\_elapsed\_time | Total time elapsed in milliseconds since the request arrived. |
| reads | Number of reads performed by this request. |
| writes | Number of writes performed by this request. |
| logical\_reads | Number of logical reads that have been performed by the request. |
| row\_count | Number of rows that have been returned to the client by this request. |
| ##SPID## | ID of the session to which this request is related. |
| prev\_error | Last error that occurred during the execution of the request. |
| nest\_level | Current nesting level of code that is executing on the request. |
| executing\_managed\_code | Indicates whether a specific request is currently executing common language runtime objects, such as routines, types, and triggers. It is set for the full time a common language runtime object is on the stack, even while running Transact-SQL from within common language runtime. |
| RessourceGroup | Name of the workload group. |
| wait\_type | If the request is currently blocked, this column returns the type of wait. |
| Wait\_type\_desc | If the request is currently blocked, this column returns the detailed description about type of wait |
| wait\_time | If the request is currently blocked, this column returns the duration in milliseconds, of the current wait. |
| last\_wait\_type | If this request has previously been blocked, this column returns the type of the last wait. |
| wait\_resource | If the request is currently blocked, this column returns the resource for which the request is currently waiting. |
| Wait\_Resource\_Auflosung |  |

|  |  |
| --- | --- |
| is\_resumable | Applies to: SQL Server 2017 (14.x) and later. Indicates whether the request is a resumable index operation. |
| page\_resource | Applies to: SQL Server 2019 (15.x) An 8-byte hexadecimal representation of the page resource if the wait\_resource column contains a page. |
| ###SPID### | ID of the session to which this request is related. |
| open\_transaction\_count | Number of transactions that are open for this request. |
| open\_resultset\_count | Number of result sets that are open for this request. |
| transaction\_id | ID of the transaction in which this request executes. |
| percent\_complete | Percentage of work completed for the following commands: ALTER INDEX REORGANIZE AUTO\_SHRINK option with ALTER DATABASE BACKUP DATABASE DBCC CHECKDB DBCC CHECKFILEGROUP DBCC CHECKTABLE DBCC INDEXDEFRAG DBCC SHRINKDATABASE DBCC SHRINKFILE RECOVERY RESTORE DATABASE ROLLBACK TDE ENCRYPTION |
| estimated\_completion | Timestump based on estimated\_completion\_time value in milliseconds added to current system date |

|  |  |
| --- | --- |
| TransactionIsolationLevel | Isolation level with which the transaction for this request is created. 0 = Unspecified 1 = ReadUncomitted 2 = ReadCommitted 3 = Repeatable 4 = Serializable 5 = Snapshot |
| sql\_handle | Is a token that uniquely identifies the batch or stored procedure that the query is part of. |
| plan\_handle | Is a token that uniquely identifies a query execution plan for a batch that is currently executing. |
| parallel\_worker\_count | Applies to: SQL Server 2016 (13.x) and later. The number of reserved parallel workers if this is a parallel query. |
| ####SPID#### | ID of the session to which this request is related. |
| SubStmt | SQL text of statement currently run |
| BatchStmt | SQL text of statement currently run in Batch |
| sql\_plan | Link to execution plan of current query |

1. **Table perf.blocking\_sessions**

Returns information about session that is blocking the request. And system information about programs, users and parameters of which blocked request depended.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| date | Current date when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| month\_id | Current month when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| guid | Uniqueidentifier |
| session\_id | ID of the session associated with the task. |
| login\_name | SQL Server login name under which the session is currently executing. |

|  |  |
| --- | --- |
| blocking\_session\_id | ID of the session that is blocking the request. If this column is NULL, the request is not blocked, or the session information of the blocking session is not available (or cannot be identified). |
| chain\_id | Not recognized field with NULL value |
| status | Status of the request. Can be one of the following values: Background Running Runnable Sleeping Suspended |
| host\_name | Name of the client workstation that is specific to a session. The value is NULL for internal sessions. |
| database\_name | Database name where session was established |
| command | Identifies the current type of command that is being processed. Common command types include the following values: SELECT INSERT UPDATE DELETE BACKUP LOG BACKUP DATABASE DBCC FOR |
| cpu\_time | CPU time in milliseconds that is used by the request. |
| disk\_to | Not recognized field with NULL value |
| last\_batch | Last time a client process executed a remote stored procedure call or an EXECUTE statement. |
| program\_name | Name of the application program. |
| spid2 | SQL Server session ID. |
| request\_id | ID of request. Used to identify requests running in a specific session. |
| blocking\_guid | Uniqueidentifier |

1. **Table perf.connected\_users**

Returns information about sessions created by current users

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| date | Current date when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| month\_id | Current month when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| guid | Uniqueidentifier |
| original\_login\_name | SQL Server login name that the client used to create this session. Can be a SQL Server authenticated login name, a Windows authenticated domain user name, or a contained database user. Note that the session could have gone through many implicit or explicit context switches after the initial connection. For example, if EXECUTE AS is used. |
| session\_count | Count of sessions created under current login. |

1. **Table perf.dm\_db\_task\_space\_usage**

Returns page allocation and deallocation activity by task for the database.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| date | Current date when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| month\_id | Current month when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| guid | Uniqueidentifier |
| session\_id | ID of the session to which this request is related. |
| request\_id | Request ID within the session. A request is also called a batch and may contain one or more queries. A session may have multiple requests active at the same time. Each query in the request may start multiple threads (tasks), if a parallel execution plan is used. |
| temp\_usage\_gb | Not recognized field with NULL value |

1. **Table perf.dm\_exec\_query\_memory\_grants**

Returns information about all queries that have requested and are waiting for a memory grant or have been given a memory grant. Queries that do not require a memory grant will not appear in this view. For example, sort and hash join operations have memory grants for query execution, while queries without an ORDER BY clause will not have a memory grant.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| date | Current date when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| month\_id | Current month when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| guid | Uniqueidentifier |
| session\_id | ID (SPID) of the session where this query is running. |
| request\_id | Request ID within the session. A request is also called a batch and may contain one or more queries. A session may have multiple requests active at the same time. Each query in the request may start multiple threads (tasks), if a parallel execution plan is used. |
| scheduler\_id | ID of the scheduler that is scheduling this query. |
| dop | Degree of parallelism of this query. |
| request\_time | Date and time when this query requested the memory grant. |
| grant\_time | Date and time when memory was granted for this query. NULL if memory is not granted yet. |
| requested\_memory\_kb | Total requested amount of memory in kilobytes. |
| granted\_memory\_kb | Total amount of memory actually granted in kilobytes. Can be NULL if the memory is not granted yet. For a typical situation, this value should be the same as requested\_memory\_kb. For index creation, the server may allow additional on-demand memory beyond initially granted memory. |
| required\_momory\_kb | Minimum memory required to run this query in kilobytes. requested\_memory\_kb is the same or larger than this amount. |
| used\_memory\_kb | Physical memory used at this moment in kilobytes. |
| max\_used\_memory\_kb | Maximum physical memory used up to this moment in kilobytes. |
| query\_cost | Estimated query cost. |

|  |  |
| --- | --- |
| timeout\_sec | Time-out in seconds before this query gives up the memory grant request. |
| resource\_semaphore\_id | Non-unique ID of the resource semaphore on which this query is waiting. |
| queue\_id | ID of waiting queue where this query waits for memory grants. NULL if the memory is already granted. |
| wait\_order | Sequential order of waiting queries within the specified queue\_id. This value can change for a given query if other queries get memory grants or time out. NULL if memory is already granted. |
| is\_nect\_candidate | Candidate for next memory grant. 1 = Yes 0 = No NULL = Memory is already granted. |
| wait\_time\_ms | Wait time in milliseconds. NULL if the memory is already granted. |
| plan\_handle | Identifier for this query plan. Use sys.dm\_exec\_query\_plan to extract the actual XML plan. |
| sql\_handle | Identifier for Transact-SQL text for this query. Use sys.dm\_exec\_sql\_text to get the actual Transact-SQL text. |
| group\_id | ID for the workload group where this query is running. |
| pool\_id | ID of the resource pool that this workload group belongs to. |
| is\_small | When set to 1, indicates that this grant uses the small resource semaphore. When set to 0, indicates that a regular semaphore is used. |
| ideal\_memory\_kb | Size, in kilobytes (KB), of the memory grant to fit everything into physical memory. This is based on the cardinality estimate. |
| reserved\_worker\_count | Number of reserved worker threads. Applies to: SQL Server (Starting with SQL Server 2016 (13.x)) and Azure SQL Database |
| used\_worker\_count | Number of worker threads used at this moment. Applies to: SQL Server (Starting with SQL Server 2016 (13.x)) and Azure SQL Database |
| reserved\_node\_bitmap | Bitmap of NUMA nodes where worker threads are reserved. |

1. **Table perf.dm\_exec\_requests**

Returns information about each request that is executing in SQL Server.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| date | Current date when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| month\_id | Current month when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| guid | Uniqueidentifier |
| session\_id | ID of the session to which this request is related. |
| request\_id | ID of the request. Unique in the context of the session. |
| start\_time | Timestamp when the request arrived. Is not nullable. |
| status | Status of the request. Can be one of the following values: Background Running Runnable Sleeping Suspended |
| command | Identifies the current type of command that is being processed. Common command types include the following values: SELECT INSERT UPDATE DELETE BACKUP LOG BACKUP DATABASE DBCC FOR  The text of the request can be retrieved by using sys.dm\_exec\_sql\_text with the corresponding sql\_handle for the request. Internal system processes set the command based on the type of task they perform. Tasks can include the following values: LOCK MONITOR CHECKPOINTLAZY WRITER |

.

|  |  |
| --- | --- |
| sql\_handle | Is a token that uniquely identifies the batch or stored procedure that the query is part of. |
| statement\_start\_offset | Indicates, in bytes, beginning with 0, the starting position of the currently executing statement for the currently executing batch or persisted object. Can be used together with the sql\_handle, the statement\_end\_offset, and the sys.dm\_exec\_sql\_text dynamic management function to retrieve the currently executing statement for the request. Is nullable. |
| statement\_end\_offset | Indicates, in bytes, starting with 0, the ending position of the currently executing statement for the currently executing batch or persisted object. Can be used together with the sql\_handle, the statement\_start\_offset, and the sys.dm\_exec\_sql\_text dynamic management function to retrieve the currently executing statement for the request. |
| plan\_handle | Is a token that uniquely identifies a query execution plan for a batch that is currently executing. |
| database\_id | ID of the database the request is executing against. |
| database\_name | Database name the request is executing against. |
| user\_id | ID of the user who submitted the request. |
| connection\_id | ID of the connection on which the request arrived. |
| blocking\_session\_id | ID of the session that is blocking the request. If this column is NULL or equal to 0, the request is not blocked, or the session information of the blocking session is not available (or cannot be identified). |
| wait\_type | If the request is currently blocked, this column returns the type of wait. |
| wait\_time | If the request is currently blocked, this column returns the duration in milliseconds, of the current wait. |
| last\_wait\_type | If this request has previously been blocked, this column returns the type of the last wait. |
| wait\_resource | If the request is currently blocked, this column returns the resource for which the request is currently waiting. |
| open\_transaction\_count | Number of transactions that are open for this request. |
| open\_result\_count | Number of result sets that are open for this request. |

|  |  |
| --- | --- |
| transaction\_id | ID of the transaction in which this request executes. |
| contex\_info | CONTEXT\_INFO value of the session. |
| percente\_complete | Percentage of work completed for the following commands: ALTER INDEX REORGANIZE AUTO\_SHRINK option with ALTER DATABASE BACKUP DATABASE DBCC CHECKDB DBCC CHECKFILEGROUP DBCC CHECKTABLE DBCC INDEXDEFRAG DBCC SHRINKDATABASE DBCC SHRINKFILE RECOVERY RESTORE DATABASE ROLLBACK TDE ENCRYPTION |
| estimated\_completion\_time | Estimated completion time in milliseconds. |
| cpu\_time | CPU time in milliseconds that is used by the request. |
| total\_elapsed\_time | Total time elapsed in milliseconds since the request arrived. |
| scheduler\_id | ID of the scheduler that is scheduling this request. |
| task\_address | Memory address allocated to the task that is associated with this request. |
| reads | Number of reads performed by this request. |
| writes | Number of writes performed by this request. |
| logical\_reades | Number of logical reads that have been performed by the request. |
| text\_size | TEXTSIZE setting for this request. |
| language | Language setting for the request. |
| date\_format | DATEFORMAT setting for the request. |
| date\_first | DATEFIRST setting for the request. |
| quoted\_identifier | 1 = QUOTED\_IDENTIFIER is ON for the request. Otherwise, it is 0. |
| arithabort | 1 = ARITHABORT setting is ON for the request. Otherwise, it is 0. |
| ansi\_null\_dflt\_on | 1 = ANSI\_NULL\_DFLT\_ON setting is ON for the request. Otherwise, it is 0. |

|  |  |
| --- | --- |
| ansi\_defaults | 1 = ANSI\_DEFAULTS setting is ON for the request. Otherwise, it is 0. |
| ansi\_warnings | 1 = ANSI\_WARNINGS setting is ON for the request. Otherwise, it is 0. |
| ansi\_padding | 1 = ANSI\_PADDING setting is ON for the request. Otherwise, it is 0. |
| ansi\_nulls | 1 = ANSI\_NULLS setting is ON for the request. Otherwise, it is 0. |
| contact\_null\_yields\_null | 1 = CONCAT\_NULL\_YIELDS\_NULL setting is ON for the request. Otherwise, it is 0. |
| transaction\_isolation\_level | Isolation level with which the transaction for this request is created. Is not nullable. 0 = Unspecified 1 = ReadUncomitted 2 = ReadCommitted 3 = Repeatable 4 = Serializable 5 = Snapshot |
| lock\_timeout | Lock time-out period in milliseconds for this request. |
| deadlock\_priority | DEADLOCK\_PRIORITY setting for the request. |
| row\_count | Number of rows that have been returned to the client by this request. |
| prev\_error | Last error that occurred during the execution of the request. |
| nest\_level | Current nesting level of code that is executing on the request. |
| granted\_query\_memory | Number of pages allocated to the execution of a query on the request. |
| executing\_managed\_code | Indicates whether a specific request is currently executing common language runtime objects, such as routines, types, and triggers. It is set for the full time a common language runtime object is on the stack, even while running Transact-SQL from within common language runtime. Is not nullable. |

|  |  |
| --- | --- |
| group\_id | ID of the workload group to which this query belongs. |
| query\_hash | Binary hash value calculated on the query and used to identify queries with similar logic. You can use the query hash to determine the aggregate resource usage for queries that differ only by literal values. |
| query\_plan\_hash | Binary hash value calculated on the query execution plan and used to identify similar query execution plans. You can use query plan hash to find the cumulative cost of queries with similar execution plans. |
| statement\_sql\_handle | Applies to: SQL Server 2014 (12.x) and later. SQL handle of the individual query. This column is NULL if Query Store is not enabled for the database. |
| statement\_context\_id | Applies to: SQL Server 2014 (12.x) and later. The optional foreign key to sys.query\_context\_settings. This column is NULL if Query Store is not enabled for the database. |
| dop | Applies to: SQL Server 2016 (13.x) and later. The degree of parallelism of the query. |
| parallel\_worker\_count | Applies to: SQL Server 2016 (13.x) and later. The number of reserved parallel workers if this is a parallel query. |
| external\_script\_request\_id | Applies to: SQL Server 2016 (13.x) and later. The external script request ID associated with the current request. |

1. **Table perf.dm\_exec\_sessions**

Returns one row per authenticated session on SQL Server. sys.dm\_exec\_sessions is a server-scope view that shows information about all active user connections and internal tasks. This information includes client version, client program name, client login time, login user, current session setting, and more. Use sys.dm\_exec\_sessions to first view the current system load and to identify a session of interest, and then learn more information about that session by using other dynamic management views or dynamic management functions.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| date | Current date when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| month\_id | Current month when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| guid | Uniqueidentifier |
| session\_id | Identifies the session associated with each active primary connection. |
| log\_time | Time when session was established. Is not nullable. Sessions that have not completed logging in at the time this DMV is queried are shown with a a login time of 1900-01-01. |
| host\_name | Name of the client workstation that is specific to a session. The value is NULL for internal sessions. |
| program\_name | Name of client program that initiated the session. The value is NULL for internal sessions. |
| host\_process\_id | Process ID of the client program that initiated the session. The value is NULL for internal sessions. |
| client\_version | TDS protocol version of the interface that is used by the client to connect to the server. The value is NULL for internal sessions. |
| client\_interface\_name | Name of library/driver being used by the client to communicate with the server. The value is NULL for internal sessions. |
| security\_id | Microsoft Windows security ID associated with the login. |
| login\_name | SQL Server login name under which the session is currently executing. For the original login name that created the session, see original\_login\_name. Can be a SQL Server authenticated login name or a Windows authenticated domain user name. |

|  |  |
| --- | --- |
| nt\_domain | Applies to: SQL Server 2008 and later. Windows domain for the client if the session is using Windows Authentication or a trusted connection. This value is NULL for internal sessions and non-domain users. |
| nt\_user\_name | Applies to: SQL Server 2008 and later. Windows user name for the client if the session is using Windows Authentication or a trusted connection. This value is NULL for internal sessions and non-domain users. |
| status | Status of the session. Possible values: Running - Currently running one or more requests Sleeping - Currently running no requests Dormant - Session has been reset because of connection pooling and is now in prelogin state. Preconnect - Session is in the Resource Governor classifier. |
| context\_info | CONTEXT\_INFO value for the session. The context information is set by the user by using the SET CONTEXT\_INFO statement. |
| cpu\_time | CPU time, in milliseconds, that was used by this session. |
| memory\_usage | Number of 8-KB pages of memory used by this session. |
| total\_schedulet\_time | Total time, in milliseconds, for which the session (requests within) were scheduled for execution. |
| total\_elapset\_time | Time, in milliseconds, since the session was established. |
| endpoint\_id | ID of the Endpoint associated with the session. |
| lest\_request\_start\_time | Time at which the last request on the session began. This includes the currently executing request. |
| lest\_request\_end\_time | Time of the last completion of a request on the session. |
| reades | Number of reads performed, by requests in this session, during this session. |
| writes | Number of writes performed, by requests in this session, during this session. |
| logical\_reads | Number of logical reads that have been performed on the session. |
| is\_user\_process | 0 if the session is a system session. Otherwise, it is 1. |

|  |  |
| --- | --- |
| text\_size | TEXTSIZE setting for the session. |
| language | LANGUAGE setting for the session. |
| date\_format | DATEFORMAT setting for the session. |
| date\_first | DATEFIRST setting for the session. |
| quoted\_identifier | QUOTED\_IDENTIFIER setting for the session. |
| arithabort | ARITHABORT setting for the session. |
| ansi\_null\_dflt\_on | ANSI\_NULL\_DFLT\_ON setting for the session. |
| ansi\_defaults | ANSI\_DEFAULTS setting for the session. |
| ansi\_warnings | ANSI\_WARNINGS setting for the session. |
| ansi\_padding | ANSI\_PADDING setting for the session. |
| ansi\_nulls | ANSI\_NULLS setting for the session. |
| contact\_null\_yields\_null | CONCAT\_NULL\_YIELDS\_NULL setting for the session. |
| transaction\_isolation\_level | Transaction isolation level of the session. 0 = Unspecified 1 = ReadUncommitted 2 = ReadCommitted 3 = RepeatableRead 4 = Serializable 5 = Snapshot |
| lock\_timeout | LOCK\_TIMEOUT setting for the session. The value is in milliseconds. |
| deadlock\_priority | DEADLOCK\_PRIORITY setting for the session. |
| row\_count | Number of rows returned on the session up to this point. |
| prev\_error | ID of the last error returned on the session. |
| original\_security\_id | Microsoft Windows security ID that is associated with the original\_login\_name. |
| origonal\_login\_name | SQL Server login name that the client used to create this session. Can be a SQL Server authenticated login name, a Windows authenticated domain user name, or a contained database user. Note that the session could have gone through many implicit or explicit context switches after the initial connection. For example, if EXECUTE AS is used. |

|  |  |
| --- | --- |
| last\_successful\_logon | Applies to: SQL Server 2008 and later. Time of the last successful logon for the original\_login\_name before the current session started. |
| last\_unsuccessful\_logon | Applies to: SQL Server 2008 and later. Time of the last unsuccessful logon attempt for the original\_login\_name before the current session started. |
| unsuccessful\_logons | Applies to: SQL Server 2008 and later. Number of unsuccessful logon attempts for the original\_login\_name between the last\_successful\_logon and login\_time. |
| group\_id | ID of the workload group to which this session belongs. |
| database\_id | Applies to: SQL Server 2012 (11.x) and later. ID of the current database for each session. |
| database\_name | Name of the current database for each session. |
| authenticating\_database\_id | Applies to: SQL Server 2012 (11.x) and later. ID of the database authenticating the principal. For Logins, the value will be 0. For contained database users, the value will be the database ID of the contained database. |
| open\_transation\_count | Applies to: SQL Server 2012 (11.x) and later. Number of open transactions per session. |

1. **Table perf.dm\_resource\_governor\_workload\_groups**

Returns workload group statistics and the current in-memory configuration of the workload group.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| date | Current date when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| month\_id | Current month when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| guid | Uniqueidentifier |
| group\_id | ID of the workload group. |
| name | Name of the workload group. |
| pool\_id | ID of the resource pool. |
| external\_pool\_id | Applies to: Starting with SQL Server 2016 (13.x). ID of the external resource pool. Is not nullable. |
| statistic\_start\_time | Time that statistics collection was reset for the workload group. |
| total\_request\_count | Cumulative count of completed requests in the workload group. |
| total\_queued\_request\_count | Cumulative count of requests queued after the GROUP\_MAX\_REQUESTS limit was reached. |
| active\_request\_count | Current request count. |
| queued\_request\_count | Current queued request count. |
| total\_cpu\_limit\_violation\_count | Cumulative count of requests exceeding the CPU limit. |
| total\_cpu\_usage\_ms | Cumulative CPU usage, in milliseconds, by this workload group. |
| max\_request\_cpu\_time\_ms | Maximum CPU usage, in milliseconds, for a single request. |
| blocked\_task\_count | Current count of blocked tasks. |
| total\_lock\_wait\_count | Cumulative count of lock waits that occurred. |
| total\_lock\_wait\_time\_ms | Cumulative sum of elapsed time, in milliseconds, a lock is held. |
| total\_query\_optimization\_count | Cumulative count of query optimizations in this workload group. |

|  |  |
| --- | --- |
| total\_suboptimal\_plan\_generation\_count | Cumulative count of suboptimal plan generations that occurred in this workload group due to memory pressure. |
| total\_reduced\_memgrant\_count | Cumulative count of memory grants that reached the maximum query size limit. |
| max\_request\_grant\_memory\_kb | Maximum memory grant size, in kilobytes, of a single request since the statistics were reset. |
| active\_parallel\_thread\_count | Current count of parallel thread usage. |
| importance | Current configuration value for the relative importance of a request in this workload group. Importance is one of the following, with Medium being the default: Low, Medium, or High. |
| request\_max\_memory\_grant\_percent | Current setting for the maximum memory grant, as a percentage, for a single request. |
| request\_max\_cpu\_time\_sec | Current setting for maximum CPU use limit, in seconds, for a single request. |
| request\_memory\_grant\_timeout\_sec | Current setting for memory grant time-out, in seconds, for a single request. |
| group\_max\_requests | Current setting for the maximum number of concurrent requests. |
| max\_drop | Configured maximum degree of parallelism for the workload group. The default value, 0, uses global settings. |
| effective\_max\_drop | Applies to: Starting with SQL Server 2012 (11.x). Effective maximum degree of parallelism for the workload group. |
| total\_cpu\_usage\_preemptive\_ms | Applies to: Starting with SQL Server 2016 (13.x). Total CPU time used while in preemptive mode scheduling for the workload group, measured in ms. Is not nullable. To execute code that is outside SQL Server (for example, extended stored procedures and distributed queries), a thread has to execute outside the control of the non-preemptive scheduler. To do this, a worker switches to preemptive mode. |

1. **Table perf.dm\_ressorce\_governor\_resource\_pools**

Returns information about the current resource pool state, the current configuration of resource pools, and resource pool statistics.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| date | Current date when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| month\_id | Current month when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| guid | Uniqueidentifier |
| pool\_id | The ID of the resource pool. |
| name | The name of the resource pool. |
| statistic\_start\_time | The time when statistics was reset for this pool. |
| total\_cpu\_usage\_ms | The cumulative CPU usage in milliseconds since the Resource Governor statistics were reset. |
| cache\_memory\_kb | The current total cache memory usage in kilobytes. |
| complete\_memory\_kb | The current total stolen memory usage in kilobytes (KB). The majority of this usage would be for compile and optimization, but it can also include other memory users. |
| used\_memgrant\_kb | The current total used (stolen) memory from memory grants. |
| total\_memgrant\_count | The cumulative count of memory grants in this resource pool. |
| total\_memgrant\_timeout\_count | The cumulative count of memory grant time-outs in this resource pool. |
| active\_memgrant\_count | The current count of memory grants. |
| active\_memgrant\_kb | The sum, in kilobytes (KB), of current memory grants. |
| memgrant\_waiter\_count | The count of queries currently pending on memory grants. |
| max\_memory\_kb | The maximum amount of memory, in kilobytes, that the resource pool can have. This is based on the current settings and server state. |
| used\_memory\_kb | The amount of memory used, in kilobytes, for the resource pool. |

|  |  |
| --- | --- |
| target\_memory\_kb | The target amount of memory, in kilobytes, the resource pool is trying to attain. This is based on the current settings and server state. |
| out\_of\_memory\_count | The number of failed memory allocations in the pool since the Resource Governor statistics were reset. |
| min\_cpu\_percent | The current configuration for the guaranteed average CPU bandwidth for all requests in the resource pool when there is CPU contention. |
| max\_cpu\_percent | The current configuration for the maximum average CPU bandwidth allowed for all requests in the resource pool when there is CPU contention. |
| cap\_cpu\_percent | Hard cap on the CPU bandwidth that all requests in the resource pool will receive. Limits the maximum CPU bandwidth level to the specified level. The allowed range for value is from 1 through 100. Is not nullable. Applies to: SQL Server (Starting with SQL Server 2012 (11.x)) |
| min\_iops\_per\_volume | The minimum I/O per second (IOPS) per disk volume setting for this Pool. Is nullable. Null if the resource pool is not governed for I/O. That is, the Resource Pool MIN\_IOPS\_PER\_VOLUME and MAX\_IOPS\_PER\_VOLUME settings are 0. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |
| max\_iops\_per\_volume | The maximum I/O per second (IOPS) per disk volume setting for this Pool. Is nullable. Null if the resource pool is not governed for I/O. That is, the Resource Pool MIN\_IOPS\_PER\_VOLUME and MAX\_IOPS\_PER\_VOLUME settings are 0. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |
| read\_io\_queued\_total | The total read I/Os enqueued since the Resource Governor was reset. Is nullable. Null if the resource pool is not governed for I/O. That is, the Resource Pool MIN\_IOPS\_PER\_VOLUME and MAX\_IOPS\_PER\_VOLUME settings are 0. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |

|  |  |
| --- | --- |
| read\_io\_issued\_total | The total read I/Os issued since the Resource Governor statistics were reset. Is nullable. Null if the resource pool is not governed for I/O. That is, the Resource Pool MIN\_IOPS\_PER\_VOLUME and MAX\_IOPS\_PER\_VOLUME settings are 0. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |
| read\_io\_completed\_total | The total read I/Os completed since the Resource Governor statistics were reset. |
| read\_io\_throtted\_total | The total read I/Os throttled since the Resource Governor statistics were reset. Is nullable. Null if the resource pool is not governed for I/O. That is, the Resource Pool MIN\_IOPS\_PER\_VOLUME and MAX\_IOPS\_PER\_VOLUME settings are 0. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |
| read\_bytes\_total | The total number of bytes read since the Resource Governor statistics were reset. Is not nullable. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |
| read\_io\_stall\_total\_ms | Total time (in milliseconds) between read I/O arrival and completion. Is not nullable. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |
| read\_io\_stall\_queued\_ms | Total time (in milliseconds) between read I/O arrival and issue. Is nullable. Null if the resource pool is not governed for I/O. That is, the Resource Pool MIN\_IOPS\_PER\_VOLUME and MAX\_IOPS\_PER\_VOLUME settings are 0. To determine if the I/O setting for the pool is causing latency, subtract read\_io\_stall\_queued\_ms from read\_io\_stall\_total\_ms. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |
| write\_io\_queued\_total | The total write I/Os enqueued since the Resource Governor statistics were reset. Is nullable. Null if the resource pool is not governed for I/O. That is, the Resource Pool MIN\_IOPS\_PER\_VOLUME and MAX\_IOPS\_PER\_VOLUME settings are 0. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |

|  |  |
| --- | --- |
| write\_io\_issued\_total | The total write I/Os issued since the Resource Governor statistics were reset. Is nullable. Null if the resource pool is not governed for I/O. That is, the Resource Pool MIN\_IOPS\_PER\_VOLUME and MAX\_IOPS\_PER\_VOLUME settings are 0. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |
| write\_io\_completed\_total | The total write I/Os completed since the Resource Governor statistics were reset. Is not nullable. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |
| write\_io\_throtted\_total | The total write I/Os throttled since the Resource Governor statistics were reset. Is not nullable. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |
| write\_bytes\_total | The total number of bytes written since the Resource Governor statistics were reset. Is not nullable. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |
| write\_io\_stall\_total\_ms | Total time (in milliseconds) between write I/O arrival and completion. Is not nullable. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |
| write\_io\_stall\_queued\_ms | Total time (in milliseconds) between write I/O arrival and issue. Is nullable. Null if the resource pool is not governed for I/O. That is, the Resource Pool MIN\_IOPS\_PER\_VOLUME and MAX\_IOPS\_PER\_VOLUME settings are 0. This is the delay introduced by I/O Resource Governance. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |

|  |  |
| --- | --- |
| io\_issue\_violations\_total | Total I/O issue violations. That is, the number of times when the rate of I/O issue was lower than the reserved rate. Is nullable. Null if the resource pool is not governed for I/O. That is, the Resource Pool MIN\_IOPS\_PER\_VOLUME and MAX\_IOPS\_PER\_VOLUME settings are 0. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |
| io\_issue\_delay\_total\_ms | Total time (in milliseconds) between the scheduled issue and actual issue of I/O. Is nullable. Null if the resource pool is not governed for I/O. That is, the Resource Pool MIN\_IOPS\_PER\_VOLUME and MAX\_IOPS\_PER\_VOLUME settings are 0. Applies to: SQL Server (Starting with SQL Server 2014 (12.x)) |
| io\_issue\_ahead\_total\_ms | Internal use only. Applies to: SQL Server (Starting with SQL Server 2016 (13.x)) |
| reseved\_io\_limited\_by\_volumes\_total | Internal use only. Applies to: SQL Server (Starting with SQL Server 2016 (13.x)) |
| io\_issue\_delay\_non\_throtted\_total\_ms | Total time (in milliseconds) between the scheduled issue and actual issue of a non-throttled I/O. Applies to: SQL Server (Starting with SQL Server 2016 (13.x)) |
| total\_cpu\_delayed\_ms | Total time (in milliseconds) between a runnable worker yields, and the operating system gives back control to another runnable worker in the Database Engine. This could be the Idle worker. Applies to: SQL Server (Starting with SQL Server 2016 (13.x)) |
| total\_cpu\_active\_ms | Total active CPU time (in milliseconds). Applies to: SQL Server (Starting with SQL Server 2016 (13.x)) |

|  |  |
| --- | --- |
| total\_cpu\_violation\_delay\_ms | Total CPU violation delays (in milliseconds). That is, total CPU time delay that was lower than the minimum guaranteed delay between a runnable worker yields, and the operating system gives back control to another runnable worker in the Database Engine. Applies to: SQL Server (Starting with SQL Server 2016 (13.x)) |
| total\_cpu\_violation\_sec | Total CPU violations (in seconds). That is, total time accrued when a CPU time violation was in-flight. Applies to: SQL Server (Starting with SQL Server 2016 (13.x)) |
| total\_cpu\_usage\_preemptive\_ms | Total CPU time used while in preemptive mode scheduling for the workload group (in milliseconds). Is not nullable. To execute code that is outside the Database Engine (for example, extended stored procedures and distributed queries), a thread has to execute outside the control of the non-preemptive scheduler. To do this, a worker switches to preemptive mode. Applies to: SQL Server (Starting with SQL Server 2016 (13.x)) |

1. **Table perf.killed\_sessions**

Returns information about killed sessions

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| date | Current date when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| month\_id | Current month when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| guid | Uniqueidentifier |
| session\_id | Identifies the session associated with each active primary connection. |

1. **Table perf.open\_transactions**

Returns information about current open transactions on SQL server.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| date | Current date when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| month\_id | Current month when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| guid | Uniqueidentifier |
| status | Process ID status. The possible values are: dormant = SQL Server is resetting the session. running = The session is running one or more batches. When Multiple Active Result Sets (MARS) is enabled, a session can run multiple batches. background = The session is running a background task, such as deadlock detection. rollback = The session has a transaction rollback in process. pending = The session is waiting for a worker thread to become available. runnable = The task in the session is in the runnable queue of a scheduler while waiting to get a time quantum. spinloop = The task in the session is waiting for a spinlock to become free. suspended = The session is waiting for an event, such as I/O, to complete. |
| session\_id | SQL Server session ID. |
| database\_name | Name of the current database for session. |
| open\_transaction\_count | Number of transactions that are open for the request. |
| login\_name | Login name. |
| login\_time | Time at which a client process logged into the server. |

|  |  |
| --- | --- |
| last\_request\_start\_time | Time at which the last request on the session began. This includes the currently executing request. |
| last\_request\_end\_time | Time of the last completion of a request on the session. |
| writes | Number of writes performed, by requests in this session, during this session. |
| program\_name | Name of the application program from which run session. |
| host\_name | Name of the workstation. |
| row\_count | Number of rows that have been returned to the client by this request. |
| transaction\_isolation\_level | Isolation level with which the transaction for this request is created. 0 = Unspecified 1 = ReadUncomitted 2 = ReadCommitted 3 = Repeatable 4 = Serializable 5 = Snapshot |
| transaction\_begin\_time | Time that the transaction started. |
| database\_transaction\_begin\_time | Time at which the database became involved in the transaction. Specifically, it is the time of the first log record in the database for the transaction. |

1. **Table perf.sql\_infos**

Returns information about executing SQL query plan.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| date | Current date when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| month\_id | Current month when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| sql\_handle | Is a token that uniquely identifies a batch that has executed or is currently executing. sql\_handle is varbinary(64). |
| plan\_handle | Is a token that uniquely identifies a query execution plan for a batch that has executed and its plan resides in the plan cache, or is currently executing. plan\_handle is varbinary(64). |

|  |  |
| --- | --- |
| t-sql | Text of the SQL query. Is NULL for encrypted objects. |
| query\_plan | Returns the Showplan in XML format for the batch specified by the plan handle. The plan specified by the plan handle can either be cached or currently executing. The XML schema for the Showplan is published and available at this Microsoft Web site. It is also available in the directory where SQL Server is installed. |

1. **Table perf.wait\_status\_log**

Returns information about current wait status

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| date | Current date when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| month\_id | Current month when execution start. This value is derived from the operating system of the computer on which the instance of SQL Server is running. |
| wait\_type | Name of the wait type. |
| waiting\_tasks\_count | Number of waits on this wait type. This counter is incremented at the start of each wait. |
| wait\_time\_ms | Total wait time for this wait type in milliseconds. This time is inclusive of signal\_wait\_time\_ms. |
| max\_wait\_time\_ms | Maximum wait time on this wait type. |
| signal\_wait\_time\_ms | Difference between the time that the waiting thread was signaled and when it started running. |

1. **Table perf.wait\_type\_blacklist**

Returns information about current wait type

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| wait\_type | Name of the wait type. |
| info | Not recognized field with NULL value |

1. **Table sched.job**

Return list of SQL Server Agent jobs and information about them.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| ident\_code | Numbers the output of a result set. For each job return number starting at 1 for the first job in resultset. |
| name | Name of the job. |
| description | Description for the job. |
| enabled | Indicates whether the job is enabled to be executed. |

1. **Table sched.job\_scheduler**

Contains information about SQL Server Agent job schedules.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| ident\_code | Numbers the output of a result set. For each job return number starting at 1 for the first job in resultset. |
| name | Name of the job. |
| enabled | Indicates whether the job is enabled to be executed. |

|  |  |
| --- | --- |
| freq\_type | How frequently a job runs for this schedule. 1 = One time only 4 = Daily 8 = Weekly 16 = Monthly 32 = Monthly, relative to freq\_interval 64 = Runs when the SQL Server Agent service starts 128 = Runs when the computer is idle |
| freq\_interval | Days that the job is executed. Depends on the value of freq\_type. The default value is 0, which indicates that freq\_interval is unused. See the table below for the possible values and their effects. |
| freq\_subday\_type | Units for the freq\_subday\_interval. The following are the possible values and their descriptions. 1 : At the specified time 2 : Seconds 4 : Minutes 8 : Hours |
| freq\_subday\_interval | Number of freq\_subday\_type periods to occur between each execution of the job. |
| freq\_relative\_interval | When freq\_interval occurs in each month, if freq\_type is 32 (monthly relative). Can be one of the following values: 0 = freq\_relative\_interval is unused 1 = First 2 = Second 4 = Third 8 = Fourth 16 = Last |
| freq\_recurrence\_factor | Number of weeks or months between the scheduled execution of a job. freq\_recurrence\_factor is used only if freq\_type is 8, 16, or 32. If this column contains 0, freq\_recurrence\_factor is unused. |
| active\_start\_date | Date on which execution of a job can begin. The date is formatted as YYYYMMDD. NULL indicates today's date. |
| active\_end\_date | Date on which execution of a job can stop. The date is formatted YYYYMMDD. |
| active\_start\_time | Time on any day between active\_start\_date and active\_end\_date that job begins executing. Time is formatted HHMMSS, using a 24-hour clock. |
| active\_end\_time | Time on any day between active\_start\_date and active\_end\_date that job stops executing. Time is formatted HHMMSS, using a 24-hour clock. |

1. **Table sched.job\_step**

Return list of SQL Server Agent job steps and information about them.

|  |  |
| --- | --- |
| **Field name** | **Field Description** |
| CatalogueID | ID from table perf\_exec\_Catalogue for current execution |
| ident\_code | Numbers the output of a result set. For each job return number starting at 1 for the first job in resultset. |
| name | Name of the job. |
| job\_id | ID of the job. |
| step\_id | ID of the step in the job. |
| command | Command to be executed by subsystem. |
| on\_success\_action | Action to be performed when a step is executed successfully. 1 = (default) Quit with success 2 = Quit with failure 3 = Go to next step 4 = Go to step on\_success\_step\_id |
| on\_success\_action\_id | ID of the next step to execute when a step is executed successfully. |
| on\_fail\_action | Action to be performed when a step is not executed successfully. 1 = Quit with success 2 = (default) Quit with failure 3 = Go to next step 4 = Go to step on\_fail\_step\_id |
| on\_fail\_action\_id | ID of the next step to execute when a step is not executed successfully. |
| database\_name | Name of the database in which command is executed if subsystem is TSQL. |
| retry\_attempts | Number of retry attempts made if the step fails. |
| retry\_interval | Amount of time to wait between retry attempts. |
| flags | Possible values: 0 = Overwrite output file 2 = Append to output file 4 = Write Transact-SQL job step output to step history 8 = Write log to table (overwrite existing history) 16 = Write log to table (append to existing history) 32 = Include step output in history 64 = Create a Windows event to use as a signal for the Cmd jobstep to abort |