Queries with complete percentage complete

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:32 AM PST

Contents

Queries that have percent complete information

Some commands will have information of the percentage completed, allowing you to have a clear picture at what stage the command is.

The list of operations that have a percent complete information are:

- 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

The list of operations is listed on the public documentation for sys.dm_exec_requests DMV.

How to check using T-SQL

By SPID number. Percentage is returned on the **percent_complete [%]** column:

```
SET NOCOUNT ON ;

SELECT      @@SERVERNAME Server,
                    [SPID] = SESSION_ID ,
            percent_complete [%] ,
            [DATABASE] = DB_NAME(SP.DBID) ,
            [STATUS] = ER.STATUS ,
            [WAIT] = WAIT_TYPE ,
            wait_resource ,
            reads ,
            writes ,
            logical_reads ,
            command ,
            [INDIVIDUAL QUERY] = SUBSTRING(QT.TEXT,
                                           ER.STATEMENT_START_OFFSET
                                           / 2,
                                           ( CASE WHEN ER.STATEMENT_END_OFFSET = -1
                                                  THEN LEN(CONVERT(NVARCHAR(MAX), QT.TEXT))
                                                  * 2
                                                  ELSE ER.STATEMENT_END_OFFSET
                                             END
                                             - ER.STATEMENT_START_OFFSET )
                                           / 2) ,
            [QUERY] = QT.TEXT ,
            PROGRAM = PROGRAM_NAME ,
            [USER] = NT_USERNAME ,
            HOSTNAME ,
            NT_DOMAIN ,
            START_TIME ,
            QP.query_plan AS xml_batch_query_plan
FROM        sys.dm_exec_requests ER WITH (NOLOCK)
            INNER JOIN sys.sysprocesses SP WITH (NOLOCK) ON ER.SESSION_ID = SP.SPID
            CROSS APPLY sys.dm_exec_sql_text(ER.SQL_HANDLE) AS QT
            CROSS APPLY sys.dm_exec_query_plan(ER.plan_handle) QP
WHERE       SESSION_ID = <spid_number>
ORDER BY    1 ,
            2
```

By command. The example below is for **RESTORE**:

```
SET NOCOUNT ON ;

SELECT      @@SERVERNAME Server,
                    [SPID] = SESSION_ID ,
            percent_complete [%] ,
            [DATABASE] = DB_NAME(SP.DBID) ,
            [STATUS] = ER.STATUS ,
            [WAIT] = WAIT_TYPE ,
            wait_resource ,
            reads ,
            writes ,
            logical_reads ,
            command ,
            [INDIVIDUAL QUERY] = SUBSTRING(QT.TEXT,
                                           ER.STATEMENT_START_OFFSET
                                           / 2,
                                           ( CASE WHEN ER.STATEMENT_END_OFFSET = -1
                                                  THEN LEN(CONVERT(NVARCHAR(MAX), QT.TEXT))
                                                  * 2
                                                  ELSE ER.STATEMENT_END_OFFSET
                                             END
                                             - ER.STATEMENT_START_OFFSET )
                                           / 2) ,
            [QUERY] = QT.TEXT ,
            PROGRAM = PROGRAM_NAME ,
            [USER] = NT_USERNAME ,
            HOSTNAME ,
            NT_DOMAIN ,
            START_TIME ,
            QP.query_plan AS xml_batch_query_plan
FROM        sys.dm_exec_requests ER WITH (NOLOCK)
            INNER JOIN sys.sysprocesses SP WITH (NOLOCK) ON ER.SESSION_ID = SP.SPID
            CROSS APPLY sys.dm_exec_sql_text(ER.SQL_HANDLE) AS QT
            CROSS APPLY sys.dm_exec_query_plan(ER.plan_handle) QP
WHERE       SESSION_ID > 50
            AND SESSION_ID NOT IN ( @@SPID )
                                   and command like 'RESTORE%'
ORDER BY    1 ,
            2
```

Calculate to estimated time for completion

Note: this is only an estimate, converting to minutes the information returned from the **estimated_completion_time** column.

Use the appropriate spid number.

```
select
percent_complete as pctcomplete,
start_time as starttime,
command as command,
b.name as databasename,
dateadd(ms,estimated_completion_time,getdate()) as estimatedendtime,
      (estimated_completion_time/1000/60) as estimatedminutestoend
from sys.dm_exec_requests a
 inner join sys.databases b on a.database_id = b.database_id
where session_id = 57 -- SPID number
```

How good have you found this content?