

How to find out the size of data in MS SQL databases

Posted At : 3 October 2010 22:15 | Posted By : Shaun McCran
Related Categories: Development, SQL

If you are running Microsoft SQL server you may already be familiar with how the database file system can enlarge itself to astronomical proportions. There are 'Auto shrink' and 'Auto grow' options, but these are really not recommended for production environments.

So how do you go about determining how much space is being used by actual data IE records, and how much is being used by log files etc? This article will show you how to run a query that will return the size of all the tables in your database, and highlight a TSQL command that will clean up the file storage of a database.

If, like me, you have a database that appears to be much larger than the data it contains would indicate, then you probably have a large volume of data files on the disk.

There is a Transact SQL command that 'Shrinks the size of the data files in the specified database'. This will shrink the log files and the data files of the specified database. As an example I recently shrunk a 890mb database down to 15mb.

```
DBCC SHRINKDATABASE
( database_name [ ,target_percent ]
[ , { NOTRUNCATE | TRUNCATEONLY } ]
)
```

A full description of the command can be found on the Microsoft MSDN site here: <http://msdn.microsoft.com/en-us/library/Aa258287>

If the command above does not resolve your issue you can interrogate the database directly as to how many records there are in each table, and how much space each table is occupying.

```
CREATE TABLE #temp (  
    table_name sysname ,  
    row_count int,  
    reserved_size varchar(50),  
    data_size varchar(50),  
    index_size varchar(50),  
    unused_size varchar(50))  
  
SET NOCOUNT ON  
INSERT #temp exec sp_msforeachtable 'sp_spaceused ''?'''  
  
SELECT  a.table_name, a.row_count, count(*) AS col_count, a.data_size  
        FROM #temp a  
        INNER JOIN information_schema.columns b on a.table_name = b.table_name  
        GROUP BY a.table_name, a.row_count, a.data_size  
        ORDER BY CAST(Replace(a.data_size, ' KB', '') AS integer) DESC  
        DROP TABLE #temp
```

This code will create a temporary table and populate it with a set of records telling you how many records are in each table, and the size of each table.