

How to create a SQL Server Clone / Statistics Only Database

Written By: Matteo Lorini -- 5/5/2010

Problem

Is there a way to reproduce and diagnose query compilation and plan issues without actually having the data or the entire database?

Solution

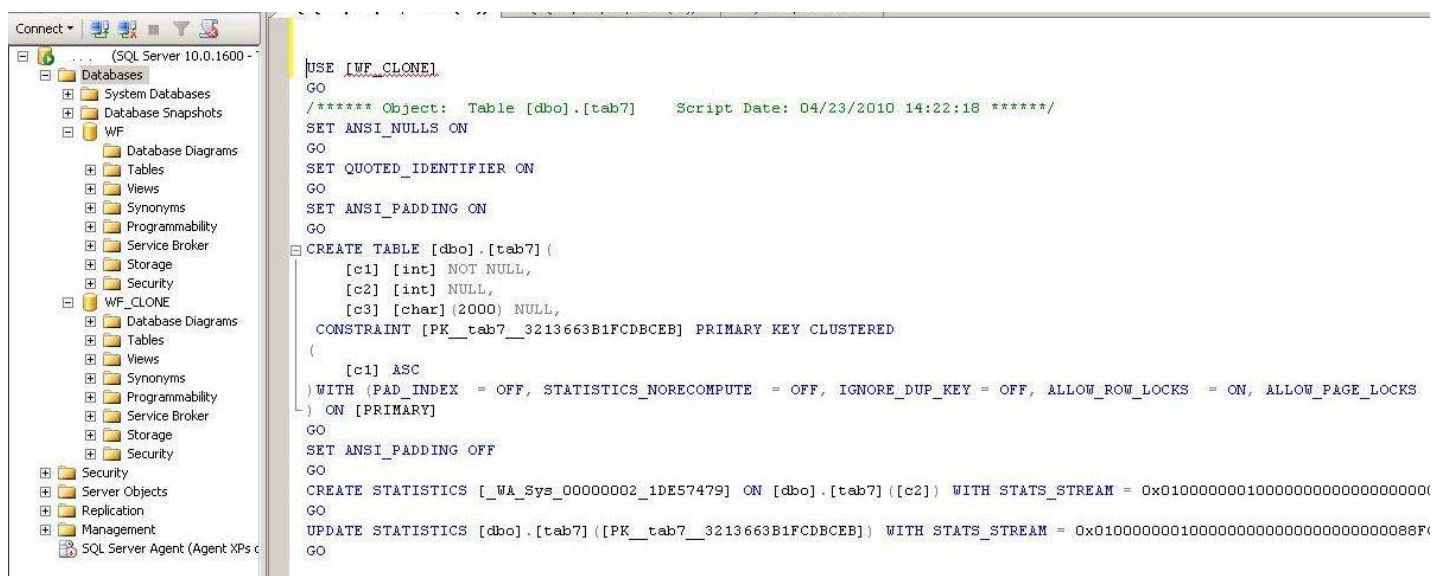
A while ago we were experiencing query performance issues and upper management decided to contact Microsoft Customer Service and Support (CSS). At the begging I was quite hesitant to contact them, because I knew they would ask for a copy of our database to help troubleshoot the issue and I did not want to do that due to the size of the database and for privacy reasons.

To my surprise, CSS did not ask us to provide a copy of our database, but instead asked us to provide a "Clone Database" or better known as a "Statistics-Only" copy of the database.

What is it Clone Database and how is it used? A clone database is a new database you create that contains the schema and statistics of the original database; it is useful because it provides a way to reproduce and diagnose query compilations and plan issues without actually having the data or the entire database, which makes it much smaller and more portable.

For example, if we have a query that does not perform well and we want to allow Microsoft Support or a third party consultant investigate and troubleshoot the query, we can provide a copy of the query plus our "Statistics-Only" copy of the database. The query, when run against the cloned database will still return the same query execution plans as it would if executed against the original database. The advantage is the size of the clone database is smaller the original database, plus the clone-database does not contain any data therefore privacy is maintained.

In the example below, I have created a clone of my test database WF. Please note the presence of the CREATE STATISTICS statements.



Now, if we check the execution plan of the following query against the clone database:

```
select a.* from dbo.tab7 a join dbo.tab7 b on a.c2=b.c2
```

We will notice the plan is exactly the same as the database that has the data. Table tab7 is empty on the clone database, however the optimizer estimated 54263.1 rows

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Databases' folder is expanded, showing a database named 'WF_CLONE'. The main window displays a script with the following SQL code:

```

[c2] [int] NULL,
[c3] [char] (2000) NULL,
CONSTRAINT [PK_tab7__3213663B1FCDBCEB] PRIMARY KEY CLUSTERED
(
    [c1] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
CREATE STATISTICS [_WA_Sys_00000002_1DE57479] ON [dbo].[tab7] ([c2]) WITH STATS_STREAM = 0x01000000010000000000000000000000
GO
UPDATE STATISTICS [dbo].[tab7] ([PK__tab7__3213663B1FCDBCEB]) WITH STATS_STREAM = 0x01000000010000000000000000000000088F
GO
select a.* from dbo.tab7 a join dbo.tab7 b on a.c2=b.c2

```

Below the script, the 'Execution plan' tab is selected, showing the query cost and the execution plan for the query:

```

Query 1: Query cost (relative to the batch): 100%
select a.* from dbo.tab7 a join dbo.tab7 b on a.c2=b.c2

```

The execution plan shows a 'Hash Match' operator (SELECT) with a cost of 47%. The plan also shows two 'Clustered Index Scan (Clustered)' operators for the tables 'dbo.tab7' and 'dbo.tab7', both with a cost of 47%.

Property	Value
Cached plan size	24 B
Estimated Operator Cost	0 (0%)
Estimated Subtree Cost	19.735
Estimated Number of Rows	54263.1

The statement being executed is:

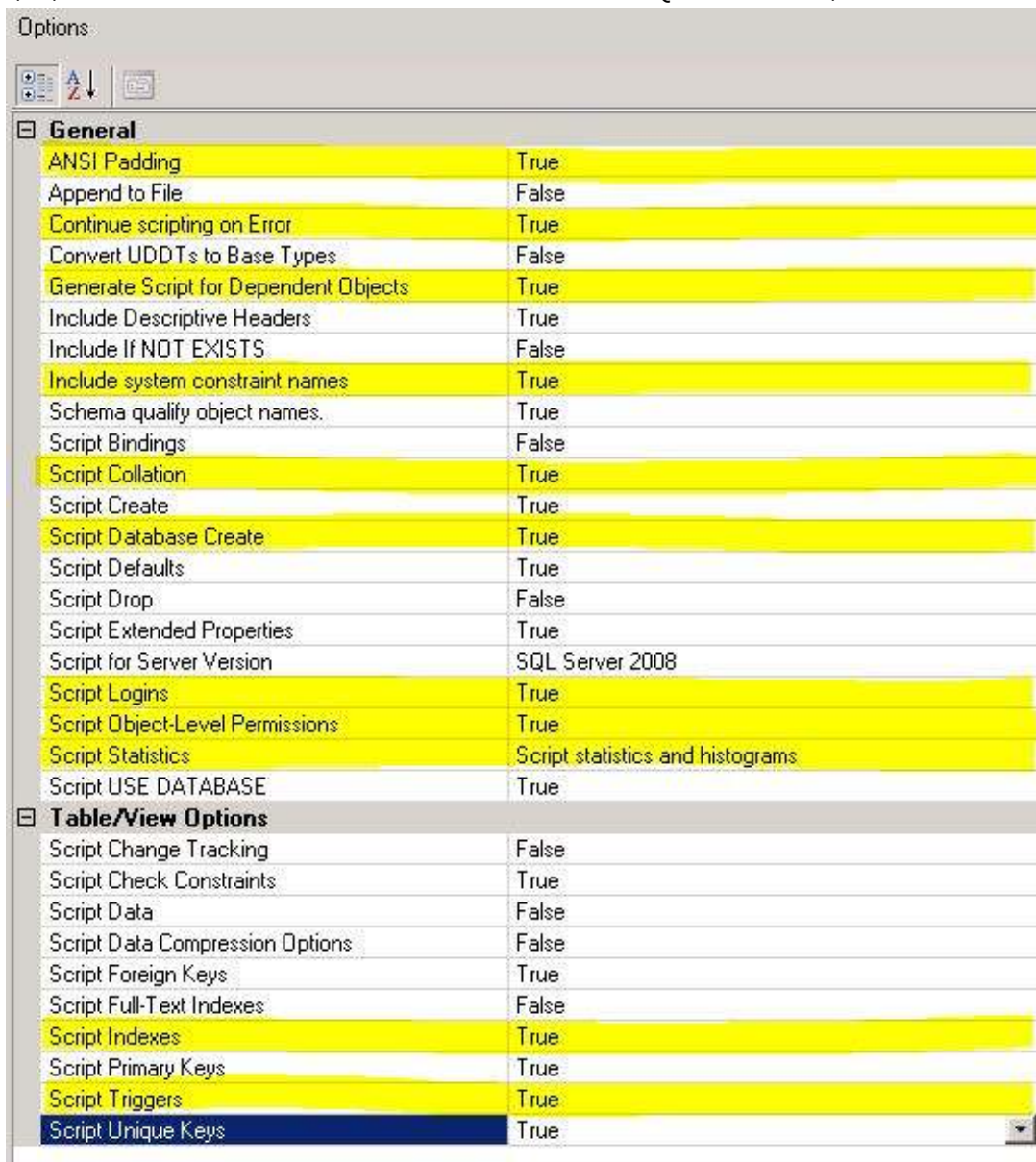
```

select a.* from dbo.tab7 a join dbo.tab7 b
on a.c2=b.c2

```

To create a clone database simply follow these steps:

- Open SQL Server Management Studio.
- In the Object Explorer, expand Databases and then locate the database that you want to script.
- Right-click on the database, point to Tasks and then click Generate Scripts...
- In the Script Wizard, verify that you have the correct database selected.
- Click to select the "Script all objects in the selected database" check box and then click Next.
- In the Choose Script Options dialog box, change the following settings from the default value to the values that shown below.
- A couple of things to note:
 - You do not need to script out logins if you are using the same instance of SQL Server.
 - If you use the USE DATABASE option make sure the script points to your clone database.
 - If you use the Script Database Create option you will need to change the script to use the name of the clone database not the original database name if you are creating this on the same instance.
 - The Script Logins option and the Script Object Level Permissions option may not be required unless the schema contains objects that are owned by logins other than dbo.



- Click Next.
- Click the Script to File option and select Single file then enter a file name and click Finish to save the script.
- Create a new empty database or use the script to create the database, it does not need to be that big since the only thing that will be in the database is the objects and the statistics, there will not be any data. If you use the script that is generated check and adjust the size of the data and log files and decrease the size instead of having very large files created.
- Execute the T-SQL in the script file that you just created to create the objects
- At this point your clone database is now ready to be used.

Once the clone database has been created it can be used to reproduce and diagnose, in a separate environment, query compilations and performance issues because the optimizer will have the following information available to select a query plan.

The following table of values has been copied from <http://support.microsoft.com/kb/914288> for your convenience.

Metadata

Constraints The query optimizer frequently uses constraints to detect contradictions between the query and the underlying schema. For example, if the query has a "WHERE col = 5" clause and a "CHECK (col < 5)" check constraint exists, the query optimizer knows that no rows will match.

The query optimizer makes similar types of deductions about nullability. For example, the "WHERE col IS NULL" clause is known to be true or false depending on the nullability of the column and whether the column is from the outer table of an outer join. The presence of FOREIGN KEY constraints is useful to determine cardinality and the appropriate join order. The query optimizer can use constraint information

to eliminate joins or simplify predicates. These changes may remove the requirement to access the base tables.

Statistics	The statistics information contains density and a histogram that shows the distribution of the leading column of the index and statistics key. Depending on the nature of the predicate, the query optimizer may use density, the histogram, or both to estimate the cardinality of a predicate. Up-to-date statistics are required for accurate cardinality estimates. The cardinality estimates are used as an input in estimating the cost of an operator. Therefore, you must have good cardinality estimates to obtain optimal query plans.
Table size (number of rows and pages)	The query optimizer uses the histograms and density to calculate the probability that a given predicate is true or false. The final cardinality estimate is calculated by multiplying the probability by the number of rows that are returned by the child operator. The number of pages in the table or the index is a factor in estimating the IO cost. The table size is used to calculate the cost of a scan, and it is useful when you estimate the number of pages that will be accessed during an index seek.
Database options	Several database options can affect optimization. The AUTO_CREATE_STATISTICS and AUTO_UPDATE_STATISTICS options affect whether the query optimizer will create new statistics or update statistics that are out of date. The PARAMETERIZATION level affects how the input query is parameterized before the input query is handed to the query optimizer. Parameterization can affect cardinality estimation and can also prevent matching against indexed views and other types of optimizations. The DATE_CORRELATION_OPTIMIZATION setting causes the optimizer to search for correlations between columns. This setting affects cardinality and cost estimation.

Environment

Session SET options	The ANSI_NULLS setting affects whether the "NULL = NULL" expression evaluates as true. Cardinality estimation for outer joins may change depending on the current setting. Additionally, ambiguous expressions may also change. For example, the "col = NULL" expression evaluates differently based on the setting. However, the "col IS NULL" expression always evaluates the same way.
Hardware resources	The cost for sort and hash operators depends on the relative amount of memory that is available to SQL Server. For example, if the size of the data is larger than the cache, the query optimizer knows that the data must always be spooled to disk. However, if the size of the data is much smaller than the cache, the operation is likely to be done in memory. SQL Server also considers different optimizations if the server has more than one processor and if parallelism has not been disabled by using a "MAXDOP" hint or the max degree of parallelism configuration option.
SQL Server Stock Keeping Unit (SKU)	Certain features are only enabled on specific editions of SQL Server 2005 or of SQL Server 2008. For example, matching a query against indexed views only occurs in SQL Server 2005 Enterprise Edition. Similarly, matching against plan guides is limited to SQL Server 2000 Standard Edition and SQL Server 2005 Enterprise Edition.

Next Steps

- Setup a clone database to see if you can troubleshoot a query performance issue. This will allow you to do this without touching your production databases.
- Take a look at these other [performance related tips](#)

Copyright (c) 2006-2010 [Edgewood Solutions, LLC](#) All rights reserved
[privacy statement](#) | [disclaimer](#) | [copyright](#)

Some names and products listed are the registered trademarks of their respective owners.