

CAPTURE AND DISPLAY EXECUTION TIME OF SQL QUERY IN SQL SERVER

This tutorial proposes 3 ways in order for you to get the **Execution time** of SQL Query or Stored Procedures are called or submitted to your SQL Server.

They will give you **durations in microseconds** and base on the execution time, you may have a deeper understand and will do some optimization for your database structure/indexing to make it runs better.

1. Using SQL Server Profiler

I think it's a easiest way for you to trace/track which Stored Procedures or SQL commands are running on SQL Server and how long it takes for each of **SQL Query/ Stored Procedure** execution.

		Fie Edit Yow Replay Taola Window Help											
-	2 2 1 1 1 1 1 1 1 1 1												
EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProcessID	SPID			
SQL:BatchCompl	DELETE FROM tblor	.Net SqlClie		wd_stu	0	34	0	2	6620	74			
SQL:BatchStarting	DELETE FROM tblor	.Net SqlClie		wd_stu					6620	76			
SQL:BatchCompl	DELETE FROM tblor	.Net SqlClie		wd_stu	16	34	0	2	6620	74			
SQL:BatchStarting	DELETE FROM tblor	.Net SqlClie		wd_stu					6620	76			
SQL:BatchCompl	DELETE FROM tblor	.Net sqlclie		wd_stu	0	0	0	0	6620	76			
RPC:Completed	exec sp_reset_con	.Net SqlClie		wd_stu	0	0	0	0	6620	76			
RPC:Completed	exec spGetCinema0	.Net sqlclie		wd_stu	0	0	0	0	662.0	76			
RPC:Completed	exec spGetCinemaS	.Net SqlClie		wd_stu	0	2	0	0	6620	76			
RPC:Completed	exec msGetNoviest	.Net sqlclie		wd_stu	67	5011	0	56	6620	63			
RPC:Completed	exec sp_reset_con	.Net SqlClie		wd_stu	0	0	0	0	6620	76			
RPC:Completed	exec spsaveorder	.Net sqlclie		wd_stu	0	6	0	0	6620	76			
RPC:Completed	exec spSaveOrderD	.Net SqlClie		wd_stu	0	9	0	0	6620	74			
RPC:Completed	exec spsaveOrderD	.Net sqlclie		wd_stu	0	9	0	0	6650	76			
SQL:BatchStarting	DELETE FROM T FRO	.Net SqlClie		wd_stu					6620	76			
SQL:BatchCompl	DELETE FROM T FRO	.Net sqlclie		wd_stu	94	2637	0	222	6620	57			
SQL:BatchStarting	DELETE FROM tb10r	.Net SqlClie		wd_stu	1000		12.1	Contraction of the local division of the loc	6620	52			
						1.0				100			

As you see, all commands are in **TextData** column and all Execution time for each are in **Duration** column respectively.

2. Using SQL Script with @StartTime and @EndTime parameters

The script should be run on SQL Server Management Studio Query.

USE AdventureWorksDW; GO
DECLARE @StartTime datetime,@EndTime datetime
SELECT @StartTime=GETDATE()
SELECT * FROM DimCustomer where Gender = 'M'
SELECT @EndTime=GETDATE()
SELECT DATEDIFF(ms,@StartTime,@EndTime) AS [Duration in microseconds]

Just replace your own SQL statements with line 2, after execute the statement, it will show the **Duration in microseconds** in another result panel.



	DECLARE @S	tartTime da	atetime,@EndTime	date	time			
	SELECT @St	artTime=GEI	TDATE ()					
	SELECT * F	ROM DimCust	comer where Gend	ler =	'M'			
	SELECT @En	dTime=GETDA	ATE ()					
	SELECT DAT	EDIFF (ms, @S	StartTime,@EndTi	.me) A	S [Durat	ion in mi	llisecs]	
_						6496		
						III		
	Results 🔓 Me	essages				m		
	Results 📑 Me CustomerKey	essages GeographyKey	CustomerAlternateKey	Title	FirstName	III MiddleName	LastName	NameStyl
			CustomerAlternate Key AW00011000	Title	FirstName Jon		LastName Yang	Name Styl
1	CustomerKey	GeographyKey		NULL	Jon	MiddleName	The second second	
1	CustomerKey 11000	GeographyKey 26	AW00011000	NULL NULL	Jon	MiddleName V	Yang	0
1 2 3	CustomerKey 11000 11001	GeographyKey 26 37	AW00011000 AW00011001	NULL NULL	Jon Eugene	MiddleName V L	Yang Huang	0
1 2 3	CustomerKey 11000 11001 11002	GeographyKey 26 37 31	AW00011000 AW00011001 AW00011002	NULL NULL NULL	Jon Eugene Ruben	MiddleName V L NULL	Yang Huang Torres	0 0 0
1 2 3 4 5	CustomerKey 11000 11001 11002 11005	GeographyKey 26 37 31 22	AW00011000 AW00011001 AW00011002 AW00011005	NULL NULL NULL NULL	Jon Eugene Ruben Julio	MiddleName V L NULL NULL	Yang Huang Torres Ruiz	0 0 0 0
	CustomerKey 11000 11001 11002 11005 11007	GeographyKey 26 37 31 22 40	AW00011000 AW00011001 AW00011002 AW00011005 AW00011007	NULL NULL NULL NULL NULL	Jon Eugene Ruben Julio Marco	MiddleName V L NULL NULL NULL	Yang Huang Torres Ruiz Mehta	0 0 0 0 0
1 2 3 3 4 5 5 7	CustomerKey 11000 11001 11002 11005 11007 11009	GeographyKey 26 37 31 22 40 25 22	AW00011000 AW00011001 AW00011002 AW00011005 AW00011007 AW00011009	NULL NULL NULL NULL NULL	Jon Eugene Ruben Julio Marco Shannon	MiddleName V L NULL NULL NULL C	Yang Huang Torres Ruiz Mehta Carlson	0 0 0 0 0 0
1 2 3 4 5	CustomerKey 11000 11001 11002 11005 11007 11009	GeographyKey 26 37 31 22 40 25 22	AW00011000 AW00011001 AW00011002 AW00011005 AW00011007 AW00011009 AW00011011	NULL NULL NULL NULL NULL	Jon Eugene Ruben Julio Marco Shannon	MiddleName V L NULL NULL NULL C	Yang Huang Torres Ruiz Mehta Carlson	0 0 0 0 0 0

3. USING SQL SCRIPT WITH SET STATISTICS TIME (TRANSACT-SQL)

It displays the number of milliseconds required to parse, compile, and execute each statement.

Run this SQL script on your SQL Query:

```
USE AdventureWorksDW;
GO
SET STATISTICS TIME ON
GO
SELECT * FROM DimCustomer where Gender = 'M'
GO
SET STATISTICS TIME OFF;
GO
```

And below is the result set:

Data: 06/12/2012 Total Chars: 1689



CAPTURE AND DISPLAY EXECUTION TIME OF SQL QUERY IN SQL SERVER 06/12/2012

```
SQL Server parse and compile time:
CPU time = 0 ms, elapsed time = 1 ms.
SQL Server parse and compile time:
CPU time = 0 ms, elapsed time = 1 ms.
(9351 row(s) affected)
SQL Server Execution Times:
CPU time = 63 ms, elapsed time = 479 ms.
SQL Server parse and compile time:
CPU time = 0 ms, elapsed time = 1 ms.
```

	The Edit Wext Replay Tools Window Help											
1 🗋 🚰 🛎 4	2 1 1 1 1 1 1 1 1 1 1											
EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProcessID	SPID		
SQL:BatchCompl	DELETE FROM tblor	.Net SqlClie		wd_stu	0	34	0	2	6620	76		
SQL:BatchStarting	DELETE FROM tblor	.Net Sqlclie		wd_stu					6620	76		
SQL:BatchCompl	DELETE FROM tblor	.Net sqlClie		wd_stu	16	34	0	2	6620	76		
SQL:BatchStarting	DELETE FROM tblor	.Net SqlClie		wd_stu					6620	76		
SQL:BatchCompl	DELETE FROM tblor	.Net SqlClie		wd_stu	0	0	0	0	6620	76		
RPC:Completed	exec sp_reset_con	.Net SqlClie		wd_stu	0	0	0	0	6620	76		
RPC:Completed	exec spGetCinema0	.Net sqlclie		wd_stu	0	0	0	0	6620	76		
RPC:Completed	exec spGetCinemaS	.Net SqlClie		wd_stu	0	2	0	0	6620	76		
RPC:Completed	exec msGetMoviest	.Net sqlclie		wd_stu	67	5011	0	56	6620	61		
RPC:Completed	exec sp_reset_con	.Net SqlClie		wd_stu	0	0	0	0	6620	76		
RPC:Completed	exec spsaveorder	.Net sqlclie		wd_stu	0	6	0	0	6620	76		
RPC:Completed	exec spSaveOrderD	.Net SqlClie		wd_stu	0	9	0	0	6620	76		
RPC:Completed	exec spsaveorderD	.Net sqlclie		wd_stu	0	9	0	0	6620	76		
SQL:BatchStarting	DELETE FROM T FRO	.Net SqlClie		wd_stu					6620	76		
SQL:BatchCompl	DELETE FROM T FRO	.Net sqlclie		wd_stu	94	2637	0	222	6620	57		
SQL:BatchStarting	DELETE FROM tb10r	.Net SqlClie		wd_stu			2.1	-	6620	57		
										2		

That's all! Feel free to contribute your own solution by submitting your comments as you are always be welcome.

