

## References

- [1] G. Ramakrishnan. Database Management Systems, Third Edition. McGraw-Hill, 2003
- [2] Fox, B. 2011. "Leveraging Big Data for Big Impact", Health Management Technology, <http://www.healthmgttech.com/>.
- [3] A. Hameurlain, "Evolution of Query Optimization Methods: From Centralized Database Systems to Data Grid Systems", Proceedings of the 20th International Conference on Database and Expert Systems Applications.
- [4]. Andrew N.K. Chen. Robust optimization for performance tuning of modern database Systems, European Journal of Operational Research. 171, 412--429 (2006)
- [5] The State of the Art in Distributed Query Processing DONALD KOSSMANN, University of Passau, ACM Computing Surveys, Vol. 32, No. 4, December 2000.
- [6] Jacobs, A. 2009. "Pathologies of Big Data", Communications of the ACM, 52(8):36-44.
- [7] JASON. 2008. "Data Analysis Challenges", The Mitre Corporation, McLean, VA, JSR-08-142
- [8] Kaisler, S. 2012. "Advanced Analytics", CATALYST Technical Report, i\_SW Corporation, Arlington, VA
- [9]. Rasha Osman, Irfan Awan, Michael E. et al.: QuePED-Revisiting queuing networks for the Performance evaluation of database designs. Simulation Modeling Practice and Theory, 19, 251--270 (2011)
- [10]. Balsamo, S., A. Di Marco, P. Inverardi and M. Simeoni, Model-based performance Prediction in software development: a survey, IEEE Transactions on Software Engineering. 30 ,5, 295--310 (2004)
- [11] K. Ono and G.M.Lohman. Measuring the complexity of join enumeration in query optimization. In D.McLeod, R Sacks-Davis, and J.-J. schek, editors,16th International Conference on Very Large Data Bases, August 13-16,1990, Brisbane, Queensland, Australia, Proceedings, pages 314-325. Morgan Kaufmann, 1990.
- [12] A. Hameurlain, "Evolution of Query Optimization Methods: From Centralized Database Systems to Data Grid Systems", Proceedings of the 20th International Conference on Database and Expert Systems Applications.

- [13] Fox, B. 2011. "Leveraging Big Data for Big Impact", Health Management Technology, <http://www.healthmgttech.com/>.
- [14] [https://hadoop.apache.org/docs/r1.2.1/mapred\\_tutorial.html](https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html).
- [15] Gantz, J. and E. Reinsel. 2011. "Extracting Value from Chaos", IDC's Digital Universe Study, sponsored by EMC.
- [16] <https://cs.uwaterloo.ca/~kmsalem/courses/.../Chalamalla-HadoopDB.pdf>
- [17] <https://en.wikipedia.org/wiki/Greenplum>
- [18] <https://pig.apache.org/>
- [19] [www.cs.rutgers.edu/~zz124/cs671.../srikanth\\_mapreducemerge.pdf](http://www.cs.rutgers.edu/~zz124/cs671.../srikanth_mapreducemerge.pdf). Map-Reduce-Merge: Simplified Relational Data. Processing on Large. Clusters. Hung-chih Yang, Ali Dasdan. Yahoo! Ruey-Lung Hsiao, D. Sto Parker.
- [20] <http://www.journalofcloudcomputing.com/content/3/1/12>. Improving the performance of Hadoop Hive by sharing scan and computation tasks Tansel Dokeroglu1, Serkan Ozall1, Murat Ali BayMuhammetSerkanCinar3 and Ahmet Cosar1.
- [21] Liu et al. "An Investigation of Practical Approximate Nearest Neighbor Algorithms", 2004. Carnegie-Mellon University, pp. 1-8.
- [22] [www.elsevier.com/locate/jcss](http://www.elsevier.com/locate/jcss), Journal of Computer and System Sciences 77 (2011) 637-651.
- [23] Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis, IJCAI-07 1606, Evgeniy Gabrilovich and Shaul Markovitch Department of Computer Science Technion—Israel Institute of Technology, 32000 Haifa, Israel
- [24] {gabr,shaulm}@cs.technion.ac.il.
- [25] <https://en.wikipedia.org/wiki/MapReduce>.
- [26] Applied Spatial Data Analysis with R Authors: Roger S. Bivand, Edzer Pebesma, Virgilio Gómez-Rubio.
- [27] A twelve-analyzer detector system for high-resolution powder diffraction P. L. Lee, D. Shu, M. Ramanathan, C. Preissner, J. Wang, M. A. Beno, R. B. Von Dreele, L. Ribaud, C. Kurtz, S. M. Antao, X. Jiao and B. H. Toby. J. Synchrotron Rad. (2008). 15, 427-432.

# Appendixes

## Appendix A - User interface and architecture diagram of the system.

### Security Module - Authentication

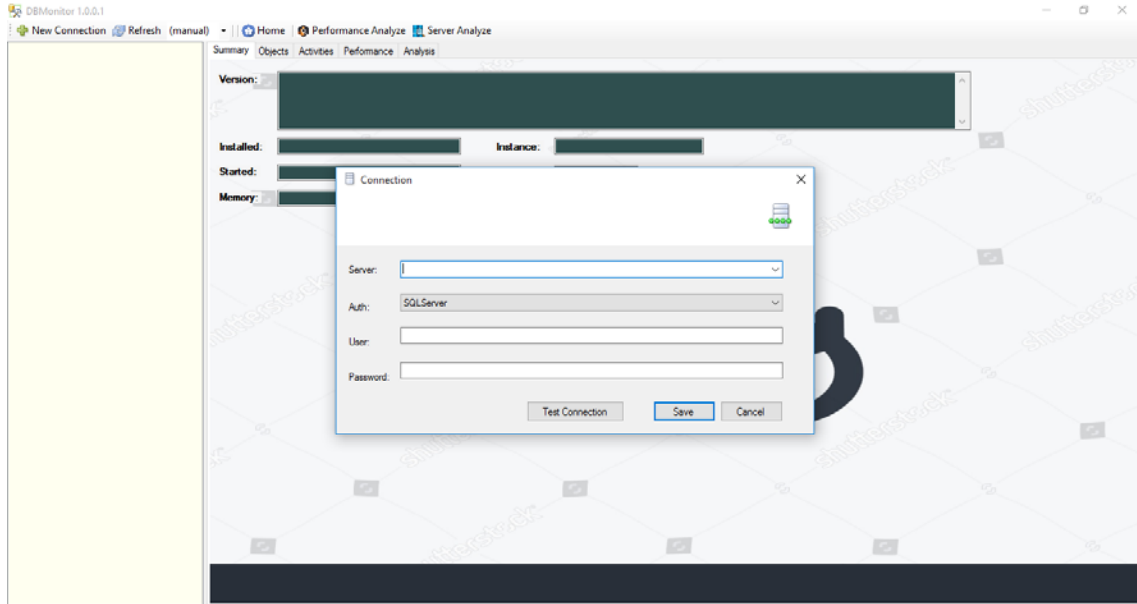


Figure 5.1 – Security Module - Authentication

### Control Module – Server and Database Information

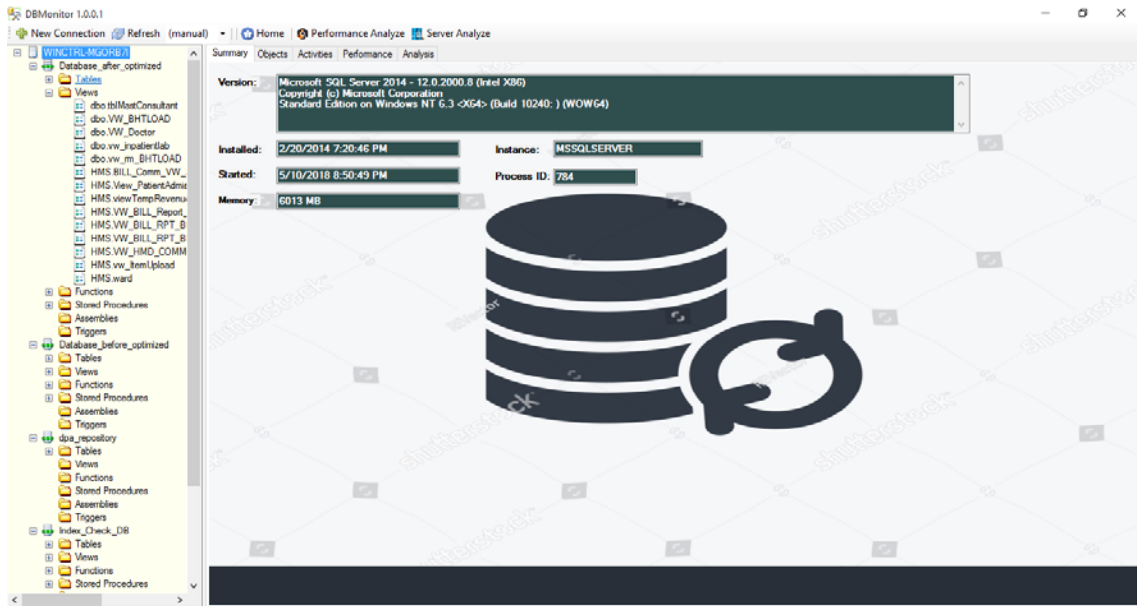


Figure 5.2 – Control Module-Server and Database Information

## Server Configuration

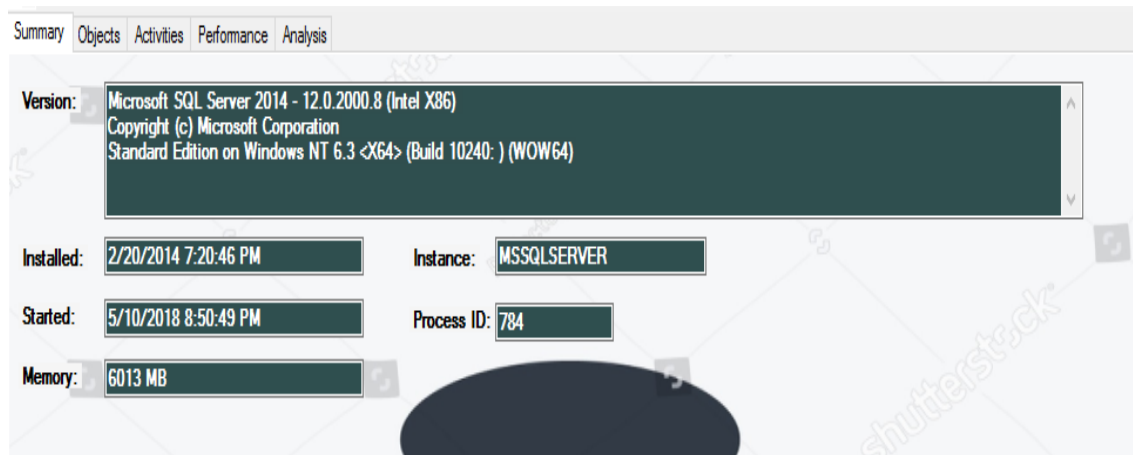


Figure 5.3 – Server Configuration

## Database Server Performance Analyzer

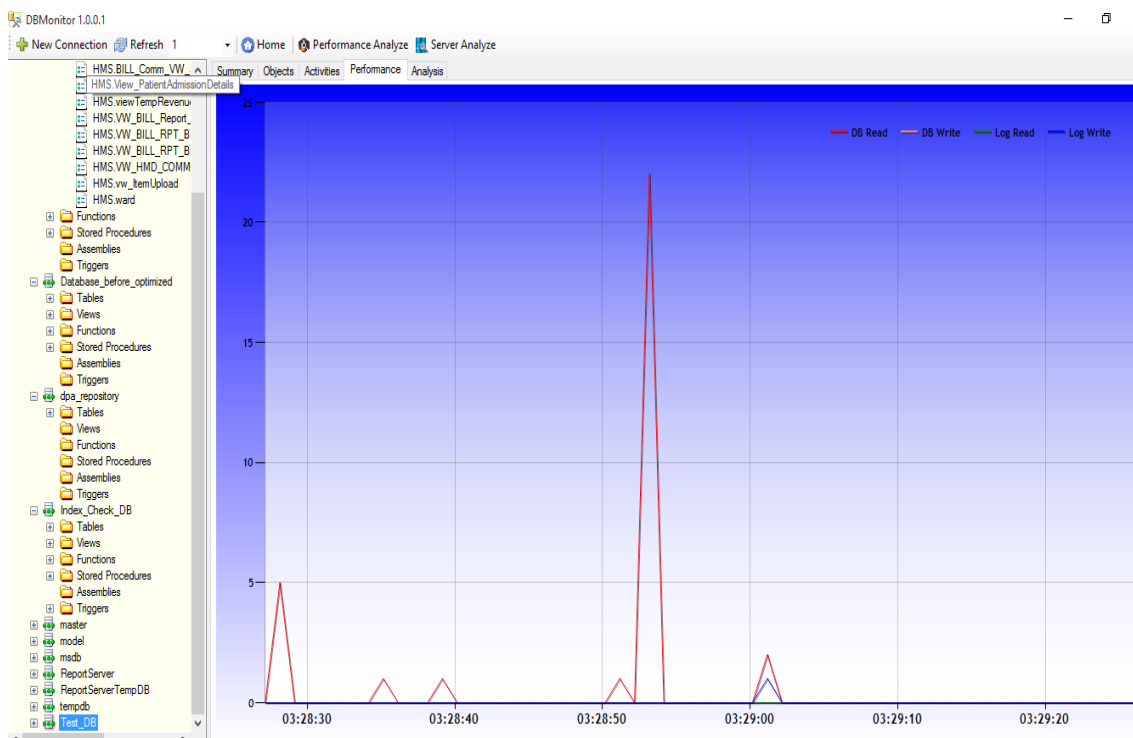


Figure 5.4 – Database Server Performance Analyzer

## Database Log Information and Suggestions

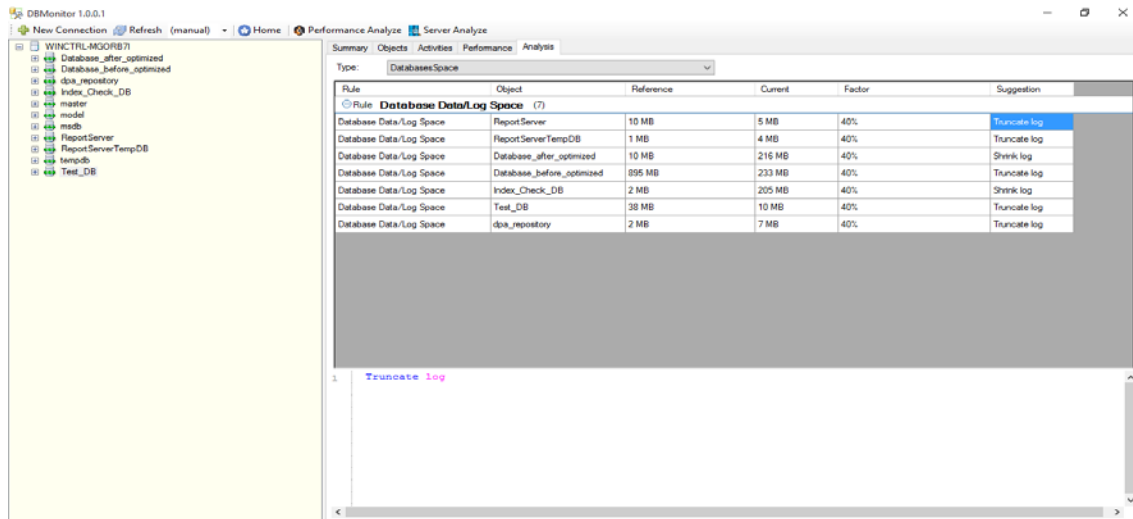


Figure 5.5 – Database Log Information and Suggestions.

## Database Performance Improvement Suggestions

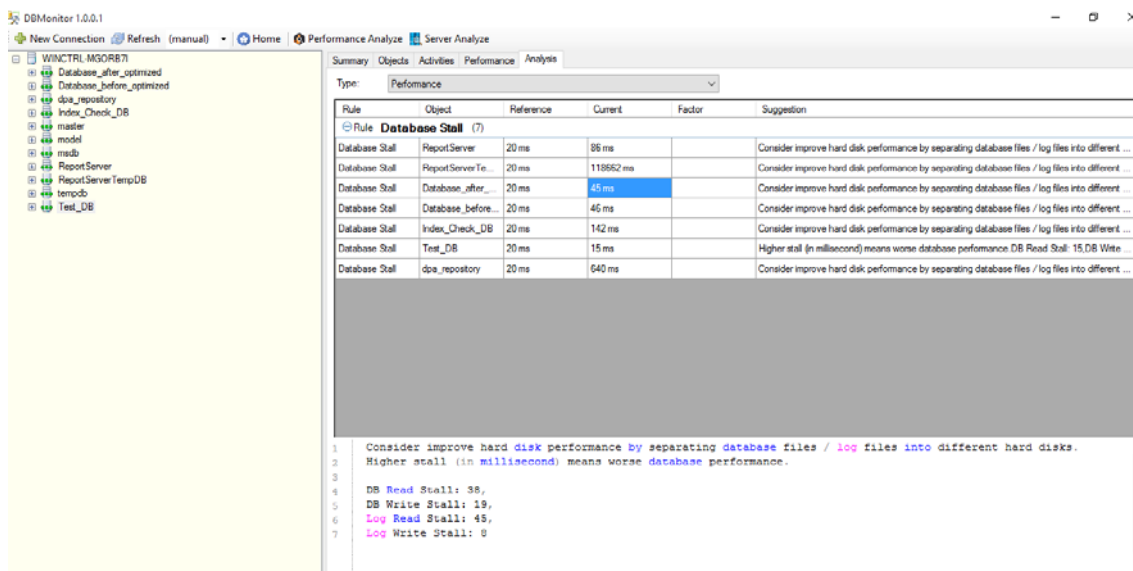


Figure 5.6 – Database Performance Improvement Suggestions.

# Database Waiting Tasks

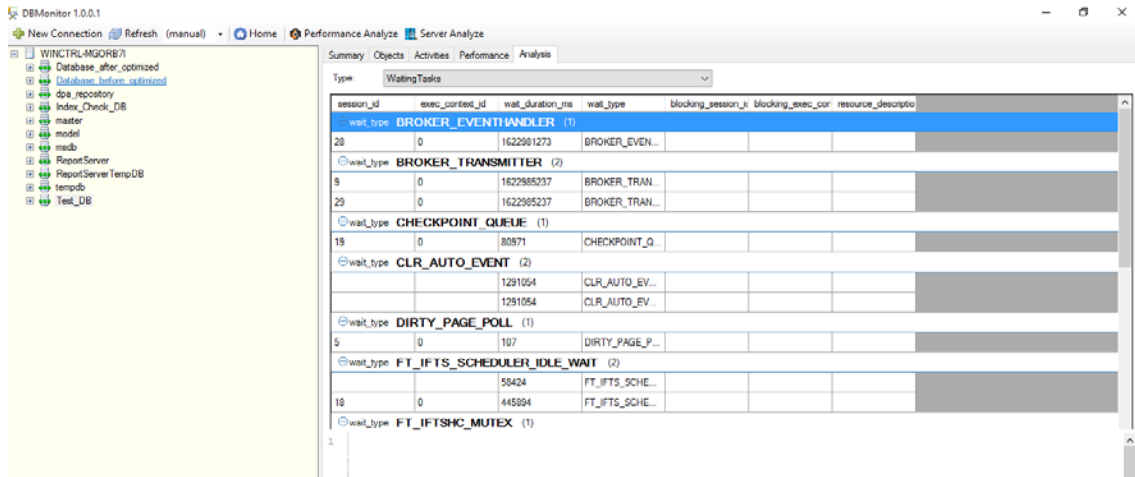


Figure 5.7 – Database Waiting Tasks.

# Database Missing Index Details and Suggestions

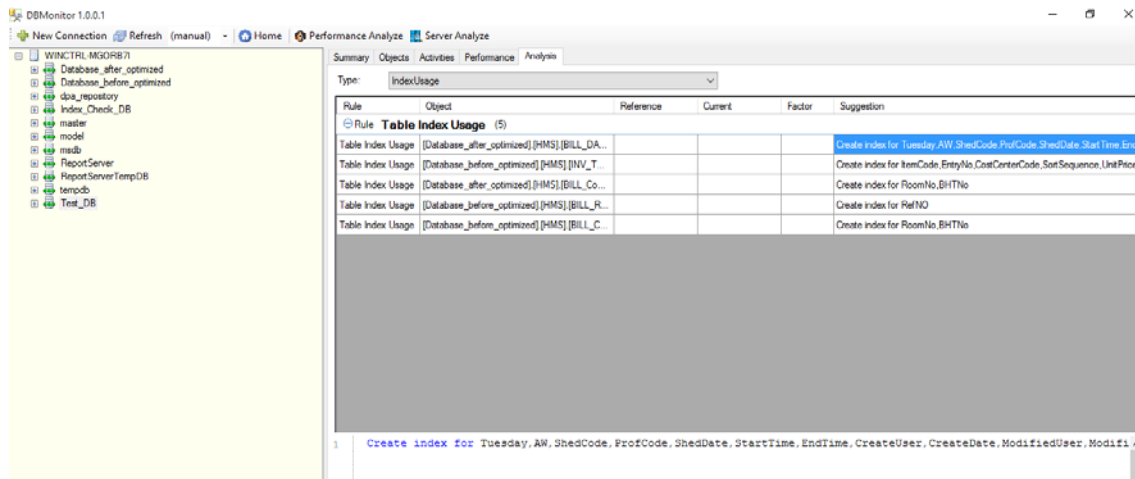


Figure 5.8 – Database Missing Index Details and Suggestions

# Database IO operations

logtime	creation_time	last_execution_time	query_text	total_worker_time	AvgCPUTime	LogicalReads	LogicalWrites	execution_count	AggIO
DatabaseName (1)									
5/29/2018 3:51 ...	5/29/2018 3:27 ...	5/29/2018 3:39 ...	insert @tab selec...	150.987000	0.262130208333...	4052	1	576	4053
DatabaseName Database after optimized (1)									
5/29/2018 3:51 ...	5/29/2018 3:21 ...	5/29/2018 3:51 ...	SELECT su.nam...	9.155000	4.577500000000...	736	0	2	736
DatabaseName dpo_repository (1)									
5/29/2018 3:51 ...	5/29/2018 3:21 ...	5/29/2018 3:51 ...	select ID,NAME...	44.346000	0.246366868686...	724	0	180	724
DatabaseName master (7)									
5/29/2018 3:51 ...	5/29/2018 3:47 ...	5/29/2018 3:47 ...	SELECTSCHEM...	90.503000	90.503000000000...	91250	0	1	91250
5/29/2018 3:51 ...	5/29/2018 3:21 ...	5/29/2018 3:51 ...	SELECT * FROM...	45.969000	3.536076923076...	3821	0	13	3821
5/29/2018 3:51 ...	5/29/2018 3:50 ...	5/29/2018 3:50 ...	select top 20 get...	26.900000	26.900000000000...	3001	66	1	3067
5/29/2018 3:51 ...	5/29/2018 3:47 ...	5/29/2018 3:47 ...	SELECTSCHEM...	29.782000	29.782000000000...	2340	0	1	2340
5/29/2018 3:51 ...	5/29/2018 3:26 ...	5/29/2018 3:47 ...	SELECTISNULL...	16.382000	0.481823529411...	1569	0	34	1569
5/29/2018 3:51 ...	5/29/2018 3:47 ...	5/29/2018 3:51 ...	SELECT TOP 20...	519.374000	4.764899032568...	545	0	109	545
5/29/2018 3:51 ...	5/29/2018 3:28 ...	5/29/2018 3:28 ...	SELECTPrognam...	4.913000	4.913000000000...	409	0	1	409
DatabaseName ReportServer (2)									
5/29/2018 3:51 ...	5/29/2018 3:21 ...	5/29/2018 3:51 ...	select top 4 ...	48.855000	0.272932960893...	1516	0	179	1516

Figure 5.9 – Database IO Operations

# Database Objects and Details

Name	Space	Count	Create Date	Modify Date	Path
Database_after_optimized	216MB / 9.9375	0			C:\Program Files (x86)\Microsoft SQL Server\MSSQL12\MSSQLSERVER\...
Database_before_optimized	233MB / 894.87...	0			C:\Program Files (x86)\Microsoft SQL Server\MSSQL12\MSSQLSERVER\...
dpo_repository	7.1875MB / 1.75...	0			C:\Program Files (x86)\Microsoft SQL Server\MSSQL12\MSSQLSERVER\...
Index_Check_DB	205.25MB / 2MB	0			C:\Program Files (x86)\Microsoft SQL Server\MSSQL12\MSSQLSERVER\...
master	4MB / 2MB	0			C:\Program Files (x86)\Microsoft SQL Server\MSSQL12\MSSQLSERVER\...
model	2.1875MB / 0.75...	0			C:\Program Files (x86)\Microsoft SQL Server\MSSQL12\MSSQLSERVER\...
msdb	15.5625MB / 1MB	0			C:\Program Files (x86)\Microsoft SQL Server\MSSQL12\MSSQLSERVER\...
ReportServer	5.1875MB / 10.1...	0			C:\Program Files (x86)\Microsoft SQL Server\MSSQL12\MSSQLSERVER\...
ReportServerTempDB	4.1875MB / 1.06...	0			C:\Program Files (x86)\Microsoft SQL Server\MSSQL12\MSSQLSERVER\...
tempdb	8MB / 0.5MB	0			C:\Program Files (x86)\Microsoft SQL Server\MSSQL12\MSSQLSERVER\...
Test_DB	10MB / 38.375MB	0			C:\Program Files (x86)\Microsoft SQL Server\MSSQL12\MSSQLSERVER\...

Figure 5.10 – Database Objects and Details

# Database Monitoring Application Options

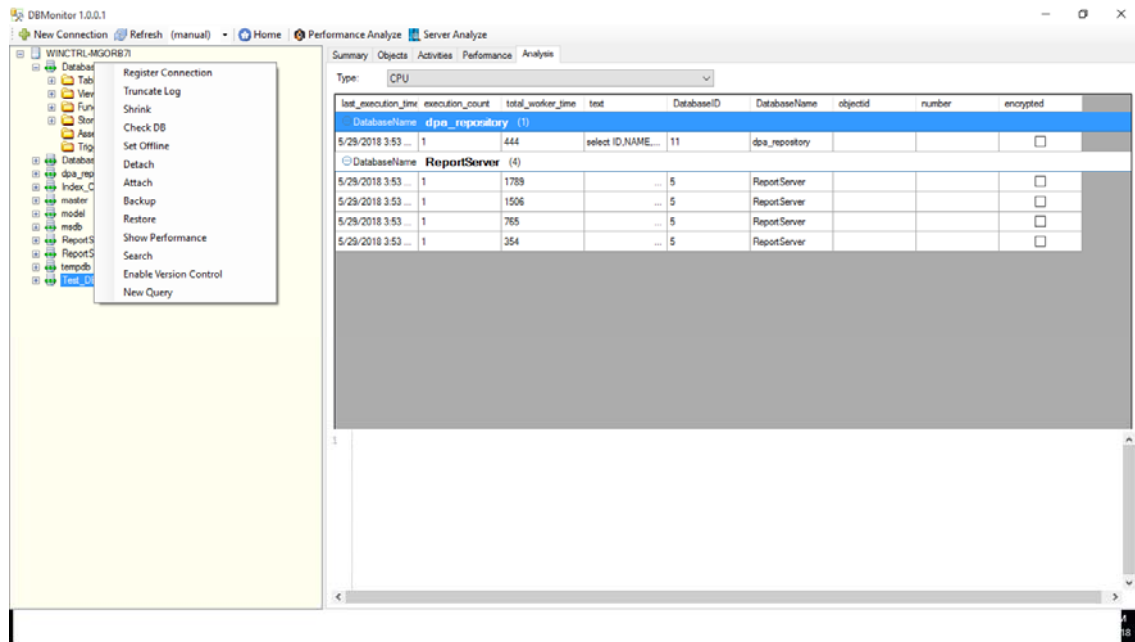


Figure 5.11 – Database Monitoring Application Options



## Appendix B – Evaluation of Database Monitoring Application

<b>Functionality</b>	<b>Result in our approach</b>	<b>The result from manually check in the database</b>	<b>Are the both results are same?</b>
<b>Server information</b>	Please refer figure 7.4.1.1	Please refer figure 7.4.1.2	<b>Yes</b>
<b>Database Table Count</b>	Please refer figure 7.4.1.3	Please refer figure 7.4.1.4	<b>Yes</b>
<b>Database current reading count</b>	Please refer figure 7.4.1.3	Please refer figure 7.4.1.4	<b>Yes</b>
<b>Database current writing count</b>	Please refer figure 7.4.1.3	Please refer figure 7.4.1.4	<b>Yes</b>
<b>Missing index details</b>	Please refer figure 7.4.1.5	Please refer figure 7.4.1.6	<b>Yes</b>
<b>Database current reading count</b>	Please refer figure 7.4.1.3	Please refer figure 7.4.1.4	<b>Yes</b>
<b>Database memory utilization details</b>	Please refer figure 7.4.1.7	Manually checked	<b>Yes</b>
<b>Database lock</b>	Please refer figure 7.4.1.8	Manually checked	<b>Yes</b>
<b>Currently running Processors</b>	Please refer figure 7.4.1.9	Manually checked	<b>Yes</b>

Table 7.1 – Evaluation functionality in database monitoring application

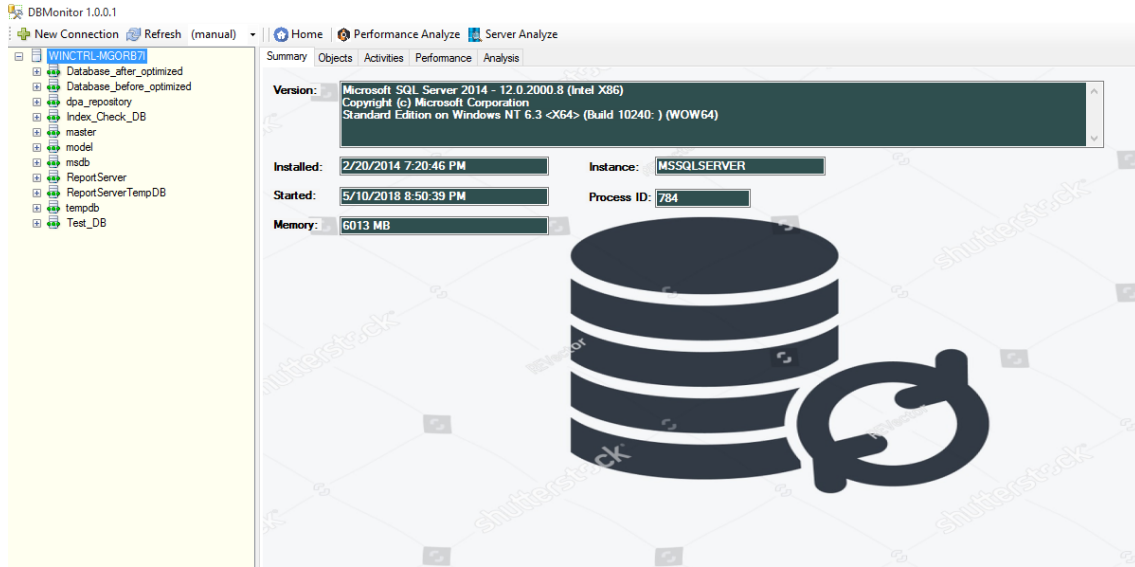


Figure 7.4.1.1 – Database server information from newly developed database monitoring application

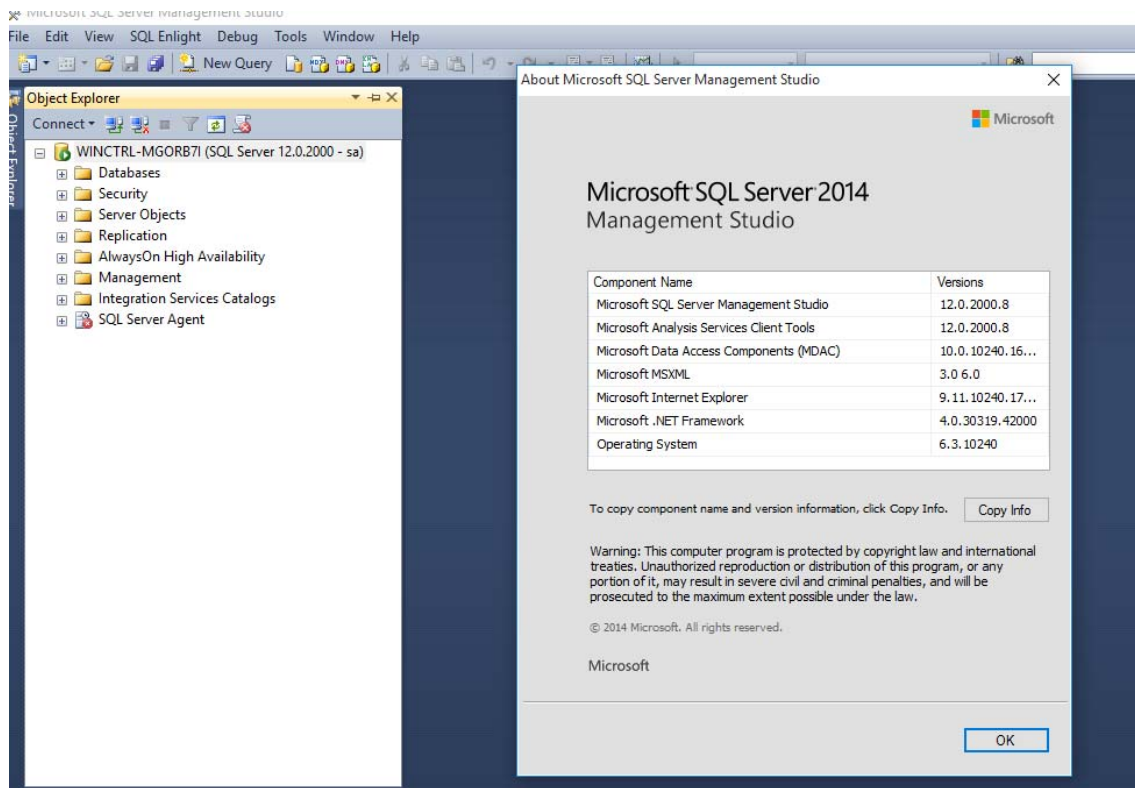


Figure 7.4.1.2 – Database server information



Figure 7.4.1.3 - Database server statistics from newly developed database monitoring application

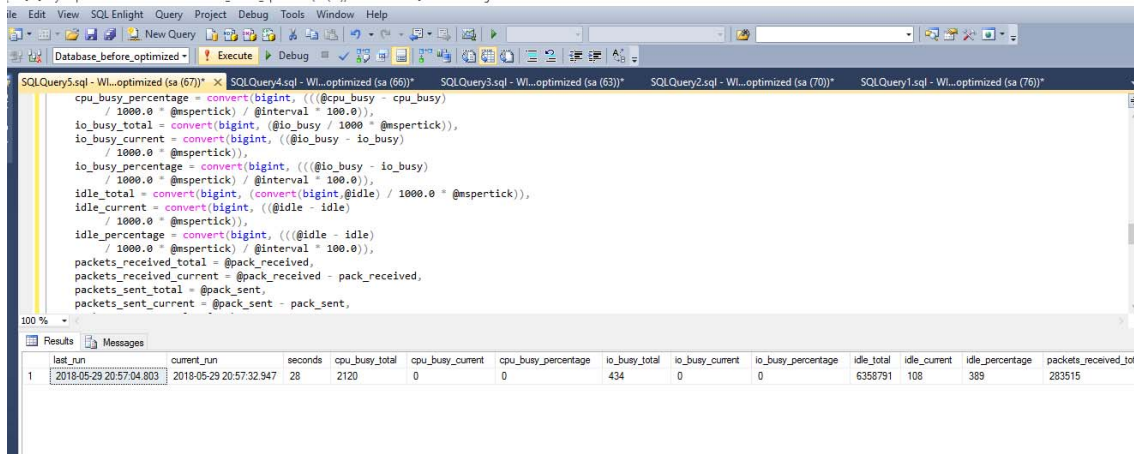


Figure 7.4.1.4 - Database server statistics

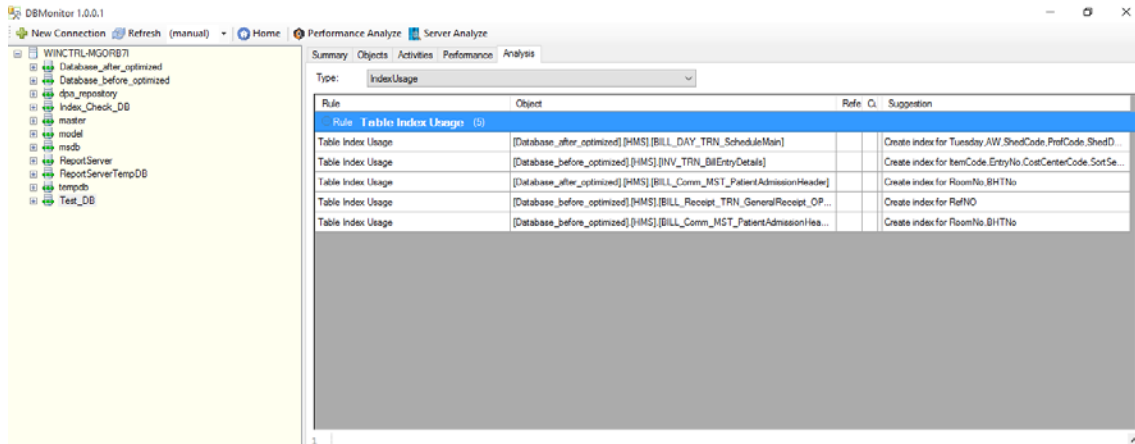


Figure 7.4.1.5 – Missing index suggestions from newly developed database monitoring application

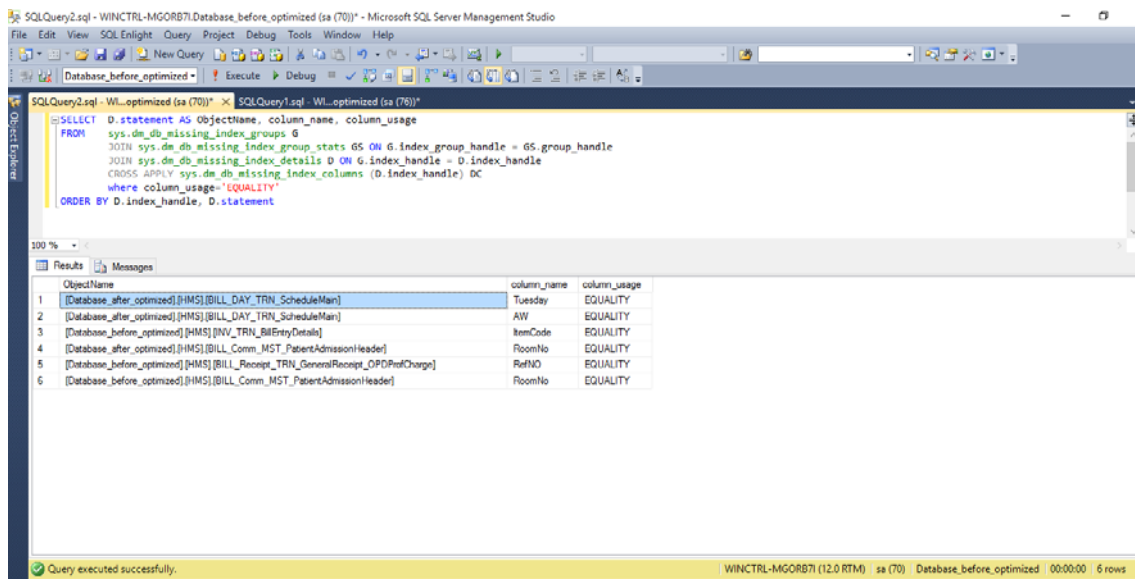


Figure 7.4.1.6 – Missing index suggestions by manually

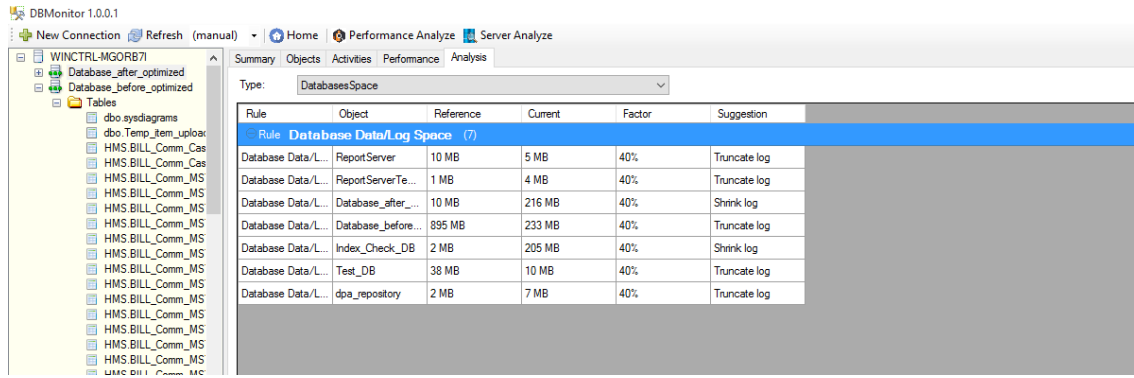


Figure 7.4.1.7 – Database memory utilization details

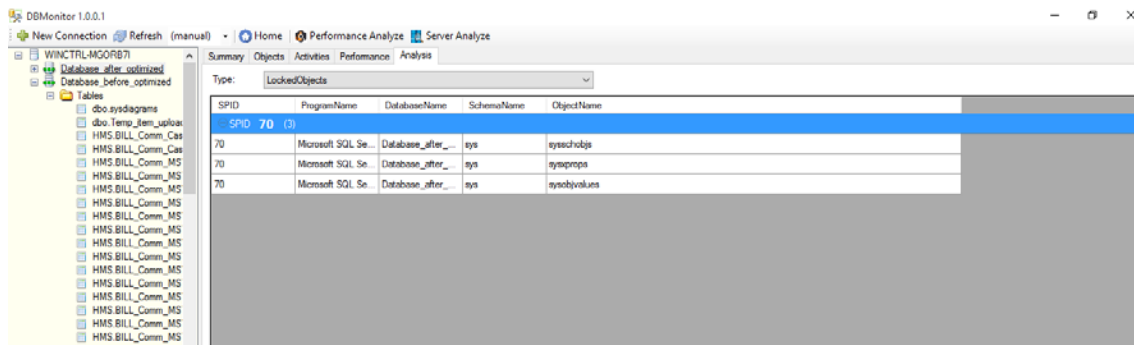


Figure 7.4.1.7 – Database lock

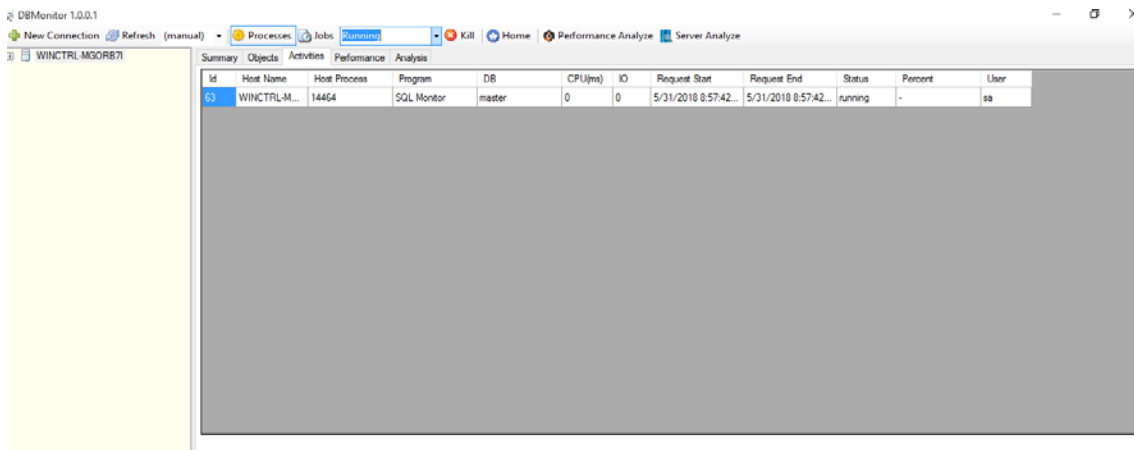


Figure 7.4.1.8 – Currently running Processors

## Appendix C – Evaluation of proposed optimization techniques

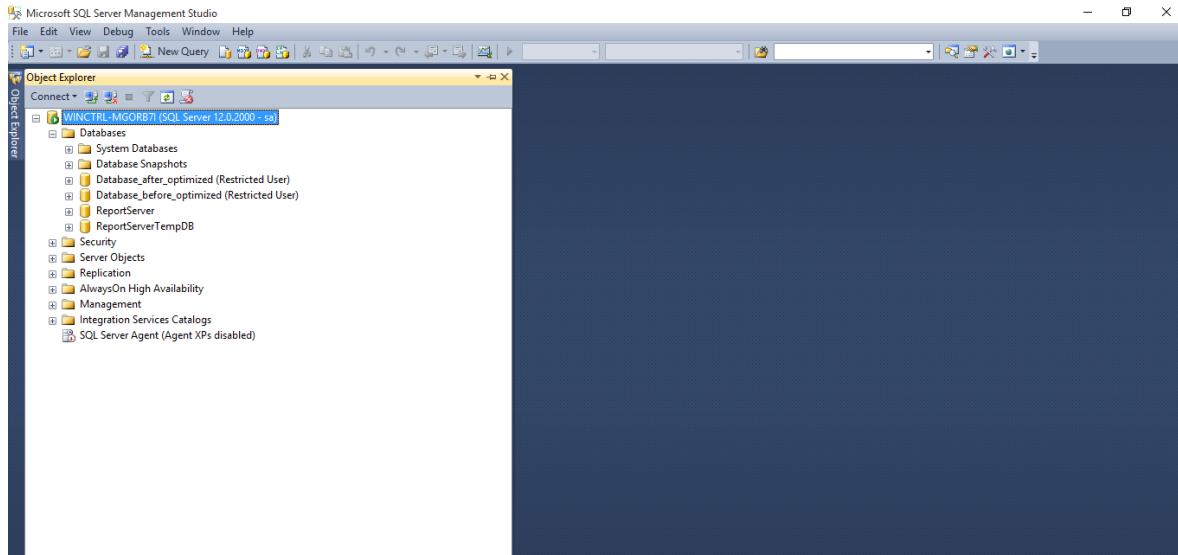


Figure 6.1 – Database Configuration

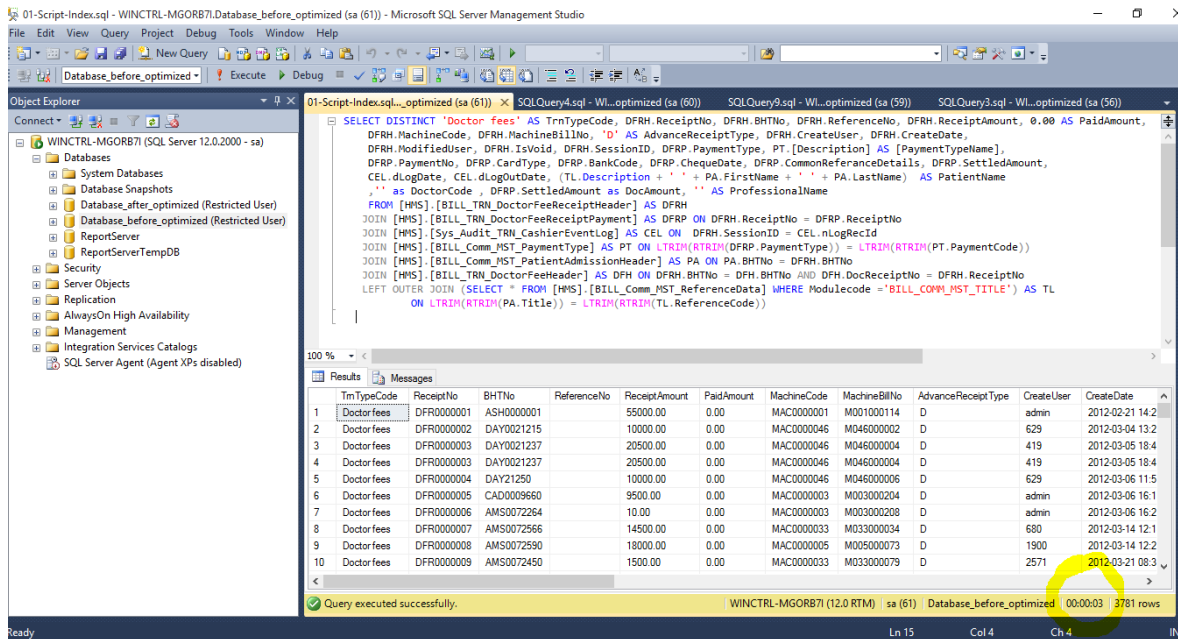


Figure 6.2 – Complex Query Execution Time

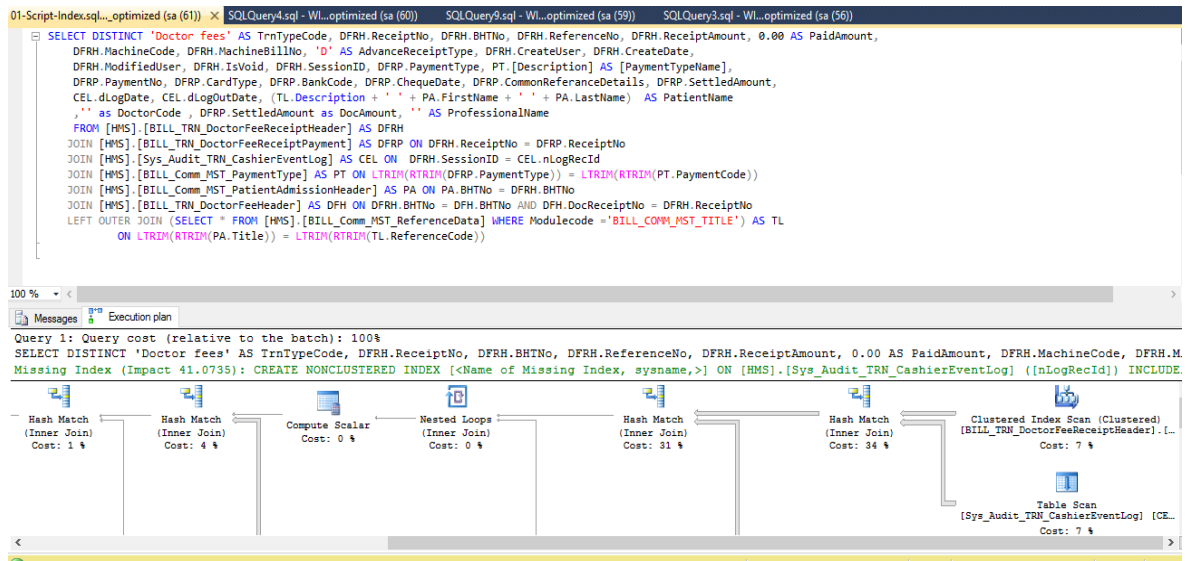


Figure 6.3 - QEP Plan

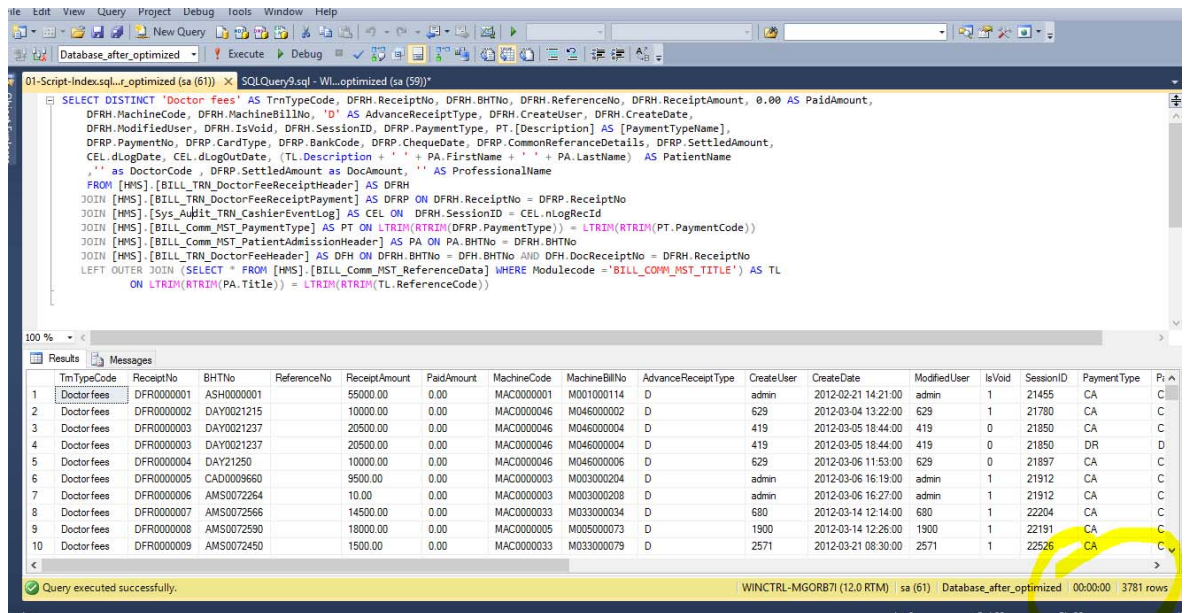


Figure 6.4 – Query Execution Time After Optimized

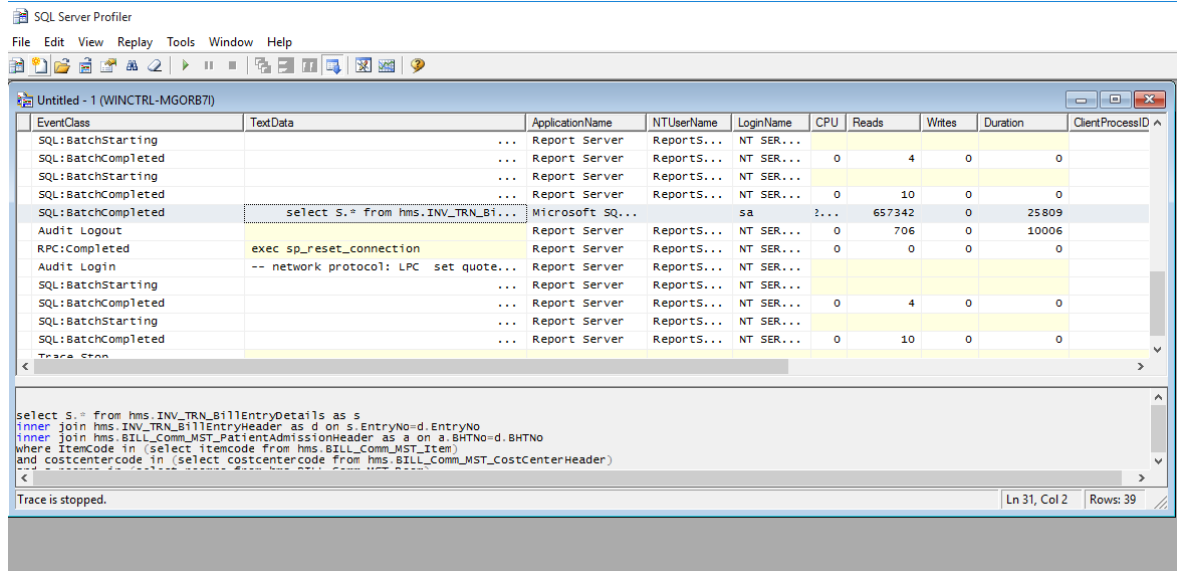


Figure 6.5 – SQL Profiler

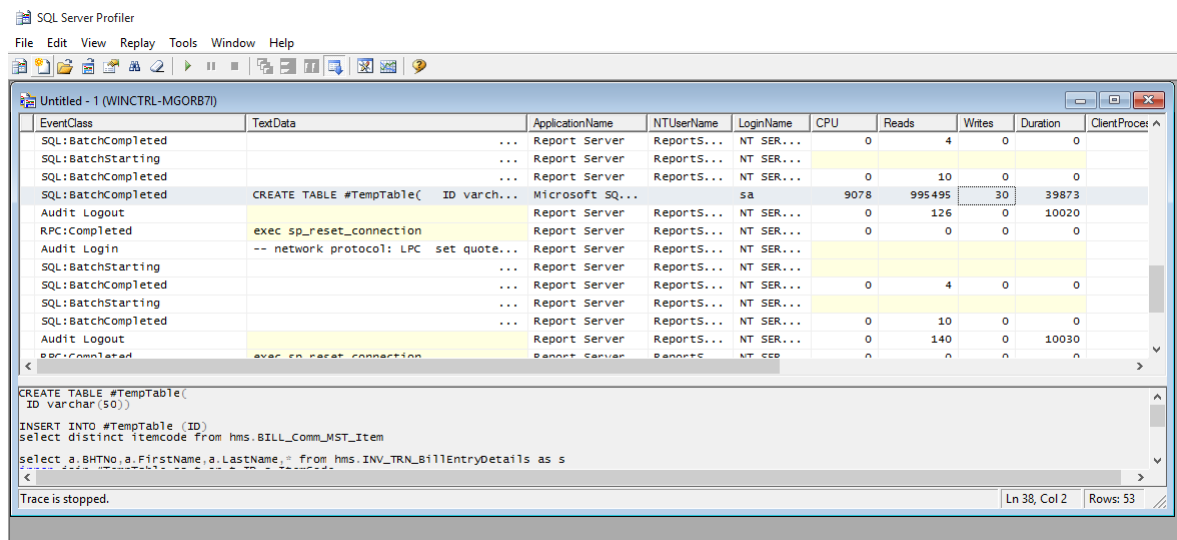


Figure 6.6 – SQL Profiler result



```

01-in operator.sql -..._optimized (sa (53)) X
set statistics time on

SELECT
S.*
FROM hms.INV_TRN_BillEntryDetails AS s
INNER JOIN hms.INV_TRN_BillEntryHeader AS d
ON s.EntryNo = d.EntryNo
INNER JOIN hms.BILL_Comm_MST_PatientAdmissionHeader AS a
ON a.BHTNo = d.BHTNo
WHERE ItemCode IN (SELECT DISTINCT
itemcode
FROM hms.BILL_Comm_MST_Item)
AND costcentercode IN (SELECT
costcentercode
FROM hms.BILL_Comm_MST_CostCenterHeader)
AND a.roomno IN (SELECT
roomno

```

100 %

Results Messages

(255180 row(s) affected)

SQL Server Execution Times:  
CPU time = 2906 ms, elapsed time = 56866 ms.

Figure 6.7 – SQL Server Execution time for Traditional query

```

02-in operator remo...optimized (sa (55)) X 01-in operator.sql -..._optimized
itemcode
FROM hms.BILL_Comm_MST_Item
create nonclustered index IX_Itemcode on #TempTable(ID
SELECT
S.*
FROM hms.INV TRN BillEntryDetails AS s

```

00 %

Results Messages

(255180 row(s) affected)

SQL Server Execution Times:  
CPU time = 4141 ms, elapsed time = 5446 ms.

SQL Server Execution Times:  
CPU time = 0 ms, elapsed time = 0 ms.

Figure 6.8 – SQL Server Execution time for our new proposed query

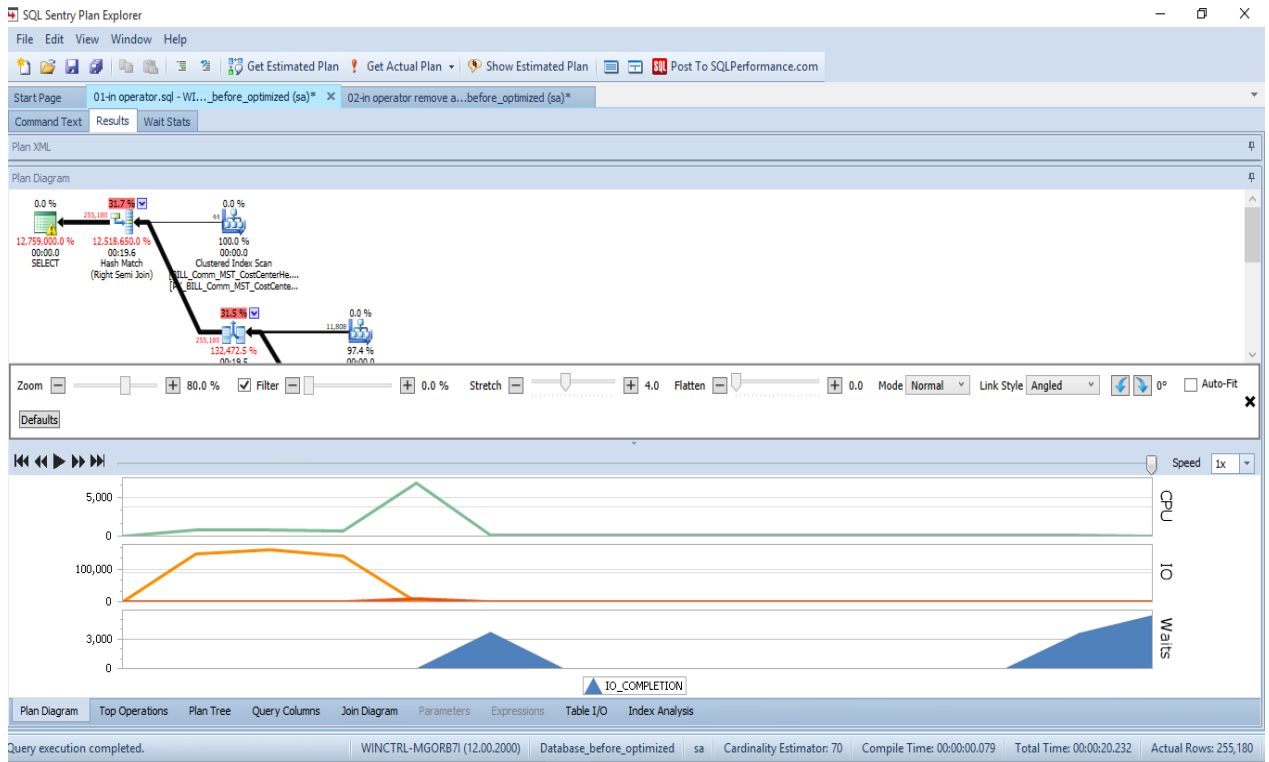


Figure 6.9 - Analyze by using Sentry Plan explore with IN

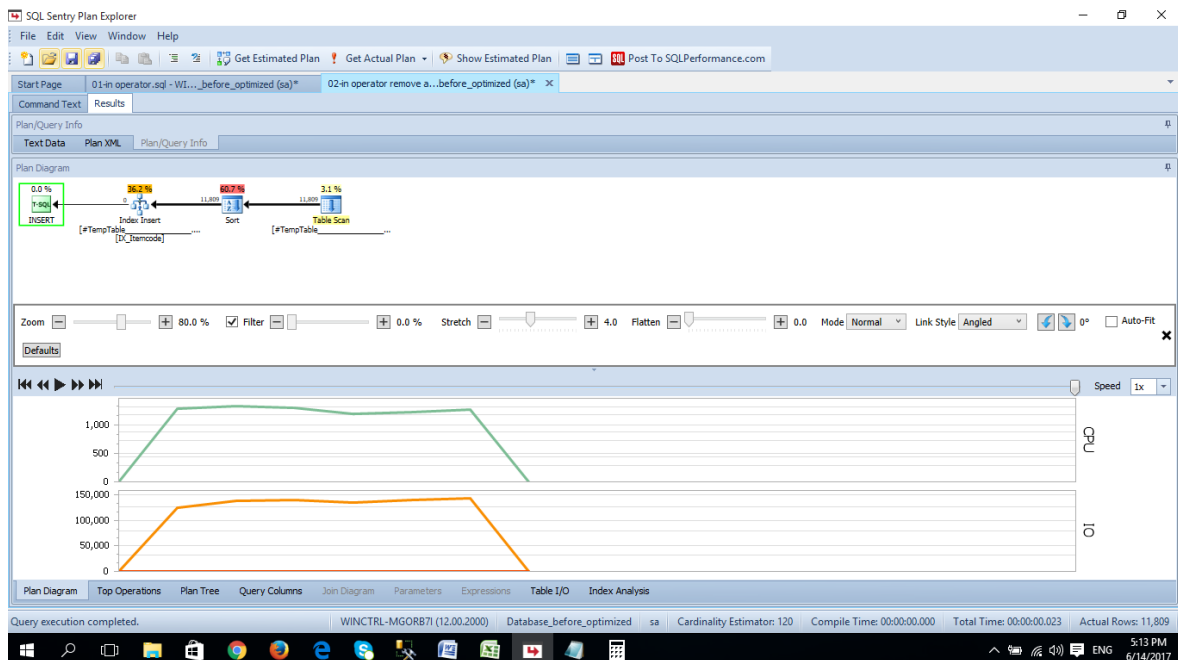


Figure 6.10 - Analyze by using Sentry Plan explore without IN

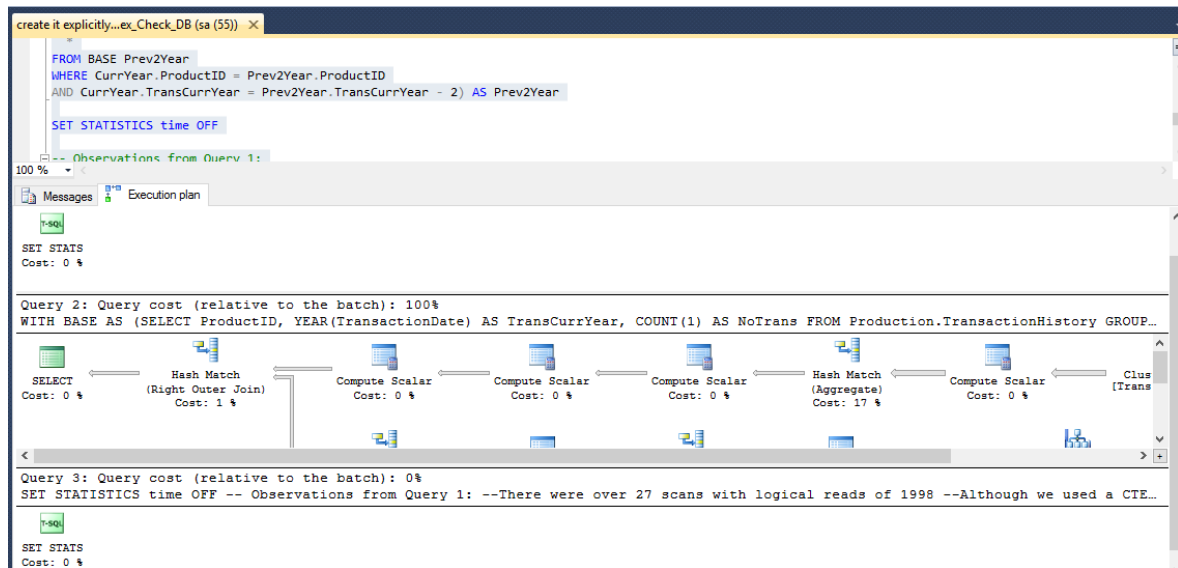


Figure 6.11 – Query cost with temp table

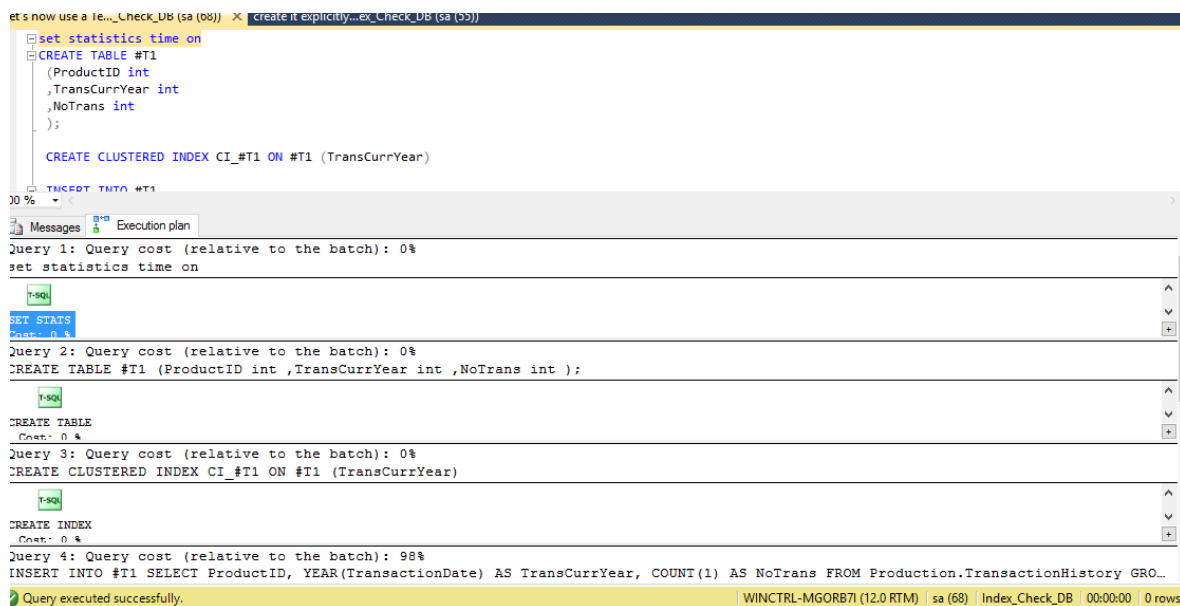


Figure 6.12 - Query cost with #temp table

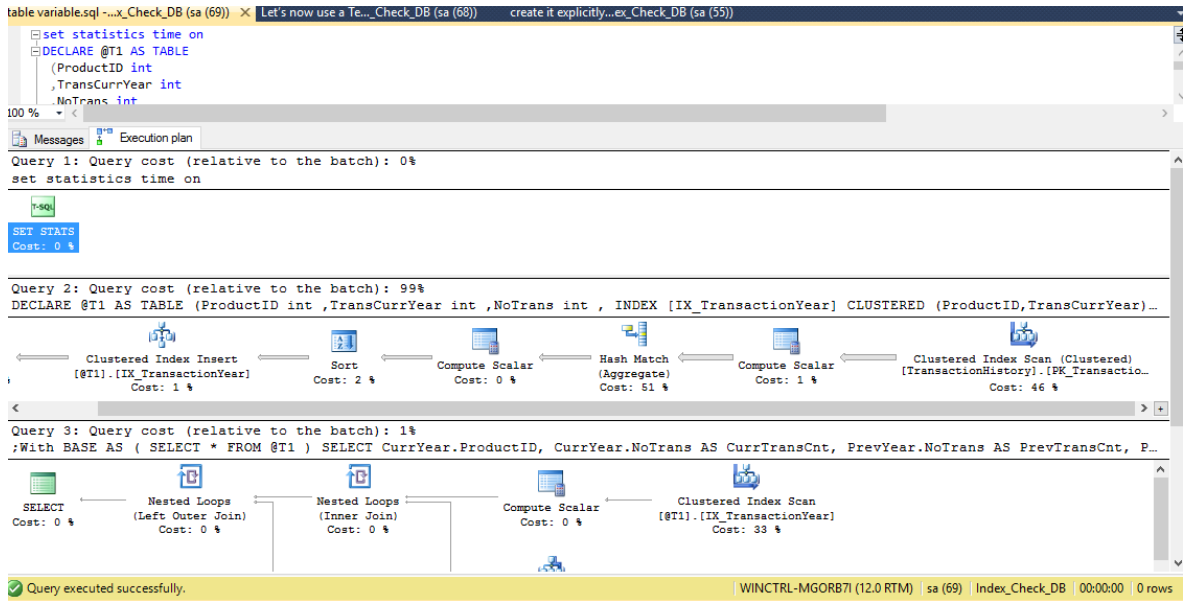
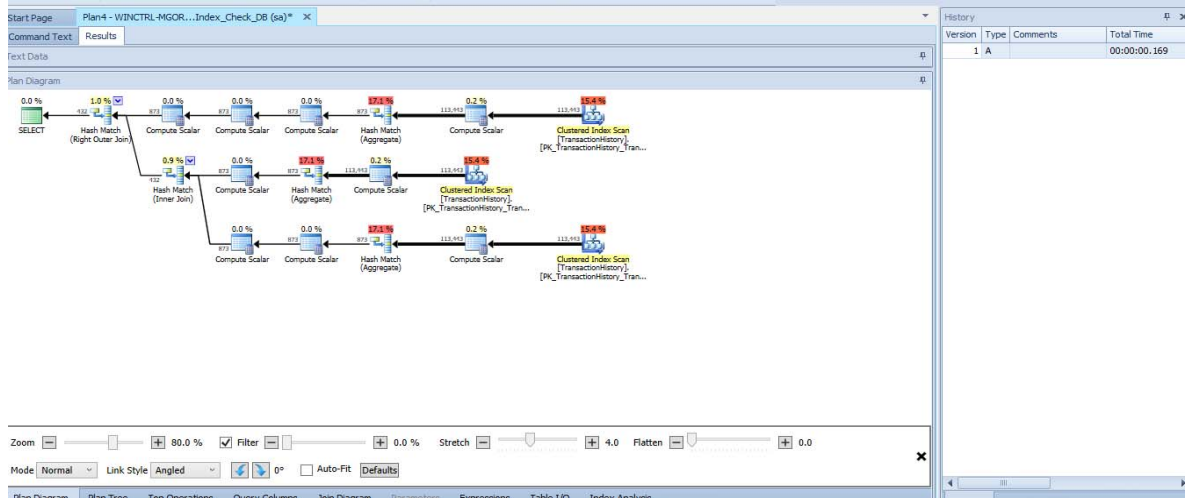
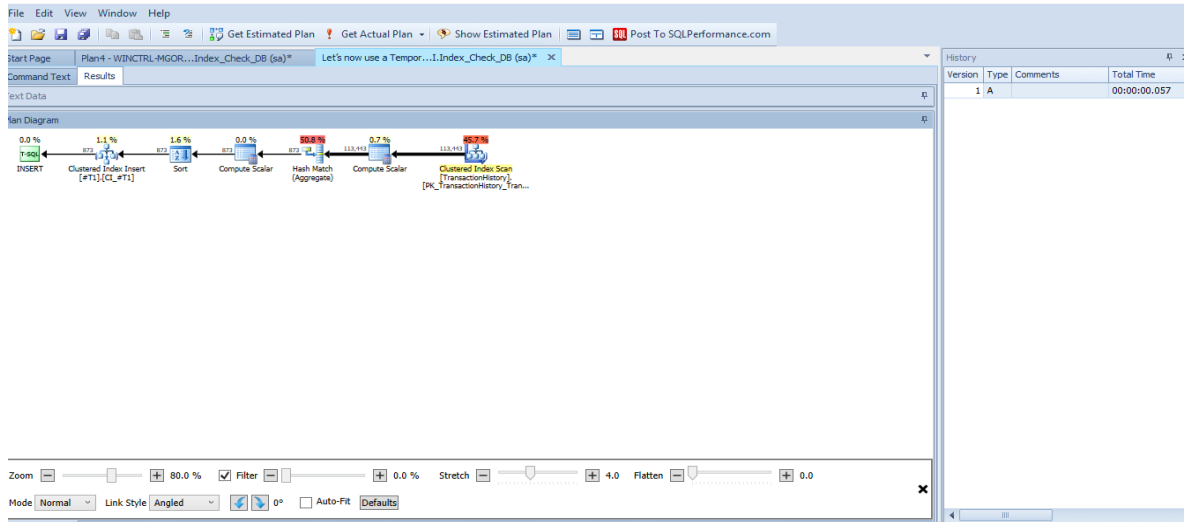


Figure 6.13 - Query cost with @temp table



6.14 - Sentry plan with #temp table



6.15 - Sentry plan with @temp table

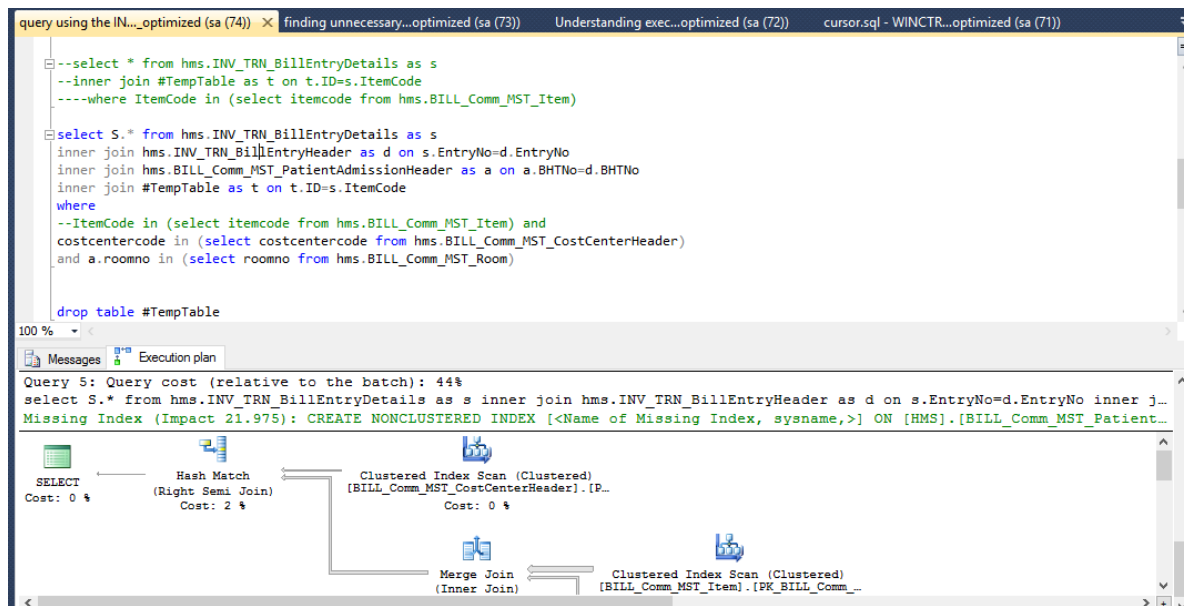


Figure 6.16 – How to find missing index

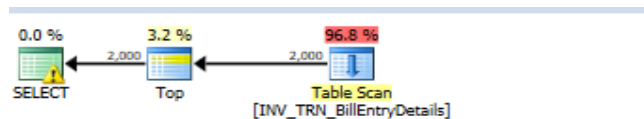


Figure 6.17 – Analyzed best practice IN and Where Clause.

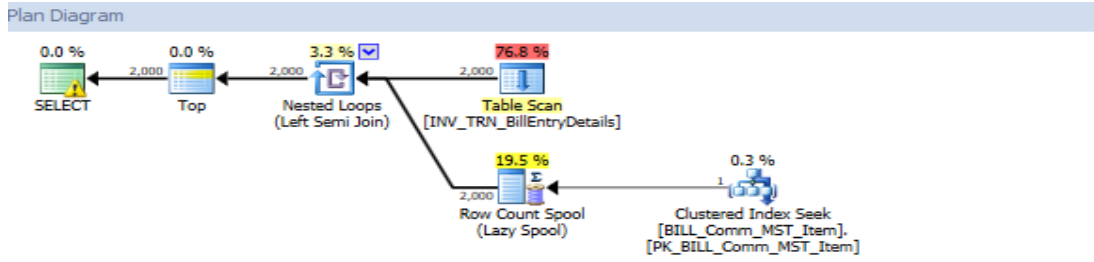


Figure 6.18 – Analyzed bad practice IN and Where Clause

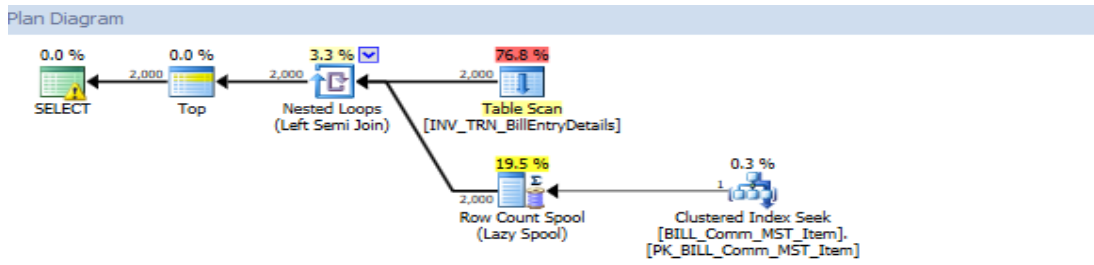


Figure 6.19 – Bad practice for IN and Where

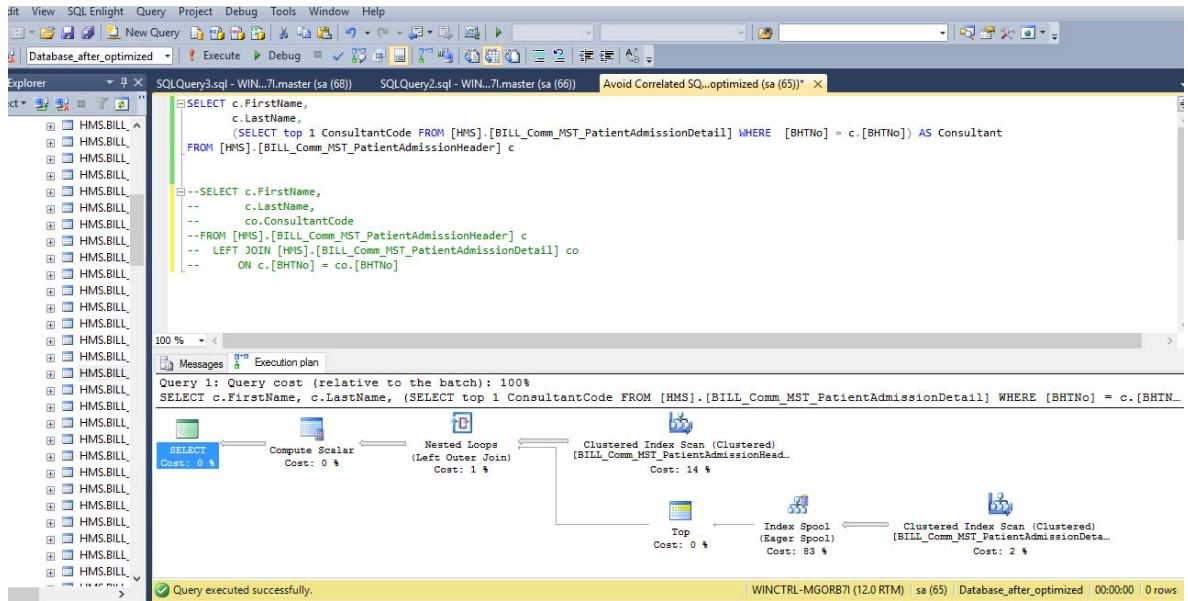


Figure 6.20 – QEP plan and Cost of Correlated SQL subqueries

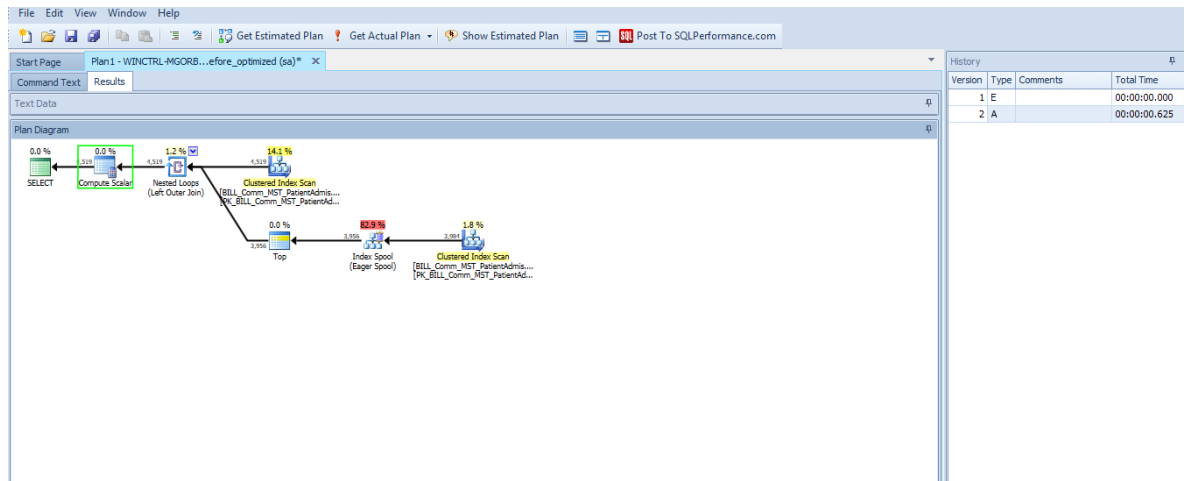


Figure 6.21- QEP plan and Cost of Correlated SQL subqueries in Sentry planner

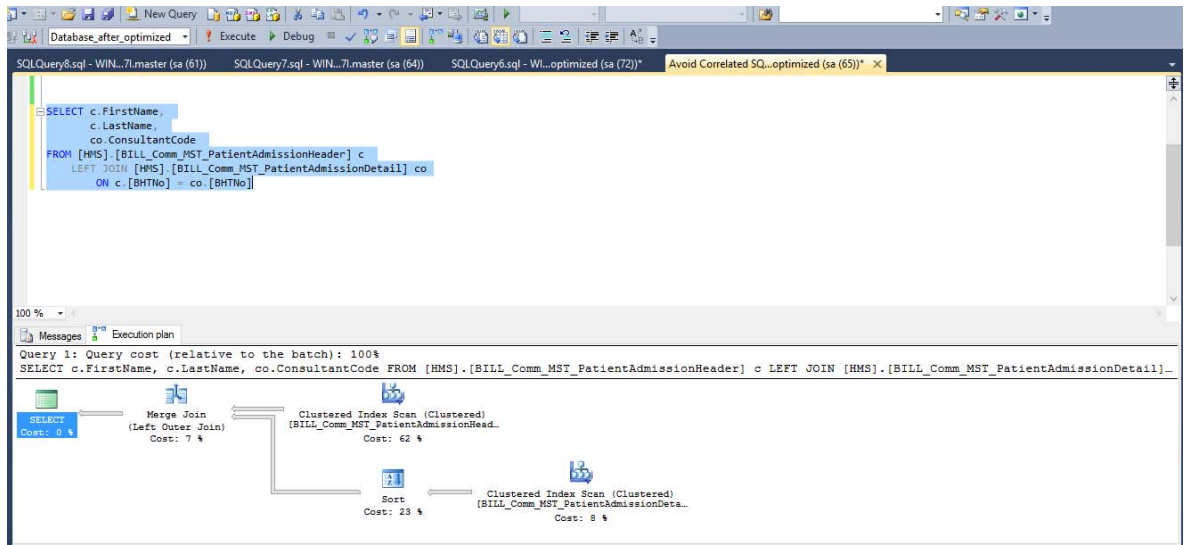


Figure 6.22- Our Query QEP plan and Cost of Correlated SQL subqueries

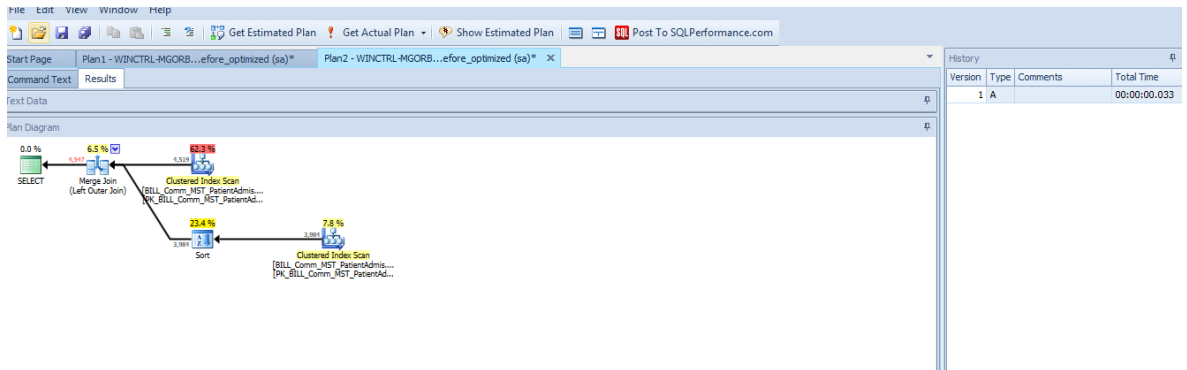


Figure 6.23- Our Query QEP plan and Cost of Correlated SQL subqueries in Sentry planner



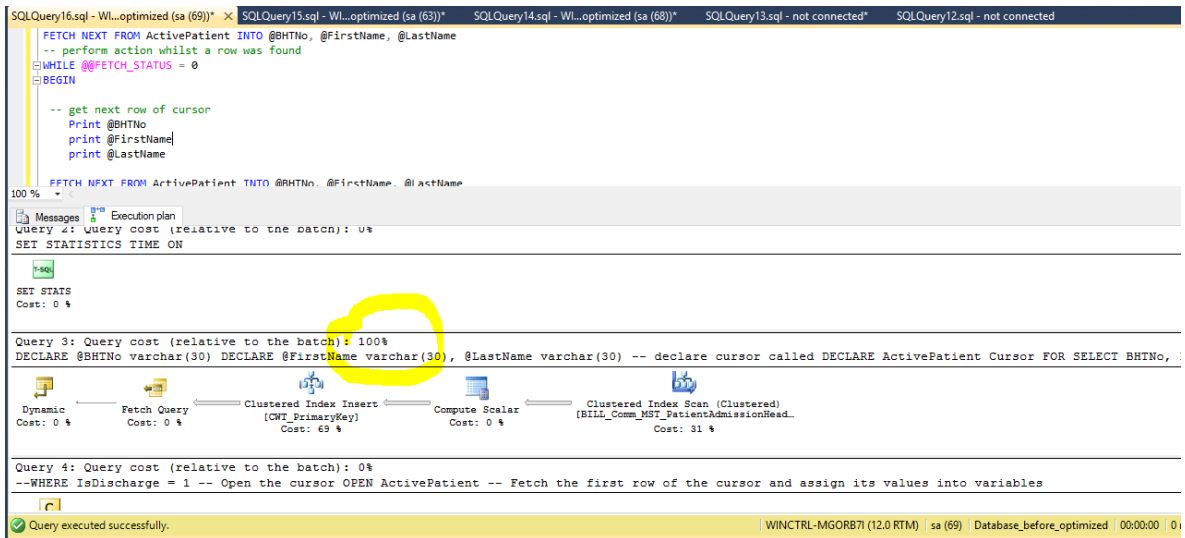


Figure 6.24 – QEP in Cursor

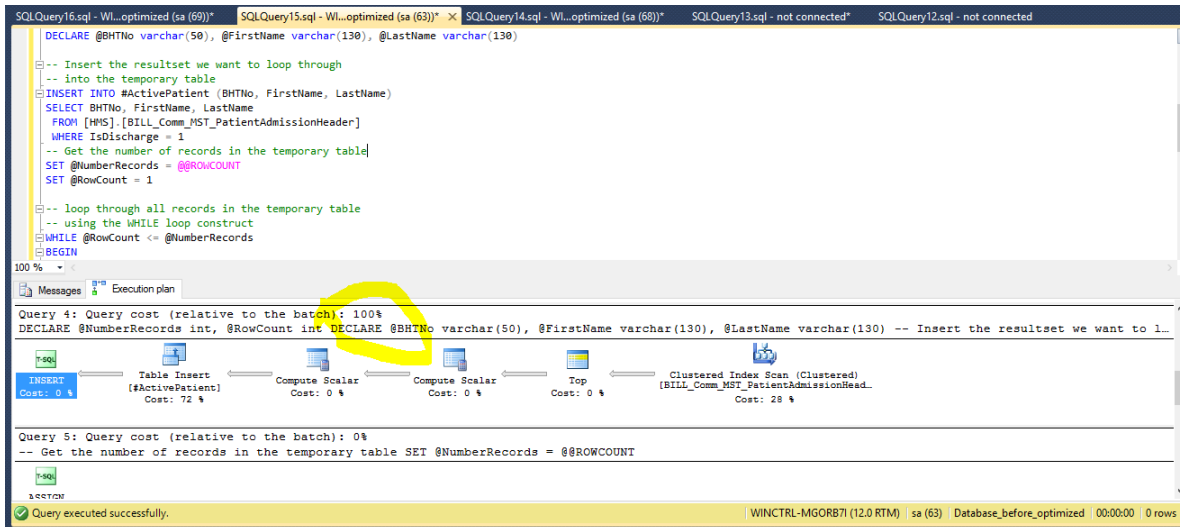


Figure 6.25 – Alternative solution on QEP plan and query cost

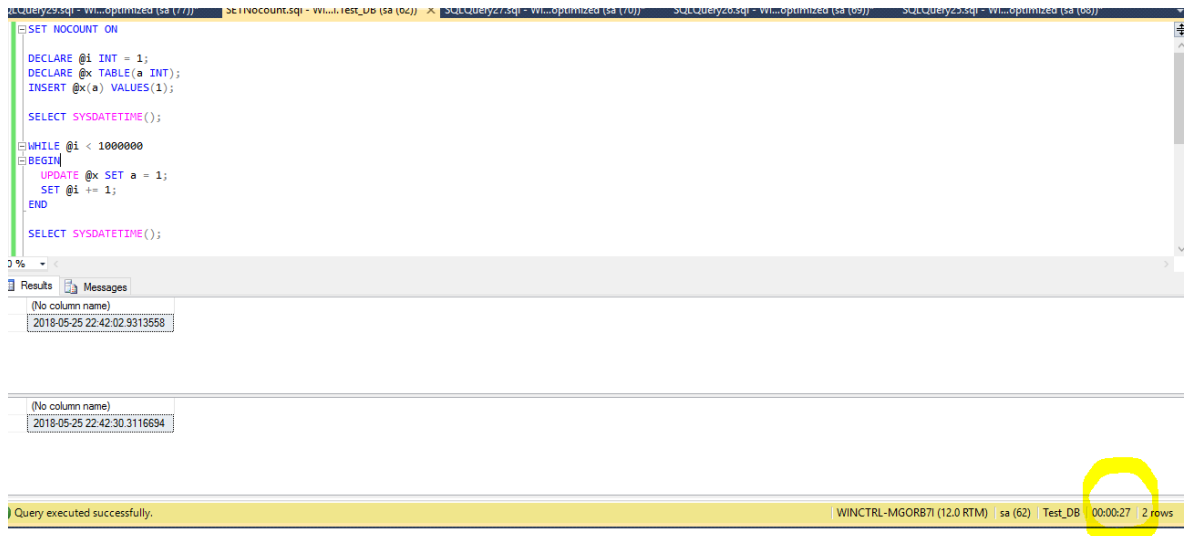


Figure 6.26 – Set no count on execution time

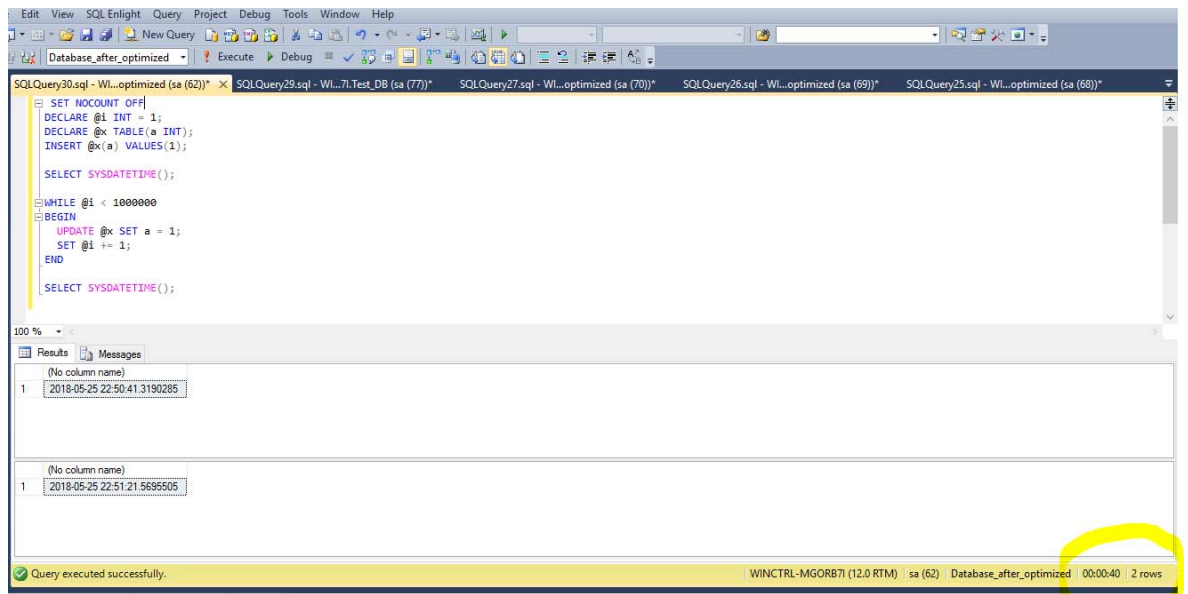


Figure 6.27 - 6.26 – Without no count execution time