Steve Stedman
Freelance SQL Server Consultant
http://stevestedman.com

SQL Server Technical Debt

As Beta 7 of the Database Health Reports approaches is just three weeks, the big new feature that has been added is the SQL Technical Debt Analysis.

The SQL Technical Debt Analysis a great check for those areas that are known problems when working on SQL Server.

But what if I like the GOTO statement?

All of the Database Health Reports checks for Technical Debt have specific over-ride options. If your coding standards state that using the GOTO statement is appropriate, then just turn that option off.

What if I use NOLOCK everywhere to improve performance?

WITH NOLOCK can be a very dangerous option if it is used incorrectly. It can however be

extremely useful if used correctly in the right situation. If you have the belief that NOLOCK should be used on every query as the "GO FASTER" option, you may want to do some research on the negative issues around using NOLOCK that way. If you have a technical standard that states when NOLOCK should be used and when it should not and you are aware of the issues, then you can just turn off the check for NOLOCK in the Database Health Reports.

Sustainable Code

Is there an section of code in your database that you know you shouldn't touch, for instance if you ever need to make changes, or fix a bug that specific area is known to be much more expensive to maintain than other areas of your system. Areas like this that are difficult to sustain and work with over time are often times the areas with the highest technical debt.

Transparency

There are some who love transparency in the process, and some who don't. Some people don't like to have their work analyzed or criticized. Technical Debt reporting allows you to establish a baseline for the quality of your T-SQL code and database design, and then determine are you getting further into debt, or is it improving.

Database Health Reports

Here is a preview of the Technical Debt analysis coming in the next beta of the Database Health Reports.

Debt By Type

Rank	Score	Debt	Type
1	433230	$1,083,075.00	Keyword should be upper case
2	62361	$155,902.00	Procedure or function too long
3	27960	$69,900.00	NOLOCK hint used
4	19756	$49,390.00	Columns in WHERE clause missing
5	9355	$23,388.00	Cursors Used
6	10	$25.00	WITH RECOMPILE should not be us
7	10	$25.00	Use of = NULL should be IS NULL

Debt By Object

Rank	Score	Debt	Object
5	11415	$28,538.00	[dbo].[BigBadMultiReturn]
6	365	$912.00	[dbo].[BadMultiReturn]
7	290	$725.00	[dbo].[Bad Sproc]
8	240	$600.00	[dbo].[Bad Table Trigger Spaces]
9	230	$575.00	[dbo].[SomeSuckierSproc0c1a]
10	230	$575.00	[dbo].[YetAnotherSuckySproc0ecba
11	230	$575.00	[dbo].[YetAnotherSuckierSproc0eda
12	230	$575.00	[dbo].[YetSomeSuckySproc0edcb1
13	230	$575.00	[dbo].[SomeSuckySproc0edca]

All Technical Debt

Score	Debt	Name	Message	Lin
13440	$33,600.00	[dbo].[BadSprocTest20K]	Column keyword not upper case in Stored Procedure (448)	htt
13440	$33,600.00	[dbo].[BadSprocTest1K]	Column keyword not upper case in Stored Procedure (448)	htt
13440	$33,600.00	[dbo].[BadSprocTest10K]	Column keyword not upper case in Stored Procedure (448)	htt
13440	$33,600.00	[dbo].[BadSprocTest30K]	Column keyword not upper case in Stored Procedure (448)	htt
9911	$24,778.00	[dbo].[BadSprocTest10K]	Length of 10011 exceeds maximum threshold	htt
4928	$12,320.00	[dbo].[BadSprocTest20K]	columns in WHERE clause missing table reference	htt
4928	$12,320.00	[dbo].[BadSprocTest1K]	columns in WHERE clause missing table reference	htt
4928	$12,320.00	[dbo].[BadSprocTest10K]	columns in WHERE clause missing table reference	htt
4928	$12,320.00	[dbo].[BadSprocTest30K]	columns in WHERE clause missing table reference	htt

See Also:

- Database Health Reports
- Technical Debt
- SQLDebt.com