Documentation

Code Analysis

Code Analysis > Code Analysis for SQL Server > Performance rules

Performance rules

Page last updated 04 February 2021

Use of SQL that could cause performance problems.

PE001 – Schema name for procedure is not specified

A stored procedure is being called via the EXECUTE command without the schema being specified.

Usually, performance is slightly better if you specify the schema, but in certain cases, you need versions of the same stored procedure to be different depending on the user role. You can put different SPs of the same name in different schemas. You then need to specify the stored procedure without the schema because SQL Server will then choose the stored procedure from the schema associated with the role of the user.

Learn more about this rule

PE002 – Schema name for table or view is not specified

It is recommended to always specify schema name.

Usually, performance is slightly better if you specify the schema because it speeds the search for the object, but in certain cases, you need versions of the same table or view to be different depending on the user role. You can put different tables or views of the same name in different schemas. You then need to specify the table without the

PE014
PE015
PE016
PE017
PE018
PE019
PE020
PE021
PE022
PE023
Script rules
Style rules

schema because SQL Server will then choose the table or view from the schema associated with the role of the user.

Learn more about this rule

PE003 – Creation of table by SELECT INTO statement

SELECT … INTO is being used to create a table. Before SQL Server 2005 this would lock out all other users.
It is usually easier to understand tables created with a CREATE TABLE statement. However, SELECT INTO can be very useful for creating temporary tables.

Learn more about this rule

PE004 – INDEX HINT is used

It is not a good idea to use query hints as in WITH (INDEX (MyIndex)).

The SQL Server query optimizer usually selects the best index for a join, so these be used only as a last resort by experienced developers and database administrators.

Learn more about this rule

PE005 – JOIN HINT is used

It is not recommended to use join hints.
It is not a good idea to use join hints as in WITH (LOOP, HASH,MERGE or REMOTE). The SQL Server query optimizer usually selects the best index for a join, so these should only be used as a last resort by experienced developers and database administrators.

Learn more about this rule

PE006 – TABLE HINT is used

It is not recommended to use table hints.
It is not a good idea to use any of the rich variety of table hints. The SQL Server query optimizer usually selects the best index for a join, so these be used only as a last resort by experienced developers and database administrators.

Learn more about this rule

PE007 – QUERY HINT is used

It is not recommended to use query hints.
It is not a good idea to use query hints. The SQL Server query optimizer usually selects the best execution plan for a query, these should be used only as a last resort by experienced developers and database administrators.

Learn more about this rule

PE008 – SET NOCOUNT OFF is used

It is not recommended to set NOCOUNT to OFF.
This is the default setting but it isn't such a good idea in a production system. The count of the result of a SQL expression is useful but it is actually sent to the application in the message queue rather than as a result. The value can be gotten using the @@ROWCOUNT function.

Learn more about this rule

PE009 – No SET NOCOUNT ON before DML

It is recommended to set NOCOUNT to ON.
It is a good idea to add a set NOCOUNT to ON statement before any DML statements in a batch to prevent the count of the result of a SQL expression being sent to the users application like a print statement in the message queue rather than as a result. The value can be gotten using the @@ROWCOUNT function.

Learn more about this rule

PE010 – Interleaving DDL and DML in stored procedure/trigger

It is not recommended to mix DML and DDL statement because recompilation issues may arise.

If DML and DDL statements are interleaved, stored procedures will often be recompiled instead of using the cached query plan, especially if there are DDL statements following DML statements. The risk of this *happening is far less if you use table variables rather than creating temporary tables or ordinary tables.*

Learn more about this rule

PE011 – PRINT statement is used in trigger

Trigger should not return data back to a client.

PRINT statements can be returned to the client. These are not in the result set but sent down the Infomessage queue, like the count messages. These are fine for temporary debugging, but should not be still there in a production system.

Learn more about this rule

PE012 – Settings lead to procedure recompilation (only in proc/trigger)

Setting this option from within batch leads to batch recompilation.

Where possible, batches, triggers and procedures should be compiled just once and their plans reused when called again. This yields better performance and uses less memory resource. Anything that changes the execution context will force a recompile.

Learn more about this rule

PE013 – COUNT used instead of EXISTS

It is not recommended to use COUNT() in this circumstance. Use EXISTS() instead for superior performance and readability.

Some programmers use COUNT(*) to check to see if there are any rows that match some criteria when EXISTS() or NOT EXISTS() could be used instead.

Learn more about this rule

PE014 – SET FORCEPLAN used

It is not recommended to use SET FORCEPLAN OFF because of the possibility of performance degradation.

SET FORCEPLAN ON overrides the logic used by the query optimizer to process a Transact-SQL SELECT statement. It prevents the optimizer from changing the order in which SQL Server processes the tables to satisfy the query. It is very unlikely to be useful.

Learn more about this rule

PE015 – No FETCH FIRST/LAST/PRIOR, but cursor is not declared as forward only

It is recommended to declare cursor with appropriate scroll options (FORWARD_ONLY,FAST_FORWARD, etc).

Learn more about this rule

PE016 – Cursor is opened but is not deallocated

It is recommended to deallocate a cursor explicitly.

Learn more about this rule

PE017 – Incorrect usage of const UDF

It is not recommended to use a UDF that returns a const value in JOIN or WHERE clauses or in a select list – it is better to store the UDF value in a variable.

Learn more about this rule

PE018 – Cursor is not declared as readonly

This cursor can be declared as readonly as there was no UPDATE or DELETE operation.

Learn more about this rule

PE019 – Consider using EXISTS instead of IN

In theory, EXISTS is faster because the search stops as soon as the condition is true, whereas IN has to collect all sub-query results before testing the condition. In practice, the query optimizer treats EXISTS and IN the same way whenever it can.

Learn more about this rule

PE020 – INSERT INTO table with ORDER BY

The SQL Query that is used to produce the result that is inserted into the table has its order specified by an ORDER BY statement. Relational tables are not ordered, so the ORDER BY is meaningless.

Learn more about this rule

PE021 – WITH RECOMPILE option is used

Consider using RECOMPILE query hint instead of WITH RECOMPILE option.

Learn more about this rule

PE022 – Foreign key is not trusted

Foreign key is not trusted.

Learn more about this rule

PE023 – DDL without specifying a schema name for table

DDL without specifying a schema name for table.

Learn more about this rule

Didn't find what you were looking for?

Visit the Redgate forum | Contact Support

Product Articles

Tips and how-to guides for Redgate products

University

Easy to follow video courses

Community Forums

Ask, discuss, and solve questions about Redgate's tools

Events & Friends

Meet us at an event, get sponsored, and join our Friends of Redgate

Simple Talk

In-depth articles and opinion from Redgate's technical journal