



Code Analysis

[Code Analysis](#) > [Code Analysis for SQL Server](#) > Execution rules

> Code Analysis for Oracle

▼ **Code Analysis for SQL Server**

> Best practice rules

> Deprecated syntax rules

▼ **Execution rules**

EI001

EI002

EI003

EI004

EI005

EI006

EI007

EI008

EI009

EI010

EI011

EI012

EI013

EI014

EI015

EI016

EI017

Execution rules

Page last updated 25 July 2018

Identify issues that may be problematic upon execution.

EI001 – Incompatible variable type for procedure call

Variable type is not fully compatible with procedure parameter type. *A parameter passed to a procedure or function must be of a type that can be cast into the variable datatype declared for that parameter in the body of the routine. It should be of exactly the same type so as to avoid the extra processing to resolve an implicit conversion.*

[Learn more about this rule](#)

EI002 – Incompatible literal type for procedure call

Literal type is not fully compatible with procedure parameter type. *A parameter passed to a procedure can be a literal (e.g. 1,'03 jun 2017' or 'hello world') but it must be possible to cast it unambiguously to the variable datatype declared for that parameter in the body of the routine.*

[Learn more about this rule](#)

EI003 – Non-scalar subquery in place of a scalar

Subquery may return more than one row.

EI018

EI019

EI020

EI021

EI022

EI023

EI024

EI025

EI026

EI027

EI028

EI029

EI030

EI031

EI032

EI033

- Miscellaneous rules
- Naming convention rules
- Performance rules
- Script rules
- Style rules

A subquery can only be scalar, meaning that it can return just one value. Even if you correctly place just one expression in your select list, you must also ensure that just one row is returned. TOP 1 can be used if there is an ORDER BY clause

[Learn more about this rule](#)

EI004 – Extra parameter passed

A named parameter is not found in parameter list of a procedure. Parameters can be passed by position in a comma-delimited list, or by name, where order position isn't required. Any parameters that are specified by name must have the name identical to the definition for that procedure

[Learn more about this rule](#)

EI005 – Unnamed call after named call

It is not recommended to use the position notation after the named notation when calling a procedure.

Parameters can be passed by position in a comma-delimited list, or by name, but it is a bad idea to mix the two methods even when it is possible. If a parameter has a default value assigned to it, it can be left out of the parameter list, and it is difficult to check whether the values you supply are for the parameters you intend.

[Learn more about this rule](#)

EI006 – Required parameter is not passed

Parameter is not passed to a procedure and no default is provided. With procedures and functions, parameters can be assigned default values that are used when a value isn't passed for that parameter when calling the procedure. However, if a parameter isn't assigned a value and there is no default provided it is an error.

[Learn more about this rule](#)

EI007 – Call parameter declared as output

Procedure parameter is not defined as OUTPUT, but marked as OUTPUT in procedure call statement.

Output scalar parameters for procedures are passed to the procedure, and can have their value altered within the procedure. This allows procedures to return scalar output. The formal parameter must be declared as an OUTPUT parameter if the actual parameter that is passed had the OUTPUT keyword. This triggers an error.

[Learn more about this rule](#)

EI008 – Call parameter is not declared as output

Procedure parameter is defined as OUTPUT, but is not marked as OUTPUT in procedure call statement.

Output scalar parameters for procedures are passed to the procedure, and can have their value altered within the procedure. This allows procedures to return scalar output. However, the matching variable passed as the output parameter in the module or command string must also have the keyword OUTPUT. There is no error but the resultant value is NULL, which you are unlikely to want.

[Learn more about this rule](#)

EI009 – Call has more parameters than required

Number of passed parameters exceeds the number of procedure parameters.

Parameters can be passed to procedures and functions in an ordered delimited list, but never more than the number of parameters. For a function, this must have the same number of list members as the parameters. For a procedure you can have fewer if defaults are declared in parameters.

[Learn more about this rule](#)

EI010 – OPEN of undefined cursor

The cursor definition is not found in the current scope – the OPEN statement will cause an error.

[Learn more about this rule](#)

EI011 – FETCH from undefined cursor

The cursor definition is not found in the current scope – the FETCH statement will cause an error.

[Learn more about this rule](#)

EI012 – CLOSE of undefined cursor

The cursor definition is not found in the current scope – the CLOSE statement will cause an error.

[Learn more about this rule](#)

EI013 – DEALLOCATE of undefined cursor

The cursor definition is not found in the current scope – the DEALLOCATE statement will cause an error.

[Learn more about this rule](#)

EI014 – Fetch from cursor with asterisk in select list, unable to check columns

It is not recommended to use the asterisk in select list when defining a cursor.

[Learn more about this rule](#)

EI015 – Incorrect number of fetch variables



Number of variables in FETCH statement differs from number of columns in the cursor definition.

[Learn more about this rule](#)

EI016 – Reference to procedure in other database

Try to avoid using hardcoded references to other databases.

There is nothing wrong in executing procedures in other databases, but it is better to avoid hard-coding these references and use synonyms

instead.

[Learn more about this rule](#)

EI017 – Hardcoded current database name in procedure call

It is better to avoid using hardcoded references to the current database as this causes problems if you later do the inconceivable by changing the database name or cut-and-pasting a routine. A hardcoded database name has no performance advantage.

[Learn more about this rule](#)

EI018 – Missing parameter(s) name in procedure call

It is recommended to specify the parameter name while calling a stored procedure.

Although delimited ordered lists are perfectly legal, it is generally wiser to specify the name of the parameter just in case you add a parameter in the middle of the list where it makes logical sense to do so. It is easy to do this with a metadata routine or SQL Prompt.

[Learn more about this rule](#)

EI019 – BEGIN TRANSACTION without ROLLBACK TRANSACTION

Take notice of using BEGIN TRANSACTION without ROLLBACK TRANSACTION.

ROLLBACK TRANSACTION rolls back a transaction to the beginning of it, or to a savepoint inside the transaction. You don't need a ROLLBACK TRANSACTION statement within a transaction, but if there isn't one, then it may be a sign that error handling has not been refined to production standards.

[Learn more about this rule](#)

EI020 – ROLLBACK TRANSACTION without BEGIN TRANSACTION

Take notice of using ROLLBACK TRANSACTION without BEGIN TRANSACTION.

It is possible to have a ROLLBACK TRANSACTION within a block where there is no explicit transaction. This will trigger an error if the code is executed outside a transaction, and suggests that transactions are being held open unnecessarily.

[Learn more about this rule](#)

EI021 – Closing of unopened cursor

It is not possible to close an unopened cursor.

[Learn more about this rule](#)

EI022 – Fetch from unopened cursor

It is not possible to fetch rows from an unopened cursor.

[Learn more about this rule](#)

EI023 – Update/delete operation on cursor, but cursor is not declared as updatable

It is recommended to specify a cursor as updatable before performing an UPDATE/DELETE operation on the cursor.

[Learn more about this rule](#)

EI024 – Stored procedure name starts with sp_

It is not recommended to start a user stored procedure name with sp_ as this prefix is used in built in stored procedures.

If you prefix a procedure name with 'sp_', the master database is searched first, which will be a performance hit. SQL Server identifiers should not have 'hungarian notation' prefixes.

[Learn more about this rule](#)

EI025 – Executing stored procedure without getting result

It is perfectly OK to execute a stored procedure without getting the result. This integer that is returned is used to pass back information about the success of the stored procedure, but the value must be assigned in code. These return codes are generally used in control-of-flow blocks within procedures to set the return code value for each possible error. To get a SQL Server error rather than a process error, you can use the @@ERROR function.

[Learn more about this rule](#)

EI026 – Reference to function in other database

It is perfectly acceptable to reference a function in another database as long as you use the standard three-part naming convention.

[Learn more about this rule](#)

EI027 – Reference to table or view in other database

It is perfectly acceptable to reference a table or view in another database as long as you use the standard three-part naming convention.

[Learn more about this rule](#)

EI028 – Adding NOT NULL column without default value

Adding a NOT NULL column without a DEFAULT value to an existing table that already contains data will fail, because SQL Server has no way of adding that column to the existing rows, as there must be a value in the column.

[Learn more about this rule](#)

EI029 – Avoid using ISNUMERIC() function

Avoid using ISNUMERIC() function because it accepts int, numeric, float and money numbers.

ISNUMERIC() is there for legacy reasons. It is answering a different

question to the one you want answered. It is better to use the LIKE command to answer the question because each numeric data type needs a different answer.

[Learn more about this rule](#)

EI030 – Usage of ORDER BY in view or single statement (inline) TVF

Views and inline Table-valued functions have no intrinsic order. The order of the result is specified only by the SELECT statement that specifies the View, iTVF or table.

If executed, this will trigger an error 'The ORDER BY clause is invalid in views, inline functions, derived tables, subqueries, and common table expressions, unless TOP or FOR XML is also specified'.

Some programmers wrongly assume that adding TOP 100 PERCENT is ok because the error disappears.

You should include the ORDER BY only in the outermost query.

[Learn more about this rule](#)

EI031 – Relying on the INSERT...EXEC statement

[Learn more about this rule](#)

EI032 – Using the xp_cmdshell system stored procedure

[Learn more about this rule](#)

EI033 – Using dynamic SQL without the EXECUTE AS clause

[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