



Emergent Software

*Three relational databases walk into a NoSQL bar. They left after 5 mins....  
They couldn't find a table! 😂*

# sp\_Develop

## Emergent Software – SQL Server Assessment

---

Successful Software & Database Projects Start Here

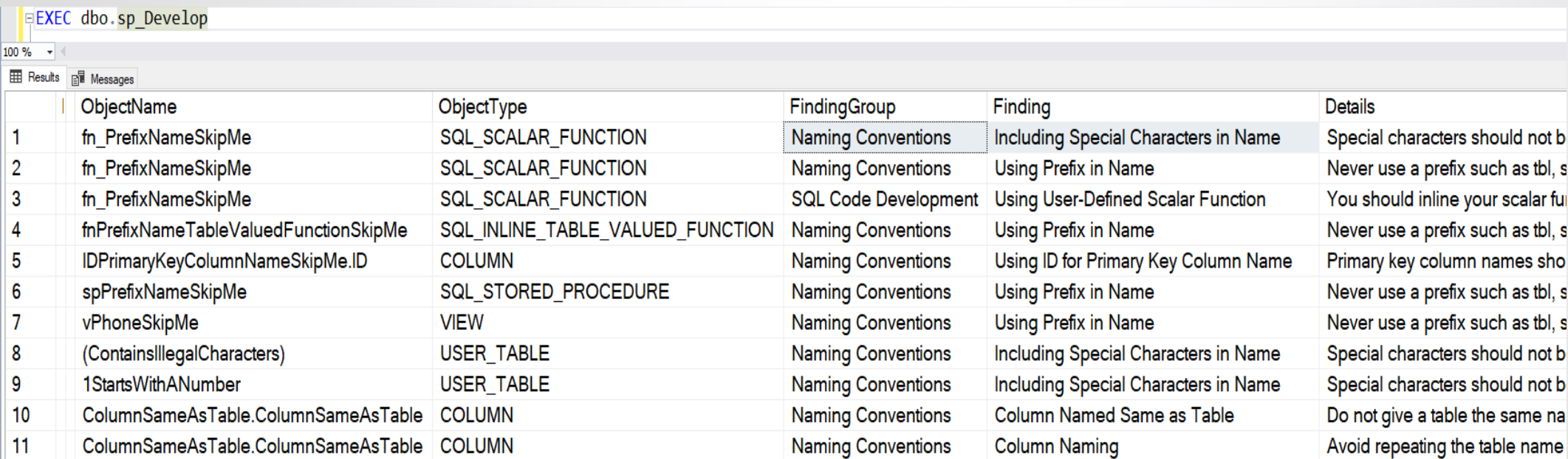
<https://github.com/EmergentSoftware/SQL-Server-Development-Assessment> - or -  
<https://emergentsoftware.github.io/SQL-Server-Development-Assessment>

Slide Deck: [https://kevinmartin.tech/sp\\_develop-session](https://kevinmartin.tech/sp_develop-session)

# What is sp\_Develop?

The open-source SQL Server Assessment project contains the sp\_Develop stored procedure. It can be used by database developers, software developers and for performing database code (smell) assessments to adhere to best practices.

sp\_Develop runs on SQL Server 2005, 2008, 2008 R2, 2012, 2014,2016, 2017, 2019, Azure SQL Server



	ObjectName	ObjectType	FindingGroup	Finding	Details
1	fn_PrefixNameSkipMe	SQL_SCALAR_FUNCTION	Naming Conventions	Including Special Characters in Name	Special characters should not b
2	fn_PrefixNameSkipMe	SQL_SCALAR_FUNCTION	Naming Conventions	Using Prefix in Name	Never use a prefix such as tbl, s
3	fn_PrefixNameSkipMe	SQL_SCALAR_FUNCTION	SQL Code Development	Using User-Defined Scalar Function	You should inline your scalar fu
4	fnPrefixNameTableValuedFunctionSkipMe	SQL_INLINE_TABLE_VALUED_FUNCTION	Naming Conventions	Using Prefix in Name	Never use a prefix such as tbl, s
5	IDPrimaryKeyColumnNameSkipMe.ID	COLUMN	Naming Conventions	Using ID for Primary Key Column Name	Primary key column names sho
6	spPrefixNameSkipMe	SQL_STORED_PROCEDURE	Naming Conventions	Using Prefix in Name	Never use a prefix such as tbl, s
7	vPhoneSkipMe	VIEW	Naming Conventions	Using Prefix in Name	Never use a prefix such as tbl, s
8	(ContainsIllegalCharacters)	USER_TABLE	Naming Conventions	Including Special Characters in Name	Special characters should not b
9	1StartsWithANumber	USER_TABLE	Naming Conventions	Including Special Characters in Name	Special characters should not b
10	ColumnSameAsTable.ColumnSameAsTable	COLUMN	Naming Conventions	Column Named Same as Table	Do not give a table the same na
11	ColumnSameAsTable.ColumnSameAsTable	COLUMN	Naming Conventions	Column Naming	Avoid repeating the table name

# About Me

---

- Hi, I'm Kevin Martin and I manage the SQL Server team at Emergent Software
- I specialize in SQL Server database development & administration
- I've done software & database development for most of my career, actively into my director level positions
- <https://www.emergentsoftware.net/contact>
- <https://kevinmartin.tech>
- <https://www.linkedin.com/in/KevinMartinTech>
- <https://twitter.com/KevinMartinTech>
- <https://github.com/KevinMartinTech>



# About Emergent Software

---

- Microsoft Gold Partner born from Emergent Networks (now Presidio)
- We develop award-winning custom software
  - websites, system integrations, mobile & web apps, ongoing maintenance & support, and much more...
- SQL Server consulting
  - modeling, query & report writing, performance tuning, data warehouse, ETL, installs, patches, migrations, best practice assessments (configuration / health, index, database development, security, usage)
  - DBA Managed Service & DBA on Vacation Service



# We Are Hiring

---

- Open Positions

- Senior SQL Server Developer & DBA
- Senior .NET Software Engineer
- Account Executive
- Senior Business Analyst, Software Development

- Why Work Here

- **Variety of Work** – you are not performing the same 1 year of work, every year
- **Flexibility** – healthy mix of office and remote for head-down time is best for our team, our clients, and us
- **Growth** – surrounded by teammates who enjoy this type of work and share their knowledge

- Hiring Process

- Pre-employment test
- Extensive vetting process with virtual machine lab challenges (better than whiteboards during interviews)
- Interview with multiple senior level experts

<https://www.emergentsoftware.net/careers/>





# Where to Find sp\_Develop

- Docs: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/>
- GitHub: <https://github.com/EmergentSoftware/SQL-Server-Development-Assessment>

The screenshot shows the 'sp\_Develop' page on the Emergent Software website. The page title is 'SQL Server Assess Overview'. The main content area contains the following text:

The open source SQL Server Assess project contains the [sp\\_Develop](#) stored procedure. It can be used by database developers, software developers and for performing database code (smell) assessments to adhere to best practices.

**NOTE:** not all checks have been developed. Browse the [Findings](#) to discover other helpful database development best practices. Use the site search to find best practices super quick!

You can use the link in the [results tab](#) to navigate to the [Findings](#) specific check details. Also, feel free to read through the sections as there might not be a check created yet to incorporate other best practices in your development.

sp\_Develop runs on SQL Server 2005, 2008, 2008 R2, 2012, 2014,2016, 2017, 2019, Azure SQL Server

Please consider [Contributing to the SQL Server Assess](#) project.

At the bottom of the page, there are two buttons: 'Get started now and install' and 'View on GitHub'.

The screenshot shows a GitHub commit history for the file `sp_Develop.sql`. The commit is by KevinMartinLink, merging pull request #120 from EmergentSoftware/dev. The commit message is 'Materializing the details of yet to be created checks into README.md' and it was made 6 days ago. The commit hash is 820d697 and it has 148 commits.

File	Commit Message	Time Ago
<code>.github</code>	added funding file to dev	15 days ago
<code>Development Application Settings</code>	Materializing the details of yet to be created checks into README.md	6 days ago
<code>Images</code>	Materializing the details of yet to be created checks into README.md	6 days ago
<code>Test Database</code>	Added table for DataType checks.	8 days ago
<code>.editorconfig</code>	added editorconfig	28 days ago
<code>.gitattributes</code>	added test database	28 days ago
<code>.gitignore</code>	changes	27 days ago
<code>CODE_OF_CONDUCT.md</code>	create community docs	27 days ago
<code>CONTRIBUTING.md</code>	change text	27 days ago
<code>LICENSE.md</code>	create community docs	27 days ago
<code>README.md</code>	Added more details	4 days ago
<code>SECURITY.md</code>	Create SECURITY.md	29 days ago
<code>sp_Develop.sql</code>	Materializing the details of yet to be created checks into README.md	6 days ago

# sp\_Develop Install Instructions

---

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/install-instructions>

sp\_Develop

Search sp\_Develop [sp\\_Develop on GitHub](#)

## sp\_Develop Install Instructions

Download a copy of the stored procedure [sp\\_Develop](#). Execute the stored procedure to create it on your SQL Server.

It is recommend installing the [sp\\_Develop](#) stored procedures in the master database for full SQL Servers, but if you want to use another one, that's totally fine.

On Azure SQL Server you will need to install the [sp\\_Develop](#) stored procedure in the user database.

[How to run sp\\_Develop](#) [View on GitHub](#)



# sp\_Develop Usage Instructions - Basic

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/usage-instructions>

sp\_Develop

Search sp\_Develop [sp\\_Develop on GitHub](#)

Home

Install Instructions

**Usage Instructions**

Results Explanations

Parameter Explanations

How to Skip Checks

Test Database Install

Development App Settings

Findings

## sp\_Develop Usage Instructions

After installing the [sp\\_Develop](#) stored procedure, open SQL Server Management Studio and run in the database you wish to check for database development best practices.

```
EXEC dbo.sp_Develop
```

That's the bare minimum you need to run the best practice checks!

Visit [Parameter Explanations for more options](#)

[What the results mean](#) [View on GitHub](#)





# sp\_Develop Results Explanations

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/results-explanations>

DatabaseName	SchemaName	ObjectName	ObjectType	FindingGroup	Finding	Details	URL	SkipCheckTSQL	Priority	CheckId	
1	spDevelop	dbo	Users	USER_TABLE	Naming Conventions	Using Plural in Names	Table and view names should be singular	<a href="https://github.com/EmergentSoftware/SQL-Server-A...">https://github.com/EmergentSoftware/SQL-Server-A...</a>	INSERT INTO DBA.DevelopCheckToSkip (ServerName, ...	20	1
2	spDevelop	dbo	vwPhones	VIEW	Naming Conventions	Using Plural in Names	Table and view names should be singular	<a href="https://github.com/EmergentSoftware/SQL-Server-A...">https://github.com/EmergentSoftware/SQL-Server-A...</a>	INSERT INTO DBA.DevelopCheckToSkip (ServerName, ...	20	1

Column Name	Details
DatabaseName	Can be run for multiple databases, this shows you the database with the potential issue
SchemaName	This is the schema for the object that might have an issue
ObjectName	This can be anything from user tables, views stored procedures, functions, ...
FindingGroup	The high-level grouping for the check (Naming Conventions, Table Conventions, Data Type Conventions, SQL Code Development, Data Issues, Configuration Issues, Running Issues)
Finding	The specific potential issue we the check is looking for
Details	Additional details about the potential issue. This does not go into in-depth details of the potential issue but should give you a heads up of what to look for
URL	Copy and paste this link into a browser to view the write up for the potential issue
SkipCheckTSQL	In this column you will find a generated TSQL script INSERT
Priority	The lower the number the more severe the potential issue is to address
CheckId	Every check is uniquely numbered

# sp\_Develop Parameters - Common

---

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/parameter-explanations>

Parameter	Details
@DatabaseName	Defaults to current DB if not specified
@GetAllDatabases	Setting = 1, runs checks across all the databases on the server instead of just your current database context. Does not work on Azure SQL Server.
@BringThePain	If you've got more than 50 databases on the server, this only works if you also pass in @BringThePain = 1, because it's gonna be slow.
@OutputType	TABLE = table COUNT = row with number found MARKDOWN = bulleted list XML = table output as XML NONE = none
@Debug	Default 0. When 1, we print out messages of what we're doing in the messages tab of SQL Server Management Studio. When 2, we print out the dynamic SQL query of the check.



# sp\_Develop Parameters - Skip Checks 1 of 4

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/how-to-skip-checks>

sp\_Develop Search sp\_Develop [sp\\_Develop on GitHub](#)

Home

Install Instructions

Usage Instructions

Results Explanations

Parameter Explanations

**How to Skip Checks**

Test Database Install

Development App Settings

Findings ▼

## How to Skip sp\_Develop Checks

Sometimes there are checks, databases or servers that you want to skip. For example, say a database is from a vendor and you are not responsible for the database development.

Another use case for skipping checks is to indicate that you have acknowledged a potential issue and you are OK with it. You can skip that check for that specific object. Using `sp_Develop` with this pattern allows you to perform your database development and iteratively check for issues.



# sp\_Develop Parameters - Skip Checks 2 of 4

---

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/how-to-skip-checks>

## Create a table to hold the list of checks you want to skip

Create this table in the database you are developing or in a database on a centralized SQL Server.

```
CREATE TABLE dbo.DevelopCheckToSkip (  
    DevelopCheckToSkipId INT IDENTITY(1, 1) NOT NULL  
    ,ServerName NVARCHAR(128) NULL  
    ,DatabaseName NVARCHAR(128) NULL  
    ,SchemaName NVARCHAR(128) NULL  
    ,ObjectName NVARCHAR(128) NULL  
    ,CheckId INT NULL  
    ,CONSTRAINT DevelopCheckToSkipId PRIMARY KEY CLUSTERED (DevelopCheckToSkipId ASC)  
);  
GO
```



# sp\_Develop Parameters - Skip Checks 3 of 4

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/how-to-skip-checks>

Results	Messages
	SkipCheckTSQL
1	INSERT INTO DBA.DevelopCheckToSkip (ServerName, DatabaseName, SchemaName, ObjectName, CheckId) VALUES (N'FAMILY\SQL2019', N'spDevelopTest', N'dbo', N'(ContainsIllegalCharacters)', 5);
2	INSERT INTO DBA.DevelopCheckToSkip (ServerName, DatabaseName, SchemaName, ObjectName, CheckId) VALUES (N'FAMILY\SQL2019', N'spDevelopTest', N'dbo', N'1StartsWithANumber', 5);
3	INSERT INTO DBA.DevelopCheckToSkip (ServerName, DatabaseName, SchemaName, ObjectName, CheckId) VALUES (N'FAMILY\SQL2019', N'spDevelopTest', N'dbo', N'ColumnSameAsTable.ColumnSame
4	INSERT INTO DBA.DevelopCheckToSkip (ServerName, DatabaseName, SchemaName, ObjectName, CheckId) VALUES (N'FAMILY\SQL2019', N'spDevelopTest', N'dbo', N'ColumnSameAsTable.ColumnSame

```
INSERT INTO
    dbo.DevelopCheckToSkip (ServerName, DatabaseName, SchemaName, ObjectName, CheckId)
VALUES
    /* Skips all checks, for every database, on the SQL2008 SQL Server */
    (N'SQL2008', NULL, NULL, NULL, NULL)
    /* Skips all checks, in the AdventureWorks2012 database, on the SQL2012 SQL Server */
    ,(N'SQL2012', N'AdventureWorks2012', NULL, NULL, NULL)
    /* Skips all checks, for the object named dbo.fn_TPSTotal, in the AdventureWorks2017 database, on
    the SQL2017 SQL Server */
    ,(N'SQL2017', N'AdventureWorks2017', N'dbo', N'fn_TPSTotal', NULL)
    /* Skips CheckId 5 (Including Special Characters in Name), for the object named dbo.[Order Details],
    in the Northwind database, on the SQL2019 SQL Server*/
    ,(N'SQL2019', N'Northwind', N'dbo', N'Order Details', 5)
    /* Skips all checks, in the AdventureWorks2017 database, on every SQL Server */
    ,(NULL, N'AdventureWorks2017', NULL, NULL, NULL)
    /* Skips all checks, for the object named dbo.vPhone, in every database, on every SQL Server */
    ,(NULL, NULL, N'dbo', N'vPhone', NULL)
    /* Skips CheckId 19 (Not Using SET NOCOUNT ON in Stored Procedure or Trigger), for the object
    dbo.CustOrderHist, in every database, on every SQL Server */
    ,(NULL, NULL, N'dbo', N'CustOrderHist', 19);
```



# sp\_Develop Parameters - Skip Checks 4 of 4

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/how-to-skip-checks>

## How to Execute the Skip Checks

```
EXEC dbo.sp_Develop
    ,@SkipCheckDatabase = N'pubs'
    ,@SkipCheckSchema = N'dbo'
    ,@SkipCheckTable = N'DevelopCheckToSkip';
```

You can also centralize this skip check table by putting it in a central location, setting up a linked server pointing to your central location, and then using the @SkipChecksServer parameter:

```
EXEC dbo.sp_Develop
    @SkipCheckServer = N'ManagementServerName'
    ,@SkipCheckDatabase = N'pubs'
    ,@SkipCheckSchema = N'dbo'
    ,@SkipCheckTable = N'DevelopCheckToSkip';
```

[Tell me about the test database](#)

[View on GitHub](#)



# sp\_Develop Test Database Install

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/test-database-install>

sp\_Develop Search sp\_Develop [sp\\_Develop on GitHub](#)

Home  
Install Instructions  
Usage Instructions  
Results Explanations  
Parameter Explanations  
How to Skip Checks  
**Test Database Install**  
Development App Settings  
Findings

## Test Database Install

The [Test Database](#) folder contains the RedGate SQL Source Control. Use this database for creating and testing checks. If you are not going to be developing [sp\\_Develop](#) checks you can skip this page.

SQL Server 2008+ is supported. You can script out the test database and downgrade schema features like `DATETIME2` that is not supported. SQL Server Developer editions are now free, go [download](#) and install the latest version for development.

### Quick Steps to Setup and Use:

- 1 Create new database 'spDevelop' and select in Object Browser
- 2 Open RedGate SQL Source Control in SQL Server Management Studio
- 3 Click 'Setup' tab
- 4 ...
- 5 **Note:** there are exclude filters setup for invalid objects created in the post script. Do not check these objects back into the branch.



# Development Application Settings Share

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/development-app-settings>

sp\_Develop

Search sp\_Develop [sp\\_Develop on GitHub](#)

- Home
- Install Instructions
- Usage Instructions
- Results Explanations
- Parameter Explanations
- How to Skip Checks
- Test Database Install
- Development App Settings**
- Findings

## Configure Development Application Settings

Included in this project are settings you can use for database development. Using the same set of settings across a team will help ensure consistent development patterns.

### SQL Server Management Studio

The settings are located in the project  
[“\SQL-Server-Assess\Development Application Settings\Microsoft\SQL Server Management Studio\General Settings”](#)

- 1 Cloned or forked the repo
- 2 In SQL Server Management Studio navigate to “Tools > Options > Environment > Import and Export Settings”
- 3 Check “Use team settings file” and browse to “..\SQL-Server-Assess\Development Application Settings\Microsoft\SQL Server Management Studio\General Settings\SSMS.vsssettings”
- 4 Click the “OK” button

### Redgate SQL Server Prompt

The settings are located in the project  
[“\SQL-Server-Assess\Development Application Settings\Red Gate\SQL Prompt”](#)

- 1 Cloned or forked the repo
- 2 Follow [these directions](#)

[Browse the check findings](#) [View on GitHub](#)



# Check Findings

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/best-practices-and-potential-findings/sql-code-conventions>

The screenshot shows a web page for 'sp\_Develop' with a search bar and a sidebar. The main content area is titled 'Findings' and contains the following text:

The sections below are where you will find the details for each of the checks performed in [sp\\_Develop](#).

Feel free to browse the finding groupings. There are best practices that do not have a check created yet.

Please consider [Contributing to the SQL Server Assess](#) project.

TABLE OF CONTENTS

- [Naming Conventions](#)
- [Table Conventions](#)
- [Data Type Conventions](#)
- [SQL Code Conventions](#)
- [Data Issue](#)
- [Configuration Issue](#)
- [Running Issue](#)
- [Current High Check Id](#)

The sidebar on the left contains the following navigation links:

- Home
- Install Instructions
- Usage Instructions
- Results Explanations
- Parameter Explanations
- How to Skip Checks
- Test Database Install
- Development App Settings
- Findings** (with an upward arrow)
- Naming Conventions
- Table Conventions
- Data Type Conventions
- SQL Code Conventions
- Data Issue
- Configuration Issue
- Running Issue
- Current High Check Id



# Check Findings – Naming Conventions

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/best-practices-and-potential-findings/naming-conventions>

sp\_Develop

## ▼ TABLE OF CONTENTS

- 1 [Using System-Generated Object Names](#)
- 2 [Concatenating Two Table Names](#)
- 3 [Variable Naming](#)
- 4 [Stored Procedures & Function Naming](#)
- 5 [Using ID for Primary Key Column Name](#)
- 6 [Not Naming Foreign Key Column the Same as Parent Table](#)
- 7 [Using Plural in Name](#)
- 8 [Using Prefix in Name](#)
- 9 [Using Prefix in Index Name](#)
- 10 [Not Using PascalCase](#)
- 11 [Using Reserved Words in Name](#)
- 12 [Including Special Characters in Name](#)
- 13 [Including Numbers in Table Name](#)
- 14 [Column Named Same as Table](#)
- 15 [Using Abbreviation](#)
- 16 [Non-Affirmative Boolean Name Use](#)
- 17 [Column Naming](#)

## Stored Procedures & Function Naming

**Check Id:** [None yet, click here to view the issue](#)

Stored procedures and functions should be named so they can be ordered by the table/business entity (ObjectAction) they perform a database operation on, and adding the database activity "Get, Update, Insert, Upsert, Delete, Merge" as a suffix, e.g., ("ProductGet" or "OrderUpdate").

[Back to top](#)

## Using ID for Primary Key Column Name

**Check Id:** 7

For columns that are the primary key for a table and uniquely identify each record in the table, the name should be [TableName] + "Id" (e.g. On the Make table, the primary key column would be "Makeld").

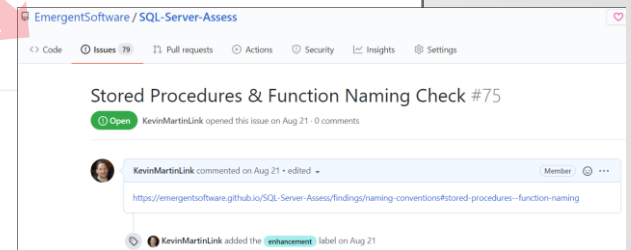
Though "Makeld" conveys no more information about the field than Make.Id and is a far wordier implementation, it is still preferable to "Id".

Naming a primary key column "Id" is also "bad" when you query from several tables you will need to rename the "Id" columns so you can distinguish them in result set.

With different column names in joins masks errors.

```
/* This has an error that is not obvious at first sight */
```

```
SELECT  
  C.Name
```





# Check Findings – Table Conventions

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/best-practices-and-potential-findings/table-conventions>

- ▼ TABLE OF CONTENTS
- 1 [Column Named ????Id But No FK Exists](#)
- 2 [More Than 5 Indexes](#)
- 3 [Less than 2 Indexes](#)
- 4 [Disabled Index](#)
- 5 [Leftover Fake Index](#)
- 6 [Column Has a Different Collation Than Database](#)
- 7 [Low Index Fill-Factor](#)
- 8 [Untrusted Foreign Key](#)
- 9 [UNIQUEIDENTIFIER in a Clustered Index](#)
- 10 [Missing Index for Foreign Key](#)
- 11 [Missing Primary Key](#)
- 12 [UNIQUEIDENTIFIER For Primary Key](#)
- 13 [Wide Table](#)
- 14 [Heap](#)

sp\_Develop  [sp\\_Develop on GitHub](#)

Home

SQL Code Conventions  
Using **NOLOCK** (READ UNCOMMITTED)

Using WITH (**NOLOCK**), WITH (READUNCOMMITTED) and TRANSACTION ISOLATION LEVEL READ UNCOMMITTED does not ...  
Can **NOLOCK** be used when the data is not changing?

on Settings



# Check Findings – Data Type Conventions

---

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/best-practices-and-potential-findings/data-type-conventions>

## ▼ TABLE OF CONTENTS

- 1 [Columns Named the Same Have Different Data Types](#)
- 2 [Using of Deprecated Data Type](#)
- 3 [Email Address Column](#)
- 4 [URL Column](#)
- 5 [Overuse of \(N\)VARCHAR\(MAX\)](#)
- 6 [Boolean Column Not Using BIT](#)
- 7 [Using FLOAT or REAL](#)
- 8 [Using SQL\\_VARIANT](#)
- 9 [Using User-Defined Data Type](#)
- 10 [Using DATETIME Instead of DATETIMEOFFSET](#)
- 11 [Using DATETIME or DATETIME2 Instead of DATE](#)
- 12 [Using DATETIME or DATETIME2 Instead of TIME](#)
- 13 [Using MONEY Data Type](#)
- 14 [Using VARCHAR Instead of NVARCHAR for Unicode Data](#)



# Check Findings – SQL Code Conventions

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/best-practices-and-potential-findings/sql-code-conventions>

## ▼ TABLE OF CONTENTS

1 [Using ORDER BY](#)

2 [Cursors](#)

a [Overview](#)

b [Valid Use Cases](#)

c [Cursor Type](#)

d [Full Cursor Syntax](#)

3 [Using WHILE Loop](#)

4 [Temporary Tables and Table Variables](#)

5 [Using Hints](#)

6 [Using Brackets](#)

7 [Using '== NULL' or '<> NULL' to Filter a Nullable Column](#)

8 [Using the NOT IN Predicate in the WHERE Clause](#)

9 [Not Using Semicolon to Terminate Statements](#)

10 [Using a Non-SARGable Expression in a WHERE Clause](#)

11 [Mixing Data Types in JOIN or WHERE Clauses](#)

12 [Stored Procedures not Using BEGIN END](#)

13 [SET ANSI\\_NULLS OFF](#)

14 [Using Types of Variable Length That Are Size 1 or 2](#)

15 [Data Type Without Length](#)

16 [COALESCE vs ISNULL](#)

17 [Using ISNUMERIC](#)

18 [Using SELECT DISTINCT](#)

19 [IN/NOT VS EXISTS/NOT EXISTS](#)

20 [Using Keyword Abbreviation](#)

21 [Using Percent at the Start of LIKE Predicate](#)

22 [Using Unfinished Notes](#)

23 [Missing Index on WHERE Clause](#)

24 [Missing Index on IN Columns](#)

25 [Converting Dates to String to Compare](#)

26 [Not Using SET XACT\\_ABORT ON](#)

27 [Scalar Function Is Not Inlineable](#)

28 [Using User-Defined Scalar Function](#)

29 [Not Using SET NOCOUNT ON in Stored Procedure or Trigger](#)

30 [Using NOLOCK \(READ UNCOMMITTED\)](#)

31 [Not Using Table Alias](#)

32 [Not Using Column List For INSERT](#)

33 [Not Using SQL Formatting](#)

34 [Not Using UPPERCASE for Keywords](#)

35 [Set Option Cause Recompile](#)

36 [Using Column Number in ORDER BY](#)

37 [Not Using Code Comments](#)

38 [Not Using Table Schema](#)

39 [Using SELECT \\*](#)

40 [Using Hardcoded Database Name Reference](#)

41 [Using @@IDENTITY Instead of SCOPE\\_IDENTITY](#)

42 [Using BETWEEN for DATETIME Ranges](#)

43 [Using Old Sybase JOIN Syntax](#)

44 [View Usage](#)

45 [Invalid Objects](#)



# Check Findings – Data Issues

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/best-practices-and-potential-findings/data-issues>

sp\_Develop

- Home
- Install Instructions
- Usage Instructions
- Results Explanations
- Parameter Explanations
- How to Skip Checks
- Test Database Install
- Development App Settings
- Findings ^
  - Naming Conventions
  - Table Conventions
  - Data Type Conventions
  - SQL Code Conventions
  - Data Issue**
  - Configuration Issue
  - Running Issue
  - Current High Check Id

▼ TABLE OF CONTENTS

- 1 [Unencrypted Data](#)

[Back to top](#)

---

## Unencrypted Data

**Check Id:** 28

The table column returned for this check might have unencrypted data that you might want to have encrypted for best practices or industry specific compliance. You will need to determine if the data needs to be protected at rest, in transit or both.

**With SQL Server you have a couple choices to implement hashing or encryption**

- [SQL Server Always Encrypt](#)
- [SQL Server Transparent Data Encryption \(TDE\)](#)
- You could develop your own or utilize a development framework pattern to implement a custom one-way hashing, hashing with salting or encryption using AES-128, AES-192, AES-256.

[Back to top](#)

---



# Check Findings – Configuration Issues

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/best-practices-and-potential-findings/configuration-issues>

Home

Install Instructions

Usage Instructions

Results Explanations

Parameter Explanations

How to Skip Checks

Test Database Install

Development App Settings

Findings ^

- Naming Conventions
- Table Conventions
- Data Type Conventions
- SQL Code Conventions
- Data Issue
- Configuration Issue**
- Running Issue
- Current High Check Id

## Object Not Owned by dbo

**Check Id:** [None yet, click here to view the issue](#)

Using dbo as the owner of all the database objects simplifies object management. dbo will always be a user in the database. If an object is owned by an account other than dbo, you must transfer ownership account needs to be deleted.

[Back to top](#)

---

## Database Compatibility Level is Lower Than the SQL Server

**Check Id:** [None yet, click here to view the issue](#)

The database compatibility level lower than the SQL Server it is running on.

There might be query optimization your are not getting running on an older database compatibility level. You might also introduce issues with a more modern database compatibility level.

[Back to top](#)





# Check Findings – Running Issues

- Visit: <https://emergentsoftware.github.io/SQL-Server-Development-Assessment/best-practices-and-potential-findings/running-issues>

sp\_Develop

Search sp\_Develop [sp\\_Develop on GitHub](#)

[Findings](#) / [Running Issues](#)

## Running Issues

These are some issues you might run into when running [sp\\_Develop](#).

▼ TABLE OF CONTENTS

- [Some Checks Skipped](#)
- [sp\\_Develop is Over 6 Months Old](#)
- [Ran on a Non-Readable Availability Group Secondary Databases](#)
- [Ran Against 50+ Databases Without @BringThePain = 1](#)

[Back to top](#)

---

### Some Checks Skipped

**Check Id:** 26

We skipped some checks that are not currently possible, relevant, or practical for the SQL Server [sp\\_Develop](#) is running against. This could be due to the SQL Server version/edition or the database compatibility level.

[Back to top](#)



# Check Code Sample

```
/******  
SELECT  
    @CheckId      = 1  
    ,@Priority     = 20  
    ,@FindingGroup = 'Naming Conventions'  
    ,@Finding      = 'Using Plural in Names'  
    ,@URLAnchor    = 'naming-conventions#using-plural-in-name';  
/******  
IF NOT EXISTS (SELECT 1 FROM #SkipCheck AS SC WHERE SC.CheckId = @CheckId AND SC.ObjectName IS NULL)  
BEGIN  
    IF @Debug IN (1, 2) RAISERROR(N'Running CheckId [%d]', 0, 1, @CheckId) WITH NOWAIT;  
  
    SET @StringToExecute = N'  
        INSERT INTO  
            #Finding (CheckId, Database_Id, DatabaseName, FindingGroup, Finding, URL, Priority, Schema_Id, SchemaName,  
                Object_Id, ObjectName, ObjectType, Details)  
  
        SELECT  
            CheckId      = ' + CAST(@CheckId AS NVARCHAR(MAX)) + N'  
            ,Database_Id = ' + CAST(@DatabaseId AS NVARCHAR(MAX)) + N'  
            ,DatabaseName = ''' + CAST(@DatabaseName AS NVARCHAR(MAX)) + N'''  
            ,FindingGroup = ''' + CAST(@FindingGroup AS NVARCHAR(MAX)) + N'''  
            ,Finding      = ''' + CAST(@Finding AS NVARCHAR(MAX)) + N'''  
            ,URL          = ''' + CAST(@URLBase + @URLAnchor AS NVARCHAR(MAX)) + N'''  
            ,Priority     = ' + CAST(@Priority AS NVARCHAR(MAX)) + N'  
            ,Schema_Id   = S.schema_id  
            ,SchemaName  = S.name  
            ,Object_Id   = O.object_id  
            ,ObjectName  = O.name  
            ,ObjectType  = O.type_desc  
            ,Details     = N'Table and view names should be singular''  
  
        FROM  
            ' + QUOTENAME(@DatabaseName) + N'.sys.objects AS O  
            INNER JOIN ' + QUOTENAME(@DatabaseName) + N'.sys.schemas AS S ON S.schema_id = O.schema_id  
  
        WHERE  
            O.type IN ('U', 'V')  
            AND RIGHT(O.name COLLATE SQL_Latin1_General_CP1_CI_AS, 1) = 'S'  
            AND RIGHT(O.name COLLATE SQL_Latin1_General_CP1_CI_AS, 2) <> 'SS'  
            AND O.NAME NOT IN ('sysdiagrams', 'database_firewall_rules');';  
  
    EXEC sys.sp_executesql @stmt = @StringToExecute;  
    IF @Debug = 2 AND @StringToExecute IS NOT NULL PRINT @StringToExecute;  
END;
```



# Testing on Multiple SQL Server Versions

- Visit: <https://my.visualstudio.com/Downloads?q=SQL%20Server%20Developer>

The screenshot displays the Redgate SQL Multi Script interface. The main window shows a SQL script being executed. The script includes a check for the existence of a procedure and its creation if it does not exist, followed by an ALTER PROCEDURE statement with various parameters.

```
28 IF OBJECT_ID('dbo.sp_Develop') IS NULL
29 BEGIN
30     EXEC dbo.sp_executesql
31         @stmt = N'CREATE PROCEDURE dbo.sp_Develop AS BEGIN SET NOCOUNT ON; END';
32 END;
33 GO
34
35 ALTER PROCEDURE dbo.sp_Develop
36     @DatabaseName NVARCHAR(128) = NULL /*Defaults to current DB if not specified'
37 , @GetAllDatabases BIT = 0
38 , @BringThePain BIT = 0
39 , @SkipCheckServer NVARCHAR(128) = NULL
40 , @SkipCheckDatabase NVARCHAR(128) = NULL
41 , @SkipCheckSchema NVARCHAR(128) = NULL
```

The 'Database Distribution List' on the right shows a list of databases to be tested, including FAMILY\SQL2019.master through FAMILY\SQL2005.master, and WIN2003STD\SQL2000.master. The 'Scripts Executed' pane on the bottom left shows that 2 scripts were executed at 12:48:23 AM against all listed databases.

The 'Results' pane on the bottom right shows a table of findings:

Server Name	Database ...	Database...	SchemaName	ObjectName	ObjectType	FindingGroup	Finding	Details	URL
FAMILY\SQ...	master	spDevelop	DBA	fn_PrefixN...	SQL_SCAL...	Naming Con...	Including S...	Special char...	https://em...
FAMILY\SQ...	master	spDevelop	DBA	fn_PrefixN...	SQL_SCAL...	Naming Con...	Using Prefix...	Never use ...	https://em...
FAMILY\SQ...	master	spDevelop	DBA	fn_PrefixN...	SQL_SCAL...	SQL Code D...	Using User...	You should i...	https://em...
FAMILY\SQ...	master	spDevelop	DBA	fnPrefixNa...	SQL_INLIN...	Naming Con...	Using Prefix...	Never use ...	https://em...
FAMILY\SQ...	master	spDevelop	DBA	IDPrimaryK...	COLUMN	Naming Con...	Using ID for...	Primary key...	https://em...
FAMILY\SQ...	master	spDevelop	DBA	spPrefixNa...	SQL_STOR...	Naming Con...	Using Prefix...	Never use ...	https://em...
FAMILY\SQ...	master	spDevelop	DBA	vPhoneSkipMe	VIEW	Naming Con...	Using Prefix...	Never use ...	https://em...
FAMILY\SQ...	master	spDevelop	dbo	(ContainsIll...	USER_TABLE	Naming Con...	Including S...	Special char...	https://em...
FAMILY\SO...	master	spDevelop	dbo	1StartsWith...	USER TABLE	Naming Con...	Including S...	Special char...	https://em...

Script executed successfully on 7 databases | 00:00:03 | 574 rows



END;

Do we have time for a few questions?

Please consider [Contributing to the SQL Server Assessment](#) project.

Slide Deck: [https://kevinmartin.tech/sp\\_develop-session](https://kevinmartin.tech/sp_develop-session)

*Alexa, do you have a boyfriend?*

*I'll leave the relations to the databases. 😄*