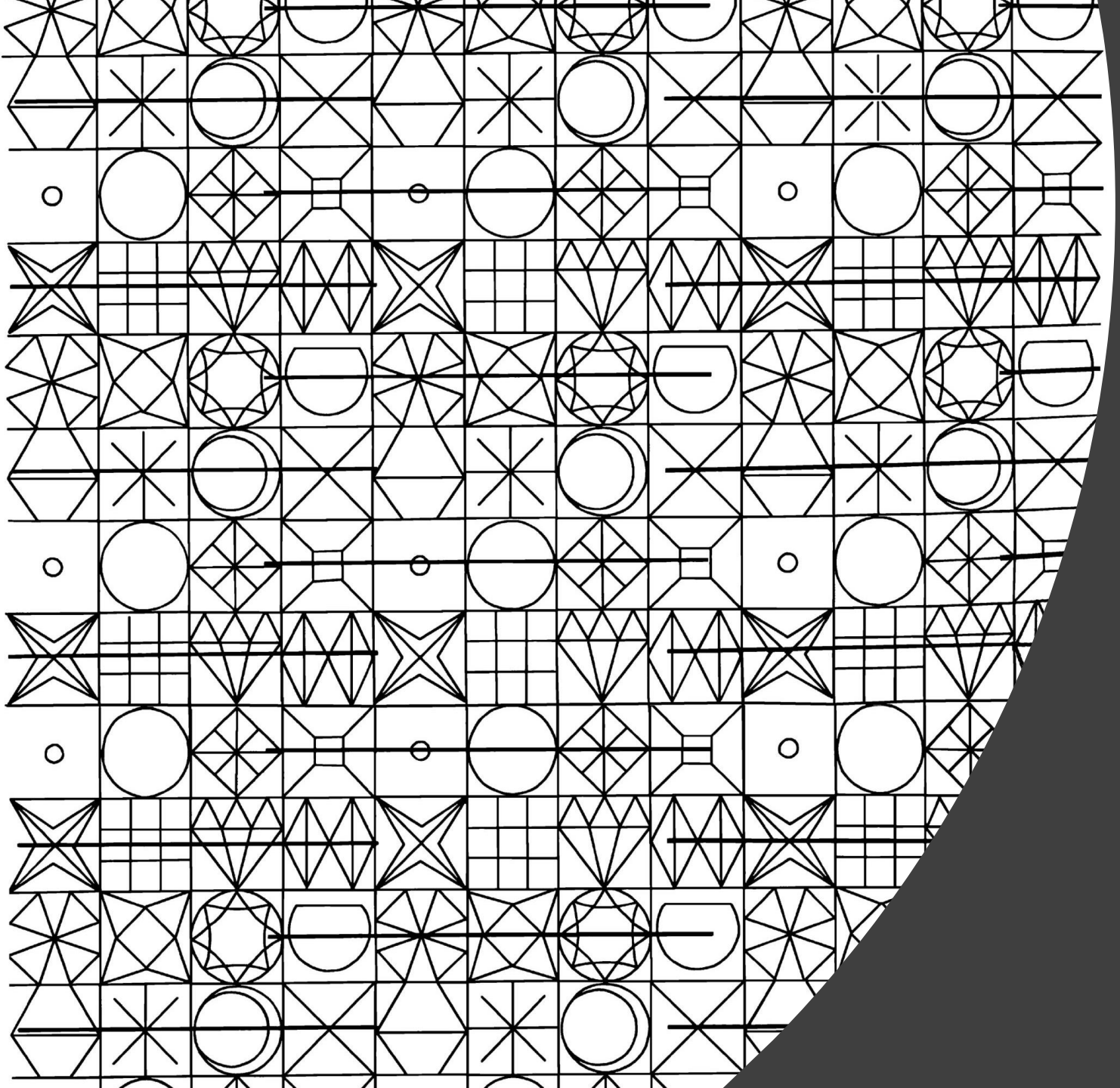


Finding TSQL Anti Patterns with ScriptDOM

About me

- SQL Server Data Engineer/DBA for 20+ years
- Co lead - Triangle SQL Server User Group, DP WIT
- Community Enthusiast

Mask up, Vaccinate, Save lives!

Takeaways

- Understand what is ScriptDOM and why it is useful
- Learn how to create simple powershell scripts using ScriptDOM to identify anti-patterns
- Learn pros and cons and comparison with similar tools.

What are ‘Anti Patterns’?

- ‘A technique intended to solve a problem but leads to other problems.’
- Practiced frequently
- Has *some* legitimate uses.

Examples

- SELECT *
- Nolock HINT
- Not aliasing tables when query uses more than one table
- Not having primary key on tables
- Using functions in where clause
- Non standard naming conventions
- Usage of cursors and linked servers more than necessary
- Non standard settings for indexes

What is ScriptDOM?

- T-SQL Script Object Model
- .NET Framework Library
- Part of DACFx package

What can it do?

Parse T-SQL scripts and find any parsing errors

Format T-SQL scripts according some general standards

Find anti-patterns

Fix *some* anti-patterns

When is this most useful?

- Large T-SQL codebase
- Legacy code
- Want to find issues before they get deployed
- Do not want any more stress on servers finding them

Why not use DMVs or Text Searches?

- Finding problems before the objects are created.
- Scientific method that is more reliable/ no false positives
- Version Compatable

Installing ScriptDOM

- Download and install sqlpackage - SQL Server | Microsoft Docs

- Separate versions for .NET core and .NET FRAMEWORK.

- All you need is C:\Program Files\Microsoft SQL Server\150\DAC\bin\Microsoft.SqlServer.TransactSql.ScriptDom.dll
- No SQL Server Install necessary

Basic Structure

- Namespace – Microsoft.sqlserver.transactsql.scriptdom
- > 1000 classes

Important classes

- TSQL170Parser…..TSQL80Parser
- TSQLFragment (Abstract Class – Parent Class for several other classes)
- TSQLFragmentVisitor (Visitor Pattern implementation)
- SQL170ScriptGenerator….SQL110ScriptGenerator
- Individual language base statement classes

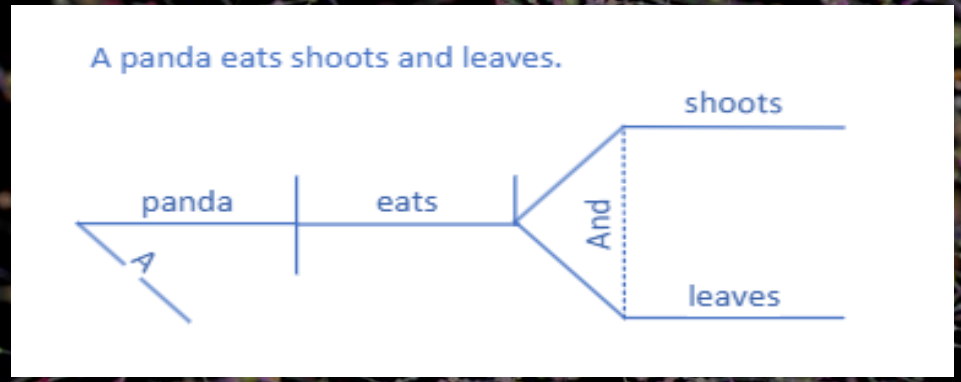
```
NamedTableReference
SelectStarExpression
```

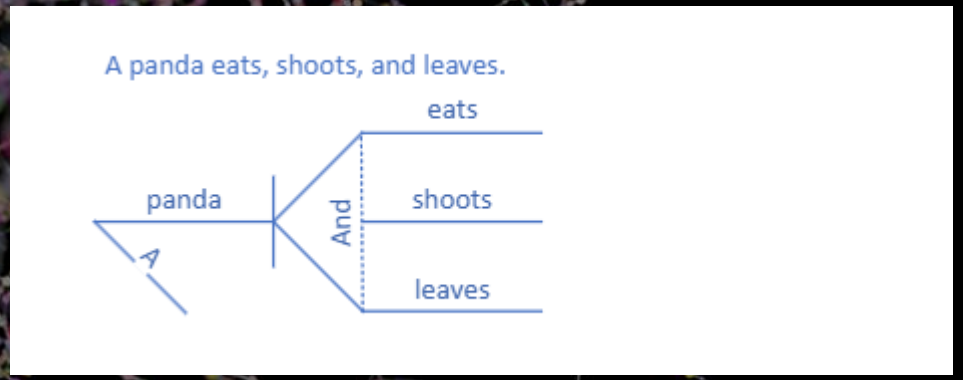
Using ScriptDOM Parser Demo

What is an ‘Abstract Syntax Tree’?

tree representation of the abstract syntactic structure of text (often source code) written in a formal language. Each node of the tree denotes a construct occurring in the text.

Why is it needed?





Exploring the ScriptDOM AST

- Basic Structure and how fragment is created
- Tokens and how they work

Using ScriptDOM to find a pattern – the long winded way

Visitor Class – What is this?

Using ScriptDOM efficiently

Using ScriptDOM as a formatter

Upvote to preserve comments
https://github.com/microsoft/DacFx/issues/41

Using ScriptDOM to fix what it finds

- Demo

Pros and Cons

- Authentic, scientific method
- Easy to deploy/maintain
- Backward compatible
- Asychronous

- Needs some .net programming
- Nuanced situations can take some programming effort
- Formatter needs to retain comments

References

- Microsoft SQL Server Script DOM | Dan Guzman's Blog (dbdelta.com)
- T-SQL Swiss Knife: Using the ScriptDom T-SQL Parser by Arvind Shyamsundar – YouTube
- Programmatically parsing Transact SQL (T-SQL) with the ScriptDom parser - Azure SQL Devs' Corner (microsoft.com)
- Finding problematic SQL NOLOCK patterns for UPDATE and DELETE (mssqltips.com)
- Episode 233: ScriptDOM (sqldatapartners.com)
- Formatting T-SQL Scripts using ScriptDOM and PowerShell – SQLServerCentral

THANK YOU!

HAPPY HOLIDAYS!