



Improve Your T-SQL  
by Changing Your Writing Habits  
– Red Gate Style

Mickey Stuewe  
Sr. SQL Database Developer



# Your Background

- DBA
- Database Developer
- Programmer
  
- Use Red Gate tools
- New to Red Gate tools



# Objectives

- Formatting T-SQL for Readability
- How SQL Prompt Can Aid You
- How to Templatize Your Favorite Scripts
- How to Add Unique Information to Generic Templates
- How to Look for Implicit Conversions
- Why Top Down Design Is Not the Best Choice for T-SQL
- Temp Tables vs Table Variables vs Materialized Tables
- How to Replace Cursors

# Formatting T-SQL for Readability

- Include a header with author, comments, and brief audit trail
- Be consistent
- The best formatting pattern is the one that is followed by the entire team



# Formatting T-SQL for Readability

- Include a header with author, comments, and brief audit trail
- Be consistent
- The best formatting pattern is the one that is followed by the entire team



# Formatting T-SQL for Readability - IMHO

- Spell words out
- Indent each sub query
- No spaces or special characters in names
- No reserved words as field or table names
- Be consistent when naming fields
- Use capitalization
- Comment, comment, comment
- Remove old code



# How SQL Prompt Can Aid You

- Standardize most of your code with THREE keys
- Find parameters that are unused
- Make SELECT \* work for you

## Quick Reference Guide:

version 6.0

<http://assets.red-gate.com/products/sql-development/sql-prompt/assets/sql-prompt-quick-reference.pdf>

version 5.0

<http://www.red-gate.com/products/sql-development/sql-prompt/assets/sql-prompt-quick-reference.pdf>



# Demo

Show me the code!





# How to Templatize Your Favorite Scripts

- Create and share a catalog of templates.
- Standardize code
- Easily access test snippets
- Easily access standard patterns you have accumulated.
- Store them in `C:\Documents and Settings\\Local Settings\Application Data\Red Gate\SQL Prompt <version>\Snippets`

# How to Add Unique Information to Generic Templates

## 1. Use Red Gate place holders (consistent information)

Prior to version 6.0

**New** in version 6.0

\$CURSOR\$

\$PASTE\$

\$DATE\$

\$SERVER\$

\$TIME\$

\$DBNAME\$

\$USER\$

\$MACHINE\$

## 2. Use SSMS place holders (dynamic information)

– <*Field Name, Data Type, Default Value*>

# Demo

Show me the code!



# How to Look for Implicit Conversions

The optimizer chooses the data type based on a predefined list.

Data Type Precedence: <http://technet.microsoft.com/en-us/library/ms190309.aspx>



# How to Look for Implicit Conversions

The optimizer chooses the data type based on a predefined list.

Data Type Precedence: <http://technet.microsoft.com/en-us/library/ms190309.aspx>

- Data types matter
- You may not get what you expect
- Implicit conversions can be costly

# Demo

Show me the code!



# Why Top Down Design Is Not the Best Choice for T-SQL

- The optimizer can't pick the best execution plan if the stored procedure is broken up into many parts
- Data is usually not narrowed quick enough
- T-SQL is built to leverage Set Theory



# Temp Tables vs Table Variables vs Materialized Tables

## Temp Tables

- Stored temporarily in temp.db

## Table Variables

- Stored temporarily in temp.db

## Materialized Tables

- Stored Permanently in database





# Temp Tables vs Table Variables vs Materialized Tables

## Temp Tables

- Stored temporarily in temp.db
- Can be indexed

## Table Variables

- Stored temporarily in temp.db
- Can be indexed

## Materialized Tables

- Stored Permanently in database
- Can be indexed



# Temp Tables vs Table Variables vs Materialized Tables

## Temp Tables

- Stored temporarily in temp.db
- Can be indexed
- Each person has their own copy

## Table Variables

- Stored temporarily in temp.db
- Can be indexed
- Each person has their own copy

## Materialized Tables

- Stored Permanently in database
- Can be indexed
- Multiple users can overwrite each others data

# Temp Tables vs Table Variables vs Materialized Tables

## Temp Tables

- Stored temporarily in temp.db
- Can be indexed
- Each person has their own copy
- Estimates are based on N rows

## Table Variables

- Stored temporarily in temp.db
- Can be indexed
- Each person has their own copy
- Estimates are based on ONE row

## Materialized Tables

- Stored Permanently in database
- Can be indexed
- Multiple users can overwrite each others data
- Estimates are based on N rows

# Temp Tables vs Table Variables vs Materialized Tables

## Temp Tables

- Good for use in stored procedures and transient data

## Table Variables

- Good for really small datasets

## Materialized Tables

- Good for single process work such as staging data to be moved to a data warehouse



# Demo

Show me the code!



# How to Replace Cursors

- Execute code for every loop
- Cursors
- While Loops
- Scalar Functions

Rows to Process	30,000
Iterations	40
Rows Processed	1,200,000

Rows to Process	30,000
1 Scalar function accessing 100 rows per function call	100
Rows Processed	3,000,000

# How to Replace Cursors

- Execute code for every loop
- Cursors
- While Loops
- Scalar Functions

## Other Options...

- Common Table Expressions (CTE)
- In-line Table Functions with CROSS APPLY

Rows to Process	30,000
Iterations	40
Rows Processed	1,200,000

Rows to Process	30,000
1 Scalar function accessing 100 rows per function call	100
Rows Processed	3,000,000

# Demo

Show me the code!





# Objectives

- Formatting T-SQL for Readability
- How SQL Prompt Can Aid You
- How to Templatize Your Favorite Scripts
- How to Add Unique Information to Generic Templates
- How to Look for Implicit Conversions
- Why Top Down Design Is Not the Best Choice for T-SQL
- Temp Tables vs Table Variables vs Materialized Tables
- How to Replace Cursors



# Questions



Please fill out an evaluation so that I can become a better instructor.

## Presentation

[www.SQLInTheCity.com](http://www.SQLInTheCity.com)

## Blogs

[MickeyStuewe.com](http://MickeyStuewe.com)

[MSSQLTips.com](http://MSSQLTips.com)

## Contact

[Mickey@MickeyStuewe.com](mailto:Mickey@MickeyStuewe.com)

Twitter: @SQLMickey

