Pro T-SQL 2008 Programmer's Guide

Michael Coles

Apress®

Pro T-SQL 2008 Programmer's Guide

ISBN-13 (pbk): 978-1-4302-1001-6

ISBN-13 (electronic): 978-1-4302-1002-3

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editors: Jonathan Gennick, Tony Campbell
Technical Reviewer: Adam Machanic
Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Kylie Johnston
Copy Editor: Damon Larson
Associate Production Director: Kari Brooks-Copony
Production Editor: Elizabeth Berry
Compositor: Lynn L'Heureux
Proofreaders: Linda Seifert, April Eddy
Indexer: Broccoli Information Management
Artist: Kinetic Publishing Services, LLC
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `http://www.springeronline.com`.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail `info@apress.com`, or visit `http://www.apress.com`.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at `http://www.apress.com/info/bulksales`.

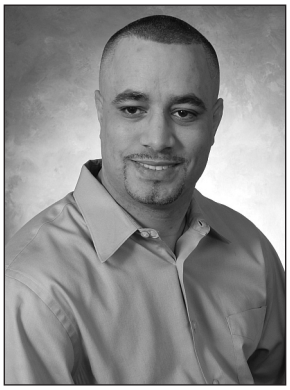
The source code for this book is available to readers at `http://www.apress.com`.

For Devoné and Rebecca

Contents at a Glance

Contents

About the Author



MICHAEL COLES is a Microsoft MVP with over a dozen years' experience in SQL database design, T-SQL development, and client-server application programming. He has consulted in a wide range of industries, including the insurance, financial, retail, and manufacturing sectors, among others. Michael's specialty is developing and performance-tuning high-profile SQL Server–based database solutions. He currently works as a consultant for a business intelligence consulting firm. He holds a degree in information technology and multiple Microsoft and other certifications.

Michael has published dozens of technical articles online and in print magazines, including SQL Server Central, ASPToday, and *SQL Server Standard*. Michael is the author of the books *Pro T-SQL 2005 Programmer's Guide* (Apress, 2007) and *Pro SQL Server 2008 XML* (Apress, 2008), and he contributed to *Accelerated SQL Server 2008* (Apress, 2008). His current projects include coauthoring the book *Pro SQL Server 2008 Full-Text Search*.

About the Technical Reviewer

ADAM MACHANIC is a Boston-based independent database consultant, writer, and speaker. He has been involved in dozens of SQL Server implementations for both high-availability OLTP and large-scale data warehouse applications, and has optimized data access layer performance for several data-intensive applications. Adam has written for numerous web sites and magazines, including SQLblog, Simple Talk, Search SQL Server, SQL Server Professional, CoDe, and VSJ. He has also contributed to several books on SQL Server, including *Expert SQL Server 2005 Development* (Apress, 2007) and *Inside SQL Server 2005: Query Tuning and Optimization* (Microsoft Press, 2007). Adam regularly speaks at user groups, community events, and conferences on a variety of SQL Server and .NET-related topics. He is a Microsoft Most Valuable Professional (MVP) for SQL Server and a Microsoft Certified IT Professional (MCITP).

Acknowledgments

I've said it before, and I'll say it again—delivering books like this into your hands takes the coordinated efforts of dozens of people working toward a common goal. There's no way you would be reading these words right now if not for the entire team at Apress. This book is the product of the work of all my Apress teammates.

With that in mind, I would like to start by thanking my editors Tony Campbell and Jonathan Gennick, who pulled this project together and oversaw it from the first page of the dedication to the last page of the index. I would also like to thank the hardest-working project manager ever, Kylie Johnston, who kept everyone on track and on schedule. I want to send a special thank you to my technical reviewer (and one of my favorite SQL authors), Adam Machanic, for keeping me honest and challenging me to "explain, explain, explain!" at every opportunity. I would also like to thank Elizabeth Berry, Damon Larson, Linda Seifert, and April Eddy, the team members who ensured that I expressed myself as clearly and cleanly as possible during the copy edit and proofreading phases. I also want to thank the numerous other team members who spent countless hours laying out pages, manipulating images, and contributing their skills in a variety of ways to bring this book to you.

I would like to thank my family, including my girlfriend, Donna; my mom; Eric; Jennifer; Chris; Desmond; and Deja. I'd also like to thank my aunt Linda and her family for their support.

Most important, thank you to Devoné and Rebecca—my little angels—for keeping a smile on my face.

Finally, I would like to send a special thank you to Steve Jones, Microsoft MVP and entrepreneur extraordinaire of SQL Server Central fame, and Chuck Heinzelman, also an MVP and editor-in-chief of *SQL Server Standard* magazine, for their continued support.

Introduction

I still recall the first "database" application I ever wrote. It was a Turbo Pascal–based application for state government, designed to keep an inventory of tools and hazardous waste materials for a state institution in the late 1980s. I recall running into a lot of issues, including performance, large data storage, extensibility, and data integrity. I mention this only because these are just the types of problems that modern enterprise-class SQL DBMSs are specifically designed to handle. What's more, they abstract away the internal workings (well, most of them anyway) so that you can concentrate more on your data and less on writing code to manipulate it. As an example, a simple sort algorithm that consumed over 100 lines of code in my custom Turbo Pascal "database" application is whittled down to a single `ORDER BY` clause in SQL.

This abstraction allows you to spend less time worrying about *how things get done* and more time thinking about *what you want to get done*. Although I don't use the term in everyday conversation, I can say that this change in thinking about storage represents a true "paradigm shift." The new version of SQL Server builds on the foundation laid out by previous releases, adding new capabilities and functionality designed to meet the increasing demands of a sophisticated developer base.

This book was originally scheduled to be an update of my *Pro T-SQL 2005 Programmer's Guide* book. The sheer number of new features, however, demanded a nearly complete rewrite. I designed this new book with the goal of helping T-SQL developers get the absolute most out of the exciting new development features and functionality in SQL Server 2008.

Who This Book Is For

This book is intended for SQL Server developers who need to port code from prior versions of SQL Server, and those who want to get the most out of T-SQL on the 2008 release. You should have a working knowledge of SQL, preferably T-SQL on SQL Server 2005 or 2000, as most of the examples in this book are written in T-SQL. In this book, I will cover some of the basics of T-SQL, including some introductory concepts like data domain and three-valued logic—but this is not a beginner's book. I will not be discussing database design, database architecture, normalization, and the most basic of SQL constructs in any kind of detail. Instead I will be focusing most of my discussion on topics of new SQL Server 2008 functionality, which assumes a basic understanding of SQL statements like `INSERT` and `SELECT`.

A working knowledge of C# and the .NET Framework is also useful (but not required), as some examples in the book will be written in C#. When C# sample code is provided, it is explained in detail, so an in-depth knowledge of the .NET Framework class library is not required.

How This Book Is Structured

This book was written to address the needs of three types of readers:

- SQL developers who are coming from other platforms to SQL Server 2008
- SQL developers who are moving from prior versions of SQL Server to SQL Server 2008
- DBAs and nondevelopers who need a working knowledge of T-SQL functionality to effectively support SQL Server 2008 instances

For all types of readers, this book is designed to act as a tutorial that describes and demonstrates new T-SQL features with working examples, and as a reference for quickly locating details about specific features. The following sections provide a chapter-by-chapter overview.

Chapter 1

Chapter 1 starts this book off by putting SQL Server 2008's implementation of T-SQL in context, including a short history of T-SQL, a discussion of T-SQL basics, and an overview of T-SQL coding best practices.

Chapter 2

Chapter 2 dives right into the new features of T-SQL on SQL Server 2008, with a discussion of productivity-enhancing features, the new `MERGE` statement, new data types like `geometry` and `hierarchyid`, and grouping sets.

Chapter 3

Chapter 3 gives an overview of the newest generation of tools available to SQL Server developers. Tools discussed include SQL Server Management Studio (SSMS), SQLCMD, Business Intelligence Development Studio (BIDS), and SQL Profiler, among others.

Chapter 4

Chapter 4 introduces T-SQL procedural code, including control-of-flow statements like `IF...THEN` and `WHILE`. In this chapter, I also discuss `CASE` expressions and `CASE`-derived functions, and provide an in-depth discussion of SQL three-valued logic.

Chapter 5

Chapter 5 discusses the various types of T-SQL user-defined functions available to encapsulate T-SQL logic on the server. I talk about all forms of T-SQL–based user-defined functions, including scalar user-defined functions, inline table-valued functions, and multistatement table-valued functions.

Chapter 6

Chapter 6 covers stored procedures, which allow you to create server-side T-SQL subroutines. In addition to describing how to create and execute stored procedures on SQL Server, I also address a thorny issue for some—the issue of why you might want to use stored procedures.

Chapter 7

Chapter 7 introduces all three types of SQL Server triggers: classic DML triggers, which fire in response to DML statements; DDL triggers, which fire in response to server and database DDL events; and logon triggers, which fire in response to server `LOGON` events.

Chapter 8

Chapter 8 discusses SQL Server encryption functionality, including the column-level encryption functionality introduced in SQL Server 2005 and the new transparent database encryption (TDE) and extensible key management (EKM) functionality, both introduced in SQL Server 2008.

Chapter 9

Chapter 9 dives into the details of common table expressions (CTEs) and windowing functions in SQL Server 2008, which feature the `OVER` clause.

Chapter 10

Chapter 10 discusses the advancements made to SQL Server 2008 integrated full-text search (iFTS), including greater integration with the SQL Server query engine and greater transparency by way of new iFTS-specific data management views and functions.

Chapter 11

Chapter 11 provides an in-depth discussion of SQL Server 2008 XML functionality, which carries forward the new features introduced in SQL Server 2005 and improves upon them. I cover several XML-related topics in this chapter, including the `xml` data type and its built-in methods, the `FOR XML` clause, and XML indexes.

Chapter 12

Chapter 12 discusses XQuery and XPath support in SQL Server 2008. SQL Server 2008 improves on the XQuery support introduced in SQL Server 2005, including support for the `xml` data type in XML DML `insert` statements and the `let` clause in FLWOR expressions.

Chapter 13

Chapter 13 introduces SQL Server 2008 catalog views, which are the preferred tools for retrieving database and database object metadata. This chapter also discusses dynamic management views and functions, which provide access to server and database state information.

Chapter 14

Chapter 14 is a discussion of SQL Common Language Runtime (SQL CLR) functionality in SQL Server 2008. In this chapter, I discuss and provide examples of SQL CLR stored procedures, user-defined functions, user-defined types, and user-defined aggregates. I also talk about the restrictions that have been removed in SQL CLR support in SQL Server 2008.

Chapter 15

Chapter 15 focuses on client-side support for SQL Server, including ADO.NET-based connectivity and one of Microsoft's newest technology offerings, LINQ to SQL.

Chapter 16

Chapter 16 discusses SQL Server connectivity using middle-tier technologies. Since native HTTP endpoints are deprecated in SQL Server 2008, I discuss them as items that may need to be supported in existing databases but should not be used for new development. I focus instead on possible replacement technologies, such as ADO.NET Data Services and IIS/.NET Web Services.

Chapter 17

Chapter 17 switches the focus back to T-SQL with a discussion of additional SQL Server 2008 features that were carried forward from their initial introduction in SQL Server 2005; features like the `INTERSECT` and `EXCEPT` operators, the DML statement `OUTPUT` clause, and improvements to the `TOP` clause. I also discuss some additional features and functionality that are new in SQL Server 2008, like new date and time functions and `FILESTREAM` support.

Chapter 18

Chapter 18 discusses improvements to server-side error handling made possible with the `TRY...CATCH` block. I also discuss various methods for debugging code, including using the Visual Studio T-SQL debugger. This chapter wraps up with a discussion of dynamic SQL and SQL injection, including the causes of SQL injection and methods you can use to protect your code against this type of attack.

Chapter 19

Chapter 19 provides an overview of performance-tuning SQL Server code. This chapter discusses SQL Server storage, indexing mechanisms, and query plans. I wrap up the chapter with a discussion of my own personal methodology for troubleshooting T-SQL performance issues.

Appendix A

Appendix A provides the answers to the exercise questions that I've included at the end of each chapter.

Appendix B

Appendix B is designed as a quick reference to the XQuery Data Model (XDM) type system.

Appendix C

Appendix C provides a quick reference glossary to several terms, many of which may be new to those using SQL Server for the first time.

Appendix D

Appendix D is a quick reference to the SQLCMD command-line tool, which allows you to execute ad hoc T-SQL statements and batches interactively, or run script files.

Conventions

To help make reading this book a more enjoyable experience, and to help you get as much out of it as possible, I've used the following standardized formatting conventions throughout.

C# code is shown in code font. Note that C# code is case sensitive. Here's an example:

```
while (i < 10)
```

T-SQL source code is also shown in code font, with keywords capitalized. Note that I've lowercased the data types in the T-SQL code to help improve readability. Here's an example:

```
DECLARE @x xml;
```

XML code is shown in code font with attribute and element content in bold for readability. Some code samples and results have been reformatted in the book for easier reading. XML ignores whitespace, so the significant content of the XML has not been altered. Here's an example:

```
<book publisher = "Apress">Pro SQL Server 2008 XML</book>:
```

Note Notes, tips, and warnings are displayed like this, in a special font with solid bars placed over and under the content.

SIDEBARS

Sidebars include additional information relevant to the current discussion and other interesting facts. Sidebars are shown on a gray background.

Prerequisites

This book requires an installation of SQL Server 2008 to run the T-SQL sample code provided. Note that the code in this book has been specifically designed to take advantage of new SQL Server 2008 features, and most of the code samples will not run on prior versions of SQL Server. The code samples presented in the book are designed to be run against the AdventureWorks 2008 sample database, available from the CodePlex web site at `www.codeplex.com/MSFTDBProdSamples`.

If you are interested in compiling and deploying the .NET code samples (the client code and SQL CLR examples) presented in the book, I highly recommend an installation of Visual Studio 2008. Although you can compile and deploy .NET code from the command line, I've provided instructions for doing so through the Visual Studio Integrated Development Environment (IDE). I find that the IDE provides a much more enjoyable experience.

Some examples, such as the ADO.NET Data Services examples in Chapter 16, require an installation of IIS as well. Other code samples presented in the book may have specific requirements, such as the LINQ samples, which require the .NET Framework 3.5. I've added notes to code samples that have additional requirements like these.

Downloading the Code

The sample code for this book is available in a ZIP file in the Downloads section of the Apress web site at `www.apress.com`. The ZIP file is structured so that each subdirectory contains all the sample code for its corresponding chapter.

Contacting the Author

The Apress team and I have made every effort to ensure that this book is free from errors and defects. Unfortunately, the occasional error does slip past us, despite our best efforts. In the event that you find an error in the book, please let us know! You can submit errors to Apress by visiting `www.apress.com`, locating the book page for this book, and clicking Submit Errata. Alternatively, feel free to drop a line directly to me at `michaelco@optonline.net`.