Pro Full-Text Search in SQL Server 2008

Michael Coles with
Hilary Cotter

Apress®

Pro Full-Text Search in SQL Server 2008

ISBN-13 (pbk): 978-1-4302-1594-3

ISBN-13 (electronic): 978-1-4302-1595-0

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

Lead Editor: Jonathan Gennick
Technical Reviewer: Steve Jones
Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Denise Santoro Lincoln
Copy Editor: Benjamin Berg
Associate Production Director: Kari Brooks-Copony
Production Editor: Laura Esterman
Compositor/Artist: Octal Publishing, Inc.
Proofreader: Patrick Vincent
Indexer: Broccoli Information Management
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 information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

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

For Devoné and Rebecca
—Michael

Contents at a Glance

Contents

About the Authors

MICHAEL COLES is a Microsoft MVP with nearly 15 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 SQL Server 2008 XML* (Apress, 2008) and *Pro T-SQL 2008 Programmer's Guide* (Apress, 2008), and he is a contributor to *Accelerated SQL Server 2008* (Apress, 2008). His current projects include speaking engagements and researching new SQL Server 2008 encryption and security functionality.

HILARY COTTER is a SQL Server MVP with more than 20 years' IT experience working for Fortune 500 clients. He graduated from University of Toronto in applied science and engineering. He is the author of a book on SQL Server replication and has written numerous white papers and articles on SQL Server and databases.

About the Technical Reviewer

STEVE JONES, a Microsoft MVP, is the founder and editor of SQLServerCentral, the largest SQL Server community on the Internet. He has been working with SQL Server since 1991 and has published numerous books and articles on all aspects of the platform. He lives in Denver with his wife, three kids, three dogs, three horses, and lots of chores.

Acknowledgments

There are several people without whom this book would not be a reality. We'd like to start by thanking our editor, Jonathan Gennick. Thanks to Steve Jones, our technical reviewer and fellow MVP, for keeping us honest. Thank you to project manager Denise Santoro Lincoln for managing this project and keeping the lines of communication open between the team members. Also thanks to Sofia Marchant for assisting with project management. We'd also like to thank Benjamin Berg and Laura Esterman for making this book print-ready.

Special thanks go to Roman Ivantsov, inventor of the Irony.NET compiler construction kit, for assisting us in the development of the Irony.NET code sample. And special thanks also to Jonathan de Halleux, creator of the .NET ternary search tree code that's the basis for our spelling suggestion code samples.

We'd also like to thank the good folks at Microsoft who provided answers to all our questions and additional guidance: Alison Brooks, Arun Krishnamoorthy, Denis Churin, Fernando Azpeitia Lopez, Jacky Chen, Jingwei Lu, Josh Teitelbaum, Margi Showman, Ramanathan Somasundaram, Somakala Jagannathan, and Venkatraman Parameswaran.

Michael Coles would also like to thank Gayle and Eric Richardson; Donna Meehan; Chris, Jennifer, Desmond, and Deja Coles; Linda Sadr and family; Rob and Laura Whitlock and family; Vitaliy Vorona; and Igor Yeliseyev. Most of all, I would like to thank my little angels, Devoné and Rebecca.

Introduction

Begin at the beginning and go on till you come to the end . . .

—Alice in Wonderland

Linguistic (language-based) searching has long been a staple of web search engines such as Google and high-end document management systems. Many developers have created custom utilities and third-party applications that implement complex search functionality similar to that provided by the most popular search engines. What many people don't realize immediately is that SQL Server provides this advanced linguistic search capability out-of-the-box. Full-Text Search (FTS) has been included with SQL Server since the SQL Server 7 release. FTS allows you to perform linguistic searches of documents and text content stored in SQL Server databases using standard T-SQL queries. FTS is a powerful tool that can be used to implement enterprise-class linguistic database searches.

SQL Server 2008 increases the power of FTS by adding a variety of new features that make it easier than ever to administer, troubleshoot, and generally use SQL Server's built-in linguistic search functionality in your own applications. In this book, we'll provide an in-depth tour of SQL Server 2008's FTS features and functionality, from both the server and client perspective.

Who This Book Is For

This book is intended for SQL Server developers and DBAs who want to get the most out of SQL Server 2008 Integrated Full-Text Search (iFTS). To get the most out of this book, you should have a working knowledge of T-SQL, as most of the sample code in the book is written in SQL Server 2008 T-SQL. Sample code is also provided in C# and C++, where appropriate. Although knowledge of these programming languages is not required, basic knowledge of procedural programming will help in understanding the code samples.

How This Book Is Structured

This book is designed to address the needs of T-SQL developers who develop SQL Server–based search applications and DBAs who support full-text search on SQL Server. For both types of readers, this book was written to act as a tutorial, describing basic full-text search functionality available through SQL Server, and as a reference to the new full-text search features and functionality available in SQL Server 2008. The following sections provide a chapter-by-chapter overview of the book's content.

Chapter 1

Chapter 1 begins by putting full-text search functionality in context. We discuss the history of SQL Server full-text search as well as the goals and purpose of full-text search, and provide an overview of SQL Server 2008 Integrated Full-Text Search (iFTS) architecture. We also define the concept of search quality and how it relates to iFTS.

Chapter 2

In Chapter 2, we discuss iFTS administration, setup, and configuration. In this chapter, we show how to set up and populate full-text indexes and full-text catalogs. We discuss full-text index change-tracking options and administration via SQL Server Management Studio (SSMS) wizards and T-SQL statements.

Chapter 3

Chapter 3 introduces iFTS basic and advanced query techniques. We use this chapter to demonstrate simple `FREETEXT`-style queries and more advanced `CONTAINS`-style query options. We look at the full range of iFTS query styles in this chapter, including Boolean search options, proximity search, prefix search, generational search, weighted search, phrase search, and other iFTS search options.

Chapter 4

Chapter 4 builds on the search techniques demonstrated in Chapter 3 and provides demonstrations of client interaction with the database via iFTS. This chapter will show you how to implement simple iFTS-based hit highlighting utilities and search engine–style search interfaces.

Chapter 5

SQL Server iFTS supports nearly 50 different languages right out of the box. In Chapter 5, we explore iFTS support for multilingual searching. We describe the factors that affect representation of international character sets and multilingual searches. We also provide best practices around multilingual searching.

Chapter 6

SQL Server 2008 provides greater flexibility and more options for storing large object (LOB) data in your databases. Chapter 6 discusses the options available for storing, managing, and indexing LOB data in your database. In this chapter, we take a look at how SQL Server indexes LOB data, including use of the new `FILESTREAM` option for efficient storage and streaming retrieval of documents from SQL Server and the NTFS file system.

Chapter 7

In Chapter 7, we discuss iFTS stoplists, which help you eliminate useless words from your searches. We discuss word frequency theory, system stoplists, and creating and managing custom stoplists.

Chapter 8

Chapter 8 provides insight into iFTS thesauruses, with examples of the types of functionality that can be built using thesaurus expansion and replacement sets, including “word bag” searches, translation, and error correction. We also discuss factors affecting thesaurus expansion and replacement, including diacritics sensitivity, nonrecursion, and overlapping rules.

Chapter 9

SQL Server 2008 iFTS provides greater transparency than any prior release of SQL Server FTS. Chapter 9 explores the new catalog views and dynamic management views and functions, all of which allow you to explore, manage, and troubleshoot your iFTS installations, full-text indexes, and full-text queries with greater insight, flexibility, and power than ever before.

Chapter 10

As with prior versions of SQL Server FTS, SQL Server 2008 iFTS depends on external components known as filters, word breakers, and stemmers. These components are critical to proper indexing and querying in iFTS. Chapter 10 discusses iFTS filters and other components, including custom filter creation. In this chapter, we explore creating a sample custom iFTS filter.

Chapter 11

SQL Server iFTS is a great tool for linguistic searches against documents and textual data, but it’s not optimized for other types of common database searches, such as name-based searching. In Chapter 11, we explore the world beyond iFTS and introduce fuzzy search technologies, such as phonetic search and n-grams, which fill the void between exact matches and linguistic full-text search.

Appendix A

In this book, we introduce several iFTS-related terms that may be unfamiliar to the uninitiated. We define these words in the body of the text where appropriate, and have included a quick reference glossary of iFTS-related search terms in Appendix A.

Appendix B

To provide more interesting examples than would be possible using the standard Adventure-Works sample database, we’ve decided to implement our own database known as `iFTS_Books`. This sample database includes the full text of dozens of public domain books in several languages, and provides concrete examples of the best practices we introduce in this book. Appendix B describes the structure and design of the `iFTS_Books` sample database.

Appendix C

Appendix C includes additional information about the mathematics and theory behind vector-space search, which is implemented in iFTS via weighted full-text searches.

Conventions

To make reading this book an enjoyable experience, and to help readers get the most out of the text, we've adopted standardized formatting conventions throughout.

C# and C++ code is shown in code font. Note that these languages are case sensitive. Here's an example of a line of C# code:

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

T-SQL source code is also shown in code font. Though T-SQL is not case sensitive, we've consistently capitalized keywords for readability. Also note that, for readability purposes, we've lowercased data type names in T-SQL code. Finally, following Microsoft's best practices, we consistently use the semicolon T-SQL statement terminator. The following demonstrates a line of T-SQL code:

```
DECLARE @x xml;
```

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

```
<book published = "Apress">Pro T-SQL 2008 Programmer&apos;s Guide</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 in order to run the T-SQL code samples provided. Note that the code in this book has been designed specifically to take advantage of SQL Server 2008 features, and most of the code in the book will either not run on prior versions of SQL Server, or will require significant modification to work on prior releases. The code samples provided in the book are designed specifically to run against the `iFTS_Books` sample database, available for download from the Apress web site at `www.apress.com` (see the following section). We describe the `iFTS_Books` database and provide installation instructions in Appendix B.

Other code samples provided in the book were written in C# (and C++ where appropriate) using Visual Studio 2008. If you're interested in compiling and executing the SQL CLR, client code, and other sample code provided, we highly recommend an installation of Visual Studio 2008 (with Service Pack 1 installed). Although you can compile the code from the command line, we find that the Visual Studio IDE provides a much more enjoyable and productive experience.

Some of the code samples may have additional requirements specified in order to use them; we will identify these special requirements as the code is presented.

Downloading the Code

The `iFTS_Books` sample database and all of the code samples presented in this book are available in a single Zip file from the Downloads section of the Apress web site at `www.apress.com`. The Zip file is structured so that each subdirectory contains a set of installation scripts or code samples presented in the book. Installation instructions for the `iFTS_Books` database and code samples are provided in Appendix B.

Contacting the Authors

The Apress team and the authors 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 directly to Apress by visiting `www.apress.com`, locating the page for this book, and clicking on Submit Errata. Alternatively, feel free to drop a line directly to the authors at `michaelco@optonline.net`.