

# SQL 2012 Developer, Part 12 of 13: Coding Techniques

page 1

**Meet the expert:** Frank Tillinghast is a senior consultant with MTOW Software Solutions. He is a Microsoft Certified Solution Developer and has been developing applications for over 15 years. Most of his time is spent consulting for companies nationwide with troubled projects or mentoring projects to successful completion. When he is not developing software or mentoring clients, Frank is teaching other developers. He has taught thousands of application developers how to create business solutions with Visual Studio .NET, VB.NET, ASP.NET, Visual C#, and SQL Server.

**Prerequisites:** This course assumes that you have a basic familiarity with the concept of relational databases and a basic understanding of what SQL Server is and the high-level tools in it, as well as how to create and manage objects using Management Studio. You should also have a basic understanding of how SQL Server implements security, including its authentication and authorization schemes, and how to assign permissions on securable objects to principals. You should know the fundamentals of Transact-SQL to write queries to retrieve and update data, and how to execute scripts using the query editor in Management Studio.

It will be very helpful, but not absolutely necessary, to have experience with .NET development using Visual Studio 2010 or 2012 for the portions of the course that deal with SQLCLR development. The .NET code included in this course is written in C#, but if you prefer Visual Basic it should be fairly easy to convert the code.

**Runtime:** 01:19:16

**Course description:** As business systems become more intricate, so does the T-SQL development needed to get at that data. In this course we'll discuss some of the challenges facing developers including complex data types and structures; the "art" of writing efficient queries; and how to tackle and simplify multi-step processes. First we'll discuss data types, implicit and explicit conversion, and precedence. Next, we'll move on to discuss working with multinational data and understanding collation—which are SQL Server's rules for sorting and comparing strings. We'll discuss working with and storing hierarchical data using the hierarchyid data type. Next you'll learn that even little things make a difference. We'll explore how to boost data operations by using JOINS rather than subqueries, and how to check for inclusion using EXISTS. We'll finish up by discussing temporary tables, table variables, user-defined table-valued functions, CTEs, worktables, and user-defined table type parameters. I'll show you how using implicit transactions can help in the debugging process and why keeping things simple usually makes the most sense.

## Course outline:

### Complex Data and Structures

- Introduction
- Complex Data and Structures
- Issues with Data Types
- Demo: CAST and TRY\_CAST
- Working with Multinational Data
- Demo: Collation
- Working with Hierarchical Data
- Demo: Hierarchical Structure
- Summary

- Demo: UDFs
- Common Table Expressions
- Worktables
- User-Defined Table Type Parameters
- Demo: CTEs
- Demo: Worktables
- Demo: Worktables Select
- Summary

### Efficient Queries

- Introduction
- Writing More Efficient Queries
- The EXISTS Clause
- Joins vs. Subqueries
- One-Pass Queries
- Demo: One-Pass Queries
- Writing More Efficient Queries
- Using Temporary Tables
- Table Variables
- User-Defined Table-Valued

### Complex Queries

- Introduction
- Working with Complex Queries
- Using Implicit Transactions
- Demo: Implicit Transactions
- Keeping it Simple
- Demo: Simplifying Procedures
- Demo: Create a Stored Procedure
- Summary