SQL 2012 Developer, Part 11 of 13: Complex Querying

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: 01:15:36

Course description: We'll start off by discussing the specifics of a topic you might not expect—handling NULLs! I know it sounds surprising, but it's important to really understand how to work with NULL values—especially if you plan to integrate with the Common Language Runtime. Next, we'll address another common and seemingly straightforward task—the ranking of data. We'll explore the specifics of how to rank grouped data using some T-SQL functions including ROW_NUMBER, RANK, DENSE_RANK, and NTILE. Next we'll take a look at how to identify correlated subqueries and how you can possibly replace them with more efficient constructs. Finally, we'll look at an alternative to subqueries, derived tables, and temporary tables—known as Common Table Expressions, or CTEs, they are similar to derived tables but they are self-referencing, can be used numerous times within the same query, and can be called recursively!

Course outline:

Complex Querying

- Introduction
- Working with NULL Values
- SqlTypes and CLR Types
- Demo: NULLs
- ANSI_NULLS
- Demo: ANSI_NULLS
- NULLs and SqlBoolean
- Assigning NULL Values
- CLR Integration
- Direct Assignment
- NULLIF
- COALESCE
- Demo: NULL Functions
- Demo: NULL Value Assignment
- Summary

Rankings

- Introduction
- Ranking Grouped Data
- The ROW_NUMBER Function
- The RANK Function
- DENSE_RANK Function
- The NTILE Function
- Demo: ROW_NUMBER
- Demo: RANK
- Demo: DENSE_RANK
- Demo: NTILE
- Summary

SubQueries

- Introduction
- Writing Correlated Subqueries
- Subquery Basics
- What is a Correlated Subquery?
- Demo: Subqueries
- Using the WHERE Clause
- ANY, SOME, and ALL Modifiers
- Including the HAVING Clause
- Subqueries and Updates
- Demo: ALL Modifier
- Demo: HAVING Clause
- Joins and Temp Tables
- Demo: Replacing a Subquery
- Summary

Common Table Expressions

- Introduction
- Using Common Table Expressions
- CTE Syntax
- Demo: Common Table Expression
- Recursive CTEs
- Demo: Recursive CTEs
- Demo: Executing a Recursive CTE
- Summary