

SQL Server Transact-SQL[®] Programming: Hands-On - 4 Days

Course 532 Overview

- You Will Learn How To**
- Develop database applications for SQL Server databases with Transact-SQL
 - Create modular code using stored procedures
 - Develop reusable code with scalar- and table-valued functions
 - Handle Transact-SQL runtime errors to create robust software
 - Audit data changes using AFTER triggers
 - Manage concurrency using transactions and locking options

Course Benefits Transact-SQL is integral to the power of SQL Server. Transact-SQL features allow developers to create scalable, distributed applications to meet the demanding requirements of modern organisations. This course provides extensive experience creating stored procedures and triggers, and developing T-SQL that utilises SQL Server to the fullest.

Who Should Attend Database developers, administrators and analysts who want to expand their knowledge of SQL Server 2005 or 2008. Experience with SQL Server at the level of Course 131, "SQL Server 2005 Comprehensive Introduction", or Course 137, "SQL Server 2008 Comprehensive Introduction", is assumed.

Hands-On Training A series of hands-on exercises provides immediate experience in Transact-SQL development, including:

- Adding a column to a database table using ALTER TABLE
- Passing data into a stored procedure using parameters
- Modifying data in a transaction
- Intercepting errors with TRY...CATCH
- Calling a user-defined function in a SQL statement
- Writing triggers to carry out advanced validation
- Tracing metadata changes with DDL triggers

SQL Server Transact-SQL[®] Programming: Hands-On - 4 Days

Course 532 Outline

SQL Server Architecture

- SQL Server edition overview
- Introducing the tools
- SQL Server Management Studio

Managing Tables with DDL

Creating schemas

- Managing schemas
- Referencing schemas versus using the default schema
- Hiding schemas with synonyms

Building tables

- Selecting appropriate SQL Server data types
- Constructing tables with CREATE TABLE

Adding constraints

- Enforcing uniqueness using PRIMARY KEY and UNIQUE constraints
- Validating relationships using FOREIGN KEY

Retrieving Data with Transact-SQL

Stored Procedures

Batch and stored procedure processing

- Minimising network traffic using batches and procedures
- Stored procedure compilation and execution
- Using scalar functions

Selecting data

- Developing stored procedures that extract data from multiple servers
- Executing dynamic queries using OPENROWSET and OPENQUERY
- Executing remote procedures
- Combining results from multiple databases
- Capturing RETURN values from stored procedures

Declaring variables and parameters

- Creating and utilising local variables
- Passing input and output parameters
- Interrogating global variables

Calling built-in scalar functions

- Converting data using CAST and CONVERT
- Ordering data with ranking functions

Maintaining Data

Modifying data

- Inserting, updating and deleting data

- Ensuring data consistency with transactions and distributed transactions
- Managing concurrency with isolation levels
- SQL Server locking fundamentals
- Avoiding blocking problems with read-committed snapshot isolation
- Managing locks using hints

Programming procedural statements

- Implementing conditions with IF...ELSE
- Looping with WHILE and GOTO
- Creating code blocks with BEGIN...END
- Debugging with PRINT
- Returning data using RETURN
- Debugging T-SQL in Management Studio

Handling errors

- Communicating problems to the client with RAISERROR
- Intercepting errors with TRY...CATCH
- Dealing with open transactions when an exception occurs

Producing server-side result sets

- Building and using temporary tables
- Processing rows on the server with a cursor
- Taking advantage of table variables

Developing Views, Functions and Triggers

Storing queries on the server

- Concealing complexity with views
- Solving business problems using multistatement table-valued functions

Creating user-defined functions

- Calculating values with scalar functions
- Processing multiple rows returned from a table-valued function
- Taking advantage of schema binding

Formulating triggers

- INSTEAD OF vs. AFTER triggers
- Detecting row changes using the inserted and deleted tables
- Tracking metadata changes with DDL triggers
- Auditing user access using a LOGON trigger
- Tracking data changes with the OUTPUT clause