

Course Outline

2784- Tuning and Optimizing Queries using Microsoft SQL Server 2005



Duration: 3 days (18 hours)

Learning Objectives:

- Normalize databases
- Design a normalized database
- Optimize a database design by denormalizing
- Optimize data storage
- Manage concurrency
- Optimize and tune queries for performance
- Optimize an indexing strategy
- Decide when cursors are appropriate
- Identify and resolve performance-limiting problems

Target Audience:

This course is intended for current professional database developers who have three or more years of on-the-job experience developing SQL Server database solutions in an enterprise environment

Prerequisites:

- Have working knowledge of data storage. Specifically, knowledge about row layout, fixed length field placement and varying length field placement.
- Be familiar with index structures and index utilization. Specifically, they must understand the interaction between non-clustered indexes, clustered indexes and heaps. They must know why a covering index can improve performance.
- Have had hands-on database developer experience. Specifically, three years of experience as a full-time database developer in an enterprise environment.
- Be familiar with the locking model. Specifically, students should have an understanding of lock modes, lock objects and isolation levels and be familiar with process blocking.
- Understand Transact-SQL syntax and programming logic. Specifically, students should be completely fluent in advanced queries, aggregate queries, subqueries, user-defined functions, cursors, control of flow statements, CASE expressions, and all types of joins.
- Be able to design a database to third normal form (3NF) and know the trade offs when backing out of the fully normalized design (denormalization) and designing for performance and business requirements in addition to being familiar with design models, such as Star and Snowflake schemas.
- Have strong monitoring and troubleshooting skills, including using monitoring tools.
- Have basic knowledge of the operating system and platform. That is, how the operating system integrates with the database, what the platform or operating system can do, and how interaction between the operating system and the database works.
- Have basic knowledge of application architecture. That is, how applications can be designed in three layers, what applications can do, how interaction between the application and the database works, and how the interaction between the database and the platform or operating system works.

- Know how to use a data modeling tool.
- Be familiar with SQL Server 2005 features, tools, and technologies.
- Have a Microsoft Certified Technology Specialist: Microsoft SQL Server 2005 credential - or equivalent experience.

In addition, it is recommended, but not required, that students have completed:

- Course 2778, Writing Queries Using Microsoft SQL Server 2005 Transact-SQL.
- Course 2779, Implementing a Microsoft SQL Server 2005 Database.
- Course 2780, Maintaining a Microsoft SQL Server 2005 Database.

Topics Covered:

- Measuring Database Performance
 - Importance of Benchmarking
 - Key Measures for Query Performance: Sysmon
 - Key Measures for Query Performance: Profiler
 - Guidelines for Identifying Locking and Blocking
 - Measuring Database Performance
 - Reviewing Tables and Scripts
 - Determining Performance Baselines
 - Prioritizing Identified Problems
- Optimizing Physical Database Design
 - Performance Optimization Model
 - Schema Optimization Strategy: Keys
 - Schema Optimization Strategy: Responsible Denormalization
 - Schema Optimization Strategy: Generalization
 - Optimizing Schemas
 - Optimizing Memberships
 - Optimizing Events
 - Normalizing Event Sponsorships
 - Denormalizing Membership Visits
 - Cleaning Up Schema
 - Adapting the Solution to the New Database Schema
 - Determining Performance
- Optimizing Queries for Performance
 - Performance Optimization Model: Queries
 - What Is Query Logical Flow?
 - Considerations for Using Subqueries
 - Guidelines for Building Efficient Queries
 - Optimizing Queries
 - Optimizing and Rewriting Slow Performing Stored Procedures
 - Optimizing and Rewriting Slow Performing Views
 - Optimizing and Rewriting Slow Performing Non-Cursor Aggregate Queries
 - Determining Performance
- Refactoring Cursors into Queries
 - Performance Optimization Model: Query-Set-based solutions
 - Five Steps to Building a Cursor

- Strategies for Refactoring Cursors
- Refactoring Cursors into Queries
- Refactoring the pMembershipCategory Cursor
- Refactoring the pCommunityImpact Cursor
- Refactoring the pMemberInvitation Cursor
- Determining Performance
- Optimizing an Indexing Strategy
 - Performance Optimization Model: Indexes
 - Considerations for Using Indexes
 - Best Uses of the Clustered Index
 - Best Practices for Non-Clustered Index Design
 - How to Document an Indexing Strategy
 - Optimizing an Indexing Strategy
 - Identifying Tables to Optimize
 - Designing Indexes
 - Determining Performance
- Managing Concurrency
 - Performance Optimization Model: Locking and Blocking
 - Multimedia - "How to Use Efficient Queries to Reduce Locking and Blocking"
 - Strategies to Reduce Locking and Blocking
 - Reducing Blocking
 - Identifying Code with Locking and Blocking Issues
 - Reducing Concurrency Issues
 - Determining Final Performance