

Tuning and Optimizing Queries using Microsoft SQL Server 2005

Course 2784A: Three days; Instructor-Led Course Syllabus

Introduction

This three-day instructor-led workshop provides database developers working in enterprise environments using Microsoft SQL Server 2005 the knowledge and skills to evaluate and improve queries and query response times. The workshop focuses on systematic identification and optimization of database factors that impact query performance.

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.

Objectives

After completing this course, students will be able to:

- Normalize databases.
- Design a normalized database.
- Optimize a database design by denormalizing.
- Optimize data storage.
- Manage concurrency
- Manage concurrency by selecting the appropriate transaction isolation level.
- Select a locking granularity level.
- Optimize and tune queries for performance.
- Optimize an indexing strategy.
- Decide when cursors are appropriate.
- Identify and resolve performance-limiting problems.

Prerequisites

Before attending this course, students must:

- 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.

Course Outline

Unit 1: Measuring Database Performance

This unit provides students with an opportunity to measure database performance and identify database performance bottlenecks. Students will use a sample script to identify performance and concurrency problems, capture baseline performance, and prioritize identified problems for optimization.

Unit 2: Optimizing Physical Database Design

In this unit, students work with strategies for optimizing physical database design. Students will optimize a database schema using normalization, generalization, and denormalization.

Unit 3: Optimizing Queries for Performance

In this unit students experience optimizing and tuning queries to improve performance. In the lab, students will optimize stored procedures, views, and non-cursor aggregate queries to improve database performance and user experience.

Each query that is optimized improves the overall system because the query will use fewer resources, freeing up those resources for other queries, and reducing the amount of locking done by the query. The domino effect is profound.

Unit 4: Refactoring Cursors into Queries

In this unit, students will work with strategies for refactoring cursors into queries. In the lab, students will work to optimize a database by replacing slow iterative code with faster set-based code.

Unit 5: Optimizing an Indexing Strategy

In this unit, students will work on optimizing indexing strategies. Students will work with a given database to add and delete indexes, by providing the optimum bridge between the query and the data without any redundancies.

Unit 6: Managing Concurrency

This unit provides students with the opportunity to work with concurrency management. Students will look for concurrency issues and then solve them by optimizing transactions and adjusting the transaction isolation level.