

Microsoft 2783

Designing the Data Tier for Microsoft SQL Server 2005

Length: One day

Introduction

Introduction

This one-day instructor-led clinic provides students with the knowledge and skills to design the data tier for Microsoft SQL Server 2005. The clinic focuses on teaching database developers working in enterprise environments to understand and decide how application developers are going to access and consume their data. This is a major failure point of database solutions today.

Audience

This clinic 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.

At Clinic Completion

After completing this clinic, students will be able to:

- Choose data access technologies and an object model to support an organization's business needs.
- Design an exception handling strategy.
- Choose a cursor strategy.
- Design query strategies using Multiple Active Result Sets (MARS).
- Design caching strategies for database applications.
- Design a scalable data tier for database applications.

Prerequisites

Before attending this clinic, students must:

- Have experience reading user requirements and business-need documents. For example, development project vision/mission statements or business analysis reports.
- Have basic knowledge of the Microsoft .NET Framework, .NET concepts, ADO.NET, and service oriented architecture (SOA).
- Be familiar with the tasks that application developers typically perform.
- Understand Transact-SQL syntax and programming logic.
- Have some experience with professional-level database design and know the tradeoffs 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 basic monitoring and troubleshooting skills. Specifically, how to use SQL Profiler and dynamic management views.
- 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.

Course Outline

Session 1: Choosing Data Access Technologies and an Object Model

This session explains how to choose data access technologies and an object model to support an organization's business needs.

Sections

- Introduction to Data Access Technologies
- Choosing Technologies for Accessing Data
- Building a Data Access Layer
- Designing Data Access from SQL Common Language Runtime (CLR) Objects
- Available Data Object Models for Administering SQL Server

After completing this session, students will be able to:

- Describe a typical database system and the role that data access technologies play in that system.
- Select appropriate technologies for accessing data stored in SQL Server 2005.
- Explain how to build a data access layer.
- Explain how to design SQL Server objects that use the In-Process data provider.
- Describe the data object models for administering SQL Server 2005 components and objects.

Session 2: Designing an Exception Handling Strategy

This session describes the various types of exceptions that can occur in a database system, how to capture them, and how to manage them appropriately.

Sections

- Exception Types and Their Purposes
- Detecting Exceptions
- Managing Exceptions

After completing this session, students will be able to:

- Describe the various types of exceptions that can be detected in a SQL Server 2005 system and how they affect applications and users.
- Design strategies to detect exceptions at the appropriate layer.
- Design strategies to log and communicate exceptions according to business requirements.

Session 3: Choosing a Cursor Strategy

This session describes when cursors are appropriate and how to use them to optimize the use of system resources.

Sections

- Common Scenarios for Row-Based vs. Set-Based Operations
- Selecting Appropriate Server-Side Cursors
- Selecting Appropriate Client-Side Cursors

After completing this session, students will be able to:

- Explain when cursors are appropriate and when they are not.
- Explain the considerations for selecting server-side cursors.
- Explain the considerations for selecting client-side cursors.

Session 4: Designing Query Strategies Using Multiple Active Result Sets

This session describes when Multiple Active Result Sets (MARS) can improve application response time and user satisfaction.

Sections

- Introduction to MARS
- Designing Query Strategies for Multiple Reads
- Designing Query Strategies for Mixing Reads and Writes in the Same Connection
- Concurrency Considerations When Using MARS

After completing this session, students will be able to:

- Explain why MARS is useful, as compared to the set-based execution of Microsoft SQL Server 2000.

- Explain when multiple simultaneous reads can be beneficial for an application, and explain the implications of using this technique.
- Explain specific scenarios in which it might be beneficial to use MARS to combine write and read operations.
- Explain the locking implications of using MARS and how these locks affect other transactions.

Session 5: Designing Caching Strategies for Database Applications

This session describes how to optimize system resources by caching data and objects in the appropriate layers.

Sections

- Why Caching Is Important
- Data and Query Caching in SQL Server 2005
- Using Caching Technologies Outside of SQL Server
- Custom Caching Techniques

After completing this session, students will be able to:

- Explain why caching is important.
- Explain the advantages of using the data and query caching automatically performed by SQL Server 2005.
- Explain how caching data outside of SQL Server works and how to manage conflicts that these technologies might produce.
- Explain the various ways to cache frequently used data, objects, and results in the appropriate tier to improve performance.

Session 6: Designing a Scalable Data Tier for Database Applications

This session describes how to assess scalability needs and design the best architecture to scale the system to meet those needs.

Sections

- Identifying the Need to Scale
- Scaling Database Applications to Avoid Concurrency Contention
- Scaling SQL Server Database Systems
- Scaling Database Applications Using a Service-Oriented Architecture
- Improving Availability and Scalability by Scaling Out Front-End Systems

After completing this session, students will be able to:

- Identify when to scale database applications and what layer to scale.
- Select an appropriate technology to avoid concurrency problems and to improve application performance.
- Evaluate whether scaling out or scaling up is appropriate for the scalability requirements of your database system.
- Explain how to improve middle tier processing by using multiple instances of Web services and object pooling.
- Explain how to improve response time and availability by scaling out front-end systems.