Programming a Microsoft SQL Server 2000 Database

Course number: 2073B
Course lenght: 5 days

Course Outline

Module 1: Overview of Programming SQL Server

This module provides students with an overview of enterprise-level application architecture and of Transact-SQL as a programming language. Transact-SQL is a data definition, manipulation, and control language. Students are assumed to be familiar with ANSI-SQL and basic programming concepts, such as functions, operators, variables, and control-of-flow statements. Students will also learn the different ways to execute Transact-SQL.

Lessons

- Introducing SQL Server Databases
- Working With SQL Server Programming Tools
- Understanding Transact-SQL Elements
- Programming Language Elements
- Executing Transact-SQL Statements

Lab : Overview of Transact-SQL

After completing this module, students will be able to:

- Describe SQL Server databases.
- Describe the primary SQL Server 2000 programming tools.
- Explain the difference between the two primary programming tools in SQL Server.
- Describe the basic elements of Transact-SQL.
- Describe the use of local variables, operators, functions, control-of-flow statements, and comments.
- Describe the various ways to execute Transact-SQL statements.

Module 2: Creating and Managing Databases

This module describes how to create a database, set database options, create filegroups, and manage a database and the transaction log. It reviews disk space allocation and how the transaction log records data modifications.

Lessons

- Defining Databases
- Using Filegroups
- Managing Databases

Lab : Creating and Managing Databases

After completing this module, students will be able to:

- Create a database.
- Work with filegroups.
- Manage a database.

Module 3: Creating Data Types and Tables

This module describes how to create data types and tables and generate Transact-SQL scripts containing statements that create a database and its objects.

Lessons

- Working with Data Types
- Working with Tables
- Generating Column Values
- Generating Scripts

Lab : Creating Data Types and Tables

After completing this module, students will be able to:

- Create and drop user-defined data types.
- Create and drop user tables.
- Generate column values.
- Generate a script.

Module 4: Implementing Data Integrity

This module shows how centrally-managed data integrity is a benefit of relational databases. Beginning with an introduction to data integrity concepts, including the methods available for enforcing data integrity, the module then introduces a section on constraints, the key method of ensuring data integrity. The module discusses the creation, implementation, and disabling of constraints and discusses how defaults and rules are an alternate way to enforce data integrity. The module concludes by comparing different data integrity methods.

Lessons

- Introducing Data Integrity
- Defining Constraints
- Understanding Constraint Types
- Disabling Constraints
- Using Defaults and Rules
- Deciding Which Enforcement Method to Use

Lab : Implementing Data Integrity

After completing this module, students will be able to:

- Describe the types of data integrity.
- Describe the methods to enforce data integrity.
- Determine which constraint to use, and create constraints.
- Define and use DEFAULT, CHECK, PRIMARY KEY, UNIQUE, and FOREIGN KEY constraints.
- Disable constraints.
- Describe and use defaults and rules.
- Determine which data-integrity enforcement methods to use.

Module 5: Planning Indexes

This module provides students with an overview of planning indexes. It explains how database performance can be improved with indexes; how clustered and nonclustered indexes are stored in SQL Server and how SQL Server retrieves rows by using indexes; and explores how SQL Server maintains indexes. The module concludes with guidelines for deciding which columns to index.

Lessons

- Introducing Indexes
- Understanding Index Architecture
- Retrieving Stored Data with SQL Server
- Maintaining Index and Heap Structures in SQL Server
- Deciding Which Columns to Index

Lab : Determining the Indexes of a Table

After completing this module, students will be able to:

- Describe why and when to use an index.
- Describe how SQL Server uses clustered and nonclustered indexes.
- Describe how SQL Server index architecture facilitates the retrieval of data.
- Describe how SQL Server maintains indexes and heaps.
- Describe the importance of selectivity, density, and distribution of data when deciding which columns to index.

Module 6: Creating and Maintaining Indexes

This module provides students with an overview of using the CREATE INDEX options to create and maintain indexes. It describes how maintenance procedures physically change the indexes; discusses maintenance tools; describes the use of statistics in SQL Server; and describes ways to verify that indexes are used and whether they perform optimally. The module concludes with a discussion of when to use the Index Tuning Wizard.

Lessons

- Creating Indexes
- Understanding Index Creation Options
- Maintaining Indexes

Lab : Creating and Maintaining Indexes

- Introducing Statistics
- Using the Index Tuning Wizard

Lab : Viewing Index Statistics

After completing this module, students will be able to:

- Create indexes and indexed views with unique or composite characteristics.
- Use the CREATE INDEX options.
- Describe how to maintain indexes over time.
- Describe how the query optimizer creates, stores, maintains, and uses statistics to optimize queries.
- Query the sysindexes system table.
- Describe how the Index Tuning Wizard works and when to use it.
- Describe performance considerations that affect creating and maintaining indexes.

Module 7: Implementing Views

This module defines views and their advantages, showing how views provide the ability to store a predefined query as an object in the database for later use. Views also offer a convenient way to hide sensitive data and the complexities of a database design and to provide a set of information without requiring the user to write or execute Transact-SQL statements. The module describes creating views and provides examples of how to include computed columns and built-in functions in the view definitions. The module then covers restrictions on modifying data through views. The last section discusses how views can improve performance.

Lessons

- Introducing Views
- Defining and Using Views
- Using Views to Optimize Performance

Lab : Implementing Views

After completing this module, students will be able to:

- Describe the concept of a view.
- List the advantages of views.
- Define a view by using the CREATE VIEW statement.
- Modify data through views.
- Optimize performance by using views.

Module 8: Implementing Stored Procedures

This module describes how to use stored procedures to improve application design and performance by encapsulating business rules. It discusses ways to process common queries and data modifications, and provides numerous examples and demonstrations of stored procedures.

Lessons

- Introducing Stored Procedures
- Creating, Modifying, Dropping, and ExecutingStored Procedures

Lab : Creating Stored Procedures

- Using Parameters in Stored Procedures
- Handling Error Messages
- Working with Stored Procedures

Lab : Creating Stored Procedures Using Parameters

After completing this module, students will be able to:

- Describe how a stored procedure is processed.
- Create, execute, modify, and drop a stored procedure.
- Create stored procedures that accept parameters.
- Create custom error messages.
- Use dynamic SQL in stored procedures.
- Execute extended stored procedures.

Module 9: Implementing User-Defined Functions

This module discusses the implementation of user-defined functions. It explains the three types of user-defined functions and the general syntax for creating and altering them, and provides an example of each type.

Lessons

- Introducing User-Defined Functions
- Implementing User-Defined Functions

Lab : Creating User-Defined Functions

After completing this module, students will be able to:

- Describe the three types of user-defined functions.
- Create and alter user-defined functions.
- Create each of the three types of user-defined functions.

Module 10: Implementing Triggers

This module shows that triggers are useful tools for database implementers who want certain actions to be performed whenever data is inserted, updated, or deleted from a specific table. Triggers are especially useful tools for cascading changes throughout other tables in the database while preserving complex referential integrity.

Lessons

- Introducing Triggers
- Creating, Altering, and Dropping Triggers
- Working with Triggers
- Implementing Triggers

Lab : Creating Triggers

After completing this module, students will be able to:

- Create a trigger.
- Drop a trigger.
- Alter a trigger.
- Evaluate the performance considerations that affect using triggers.

Module 11: Programming Across Multiple Servers

This module provides students with information on how to design security for a multi-server environment. It also explains the construction of distributed queries, distributed transactions, and partitioned views.

Lessons

- Introducing Distributed Queries
- Setting Up a Linked Server Environment
- Working with Linked Servers
- Using Partitioned Views

Lab : Using Distributed Data

After completing this module, students will be able to:

- Describe distributed queries.
- Write ad hoc queries that access data that is stored on a remote computer running Microsoft SQL Server 2000 or in an object linking and embedding database (OLE DB) data source.
- Set up a linked server environment to access data that is stored on a remote computer running SQL Server 2000 or in an OLE DB data source.
- Write queries that access data from a linked server.
- Execute stored procedures on a remote server or linked server.
- Manage distributed transactions.
- Use distributed transactions to modify distributed data.
- Use partitioned views to increase performance.

Module 12: Optimizing Query Performance

This module provides students with an in-depth look at how the query optimizer works, how to obtain query plan information, and how to implement indexing strategies.

Lessons

- Introducing the Query Optimizer
- Tuning Performance Using SQL Utilities
- Using an Index to Cover a Query
- Overriding the Query Optimizer
- Understanding Indexing Strategies and Guidelines

Lab : Optimizing Query Performance

After completing this module, students will be able to:

- Explain the role of the query optimizer and how it works to ensure that queries are optimized.
- Use various methods for obtaining execution plan information so that students can determine how the query optimizer processed a query and can validate that the most efficient execution plan was generated.
- Create indexes that cover queries.
- Identify indexing strategies that reduce page reads.
- Evaluate when to override the query optimizer.

Module 13: Performing Advance Query Analysis

This module describes how the query optimizer evaluates and processes queries that contain the AND operator, the OR operator, and join operations.

Lessons

- Analyzing Queries That Use the AND and OR Operator
- Analyzing Queries That Use Join Operations

Lab : Analyzing Queries That Use the AND and OR Operators
Lab : Analyzing Queries That Use Different Join Strategies

After completing this module, students will be able to:

- Analyze the performance gain of writing efficient queries while creating useful indexes for queries that contain the AND logical operator.
- Analyze the performance gain of writing efficient queries while creating useful indexes for queries that contain the OR logical operator.
- Evaluate how the query optimizer uses different join strategies for query optimization.

Module 14: Managing Transactions and Locks

This module discusses how transactions and locks ensure transaction integrity to accommodate multiple users. The module continues with a discussion of how transactions are executed and rolled back. A short animation helps to convey how transaction processing works. The module next describes how SQL Server locks maintain data consistency and concurrency. The module then introduces resources that can be locked, the different types of locks, and lock compatibility. A discussion follows on SQL Server dynamic locking based on schema and query. The final section describes locking options, discusses deadlocks, and explains how to display information on active locks.

Lessons

- Introducing Transactions and Locks
- Managing Transactions
- Understanding SQL Server Locking Architecture
- Managing Locks

Lab : Managing Transactions and Locks

After completing this module, students will be able to:

- Describe transaction processing.
- Execute, cancel, or roll back a transaction.
- Identify locking concurrency issues.
- Identify resource items that can be locked and the types of locks.
- Describe lock compatibility.
- Describe how SQL Server uses dynamic locking.
- Set locking options and display locking information.