



MICROSOFT
TRAINING
AND CERTIFICATION

Programming a Microsoft SQL Server 2000 Database

Delivery Guide

Course Number: 2073A

Information in this document is subject to change without notice. The names of companies, products, people, characters, and/or data mentioned herein are fictitious and are in no way intended to represent any real individual, company, product, or event, unless otherwise noted. Complying with all applicable copyright laws is the responsibility of the user. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Microsoft Corporation. If, however, your only means of access is electronic, permission to print one copy is hereby granted.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2000 Microsoft Corporation. All rights reserved.

Microsoft, ActiveX, BackOffice, MS-DOS, PowerPoint, Visual Basic, Visual C++, Visual Studio, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the U.S.A. and/or other countries.

Other product and company names mentioned herein may be the trademarks of their respective owners.

Project Lead: Rich Rose

Contents

Introduction	
Course Materials	2
Prerequisites	3
Course Outline	5
Setup	9
Microsoft Official Curriculum	12
Microsoft Certified Professional Program	13
Facilities	15
Module 1: SQL Server Overview	
Overview	1
What Is SQL Server?	2
SQL Server Integration	10
SQL Server Databases	15
SQL Server Security	25
Working with SQL Server	31
Lab A: SQL Server Overview	38
Review	44
Module 2: Overview of Programming SQL Server	
Overview	1
Designing Enterprise Application Architecture	2
SQL Server Programming Tools	7
The Transact-SQL Programming Language	9
Elements of Transact-SQL	10
Additional Language Elements	17
Ways to Execute Transact-SQL Statements	30
Recommended Practices	41
Lab A: Overview of Transact-SQL	42
Review	50
Module 3: Creating and Managing Databases	
Overview	1
Creating Databases	2
Creating Filegroups	10
Managing Databases	12
Introduction to Data Structures	21
Recommended Practices	27
Lab A: Creating and Managing Databases	28
Review	34

Module 4: Creating Data Types and Tables

Overview.....	1
Creating Data Types.....	2
Creating Tables.....	9
Generating Column Values.....	17
Generating Scripts.....	21
Recommended Practices.....	22
Lab A: Creating Data Types and Tables.....	23
Review.....	33

Module 5: Implementing Data Integrity

Overview.....	1
Types of Data Integrity.....	2
Enforcing Data Integrity.....	3
Defining Constraints.....	4
Types of Constraints.....	9
Disabling Constraints.....	17
Using Defaults and Rules.....	21
Deciding Which Enforcement Method to Use.....	23
Recommended Practices.....	24
Lab A: Implementing Data Integrity.....	25
Review.....	34

Module 6: Planning Indexes

Overview.....	1
Introduction to Indexes.....	2
Index Architecture.....	7
How SQL Server Retrieves Stored Data.....	12
How SQL Server Maintains Index and Heap Structures.....	19
Deciding Which Columns to Index.....	25
Recommended Practices.....	36
Lab A: Determining the Indexes of a Table.....	37
Review.....	41

Module 7: Creating and Maintaining Indexes

Overview.....	1
Creating Indexes.....	2
Creating Index Options.....	11
Maintaining Indexes.....	16
Lab A: Creating and Maintaining Indexes.....	25
Introduction to Statistics.....	35
Querying the sysindexes Table.....	45
Setting Up Indexes Using the Index Tuning Wizard.....	47
Performance Considerations.....	49
Recommended Practices.....	50
Lab B: Viewing Index Statistics.....	52
Review.....	61

Module 8: Implementing Views

Overview.....	1
Introduction to Views	2
Advantages of Views	3
Defining Views	5
Modifying Data Through Views.....	15
Optimizing Performance by Using Views	16
Recommended Practices	22
Lab A: Implementing Views.....	24
Review	33

Module 9: Implementing Stored Procedures

Overview.....	1
Introduction to Stored Procedures.....	2
Creating, Executing, Modifying, and Dropping Stored Procedures.....	10
Lab A: Creating Stored Procedures	20
Using Parameters in Stored Procedures	25
Executing Extended Stored Procedures	35
Handling Error Messages.....	37
Performance Considerations	43
Recommended Practices	45
Lab B: Creating Stored Procedures Using Parameters.....	46
Review	56

Module 10: Implementing User-defined Functions

Overview.....	1
What Is a User-defined Function?	2
Defining User-defined Functions.....	3
Examples of User-defined Functions	9
Recommended Practices	17
Lab A: Creating User-defined Functions	18
Review	24

Module 11: Implementing Triggers

Overview.....	1
Introduction to Triggers	2
Defining Triggers.....	9
How Triggers Work	15
Examples of Triggers.....	26
Performance Considerations	29
Recommended Practices	30
Lab A: Creating Triggers.....	31
Review	39

Module 12: Programming Across Multiple Servers

Overview.....	1
Introduction to Distributed Queries	2
Executing an Ad Hoc Query on a Remote Data Source.....	4
Setting Up a Linked Server Environment	7
Executing a Query on a Linked Server	16
Executing a Stored Procedure on a Linked Server.....	21
Managing Distributed Transactions	22
Modifying Data on a Linked Server.....	23
Using Partitioned Views	25
Recommended Practices	31
Lab A: Using Distributed Data	32
Review	43

Module 13: Optimizing Query Performance

Overview.....	1
Introduction to the Query Optimizer.....	2
Obtaining Execution Plan Information	13
Using an Index to Cover a Query.....	25
Indexing Strategies.....	35
Overriding the Query Optimizer	42
Recommended Practices	48
Lab A: Optimizing Query Performance.....	50
Review	65

Module 14: Analyzing Queries

Overview.....	1
Queries That Use the AND Operator.....	2
Queries That Use the OR Operator	4
Lab A: Analyzing Queries That Use the AND and OR Operators	5
Queries That Use Join Operations.....	32
Lab B: Analyzing Queries That Use Different Join Strategies	45
Recommended Practices	54
Review	55

Module 15: Managing Transactions and Locks

Overview.....	1
Introduction to Transactions and Locks.....	2
Managing Transactions.....	4
SQL Server Locking	12
Managing Locks.....	18
Recommended Practices	29
Lab A: Managing Transactions and Locks	30
Review	41

About This Course

This section provides you with a brief description of the course, audience, suggested prerequisites, and course objectives.

Description

This five-day course provides students with the technical skills required to program a database by using Microsoft® SQL Server™ 2000.

Course 2073A is a major revision of course 833, *Implementing a Database on Microsoft SQL Server 7.0*. This course incorporates new features of SQL Server 2000. The course omits some of the content on querying. The deleted content will be offered in a separate two-day course, course 2071, *Querying Microsoft SQL Server 2000 with Transact-SQL*. The course adds content from course 2013, *Optimizing Microsoft SQL Server 7.0*. The course contains a new module on user-defined functions.

Audience

This course is designed for those who are responsible for implementing database objects and programming SQL Server databases by using Transact-SQL.

Student Prerequisites

This course requires that students meet the following prerequisites:

- Experience using the Microsoft Windows® 2000 operating system to:
 - Connect clients running Windows 2000 to networks and the Internet.
 - Configure the Windows 2000 environment.
 - Create and manage user accounts.
 - Manage access to resources by using groups.
 - Configure and manage disks and partitions, including disk striping and mirroring.
 - Manage data by using NTFS.
 - Implement Windows 2000 security.
 - Optimize performance in Windows 2000.

For students who do not meet these prerequisites, the following courses provide students with the necessary knowledge and skills:

- Course 2051, *Microsoft Windows 2000 Network and Operating System Essentials*
- Course 2052, *Supporting Microsoft Windows 2000 Professional and Server*

- An understanding of basic relational database concepts, including:
 - Logical and physical database design.
 - Data integrity concepts.
 - Relationships between tables and columns (primary key and foreign key, one-to-one, one-to-many, many-to-many).
 - How data is stored in tables (rows and columns).

For students who do not meet these prerequisites, the following course provides students with the necessary knowledge and skills:

- Course 1609, *Designing Data Services and Data Models*
- Knowledge of basic Transact-SQL syntax (SELECT, UPDATE, and INSERT statements).

For students who do not meet these prerequisites, the following course provides students with the necessary knowledge and skills:

- Course 2071A, *Querying Microsoft SQL Server 2000 with Transact-SQL*
- Familiarity with the role of the database administrator.

Course Objectives

After completing this course, students will be able to:

- Describe the elements of SQL Server.
- Design a SQL Server enterprise application architecture.
- Describe the conceptual basis of programming in Transact-SQL.
- Create and manage databases and their related components.
- Implement data integrity by using the IDENTITY column property, constraints, defaults, rules, and unique identifiers.
- Plan for the use of indexes.
- Create and maintain indexes.
- Create, use, and maintain data views.
- Design, create, and use stored procedures.
- Implement user-defined functions.
- Create and implement triggers.
- Program across multiple servers by using distributed queries, distributed transactions, and partitioned views.
- Optimize query performance.
- Analyze queries.
- Manage transactions and locks to ensure data concurrency and recoverability.

Course Timing

The following schedule is an estimate of the course timing. Your timing may vary.

Day 1

Start	End	Module
9:00	9:30	Introduction
9:30	10:30	Module 1: SQL Server Overview
10:30	10:45	Break
10:45	11:15	Lab A: SQL Server Overview
11:15	12:00	Module 2: Overview of Programming SQL Server
12:00	1:00	Lunch
1:00	1:30	Lab A: Overview of Transact-SQL
1:30	2:00	Module 3: Creating and Managing Databases
2:00	2:30	Lab A: Creating and Managing Databases
2:30	2:45	Break
2:45	3:15	Module 4: Creating Data Types and Tables
3:15	4:00	Lab A: Creating Data Types and Tables

Day 2

Start	End	Module
9:00	9:30	Day 1 review
9:30	10:15	Module 5: Implementing Data Integrity
10:15	10:30	Break
10:30	11:00	Lab A: Implementing Data Integrity
11:00	12:30	Module 6: Planning Indexes
12:30	1:30	Lunch
1:30	1:45	Lab A: Determining the Indexes of a Table
1:45	2:15	Module 7: Creating and Maintaining Indexes
2:15	2:45	Lab A: Creating and Maintaining Indexes
2:45	3:00	Break
3:00	3:30	Module 7: Creating and Maintaining Indexes (continued)
3:30	4:00	Lab B: Viewing Index Statistics

Day 3

Start	End	Module
9:00	9:30	Day 2 review
9:30	10:15	Module 8: Implementing Views
10:15	10:45	Lab A: Implementing Views
10:45	11:00	Break
11:00	11:30	Module 9: Implementing Stored Procedures
11:30	11:45	Lab A: Creating Stored Procedures
11:45	12:45	Lunch
12:45	1:45	Module 9: Implementing Stored Procedures (continued)
1:45	2:30	Lab B: Creating Stored Procedures Using Parameters
2:30	2:45	Break
2:45	3:15	Module 10: Implementing User-defined Functions
3:15	4:00	Lab A: Creating User-defined Functions

Day 4

Start	End	Module
9:00	9:30	Day 3 review
9:30	10:15	Module 11: Implementing Triggers
10:15	10:30	Break
10:30	11:00	Lab A: Creating Triggers
11:00	12:00	Module 12: Programming Across Multiple Servers
12:00	1:00	Lunch
1:00	2:00	Lab A: Using Distributed Data
2:00	2:15	Break
2:15	4:00	Module 13: Optimizing Query Performance

Day 5

Start	End	Module
9:00	9:30	Day 4 review
9:30	10:15	Lab A: Optimizing Query Performance
10:15	10:30	Break
10:30	11:00	Module 14: Analyzing Queries
11:00	12:00	Lab A: Analyzing Queries That Use the AND and OR Operators
12:00	1:00	Lunch
1:00	1:30	Module 14: Analyzing Queries (continued)
1:30	2:00	Lab B: Analyzing Queries That Use Different Join Strategies
2:00	2:15	Break
2:15	3:00	Module 15: Managing Transactions and Locks
3:00	4:00	Lab A: Managing Transactions and Locks

Trainer Materials Compact Disc Contents

The Trainer Materials compact disc contains the following files and folders:

- *Default.htm*. This file opens the Trainer Materials Web page.
- *Readme.txt*. This file contains a description of the compact disc contents and setup instructions in ASCII format (non-Microsoft Word document).
- *2073A_ms.doc*. This file is the Manual Classroom Setup Guide. It contains the steps for manually installing the classroom computers.
- *2073A_sg.doc*. This file is the Classroom Setup Guide. It contains a description of classroom requirements, classroom configuration, instructions for using the automated classroom setup scripts, and the Classroom Setup Checklist.
- *Errorlog*. This folder contains a template that is used to record any errors and corrections that you find in the course.
- *Fonts*. This folder contains fonts that are required to view the Microsoft PowerPoint® presentation and Web-based materials.
- *Mplayer*. This folder contains files that are required to install Microsoft Windows Media™ Player.
- *Powerpnt*. This folder contains the PowerPoint slides that are used in this course.
- *Pptview*. This folder contains the PowerPoint Viewer, which is used to display the PowerPoint slides.
- *Studentcd*. This folder contains the Web page that provides students with links to resources pertaining to this course, including additional reading, review and lab answers, lab files, multimedia presentations, and course-related Web sites.
- *Tprep*. This folder contains the Trainer Preparation Presentation, a narrated presentation that explains the instructional strategy for the course and presentation tips and caveats. To open the presentation, on the Trainer Materials Web page, click **Trainer Preparation Presentation**.

Student Materials Compact Disc Contents

The Student Materials compact disc contains the following files and folders:

- *Default.htm*. This file opens the Student Materials Web page. It provides you with resources pertaining to this course, including additional reading, review and lab answers, lab files, multimedia presentations, and course-related Web sites.
- *Readme.txt*. This file contains a description of the compact disc contents and setup instructions in ASCII format (non-Microsoft Word document).
- *AddRead*. This folder contains additional reading pertaining to this course.
- *Answers*. This folder contains answers to any questions in the modules and hands-on labs.
- *Appendix*. This folder contains Appendix files for this course.
- *Fonts*. This folder contains fonts that are required to view the PowerPoint presentation and Web-based materials.
- *Labfiles*. This folder contains files that are used in the hands-on labs. These files may be used to prepare the student computers for the hands-on labs.
- *Media*. This folder contains files that are used in multimedia presentations for this course.
- *Mplayer*. This folder contains files that are required to install Microsoft Windows Media™ Player.
- *Pptview*. This folder contains the PowerPoint Viewer, which is used to display the PowerPoint presentations that accompany the additional reading.
- *Webfiles*. This folder contains the files that are required to view the Student Materials Web page.
- *Wordview*. This folder contains the Word Viewer that is used to view any Word document (.doc) files that are included on the compact disc.

Document Conventions

The following conventions are used in course materials to distinguish elements of the text.

Convention	Use
◆	Indicates an introductory page. This symbol appears next to a topic heading when additional information on the topic is covered on the page or pages that follow it.
bold	Represents commands, command options, and syntax that must be typed exactly as shown. It also indicates commands on menus and buttons, dialog box titles and options, and icon and menu names.
<i>italic</i>	In syntax statements or descriptive text, indicates argument names or placeholders for variable information. Italic is also used for introducing new terms, for book titles, and for emphasis in the text.
Title Capitals	Indicate domain names, user names, computer names, directory names, and folder and file names, except when specifically referring to case-sensitive names. Unless otherwise indicated, you can use lowercase letters when you type a directory name or file name in a dialog box or at a command prompt.
ALL CAPITALS	Indicate the names of keys, key sequences, and key combinations—for example, ALT+SPACEBAR.
monospace	Represents code samples or examples of screen text.
[]	In syntax statements, enclose optional items. For example, <i>[filename]</i> in command syntax indicates that you can choose to type a file name with the command. Type only the information within the brackets, not the brackets themselves.
{ }	In syntax statements, enclose required items. Type only the information within the braces, not the braces themselves.
	In syntax statements, separates an either/or choice.
▶	Indicates a procedure with sequential steps.
...	In syntax statements, specifies that the preceding item may be repeated.
.	Represents an omitted portion of a code sample.

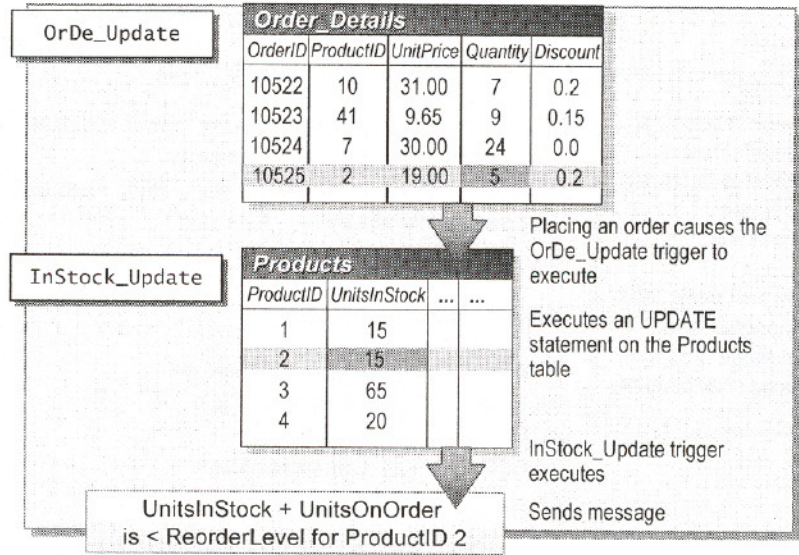
How Nested Triggers Work

Topic Objective

To discuss the use of nested triggers.

Lead-in

As mentioned previously, triggers can be nested up to 32 levels deep. If nested triggers are enabled, a trigger that changes a table can activate a second trigger, which in turn can activate a third trigger, and so on.



Any trigger can contain an UPDATE, INSERT, or DELETE statement that affects another table. With nesting enabled, a trigger that changes a table can activate a second trigger, which in turn can activate a third trigger, and so on. Nesting is enabled at installation, but you can disable and re-enable it by using the **sp_configure** system stored procedure.

Triggers can be nested up to 32 levels deep. If any trigger in a nested chain sets off an infinite loop, the nesting level is exceeded. The trigger then terminates and rolls back the transaction. You can use nested triggers to perform functions, such as the storage of a backup copy of rows that were affected by a previous trigger. Consider the following facts when you use nested triggers:

- By default, the nested triggers configuration option is on.
- A nested trigger will not fire twice in the same trigger transaction; a trigger does not call itself in response to a second update to the same table within the trigger. For example, if a trigger modifies a table that, in turn, modifies the original trigger table, the trigger does not fire again.
- Because a trigger is a transaction, a failure at any level of a set of nested triggers cancels the entire transaction, and all data modifications are rolled back. Therefore, you should include PRINT statements when you test triggers so that you can determine where the failure occurred.

Delivery Tip

The @@NESTLEVEL function is useful when testing and troubleshooting triggers but would not typically be included in a production environment.

Checking the Nesting Level

Each time that a nested trigger fires, the nesting level increments. SQL Server supports up to 32 levels of nesting, but you may want to limit the levels of nesting to avoid exceeding the maximum nesting level. You can use the @@NESTLEVEL function to see the current levels of nesting.

Determining Whether to Use Nesting

Nesting is a powerful feature that you can use to maintain data integrity throughout a database. Occasionally, however, you may want to disable nesting. If nesting is disabled, a trigger that modifies another table does not invoke any of the triggers on the second table.

Use the following statement to disable nesting:

Syntax

```
sp_configure 'nested triggers', 0
```

You may decide to disable nesting because:

- Nested triggers require a complex and well-planned design. Cascading changes can modify data that you did not intend to affect.
- A data modification at any point in a series of nested triggers sets off the trigger series. Although this offers powerful protection for your data, it can be a problem if your tables must be updated in a specific order.

You can create the same functionality with or without the nesting feature; however, your trigger design will differ substantially. In designing nested triggers, each trigger should initiate only the next data modification—the design should be modular. In designing non-nested triggers, each trigger should initiate all data modifications that you want it to make.

Example

This example shows how placing an order causes the **OrDe_Update** trigger to execute. This trigger executes an UPDATE statement on the **UnitsInStock** column of the **Products** table. When the update occurs, it fires the **Products_Update** trigger and compares the new value of the stock in inventory, plus the stock on order, to the reorder level. If the stock in inventory plus the stock on order falls below the reorder level, a message is sent alerting the buyer to purchase more stock.

```
USE Northwind
GO
CREATE TRIGGER Products_Update
  ON Products
  FOR UPDATE
AS
IF UPDATE (UnitsInStock)
  IF (Products.UnitsInStock + Products.UnitsOnOrder) <
  Products.ReorderLevel
BEGIN
  --Send message to the purchasing department
END
```