



## Course Outline

### Implementing a Microsoft SQL Server 2008 Database

Course 6232A: Five days; Instructor-Led

#### Introduction

Elements of this syllabus are subject to change.

This five-day instructor-led course provides students with the knowledge and skills to implement a Microsoft SQL Server 2008 database. The course focuses on teaching individuals how to use SQL Server 2008 product features and tools related to implementing a database.

#### Audience

This course is intended for IT Professionals who administer and maintain SQL Server databases.

#### At Course Completion

After completing this course, students will be able to:

- Create databases and database files.
- Create data types and tables.
- Use XML-related features in Microsoft SQL Server 2008.
- Plan, create, and optimize indexes.
- Implement data integrity in Microsoft SQL Server 2008 databases by using constraints.
- Implement data integrity in Microsoft SQL Server 2008 by using triggers.
- Implement views.
- Implement stored procedures.
- Implement functions.
- Implement managed code in the database.
- Manage transactions and locks.
- Use Service Broker to build a messaging-based solution.

#### Prerequisites

Before attending this course, students must have:

- Basic knowledge of the Microsoft Windows operating system and its core functionality.
- Working knowledge of Transact-SQL.
- Working knowledge of relational databases.
- Some experience with database design.



## Course Outline

### Course Outline

#### ***Module 1: Creating Databases and Database Files***

The students will learn one of the most fundamental tasks that a database developer must perform, the creation of a database and its major components, such as creating databases, setting database options, creating filegroups, schemas, and database snapshots.

##### **Lessons**

- Lesson 1: Creating Databases
- Lesson 2: Creating Filegroups
- Lesson 3: Creating Schemas
- Lesson 4: Creating Database Snapshots

##### **Lab: Creating Databases and Database Files**

- Exercise 1: (Level 200) Creating a Database
- Exercise 2: (Level 200) Creating Schemas
- Exercise 3: (Level 300) Creating a Database Snapshot

After completing this module, students will be able to:

- Create databases
- Create filegroups
- Create schemas
- Create database snapshots

#### ***Module 2: Creating Data Types and Tables***

The students will learn about the system-supplied data types in SQL Server 2008. They will learn how to define custom Transact-SQL data types and how to create tables and how to use partitioned tables to organize data into multiple partitions.

##### **Lessons**

- Lesson 1: Creating Data Types
- Lesson 2: Creating Tables
- Lesson 3: Creating Partitioned Tables

##### **Lab: Creating Data Types and Tables**

- Exercise 1 (Level 200): Creating Data Types
- Exercise 2 (Level 200): Using New Date and Time Data Types
- Exercise 3 (Level 200): Creating Tables
- Exercise 4: (Level 300) Creating Partitioned Tables

After completing this module, students will be able to:



## Course Outline

- Create new data types.
- Create new tables.
- Create partitioned tables.

### ***Module 3: Creating and Tuning Indexes***

The students will learn how to plan, create, and optimize indexes to attain optimal performance benefits.

#### **Lessons**

- Lesson 1: Planning Indexes
- Lesson 2: Creating Indexes
- Lesson 3: Optimizing Indexes

#### **Lab: Creating and Tuning Indexes**

- Exercise 1: (Level 200) Creating Indexes
- Exercise 2: (Level 300) Tuning Indexes

After completing this module, students will be able to:

- Plan indexes.
- Create indexes.
- Optimize indexes.

### ***Module 4: Implementing Data Integrity by Using Constraints and Triggers***

The students will learn about implementing data integrity in SQL Server 2008 by using constraints. They will also implement data integrity by using triggers.

#### **Lessons**

- Lesson 1: Data Integrity Overview
- Lesson 2: Implementing Constraints
- Lesson 3: Implementing Triggers

#### **Lab: Implementing Data Integrity by Using Constraints and Triggers**

- Exercise 1: (Level 200) Creating Constraints
- Exercise 2: (Level 200) Disabling Constraints
- Exercise 3: (Level 300) Creating Triggers

After completing this module, students will be able to:

- Describe the options for enforcing data integrity in SQL Server 2008.
- Implement data integrity in SQL Server 2008 databases by using constraints.
- Implement data integrity in SQL Server 2008 databases by using triggers.



## Course Outline

### **Module 5: Using XML**

The students will learn how to work with XML, including use of the FOR XML clause, the OPENXML function, XQuery expressions, and the xml native data type. They will learn the considerations to be taken into account when creating XML indexes and the syntax used to create the XML indexes. They will also learn what XML schemas and XML schema collections are as well as how to use them to implement typed XML data.

#### **Lessons**

- Lesson 1: Using the XML Data Type
- Lesson 2: Retrieving XML by Using FOR XML
- Lesson 3: Shredding XML by Using OPENXML
- Lesson 4: Introducing XQuery
- Lesson 5: Creating XML Indexes
- Lesson 6: Implementing XML Schemas

#### **Lab: Using XML**

- Exercise 1: (Level 200) Mapping Relational Data and XML
- Exercise 2: (Level 200) Storing XML Natively in the Database
- Exercise 3: (Level 300) Using XQuery with XML Methods
- Exercise 4: (Level 200) Create XML Indexes
- Exercise 5: (Level 300) Implementing XML Schemas

After completing this module, students will be able to:

- Use the xml data type.
- Retrieve XML by using the FOR XML clause.
- Shred XML by using the OPENXML function.
- Use XQuery expressions.
- Create XML indexes.
- Implement data integrity in SQL Server 2008 databases by using XML schemas.

### **Module 6: Implementing Views**

The students will be introduced to the different types of views available in Microsoft SQL Server 2008 which provide a convenient way to access data through a predefined query.

#### **Lessons**

- Lesson 1: Introduction to Views
- Lesson 2: Creating and Managing Views
- Lesson 3: Optimizing Performance by Using Views

#### **Lab: Implementing Views**

- Exercise 1: (Level 200) Creating Views
- Exercise 2: (Level 200) Creating Indexed Views



## Course Outline

- Exercise 3: (Level 200) Creating Partitioned Views

After completing this module, students will be able to:

- Describe the purpose of views.
- Create and manage views.
- Explain how to optimize query performance by using views.

### ***Module 7: Implementing Stored Procedures***

The students will learn the design and implementation of stored procedures to enforce business rules or data consistency, or to modify and maintain existing stored procedures written by other developers.

#### **Lessons**

- Lesson 1: Implementing Stored Procedures
- Lesson 2: Creating Parameterized Stored Procedures
- Lesson 3: Working With Execution Plans
- Lesson 4: Handling Errors

#### **Lab: Implementing Stored Procedures**

- Exercise 1: (Level 300) Creating Stored Procedures
- Exercise 2: (Level 300) Working with Execution Plans

After completing this module, students will be able to:

- Implement stored procedures.
- Create parameterized stored procedures.
- Work with execution plans.
- Handle errors in stored procedures.

### ***Module 8: Implementing Functions***

The students will learn the design and implementation of user-defined functions that enforce business rules or data consistency, or to modify and maintain existing functions written by other developers.

#### **Lessons**

- Lesson 1: Creating and Using Functions
- Lesson 2: Working with Functions
- Lesson 3: Controlling Execution Context

#### **Lab: Implementing Functions**

- Exercise 1: (Level 300) Creating Functions
- Exercise 2: (Level 300) Controlling Execution Context



## Course Outline

After completing this module, students will be able to:

- Create and use functions.
- Work with functions.
- Control execution context.

### ***Module 9: Implementing Managed Code in the Database***

The students will learn to use managed code to implement database objects, such as stored procedures, user-defined data types, user-defined functions, and triggers.

#### **Lessons**

- Lesson 1: Introduction to the SQL Server Common Language Runtime
- Lesson 2: Importing and Configuring Assemblies
- Lesson 3: Creating Managed Database Objects

#### **Lab: Implementing Managed Code in the Database**

- Exercise 1: (Level 300) Importing an Assembly
- Exercise 2: (Level 300) Creating Managed Database Objects

After completing this module, students will be able to:

- Identify appropriate scenarios for managed code in the database.
- Import and configure assemblies.
- Create managed database objects.

### ***Module 10: Managing Transactions and Locks***

The students will learn to use transactions and SQL Server locking mechanisms to meet the performance and data integrity requirements of their applications.

#### **Lessons**

- Lesson 1: Overview of Transactions and Locks
- Lesson 2: Managing Transactions
- Lesson 3: Understanding SQL Server Locking Architecture
- Lesson 4: Managing Locks

#### **Lab: Managing Transactions and Locks**

- Exercise 1: (Level 300) Using Transactions
- Exercise 2: (Level 300) Managing Locks
- Exercise 3: (Level 300) Using Partition Locking

After completing this module, students will be able to:



## Course Outline

- Describe how SQL Server 2008 transactions use locks.
- Execute and cancel a transaction.
- Describe concurrency issues and SQL Server 2008 locking mechanisms.
- Manage locks.

### ***Module 11: Using Service Broker***

The students will learn to use Service Broker to create secure, reliable, and scalable applications.

#### **Lessons**

- Lesson 1: Service Broker Overview
- Lesson 2: Creating Service Broker Objects
- Lesson 3: Sending and Receiving Messages

#### **Lab: Using Service Broker**

- Exercise 1: (Level 300) Creating Service Broker Objects
- Exercise 2: (Level 300) Implementing the Initiating Service
- Exercise 3: (Level 300) Implementing the Target Service

After completing this module, students will be able to:

- Describe Service Broker functionality and architecture.
- Create Service Broker objects.
- Send and receive Service Broker messages.