Developing SQL Databases

OD20762B; On-Demand, Video-based

Course Description

This course provides students with the knowledge and skills to develop a Microsoft SQL Server 2016 database. The course focuses on teaching individuals how to use SQL Server 2016 product features and tools related to developing a database.

Course Objectives

After completing this course, students will be able to:

- Design and Implement Tables
- Describe advanced table designs
- Ensure Data Integrity through Constraints
- Describe indexes, including Optimized and Columnstore indexes
- Design and Implement Views
- Design and Implement Stored Procedures
- Design and Implement User Defined Functions
- Respond to data manipulation using triggers
- Design and Implement In-Memory Tables
- Implement Managed Code in SQL Server
- Store and Query XML Data
- Work with Spatial Data
- Store and Query Blobs and Text Documents

Audience

The primary audience for this course is IT Professionals who want to become skilled on SQL Server 2016 product features and technologies for implementing a database.

The secondary audiences for this course are individuals who are developers from other product platforms looking to become skilled in the implementation of a SQL Server 2016 database.

Prerequisites

- Basic knowledge of the Microsoft Windows operating system and its core functionality
- Working knowledge of Transact-SQL

- Working knowledge of relational databases

Course Outline

Module 1: Introduction to Database Development

Before beginning to work with Microsoft SQL Server in either a development or an administration role, it is important to understand the scope of the SQL Server platform. In particular, it is useful to understand that SQL Server is not just a database engine—it is a complete platform for managing enterprise data. SQL Server provides a strong data platform for all sizes of organizations, in addition to a comprehensive set of tools to make development easier, and more robust.

Lessons

- Introduction to the SQL Server Platform
- SQL Server Database Development Tasks

After completing this module, you will be able to:

- Describe the SQL Server platform
- Use SQL Server administration tools

Module 2: Designing and Implementing Tables

In a relational database management system (RDBMS), user and system data is stored in tables. Each table consists of a set of rows that describe entities and a set of columns that hold the attributes of an entity. For example, a Customer table might have columns such as CustomerName and CreditLimit, and a row for each customer. In Microsoft SQL Server data management software tables are contained within schemas that are very similar in concept to folders that contain files in the operating system. Designing tables is one of the most important tasks that a database developer undertakes, because incorrect table design leads to the inability to query the data efficiently. After an appropriate design has been created, it is important to know how to correctly implement the design.

Lessons

- Designing Tables
- Data Types
- Working with Schemas
- Creating and Altering Tables

Lab: Designing and Implementing Tables

- Designing Tables
- Creating Schemas
- Creating Tables

After completing this module, you will be able to:

- Design tables using normalization, primary and foreign keys
- Work with identity columns
- Understand built-in and user data types
- Use schemas in your database designs to organize data, and manage object security
- Work with computed columns and temporary tables

Module 3: Advanced Table Designs

The physical design of a database can have a significant impact on the ability of the database to meet the storage and performance requirements set out by the stakeholders. Designing a physical database implementation includes planning the filegroups, how to use partitioning to manage large tables, and using compression to improve storage and performance. Temporal tables are a new feature in SQL Server 2016 and offer a straightforward solution to collecting changes to your data.

Lessons

- Partitioning Data
- Compressing Data
- Temporal Tables

Lab: Using Advanced Table Designs

- Partitioning Data
- Compressing Data

After completing this module, you will be able to:

- Describe the considerations for using partitioned tables in a SQL Server database
- Plan for using data compression in a SQL Server database
- Use temporal tables to store and query changes to your data

Module 4: Ensuring Data Integrity through Constraints

The quality of data in your database largely determines the usefulness and effectiveness of applications that rely on it—the success or failure of an organization or a business venture could depend on it. Ensuring data integrity is a critical step in maintaining high-quality data. You should enforce data integrity at all levels of an application from first entry or collection through storage. Microsoft SQL Server data management software provides a range of features to simplify the job.

Lessons

- Enforcing Data Integrity
- Implementing Data Domain Integrity
- Implementing Entity and Referential Integrity

Lab: Using Data Integrity Through Constraints

- Add Constraints
- Test the Constraints

After completing this module, you will be able to:

Describe the options for enforcing data integrity, and the levels at which they should be applied.

Implement domain integrity through options such as check, unique, and default constraints.

Implement referential integrity through primary and foreign key constraints.

Module 5: Introduction to Indexes

An index is a collection of pages associated with a table. Indexes are used to improve the performance of queries or enforce uniqueness. Before learning to implement indexes, it is helpful to understand how they work, how effective different data types are when used within indexes, and how indexes can be constructed from multiple columns. This module discusses table structures that do not have indexes, and the different index types available in Microsoft SQL Server.

Lessons

- Core Indexing Concepts
- Data Types and Indexes
- Heaps, Clustered, and Nonclustered Indexes
- Single Column and Composite Indexes

Lab: Implementing Indexes

- Creating a Heap
- Creating a Clustered Index
- Creating a Covered Index

After completing this module, you will be able to:

- Explain core indexing concepts
- Evaluate which index to use for different data types
- Describe the difference between single and composite column indexes.

Module 6: Designing Optimized Index Strategies

Indexes play an important role in enabling SQL Server to retrieve data from a database quickly and efficiently. This module discusses advanced index topics including covering indexes, the INCLUDE clause, query hints, padding and fill factor, statistics, using DMOs, the Database Tuning Advisor, and Query Store.

Lessons

- Index Strategies
- Managing Indexes
- Execution Plans
- The Database Engine Tuning Advisor
- Query Store

Lab: Optimizing Indexes

- Using Query Store
- Heaps and Clustered Indexes
- Creating a Covered Index

After completing this module, you will be able to:

- What a covering index is, and when to use one
- The issues involved in managing indexes
- Actual and estimated execution plans
- How to use Database Tuning Advisor to improve the performance of queries
- How to use Query Store to improve query performance

Module 7: Columnstore Indexes

Introduced in Microsoft SQL Server 2012, columnstore indexes are used in large data warehouse solutions by many organizations. This module highlights the benefits of using these indexes on large datasets; the improvements made to columnstore indexes in SQL Server 2016; and the considerations needed to use columnstore indexes effectively in your solutions.

Lessons

- Introduction to Columnstore Indexes
- Creating Columnstore Indexes
- Working with Columnstore Indexes

Lab: Using Columnstore Indexes

- Creating a Columnstore Index
- Create a Memory Optimized Columnstore Table

After completing this module, you will be able to:

- Describe columnstore indexes and identify suitable scenarios for their use
- Create clustered and nonclustered columnstore indexes
- Describe considerations for using columnstore indexes

Module 8: Designing and Implementing Views

This module describes the design and implementation of views. A view is a special type of query—one that is stored and can be used in other queries—just like a table. With a view, only the query definition is stored on disk; not the result set. The only exception to this is indexed views, when the result set is also stored on disk, just like a table. Views simplify the design of a database by providing a layer of abstraction, and hiding the complexity of table joins. Views are also a way of securing your data by giving users permissions to use a view, without giving them permissions to the underlying objects. This means data can be kept private, and can only be viewed by appropriate users.

Lessons

- Introduction to Views
- Creating and Managing Views
- Performance Considerations for Views

Lab: Designing and Implementing Views

- Creating Standard Views
- Creating an Updateable view

After completing this module, you will be able to:

- Understand the role of views in database design
- Create and manage views
- Understand the performance considerations with views

Module 9: Designing and Implementing Stored Procedures

This module describes the design and implementation of stored procedures.

Lessons

- Introduction to Stored Procedures
- Working with Stored Procedures
- Implementing Parameterized Stored Procedures
- Controlling Execution Context

Lab: Designing and Implementing Stored Procedures

- Create Stored procedures
- Create Parameterized Stored procedures
- Changes Stored Procedure Execution Context

After completing this module, you will be able to:

- Understand what stored procedures are, and what benefits they have
- Design, create, and alter stored procedures
- Control the execution context of stored procedures
- Implement stored procedures that use parameters

Module 10: Designing and Implementing User-Defined Functions

Functions are routines that you use to encapsulate frequently performed logic. Rather than having to repeat the function logic in many places, code can call the function. This makes code more maintainable, and easier to debug. In this module, you will learn to design and implement user-defined functions (UDFs) that enforce business rules or data consistency. You will also learn how to modify and maintain existing functions.

Lessons

- Overview of Functions
- Designing and Implementing Scalar Functions
- Designing and Implementing Table-Valued Functions
- Considerations for Implementing Functions
- Alternatives to Functions

Lab: Designing and Implementing User-Defined Functions

- Format Phone numbers
- Modify an Existing Function

After completing this module, you will be able to:

- Describe different types of functions
- Design and implement scalar functions
- Design and implement table-valued functions (TVFs)
- Describe considerations for implementing functions
- Describe alternatives to functions

Module 11: Responding to Data Manipulation via Triggers

Data Manipulation Language (DML) triggers are powerful tools that you can use to enforce domain, entity, referential data integrity and business logic. The enforcement of integrity helps you to build reliable applications. In this module, you will learn what DML triggers are, how they enforce data integrity, the different types of trigger that are available to you, and how to define them in your database.

Lessons

- Designing DML Triggers
- Implementing DML Triggers

- Advanced Trigger Concepts

Lab: Responding to Data Manipulation by Using Triggers

- Create and Test the Audit Trigger
- Improve the Audit Trigger

After completing this module, you will be able to:

- Design DML triggers
- Implement DML triggers
- Explain advanced DML trigger concepts, such as nesting and recursion

Module 12: Using In-Memory Tables

Microsoft SQL Server 2014 data management software introduced in-memory online transaction processing (OLTP) functionality features to improve the performance of OLTP workloads. SQL Server 2016 adds several enhancements, such as the ability to alter a memory-optimized table without recreating it. Memory-optimized tables are primarily stored in memory, which provides the improved performance by reducing hard disk access. Natively compiled stored procedures further improve performance over traditional interpreted Transact-SQL.

Lessons

- Memory-Optimized Tables
- Natively Compiled Stored Procedures

Lab: Using In-Memory Database Capabilities

- Using Memory-Optimized Tables
- Using Natively Compiled Stored procedures

After completing this module, you will be able to:

- Use memory-optimized tables to improve performance for latch-bound workloads
- Use natively compiled stored procedures

Module 13: Implementing Managed Code in SQL Server

As a SQL Server professional, you are likely to be asked to create databases that meet business needs. Most requirements can be met using Transact-SQL. However, occasionally you may need additional capabilities that can only be met by using common language runtime (CLR) code. As functionality is added to SQL Server with each new release, the necessity to use managed code decreases. However, there are times when you might need to create aggregates, stored procedures, triggers, user-defined functions, or user-defined types. You can use any .NET Framework language to develop these objects. In this module, you will learn how to use CLR managed code to create user-defined database objects for

SQL Server.

Lessons

- Introduction to CLR Integration in SQL Server
- Implementing and Publishing CLR Assemblies

Lab: Implementing Managed Code in SQL Server

- Assessing Proposed CLR Code
- Creating a Scalar-Valued CLR Function
- Creating a Table Valued CLR Function

After completing this module, you will be able to:

- Explain the importance of CLR integration in SQL Server
- Implement and publish CLR assemblies using SQL Server Data Tools (SSDT)

Module 14: Storing and Querying XML Data in SQL Server

XML provides rules for encoding documents in a machine-readable form. It has become a widely adopted standard for representing data structures, rather than sending unstructured documents. Servers that are running Microsoft SQL Server data management software often need to use XML to interchange data with other systems; many SQL Server tools provide an XML-based interface. SQL Server offers extensive handling of XML, both for storage and querying. This module introduces XML, shows how to store XML data within SQL Server, and shows how to query the XML data. The ability to query XML data directly avoids the need to extract data into a relational format before executing Structured Query Language (SQL) queries. To effectively process XML, you need to be able to query XML data in several ways: returning existing relational data as XML, and querying data that is already XML.

Lessons

- Introduction to XML and XML Schemas
- Storing XML Data and Schemas in SQL Server
- Implementing the XML Data Type
- Using the Transact-SQL FOR XML Statement
- Getting Started with XQuery
- Shredding XML

Lab: Storing and Querying XML Data in SQL Server

- Determining when to use XML
- Testing XML Data Storage in Variables
- Using XML Schemas
- Using FOR XML Queries
- Creating a Stored Procedure to Return XML

After completing this module, you will be able to:

- Describe XML and XML schemas
- Store XML data and associated XML schemas in SQL Server
- Implement XML indexes within SQL Server
- Use the Transact-SQL FOR XML statement
- Work with basic XQuery queries

Module 15: Storing and Querying Spatial Data in SQL Server

This module describes spatial data and how this data can be implemented within SQL Server.

Lessons

- Introduction to Spatial Data
- Working with SQL Server Spatial Data Types
- Using Spatial Data in Applications

Lab: Working with SQL Server Spatial Data

- Become Familiar with the Geometry Data Type
- Add Spatial Data to an Existing Table
- Find Nearby Locations

After completing this module, you will be able to:

- Describe how spatial data can be stored in SQL Server
- Use basic methods of the GEOMETRY and GEOGRAPHY data types
- Query databases containing spatial data

Module 16: Storing and Querying BLOBs and Text Documents in SQL Server

Traditionally, databases have been used to store information in the form of simple values—such as integers, dates, and strings—that contrast with more complex data formats, such as documents, spreadsheets, image files, and video files. As the systems that databases support have become more complex, administrators have found it necessary to integrate this more complex file data with the structured data in database tables. For example, in a product database, it can be helpful to associate a product record with the service manual or instructional videos for that product. SQL Server provides several ways to integrate these files—that are often known as Binary Large Objects (BLOBs)—and enable their content to be indexed and included in search results. In this module, you will learn how to design and optimize a database that includes BLOBs.

Lessons

- Considerations for BLOB Data
- Working with FILESTREAM

- Using Full-Text Search

Lab: Storing and Querying BLOBs and Text Documents in SQL Server

- Enabling and Using FILESTREAM Columns
- Enabling and Using File Tables
- Using a Full-Text Index

After completing this module, you will be able to:

- Describe the considerations for designing databases that incorporate BLOB data
- Describe the benefits and design considerations for using FILESTREAM to store BLOB data on a Windows file system
- Describe the benefits of using full-text indexing and Semantic Search, and explain how to use these features to search SQL Server data, including unstructured data

Module 17: SQL Server Concurrency

This module explains how to name, declare, assign values to, and use variables. It also describes how to store data in an array. Concurrency control is a critical feature of multiuser database systems; it allows data to remain consistent when many users are modifying data at the same time. This module covers the implementation of concurrency in Microsoft SQL Server. You will learn about how SQL Server implements concurrency controls, and the different ways you can configure and work with concurrency settings.

Lessons

- Concurrency and Transactions
- Locking Internals

Lab: SQL Server Concurrency

- Implement Snapshot Isolation
- Implement Partition Level Locking

After completing this module, you will be able to:

- Describe concurrency and transactions in SQL Server
- Describe SQL Server locking

Module 18: Performance and Monitoring

This module explains how to name, declare, assign values to, and use variables. It also describes how to store data in an array. This module looks at how to measure and monitor the performance of your SQL Server databases. The first two lessons look at SQL Server Extended Events, a flexible, lightweight event-handling system built into the Microsoft SQL Server Database Engine. These lessons focus on the

architectural concepts, troubleshooting strategies and usage scenarios.

Lessons

- Extended Events
- Working with extended Events
- Live Query Statistics
- Optimize Database File Configuration
- Metrics

Lab: Monitoring, Tracing, and Baselining

- Collecting and Analyzing Data Using Extended Events
- Implementing Baseline Methodology

After completing this module, you will be able to:

- Understand Extended Events and how to use them
- Work with Extended Events
- Understand Live Query Statistics
- Optimize the file configuration of your databases
- Use DMVs and Performance Monitor to create baselines and gather performance metrics