

Course Outline | MS SQL Server 2016/17 Programming

5 day(s)

This 5 day course is intended for students new to querying and programming with Microsoft SQL Server products who need a total immersion in the subject on a SQL Server 2016 or 2017 platform. This is a very comprehensive and intensive course with plenty of illustrated examples and augmented with practical hands-on exercises to enhance the learning experience.

Additionally, with the course structure being fully modularised, customised versions of this course can also be devised and delivered to suit individual requirements. In fact, specific content from this course could also be combined with material from other related courses to produce a bespoke training package – just ask for details.

Prerequisites

Students should be familiar with basic programming concepts and also understand the fundamental design of relational databases including, but not restricted to, the purpose of primary and foreign keys, entity relationships, and data normalization, such as can be found in the following course.

- SQL Server 2016 and 2017 Relational Database Design and Querying Fundamentals

Course Objectives

Upon successful completion of this course, students will be able to:

- Identify the Primary Components of a Typical SQL Server 2016 or 2017 Installation
- Identify the Tools Available in SQL Server 2016 or 2017
- Use SQL Server Management Studio (SSMS)
- Understand the Components of Transact-SQL (T-SQL)
- Retrieve, Filter and Sort Data
- Work with the Native SQL Server Data Types
- Query Data from Multiple Tables Using Joins
- Summarize and Group Data
- Use Window Functions for Ranking and Aggregating Data
- Combine and Limit Result Sets
- Work with Subqueries
- Use Crosstab Queries
- Use Common Table Expressions
- Work with SQL Server Date and Time Data
- Modify Data with INSERT, DELETE and UPDATE
- Merge Data with the MERGE Statement
- Understand and Implement Transactions
- Understand SQL Server Locks
- Understand SQL Server Indexes
- Understand Query Performance Issues
- Create and Manage Views
- Control the Flow of Program Execution
- Create and Implement Stored Procedures
- Handle Errors and Exceptions
- Create and Implement Triggers

Course Outline

Introduction to SQL Server Management Studio Introduction to SQL Server Integration Services

- Identify the Tools and Components of SQL Server
- Identifying the Elements of SSMS
- Objects in a SQL Server Database
- Database Diagrams
- Components of a SQL Server Database Table
- The Querying Environment

Course Outline | MS SQL Server 2016/17 Programming

- Creating a Script
- Adding Comments to your Code

Introduction to SQL and Transact-SQL

- ANSI-SQL and Transact-SQL (T-SQL)
- Categories of SQL Statements

Using the SELECT Statement to Retrieve Data

- Examine the Basic Syntax of the SELECT Statement
- Using String Literals and Expressions
- Retrieving Columns of Data from a Table
- Using Aliases for Column Names

Filtering Rows of Data with the WHERE Clause

- How to Use the WHERE Clause
- Overview of Operators
- Filter Data by Using Comparison Operators
- Filter Data by Using String Comparisons
- Filter Data by Using Logical Operators
- Filter Data Using a Range of Values
- Filter Data Using a List of Values
- Filter Data with NULL Values
- Filtering Date and Time Data

Formatting and Sorting Result Sets

- Sorting Data Using the ORDER BY Clause
- Eliminating Duplicate Rows with DISTINCT
- Introducing Native SQL Server Functions
- Getting Help on Functions
- Functions Relating to NULL Values
- Using the CASE Expression

Working with SQL Server Data Types

- Native SQL Server Data Types
- Data Type Precedence
- Implicit Data Type Conversions
- Explicit Data Type Conversion Functions
- Working with Variables
- Using Batches

Querying Data from Multiple Tables Using Joins

- Introduction to Joins
- Using Inner Joins
- Using Outer Joins
- Using Cross Joins
- Using Aliases for Table Names
- Joining More than Two Tables
- Joining a Table to Itself (Self Join)
- Using Non-Equi Joins

Summarizing Data Using Aggregate Functions

- Using Aggregate Functions Native to SQL Server
- Using Aggregate Functions with NULL Values
- Grouping Summarized Data with GROUP BY

Course Outline | MS SQL Server 2016/17 Programming

- Filtering Grouped Data Using the HAVING Clause
- Using the ROLLUP and CUBE Operators
- Using GROUPING SETS

Window Functions

- Aggregate Window Functions
- Ranking Functions
- Analytic Window Functions

Combining and Limiting Result Sets

- Combining Result Sets Using the UNION Operator
- Using the EXCEPT and INTERSECT Operators
- Using UNION, EXCEPT, and INTERSECT Together
- Limiting Result Sets Using the TOP Operator
- Using OFFSET/FETCH NEXT Filters

Working with Subqueries

- Introduction to Subqueries
- Using Subqueries as Expressions and Derived Tables
- Using Correlated Subqueries
- Using EXISTS with Correlated Subqueries
- Using the APPLY Operator

Temporary Storage

- Table Variables
- Temporary Tables

Crosstab Queries

- Using the PIVOT Operator
- Using the UNPIVOT Operator

Common Table Expressions

- Introduction to Common Table Expressions
- Using Common Table Expressions
- Recursive Queries Using Common Table Expressions

SQL Server Date and Time Data Types

- Understanding SQL Server Date and Time Data Types
- Querying and Modifying Date and Time Data
- Native SQL Server Date and Time Functions
- Guidelines and Techniques for Querying Dates

Inserting Data into Tables

- Inserting a Single Row into a Table
- Inserting Multiple Rows into a Table
- Inserting Values into Identity Columns
- The SEQUENCE Object
- Differentiating Various INSERT Statements
- Using the OUTPUT Clause with INSERT

Deleting Data from Tables

- Deleting Rows from a Table
- Truncating a Table

Course Outline | MS SQL Server 2016/17 Programming

- Deleting Rows Based on Data in Other Tables
- Using the OUTPUT Clause with DELETE

Updating Data in Tables

- Updating Rows in a Table
- Updating Rows Based on Data in Other Tables
- Using the OUTPUT Clause with UPDATE

Merging Data

- The MERGE Statement
- Practical Examples of Using MERGE

Transactions and Locking

- What Is a Transaction?
- How SQL Server Modifies Data in Tables
- Managing Transactions
- Nested Transactions
- Transaction Isolation Levels
- Delayed Durability
- SQL Server Locking Architecture
- Managing Locks

SQL Server Indexes

- Understanding SQL Server Index Architecture
- Creating and Using Nonclustered Indexes
- Creating and Using Clustered Indexes
- Filtered Indexes

Query Performance Considerations

- How SQL Server Processes T-SQL Queries
- Examining Execution Plans
- Dynamically Generating T-SQL Code
- Writing Efficient Search Arguments

Implementing Views

- Overview of Views
- Creating and Modifying a View
- Guidelines for Creating Views
- Examining the Impact of Using SELECT * in Views
- Restrictions for Modifying Data by Using Views
- Indexed Views
- Partitioned Views

Controlling Program Execution

- Control-of-Flow Language Statements
- Logical Functions

Implementing Stored Procedures

- Overview of Stored Procedures
- How Stored Procedures Are Executed by SQL Server
- Creating and Using a Stored Procedure
- Parameterising Stored Procedures
- Returning Values from a Stored Procedure
- Using Table Valued Parameters (TVPs)

Course Outline | MS SQL Server 2016/17 Programming

- Controlling Execution Context
- Parameter Sniffing

Error Handling

- Using @@ERROR
- Using RAISERROR
- Using TRY...CATCH
- Using the THROW Statement

Implementing User Defined Functions

- Overview of User Defined Functions (UDFs)
- Creating and Modifying Scalar UDFs
- Creating and Modifying Table-Valued UDFs
- Guidelines for Implementing UDFs

Implementing Triggers

- Overview of Triggers
- How Triggers Work
- AFTER Triggers
- INSTEAD OF Triggers
- DDL Triggers

Appendix 1. Querying Metadata

- Understanding Metadata
- Querying Metadata by Using Views
- Querying Metadata by Using Stored Procedures
- Querying Metadata by Using Functions
- Metadata Discovery in SQL Server

Appendix 2. Distributed Queries

- Ad Hoc Distributed Queries
- Linked Servers
- Distributed Queries Against Linked Servers