

Microsoft SQL 2016 Database Development

**Bootcamp Title – MCSA: SQL 2016 Database Development**   
Number of Days – 5

Course Description:

The MCSA SQL 2016 Database Development certification boot camp is a 5-day comprehensive deep dive into the SQL Server covering topics such as planning, monitoring, and configuring. This instructor led face to face training camp will teach you the skills needed to support a SQL Server environment.

|  |
| --- |
|  |
| **Course Outline**  **Module 1: Introduction to Microsoft SQL Server**  This module introduces SQL Server, the versions of SQL Server, including cloud versions, and how to connect to  SQL Server using SQL Server Management Studio.  **Lessons**   * The Basic Architecture of SQL Server * SQL Server Editions and Versions * Getting Started with SQL Server Management Studio   Lab : Working with SQL Server Tools   * Working with SQL Server Management Studio * Creating and Organizing T-SQL Scripts * Using Books Online   After completing this module, you will be able to:   * Describe relational databases and Transact-SQL queries. * Describe the on-premise and cloud-based editions and versions of SQL Server. * Describe how to use SQL Server Management Studio (SSMS) to connect to an instance of SQL Server, explore   the databases contained in the instance, and work with script files that contain T-SQL queries.  **Module 2: Introduction to T-SQL Querying**  This module describes the elements of T-SQL and their role in writing queries. Describe the use of sets in SQL  Server. Describe the use of predicate logic in SQL Server. Describe the logical order of operations in SELECT  statements.  **Lessons**   * Introducing T-SQL * Understanding Sets * Understanding Predicate Logic * Understanding the Logical Order of Operations in SELECT statements   Lab : Introduction to T-SQL Querying   * Executing Basic SELECT Statements * Executing Queries that Filter Data using Predicates * Executing Queries That Sort Data Using ORDER BY   After completing this module, you will be able to:   * Describe the role of T-SQL in writing SELECT statements. * Describe the elements of the T-SQL language and which elements will be useful in writing queries. * Describe the concepts of the set theory, one of the mathematical underpinnings of relational databases, and to   help you apply it to how you think about querying SQL Server   * Describe predicate logic and examine its application to querying SQL Server. * Explain the elements of a SELECT statement, delineate the order in which the elements are evaluated, and then apply this understanding to a practical approach to writing queries.   **Module 3: Writing SELECT Queries**  This module introduces the fundamentals of the SELECT statement, focusing on queries against a single table.  **Lessons**   * Writing Simple SELECT Statements * Eliminating Duplicates with DISTINCT * Using Column and Table Aliases * Writing Simple CASE Expressions   Lab : Writing Basic SELECT Statements   * Writing Simple SELECT Statements * Eliminating Duplicates Using DISTINCT * Using Column and Table Aliases * Using a Simple CASE Expression   After completing this module, you will be able to:   * Describe the structure and format of the SELECT statement, as well as enhancements that will add functionality   and readability to your queries   * Describe how to eliminate duplicates using the DISTINCT clause * Describe the use of column and table aliases * Understand and use CASE expressions   **Module 4: Querying Multiple Tables**  This module describes how to write queries that combine data from multiple sources in Microsoft SQL Server.  **Lessons**   * Understanding Joins * Querying with Inner Joins * Querying with Outer Joins * Querying with Cross Joins and Self Joins   Lab : Querying Multiple Tables   * Writing Queries that use Inner Joins * Writing Queries that use Multiple-Table Inner Joins * Writing Queries that use Self-Joins * Writing Queries that use Outer Joins * Writing Queries that use Cross Joins   After completing this module, you will be able to:   * Explain the fundamentals of joins in SQL Server * Write inner join queries * Write queries that use outer joins * Use additional join types   **Module 5: Sorting and Filtering Data**  This module describes how to implement sorting and filtering.  **Lessons**   * Sorting Data * Filtering Data with Predicates * Filtering Data with TOP and OFFSET-FETCH * Working with Unknown Values   Lab : Sorting and Filtering Data   * Writing Queries that Filter Data using a WHERE Clause * Writing Queries that Sort Data Using an ORDER BY Clause * Writing Queries that Filter Data Using the TOP Option * Write Queries that filter data using the OFFSET-FETCH clause   After completing this module, you will be able to:   * Explain how to add an ORDER BY clause to your queries to control the order of rows displayed in your query's output * Explain how to construct WHERE clauses to filter out rows that do not match the predicate. * Explain how to limit ranges of rows in the SELECT clause using a TOP option. * Explain how to limit ranges of rows using the OFFSET-FETCH option of an ORDER BY clause. * Explain how three-valued logic accounts for unknown and missing values, how SQL Server uses NULL to mark missing values, and how to test for NULL in your queries.   .  **Module 6: Working with SQL Server Data Types**  This module introduces the data types SQL Server uses to store data.  **Lessons**   * Introducing SQL Server Data Types * Working with Character Data * Working with Date and Time Data   Lab : Working with SQL Server Data Types   * Writing Queries that Return Date and Time Data * Writing Queries that use Date and Time Functions * Writing Queries That Return Character Data * Writing Queries That Return Character Functions   After completing this module, you will be able to:   * Explore many of the data types SQL Server uses to store data and how data types are converted between types * Explain the SQL Server character-based data types, how character comparisons work, and some common   functions you may find useful in your queries   * Describe data types that are used to store temporal data, how to enter dates and times so they will be properly   parsed by SQL Server, and how to manipulate dates and times with built-in functions.  **Module 7: Using DML to Modify Data**  This module describes how to create DML queries, and why you would want to.  **Lessons**   * Adding Data to Tables * Modifying and Removing Data * Generating automatic column values   Lab : Using DML to Modify Data   * Inserting Records with DML * Updating and Deleting Records Using DML   After completing this module, you will be able to:   * Use INSERT and SELECT INTO statements * Use UPDATE, MERGE, DELETE, and TRUNCATE.   **Module 8: Using Built-In Functions**  This module introduces some of the many built in functions in SQL Server.  **Lessons**   * Writing Queries with Built-In Functions * Using Conversion Functions * Using Logical Functions * Using Functions to Work with NULL   **Lab : Using Built-In Functions**   * Writing Queries That Use Conversion Functions * Writing Queries that use Logical Functions * Writing Queries that Test for Nullability   After completing this module, you will be able to:   * Describe the types of functions provided by SQL Server, and then focus on working with scalar functions * Explain how to explicitly convert data between types using several SQL Server functions * Describe how to use logical functions that evaluate an expression and return a scalar result. * Describe additional functions for working with NULL   **Module 9: Grouping and Aggregating Data**  This module describes how to use aggregate functions.  **Lessons**   * Using Aggregate Functions * Using the GROUP BY Clause * Filtering Groups with HAVING   **Lab : Grouping and Aggregating Data**   * Writing Queries That Use the GROUP BY Clause * Writing Queries that Use Aggregate Functions * Writing Queries that Use Distinct Aggregate Functions * Writing Queries that Filter Groups with the HAVING Clause   After completing this module, you will be able to:   * Describe the built-in aggregate function in SQL Server and write queries using it. * Write queries that separate rows using the GROUP BY clause. * Write queries that use the HAVING clause to filter groups.   Module 10: Using Subqueries  This module describes several types of subquery and how and when to use them.  Lessons   * Writing Self-Contained Subqueries * Writing Correlated Subqueries * Using the EXISTS Predicate with Subqueries   **Lab : Using Subqueries**   * Writing Queries That Use Self-Contained Subqueries * Writing Queries That Use Scalar and Multi-Result Subqueries * Writing Queries That Use Correlated Subqueries and an EXISTS Clause   After completing this module, you will be able to:   * Describe where subqueries may be used in a SELECT statement. * Write queries that use correlated subqueries in a SELECT statement * Write queries that use EXISTS predicates in a WHERE clause to test for the existence of qualifying rows * Use the EXISTS predicate to efficiently check for the existence of rows in a subquery.   **Module 11: Using Table Expressions**  Previously in this course, you learned about using subqueries as an expression that returned results to an outer  calling query. Like subqueries, table expressions are query expressions, but table expressions extend this idea by allowing you to name them and to work with their results as you would work with data in any valid relational table. Microsoft SQL Server supports four types of table expressions: derived tables, common table expression (CTEs),  views, and inline table-valued functions (TVFs). In this module, you will learn to work with these forms of table expressions and learn how to use them to help create a modular approach to writing queries.  **Lessons**   * Using Views * Using Inline Table-Valued Functions * Using Derived Tables * Using Common Table Expressions   **Lab : Using Table Expressions**   * Writing Queries That Use Views * Writing Queries That Use Derived Tables * Writing Queries That Use Common Table Expressions (CTEs) * Writing Queries That Use Inline Table-Valued Expressions (TVFs)   After completing this module, you will be able to:   * Write queries that return results from views. * Use the CREATE FUNCTION statement to create simple inline TVFs. * Write queries that create and retrieve results from derived tables. * Write queries that create CTEs and return results from the table expression.   **Module 12: Using Set Operators**  This module introduces how to use the set operators UNION, INTERSECT, and EXCEPT to compare rows between two input sets.  **Lessons**   * Writing Queries with the UNION operator * Using EXCEPT and INTERSECT * Using APPLY   **Lab : Using Set Operators**   * Writing Queries That Use UNION Set Operators and UNION ALL * Writing Queries That Use CROSS APPLY and OUTER APPLY Operators * Writing Queries That Use the EXCEPT and INTERSECT Operators   After completing this module, students will be able to:   * Write queries that use UNION to combine input sets. * Write queries that use UNION ALL to combine input sets * Write queries that use the EXCEPT operator to return only rows in one set but not another. * Write queries that use the INTERSECT operator to return only rows that are present in both sets * Write queries using the CROSS APPLY operator. * Write queries using the OUTER APPLY operator   **Module 13: Using Windows Ranking, Offset, and Aggregate Functions**  This module describes the benefits to using window functions. Restrict window functions to rows defined in an  OVER clause, including partitions and frames. Write queries that use window functions to operate on a window of  rows and return ranking, aggregation, and offset comparison results.  **Lessons**   * Creating Windows with OVER * Exploring Window Functions   **Lab : Using Windows Ranking, Offset, and Aggregate Functions**   * Writing Queries that use Ranking Functions * Writing Queries that use Offset Functions * Writing Queries that use Window Aggregate Functions   After completing this module, students will be able to:   * Describe the T-SQL components used to define windows, and the relationships between them. * Write queries that use the OVER clause, with partitioning, ordering, and framing to define windows * Write queries that use window aggregate functions. * Write queries that use window ranking functions. * Write queries that use window offset functions   **Module 14: Pivoting and Grouping Sets**  This module describes write queries that pivot and unpivot result sets. Write queries that specify multiple groupings with grouping sets  **Lessons**   * Writing Queries with PIVOT and UNPIVOT * Working with Grouping Sets * **Lab : Pivoting and Grouping Sets** * Writing Queries that use the PIVOT Operator * Writing Queries that use the UNPIVOT Operator * Writing Queries that use the GROUPING SETS CUBE and ROLLUP Subclauses   After completing this module, students will be able to:   * Describe how pivoting data can be used in T-SQL queries. * Write queries that pivot data from rows to columns using the PIVOT operator. * Write queries that unpivot data from columns back to rows using the UNPIVOT operator. * Write queries using the GROUPING SETS subclause. * Write queries that use ROLLUP AND CUBE. * Write queries that use the GROUPING\_ID function.   This module describes how to return results by executing stored procedures. Pass parameters to procedures. Create simple stored procedures that encapsulate a SELECT statement. Construct and execute dynamic SQL with EXEC  and sp\_executesql.  **Lessons**   * Querying Data with Stored Procedures * Passing Parameters to Stored procedures * Creating Simple Stored Procedures * Working with Dynamic SQL   **Lab : Executing Stored Procedures**   * Using the EXECUTE statement to Invoke Stored Procedures * Passing Parameters to Stored procedures * Executing System Stored Procedures   After completing this module, students will be able to:   * Describe stored procedures and their use. * Write T-SQL statements that execute stored procedures to return data. * Write EXECUTE statements that pass input parameters to stored procedures. * Write T-SQL batches that prepare output parameters and execute stored procedures. * Use the CREATE PROCEDURE statement to write a stored procedure. * Create a stored procedure that accepts input parameters. * Describe how T-SQL can be dynamically constructed. * Write queries that use dynamic SQL.   **Module 16: Programming with T-SQL**  This module describes how to enhance your T-SQL code with programming elements.  **Lessons**   * T-SQL Programming Elements * Controlling Program Flow   **Lab : Programming with T-SQL**   * Declaring Variables and Delimiting Batches * Using Control-Of-Flow Elements * Using Variables in a Dynamic SQL Statement * Using Synonyms   After completing this module, students will be able to:   * Describe how Microsoft SQL Server treats collections of statements as batches. * Create and submit batches of T-SQL code for execution by SQL Server. * Describe how SQL Server stores temporary objects as variables. * Write code that declares and assigns variables. * Create and invoke synonyms * Describe the control-of-flow elements in T-SQL. * Write T-SQL code using IF...ELSE blocks. * Write T-SQL code that uses WHILE.   **Module 17: Implementing Error Handling**  This module introduces error handling for T-SQL.  **Lessons**   * Implementing T-SQL error handling * Implementing structured exception handling   **Lab : Implementing Error Handling**   * Redirecting errors with TRY/CATCH * Using THROW to pass an error message back to a client   After completing this module, students will be able to:   * Implement T-SQL error handling. * Implement structured exception handling.   **Module 18: Implementing Transactions**  This module describes how to implement transactions.  **Lessons**   * Transactions and the database engines * Controlling transactions   **Lab : Implementing Transactions**   * Controlling transactions with BEGIN, COMMIT, and ROLLBACK * Adding error handling to a CATCH block   After completing this module, students will be able to:   * Describe transactions and the differences between batches and transactions. * Describe batches and how they are handled by SQL Server. * Create and manage transactions with transaction control language (TCL) statements. * Use SET XACT\_ABORT to define SQL Servers handling of transactions outside TRY/CATCH blocks. |
|  |

**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 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 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.