

Querying data with transact sql :

(24 Hours)

Exam 70-761

Prerequisites

- Working knowledge of relational databases.
- Basic knowledge of the Microsoft Windows operating system and its core functionality.

Course contents :

Module 1: Introduction to Microsoft SQL Server 2016

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 2016 Tools

After completing this module, you will be able to:

- Describe the architecture and editions of SQL Server 2012.
- Work with SQL Server Management Studio.

Module 2: Introduction to T-SQL Querying

This module introduces the elements of T-SQL and their role in writing queries, describes the use of sets in SQL Server, describes the use of predicate logic in SQL Server, and describes 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 Transact-SQL Querying

- After completing this module, you will be able to:
- Describe 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

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

After completing this module, you will be able to:

- Write simple SELECT statements.
- Eliminate duplicates using the DISTINCT clause.
- Use column and table aliases.
- Write simple CASE expressions.

Module 4: Querying Multiple Tables

This module explains how to write queries which combine data from multiple sources in SQL Server. The module introduces the use of JOINS in T-SQL queries as a mechanism for retrieving data from multiple tables.

Lessons

- Understanding Joins
- Querying with Inner Joins
- Querying with Outer Joins
- Querying with Cross Joins and Self Joins

Lab : Querying Multiple Tables

After completing this module, you will be able to:

- Describe how multiple tables may be queried in a SELECT statement using joins.
- Write queries that use inner joins.
- Write queries that use outer joins.
- Write queries that use self-joins and cross joins.

Module 5: Sorting and Filtering Data

This module explains how to enhance queries to limit the rows they return, and to control the order in which the rows are displayed. The module also discusses how to resolve missing and unknown results.

Lessons

- Sorting Data
- Filtering Data with Predicates
- Filtering with the TOP and OFFSET-FETCH Options
- Working with Unknown Values

Lab : Sorting and Filtering Data

After completing this module, you will be able to:

- Filter data with predicates in the WHERE clause.
- Sort data using ORDER BY.
- Filter data in the SELECT clause with TOP.
- Filter data with OFFSET and FETCH.

Module 6: Working with SQL Server 2016 Data Types

This module explains the data types SQL Server uses to store data. It introduces the many types of numeric and special-use data types. It also explains conversions between data types, and the importance of type precedence.

Lessons

- Introducing SQL Server 2016 Data Types
- Working with Character Data
- Working with Date and Time Data

Lab : Working with SQL Server 2016 Data Types

After completing this module, you will be able to:

- Describe numeric data types, type precedence and type conversions.
- Write queries using character data types.
- Write queries using date and time data types.

Module 7: Using DML to Modify Data This module describes the use of Transact-SQL Data Manipulation Language to perform inserts, updates, and deletes to your data.

Lessons

- Inserting Data
- Modifying and Deleting Data

Lab : Using DML to Modify Data

After completing this module, you will be able to:

- Insert new data into your tables.
- Update and delete existing records in your tables

Module 8: Using Built-In Functions

This module introduces the use of functions that are built in to SQL Server Denali, and will discuss some common usages including data type conversion, testing for logical results and nullability.

Lessons

- Writing Queries with Built-In Functions
- Using Conversion Functions
- Using Logical Functions
- Using Functions to Work with NULL

Lab : Using Built-In Functions

After completing this module, you will be able to:

- Write queries with built-in scalar functions.
- Use conversion functions.
- Use logical functions.
- Use functions that work with NULL.

Module 9: Grouping and Aggregating Data

This module introduces methods for grouping data within a query, aggregating the grouped data and filtering groups with HAVING. The module is designed to help the student grasp why a SELECT clause has restrictions placed upon column naming in the GROUP BY clause as well as which columns may be listed in the SELECT clause.

Lessons

Using Aggregate Functions
Using the GROUP BY Clause
Filtering Groups with HAVING

Lab : Grouping and Aggregating Data

After completing this module, you will be able to:

- Write queries which summarize data using built-in aggregate functions.
- Use the GROUP BY clause to arrange rows into groups.
- Use the HAVING clause to filter out groups based on a search condition.

Module 10: Using Subqueries

This module will introduce the use of subqueries in various parts of a SELECT statement. It will include the use of scalar and multi-result subqueries, and the use of the IN and EXISTS operators.

Lessons

- Writing Self-Contained Subqueries
- Writing Correlated Subqueries
- Using the EXISTS Predicate with Subqueries

Lab : Using Subqueries

After completing this module, you will be able to:

- Describe the uses of queries which are nested within other queries.
- Write self-contained subqueries which return scalar or multi-valued results.
- Write correlated subqueries which return scalar or multi-valued results.
- Use the EXISTS predicate to efficiently check for the existence of rows in a subquery.

Module 11: Using Set Operators This module introduces 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

After completing this module, you will be able to:

- Write queries using UNION, EXCEPT, and INTERSECT operators.
- Use the APPLY operator.

Developing SQL Databases

(40 Hours)

Exam 70-762

Prerequisites

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

Course contents :

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 Baselineing

- 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