

Module 2

Introduction to T-SQL Querying

Module Overview

- Introducing T-SQL
- Understanding Sets
- Understanding Predicate Logic
- Understanding the Logical Order of Operations in SELECT Statements

Lesson 1: Introducing T-SQL

- About T-SQL
- Categories of T-SQL Statements
- T-SQL Language Elements
- T-SQL Language Elements: Predicates and Operators
- T-SQL Language Elements: Functions
- T-SQL Language Elements: Variables
- T-SQL Language Elements: Expressions
- T-SQL Language Elements: Control of Flow, Errors, and Transactions
- T-SQL Language Elements: Comments
- T-SQL Language Elements: Batch Separators
- Demonstration: T-SQL Language Elements

About T-SQL

- Structured Query Language (SQL)
 - Developed by IBM in the 1970s
 - Adopted by ANSI and ISO standards bodies
 - Widely used in the industry
 - PL/SQL (Oracle), SQL Procedural Language (IBM), Transact-SQL (Microsoft)
- Transact-SQL is commonly referred to as T-SQL
 - The querying language of SQL Server 2016
- SQL is declarative
 - Describe what you want, not the individual steps

Categories of T-SQL Statements

DML*

- Data Manipulation Language
- Used to query and manipulate data
- SELECT, INSERT, UPDATE, DELETE

DDL

- Data Definition Language
- Used to define database objects
- CREATE, ALTER, DROP

DCL

- Data Control Language
- Used to manage security permissions
- GRANT, REVOKE, DENY

*DML with SELECT is the focus of this course

T-SQL Language Elements

- Predicates and Operators
- Functions
- Variables
- Expressions
- Batch Separators
- Control of Flow
- Comments

T-SQL Language Elements: Predicates and Operators

Elements:	Predicates and Operators:
Predicates	ALL, ANY, BETWEEN, IN, LIKE, OR, SOME
Comparison Operators	=, >, <, >=, <=, <>, !=, !>, !<
Logical Operators	AND, OR, NOT
Arithmetic Operators	*, /, %, +, -, ,
Concatenation	+

T-SQL Language Elements: Functions

String Functions

- SUBSTRING
- LEFT, RIGHT
- LEN
- REPLACE
- REPLICATE
- UPPER, LOWER
- LTRIM, RTRIM
- STUFF
- SOUNDEX

Date and Time Functions

- GETDATE
- SYSDATETIME
- GETUTCDATE
- DATEADD
- DATEDIFF
- YEAR
- MONTH
- DAY
- DATENAME
- DATEPART
- ISDATE

Aggregate Functions

- SUM
- MIN
- MAX
- AVG
- COUNT
- COUNT_BIG
- STDEV
- STDEVP
- VAR

T-SQL Language Elements: Variables

- Local variables in T-SQL temporarily store a value of a specific data type
- Name begins with single @ sign
 - @@ reserved for system functions
- Assigned a data type
- Must be declared and used within the same batch
- In SQL Server 2016, you can declare and initialize a variable in the same statement

```
DECLARE @search varchar(30) = 'Match%';
```

T-SQL Language Elements: Expressions

- Combination of identifiers, values, and operators evaluated to obtain a single result
- Can be used in SELECT statements
 - SELECT clause
 - WHERE clause
- Can be single constant, single-valued function, or variable
- Can be combined if expressions have the same data type

```
SELECT YEAR(orderdate) + 1 ...  
SELECT qty * unitprice ...
```

T-SQL Language Elements: Control of Flow, Errors, and Transactions

Control of Flow

- IF ... ELSE
- WHILE
- BREAK
- CONTINUE
- BEGIN ... END
- WAITFOR

Error Handling

- TRY
- CATCH
- THROW

Transaction Control

- BEGIN TRANSACTION
- ROLLBACK TRANSACTION
- COMMIT TRANSACTION
- ROLLBACK WORK
- SAVE TRANSACTION

The above are used in programmatic code objects

T-SQL Language Elements: Comments

- Two methods for marking text as comments
 - A block comment, surround text with /* and */

```
/*  
    All the text in this paragraph will be treated as  
    comments by SQL Server.  
*/
```

- An inline comment, precede text with --

```
-- This is an inline comment
```

- Many T-SQL editors will color comments as above

T-SQL Language Elements: Batch Separators

- Batches are sets of commands sent to SQL Server as a unit
- Batches determine variable scope, name resolution
- To separate statements into batches, use a separator:
 - SQL Server tools use the GO keyword
 - GO is not an SQL Server T-SQL command
 - GO [count] executes the preceding batch [count] times

Demonstration: T-SQL Language Elements

In this demonstration, you will see how to:

- Use T-SQL language elements

Lesson 2: Understanding Sets

- Set Theory and SQL Server
- Set Theory Applied to SQL Server Queries

Set Theory and SQL Server

Characteristics of a Set	Example
Elements of a set called Members	Customer as a member of set called Customers
Elements of a set are described by attributes	First name, Last name, Age
Elements must be unique	Customer ID

Set theory does not specify the order of its members

Set Theory Applied to SQL Server Queries

Application of Set Theory	Comments
Acts on all elements at once	Query the whole table
Use set-based processing	Tell the engine what you want to retrieve
Avoid cursors or loops	Do not process each item individually
Members of a set must be unique	Define unique keys in a table
No defined order to result set	Use ORDER BY clause if results need to be ordered

Lesson 3: Understanding Predicate Logic

- Predicate Logic and SQL Server
- Predicate Logic Applied to SQL Server Queries

Predicate Logic and SQL Server

- Predicate logic is another mathematical basis for the relational database model
- In theory, a predicate is a property or expression that is either true or false
- Predicate is also referred to as a Boolean expression

Predicate Logic Applied to SQL Server Queries

Uses for Predicates

- Filtering data in queries
- Providing conditional logic to CASE expressions
- Joining tables
- Defining subqueries
- Enforcing data integrity
- Control of flow

Lesson 4: Understanding the Logical Order of Operations in SELECT Statements

- Elements of a SELECT Statement
- Logical Query Processing
- Applying the Logical Order of Operations to Writing SELECT Statements
- Demonstration: Logical Query Processing

Elements of a SELECT Statement

Element	Expression	Role
SELECT	<select list>	Defines which columns to return
FROM	<table source>	Defines table(s) to query
WHERE	<search condition>	Filters returned data using a predicate
GROUP BY	<group by list>	Arranges rows by groups
HAVING	<search condition>	Filters groups by a predicate
ORDER BY	<order by list>	Sorts the results

Logical Query Processing

5.	SELECT	<select list>
1.	FROM	<table source>
2.	WHERE	<search condition>
3.	GROUP BY	<group by list>
4.	HAVING	<search condition>
6.	ORDER BY	<order by list>

The order in which a query is written is not the order in which it is evaluated by SQL Server

Applying the Logical Order of Operations to Writing SELECT Statements

```
USE TSQL;
```

```
SELECT EmployeeId, YEAR(OrderDate) AS OrderYear  
FROM Sales.Orders  
WHERE CustomerId = 71  
GROUP BY EmployeeId, YEAR(OrderDate)  
HAVING COUNT(*) > 1  
ORDER BY EmployeeId, OrderYear;
```


Demonstration: Logical Query Processing

In this demonstration, you will see how to:

- View query output that illustrates logical processing order

Lab: Introduction to T-SQL Querying

- Exercise 1: Executing Basic SELECT Statements
- Exercise 2: Executing Queries That Filter Data Using Predicates
- Exercise 3: Executing Queries That Sort Data Using ORDER BY

Logon Information

Virtual machine: **20761C-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa55w.rd**

Estimated Time: 30 minutes

Lab Scenario

You are an Adventure Works business analyst, who will be writing reports against corporate databases stored in SQL Server. To help you become more comfortable with SQL Server querying, the Adventure Works IT department has provided some common queries to run against their databases. You will review and execute these queries.

Module Review and Takeaways

- Review Question(s)