Intermediate SQL

SCRIPTS

Scripts

- In T-SQL we have the ability to code scripts with functionality similar to the functionality provided by other programming languages like C#, Java, Python, etc.

- The scripting ability is however limited compared to other languages since T-SQL is designed to work with SQL Server Databases rather than a general-purpose programming language

Copyright Murach SQL Server 2019

Batches

```
CREATE DATABASE MyDatabase;
GO

USE MyDatabase;

CREATE TABLE MyTable (
    MyId INT NOT NULL IDENTITY PRIMARY KEY,
    MyField NVARCHAR(50) NOT NULL);

CREATE TABLE MyOtherTable (
    MyOtherId INT NOT NULL IDENTITY PRIMARY KEY,
    MyOtherField NVARCHAR(50) NOT NULL);

GO
```

- In Introduction to SQL, we learned some DDL – specifically CREATE DATABASE and CREATE TABLE
- We used the GO command to indicate the end of a batch (GO isn't actually a T-SQL statement it's just interpreted by SSMS)
- In the example to the left, we have 2 separate batches
  - One that creates the database
  - Another that creates 3 tables
- They must be coded in separate batches because the database needs to exist first before tables can be added to it.

Copyright Murach SQL Server 2019

Batches

- CREATE TABLE statements do not have to be in their own batch, but the following statements do have to be in separate batches – the must be the first and only statement in a batch

  - CREATE VIEW
  - CREATE TRIGGER
  - CREATE PROCEDURE
  - CREATE SCHEMA
  - CREATE FUNCTION

Copyright Murach SQL Server 2019

Batches

- Each script can contain one or more batches
- GO command is required to signal the end of a batch (unless it is the last one in the script)
- Statements are executed in the order that the appear in the batch, therefore you need to code statements that depend on other statements AFTER the statements they depend on

Copyright Murach SQL Server 2019

T-SQL Statements for Script Processing

Keyword	Description
USE	Changes the database context to the specified database
PRINT	Returns a message to the client
DECLARE	Declares a local variable
SET	Sets the value of a local variable or session variable
EXEC	Executes a dynamic SQL statement or stored procedure

Copyright Murach SQL Server 2019

T-SQL Statements for controlling the flow of execution

Keyword	Description
IF…ELSE	Controls the flow of execution based on a condtion
BEGIN…END	Defines a statement block
WHILE	Repeats statements while specific condition is true
BREAK	Exits the innermost WHILE LOOP
CONTINUE	Returns to the beginning of a WHILE LOOP
TRY…CATCH	Controls the flow of execution when an error occurs
GOTO	Do not use this
RETURN	Exits unconditionaly

Copyright Murach SQL Server 2019

Demos

- In the class demos we will cover the following from Chapter 14
  - Declaring scalar variables
  - Setting the value for scalar variables using the SET keyword
  - Setting the value of scalar variables using the SELECT statement
  - Declaring table variables
  - Working with temporary tables
  - Common Table Expressions (CTE)
  - Conditional Processing (IF/ELSE/ELSE IF)
  - Loops
  - Cursors
  - Try/Catch