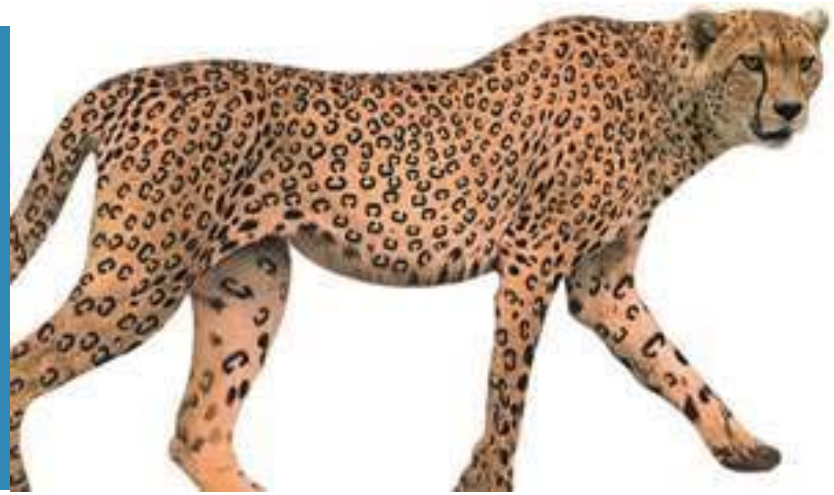
Cognizant | Academy
Passion for making a difference

SQL Server 2005

Targeted at: Entry Level Trainees



Session 19: Control Flow Constructs

Academy

About the Author

Created By:	Shashidharamurthy R L (124294)
Credential Information:	Senior Instructor, Cognizant Academy MCP, MCTS, SCJP, SCWCD
Version and Date:	SQLSERVER2005/PPT/0408/1.0

Cognizant Certified Official Curriculum

Icons Used

Questions

Tools

Hands on Exercise

Coding Standards

Test Your Understanding

Reference

Demonstration

A Welcome Break

Contacts

SQL Server 2005 Session 19: Overview

- **Introduction:**

  This session provides a brief overview about writing control flow constructs.

SQL Server 2005 Session 19: Objective

- **Objective:**

After completing this session, you will be able to:

- Work with the Control Flow Constructs as follows:
  - `If - else`
  - `Case`
  - `While`
  - `Return`
  - `WaitFor`

Control Flow Constructs

- Control-of-flow language controls the flow of execution of Transact-SQL statements, statement blocks, and stored procedures.
- Control-of-flow keywords are as follows:
  - `BEGIN...END`
  - `IF...ELSE`
  - `WHILE`
  - `CASE`
  - `BREAK`
  - `CONTINUE`
  - `RETURN`
  - `GOTO`
  - `WAITFOR`

Control Flow Constructs (Contd.)

- IF...ELSE:
  - The IF statement is applied to test for a condition
  - The Transact-SQL statement that follows an IF keyword and its condition is executed if the condition is satisfied, then the Boolean expression returns TRUE.
  - The optional ELSE keyword introduces another Transact-SQL statement that is executed when the IF condition is not satisfied then the Boolean expression returns FALSE.

```
IF Boolean_expression
 { sql_statement | statement_block }
 [ ELSE
     { sql_statement | statement_block } ]
```

Control Flow Constructs (Contd.)

- WHILE:
  - Sets a condition for the repeated execution of an SQL statement or statement block.
  - The statements are executed repeatedly as long as the specified condition is true.
  - The execution of statements in the WHILE loop can be controlled from inside the loop with the BREAK and CONTINUE keywords.

```
WHILE Boolean_expression
{ sql_statement | statement_block }
[ BREAK ]
{ sql_statement | statement_block }
[ CONTINUE ]
 { sql_statement | statement_block }
```

Control Flow Constructs (Contd.)

- CASE:
  - » Evaluates a list of conditions and returns one of multiple possible result expressions.
  - » CASE has two formats:
    - The simple CASE function compares an expression to a set of simple expressions to determine the result.
    - The searched CASE function evaluates a set of Boolean expressions to determine the result.

Simple CASE function:

```
CASE input_expression
     WHEN when_expression THEN result_expression
    [ ...n ]
     [
    ELSE else_result_expression
     ]
END
```

Searched CASE function:

```
CASE
     WHEN Boolean_expression THEN result_expression
    [ ...n ]
     [
    ELSE else_result_expression
     ]
END
```

Control Flow Constructs (Contd.)

- CASE:
  - Using a SELECT statement with a simple CASE function:

```
USE AdventureWorks;
GO
SELECT   ProductNumber, Category =
      CASE ProductLine
         WHEN 'R' THEN 'Road'
         WHEN 'M' THEN 'Mountain'
         WHEN 'T' THEN 'Touring'
         WHEN 'S' THEN 'Other sale items'
         ELSE 'Not for sale'
      END,
   Name
FROM Production.Product
ORDER BY ProductNumber;
GO
```

Control Flow Constructs (Contd.)

- CASE:
  - Using a SELECT statement with simple and searched CASE function

```
USE AdventureWorks;
GO
SELECT   ProductNumber, Name, 'Price Range' =
      CASE
         WHEN ListPrice =  0 THEN 'Mfg item - not for resale'
         WHEN ListPrice < 50 THEN 'nder $50'
         WHEN ListPrice >= 50 and ListPrice < 250 THEN 'Under $250'
         WHEN ListPrice >= 250 and ListPrice < 1000 THEN 'Under $1000'
         ELSE 'Over $1000'
      END
FROM Production.Product
ORDER BY ProductNumber ;
GO
```

Control Flow Constructs (Contd.)

Simple WHILE construct with BREAK and CONTINUE keyword.

```
WHILE Boolean_expression
{ sql_statement | statement_block }
[ BREAK ]
{ sql_statement | statement_block }
[ CONTINUE ]
 { sql_statement | statement_block }
```

Note:
BREAK: Exits the innermost loop in a WHILE or IF…ELSE statement
CONTINUE: Restarts a WHILE loop. Any statements after the CONTINUE keyword are ignored

Control Flow Constructs (Contd.)

- RETURN:
  - Exits unconditionally from a query or procedure.
  - RETURN is immediate and complete and can be used at any point to exit from a procedure, batch or statement block.
  - Statements that follow RETURN are not executed.

```
RETURN [ integer_expression ]
```

Control Flow Constructs (Contd.)

- WAITFOR:
  - Blocks the execution of a batch, stored procedure or transaction until a specified time or time interval is reached.

```
WAITFOR
{
    DELAY ‘time_to_pass’
  | TIME ‘time_to_execute’
  | ( receive_statement ) [ , TIMEOUT timeout ]
}
```

Q & A

- Allow time for questions from participants

Try it Out

Problem Statement:

- Consider `ProductLine` table in AdventureWorks Database uses the `CASE` function to change the display of product line categories to make them more understandable.

Try it Out (Contd.)

Code:

```
USE AdventureWorks;
GO
SELECT   ProductNumber, Category =
      CASE ProductLine
         WHEN 'R' THEN 'Road'
         WHEN 'M' THEN 'Mountain'
         WHEN 'T' THEN 'Touring'
         WHEN 'S' THEN 'Other sale items'
         ELSE 'Not for sale'
      END,
   Name
FROM Production.Product
ORDER BY ProductNumber;
GO
```

Refer file Session_19_SQL_Server_2005_Presentation.sql for the soft copy of the code

Try it Out (Contd.)

How it Works:

- When script is executed it uses the CASE function to change the display of product line categories to make them more understandable.

Q & A

- Allow time for questions from participants

Test Your Understanding

1. The construct which evaluates a list of conditions and returns one of multiple possible result expressions.

   a. CASE

   b. IF..ELSE

   c. WHILE

   d. DO..WHILE

SQL Server 2005 Session 19: Summary

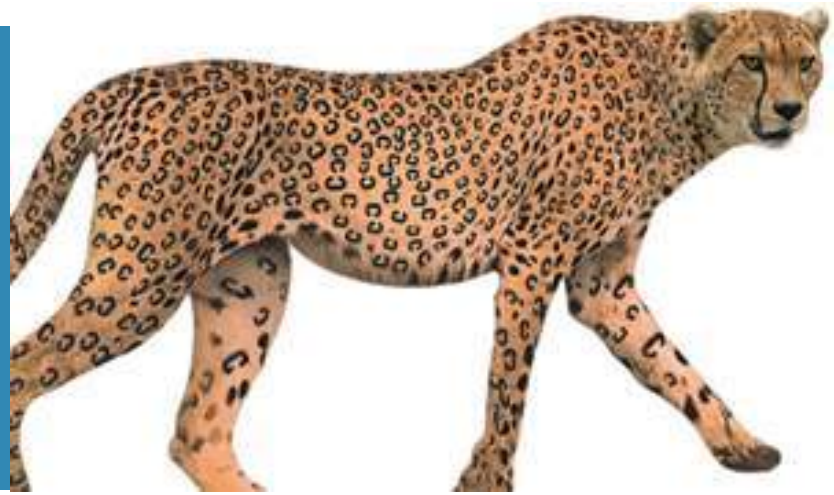
- CASE has two formats. Both formats support an optional ELSE argument:
  - » The simple CASE function compares an expression to a set of simple expressions to determine the result.
    The searched CASE function evaluates a set of Boolean expressions to determine the result.
    WHILE construct sets condition for the repeated execution of an SQL statement or statement block.
- WAITFOR blocks the execution of a batch, stored procedure or transaction until a specified time or time interval is reached.

SQL Server 2005 Session 19: Source

- SQL Server Books Online
- Microsoft Official Curriculum:
  - 2779A Implementing a Microsoft SQL Server 2005 Database

Cognizant | Academy
Passion for making a difference

You have completed the Session 19 of SQL Server 2005.



Academy