(T35)討論 IIFFunction
CourseGUID: e48417fc-9db5-4e99-822c-706c5ccef6cc
==========
(T35)討論 IIFFunction
==========

==========

0. Summary

1.
The following clauses are equivalent:
1.1.
IFF Syntax:
--IIF (boolCondition, trueValue, falseValue)
1.2.
CaseWhen Syntax :
--CASE WHEN @Grades >= 50
-- THEN 'Pass'
-- ELSE 'Fail'
--END;
1.3.
E.g.
--DECLARE @Grades INT = 50;
--DECLARE @Result1 NVARCHAR(10) = IIF(@Grades >= 50, 'Pass', 'Fail');
--DECLARE @Result2 NVARCHAR(10) = CASE WHEN @Grades >= 50 THEN 'Pass'
-- ELSE 'Fail'
-- END;

2.
The following clauses are equivalent:
2.1.
CHOOSE Syntax:
--CHOOSE(IndexValue , Value01, Value02, ...)
2.2.
--CASE @IndexValue
-- WHEN 1 THEN 'Value01'
-- WHEN 2 THEN 'Value02'
-- ...
--END;
2.3.
DECLARE @Grades INT = 4;
--DECLARE @Result1 NVARCHAR(10) = CHOOSE(@Grades, 'Fail 1', 'Fail 2', 'Fail 3',
-- 'Pass', 'Credit', 'Distinction',
-- 'High Distinction');
--DECLARE @Result2 NVARCHAR(10) = CASE @Grades
-- WHEN 1 THEN 'Fail 1'
-- WHEN 2 THEN 'Fail 2'
-- WHEN 3 THEN 'Fail 3'
-- WHEN 4 THEN 'Pass'
-- WHEN 5 THEN 'Credit'
-- WHEN 6 THEN 'Distinction'
-- WHEN 7 THEN 'High Distinction'
-- END;

==========

1. IIF Function

```
--============================================================
--T035_01
--The following clauses are equivalent:
DECLARE @Grades INT = 50;
DECLARE @Result1 NVARCHAR(10) = IIF(@Grades >= 50, 'Pass', 'Fail');
DECLARE @Result2 NVARCHAR(10) = CASE WHEN @Grades >= 50 THEN 'Pass'
                                     ELSE 'Fail'
                                END;
PRINT @Result1;
PRINT @Result2;
GO -- Run the previous command and begins new batch
--Return Pass
```

Messages

```
Pass
Pass
```

```
--============================================================
--T035_02
--Create sample data
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE     TABLE_NAME = 'StudentGrades' ) )
    BEGIN
        TRUNCATE TABLE dbo.StudentGrades;
        DROP TABLE StudentGrades;
    END;
GO -- Run the previous command and begins new batch
CREATE TABLE StudentGrades
    (
      Id INT IDENTITY(1, 1)
             PRIMARY KEY ,
      [Name] NVARCHAR(100) ,
      Grades INT
    );
GO -- Run the previous command and begins new batch
INSERT  INTO StudentGrades
VALUES  ( 'Name01', 50 );
INSERT  INTO StudentGrades
VALUES  ( 'Name02', 51 );
INSERT  INTO StudentGrades
VALUES  ( 'Name03', 49 );
INSERT  INTO StudentGrades
VALUES  ( 'Name04', 30 );
INSERT  INTO StudentGrades
VALUES  ( 'Name05', 75 );
INSERT  INTO StudentGrades
VALUES  ( 'Name06', 85 );
INSERT  INTO StudentGrades
VALUES  ( 'Name07', 20 );
GO -- Run the previous command and begins new batch
SELECT  *
FROM    StudentGrades;
GO -- Run the previous command and begins new batch
```

	Id	Name	Grades
1	1	Name01	50
2	2	Name02	51
3	3	Name03	49
4	4	Name04	30
5	5	Name05	75
6	6	Name06	85
7	7	Name07	20

```
--=============================================================================
--T035_03
--The following clauses are equivalent:
--T035_03_01
--CASE WHEN(condition) THEN ... ELSE ...
SELECT  [Name],
        Grades,
        CASE WHEN Grades >= 50 THEN 'Pass'
             ELSE 'Fail'
        END AS [Result]
FROM    StudentGrades;
GO -- Run the previous command and begins new batch
--T035_03_02
SELECT  [Name],
        Grades,
        IIF(Grades >= 50, 'Pass', 'Fail') AS [Result]
FROM    StudentGrades;
GO -- Run the previous command and begins new batch
```

	Name	Grades	Result
1	Name01	50	Pass
2	Name02	51	Pass
3	Name03	49	Fail
4	Name04	30	Fail
5	Name05	75	Pass
6	Name06	85	Pass
7	Name07	20	Fail

```
--=============================================================
--T035_04
--Clean up
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE     TABLE_NAME = 'StudentGrades' ) )
    BEGIN
        TRUNCATE TABLE dbo.StudentGrades;
        DROP TABLE StudentGrades;
    END;
GO -- Run the previous command and begins new batch
```