

PENDANT (Transact-SQL)

15/03/2017 • 2 minutes pour lire •  +1

In this article

[Syntax](#)





[Arguments](#)

[Remarks](#)

[Examples](#)

[Examples: Azure SQL Data Warehouse and Parallel Data Warehouse](#)


[See Also](#)


S'applique à: Base de données  SQL Server  Azure SQL  Entrepôt de données parallèle Azure SQL Data Warehouse 

Définit une condition pour l'exécution répétée d'une instruction SQL ou d'un bloc d'instructions. Les instructions sont exécutées à plusieurs reprises tant que la condition spécifiée est vraie. L'exécution des instructions de la boucle WHILE peut être contrôlée de l'intérieur à l'aide des mots-clés BREAK et CONTINUE.

 [Conventions de syntaxe Transact-SQL](#)

Syntaxe

	 Copie
<pre>-- Syntax for SQL Server and Azure SQL Database WHILE Boolean_expression { sql_statement statement_block BREAK CONTINUE }</pre>	

	 Copie
<pre>-- Syntax for Azure SQL Data Warehouse and Parallel Data Warehouse WHILE Boolean_expression { sql_statement statement_block BREAK }</pre>	

Arguments

Expression booléenne

Est une [expression](#) qui renvoie **VRAI** ou **FAUX** . Si l'expression booléenne contient une instruction SELECT, cette dernière doit être entre parenthèses.

{ sql_statement | statement_block }

S'agit-il d'une instruction ou d'un regroupement d'instructions Transact-SQL tel que défini avec un bloc d'instructions? Pour définir un bloc d'instructions, utilisez les mots-clés BEGIN et END du contrôle de flux.

BREAK

Provoque une sortie de la boucle WHILE la plus interne. Toutes les instructions qui apparaissent après le mot clé END, marquant la fin de la boucle, sont exécutées.

CONTINUE

Fait redémarrer la boucle WHILE en ignorant les instructions après le mot clé CONTINUE.

Remarques

Si deux ou plusieurs boucles WHILE sont imbriquées, le BREAK interne quitte la prochaine boucle. Toutes les instructions après la fin de la boucle interne sont exécutées en premier, puis la boucle la plus externe suivante redémarre.

Exemples

A. Utilisation de BREAK et CONTINUE avec imbriqué SI ... ELSE et WHILE

In the following example, if the average list price of a product is less than \$300, the WHILE loop doubles the prices and then selects the maximum price. If the maximum price is less than or equal to \$500, the WHILE loop restarts and doubles the prices again. This loop continues doubling the prices until the maximum price is greater than \$500, and then exits the WHILE loop and prints a message.

```
USE AdventureWorks2012;
GO
WHILE (SELECT AVG(ListPrice) FROM Production.Product) < $300
BEGIN
    UPDATE Production.Product
        SET ListPrice = ListPrice * 2
    SELECT MAX(ListPrice) FROM Production.Product
    IF (SELECT MAX(ListPrice) FROM Production.Product) > $500
        BREAK
```



```
ELSE
    CONTINUE
END
PRINT 'Too much for the market to bear';
```

B. Using WHILE in a cursor

The following example uses @@FETCH_STATUS to control cursor activities in a WHILE loop.

Copy

```
DECLARE @EmployeeID as nvarchar(256)
DECLARE @Title as nvarchar(50)

DECLARE Employee_Cursor CURSOR FOR
SELECT LoginID, JobTitle
FROM AdventureWorks2012.HumanResources.Employee
WHERE JobTitle = 'Marketing Specialist';
OPEN Employee_Cursor;
FETCH NEXT FROM Employee_Cursor;
FETCH NEXT FROM Employee_Cursor INTO @EmployeeID, @Title;
WHILE @@FETCH_STATUS = 0
BEGIN
    Print '    ' + @EmployeeID + '    ' + @Title
    FETCH NEXT FROM Employee_Cursor;
END;
CLOSE Employee_Cursor;
DEALLOCATE Employee_Cursor;
GO
```

Examples: Azure SQL Data Warehouse and Parallel Data Warehouse

C: Simple While Loop

Dans l'exemple suivant, si le prix affiché moyen d'un produit est inférieur à \$300, la WHILE boucle double les prix et sélectionne ensuite le prix maximum. Si le prix maximum est inférieur ou égal à \$500, la WHILE boucle redémarre et double les prix. Cette boucle continue à doubler les prix jusqu'à ce que le prix maximum soit supérieur à \$500, puis quitte la WHILE boucle.

Copie

```
-- Uses AdventureWorks
```

```
WHILE ( SELECT AVG(ListPrice) FROM dbo.DimProduct) < $300
BEGIN
    UPDATE dbo.DimProduct
        SET ListPrice = ListPrice * 2;
    SELECT MAX ( ListPrice) FROM dbo.DimProduct
    IF ( SELECT MAX (ListPrice) FROM dbo.DimProduct) > $500
        BREAK;
END
```

Voir également

[ALTER TRIGGER \(Transact-SQL\)](#)

[Langage de contrôle de flux \(Transact-SQL\)](#)

[CREATE TRIGGER \(Transact-SQL\)](#)

[Curseurs \(Transact-SQL\)](#)

[SELECT \(Transact-SQL\)](#)