

WHILE (Transact-SQL)

15/03/2017 • 2 minutes de lecture •   

Dans cet article

[Syntaxe](#)

[Arguments](#)

[Notes](#)

[Exemples](#)

[Exemples : Azure SQL Data Warehouse et Parallel Data Warehouse](#)


[Voir aussi](#)


S'APPLIQUE À :  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Définit la condition qui fera se répéter l'exécution d'une instruction SQL ou d'un bloc d'instructions. L'exécution des instructions se répète aussi longtemps que la condition spécifiée demeure vraie. L'exécution des instructions de la boucle WHILE peut être contrôlée de l'intérieur de la boucle avec les mots clés BREAK et CONTINUE.

 [Conventions de la syntaxe Transact-SQL](#)

Syntaxe

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

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

Arguments

Boolean_expression

[Expression](#) qui renvoie **TRUE** ou **FALSE**. Si l'expression booléenne contient une instruction SELECT, cette dernière doit être mise entre parenthèses.

{sql_statement | statement_block}

Toute instruction ou tout groupe d'instructions Transact-SQL tels que définis dans un bloc d'instructions. Pour définir un bloc d'instructions, utilisez les mots clés de contrôle de flux BEGIN et END.

BREAK

Provoque la sortie de la boucle WHILE la plus récente. Toutes les instructions situées après le mot clé END, marquant la fin de cette boucle, sont alors exécutées.

CONTINUE

Provoque le redémarrage de la boucle WHILE, en ignorant toutes les instructions qui suivent le mot clé CONTINUE.

Notes

Si plusieurs boucles WHILE sont imbriquées, le BREAK à l'intérieur de la boucle provoque le retour à la boucle précédente. Avant que la boucle précédente ne prenne le relais, toutes les instructions situées après la fin de la boucle intérieure sont d'abord exécutées.

Exemples

A. Utilisation de BREAK et CONTINUE avec des IF...ELSE et des WHILE imbriqués

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

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

 Copier

```
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. Utilisation de WHILE dans un curseur

L'exemple suivant utilise @@FETCH_STATUS pour contrôler les activités d'un curseur dans une boucle WHILE.

 Copier

```
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
```

Exemples : Azure SQL Data Warehouse et Parallel Data Warehouse

C : Boucle While simple

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

 Copier

```
-- 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 aussi

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

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

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

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

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