

Module 07 – T-SQL



Objectifs

- Introduction T-SQL
- Quelques éléments de langage

T-SQL

- T-SQL pour Transact-SQL est le langage de programmation d'SQL Server
- vous l'utilisez déjà depuis le début du cours
- Il permet d'effectuer des opérations de type DDL, DML, etc. mais propose aussi des outils de contrôle de flot (répétitions / alternatives) et de création de fonctions (fonctions, procédures stockées, trigger, etc.)

Déclaration de variables

- Création de variables :
 - DECLARE @<nomVariable> TYPE[= <valeur>] [, @<nomVariable> TYPE[= <valeur>]]*;

```
DECLARE @LastName NVARCHAR(30), @FirstName NVARCHAR(20), @StateProvince NCHAR(2);
```

Déclaration de variables

- Affectation :
 - SET @<nomVariable> = <valeur>; => préférez cette syntaxe
 - SELECT @<nomVariable> = <valeur>; => utilisez cette syntaxe pour affecter des variables à partir de valeurs prises à partir d'une requête

```
USE AdventureWorks2014;
GO
-- Declare two variables.
DECLARE @FirstNameVariable NVARCHAR(50),
        @PostalCodeVariable NVARCHAR(15);

-- Set their values.      • <nomVariable> = <valeur>;
SET @FirstNameVariable = N'Amy';
SET @PostalCodeVariable = N'BA5 3HX';

-- Use them in the WHERE clause of a SELECT statement.
SELECT LastName, FirstName, JobTitle, City, StateProvinceName, CountryRegionName
FROM HumanResources.vEmployee
WHERE FirstName = @FirstNameVariable
      OR PostalCode = @PostalCodeVariable;
GO
```

Quelques commandes

- PRINT 'mon message' : affiche un message
- EXECUTE '<script>' : exécute un script. C'est pratique si vous avez à générer dynamiquement un script et à l'exécuter après
- EXECUTE [@<nomVariable> =] <nomProcEDUREStockee> [[param1] [paramn]*] : exécution d'une procédure stockée (vue plus loin)
- GO : termine un script T-SQL, simule le démarrage d'un nouveau script (i.e. : les variables définies avant le GO ne le sont pas après)

Contrôle de flot

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

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

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

Conventions : toujours utiliser des blocs pour les IF/ELSE/WHILE

```
BEGIN  
    { sql_statement | statement_block }  
END
```

Références

- <https://learn.microsoft.com/en-us/sql/t-sql/language-elements/language-elements-transact-sql?view=sql-server-ver16>