

MICROSOFT OFFICIAL COURSE

**Module 8 :
Utilisation d'objets de
programmation pour la
récupération de données**

Module 8 : Utilisation d'objets de programmation pour la récupération de données

- Vue d'ensemble des vues
- Vue d'ensemble des fonctions définies par l'utilisateur
- Vue d'ensemble des procédures stockées
- Vue d'ensemble des déclencheurs
- Écriture de requêtes distribuées


Leçon 1 : Vue d'ensemble des vues

- Qu'est-ce qu'une vue ?
- Création et gestion d'une vue
- Éléments à prendre en compte pour la création de vues
- Restrictions liées à la modification de données à l'aide de vues
- Vues indexées
- Exemple de vue indexée
- Vues partitionnées
- Exemple de vue partitionnée

Qu'est-ce qu'une vue ?

Une vue est une requête stockée qui produit une table virtuelle

Employee (table)				
EmployeeID	LastName	FirstName	Title	...
287	Mensa-Annan	Tete	Mr.	...
288	Abbas	Syed	Mr.	...
289	Valdez	Rachel	NULL	...



vEmployee (vue)	
LastName	FirstName
Mensa-Annan	Tete
Abbas	Syed
Valdez	Rachel

Création et gestion d'une vue

Création ou modification d'une vue

```
[CREATE|ALTER] VIEW HumanResources.vEmployee  
AS  
BEGIN  
  
SELECT EmployeeID, FirstName, LastName,  
EmailAddress  
FROM HumanResources.Employee  
  
END
```

Suppression d'une vue

```
DROP VIEW HumanResources.vEmployee
```

Éléments à prendre en compte pour la création de vues

Restriction	Description
Limite de colonne	<ul style="list-style-type: none">Le nombre total de colonnes référencées dans la vue ne peut pas dépasser 1 024
COMPUTE	<ul style="list-style-type: none">Ne peut pas être utilisé dans une définition CREATE VIEW
ORDER BY	<ul style="list-style-type: none">Ne peut pas être utilisé dans les vues, les fonctions incluses, les tables dérivées et les sous-requêtes
INTO	<ul style="list-style-type: none">Ne peut pas être utilisé avec l'instruction SELECT dans une définition de vue
Table temporaire	<ul style="list-style-type: none">Ne peut pas être référencé dans une vue
GO	<ul style="list-style-type: none">CREATE VIEW doit être unique dans un lot unique
SELECT *	<ul style="list-style-type: none">Peut être utilisé dans une définition de vue si la clause SCHEMABINDING n'est pas spécifiée

Restrictions liées à la modification de données à l'aide de vues

Restrictions relatives à l'écriture d'instructions qui modifient des données :

- Les instructions doivent modifier les colonnes d'une seule table de base
- Des critères doivent être respectés lorsque WITH CHECK OPTION est utilisé
- INSERT doit spécifier des valeurs pour toutes les colonnes qui n'acceptent pas les valeurs NULL

Restrictions liées aux colonnes lors de la modification de données :

- Les colonnes de table doivent être référencées directement
- Elles utilisent un déclencheur INSTEAD OF
- Elles ne peuvent pas être affectées par les clauses GROUP BY, HAVING ou DISTINCT
- Les données doivent respecter les restrictions sur les colonnes modifiées

Démonstration : Génération d'une vue

Dans cette démonstration, vous allez apprendre à :

- Créer une vue
- Interroger une vue
- Générer un script pour une vue

Leçon 2 : Vue d'ensemble des fonctions définies par l'utilisateur

- Qu'est-ce qu'une fonction définie par l'utilisateur ?
- Création et gestion de fonctions définies par l'utilisateur
- Création d'une fonction table définie par l'utilisateur
- Restrictions liées à la création de fonctions définies par l'utilisateur
- Comment implémenter différents types de fonctions définies par l'utilisateur
- Considérations relatives aux performances pour l'utilisation de fonctions définies par l'utilisateur

Qu'est-ce qu'une fonction définie par l'utilisateur ?

Une fonction définie par l'utilisateur est une routine qui accepte des paramètres, effectue une action et retourne le résultat de cette action sous la forme d'une valeur

Avantages liés à l'utilisation de fonctions définies par l'utilisateur

- Programmation modulaire pour logique réutilisable
- Les opérations complexes peuvent être optimisées pour une exécution plus rapide
- La logique effectuée dans la base de données réduit le trafic réseau

Création et gestion de fonctions définies par l'utilisateur

Création ou modification d'une fonction définie par l'utilisateur

```
[CREATE|ALTER] FUNCTION fEmployeeEmail(@ID int)
RETURNS varchar(50)
AS
BEGIN
  DECLARE @email varchar(50)

  SELECT @email = EmailAddress
  FROM HumanResources.Employee
  WHERE EmployeeID = @ID

  RETURN @email
END
```

Suppression d'une vue

```
DROP FUNCTION fEmployeeEmail
```

Création d'une fonction table définie par l'utilisateur

Création d'une fonction table définie par l'utilisateur

```
CREATE FUNCTION fEmployeeByGender(@Gender nchar(1))  
RETURNS table  
AS  
BEGIN  
  
    RETURN (SELECT *  
            FROM HumanResources.Employee  
            WHERE Gender = @Gender)  
  
END
```

Interrogation d'une fonction table définie par l'utilisateur

```
SELECT * FROM fEmployeeByGender('F')
```

Restrictions liées à la création de fonctions définies par l'utilisateur

Vous ne pouvez pas utiliser des fonctions définies par l'utilisateur pour :

- Mettre à jour des données
 - Utilisez à la place des procédures stockées
- Définir ou créer des objets dans la base de données
 - Les objets référencés par la fonction doivent être préalablement déclarés et créés
- Effectuer des transactions

Comment implémenter différents types de fonctions définies par l'utilisateur

Types	Utilisation
Scalaire	<ul style="list-style-type: none">• SCALAR est spécifié dans la clause RETURNS• Peut être défini à l'aide de plusieurs instructions T-SQL
Table incluse	<ul style="list-style-type: none">• TABLE est spécifié dans la clause RETURNS• N'est associé à aucune variable de retour• <i>select_stmt</i> est la seule instruction SELECT qui définit la valeur de retour
Table à instructions multiples	<ul style="list-style-type: none">• TABLE est spécifié dans la clause RETURNS• <i>function_body</i> est utilisé comme une série d'instructions T-SQL qui remplissent une variable de retour TABLE• <i>@return_variable</i> est utilisé pour stocker et accumuler des lignes qui sont retournées sous forme de valeur

Considérations relatives à l'utilisation de fonctions définies par l'utilisateur

```
SELECT MyCalculation(nom_colonne)  
FROM nom_table  
WHERE MyCondition(nom_colonne)
```

- Les deux fonctions seront appelées une fois pour chaque ligne dans la table
- Si les deux fonctions requièrent 0,1 seconde pour chaque exécution, la requête nécessitera :
 - 1 seconde pour 5 lignes
 - 10 secondes pour 50 lignes
 - 1 heure pour 18 000 lignes

Démonstration : Génération d'une fonction définie par l'utilisateur

Dans cette démonstration, vous allez apprendre à :

- Créer une fonction définie par l'utilisateur
- Appeler une fonction définie par l'utilisateur

Leçon 3 : Vue d'ensemble des procédures stockées

- Qu'est-ce qu'une procédure stockée ?
- Comment les procédures stockées sont-elles créées ?
- Exécution initiale des procédures stockées
- Applications pratiques liées aux procédures stockées

Qu'est-ce qu'une procédure stockée ?

Collection d'instructions T-SQL stockées sur le serveur

Les procédures stockées peuvent :

- accepter des paramètres d'entrée
- retourner des paramètres de sortie ou un ensemble de lignes
- retourner une valeur d'état indiquant le succès ou l'échec

Avantages liés à l'utilisation des procédures stockées :

- permet de promouvoir la programmation modulaire
- fournit des attributs de sécurité et un chaînage d'autorisations
- autorise la liaison différée et la réutilisation de code
- réduit le trafic réseau

Comment les procédures stockées sont-elles créées ?

Création d'une procédure stockée

```
CREATE PROCEDURE HumanResources.usp_GetEmployeesName  
@NamePrefix char(1)  
AS  
BEGIN  
SELECT BusinessEntityID, FirstName, LastName,  
EmailAddress  
FROM HumanResources.vEmployee  
WHERE FirstName LIKE @NamePrefix + '%'  
ORDER BY FirstName  
END
```

Appel d'une procédure stockée

```
EXECUTE HumanResources.usp_GetEmployeesName 'A'
```

Démonstration : Création de procédures stockées

Dans cette démonstration, vous allez apprendre à :

- Créer une procédure stockée
- Appeler une procédure stockée

Leçon 4 : Vue d'ensemble des déclencheurs

- Qu'est-ce qu'un déclencheur ?
- Comment les déclencheurs sont-ils créés ?
- Mode de fonctionnement des déclencheurs
- Types de déclencheurs et limitations

Qu'est-ce qu'un déclencheur ?

Collection d'instructions T-SQL stockées sur le serveur qui évalue les données avant ou après leur insertion, leur modification ou leur suppression

- Type spécial de procédure stockée qui s'exécute lors d'une tentative de modification des données dans une table
- Généralement utilisée pour gérer l'intégrité des données de bas niveau et non utilisée pour retourner des résultats de requête
- Deux catégories de déclencheurs DML : INSTEAD OF et AFTER. Le déclencheur INSTEAD OF est également appelé déclencheur BEFORE

Comment les déclencheurs sont-ils créés ?

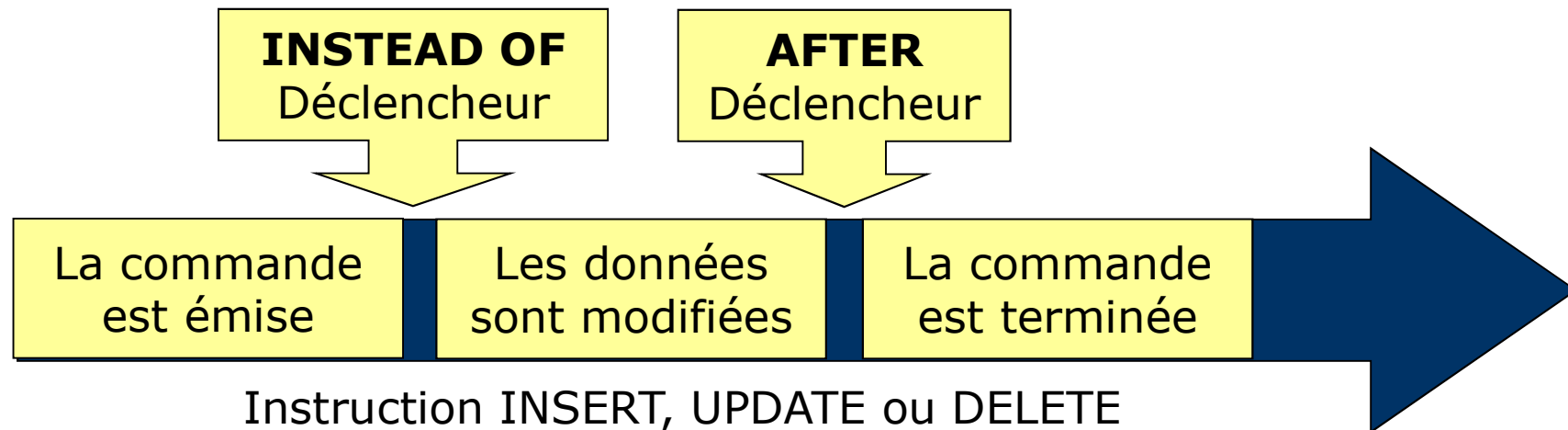
Création d'un déclencheur

```
CREATE TRIGGER Sales.trigCurrency
ON Sales.Currency
AFTER INSERT
AS
BEGIN
    DECLARE @name nvarchar(50)
    SELECT @name = Name
    FROM inserted
    IF len(@name) < 5
    BEGIN
        ROLLBACK TRANSACTION
    END
END
```

Mode de fonctionnement des déclencheurs

- Les déclencheurs peuvent restaurer des transactions lorsqu'une règle métier spécifique n'est pas satisfaite
- Lorsqu'un déclencheur contenant une instruction de restauration est exécuté à partir d'un lot SQL, le lot entier est annulé
- Toute instruction qui suit l'instruction ROLLBACK TRANSACTION sera cependant exécutée
- Toutes les modifications qui ont lieu après la restauration ne sont pas restaurées

Types de déclencheurs et limitations



Limitations relatives aux déclencheurs

- Un déclencheur ne peut s'appliquer qu'à une seule table
- Les déclencheurs sont exécutés uniquement dans la base de données actuelle
- Les déclencheurs doivent appartenir au même schéma que leur table cible
- Les déclencheurs **INSTEAD OF DELETE/UPDATE** ne peuvent pas être créés sur une table dans laquelle une clé étrangère en cascade est définie

Démonstration : Création de déclencheurs

Dans cette démonstration, vous allez apprendre à :

- Créer un déclencheur
- Afficher les résultats d'un déclencheur