

Chapitre 4 Les déclencheurs

1) Introduction

Les déclencheurs sont des objets de la base qui nous permettent d'automatiser des actions. Ils vont nous permettre d'effectuer une instruction, à chaque fois qu'une instruction SQL (Ajout, Suppression ou Modification) est effectuée pour une table donnée.

2) Création d'un déclencheur :

L'instruction CREATE TRIGGER est l'instruction SQL qui va nous permettre de créer un déclencheur. Il est nécessaire de préciser le nom du déclencheur à la suite de cette instruction.

La clause **ON** va nous permettre de définir sur quelle table nous allons définir le déclencheur que nous créons et les clauses **AFTER, INSTEAD OF** nous permettront par la suite, de donner les conditions d'action du déclencheur.

- **AFTER** : le déclencheur s'exécutera après insertion, mise à jour ou suppression dans la table qui contient le déclencheur.

- **INSTEAD OF** : le déclencheur s'exécutera avant insertion, mise à jour ou suppression dans la table qui contient le déclencheur.

Comme pour une fonction, les délimiteurs des actions dans un déclencheur sont les clauses AS BEGIN et END.

Syntaxe :

```
CREATE TRIGGER Nom_TRG ON Nom_Table  
FOR/AFTER/INSTEAD OF Action(Insert/delete/update)  
AS BEGIN  
--Instruction SQL  
END
```

Exemple :

Créer un déclencheur, dont le nom est Soustraction_Stock, sur la table commande. Nous y associons l'événement AFTER INSERT, qui précise qu'après insertions de données dans la table Commande, le code contenu entre les délimiteurs AS BEGIN et END doit être exécuté.

```
CREATE TRIGGER Soustraction_Stock ON Commande AFTER INSERT  
AS BEGIN
```

```

DECLARE @Id_Stock int, @Quantite int
SELECT @Id_Stock = Id_stock FROM INSERTED
SELECT @Quantite = Quantite FROM INSERTED
UPDATE Stock
SET Quantite = Quantite-@Quantite
WHERE Id_Stock = @Id_Stock
END

```

Le code contenu entre les délimiteurs, dans cet exemple, récupère certaines valeurs insérées dans la table Commande, en l'occurrence Id_Stock et Quantité, dans des variables déclarées en début de lot grâce aux mots clé **FROM INSERTED**, et soustrait la quantité de la commande, à la quantité du stock, en fonction d'un Id_Stock donné, grâce à un UPDATE de la table Stock.

NB :

- ✓ On cas d'insertion d'un enregistrement le trigger d'insertion crée une table appeler **inserted** (table virtuelle) cette table contient l'enregistrement à insérer quand le trouve aussi dans la table réel.
- ✓ On cas de suppression le moteur de base de donnée crée la table **deleted** cette table contient l'enregistrement à supprimer et la table d'origine réel ne contient plus cette enregistrement.
- ✓ On cas de modification la table **inserted** et la table d'origine contient la modification, la table **deleted** contient l'ancien enregistrement

Modifier un déclencheur :

```

ALTER TRIGGER commandes_insert
... -- son nouveau code

```

3) Suppression d'un déclencheur

La structure générale de suppression par code T-SQL est la même que pour n'importe quel objet de la base puisque nous allons utiliser DROP.

Voyons un exemple :

```

DROP TRIGGER Soustraction_Stock

```

Indication :

On peut créer un déclencheur pour deux ou trois événements à la fois.

Syntaxe :

```
CREATE TRIGGER ...  
ON ... AFTER INSERT, UPDATE  
AS  
....
```

Lorsque vous créez un trigger, vous spécifiez la table et les commandes de modification de données qui le déclenchent ou l'activent. Vous indiquez ensuite la ou les actions que le trigger doit entreprendre.