



Module 3 Développer la persistance des données

SQL – Implémenter SGBDR	Séance	S02	Activité	A-003
-------------------------	--------	-----	----------	-------

L'activité proposée doit vous permettre de mettre en œuvre les techniques de programmation des triggers pour :

- La vérification d'une règle de gestion.

Sommaire de l'activité proposée :

1	Vérification de règles dynamiques	2
1.1	Vérification du Délai de livraison	2

1 Vérification de règles dynamiques

Nous pouvons, pour vérifier que le contenu inséré dans une colonne respecte une règle de gestion spécifique, mettre en place une contrainte d'intégrité sur domaine (de type CONSTRAINT CHECK).

Cette approche reste toutefois limitée : la règle doit être exprimée sous forme de conditions mettant en œuvre des expressions manipulant des valeurs constantes ou des valeurs issues d'une colonne de cette même table.

Ainsi, si je souhaite mettre en place une contrainte sur domaine CHECK1 sur une colonne COL2 d'une table T1, comportant aussi une colonne COL1 je peux écrire :

COL2 > T1 : Par exemple pour la table Order_Details que Discount < 10 % si Quantity < 5 les colonnes quantity et discount se trouvant dans cette même table.

Mais nous ne pouvons pas exprimer sous cette forme une contrainte qui mettrait en œuvre des colonnes issues d'une autre table T2, comme :

T1.COL2 > T2.COL1 .

Si nous souhaitons mettre en place la vérification d'une contrainte exprimée dans cette forme, nous devons la programmer au sein d'un programme déclencheur, dit Trigger.

1.1 Vérification du Délai de livraison

Vous devez programmer un trigger qui interdise que la date de livraison prévue d'une commande RequiredDate de Orders soit inférieure à la date de commande + 3 semaines si l'un des articles commandés est en rupture de stock (Discontinued = true) dans la table Products.

Dans un premier temps testez uniquement l'insertion d'une **seule** ligne sur une commande existante. Programmez ce trigger pour les opérations d'insertion et de modification.

Rappel de la règle de gestion :

Une commande est considérée non valide si un produit en rupture de stock doit être livré dans un délai inférieur à 3 semaines à partir de la date de commande.

1 commande représente une transaction atomique qui prend en compte l'ensemble des opérations d'insertion suivantes :

- 1 ligne dans la table Orders
- x lignes dans la table Order_Details

Pour vérifier la règle de gestion énoncée, nous avons besoin :

De la date de commande : Orders.OrderDate

De la date de livraison exigée : Orders.RequiredDate

De la valeur de la colonne Products.Discontinued

La transaction sera abandonnée dès lors qu'une ligne détail de commande comportera un produit en rupture avec une livraison < 3 semaines

Le trigger sera donc associé à la table Order_Details sur opération INSERT

Pour annuler la transaction, vous pourrez, au choix, exécuter au sein du trigger considéré :

- Un ordre ROLLBACK
- Un ordre RAISERROR avec un message d'une gravité supérieure à 15
- Un ordre THROW

Pour apprécier différences entre Raiserror et Throw <https://docs.microsoft.com/fr-fr/sql/t-sql/language-elements/throw-transact-sql?view=sql-server-ver15>

Pour tester le bon fonctionnement du trigger, réalisez sous forme d'une transaction complexe une opération de prise de commande.

- Modifier la date de livraison requise pour qu'elle soit inférieure à DateCommande + 3 semaines
- Insérer une ligne de commande avec produit en stock OK
- Insérer une ligne de commande avec produit en rupture

Exemple de script intégrant une transaction pour la création d'une nouvelle commande avec deux références de produit pour tester la rupture de stock. A réaliser dans la base ComptoirAnglais.

```
DECLARE @NumCde Int

-- Début de la Transaction

BEGIN TRAN

INSERT INTO [Orders]
    ([CustomerID], [EmployeeID], [OrderDate], [RequiredDate], [ShippedDate]
, [ShipVia], [Freight], [ShipName], [ShipAddress], [ShipCity], [ShipRegion]
, [ShipPostalCode], [ShipCountry])

SELECT
    [CustomerID], [EmployeeID], [OrderDate], [RequiredDate], [ShippedDate], [ShipVia], [Freight]
, [ShipName], [ShipAddress], [ShipCity]
, [ShipRegion]
, [ShipPostalCode]
, [ShipCountry]
        FROM Orders
        WHERE OrderId = 10248
-- Récupération du N° de commande
SET @NumCde = SCOPE_IDENTITY()

PRINT @NUMCDE
-- Insertion des lignes de commandes

INSERT INTO [Order Details]
    ([OrderID]
, [ProductID]
, [UnitPrice]
, [Quantity]
, [Discount])
    VALUES
    (@Numcde
, 11 -- Mettre Code produit OK ( non en rupture)
, 20.34
, 10
, 0.259)
INSERT INTO [Order Details]
    ([OrderID]
, [ProductID]
, [UnitPrice]
, [Quantity]
, [Discount])
    VALUES
    (@Numcde
, 20 -- Code produit non OK (en rupture)
, 20.34
, 10
, 0.259)
-- Validation de la transaction
COMMIT TRAN
```