

PL/SQL - 4

Objectifs du cours

- Déclencheurs

Distinguer

- Les déclencheurs de base de données
 - Enregistrés dans la bd (côté serveur)
 - Contrôlent la dynamique de la bd
 - Par exemple : quand un tuple est inséré dans une table
- Les déclencheurs d'application
 - Enregistrés avec l'application (côté client)
 - Contrôlent la dynamique de l'interface
 - Par exemple : quand l'utilisateur clique sur un bouton, WHEN-BUTTON-PRESSED

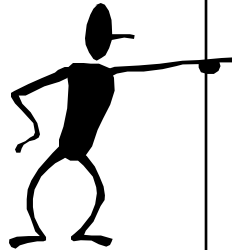
Déclencheurs de Base de Données

- Pourquoi faire ?
- Comment les
 - créer
 - activer
 - désactiver
 - supprimer
- Exemples : application d'une règle de sécurité avec des déclencheurs
- Règles d'utilisation des déclencheurs

Déclencheurs BD - intérêts

J'ai besoin de
toutes les pièces
N° 110

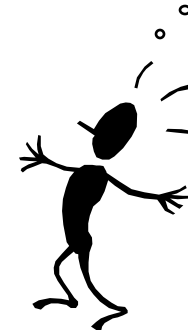
°
°



Stock

N°Pièce	quantitéDispo
100	12
105	23
110	50
220	200
222	201
228	87

Eh ...
mais il faudra remplir
le stock de pièces
N° 110 !!!



- En programmation classique :
 - écrire une routine ou une procédure appliquant la règle
 - appeler cette routine à chaque point précis de l'application ou la règle doit être mise en oeuvre
 - => la règle n'est appliquée que si elle est invoquée explicitement
 - => difficile d'obtenir la liste des endroits où la règle est mise en oeuvre

Déclencheurs - intérêts

- **Définition :**
Un déclencheur de base de données est un ensemble d'instructions qui s'exécutent lorsque le contenu d'une table est modifié par l'intermédiaire d'une instruction INSERT, UPDATE ou DELETE
- **Les déclencheurs permettent :**
 - d'implémenter des règles de sécurité
 - de contrôler la cohérence des données contenues dans une table lors de leur modification
 - d'effectuer des validations complexes en fonctions de données provenant de tables multiples
 - d'implémenter des règles de gestion
 - de garder trace des modifications en reportant les valeurs modifiées dans une autre table
 - de calculer des valeurs de colonnes par défaut
 - de décrire le comportement des composants des interfaces graphiques développées dans Developer 2000

Déclencheurs - intérêts

- Ces règles sont implémentées une fois pour toutes; elles n'ont pas à être vérifiées à chaque utilisation des tables sur laquelle elles portent
 - compatibilité avec le paradigme événementiel
 - => pas d'invocation explicite de la règle
 - => contrôle systématique de chaque modification des tables
 - facilité de mise à jour et de maintenance
 - => code de la règle séparé de l'applicatif de traitement normal
 - => contrôle centralisé de toutes les modifications d'une table
- Les déclencheurs sont des block PL/SQL qui s'exécutent *lorsque une table est modifiée* =>
 - impossible de savoir à l'avance quand un déclencheur sera exécuté
 - difficile de savoir quand un déclencheur s'exécute
 - plus facile de savoir quels déclencheurs se sont exécutés

Manipulation des déclencheurs BD

- créer
- activer
- désactiver
- supprimer

Création des déclencheurs

- Syntaxe :
CREATE [OR REPLACE] TRIGGER <nom de déclencheur>
{BEFORE | AFTER} <événement déclenchant> ON <nom table>
[FOR EACH ROW]
[WHEN (<condition>)]
<Bloc PL/SQL>

avec :

<événement déclenchant> =

- INSERT : insertion d'une nouvelle ligne dans la table
- UPDATE : modifications des attributs d'une ligne de la table
- DELETE : suppression d'une ligne de la table
- une combinaison quelconque des trois (par ex : INSERT OR DELETE)

Création des déclencheurs - exemple

```
CREATE OR REPLACE TRIGGER maintientDuStock
AFTER INSERT OR UPDATE ON Stock
FOR EACH ROW
WHEN (:new.quantiteDispo < 10 OR :new.quantiteDispo IS NULL)
BEGIN
    INSERT INTO Commandes (N°Produit, quantiteCommandee)
        VALUES (:new.N°Produit, 200) ;
END ;
```

Déclencheurs de niveau instruction / ligne

- Un déclencheur de niveau instruction :
 - *n'inclue pas la clause* FOR EACH ROW dans l'instruction CREATE TRIGGER
 - ne s'exécute qu'une fois pour un événement particulier
 - ne peut accéder aux valeurs des lignes qui pourraient être modifiées par l'événement déclenchant
 - est adapté pour retrouver l'auteur ou la date de l'événement déclenchant
- Un déclencheur de niveau ligne :
 - *doit inclure la clause* FOR EACH ROW dans l'instruction CREATE TRIGGER
 - est déclenché pour chaque ligne modifiée par l'événement déclenchant
 - peut accéder aux anciennes et aux nouvelles valeurs modifiées par cet événement
 - est adapté pour mettre en application les règles de gestion et de sécurité

Référence aux colonnes de la table sur laquelle porte le déclencheur

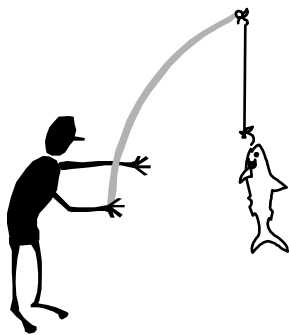
- Dans le code associé aux déclencheurs de niveau ligne, on veut accéder aux valeurs des attributs de la ligne modifiée
=> utilisation de deux variables :
 - :old
 - :new
- Pour un déclencheur sur INSERT
 - les nouvelles valeurs sont dans :new.<nom d'attribut>
- Pour un déclencheur sur UPDATE
 - les anciennes valeurs sont dans :old.<nom d'attribut>
 - les nouvelles valeurs sont dans :new.<nom d'attribut>
- Pour un déclencheur sur DELETE
 - les anciennes valeurs sont dans :old.<nom d'attribut>

Déclencheurs BEFORE et AFTER

- Tout déclencheur BEFORE :
 - est exécuté avant que l'événement déclenchant n'ait lieu
- Tout déclencheur AFTER :
 - est exécuté après que l'événement déclenchant ait lieu
- Un déclencheur BEFORE de niveau ligne :
 - est exécuté avant que l'événement déclenchant n'ait lieu
 - peut affecter les valeurs de la ligne insérée ou modifiée
- Un déclencheur AFTER de niveau ligne :
 - est exécuté après que l'événement déclenchant ait lieu
 - ***ne peut pas*** affecter les valeurs de la ligne insérée ou modifiée

Récapitulatif des déclencheurs possibles

- Six déclencheurs de niveau ligne :
 - BEFORE INSERT, BEFORE UPDATE, BEFORE DELETE
 - AFTER INSERT, AFTER UPDATE, AFTER DELETE
- Six déclencheurs de niveau instruction :
 - BEFORE INSERT, BEFORE UPDATE, BEFORE DELETE
 - AFTER INSERT, AFTER UPDATE, AFTER DELETE



=> à vous de choisir le(s) déclencheur(s) dont vous avez besoin en fonction des règles que je viens de donner

Activation / désactivation / suppression des déclencheurs

- Désactivation de déclencheur - Syntaxe :
ALTER TRIGGER <nom du déclencheur> DISABLE ;
- Activation de déclencheur - Syntaxe :
ALTER TRIGGER <nom du déclencheur> ENABLE ;
- Suppression de déclencheur - Syntaxe :
DROP TRIGGER <nom du déclencheur> ;

Mise en œuvre d'une politique de sécurité avec des déclencheurs - exemple

- Domaine : crédit par carte
 - probabilité de fraude >80% lorsque le cumul des débits dépasse 5000 euros pour les trois derniers jours
 - décision : enregistrer tous les numéros de compte concernés dans une table séparée afin d'en faire une analyse détaillée
- Créer un déclencheur sur la table Debits
 - qui s'exécute après l'insertion d'une ligne dans cette table
 - qui évalue le montant des débits du compte concerné pour les trois derniers jours
 - qui insère une ligne dans la table ComptesMenacés si ce montant est supérieur à 5000

Debits (numTransaction , numCompte, dateTransaction, montant , numCarte)

ComptesMenacés (numCompte, numCarte, montantDerniersDebits, numTransaction)

Mise en œuvre d'une politique de sécurité avec des déclencheurs - exemple

```
CREATE OR REPLACE TRIGGER SurveillanceTentativesFraudes
AFTER INSERT ON Debits
FOR EACH ROW
DECLARE
    montant3DerniersJours_v    number ;
BEGIN
    SELECT SUM (montant)
    INTO montant3DerniersJours_v
    FROM Debits
    WHERE      numCompte = :new.numCompte
              AND  dateTransaction <= SYSDATE -3 ;
    ...
```

Mise en œuvre d'une politique de sécurité avec des déclencheurs - exemple

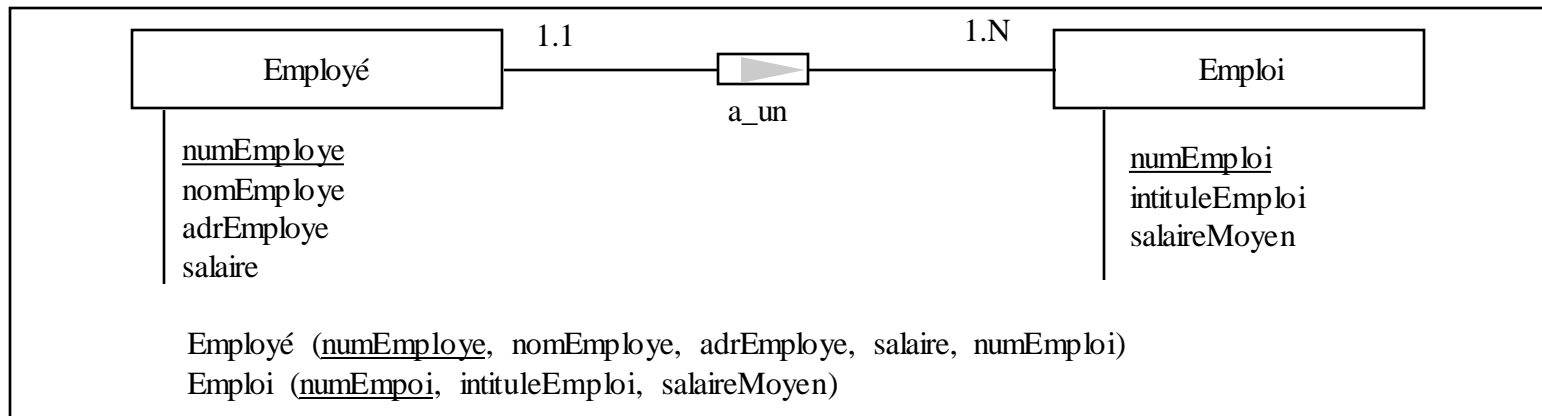
```
IF (montant3DerniersJours_v > 5000) THEN
    INSERT INTO ComptesMenacés
        (numCompte, numCarte, montantDerniersDebits, numTransaction)
    VALUES
        (:new.numCompte, :new.numCarte,
        montant3DerniersJours_v, :new.numTransaction)
END IF;
END ;
```

Quelques remarques sur les déclencheurs

- Déclencheurs en cascade : un déclencheur réalisant de INSERT, UPDATE ou DELETE peut générer des événements amenant à l'exécution d'un ou plusieurs autres déclencheurs, et ainsi de suite; on parle alors de déclencheurs en cascade. ***Evitez plus de deux niveau de cascade.***
- Mutation : tant que les modifications (INSERT, UPDATE ou DELETE) d'une table ne sont pas terminées, celle-ci est dite en cours de mutation. ***Un déclencheur de niveau ligne ne peut lire ou modifier une table en cours de mutation*** (car elle est instable).
- Validation : un déclencheur ne peut ***ni*** exécuter d'instruction COMMIT ou ROLLBACK, ***ni*** appeler de fonction, procédure ou sous-programme de package invoquant ces instructions.

Exercice

- Chercher les déclencheurs nécessaires au contrôle des contraintes d'intégrité identifiée par le schéma Entité/Relation ci-dessous
- Ecrire le code l'un des déclencheurs permettant d'effectuer ce contrôle



Notes :

- le salaire moyen identifié dans la relation emploi doit valoir la moyenne des salaires de tous les employés qui détiennent cet emploi
- il y a au moins quatre déclencheurs, aidez-vous du transparent N°12