

# **PL/SQL**

---

## **TP triggers**

---

# 1 Aide mémoire PL/SQL

Bloc PL/SQL	<pre> DECLARE     -- Déclaration constantes/variables BEGIN     -- Commandes/instructions EXCEPTION     -- Traitement des erreurs à l'exécution END; / -- Exécution automatique lors du « start » </pre>
Déclaration de variable	Nom_Variable TYPE_VARIABLE;
Affectation	Nom_Variable := valeur;
Tests	<pre> SELECT attribut INTO Nom_Variable FROM table; IF condition1 THEN     -- Instructions ELSIF condition2 THEN -- (Optionnel)     -- Instructions ELSE -- (Optionnel)     -- Instructions END IF; </pre>
Boucles	<pre> FOR compteur IN [REVERSE] min..max LOOP     -- Instructions END LOOP; WHILE condition LOOP     -- Instructions END LOOP; </pre>
Curseurs - Déclaration - Utilisation	<pre> CURSOR Nom_Curseur IS Requête_SQL; FOR nuplet IN Nom_Curseur LOOP     -- Instructions     -- Ex. Nom_Variable := nuplet.attribut; END LOOP;-- NB : nuplet est de type Nom_Curseur%ROWTYPE </pre>
Exceptions - Déclarer - Lever - Traiter	<pre> Nom_Exception EXCEPTION; RAISE Nom_Exception; WHEN Nom_Exception THEN -- Instruction; </pre>
Documentation PL/SQL	<a href="http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28370/toc.htm">http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28370/toc.htm</a>

Si ce n'est déjà fait, vous allez réaliser une copie des tables suivantes appartenant au schéma SCOTT pour pouvoir travailler confortablement avec des tables appartenant à votre schéma :

- EMP
- DEPT

Pour mémoire, vous pouvez réaliser simplement une copie d'une table (structure + données) via l'ordre SQL suivant :

```
CREATE TABLE matable
AS SELECT * FROM AUTRESHEMA.tablesource ;
```

## 2 Déclencheurs

### 2.1 Numérotation automatique

Soit une table quelconque TABL, dont la clé primaire CLENUM est numérique. Définir un trigger en insertion permettant d'implémenter une numérotation automatique de la clé. Le premier numéro doit être 1.

### 2.2 Règles métiers

Soit le schéma relationnel d'une agence bancaire régionale.

- CLIENT (NUMCL, NOM, PRENOM, ADR, CP, VILLE, SALAIRE, CONJOINT)
- DETENTEUR (NUMCL#, NUMCP#)
- COMPTE (NUMCP, DATEOUVR, SOLDE)

Attributs soulignés : Clés primaires.

Attributs suffixés par # : Clés étrangères.

Écrire un trigger en insertion permettant d'implémenter les contraintes suivantes :

- le département dans lequel habite le client doit être 01, 07, 26, 38, 42, 69, 73, ou 74 ;
- le nom du conjoint doit être le même que celui du client.

### 2.3 Logging

Ajouter à votre schéma la table suivante :

JOURNAL(TSTAMP , USRNAME , TABLE , ACTION) ;

Conseil : pour l'attribut TSTAMP, utilisez le type de données TIMESTAMP.

La raison d'être de cette table est de consigner les actions effectuées par les utilisateurs, on retrouvera donc par exemple le contenu suivant :

TSTAMP	USRNAME	TABLE	ACTION
16/10/13 08:45:30,723000	SYS	EMP	INSERT
16/10/13 09:00:30,787000	SYS	DEPT	UPDATE
16/10/13 09:12:30,724000	SYS	DEPT	DELETE
16/10/13 09:25:30,773000	SYS	DEPT	INSERT
16/10/13 10:05:30,824000	SYS	EMP	UPDATE

Vous pouvez récupérer le nom d'utilisateur grâce à la fonction PL/SQL USER et le timestamp courant grâce à la fonction PL/SQL CURRENT\_TIMESTAMP :

```
SYS:LYON1>select CURRENT_TIMESTAMP from DUAL ;
CURRENT_TIMESTAMP
-----
16/10/13 08:54:03,080000 +02:00

SYS:LYON1>select USER from DUAL ;
USER
-----
SYS
```

Ajouter à la table DEPT un trigger qui aura en charge de créer les entrées correspondantes dans la table LOG lorsqu'un utilisateur exécute un ordre de type DML sur la table DEPT (INSERT, UPDATE, DELETE).