

# Bases de Données

Module : BDD – Administration des données

- Evaluation Individuelle
  - durée 1H30
  - jeudi 14 décembre 2023 matin
- Professeur : David MOTSCH
- Mail : motsch@3il.fr



# Bases de Données

## Contraintes d'intégrité

- ▶ Définition :
  - Une contrainte d'intégrité est en général un prédicat que doit vérifier un sous ensemble de la base afin que l'on puisse considérer les informations comme cohérentes.

# Bases de Données

## ► Les CI :

- Unicité de la clé primaire (primary key)
- Clé étrangère (foreign key, references)
- Contrôle des données (check)
  
- Exemple :
  - Valeur en majuscule ou première lettre en majuscule
  - Vérifier  $\text{datefin} \geq \text{datedebut}$
  - Not null

# Bases de Données

## ► Les CI dynamiques :

- Elles concernent le passage d'un état de la base de données à un autre état.
- Le trigger (déclencheur)
  - C'est un traitement qui s'exécute automatiquement sur un ordre (INSERT, UPDATE, DELETE) sur une table.



- ▶ Création des tables :

- CREATE TABLE *nom\_table* (  
    *Col1* type [*constraint nomcontrainte lacontrainte*],  
    *Col2* type [*constraint nomcontrainte2 lacontrainte*],  
    ...  
    *Coln* type [*constraint nomcontrainten lacontrainte*],  
    [*constraint nomcontrainten lacontrainte*]);

# Bases de Données

## ▶ PL/SQL

- Le PL/SQL est un langage qui intègre le SQL et permet de programmer de manière procédurale.
  - Interactif
  - Séquences
  - Curseurs
  - Procédures Stockées
  - Fonctions stockées
  - Packages
  - Triggers
- C'est un langage propre à Oracle. Il peut être exécuté dans un script ou directement stockées en base

# Bases de Données

► Interactif :

- BEGIN  
FOR i in 1..9 LOOP  
INSERT INTO FOURNISSEUR VALUES ('F00' || i,  
'nom ' || i, 'adresse ' || i, 'ville || i);  
END LOOP;  
END;  
/

## ► Les Séquences :

Create sequence nom\_seq  
Increment by n  
Start with v;

- Avec  $n \geq 1$  et  $v \geq 1$ .
- Si increment by est omis, alors la séquence sera incrémentée de 1 ( $n=1$ )
- Si Start with est omis, la séquence commencera à 1 ( $v=1$ )





# Bases de Données

- ▶ Utilisation des séquences :
  - Nextval et Currval sont des Pseudo colonnes de la séquence
  - Nextval : prochaine valeur de la séquence
  - Currval : Dernière valeur utilisée de la séquence dans la session.



# Bases de Données

## ► Les curseurs :

- PL/SQL utilise les curseurs pour les accès à des informations de la BD.
- Déclaration : `Cursor nom_curseur IS SELECT ....`
- Ouverture : `OPEN nom_curseur;`
- Fermeture : `CLOSE nom_curseur;`
- Extraction d'une ligne : `FETCH nom_curseur INTO Var1, Var2, ...`

## ► Les curseurs :

- Il y a 4 attributs associés aux curseurs PL/SQL
  - %NOTFOUND
  - %FOUND
  - %ROWCOUNT
  - %ISOPEN

## ► Les curseurs :

### ◦ Declare

```
CURSOR curs_liste IS select nom from fournisseur;  
BEGIN  
  FOR enreg in curs_liste loop  
    DBMS_OUTPUT.PUT_LINE('Nom : ' || enreg.nom);  
  END LOOP;  
END;
```

- ▶ Les procédures indépendantes :

```
Create [OR REPLACE] PROCEDURE nom_procedure  
[(argument1 type ..., argumentN type)]  
IS  
[déclaration_variables_locales]  
BEGIN  
    Traitement_Procedure  
END [nom_procedure];
```

- ▶ Les procédures indépendantes :
  - Pour exécuter une procédure :
    - Sous SQLPLUS, On tape la commande :
      - SQL> execute nom\_procedure (Argument1, Argument2, ..., ArgumentN);
    - Dans un bloc PL/SQL :

```
BEGIN  
    nom_procedure(Argument1, Argument2, ..., ArgumentN);  
END;
```

- ▶ Les fonctions stockées :

```
Create [OR REPLACE] FONCTION nom_fonction
[(argument1 type ..., argumentN type)]
RETURN type_donnees IS
[déclaration_variables_locales]
BEGIN
    Traitement_fonction
RETURN valeur_fonction;
END [nom_fonction];
```

# Bases de Données

- ▶ Les fonctions stockées :
  - Pour exécuter une fonction, on utilise la commande SELECT :
  - `SELECT nom_fonction (Argument1, Argument2, ..., ArgumentN)  
FROM DUAL;`





# Bases de Données

- ▶ Modes des paramètres de sous-programmes
  - Les paramètres peuvent être définis IN (mode par défaut), IN OUT ou OUT, en fonction de la nature de l'information devant être passée.
  - IN : Le paramètre ne peut recevoir de valeur.
  - OUT : On ne peut pas utiliser le paramètre dans un sous-programme
  - IN OUT : On peut réaffecter la valeur dans un sous-programme.



# Bases de Données

Pour consulter le code source de votre procédure, vous pouvez utiliser la vue du dictionnaire de données Oracle : `user_source`.

Exemple :

```
select line, text from user_source  
Where name = 'nom_procedure';
```



- ▶ Les packages:

```
Create [OR REPLACE] PACKAGE nom_package IS  
  Section_declaration;  
END nom_package;
```

```
Create PACKAGE BODY nom_package IS  
  Section_declaration;  
  Corps_procedure;  
  Corps_fonction ;  
END nom_package;
```

## ► TRIGGER

```
CREATE OR REPLACE TRIGGER nom_trigger
Before/After insert or update or delete on nom_table
For each row
Declare
    [déclaration_variables]
Begin
    Traitement_Procedure
End;
/
```