

I-Introduction aux SGBDs

2^{ème} Ingénieurs info

Année Universitaire 2019-2020

R.MGHIRBI

Plan du module

- ◆ **Partie 1- I. Introduction aux SGBDs**
 - Chapitre 1: Présentation des SGBDs
 - Chapitre 2: Définition et Evolution des données
 - Chapitre 3: Contrôle des données
 - Chapitre 4: Gestion des objets utilisateurs
 - (Vues, séquences et Index)
- ◆ **Partie 2- II. Langage procédural: PL/SQL**
- ◆ **Partie 3- III. Les Transactions**
- ◆ **Partie 4- IV. L'interface JDBC**
 - Connexion à une base et état de connexion
 - Interaction avec la base et extraction des données

Rappel de définition des tables

Syntaxe de création

```
CREATE TABLE [schéma.]nomTable  
  ( colonne1 type1 [DEFAULT valeur1] [NOT NULL]  
    [, colonne2 type2 [DEFAULT valeur2] [NOT NULL] ]  
    [CONSTRAINT nomContraintel typeContraintel]...) ;
```

◆ *schéma* :

- s'il est omis, il sera assimilé au nom de l'utilisateur connecté.
- S'il est précisé, il désigne soit l'utilisateur courant soit un autre utilisateur de la base
 - Dans ce cas, il faut que l'utilisateur courant ait le droit de créer une table dans un autre schéma.

Rappel de définition des tables

Définition de contraintes:

- ◆ 2 manières de définition: niveau ligne et niveau relation
- ◆ 4 types de contraintes:

```
CONSTRAINT nomContrainte  
  • UNIQUE (colonne1 [,colonne2]...)  
  • PRIMARY KEY (colonne1 [,colonne2]...)  
  • FOREIGN KEY (colonne1 [,colonne2]...)  
    REFERENCES [schéma.]nomTablePere (colonne1 [,colonne2]...)  
    [ON DELETE { CASCADE | SET NULL }]  
  • CHECK (condition)
```

- ◆ CASCADE propagera la suppression de tous les enregistrements fils rattachés à l'enregistrement père supprimé,
- ◆ SET NULL positionnera seulement leur clé étrangère à NULL

Rappel de définition des tables

Destruction d'un schéma:

```
DROP TABLE [schéma.]nomTable [CASCADE CONSTRAINTS];
```

- ◆ CASCADE CONSTRAINTS permet de s'affranchir des clés étrangères actives contenues dans d'autres tables et qui référencent la table à supprimer.
 - Cette option détruit les contraintes des tables « fils » associées sans rien modifier aux données qui y sont stockées.
- ◆ RQ: Les éléments qui utilisaient la table (vues, synonymes, fonctions ou procédures) ne sont pas supprimés mais sont temporairement inopérants.
- ◆ Attention, une suppression ne peut pas être par la suite annulée.

5. EVOLUTION D'UN SCHÉMA

5.1. Renommage de tables

Ordre Remane: `RENAME ancienNom TO nouveauNom;`

♦ *Exemple 1:* `RENAME Pilote TO Naviguant;`

Ou

ALTER: `ALTER TABLE ancienNom RENAME TO nouveauNom;`

♦ *Exemple 2:*

`ALTER TABLE Pilote RENAME TO Naviguant;`

5.2. Modification structurelle

Syntaxe

ALTER TABLE *Nom de Table* <Altération>

Différents types d'altérations sont possibles :

- Ajout d'une colonne **ADD nomCOLUMN (définition_col)**
- Modification du nom /définition d'une colonne
RENAME COLUMN TO (NOM)
MODIFY (Définition)
- Suppression d'une colonne **(DROP COLUMN)**
- Suppression de plusieurs colonnes **DROP**

Exemples

- ♦ Soit la table pilote (brevet, nom)
- ♦ *Exemple 3*: ajout de colonnes:

```
ALTER TABLE Pilote ADD (nbHVol NUMBER(7,2));
```

```
ALTER TABLE Pilote  
ADD (compa VARCHAR2(4) DEFAULT 'AF',  
ville VARCHAR2(30) DEFAULT 'Paris' NOT NULL);
```

- ♦ *Exemple 4*: renommer une colonne

```
ALTER TABLE Pilote RENAME COLUMN ville TO adresse;
```

Exemples

- ♦ Modifier la définition d'une colonne
 - Type, taille ou contraintes

```
ALTER TABLE Pilote  
MODIFY compa VARCHAR(6)  
DEFAULT 'SING';  
INSERT INTO Pilote (brevet, nom)  
VALUES ('PL-2', 'Laurent Boutrand');
```

Augmente la taille de la colonne *compa* et change la contrainte de valeur par défaut.

```
ALTER TABLE Pilote  
MODIFY compa CHAR(4) NOT NULL;
```

Diminue la colonne et modifie également son type de VARCHAR2 en CHAR tout en le déclarant NOT NULL (possible car les données contenues dans la colonne ne dépassent pas quatre caractères).

```
ALTER TABLE Pilote MODIFY compa NULL;
```

Rend possible l'insertion de valeur nulle dans la colonne *compa*.

```
ALTER TABLE Pilote DROP COLUMN adresse;
```

5.2. Modification structurelle

- ◆ RQ: Il n'est pas possible de supprimer avec DROP COLUMN:
 - Des clés primaires (ou candidates par UNIQUE) référencées par des clés étrangères ;
 - Des colonnes à partir desquelles un index a été construit ;
 - Des pseudo-colonnes (ROWID et LEVEL) ou des colonnes de tables objets ;
 - Toutes les colonnes d'une table.

5.3. Modification comportementale

ALTER TABLE *Nom_de_Table* <Altération>

- Ajout d'une contrainte (ADD CONSTRAINT)
- Suppression d'une contrainte (DROP CONSTRAINT).
- Désactiver/réactiver une contrainte:

5.3. Modification comportementale

Ajout d'une contrainte:

```
ALTER TABLE [schéma.]nomTable  
ADD [CONSTRAINT nomContrainte] typeContrainte;
```

- ◆ Les types de contraintes étant comme pour la création
 - UNIQUE (colonne1 [,colonne2]...)
 - PRIMARY KEY (colonne1 [,colonne2]...)
 - FOREIGN KEY (colonne1 [,colonne2]...)
REFERENCES [schéma.]nomTablePère (colonne1 [,colonne2]...)
[ON DELETE { CASCADE | SET NULL }]
 - CHECK (condition)
- ◆ Exemple 3 :

```
ALTER TABLE Avion  
ADD (CONSTRAINT nn_proprio CHECK (proprio IS NOT NULL),  
CONSTRAINT fk_Avion_comp_Compag FOREIGN KEY(proprio)  
REFERENCES Compagnie(comp));
```

2^{ème} Ing.Inf

5.3. Modification comportementale

Suppression de contraintes

```
ALTER TABLE [schéma.]nomTable DROP CONSTRAINT nomContrainte [CASCADE];
```

- ◆ Exemple 5:

```
ALTER TABLE Avion DROP CONSTRAINT pk_Avion CASCADE;
```
- ◆ Cette commande supprime à la fois la clé primaire de la table Avion mais aussi les contraintes clés étrangères des tables dépendantes (pas les attributs).
- ◆ Si l'option CASCADE n'avait pas été spécifiée, Oracle aurait renvoyé une erreur: cette clé unique/primaire est référencée par des clés étrangères ».

2^{ème} Ing.Inf

5.3. Modification comportementale

Désactivation de contraintes

```
ALTER TABLE [schéma.] nomTable
```

```
    DISABLE [ VALIDATE | NOVALIDATE ] CONSTRAINT nomContrainte  
    [ CASCADE ] [ { KEEP | DROP } INDEX ] ;
```

- ♦ *CASCADE* répercute la désactivation des clés étrangères des tables « fils » dépendantes.
message d'Oracle renvoyé est : « ORA-02297: impossible désactiver une clé primaire référencée par une clé étrangère sans cette option, le ver contrainte... - les dépendences existent ».
- ♦ Les options *KEEP INDEX* et *DROP INDEX* permettent de préserver ou de détruire l'index dans le cas de la désactivation d'une clé primaire.
- ♦ Exemple 6:

```
ALTER TABLE Avion DISABLE CONSTRAINT pk_Avion CASCADE DROP INDEX;  
INSERT INTO Avion VALUES ('Bidon1', 'TB-21', 1000, 'AF');
```

- ♦ La désactivation de cette contrainte par *CASCADE* désactive aussi une des clés étrangères de la table référençant la première.

5.3. Modification comportementale

Réactivation de contraintes

```
ALTER TABLE [schéma.] nomTable
```

```
    ENABLE [ VALIDATE | NOVALIDATE ] CONSTRAINT nomContrainte  
    [ USING INDEX ClauseIndex ] [ EXCEPTIONS INTO tableErreurs ] ;
```

- ♦ *VALIDATE* vérifie que les données courantes de la table respectent la contrainte ;
- ♦ La clause d'index permet, dans le cas des clés primaires ou candidates (*UNIQUE*), de pouvoir recréer l'index associé.
- ♦ La clause d'exceptions permet de retrouver les enregistrements ne vérifiant pas la nouvelle contrainte.
- ♦ RQ: Il n'est pas possible de réactiver une clé étrangère tant que la contrainte de clé primaire référencée n'est pas active.

- ♦ Exemple 7

```
ALTER TABLE Avion ENABLE CONSTRAINT pk_Avion  
    USING INDEX (CREATE UNIQUE INDEX pk_Avion ON Avion (immat));
```


5.4 Exploration des contraintes dans le DD oracle

OWNER	Propriétaire de la contrainte
CONSTRAINT_NAME	Nom de la contrainte
CONSTRAINT_TYPE	Type de la contrainte <ul style="list-style-type: none"> • C (check constraint on a table) • P (primary key) • U (unique key) • R (referential integrity) • V (with check option, on a view) • O (with read only, on a view)
TABLE_NAME	Nom de la table ou la contrainte a été définie
SEARCH_CONDITION	Texte de la condition de recherche pour une contrainte check
R_OWNER	Propriétaire de la table référencée par la contrainte
R_CONSTRAINT_NAME	Nom de la contrainte unique ou primaire référencée par la contrainte (s'il s'agit d'une clé étrangère)
STATUS	Statut d'une contrainte (ENABLED or DISABLED)
LAST_CHANGE	La dernière fois quand la contrainte a été désactivée ou réactivée
INDEX_OWNER	Nom de l'utilisateur propriétaire de l'index
INDEX_NAME	Nom de l'index (seulement pour les contraintes unique et primary-key)
VIEW_RELATED	Si la contrainte est reliée à une vue

2^{ème} Ing.Inf

18

5.4 Exploration des contraintes dans le DD oracle

- ♦ On peut visionner les contraintes grâce aux tables du dictionnaire de données.
- ♦ Ces tables sont:
 - [DBA/USER/ALL]_CONSTRAINTS et
 - [DBA/USER/ALL] _CONS_COLUMNS.
- ♦ *Exemple 8 :*
 - Afficher les noms des contraintes (seulement 10 caractères), leurs types, et leurs contenus pour l'utilisateur courant relativement à la table EMP
 - `Select substr(constraint_name, 1, 10), constraint_type, search_condition From USER_CONSTRAINTS`
`Where upper(table_name) = 'EMP';`
- ♦ *Exemple 9 :*
 - Afficher les contraintes (leurs noms et colonnes concernées) de l'utilisateur connecté pour la table DEPT
 - `Select constraint_name, column_name From USER_CONS_COLUMNS Where`
`upper(table_name) ='DEPT';`

2^{ème} Ing.Inf

19

Exemple

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE,  
SEARCH_CONDITION, INDEX_NAME  
FROM DBA_CONSTRAINTS  
WHERE upper(TABLE_NAME) = UPPER('Nom_TABLE') ;
```

Bibliographie

- ◆ Livres:
 - C.Soutou, SQL pour Oracle, editions 3 et 7
- ◆ Supports de cours
 - J.Y. Antoine, Administration des bases de données