

SMI-S5

SGBD ORACLE : LDD

Pr. Issam QAFFOU

Laboratoire Ingénierie des Systèmes d'Information
Département d'Informatique
FSSM-UCA

Définition

- Dans le modèle relationnel, les données sont organisées et structurées suivant des relation qu'on appelle des tables (Colonnes X Lignes).
- LDD (Langage de définition des données) est un sous-ensemble du langage SQL pour manipuler les structures des tables d'une base de données, et non les données elles-mêmes.
- Commandes: create, alter, drop, rename

Création de table

La commande CREATE TABLE permet de créer une table dans la base de données courante.

Sa syntaxe est la suivante:

```
CREATE TABLE Nom_de_la_table (champ1 Type  
CONSTRAINT champ1 Propriété, champ2 Type  
CONSTRAINT Propriété ...);
```

Champ: Nom du champ de la table (colonne)

Type: Type de données

Propriété: La contrainte d'intégrité

Création de table

Les types de données

- **INTEGER**: Ce type permet de stocker des entiers signés codés sur 4 octets.
- **BIGINT**: Ce type permet de stocker des entiers signés codés sur 8 octets.
- **REAL**: Ce type permet de stocker des réels comportant 6 chiffres significatifs codés sur 4 octets.
- **DOUBLE PRECISION**: Ce type permet de stocker des réels comportant 15 chiffres significatifs codés sur 8 octets.
- **BOOLEAN**: Ce type de données permet de stocker des valeurs Booléenne.

Création de table

Les types de données

- **NUMERIC**[(précision,[longueur])]: Ce type de données permet de stocker des données numériques à la fois entières et réelles avec une précision de 1000 chiffres significatifs. longueur précise le nombre maximum de chiffres significatifs stockés et précision donne le nombre maximum de chiffres après la virgule.
- **CHAR**(longueur): Ce type de données permet de stocker des chaînes de caractères de longueur fixe. longueur doit être inférieur à 255, sa valeur par défaut est 1.
- **VARCHAR**(longueur): Ce type de données permet de stocker des chaînes de caractères de longueur variable. longueur doit être inférieur à 2000, il n'y a pas de valeur par défaut.
- **DATE**: Ce type de données permet de stocker des données constituées d'une date.

Commentaires

Il est possible d'inclure des retours chariots, des tabulations, espaces et commentaires **sur une ligne** précédée de deux tirets --, sur **plusieurs lignes** entre /* et */

```
CREATE TABLE
-- nom de la table
Test(
-- description
colonne NUMBER(38,8)
)
/* fin,
ne pas oublier le point-virgule. */
;
```

Convention

- La casse n'a pas d'importance au niveau des mots-clés de SQL, des noms de tables, colonnes, index, etc.
- il est préférable d'utiliser les conventions suivantes :
 - Tous les mots-clés de SQL sont notés en MAJUSCULES ;
 - Les noms de tables sont notés en Minuscules (excepté la première lettre) ;
 - Les noms de colonnes et de contraintes en minuscules.

Exemple de création de table

```
CREATE TABLE Etudiant  
( Code NUMBER(4),  
  Nom CHAR(10),  
  Prenom CHAR(20),  
  ville CHAR(15),  
  Date_Naissance DATE ) ;
```


Contraintes

Les contraintes ont pour but de programmer des règles de gestion au niveau des colonnes des tables.

Exemple : une note doit être comprise entre 0 et 20

Deux manières pour déclarer les contraintes :

- En même temps que la colonne dites **en ligne**.
- Après la déclaration des colonne, des contraintes peuvent être personnalisées **par un nom** (peuvent concerner plusieurs colonnes).

Contraintes : UNIQUE

CONSTRAINT *nomContrainte*

UNIQUE (*colonne1* [,*colonne2*] ...)

- La contrainte **UNIQUE** impose une valeur distincte au niveau de la table
- Les valeurs nulles font exception à moins que NOT NULL soit aussi appliquée sur les colonnes.

Contraintes : PRIMARY KEY

CONSTRAINT *nomContrainte*

PRIMARY KEY (*colonne1* [,*colonne2*]...)

- La contrainte PRIMARY KEY déclare la clé primaire de la table. Un index est généré automatiquement sur la ou les colonnes concernées.
- Les colonnes clés primaires ne peuvent **être ni nulles ni identiques** (en totalité si elles sont composées de plusieurs colonnes).

Contraintes : FOREIGN KEY

CONSTRAINT *nomContrainte*

FOREIGN KEY (colonne1 [,colonne2]...)

REFERENCES *nomTablePere* (colonne1
[,colonne2]...)

- La contrainte FOREIGN KEY déclare une clé étrangère entre une table fille et une table mère.
- Ces contraintes définissent l'intégrité référentielle.

Contraintes : CHECK

CONSTRAINT *nomContrainte*

CHECK (*condition*)

- La contrainte CHECK impose un domaine de valeurs ou une condition simple ou complexe entre colonnes
- Exemples :
 - CHECK (note BETWEEN 0 AND 20)
 - CHECK (grade='Copilote' OR grade='Commandant')).

Exemple

```
CREATE TABLE Pilote  
(matricule CHAR(6), nom CHAR(15),  
nbHVol NUMBER(7,2), compa CHAR(4),  
CONSTRAINT pk_Pilote PRIMARY KEY(matricule),  
CONSTRAINT nn_nom CHECK (nom IS NOT NULL),  
CONSTRAINT ck_nbHVol CHECK (nbHVol BETWEEN 0 AND 20000),  
CONSTRAINT un_nom UNIQUE (nom),  
CONSTRAINT fk_Pil_compa_Comp FOREIGN KEY (compa)  
REFERENCES Compagnie(comp));
```

Remarque

- **DESC** (raccourci de DESCRIBE) est une commande SQL*Plus. Elle permet d'extraire la structure brute d'une table.
- **L'ordre de création** des tables est important quand on définit les contraintes en même temps que les tables. Il faut créer d'abord les tables « **pères** » **puis** les tables « **fils** ».
- Le script de **destruction** des tables suit le raisonnement **inverse**.

Les index

- Un index Oracle permet **d'accélérer** l'accès aux données d'une table.
- Le **but** principal d'un index est **d'éviter de parcourir** une table séquentiellement du premier enregistrement jusqu'à celui visé.
- **Problème** : si on cherche le nom «laila» dans une table non indexée de plus de cent millions d'enregistrements...

Création d'un index

Privilège demandé : CREATE INDEX ou CREATE ANY INDEX

La syntaxe de création :

CREATE INDEX

[UNIQUE] *nomIndex*

ON *nomTable* (*colonne* [ASC / DESC] ...) ;

- UNIQUE permet de créer un index qui ne supporte pas les doublons.
- ASC et DESC précisent l'ordre (croissant ou décroissant).

Création d'un index

- Il est conseillé de créer des index sur des colonnes (majoritairement des clés étrangères) utilisées dans les clauses de jointures
- Les index sont pénalisants lorsqu'ils sont définis sur des colonnes très souvent modifiées ou si la table contient peu de lignes.

Destruction

Avec CASCADE CONSTRAINTS

**DROP TABLE Compagnie CASCADE
CONSTRAINTS;**

DROP TABLE Pilote;

Les « fils » puis les « pères »

DROP TABLE Pilote;

DROP TABLE Compagnie;

Remarque : DROP TABLE supprime aussi les **index** associés.

Renommer une table

- Pour renommer une table on utilise la commande " RENAME " .

RENAME Pilote **TO** Naviguant;

Ou encore :

ALTER TABLE Pilote **RENAME TO** Naviguant;

- Alter table permet de changer dans la structure d'une table.

Ajout de colonnes

- La directive **ADD** de l'instruction ALTER TABLE permet d'ajouter une nouvelle colonne à une table.
- Cette colonne est initialisée à NULL pour tous les enregistrements (à moins de spécifier une contrainte DEFAULT).
- Exemples :

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

```
ALTER TABLE Pilote
```

```
ADD (compa VARCHAR2(4) DEFAULT 'AF',
```

```
ville VARCHAR2(30) DEFAULT 'Rabat' NOT NULL);
```

Modifier le type des colonnes

- La directive **MODIFY** de l'instruction **ALTER TABLE** modifie le **type** ou les **contraintes** en ligne d'une colonne (tant que les données contenues sont compatibles avec la nouvelle modification).

- Exemples :

ALTER TABLE Pilote

MODIFY compa **VARCHAR(6) DEFAULT 'SING';**

ALTER TABLE Pilote

MODIFY compa **CHAR(4) NOT NULL;**

Supprimer des colonnes

- ➡ Une colonne est un cas particulier d'une table, alors on utilise la commande DROP comme il est le cas pour les tables.

ALTER TABLE Pilote DROP COLUMN adresse;

- ➡ Il n'est pas possible de supprimer avec cette instruction :
 - ✓ des clés primaires référencées par des **clés étrangères**
 - ✓ des colonnes à partir desquelles **un index** a été construit
 - ✓ toutes les colonnes d'une table

Ajout et suppression de contraintes

➤ Ajout de contraintes :

ALTER TABLE Avion

ADD (

CONSTRAINT nn_proprio CHECK (proprio IS NOT NULL),

CONSTRAINT fk_Avion_comp_Compag FOREIGN KEY(proprio) REFERENCES Compagnie(comp));

➤ Suppression de contraintes :

ALTER TABLE Avion DROP CONSTRAINT nn_proprio;

ALTER TABLE Avion DROP CONSTRAINT pk_Avion CASCADE;