Instructions SQL

Première partie : Langage de description et de gestion des données

Quelques instructions et leur syntaxe

1. Introduction

Trois principales catégories d'instructions.

Instructions de création :

- Création de table
- Création de domaine
- Création de vue
- Création d'assertion
- Création de motif
- Création de jeu de caractères
- Création de traducteur

Nous ne verrons que la création de table et de domaine.

Instructions de mise à jour :

- Suppression de table
- Suppression de domaine
- Suppression d'assertion
- Suppression de motif
- Suppression de jeu de caractères
- Suppression de traducteur
- Modification de domaine
- Modification de table (structure)

Nous verrons les suppressions et modifications de table et de domaine

Instructions d'insertion et de suppression et de mise à jour d'enregistrements:

- Insertion d'enregistrements dans des tables.
- Suppression d'enregistrements dans des tables.
- Modification de valeurs dans des enregistrements existant dans des tables.

2. Création de table CREATE TABLE

CREATE [{GLOBAL | LOCAL} TEMPORARY] TABLE <nom-table>
(<élément-table> [, <élément table >}…])
[ON COMMIT {PRESERVE | DELETE} ROWS]

si GLOBAL ou LOCAL ne sont pas spécifiés alors c'est une table de la base sinon ce doit être une table temporaire.

Une table temporaire globale a une copie pour chaque programme et est disponible pour chaque module de ce programme, alors qu'une table temporaire locale a une copie pour chaque module de chaque programme.

La clause ON COMMIT définit ce qui arrive aux enregistrements d'une table temporaire quand la transaction se termine (sans échec). PRESERVE indique que les enregistrements sont sauvegardés, DELETE signifie qu'il faut vider la table après chaque transaction.

<élément table > : := <définition colonne> | <définition de contrainte de table>

<définition colonne> : := <nom attribut> {<type de donnée> | <nom-domaine>}
[<clause de défaut>]
[<définition de contrainte de colonne>...]
[<clause de motif>]

2.1 Différents types de données

Données alphanumériques

CHARACTER
Caractère unique
CHARACTER VARYING (nombre)
Chaîne de longueur variable
CHARACTER (nombre)
Chaîne de longueur fixe

On peut rajouter un complément :
CHARACTER SET nom- del'ensemble
Ex : CHARACTER (40) CHARACTER SET Kanji

1.1.1. Données binaires

BIT [VARYING] [(<longueur>)]
Littéraux binaires
B'chaîne littérale'
Exemple : B'001001101'

1.1.2. Données numériques

1.1.2.1. Données numériques exactes

Nombres avec des représentations exactes
NUMERIC [(<précision> [,<échelle>)]
DECIMAL[<précision> [,<échelle>)]
INTEGER
SMALLINT
Exemples :
NUMERIC (5) nombre sur 5 positions
NUMERIC (5,2) nombre sur cinq positions dont la valeur doit être multipliée par 10 puissance 2.

1.1.2.2. Données numériques approximatives

Plutôt des nombres réels.
FLOAT [(<précision>)]
REAL
DOUBLE PRECISION

1.1.2.3. Données temporelles

Trois types de données temporelles :
DATE
Qui accepte les champs
YEAR

MONTH
DAY
Syntaxe : DATE
TIME
Qui accepte les champs
HOUR
MINUTE
SECOND valeur entière et fractions de seconde
Syntaxe :
TIME [(<précision>)][WITH TIME ZONE]
L'option WITH TIME ZONE inclut les champs
TIMEZONE_HOUR (heure universelle)
TIMEZONE_MINUTE
Exemples :
TIME
TIME (2) jusqu'au centième de seconde
TIME (3) WITH TIME ZONE jusqu'au millième de seconde en temps universel
TIMESTAMP
qui accepte tous les champs depuis year jusqu'à second.
Syntaxe :
TIMESTAMP [(<précision>)][WITH TIME ZONE]
Les littéraux sont précédés du nom de type.
Exemple DATE'2001-04-16'
Il y a aussi des types de données temporelles sous forme d'intervalles.

1.2. Clause de défaut

Spécification d'une valeur par défaut à l'insertion d'un nouvel enregistrement.
DEFAULT {
<litteral>
| < fonction de valeur temporelle> | USER | CURRENT_USER |SESSION_USER
|SYSTEM_USER | NULL }
NULL est la valeur par défaut si aucune clause de défaut n'est spécifiée.

1.3. Différents types de contraintes

On peut avoir les contraintes (qui contrôlent l'intégrité) directement dans la définition des colonnes (attributs) ou ensuite, associées à la table. Elles ne sont simplement pas placées de la même façon (pas la même syntaxe exactement).
NOT NULL
UNIQUE/ PRIMARY KEY
FOREIGN KEY/REFERENCES
CHECK

1.3.1. Contraintes sur les colonnes

Syntaxe générale
<contrainte sur colonne > : := [CONSTRAINT <nom-de-contrainte]
{NOT NULL
| <contrainte d'unicité>| <contrainte de référence externe>| <contrainte d'intégrité>}
[<attributs de la contrainte>]

Les attributs de la contrainte ne seront pas développés ici.

NOT NULL

NOT NULL impose que l'attribut (colonne) doit toujours avoir une valeur.

Exemple :

```
CREATE TABLE personne
(
nom    CHARACTER (40) NOT NULL,
…)
```

l'enregistrement ne peut être stocké que si la rubrique nom a été remplie.

UNIQUE, PRIMARY KEY (contrainte d'unicité)

Permet de vérifier que deux enregistrements d'une même table ne peuvent avoir la même valeur (l'attribut est en partie gauche de dépendance fonctionnelle).

Par exemple si à la personne on veut rajouter un unique numéro INSEE

```
CREATE TABLE personne
(
nom    CHARACTER (40) NOT NULL,
Insee   NUMERIC(12) UNIQUE,
…)
```

La contrainte PRIMARY KEY a le même effet, mais spécifie en plus que l'attribut en question est la clé du schéma de relation.

Exemple

```
CREATE TABLE personne
(
num    NUMERIC (5) PRIMARY KEY,
nom    CHARACTER (40) NOT NULL,
Insee   NUMERIC(12) UNIQUE,
…)
```

REFERENCES (contrainte de référence externe)

Permettant la connexion de deux tables à travers l'importation d'une clé étrangère.

Syntaxe

```
REFERENCES <nom-table> [(<nom-de-colonne>)]
                [MATCH {FULL | PARTIAL }]
                [<action a réaliser>]
```

Le nom de la table est celui de la table à connecter.

Le nom-de-colonne est le nom de la clé étrangère (qui doit avoir comme contrainte PRIMARY KEY dans sa table)

Exemple :

```
CREATE TABLE Departement
(
nomdep          CHARACTER (5) PRIMARY KEY,
…)
CREATE TABLE personne
(
num    NUMERIC (5) PRIMARY KEY,
nom    CHARACTER (40) NOT NULL,
Insee   NUMERIC(12) UNIQUE,
nomdep          CHARACTER (5) REFERENCES  Departement (nomdep),
…)
```

CHECK (contrainte d'intégrité)
Permet de vérifier rapidement l'intégrité des valeurs pour certains attributs.
Syntaxe :
CHECK (<conditions de recherche>)
Exemple :

```
CREATE TABLE personne
(
num    NUMERIC (5) PRIMARY KEY,
nom    CHARACTER (40) NOT NULL,
Insee  NUMERIC (12) UNIQUE,
sexe   CHARACTER CHECK (sexe IN ('M' ;'F')),
nomdep        CHARACTER (5) REFERENCES  Departement (nomdep),
…)
```

1.1.2. Contraintes sur les tables

Pratiquement les mêmes que sur les colonnes, mais permettent une plus grande souplesse d'écriture, ont une syntaxe beaucoup plus complète et une échelle d'action valable sur plusieurs attributs à la fois.
Syntaxe :
<définition contrainte de table> : := [CONSTRAINT <nom-de-contrainte>]
[<contrainte d'unicité> | <contrainte de référence externe > | <contrainte de validité>
<attributs de contraintes>]

1.1.1.1. Contrainte d'unicité :

Par UNIQUE et PRIMARY KEY.
Syntaxe :
UNIQUE ou | PRIMARY KEY (<nom d'attribut> [{,<nomd'attribut>}…])
Exemples :

```
CREATE TABLE personne
(
num             NUMERIC (5),
nom             CHARACTER (40) NOT NULL,
adresse         CHARACTER VARYING (200),
Insee           NUMERIC(12) ,
sexe            CHARACTER ,
UNIQUE (Insee, adresse)
PRIMARY KEY (num)
…)
```

1.1.1.2. Contrainte de référence externe

Permet de noter les contraintes de clé étrangère. La syntaxe est assez complexe car elle doit intégrer les problèmes de mise à jour concomittante des deux tables liées.
FOREIGN KEY (<clés etrangères>)
REFERENCES <tables concernées>
[MATCH { FULL | PARTIAL}]
[<actions de référence à réaliser>]

<actions de référence à réaliser> : := <règle de mise à jour>[<règle de suppression>] |
<règle de suppression> >[<règle de mise à jour>]

<règle de mise à jour> : := ON UPDATE <action de référence>
<règle de suppression> : := ON DELETE <action de référence>
<action de référence> : := [CASCADE | SET NULL |SET DEFAULT | NO ACTION]
Dans les deux cas, NO ACTION signifie ne rien faire.

Si règle de mise à jour :

CASCADE	Les clés étrangères des lignes qui correspondent dans les deux tables sont mises à jour à la même valeur (celle de la clé mise à jour)
SET NULL	En relation avec MATCH FULL. Met à « null » la colonne correspondant à la clé importée lorsque la table a été mise à jour.
SET DEFAULT	Met à la valeur de défaut la colonne correspondant à la clé importée.

Si règle de suppression

CASCADE	Les occurrences qui sont appariées sont supprimées
SET NULL	Les clés étrangères correspondant aux occurrences appariées sont mise à « null ».
SET DEFAULT	Les clés étrangères correspondant aux occurrences appariées sont mises à la valeur de défaut correspondante.

Exemple :

```
CREATE TABLE personne
(
num             NUMERIC (5),
nom             CHARACTER (40) NOT NULL,
adresse         CHARACTER VARYING (200),
Insee           NUMERIC(12) ,
sexe            CHARACTER ,
nomdep          CHARACTER (5) ,
UNIQUE (Insee, adresse)
PRIMARY KEY (num)
FOREIGN KEY (nomdep) REFERENCES departement
ON DELETE CASCADE
ON UPDATE CASCADE
…)
```

1.1.1.3. Contrainte d'intégrité

Syntaxe
CHECK (<conditions à vérifier>)
Dans les conditions on peut avoir des requêtes SQL (à voir dans la deuxième partie.
Exemple de contraintes sur les tables mélangées à des contraintes sur colonnes:

```
CREATE TABLE employe
(
num-emp         NUMERIC (5) CONSTRAINT Employe_PK PRIMARY KEY,
numdep          NUMERIC (3),
```

nom CHARACTER (40) NOT NULL,
adresse CHARACTER VARYING (200),
sexe CHARACTER NOT NULL,
telperso tel,
telburo tel UNIQUE,
indice-sal NUMERIC (6) NOT NULL
commission NUMERIC (3,2),
FOREIGN KEY (numdep) REFERENCES departement
CHECK (sexe IN ('M', 'F')),

« tel » correspond à un non de domaine.

3. Création de domaine CREATE DOMAIN

Syntaxe
CREATE DOMAIN <nom de domaine> AS <type de données>
[<clauses de défaut>]
[<contraintes de domaine> ...]
[<clause de motif]

<contrainte de domaine> : := [CONSTRAINT <nom-de contraintes>]
<contrainte d'intégrité>[<attributs de contrainte>]
Exemples
CREATE DOMAINE date-acquisition AS DATE DEFAULT CURRENT_DATE
CREATE DOMAINE tel AS CHARACTER (13)
CHECK (
VALUE LIKE '%-%-%-%-%'
OR VALUE LIKE '----')

4. Suppression de table DROP TABLE

Syntaxe :
DROP TABLE <nom-table> {RESTRICT | CASCADE}
RESTRICT signifie que la table ne doit pas être référencée ailleurs (dans d'autres tables domaines ou vues)
CASCADE signifie que la suppression de la table entraîne aussi l'effacement de la contrainte FOREIGN KEY qui référencerait cette table dans d'autres tables.
Exemple
DROP Personne RESTRICT fonctionne
Mais DROP Departement RESTRICT ira en échec
Alors que :
DROP Departement CASCADE va entraîner l'effacement de contrainte FOREIGN KEY (nomdep)REFERENCES Departement

5. Suppression de domaine DROP DOMAIN

Syntaxe :
DROP DOMAIN <nom-domaine> {RESTRICT | CASCADE}
RESTRICT peut être utilisé si le domaine n'est utilisé par aucune table, sinon, il y a échec.
CASCADE va entraîner les actions suivantes :

- Le nom de domaine est remplacé par le type de donnée du domaine
- Si le domaine a une contrainte alors un équivalent de contrainte sur colonne est ajouté

- Si l'attribut (colonne) n'a pas de clause de défaut, c'est celle du domaine (si elle existe) qui est reportée sur la colonne.

6. Modification de table ALTER TABLE

Pour modifier la structure (et non pas le contenu) d'une table ;
Plusieurs modifications existent :
Ajouter/supprimer une colonne (un attribut)
Ajouter/supprimer une contrainte de table
Mettre /enlever une valeur de défaut pour une colonne

6.1. Ajouter une colonne

Syntaxe
ALTER TABLE <nom-table> ADD [COLUMN] <définition de colonne>
Exemple
ALTER TABLE Personne ADD datenaissance DATE
La date de naissance est la dernière colonne de la table et elle est mise à « null ».

1.2. Supprimer une colonne

ALTER TABLE <nom-table> DROP [COLUMN] <nom colonne> {RESTRICT | CASCADE}
Si RESTRICT est spécifié la colonne ne doit pas être référencée ailleurs.
Si CASCADE est spécifié alors cela permet d'effacer la colonne dans les tables où elle est référencée.

1.3. Ajouter une contrainte de table

ALTER TABLE <nom-table> ADD <définition de contrainte de table>
Exemple :
Si on a rajouté la date de naissance à la table Personne alors
ALTER TABLE personne ADD
CHECK (datenaissance YEAR < CURRENT_DATE)
Vérifie que l'année de naissance est inférieure a l'année en cours.

1.4. Supprimer une contrainte de table

ALTER TABLE <nom-table> DROP <nom de contrainte>{RESTRICT | CASCADE}
RESTRICT ne peut être spécifié que pour des contraintes de type UNIQUE
Si CASCADE est spécifié alors toutes les contraintes de référence dépendantes sont aussi supprimées.

1.5. Mettre une valeur de défaut à une colonne

Syntaxe
ALTER TABLE <nom-table> ALTER [COLUMN] <nom colonne> SET <clause de défaut>
Exemple
ALTER TABLE enployé ALTER COLUMN Indice-sal SET DEFAUT 100

1.6. Supprimer une valeur de défaut d'une colonne

Syntaxe :
ALTER TABLE <nom-table> ALTER [COLUMN] <nom colonne> DROP DEFAULT

7. Insertion d'un enregistrement (ligne) dans une table INSERT INTO

Syntaxe
INSERT INTO <nom-table>
[(<nom-colonne> [{,<nom-colonne>}…])] <requête> | DEFAULT VALUES

DEFAULT VALUES crée une ligne entière avec les valeurs par défaut des colonnes
Si on veut insérer un enregistrement tout entier, il suffit de mettre les valeurs correspondant à chaque terme du n-uplet précédé du mot-clé VALUES.
Exemple :
Pour la table personne créée de la manière suivante

```
CREATE TABLE personne
(
num             NUMERIC (5),
nom             CHARACTER (40) NOT NULL,
adresse         CHARACTER VARYING (200),
sexe            CHARACTER ,
UNIQUE (adresse)
PRIMARY KEY (num)
ON DELETE CASCADE
ON UPDATE CASCADE
…)
```

On insère des enregistrements un à un par :
INSERT INTO personne VALUES (135,'Durand ','13 rue des pensées, 34000 Montpellier','M')
En ce qui concerne la requête, elle sera vue dans la deuxième partie.

8. Suppression d'enregistrements dans une table DELETE FROM

Syntaxe
DELETE FROM <nom-table> [WHERE <conditions de recherche>]

Si DELETE FROM est utilisé tout seul, alors tous les enregistrements de la table sont supprimés.
Si WHERE est spécifié alorson ne supprime que les enregistrements vérifiant les conditions de recherche.
<conditions de recherche> : :=<expression logique simple> | <expression logique avec requête>
Exemple (avec une expression logique)
DELETE FROM personne WHERE sexe ='M'
DELETE FROM personne WHERE num=135

9. Modification d'enregistrements dans une table UPDATE

Syntaxe

```
UPDATE <nom-table>
        SET <nom-colonne> = {<expression de valeur> | NULL | DEFAULT}
                [{,<nom-colonne> = {<expression de valeur> | NULL | DEFAULT}}…
                [WHERE <conditions de recherche>]
```

Si UPDATE est utilisé tout seul, alors tous les enregistrements de la table sont modifiés.
Exemples Soit la table employe créée ainsi

```
CREATE TABLE employe
(
num-emp      NUMERIC (5) CONSTRAINT Employe_PK PRIMARY KEY,
numdep       NUMERIC (3),
nom          CHARACTER (40) NOT NULL,
adresse      CHARACTER VARYING (200),
sexe         CHARACTER NOT NULL,
telperso     tel,
telburo      tel UNIQUE,
indice-sal   NUMERIC (6) NOT NULL
commission   NUMERIC (3,2),
FOREIGN KEY (numdep) REFERENCES departement
CHECK (sexe IN ('M', 'F')),
```

Des exemples de modification :

```
UPDATE employe SET indice-sal =indice-sal +5, commission=commission+indice-sal/10
UPDATE Employé SET indice-sal =indice-sal+5 WHERE indice-sal <200
UPDATE Employé SET commission=NULL WHERE numdep=1
```