

Les ordres SQL

SELECT	Recherche des données
INSERT UPDATE DELETE	Langage de manipulation des données(LMD)
CREATE ALTER DROP	Langage de définition des données(LDD)
COMMIT ROLLBACK	Contrôle des transactions
GRANT REVOKE	Langage de contrôle des données (LCD)

L'ordre CREATE TABLE

Pour créer une table, l'utilisateur doit disposer du privilège CREATE TABLE et d'un espace de stockage dans la quelle il pourra créer des objets.

Syntaxe :

***CREATE TABLE [schema.]nom_table
(column datatype [default expr],.....) ;***

[] : Pour dire que c'est facultatif

schema : nom du propriétaire

default expr : spécifie une valeur par défaut à utiliser en cas d'omission d'une valeur dans l'ordre INSERT

expr :

- *valeurs autorisées : expressions, fonctions SQL*
- *valeurs non autorisées : noms d'autres colonnes*

column : nom de la colonne

datatype : type de données et longueur de la colonne

Exemple :

```
SQL> CREATE TABLE dept  
      (deptno number(2),  
       dname varchar2(15) ,  
       loc varchar2(14)) ;
```

Pour vérifier la création de la table on utilise, on utilise la commande **DESCRIBE** ou **DESC**.

Exemple :

SQL> **DESC** dept

Le resultat est :

Name	NULL	TYPE
deptno		number(2)
dname		varchar2(15)
loc		varchar2(14)

Les types de données les plus utilisées

Types de données	Description
VARCHAR2(size)	Données caractères de longueur variable. Longueur minimal est 1. Longueur maximal est 4000.
CHAR(size)	données caractères de longueur fixe
NUMBER(p,s)	numérique de longueur variable.
LONG	données caractères de longueur variable jusqu'à 2 giga-octets
DATE	valeurs de date et d'heure

L'ordre ALTER TABLE

Après avoir crée les tables, il arrive qu'on doit modifier la structure pour ajouter une colonne oubliée ou qu'on décide de changer une définition de colonne. Cela est possible grâce à l'ordre **ALTER TABLE** .

On peut ajouter des colonnes à une table en utilisant l'ordre **ALTER TABLE** avec la clause **ADD**. La syntaxe est la suivante :

```
ALTER TABLE nom_table  
ADD ( Column datatype [default expr]  
[,Column datatype]...);
```

On peut aussi modifier les colonnes existantes d'une table en utilisant l'ordre **ALTER TABLE** avec la clause **ADD**. La syntaxe est la suivante :

```
ALTER TABLE nom_table  
MODIFY ( Column datatype [default expr]  
[,Column datatype]...);
```

Exemple :

```
SQL>ALTER TABLE dept
      ADD      ( tel varchar2(15)) ;
Table altered.
```

Cette nouvelle colonne sera placée à la fin. Si la table contient déjà des lignes, la nouvelle colonne sera initialisée à Null pour toutes les lignes sauf si l'on précise une valeur par défaut.

deptno	dname	loc	tel
10	informatique	pavillon principal	
20	math	pavillon principal	

L'ordre DROP TABLE

Pour supprimer une table, on utilise l'ordre DROP TABLE.

Syntaxe :

DROP TABLE [schema.]nom_table ;

Cet ordre permet de

Exemple :

DROP TABLE dept ;

NB : Cet ordre ne peut être annulé après exécution.

Les contraintes

Oracle fait appel à des contraintes pour empêcher l'entrée de données incorrectes dans des tables.

On utilise les contraintes pour :

- Appliquer des règles au niveau d'une table chaque fois qu'une ligne est insérée, mise à jour ou supprimée dans cette table. La contrainte doit être satisfaite pour que l'opération réussisse.
- Empêcher la suppression d'une table si il y a des dépendances avec d'autres tables.

Les types de contraintes valides dans Oracle :

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK

Syntaxe :

```
CREATE TABLE [schema].nom_table  
    (column datatype [default expr] [column_constraint],  
    .....  
    [table_constraint]);
```

column_constraint : contrainte d'intégrité incluse dans la définition de la colonne

table_constraint : contrainte d'intégrité incluse dans la définition de la table

Les contraintes peuvent être définies donc soit au niveau colonne, soit au niveau table sauf la contrainte NOT NULL qui ne peut être défini qu'au niveau colonne.

Exemple :

```
SQL> CREATE TABLE dept  
    (deptno number(2),  
    dname varchar2(15) ,  
    loc varchar2(14),  
    CONSTRAINT emp_empno_pk PRIMARY KEY(empno));
```

La contrainte NOT NULL :

Cette contrainte interdit la présence de valeurs NULL dans la colonne à la quelle elle s'applique. Par défaut, les colonnes qui ne sont pas associées à la contrainte NOT NULL peuvent contenir des valeurs NULL.

deptno	dname	loc
10	informatique	pavillon principal
20	math	pavillon principal
30	médecine	

Contrainte NOT NULL (aucune ligne ne peut avoir de valeur NULL dans cette colonne)	↑
Absence de contrainte NOT NULL (toute ligne ne peut avoir une valeur NULL dans cette colonne)	↑

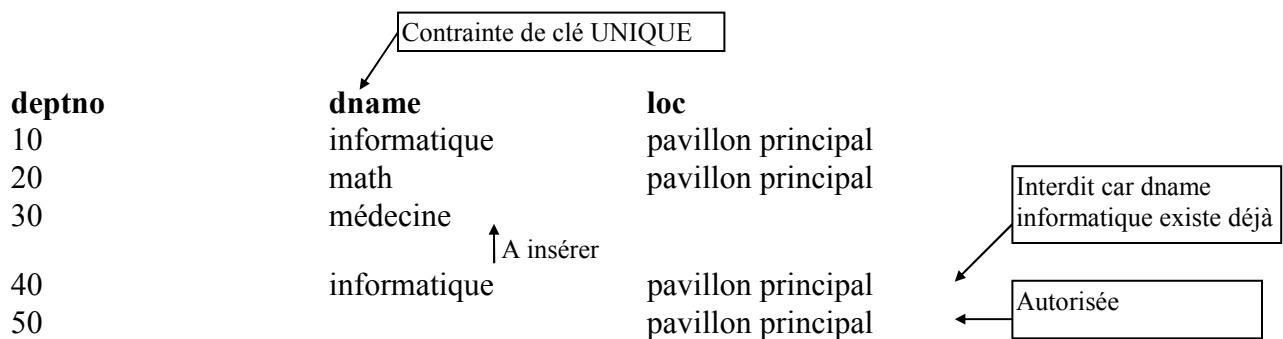
Exemple :

```
SQL> CREATE TABLE dept
      (deptno number(2),
       dname varchar2(15) NOT NULL,
       loc varchar2(14)) ;
```

La contrainte de clef UNIQUE :

Une contrainte d'intégrité de type clé UNIQUE exige que chaque valeur dans une colonne ou dans un ensemble de colonnes soit unique, c'est à dire qu'elle n'existe pas dans plusieurs lignes pour la colonne ou l'ensemble de colonne spécifiés.

Les contraintes UNIQUE autorisent la saisie des valeurs NULL, à moins qu'on définit des contraintes NOT NULL sur les mêmes colonnes.



Exemple de la contrainte clef UNIQUE définit au niveau table :

```
SQL> CREATE TABLE dept
      (deptno number(2),
       dname varchar2(15) ,
       loc varchar2(14),
       CONSTRAINT dept_dname_uk UNIQUE (dname)) ;
```

Exemple de la contrainte clef UNIQUE définit au niveau colonne :

```
SQL> CREATE TABLE dept
      (deptno number(2),
       dname varchar2(15) UNIQUE ,
       loc varchar2(14)) ;
```

La contrainte PRIMARY KEY :

Une contrainte PRIMARY KEY crée une clé primaire pour la table. Une seule clé primaire peut être créée par table. La contrainte PRIMARY KEY est une colonne ou un ensemble de colonnes qui identifie de manière unique chaque ligne d'une table. Elle établit une règle d'unicité de la

colonne ou d'une combinaison de colonnes et garantit qu'aucune colonne faisant partie de la clef primaire ne contient de valeur NULL.

deptno	dname	loc
10	informatique	pavillon principal
20	math	pavillon principal
30	médecine	
20	informatique	pavillon principal
		pavillon principal

↑ A insérer

CLE PRIMAIRE

Interdit car deptno 20 existe déjà

Interdit deptno est null

Exemple de la contrainte PRIMARY KEY définit au niveau table :

```
SQL> CREATE TABLE dept
      (deptno number(2),
       dname varchar2(15),
       loc varchar2(14),
       CONSTRAINT dept_deptno_pk PRIMARY KEY (deptno)) ;
```

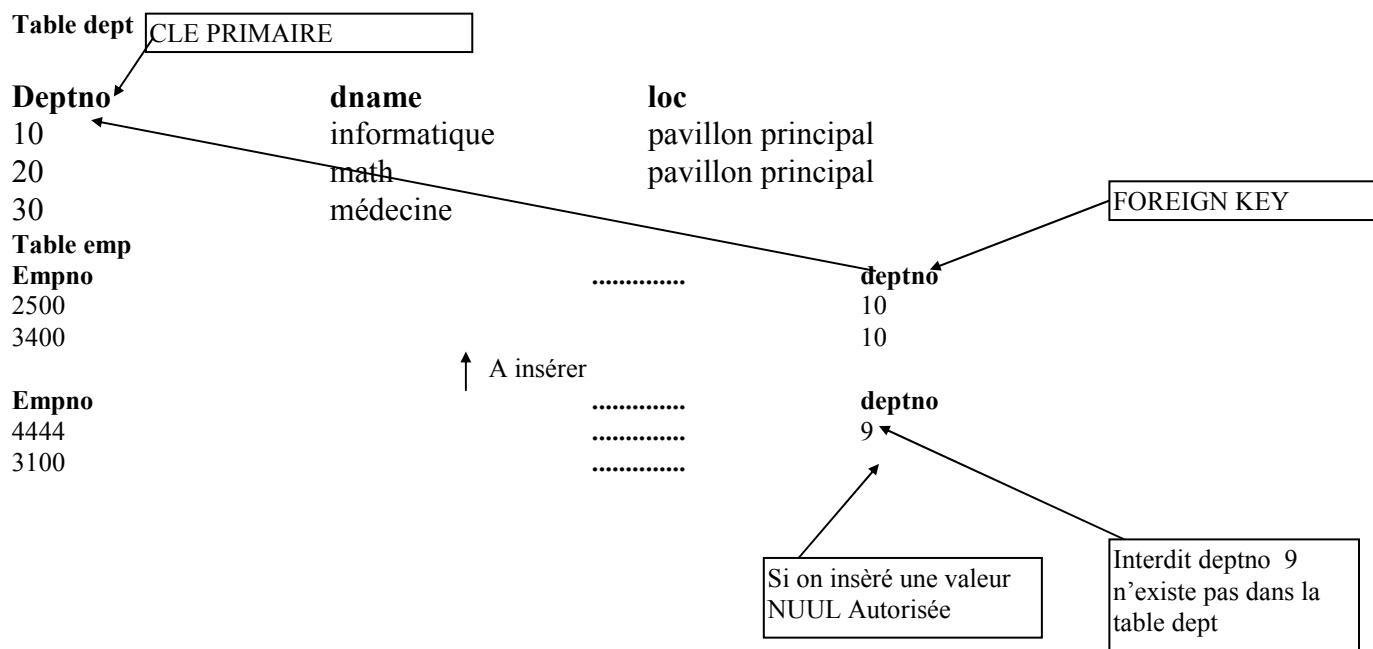
Exemple de la contrainte PRIMARY KEY définit au niveau colonne :

```
SQL> CREATE TABLE dept
      (deptno number(2) PRIMARY KEY ,
       dname varchar2(15),
       loc varchar2(14)) ;
```

La contrainte FOREIGN KEY :

La contrainte FOREIGN KEY, ou contrainte d'intégrité référentielle, désigne une colonne ou une combinaison de colonnes comme étant une clef étrangère et établit une relation avec une clef primaire d'une table différente. Dans l'exemple ci-dessous, la colonne deptno a été définie comme clef étrangère dans la table emp, elle fait référence à la colonne deptno de la table dept.

Une valeur de clef étrangère doit obligatoirement correspondre à une valeur existante de la table maître ou être NULL



Exemple de la contrainte FOREIGN KEY définit au niveau table :

```
SQL> CREATE TABLE emp
      (empno number(4),
      .....
      deptno number(2),
      CONSTRAINT emp_deptno_fk FOREIGN KEY (deptno)
      REFERENCES dept (deptno) ON DELETE CASCADE) ;
```

Exemple de la contrainte FOREIGN KEY définit au niveau colonne :

```
SQL> CREATE TABLE emp
      (empno number(4),
      .....
      deptno number(2) FOREIGN KEY REFERENCES dept (deptno) ON
      DELETE CASCADE) ;
```

FOREIGN KEY : définit la colonne dans la table détail

REFERENCES : identifie la table et la colonne de la table maître

ON DELETE CASCADE : autorise la suppression d'une ligne dans la table maître et des lignes dépendants dans la table détail.

La contrainte CHECK :

Elle définit une condition que chaque ligne doit obligatoirement satisfaire.

Exemple :

```
SQL> CREATE TABLE emp
      (empno number(4),
      .....
      deptno number(2),
      CONSTRAINT emp_deptno_ck CHECK (deptno BETWEEN 10 AND 99)) ;
```