

Memento des ordres SQL*Plus

Première partie

SQL comme langage de définition de données (LDD)

Types syntaxiques des attributs : VARCHAR2(n) CHAR[(n)] NUMBER[(n[,m])] DATE LONG

1 Création de relation

```
CREATE TABLE <nom_table>
    (nom_colonne1 <type1> [DEFAULT <expression1>]
    [, nom_colonne2 <type2> [DEFAULT <expression2>]]
    [, <contrainte1>[, <contrainte2>]])
```

où :

```
<contrainte1> = CONSTRAINT <nom_contrainte> <spec_contrainte> [<etat>]
<spec_contrainte> = PRIMARY KEY(<attribut1>[, <attribut2>, ...])
                    | FOREIGN KEY(<attribut1>[, <attribut2>, ...])
                      REFERENCES <nom_relation_associée>(<attribut1>[, <attribut2>, ...])
                      [ON DELETE <CASCADE|SET NULL>]
                    | CHECK(<nom_attribut | expression> <condition>)
<etat>= ENABLE | DISABLE
```

```
CREATE TABLE <nom_relation> [(liste_attributs>, <liste_contraintes>)]
AS <requete>
```

2 Ajout d'attributs et de contraintes dans une relation

```
ALTER TABLE <nom_table>
ADD ([nom_colonne1 <type1>] [DEFAULT <expression1>] [NOT NULL] [UNIQUE]
    [, <nom_colonne2> <type2> [DEFAULT <expression2>] [NOT NULL] [UNIQUE]...]
    [, <contrainte1> ...])
```

où

```
<contrainte1> = CONSTRAINT <nom_contrainte> <spec_contrainte> [<etat>]
<spec_contrainte> = PRIMARY KEY(<attribut1>[, <attribut2>, ...])
                    | FOREIGN KEY(<attribut1>[, <attribut2>, ...])
                      REFERENCES <nom_relation_associée>(<attribut1>[, <attribut2>, ...])
                      [ON DELETE <CASCADE|SET NULL>]
                    | CHECK(<nom_attribut | expression> <condition>)
```

```
<etat>=  ENABLE | DISABLE | VALIDATE | NOVALIDATE | ENABLE VALIDATE | ENABLE NOVALIDATE  
         |  DISABLE VALIDATE |  DISABLE NOVALIDATE
```

3 Modification de la définition d'un attribut

```
ALTER TABLE <nom_table>  
MODIFY  [(]<nom_colonne1> [<nouveau_type1>] [DEFAULT <expression1>] [NOT NULL]  
        [,<nom_colonne2> [<nouveau_type2>] [DEFAULT <expression2>] [NOT NULL] ... []]
```

4 Modification de l'état d'une contrainte

```
ALTER TABLE <nom_table>  
MODIFY CONSTRAINT <nom_contrainte> <etat_contrainte>
```

5 Suppression de contrainte dans une relation

```
ALTER TABLE <nom_table> DROP CONSTRAINT <nom_contrainte> [CASCADE]  
  
ALTER TABLE <nom_table> DROP UNIQUE(<nom_attribut>) [CASCADE]  
  
ALTER TABLE <nom_table> DROP PRIMARY KEY [CASCADE]
```

6 Suppression d'attribut dans une relation

```
ALTER TABLE <nom_table> SET UNUSED COLUMN <nom_attribut>  
  
ALTER TABLE <nom_table> SET UNUSED(<nom_attribut1>[, <nom_attribut2> ...])  
  
ALTER TABLE <nom_table> DROP COLUMN <nom_attribut> [CASCADE CONSTRAINTS]  
  
ALTER TABLE <nom_table> DDROP(<nom_attribut1>[, <nom_attribut2>, ...]) [CASCADE CONSTRAINTS]  
  
ALTER TABLE <nom_table> DROP UNUSED COLUMNS
```

7 Suppression de relation

```
DROP TABLE <nom_table> [CASCADE CONSTRAINTS]
```

8 Création/suppression de synonyme et changement du nom d'une relation

```
CREATE [PUBLIC] SYNONYM <nom_synonyme> FOR <nom_objet>  
  
DROP SYNONYM <nom_synonyme>  
  
RENAME <ancien_nom> TO <nouveau_nom>
```

9 Gestion de séquences

```
CREATE SEQUENCE <nom_sequence> [START WITH <valeur_initiale>]
                                [INCREMENT BY <valeur_increment>]
                                [MAXVALUE <valeur_maximale> | NOMAXVALUE]
                                [MINVALUE <valeur_minimale> | NOMINVALUE]
                                [CYCLE | NOCYCLE]
```

```
DROP SEQUENCE <nom_sequence>
```

```
ALTER SEQUENCE <nom_sequence> [INCREMENT BY <valeur_increment>]
                                [MAXVALUE <valeur_maximale> | NOMAXVALUE]
                                [MINVALUE <valeur_minimale> | NOMINVALUE]
                                [CYCLE | NOCYCLE]
```

10 Index sur les relations

```
CREATE [UNIQUE | BITMAP] INDEX <nom_index>
ON <nom_table> ([<nom_colonne1>[, <nom_colonne2>, ...])
```

```
ALTER TABLE <nom_index> RENAME TO <nouveau_nom>
```

```
DROP INDEX <nom_index>
```

11 Principales tables systèmes Oracle

ALL_CONS_COLUMNS	(OWNER, CONSTRAINT_NAME, TABLE_NAME, COLUMN_NAME,...)
ALL_CONSTRAINTS	(OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME, SEARCH_CONDITION,...)
ALL_INDEXES	(OWNER, INDEX_NAME, INDEX_TYPE, TABLE_OWNER, TABLE_NAME, TABLE_TYPE, UNIQUENESS, COMPRESSION)
ALL_OBJECTS	(OWNER, OBJECT_NAME, OBJECT_ID, DATA_OBJECT_ID, OBJECT_TYPE, CREATED, ...)
ALL_SEQUENCES	(SEQUENCE_OWNER, SEQUENCE_NAME, MIN_VALUE, MAX_VALUE, INCREMENT, CYCLE_FLAG)
ALL_SYNONYMS	(OWNER, SYNONYM_NAME, TABLE_OWNER, TABLE_NAME)
ALL_TAB_COLUMNS	(OWNER, TABLE_NAME, COLUMN_NAME, DATA_TYPE, DATA_LENGTH)
ALL_TABLES	(OWNER, TABLE_NAME, TABLESPACE_NAME)
ALL_VIEWS	(OWNER, VIEW_NAME, TEXT_LENGTH, TEXT,...)

Les tables de même nom préfixées par USER_ ont la même structure hormis l'attribut OWNER, et décrivent seulement les composants du schéma de l'utilisateur.

Pseudo-colonnes

<nom_sequence>.CURRVAL , <nom_sequence>.NEXTVAL , LEVEL , ROWID , ROWNUM , USER

Deuxième partie

SQL comme langage de manipulation de données (LMD)

12 Requêtes

```
<requete> =  SELECT <liste_resultat | * >
              FROM <liste_relations>
              [WHERE <liste_conditions>]
              [GROUP BY <liste_attributs_de_partitionnement>]
              [HAVING <liste_conditions_de_partitionnement>]]
              [ORDER BY <liste_attributs_a_trier>]
```

où :

```
liste_resultat = [DISTINCT] <attribut1 | expr1 | requete1> [<alias1>]
                  [, <attribut2 | expr2 | requete2> [<alias2>] ...]
liste_relations = <relation1 | vue1 | requete1> [alias1]
                  [, <relation2 | vue 2 | requete2> |alias2>]...
liste_conditions = [NOT] <condition1> [AND | OR <condition2> ...]
```

Condition de sélection :

```
<condition1> = <attribut [(+)] | expression> <comparateur | predicat_conditionnel> <constante>
```

```
<predicat_cond> =  IS NULL | IN | BETWEEN ... AND | LIKE
                   | IS NOT NULL | NOT IN | NOT BETWEEN | NOT LIKE
```

Condition de jointure prédicative :

```
<conditionj> = <attribut1[(+)] | expr1> <comparateur> <attribut2[(+)] | expr2 >
```

Condition de jointure imbriquée :

```
<conditionji> =  <expression1>[, <expression2>,...]  $\theta$  (<requete>)
                  | <expression1>[, <expression2>,...]  $\theta$  ANY | IN (<requete>)
                  | <expression1>[, <expression2>,...]  $\theta$  ANY (<requete>)
```

13 Calculs verticaux (fonctions agrégatives)

```
<nom_fonction> ([DISTINCT] <nom_colonne>)
```

où :

```
<nom_fonction> = SUM | AVG | COUNT | MAX | MIN | STDDEV | VARIANCE
```

14 Tri des résultats

```
ORDER BY <expression1> [ASC | DESC] [NULLS FIRST | NULLS LAST] [, <expression2> ...]
```

15 Opérateurs ensemblistes

```
<requete1>  
UNION | INTERSECT | MINUS  
<requete2>
```

16 Test d'absence ou d'existence de données

```
SELECT <liste_attributs>  
FROM <relation1> [<alias1>] [,<relation2> [<alias2>] ...]  
WHERE [<liste_conditions> AND | OR] [NOT] EXISTS (<sous_requete>)
```

17 Classifications ou partitionnement

```
ORDER BY <colonne1> [, <colonne2>, ...]  
HAVING <liste_conditions_classe>
```

18 Recherche dans une arborescence

```
SELECT <colonne1> [, <colonne2>, ...]  
FROM <table> [<alias>]  
[WHERE <liste_conditions>]  
CONNECT BY [PRIOR] <colonne1> = [PRIOR] <colonne2>  
[AND <condition_hierarchique>]  
[START WITH <condition_depart>]  
[ORDER BY LEVEL]
```

19 Mise à jour des données

```
UPDATE <nom_table>  
SET <nom_colonne1> = <expression1> [, <nom_colonne2> = <expression2>, ...]  
[WHERE <condition_selection>]
```

```
UPDATE <nom_table>  
SET (<nom_colonne1>[, <nom_colonne2>, ...]) = (SELECT <colonne1>[,<colonne2>, ...]  
FROM ... WHERE ...)  
[WHERE <condition_selection>]
```

```
INSERT INTO <nom_table> [(<liste_attributs>)]  
VALUES (<valeur1>[,<valeur2>, ...])
```

```
INSERT INTO <nom_table> [(<liste_attributs>)] <requête>
```

```
DELETE FROM <nom_table> WHERE <condition>
```

Troisième partie

SQL comme langage de contrôle des données (LCD)

20 Gestion des transactions

COMMIT, ROLLBACK

21 Création et suppression de rôles et d'utilisateurs

CREATE ROLE <nom_role> [*IDENTIFIED BY* <mot_de_passe>]

ALTER ROLE <nom_role> [*IDENTIFIED BY* <nouveau_mot_de_passe>]

DROP ROLE <nom_role>

CREATE USER <nom_utilisateur> [*IDENTIFIED BY* <mot_de_passe>]

DEFAULT TABLESPACE <nom_tablespace>

QUOTA <taille> *PROFILE* <nom_profil>

ALTER USER <nom_utilisateur> [*IDENTIFIED BY* <nouveau_mot_de_passe>]

DROP USER <nom_utilisateur>

22 Attribution ou suppression de privilèges

GRANT <systeme_privileges | *ALL* [*PRIVILEGES*]>

TO <liste_role_utilisateur | *PUBLIC*>

[*WITH ADMIN OPTION*]

où :

<systeme_privilege> = *CREATE ROLE* | *CREATE SEQUENCE* | *CREATE SESSION* | *CREATE SYNONYM*
| *CREATE PUBLIC* | *CREATE TABLE* | *CREATE USER* | *CREATE VIEW*

GRANT <liste_droits>

TO <nom_composant>

[*WITH GRANT OPTION*]

où :

<liste_droits> = *SELECT* | *INSERT* | *UPDATE* [nom_colonne1,...] | *DELETE* | *ALTER*
| *INDEX* | *REFERENCES* | *ALL* [*PRIVILEGES*]

GRANT <liste_roles_attribues>

TO <liste_roles_utilisateurs>

[*WITH ADMIN OPTION*]

23 Gestion de vues

```
CREATE [OR REPLACE] [[NO] FORCE]  
VIEW <nom_vue> [(alias1[,alias2,...])]  
AS <requete>  
[WITH CHECK OPTION | WITH READ ONLY]
```

```
ALTER VIEW <nom_vue> COMPILE
```

```
DROP VIEW <nom_vue>
```