

Langage SQL :**Création de tables**

CREATE TABLE [schéma.]nomtable
(colonne1 type1 [DEFAULT valeur1] [NOT NULL]
[,colonne2 type2 [DEFAULT valeur2] [NOT NULL]]
[CONSTRAINT nomContrainte1 TypeContrainte1]...);

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)

Poser un commentaire sur une table ou une colonne

COMMENT ON TABLE [schéma.]NomTable
COLUMN [schéma.]NomTable.nomColonne
IS 'MON COMMENTAIRE' ;

Obtenir les nom et descriptions des tables

Obtenir les noms des tables
SELECT TABLE_NAME FROM USER_TABLES ;
DESC nomtable ;

Supprimer une table

DROP TABLE [schéma.]NomTable [CASCADE
CONSTRAINTS] ;

Renommer une table

RENAME ancienNom TO nouveauNom ;
ALTER TABLE noTable RENAME TO NouveauNom ;

Ajouter une colonne

ALTER TABLE NomTable ADD NomColonne TypeCol
CONSTRAINTS nomContraintes typeContraintes ;

Renommer colonne

ALTER TABLE NomTable RENAME COLUMN Nomcolonne
TO Nouveau NomColonne ;

Modifier type des colonnes

ALTER TABLE Nomtable MODIFY Nomcolonne
TypeColonne Contraintes ;

Supprimer des colonnes

ALTER TABLE nomTable DROP COLUMN nomColonne ;

Création des séquences

CREATE SEQUENCE nomSequence
MINVALUE valeurMin
MAXVALUE valeurMax
START WITH valeurDeDebut
INCREMENT BY valeurIncrément
CYCLE|NOCYCLE ;

nomSequence.NEXTVAL;
nomSequence.CURVAL;

Insérer des données dans vos tables

INSERT INTO nomtable(colonne1,colonne2,...)
VALUES(valeur1,valeur2,...) ;

Modifier des données dans vos tables

UPDATE nomTable SET colonne='valeur' [WHERE
condition] ;

Supprimer des données dans vos tables

DELETE FROM nomTable [WHERE condition] ;

Sélection de données

SELECT colonnes FROM NomTable WHERE Conditions ;

Caractères jocker : * plusieurs caractères, ? Un seul caractère.

Les fonctions d'agrégation SQL

Syntaxe	Description
COUNT	Dénombrement des lignes
SUM	Somme des valeurs numériques
MIN	Valeur Numérique Minimale
MAX	Valeur Numérique Maximale
AVG	Moyenne des valeurs numériques

Les Alias : nomChamp AS nouveauNom

Les opérateurs de comparaisons : < = > <= >= <>

Comparateur de vacuité : IS [NOT] NULL

Bornes d'inclusions : BETWEEN v1 AND v2

Sélection dans une liste : [NOT] IN liste

Comparateur de motifs : LIKE '%d%'

Trie sur résultat :

DISTINCT
UNIQUE
ALL
ASC pour classer dans l'ordre croissant
DESC pour classe dans l'ordre décroissant
NULLS FIRST les valeurs nulles en haut
NULLS LAST les valeurs nulles en bas

Opérations ensemblistes :

MINUS
UNION
INTERSECT

Jopintures :

Equijointure : jointure entre 2 tables

Autojointure : Jointure avec elle même

On va trouver deux formes d'écritures, relationnelle (=) et SQL2 (JOIN...ON condition), l'utilisation de INNER devant JOIN est

optionnelle et est appliqué par défaut.

Jointure externe :

La directive de jointure externe (+) se place du côté de la table subordonnée.

Le sens de la directive de jointure externe LEFT ou RIGHT de la clause OUTER JOIN désigne la table dominante.

Jointure précedurales ou sosu requête :

ALL ⇔ a tous

ANY ⇔ au moins un

IN ⇔ = ANY

NOT IN ⇔ <> ALL

Clause de groupement : GROUP BY

2 contraintes impératives sur les colonnes de la clause SELECT ;

Toutes les colonnes de la clause GROUP BY doivent figurer sur la clause GROUP BY

Toutes les colonnes de la clause SELECT sans fonction d'agrégat, doivent figurer dans la clause GROUP BY

Clause de restrictions sur groupement: HAVING

Créer des Vues

CREATE OR REPLACE VIEW nomVue

(nomcolvue1,nomcolvue2,...)

AS

SELECT col1,col2,

FROM Nomtables

WHERE conditions

Accélérer les recherche grâce aux Index

CREATE INDEX {UNIQUE|BITMAP} [schéma.]nomIndex

ON [schéma.]nomTable ({colonne1 | expressionCol1} [ASC | DESC]);

Mettre en place les déclencheurs

CREATE [OR REPLACE] TRIGGER [schéma.]

nom_déclencheur

{

```

AFTER | BEFORE | INSTEAD OF
}
{
{
[ INSERT
[ OR DELETE
[ OR UPDATE [ OF nom_colonne [,
nom_colonneN ] ] ] ] ]
}
ON
{
{ [ schéma. ] nom_table }
|
{ [ NESTED TABLE colonne_emboîtée OF ]
[ schéma . ] nom_vue }
}
[ REFERENCING ]
{
[ OLD [ AS ] ancienne_colonne
| NEW [ AS ] nouvelle_colonne
| PARENT [ AS ] colonne_parente ]
}
}
[ FOR EACH ROW ] instruction_SQL;
}
|
{
{ Evénement_LDD [ OR EvénementN_LDD ] }
|
{ Evénement_Base_Données [ OR
EvénementN_Base_Données ]
}
}
ON
{
{ [ schéma. ] SCHEMA } | DATABASE
}
WHEN ( Condition )
{ instruction_PL/SQL
instruction_procédure };
}

```

Quelques fonctions Oracles :

Manipulations des cahines de caractères :

ASCII(c) Retourne la valeur du caractère ASCII équivalent.

CHR(n) Retourne le caractère ASCII équivalent

CONCAT (c1,c2) Equivalent à ||

INITCAP (c) Première lettre de chaque mot mis en majuscule

UPPER (c) Met en majuscule la chaîne

LENGTH(c) Longueur de la chaîne

REPLACE(c1,c2,c3) recherche c2 dans c1 et le remplace par c3

Manipulations des nombres :

ABS(n) valeur absolue

CEIL(n) plus petit entier

FLOOR(n) plus grand entier

ROUND(n,d) arrondi a d décimale la valeur n

SIGN(n) signe de n

TRUNC(n,d) coupure de n à d décimales

Manipulations des dates :

SYSDATE ou CURRENT_DATE Date courante

LAST_DAY(d) retourne le dernier jour du mois

Comparateurs de grandeurs :

GREATEST(exp1,exp2,exp3) retourne la plus grande des expressions

LEAST(exp1,exp2,exp3) retourne la plus petite des expression

NULLIF(exp1,exp2) retour NULL si exp1=exp2 sinon Exp1

NVL(exp1,exp2) converti exp1 en exp2 si exp1=NULL