

6 Créer et gérer des tables

Objectifs

- décrire les principaux objets de base de données
- créer des tables
- décrire les différents types de données utilisables pour les définitions de colonne
- modifier des définitions de table
- supprimer, renommer et vider des tables

Objets de base de données

Objet	Description
Table	Unité de stockage élémentaire, composée de lignes et de colonnes
Vue	Représentation logique de sous-ensembles de données issus d'une ou de plusieurs tables
Séquence	Générateur de valeurs numériques
Index	Améliore les performances de certaines interrogations
Synonyme	Permet d'affecter un autre nom à un objet

Règles d'appellation

Les noms de table et de colonne :

- doivent commencer par une lettre,
- peuvent comporter de 1 à 30 caractères,
- ne peuvent contenir que les caractères A à Z, a à z, 0 à 9, _, \$, et #,
- ne doivent pas être identiques à ceux d'un autre objet appartenant au même utilisateur,
- ne doivent pas correspondre à des mots réservés du serveur Oracle.

Instruction CREATE TABLE

- Vous devez disposer :
 - du privilège CREATE TABLE,
 - d'un espace de stockage

```
CREATE TABLE [schema.]table  
  (column datatype [DEFAULT expr][, ...]);
```

- Vous devez indiquer :
 - le nom de la table,
 - le nom, le type de données et la taille des colonnes.

ORACLE

6-5

Référencer des tables appartenant à un autre utilisateur

- Les tables appartenant à d'autres utilisateurs ne se trouvent pas dans le schéma utilisateur.
- Le nom du propriétaire doit être placé devant le nom des tables.

ORACLE

6-6

Option DEFAULT

- Permet d'indiquer la valeur par défaut d'une colonne lors d'une insertion.

```
... hire_date DATE DEFAULT SYSDATE, ...
```

- Valeurs autorisées : valeurs littérales, expressions et fonctions SQL.
- Valeurs non autorisées : noms d'autres colonnes ou pseudo-colonnes.
- Le type de données par défaut doit correspondre à celui de la colonne.

ORACLE

6-7

Créer des tables

- Créez la table.

```
CREATE TABLE dept  
  (deptno NUMBER(2),  
   dname  VARCHAR2(14),  
   loc    VARCHAR2(13));
```

Table created.

- Vérifiez la création de la table.

```
DESCRIBE dept
```

Name	Null?	Type
DEPTNO		NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

ORACLE

6-8

Interroger le dictionnaire de données

- Consultez le nom des tables appartenant à l'utilisateur.

```
SELECT table_name  
FROM user_tables ;
```

- Affichez les différents types d'objet appartenant à l'utilisateur.

```
SELECT DISTINCT object_type  
FROM user_objects ;
```

- Affichez les tables, vues, synonymes et séquences appartenant à l'utilisateur.

```
SELECT *  
FROM user_catalog ;
```

ORACLE

6-10

Créer une table à l'aide d'une syntaxe de sous-interrogation

- Créez une table et insérez des lignes en associant l'instruction CREATE TABLE et l'option AS subquery.

```
CREATE TABLE table  
[(column, column...)]  
AS subquery;
```

- Faites correspondre le nombre de colonnes indiquées au nombre de colonnes de la sous-interrogation.
- Définissez des colonnes à l'aide de noms de colonne et de valeurs par défaut.

ORACLE

6-11

Créer une table à l'aide d'une sous-interrogation

```
CREATE TABLE dept80  
AS  
SELECT employee_id, last_name,  
       salary*12 ANNSAL,  
       hire_date  
FROM employees  
WHERE department_id = 80;
```

Table created.

```
DESCRIBE dept80
```

Name	Null?	Type
EMPLOYEE_ID		NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
ANNSAL		NUMBER
HIRE_DATE	NOT NULL	DATE

ORACLE

6-12

Instruction ALTER TABLE

L'instruction ALTER TABLE permet :

- d'ajouter une nouvelle colonne,
- de modifier une colonne existante,
- de définir une valeur par défaut pour une nouvelle colonne,
- de supprimer une colonne.

ORACLE

6-13

Instruction ALTER TABLE

L'instruction `ALTER TABLE` permet d'ajouter, de modifier ou de supprimer des colonnes.

```
ALTER TABLE table
ADD      (column datatype [DEFAULT expr]
         [, column datatype]...);
```

```
ALTER TABLE table
MODIFY   (column datatype [DEFAULT expr]
         [, column datatype]...);
```

```
ALTER TABLE table
DROP     (column);
```

ORACLE

6-14

Ajouter une colonne

Nouvelle colonne

DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE
149	Zlotkey	126000	29-JAN-00
174	Abel	132000	11-MAY-96
176	Taylor	103200	24-MAR-98

JOB_ID

"Ajoutez une colonne à la table DEPT80."

DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
149	Zlotkey	126000	29-JAN-00	
174	Abel	132000	11-MAY-96	
176	Taylor	103200	24-MAR-98	

ORACLE

6-15

Ajouter une colonne

- La clause `ADD` permet d'ajouter des colonnes.

```
ALTER TABLE dept80
ADD      (job_id VARCHAR2(9));
Table altered.
```

- La nouvelle colonne est placée à la fin de la table.

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
149	Zlotkey	126000	29-JAN-00	
174	Abel	132000	11-MAY-96	
176	Taylor	103200	24-MAR-98	

ORACLE

6-16

Modifier une colonne

- Vous pouvez modifier le type de données, la taille et la valeur par défaut d'une colonne.

```
ALTER TABLE dept80
MODIFY   (last_name VARCHAR2(30));
Table altered.
```

- La modification d'une valeur par défaut ne s'applique qu'aux insertions ultérieures dans la table.

ORACLE

6-17

Supprimer une colonne

La clause `DROP COLUMN` permet de supprimer d'une table les colonnes qui ne sont plus utiles.

```
ALTER TABLE dept80
DROP COLUMN job_id;
Table altered.
```

Option SET UNUSED

- L'option `SET UNUSED` permet d'identifier une ou plusieurs colonnes comme non utilisées.
- L'option `DROP UNUSED COLUMNS` permet de supprimer les colonnes identifiées comme non utilisées.

```
ALTER TABLE table
SET UNUSED (column);
OR
ALTER TABLE table
SET UNUSED COLUMN column;
```

```
ALTER TABLE table
DROP UNUSED COLUMNS;
```

Supprimer une table

- La structure et l'ensemble des données de la table sont supprimées.
- Toutes les transactions en cours sont validées.
- Tous les index sont supprimés.
- Vous ne pouvez pas annuler une instruction `DROP TABLE`.

```
DROP TABLE dept80;
Table dropped.
```

Renommer un objet

- L'instruction `RENAME` permet de renommer une table, une vue, une séquence ou un synonyme.

```
RENAME dept TO detail_dept;
Table renamed.
```

- Pour effectuer cette opération, vous devez être propriétaire de l'objet.

Vider une table

- L'instruction `TRUNCATE TABLE` :
 - supprime toutes les lignes d'une table,
 - libère l'espace de stockage utilisé par la table

```
TRUNCATE TABLE detail_dept;  
Table truncated.
```

- Vous ne pouvez pas annuler une instruction `TRUNCATE`.
- Vous pouvez également supprimer des lignes à l'aide de l'instruction `DELETE`.

Ajouter des commentaires à une table

- L'instruction `COMMENT` permet d'ajouter des commentaires à une table ou à une colonne.

```
COMMENT ON TABLE employees  
IS 'Employee Information';  
Comment created.
```

- Vous pouvez visualiser les commentaires grâce aux vues suivantes du dictionnaire de données :
 - `ALL_COL_COMMENTS`
 - `USER_COL_COMMENTS`
 - `ALL_TAB_COMMENTS`
 - `USER_TAB_COMMENTS`

Synthèse

Utiliser des instructions LDD pour créer, modifier, supprimer et renommer des tables.

Instruction	Description
<code>CREATE TABLE</code>	Crée une table
<code>ALTER TABLE</code>	Modifie la structure d'une table
<code>DROP TABLE</code>	Supprime les lignes et la structure d'une table
<code>RENAME</code>	Renomme une table, une vue, une séquence ou un synonyme
<code>TRUNCATE</code>	Supprime toutes les lignes d'une table et libère l'espace de stockage occupé par cette table
<code>COMMENT</code>	Ajoute des commentaires à une table ou à une vue

Présentation de l'exercice 6

Dans cet exercice, vous allez :

- créer des tables
- créer une table à l'aide de la syntaxe `CREATE TABLE AS`
- modifier des définitions de colonne
- vérifier l'existence des tables
- ajouter des commentaires aux tables
- supprimer des tables
- modifier des tables

7

Inclure des contraintes

ORACLE

Objectifs

- définir des contraintes
- créer et gérer des contraintes

6-29

ORACLE

Qu'est-ce qu'une contrainte ?

- Les contraintes appliquent des règles au niveau d'une table.
- Les contraintes empêchent la suppression d'une table lorsqu'il existe des dépendances.
- Les types de contrainte suivants sont utilisés :
 - NOT NULL
 - UNIQUE
 - PRIMARY KEY
 - FOREIGN KEY
 - CHECK

ORACLE

6-30

Règles applicables aux contraintes

- Vous pouvez affecter un nom aux contraintes ou laisser le serveur Oracle en générer un au format `SYS_Cn`.
- Vous pouvez créer une contrainte :
 - au moment de la création de la table,
 - une fois que la table est créée.
- Définissez une contrainte au niveau table ou colonne.
- Consultez le dictionnaire de données pour visualiser une contrainte.

ORACLE

6-31

Définir des contraintes

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

```
CREATE TABLE employees(
  employee_id  NUMBER(6),
  first_name   VARCHAR2(20),
  ...
  job_id       VARCHAR2(10) NOT NULL,
  CONSTRAINT emp_emp_id_pk
  PRIMARY KEY (EMPLOYEE_ID));
```

ORACLE

6-32

Définir des contraintes

- Contrainte au niveau colonne

```
column [CONSTRAINT constraint_name] constraint_type,
```

- Contrainte au niveau table

```
column,...
 [CONSTRAINT constraint_name] constraint_type
 (column, ...),
```

ORACLE

6-33

Contrainte NOT NULL

Interdit les valeurs NULL dans la colonne :

EMPLOYEE_ID	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID
100	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	90
101	Kochhar	NIKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	90
102	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	90
103	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	60
104	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	60
178	Grant	KGRANT	011.44.1644.429263	24-MAY-99	SA_REP	7000	
200	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400	10

...
20 rows selected.

↑
Contrainte NOT NULL
(aucune ligne de cette
colonne ne peut
contenir de valeur
NULL)

↑
Contrainte
NOT NULL

↑
Absence de contrainte
NOT NULL
(les lignes de cette
colonne peuvent
contenir une valeur NULL)

ORACLE

6-34

Contrainte NOT NULL

Cette contrainte est définie au niveau colonne :

```
CREATE TABLE employees(
  employee_id  NUMBER(6),
  last_name    VARCHAR2(25) NOT NULL,
  salary       NUMBER(8,2),
  commission_pct NUMBER(2,2),
  hire_date    DATE
  CONSTRAINT emp_hire_date_nn
  NOT NULL,
  ...
```

←
Nom
attribué par
le système

←
Nom
attribué par
l'utilisateur

ORACLE

6-35

Contrainte UNIQUE

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	EMAIL
100	King	SKING
101	Kochhar	NKOCHHAR
102	De Haan	LDEHAAN
103	Hunold	AHUNOLD
104	Ernst	BERNST

Contrainte UNIQUE

...



208	Smith	JSMITH
209	Smith	JSMITH

Autorisé
Non autorisé :
existe déjà

ORACLE

6-36

Contrainte UNIQUE

Cette contrainte est définie au niveau table ou colonne :

```
CREATE TABLE employees(  
  employee_id      NUMBER(6),  
  last_name        VARCHAR2(25) NOT NULL,  
  email            VARCHAR2(25),  
  salary           NUMBER(8,2),  
  commission_pct  NUMBER(2,2),  
  hire_date        DATE NOT NULL,  
  ...  
  CONSTRAINT emp_email_uk UNIQUE(email));
```

ORACLE

6-37

Contrainte PRIMARY KEY

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500

PRIMARY KEY

...



Non autorisé
(valeur NULL)

	Public Accounting		1400
50	Finance	124	1500

Non autorisé
(50 existe déjà)

ORACLE

6-38

Contrainte PRIMARY KEY

Cette contrainte est définie au niveau table ou colonne :

```
CREATE TABLE departments(  
  department_id    NUMBER(4),  
  department_name  VARCHAR2(30)  
  CONSTRAINT dept_name_nn NOT NULL,  
  manager_id       NUMBER(6),  
  location_id      NUMBER(4),  
  CONSTRAINT dept_id_pk PRIMARY KEY(department_id));
```

ORACLE

6-39

Contrainte FOREIGN KEY

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500

PRIMARY
KEY →

...

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
102	De Haan	90
103	Hunold	60
104	Ernst	60
107	Lorentz	60

← FOREIGN
KEY

...

↑ INSERT INTO

200	Ford	9
201	Ford	60

← Non autorisé
(9 n'existe pas)

← Autorisé

ORACLE

6-40

Contrainte FOREIGN KEY

Cette contrainte est définie au niveau table ou colonne :

```
CREATE TABLE employees(  
  employee_id      NUMBER(6),  
  last_name        VARCHAR2(25) NOT NULL,  
  email            VARCHAR2(25),  
  salary           NUMBER(8,2),  
  commission_pct  NUMBER(2,2),  
  hire_date        DATE NOT NULL,  
  ...  
  department_id   NUMBER(4),  
  CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)  
    REFERENCES departments(department_id),  
  CONSTRAINT emp_email_uk UNIQUE(email));
```

ORACLE

6-41

Mots-clés associés à la contrainte FOREIGN KEY

- **FOREIGN KEY** : définit une colonne de la table enfant au niveau table.
- **REFERENCES** : identifie la table et la colonne dans la table parent.
- **ON DELETE CASCADE** : supprime les lignes dépendantes de la table enfant lorsqu'une ligne de la table parent est supprimée.
- **ON DELETE SET NULL** : affecte la valeur NULL aux clés étrangères dépendantes.

ORACLE

6-42

Contrainte CHECK

- Définit une condition que chaque ligne doit satisfaire.
- Les expressions suivantes ne sont pas autorisées :
 - Références aux pseudo-colonnes CURRVAL, NEXTVAL, LEVEL et ROWNUM
 - Appels des fonctions SYSDATE, UID, USER et USERENV
 - Interrogations faisant référence à d'autres valeurs dans d'autres lignes

```
..., salary NUMBER(2)  
  CONSTRAINT emp_salary_min  
    CHECK (salary > 0),...
```

ORACLE

6-43

Ajouter une syntaxe de contrainte

Utilisez l'instruction `ALTER TABLE` pour :

- ajouter ou supprimer une contrainte sans modifier sa structure,
- activer ou désactiver des contraintes,
- ajouter une contrainte `NOT NULL` à l'aide de la clause `MODIFY`.

```
ALTER TABLE table
ADD [CONSTRAINT constraint] type (column);
```

ORACLE

6-44

Ajouter une contrainte

Ajoutez à la table `EMPLOYEES` une contrainte `FOREIGN KEY` précisant qu'un manager doit déjà exister dans cette table en tant qu'employé valide.

```
ALTER TABLE employees
ADD CONSTRAINT emp_manager_fk
FOREIGN KEY(manager_id)
REFERENCES employees(employee_id);
Table altered.
```

ORACLE

6-45

Supprimer une contrainte

- Supprimez de la table `EMPLOYEES` la contrainte relative au manager.

```
ALTER TABLE employees
DROP CONSTRAINT emp_manager_fk;
Table altered.
```

- Supprimez la contrainte `PRIMARY KEY` de la table `DEPARTMENTS`, ainsi que la contrainte `FOREIGN KEY` associée définie sur la colonne `EMPLOYEES.DEPARTMENT_ID`.

```
ALTER TABLE departments
DROP PRIMARY KEY CASCADE;
Table altered.
```

ORACLE

6-46

Désactiver des contraintes

- Pour désactiver une contrainte d'intégrité, exécutez la clause `DISABLE` de l'instruction `ALTER TABLE`.
- Pour désactiver les contraintes d'intégrité dépendantes, appliquez l'option `CASCADE`.

```
ALTER TABLE employees
DISABLE CONSTRAINT emp_emp_id_pk CASCADE;
Table altered.
```

ORACLE

6-47

Activer des contraintes

- Pour activer une contrainte d'intégrité désactivée dans la définition de la table, utilisez la clause `ENABLE`.

```
ALTER TABLE      employees
ENABLE CONSTRAINT emp_emp_id_pk;
Table altered.
```

- Si vous activez une contrainte de clé `UNIQUE` ou `PRIMARY KEY`, un index correspondant est automatiquement créé.

ORACLE

6-48

Contraintes en cascade

- La clause `CASCADE CONSTRAINTS` s'utilise avec la clause `DROP COLUMN`.
- La clause `CASCADE CONSTRAINTS` supprime toutes les contraintes d'intégrité référentielle qui font référence aux clés uniques et aux clés primaires définies sur les colonnes supprimées.
- La clause `CASCADE CONSTRAINTS` supprime également toutes les contraintes multicolonne des colonnes supprimées.

ORACLE

6-49

Contraintes en cascade

Exemple :

```
ALTER TABLE test1
DROP (pk) CASCADE CONSTRAINTS;
Table altered.
```

```
ALTER TABLE test1
DROP (pk, fk, coll) CASCADE CONSTRAINTS;
Table altered.
```

ORACLE

6-50

Afficher des contraintes

Pour afficher la définition et le nom de toutes les contraintes, interrogez la table `USER_CONSTRAINTS`.

```
SELECT  constraint_name, constraint_type,
        search_condition
FROM    user_constraints
WHERE   table_name = 'EMPLOYEES';
```

CONSTRAINT_NAME	C	SEARCH_CONDITION
EMP_LAST_NAME_NN	C	"LAST_NAME" IS NOT NULL
EMP_EMAIL_NN	C	"EMAIL" IS NOT NULL
EMP_HIRE_DATE_NN	C	"HIRE_DATE" IS NOT NULL
EMP_JOB_NN	C	"JOB_ID" IS NOT NULL
EMP_SALARY_MIN	C	salary > 0
EMP_EMAIL_UK	U	

...

ORACLE

6-51

Afficher les colonnes associées à des contraintes

Affichez les colonnes associées aux noms de contrainte à l'aide de la vue `USER_CONS_COLUMNS`.

```
SELECT constraint_name, column_name
FROM user_cons_columns
WHERE table_name = 'EMPLOYEES';
```

CONSTRAINT_NAME	COLUMN_NAME
EMP_DEPT_FK	DEPARTMENT_ID
EMP_EMAIL_NN	EMAIL
EMP_EMAIL_UK	EMAIL
EMP_EMP_ID_PK	EMPLOYEE_ID
EMP_HIRE_DATE_NN	HIRE_DATE
EMP_JOB_FK	JOB_ID
EMP_JOB_NN	JOB_ID
...	

ORACLE

6-52

Synthèse

- Types de contrainte :
 - NOT NULL
 - UNIQUE
 - PRIMARY KEY
 - FOREIGN KEY
 - CHECK
- Vous pouvez interroger la table `USER_CONSTRAINTS` pour voir le nom et la définition de toutes les contraintes.

ORACLE

6-53

Présentation de l'exercice 7

Dans cet exercice, vous allez :

- ajouter des contraintes à des tables existantes,
- ajouter des colonnes à une table,
- afficher des informations provenant des vues du dictionnaire de données.

ORACLE

6-54

8

Créer des vues

ORACLE

Objectifs

A la fin de ce chapitre, vous pourrez :

- décrire une vue
- créer, modifier et supprimer une vue
- extraire des données depuis une vue
- insérer, mettre à jour et supprimer des données depuis une vue
- créer et utiliser une vue en ligne
- réaliser une analyse de type n-premiers

ORACLE

8-57

Objets de base de données

Objet	Description
Table	Unité de stockage de base constituée de lignes et de colonnes
Vue	Représentation logique de sous-ensembles de données d'une ou de plusieurs tables
Séquence	Génère des valeurs de clé primaire
Index	Améliore les performances de certaines interrogations
Synonyme	Autre nom attribué à un objet

ORACLE

8-58

Définition d'une vue

Table EMPLOYEES :

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_FRES	2400
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	1700
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	1700
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	8000
107	Diana	Lorentz	DLORENTZ	590.423.4567	07-FEB-98	IT_PROG	4200
111	Irene	Baer	IBAER	590.123.4567	16-MAY-99	ST_MAN	5600
114	Matthew	King	MKAIS	650.121.2019	17-FEB-95	ST_CLERK	3600
141	Cliff	Johnson	CJAMES	650.121.2014	05-JAN-97	ST_CLERK	3100
142	Randall	Mateo	RMATEO	650.121.2014	15-MAR-95	ST_CLERK	2600
149	Zlotkey			10000	JAN-00	SA_MAN	10500
174	Abel			11000	MAY-96	SA_REP	11000
170	Taylor			0000	MAR-98	SA_REP	8600
170	Rudenberg	Stark	SRSTARK	011.985.1098.4290	24-MAY-99	SA_REP	7000
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-96	MK_MAN	13000
202	Pat	Fay	PFAY	603.123.6666	17-AUG-97	MK_REP	6000
205	Shelley	Higgins	SHIGGINS	515.123.0000	07-JUN-94	AC_MGR	12000
206	William	Gietz	WGIEZT	515.123.8181	07-JUN-94	AC_ACCOUNT	8300

20 rows selected.

ORACLE

8-59

Avantages des vues

- Limitent l'accès aux données
- Facilitent la création d'interrogations complexes
- Garantissent l'indépendance des données
- Présentent les mêmes données sous différentes vues

ORACLE

8-60

Vues simples et vues complexes

Caractéristiques	Vues simples	Vues complexes
Nombre de tables	Une	Une ou plusieurs
Fonctions	Non	Oui
Groupes de données	Non	Oui
Opérations LMD sur une vue	Oui	Pas toujours

ORACLE

8-61

Créer une vue

- Imbriguez une sous-interrogation dans l'instruction CREATE VIEW.

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW view
[(alias[, alias]...)]
AS subquery
[WITH CHECK OPTION [CONSTRAINT constraint]]
[WITH READ ONLY [CONSTRAINT constraint]];
```

- La sous-interrogation peut contenir une syntaxe SELECT complexe.

ORACLE

8-62

Créer une vue

- Créez la vue EMPVU80 qui doit contenir des informations sur les employés du service 80.

```
CREATE VIEW empvu80
AS SELECT employee_id, last_name, salary
FROM employees
WHERE department_id = 80;
```

View created.

- Décrivez la structure de la vue à l'aide de la commande iSQL*Plus DESCRIBE.

```
DESCRIBE empvu80
```

ORACLE

8-63

Créer une vue

- Créez une vue en utilisant des alias de colonne dans la sous-interrogation.

```
CREATE VIEW salvu50
AS SELECT employee_id ID_NUMBER, last_name NAME,
salary*12 ANN_SALARY
FROM employees
WHERE department_id = 50;
```

View created.

- Sélectionnez les colonnes de cette vue par leur nom d'alias.

ORACLE

8-64

Extraire des données d'une vue

```
SELECT *  
FROM salvu50;
```

ID_NUMBER	NAME	ANN_SALARY
124	Mourgos	69600
141	Rajs	42000
142	Davies	37200
143	Matos	31200
144	Vargas	30000

ORACLE

8-65

Interroger une vue

iSQL*Plus

```
SELECT *  
FROM empvu80;
```

EMPLOYEE_ID	LAST_NAME	SALARY
149	Zlotkey	10500
174	Abel	11000
176	Taylor	8600

Serveur Oracle

USER_VIEWS

```
EMPVU80  
SELECT employee_id,  
last_name, salary  
FROM employees  
WHERE department_id=80;
```

EMPLOYEES

ORACLE

8-66

Modifier une vue

- Modifiez la vue EMPVU80 à l'aide de la clause `CREATE OR REPLACE VIEW`. Ajoutez un alias pour chaque nom de colonne.

```
CREATE OR REPLACE VIEW empvu80  
(id_number, name, sal, department_id)  
AS SELECT employee_id, first_name || ' ' || last_name,  
salary, department_id  
FROM employees  
WHERE department_id = 80;
```

View created.

- Les alias de colonne de la clause `CREATE VIEW` s'affichent dans le même ordre que les colonnes de la sous-interrogation.

ORACLE

8-67

Créer une vue complexe

Créez une vue complexe contenant des fonctions de groupe pour afficher des valeurs provenant de deux tables.

```
CREATE VIEW dept_sum_vu  
(name, minsal, maxsal, avgsal)  
AS SELECT d.department_name, MIN(e.salary),  
MAX(e.salary),AVG(e.salary)  
FROM employees e, departments d  
WHERE e.department_id = d.department_id  
GROUP BY d.department_name;
```

View created.

ORACLE

8-68

Règles d'exécution des opérations LMD sur une vue

- Vous pouvez exécuter des opérations LMD sur des vues simples.
- Vous ne pouvez pas supprimer une ligne si la vue contient :
 - des fonctions de groupe,
 - une clause `GROUP BY`,
 - le mot-clé `DISTINCT`,
 - la pseudo-colonne `ROWNUM`.

ORACLE

8-69

Règles d'exécution des opérations LMD sur une vue

Vous ne pouvez pas modifier les données d'une vue si elle contient :

- des fonctions de groupe,
- une clause `GROUP BY`,
- le mot-clé `DISTINCT`,
- la pseudo-colonne `ROWNUM`,
- des colonnes définies par des expressions.

ORACLE

8-70

Règles d'exécution des opérations LMD sur une vue

Vous ne pouvez pas ajouter de données dans une vue si celle-ci comporte :

- des fonctions de groupe,
- une clause `GROUP BY`,
- le mot-clé `DISTINCT`,
- la pseudo-colonne `ROWNUM`,
- des colonnes définies par des expressions,
- des colonnes `NOT NULL` se trouvant dans les tables de base qui ne sont pas sélectionnées par la vue.

ORACLE

8-71

Utiliser la clause `WITH CHECK OPTION`

- Vous pouvez vous assurer que les opérations LMD effectuées sur la vue restent dans le domaine de la vue à l'aide de la clause `WITH CHECK OPTION`.

```
CREATE OR REPLACE VIEW empvu20
AS SELECT *
FROM employees
WHERE department_id = 20
WITH CHECK OPTION CONSTRAINT empvu20_ck ;
View created.
```

- Toute tentative de modification du numéro de service dans une ligne de la vue échouera, car elle transgresse la contrainte `WITH CHECK OPTION`.

ORACLE

8-72

Refuser des opérations LMD

- Aucune opération LMD ne pourra être exécutée si vous ajoutez l'option `WITH READ ONLY` dans la définition de votre vue.
- Le serveur Oracle envoie un message d'erreur lors de toute tentative d'exécution d'une instruction LMD sur une ligne de la vue.

Refuser des opérations LMD

```
CREATE OR REPLACE VIEW empvu10
  (employee_number, employee_name, job_title)
AS SELECT  employee_id, last_name, job_id
  FROM      employees
  WHERE     department_id = 10
  WITH READ ONLY;
View created.
```

Supprimer une vue

La suppression d'une vue n'entraîne pas la perte des données, car toute vue est basée sur des tables sous-jacentes de la base de données.

```
DROP VIEW view;
```

```
DROP VIEW empvu80;
View dropped.
```

Vues en ligne

- Une vue en ligne est une sous-interrogation intégrant un alias (ou nom de corrélation) que vous pouvez utiliser dans une instruction SQL.
- Une sous-interrogation nommée, contenue dans la clause `FROM` de l'interrogation principale, est un exemple de vue en ligne.
- Une vue en ligne n'est pas un objet de schéma.

Analyse de type n-premiers

- Les interrogations n-premiers vous permettent d'identifier les n valeurs les plus petites ou les plus grandes présentes d'une colonne.
Par exemple :
 - Quels sont les 10 produits les mieux vendus ?
 - Quels sont les 10 produits les moins vendus ?
- Les ensembles de valeurs les plus grandes et les plus petites correspondent à des interrogations n-premiers.

ORACLE

8-77

Réaliser une analyse de type n-premiers

La structure de haut niveau d'une analyse de type n-premiers se présente comme suit :

```
SELECT [column_list], ROWNUM
FROM (SELECT [column_list]
      FROM table
      ORDER BY Top-N_column)
WHERE ROWNUM <= N;
```

ORACLE

8-78

Exemple d'analyse de type n-premiers

Pour afficher le nom et le salaire des trois employés de la table `EMPLOYEES` qui touchent les salaires les plus élevés, procédez comme suit :

1 2 3

```
SELECT ROWNUM as RANK, last_name, salary
FROM (SELECT last_name,salary FROM employees
      ORDER BY salary DESC)
WHERE ROWNUM <= 3;
```

RANK	LAST_NAME	SALARY
1	King	24000
2	Kochhar	17000
3	De Haan	17000

1 2 3

ORACLE

8-79

Synthèse

Ce chapitre vous a permis d'apprendre qu'une vue est issue de données provenant d'autres tables ou vues. Elle présente les avantages suivants :

- Elle restreint l'accès à la base de données.
- Elle simplifie les interrogations.
- Elle garantit l'indépendance des données.
- Elle permet de visualiser les mêmes données sous différentes formes.
- Elle peut être supprimée sans perte des données sous-jacentes.
- Une vue en ligne est une sous-interrogation comportant un alias.
- Une analyse de type n-premiers peut être réalisée à l'aide de sous-interrogations et d'interrogations externes.

ORACLE

8-80

Présentation de l'exercice 8

Dans cet exercice, vous allez :

- créer une vue simple
- créer une vue complexe
- créer une vue avec une contrainte CHECK
- tenter de modifier des données dans une vue
- afficher des définitions de vue
- supprimer des vues

8-81

ORACLE

Autres objets de base de données



ORACLE

Objectifs

- créer, mettre à jour et utiliser des séquences
- créer et mettre à jour des index
- créer des synonymes privés et publics

8-85

ORACLE

Objets de base de données

Objet	Description
Table	Unité de stockage de base constituée de lignes et de colonnes
Vue	Représentation logique de sous-ensembles de données d'une ou de plusieurs tables
Séquence	Génère des valeurs de clé primaire
Index	Améliore les performances de certaines interrogations
Synonyme	Autre nom attribué à un objet

8-86

ORACLE

Définition d'une séquence

Une séquence :

- génère automatiquement des numéros uniques,
- est un objet partageable,
- s'utilise généralement pour créer une valeur de clé primaire,
- remplace le code applicatif,
- permet un accès plus rapide aux valeurs de séquence mises en mémoire cache

ORACLE

8-87

Syntaxe de l'instruction CREATE SEQUENCE

Définissez une séquence permettant de générer automatiquement des numéros séquentiels :

```
CREATE SEQUENCE sequence
  [INCREMENT BY n]
  [START WITH n]
  [{MAXVALUE n | NOMAXVALUE}]
  [{MINVALUE n | NOMINVALUE}]
  [{CYCLE | NOCYCLE}]
  [{CACHE n | NOCACHE}];
```

ORACLE

8-88

Créer une séquence

- Créez une séquence nommée DEPT_DEPTID_SEQ à utiliser pour la clé primaire de la table DEPARTMENTS.
- N'utilisez pas l'option CYCLE.

```
CREATE SEQUENCE dept_deptid_seq
  INCREMENT BY 10
  START WITH 120
  MAXVALUE 9999
  NOCACHE
  NOCYCLE;
```

Sequence created.

ORACLE

8-89

Vérifier des séquences

- Vérifiez les valeurs de votre séquence dans la table USER_SEQUENCES du dictionnaire de données.

```
SELECT  sequence_name, min_value, max_value,
        increment_by, last_number
FROM    user_sequences;
```

- Si l'option NOCACHE est indiquée, la colonne LAST_NUMBER affiche le prochain numéro de séquence disponible.

ORACLE

8-90

Pseudo-colonnes NEXTVAL et CURRVAL

- NEXTVAL renvoie la prochaine valeur de séquence disponible. Elle renvoie une valeur unique à chaque fois qu'elle est référencée, même pour différents utilisateurs.
- CURRVAL renvoie la valeur de séquence en cours.
- CURRVAL ne peut contenir une valeur que si NEXTVAL a été utilisée pour la séquence.

Utiliser une séquence

- Insérez un nouveau service nommé "Support" au niveau de l'ID 2500.

```
INSERT INTO departments(department_id,
                        department_name, location_id)
VALUES      (dept_deptid_seq.NEXTVAL,
            'Support', 2500);
```

1 row created.

- Affichez la valeur actuelle de la séquence DEPT_DEPTID_SEQ.

```
SELECT dept_deptid_seq.CURRVAL
FROM    dual;
```

Utiliser une séquence

- La mise en mémoire cache des valeurs de séquence permet un accès plus rapide à ces valeurs.
- Les valeurs de séquence peuvent présenter des trous :
 - en cas d'annulation,
 - en cas de panne du système,
 - si une séquence est utilisée dans une autre table.
- Si la séquence a été créée avec l'option NOCACHE, interrogez la table USER_SEQUENCES pour afficher la prochaine valeur disponible.

Modifier une séquence

Modifiez la valeur d'incrément, les valeurs minimale et maximale, ainsi que l'option CYCLE ou CACHE.

```
ALTER SEQUENCE dept_deptid_seq
            INCREMENT BY 20
            MAXVALUE 999999
            NOCACHE
            NOCYCLE;
```

Sequence altered.

Règles de modification d'une séquence

- Vous devez être le propriétaire de la séquence ou disposer du privilège `ALTER`.
- Seuls les numéros de séquence à venir seront modifiés.
- Vous devez supprimer, puis recréer la séquence pour modifier son premier numéro.
- La validité des valeurs est contrôlée.

Supprimer une séquence

- Supprimez une séquence du dictionnaire de données à l'aide de l'instruction `DROP SEQUENCE`.
- Une fois supprimée, la séquence ne peut plus être référencée.

```
DROP SEQUENCE dept_deptid_seq;  
Sequence dropped.
```

Définition d'un index

Un index :

- est un objet de schéma,
- permet au serveur Oracle d'optimiser le temps d'extraction de lignes par le biais d'un pointeur,
- peut réduire les E/S disque grâce à une méthode d'accès permettant d'identifier rapidement l'emplacement des données,
- est indépendant de la table qu'il indexe,
- est utilisé et mis à jour automatiquement par le serveur Oracle.

Mode de création des index

- **Création automatique** : un index unique est créé automatiquement lorsque vous définissez une contrainte `PRIMARY KEY` ou `UNIQUE` dans une définition de table.
- **Création manuelle** : vous pouvez créer des index non-unique sur des colonnes pour réduire le temps d'accès aux lignes.

Créer un index

- Créez un index sur une ou plusieurs colonnes.

```
CREATE INDEX index
ON table (column[, column]...);
```

- Réduisez le temps d'accès d'une interrogation à la colonne `LAST_NAME` de la table `EMPLOYEES`.

```
CREATE INDEX emp_last_name_idx
ON employees(last_name);
Index created.
```

ORACLE

8-100

Conditions exigeant la création d'index

Vous devez créer un index si :

- une colonne contient un grand nombre de valeurs,
- une colonne contient un grand nombre de valeurs NULL,
- des colonnes sont fréquemment utilisées conjointement dans une clause `WHERE` ou une condition de jointure,
- la table est volumineuse et la plupart des interrogations doivent extraire moins de 2 à 4 % des lignes.

ORACLE

8-101

Conditions n'exigeant pas la création d'index

Il n'est généralement pas nécessaire de créer un index si :

- la table est de taille réduite,
- les colonnes ne sont pas fréquemment utilisées comme condition dans l'interrogation,
- la plupart des interrogations doivent extraire plus de 2 à 4 % des lignes de la table,
- la table est fréquemment mise à jour,
- les colonnes indexées sont référencées comme faisant partie d'une expression.

ORACLE

8-102

Vérifier des index

- La vue `USER_INDEXES` du dictionnaire de données contient le nom de l'index et son unicité.
- La vue `USER_IND_COLUMNS` contient les noms d'index, de table et de colonne.

```
SELECT ic.index_name, ic.column_name,
       ic.column_position col_pos, ix.uniqueness
FROM user_indexes ix, user_ind_columns ic
WHERE ic.index_name = ix.index_name
AND ic.table_name = 'EMPLOYEES';
```

ORACLE

8-103

Index basés sur une fonction

- Un index basé sur une fonction est un index basé sur des expressions.
- L'expression d'un index est constituée à partir de colonnes de table, de constantes, de fonctions SQL et de fonctions définies par l'utilisateur.

```
CREATE INDEX upper_dept_name_idx  
ON departments(UPPER(department_name));
```

Index created.

```
SELECT *  
FROM departments  
WHERE UPPER(department_name) = 'SALES';
```

ORACLE

8-104

Supprimer un index

- Supprimez un index du dictionnaire de données à l'aide de la commande `DROP INDEX`.

```
DROP INDEX index;
```

- Supprimez l'index `UPPER_LAST_NAME_IDX` du dictionnaire de données.

```
DROP INDEX upper_last_name_idx;  
Index dropped.
```

- Pour supprimer un index, vous devez en être le propriétaire ou disposer du privilège `DROP ANY INDEX`.

ORACLE

8-106

Synonymes

Les synonymes (autres noms d'objet) simplifient l'accès aux objets. Ils permettent :

- de référencer plus facilement une table appartenant à un autre utilisateur,
- d'abrégier les noms d'objet.

```
CREATE [PUBLIC] SYNONYM synonym  
FOR object;
```

ORACLE

8-107

Créer et supprimer des synonymes

- Créez un nom abrégé pour la vue `DEPT_SUM_VU`.

```
CREATE SYNONYM d_sum  
FOR dept_sum_vu;  
Synonym Created.
```

- Supprimez un synonyme.

```
DROP SYNONYM d_sum;  
Synonym dropped.
```

ORACLE

8-108

Synthèse

- générer automatiquement des numéros de séquence à l'aide d'un générateur de séquences
- visualiser des informations relatives aux séquences dans la table `USER_SEQUENCES` du dictionnaire de données
- créer des index pour optimiser le temps d'extraction des interrogations
- visualiser des informations relatives aux index dans la table `USER_INDEXES` du dictionnaire de données
- attribuer d'autres noms aux objets à l'aide de synonymes

Présentation de l'exercice 9

Dans cet exercice, vous allez :

- créer des séquences
- utiliser des séquences
- créer des index non-uniqes
- afficher des informations du dictionnaire de données relatives aux séquences et aux index
- supprimer des index