

# 10

## Inclure des contraintes

# Objectifs

**A la fin de ce chapitre, vous pourrez :**

- **définir des contraintes**
- **créer et gérer des contraintes**

# 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

# 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.

# 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));
```

# Définir des contraintes

- **Contrainte au niveau colonne**

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

- **Contrainte au niveau table**

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

# 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	NKOCHHAR	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®**

# 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



# Contrainte UNIQUE

## EMPLOYEES

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

...

INSERT INTO

208	Smith	JSMITH
209	Smith	JSMITH

← Autorisé

← Non autorisé :  
existe déjà

# 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));
```

# Contrainte PRIMARY KEY

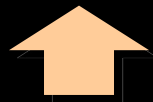
## DEPARTMENTS



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

...

Non autorisé  
(valeur NULL)



**INSERT INTO**

	Public Accounting		1400
50	Finance	124	1500

Non autorisé  
(50 existe déjà)



# 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));
```

# 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**



# 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));
```

# 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.

# 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), ...
```



# 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);
```

# 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.**

# 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.
```

# 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.
```

# 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.

# 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.

# Contraintes en cascade

Exemple :

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

```
ALTER TABLE test1  
DROP (pk, fk, col1) CASCADE CONSTRAINTS;  
Table altered.
```

# 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	

...



# 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

...

# Synthèse

**Ce chapitre vous a permis d'apprendre à créer des contraintes.**

- **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.**

# Présentation de l'exercice 10

**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.**