

Bases de Données Avancées

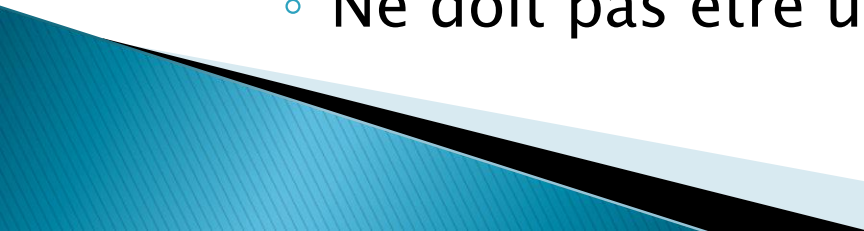
M. ZOUHRI

Objets d'une Base de données

Objet	Description
Table	Unité de stockage élémentaire, composée de lignes et de colonnes
Vue	Représente de manière logique des sous-groupes de données issues d'une ou plusieurs tables
Séquence	Génère des valeurs de clés primaires
Index	Améliore les performances de certaines requêtes
Synonyme	Permet de donner un autre nom à un objet

Conventions de Dénomination

Un nom :

- Doit commencer par une lettre
 - Peut comporter de 1 à 30 caractères
 - Ne peut contenir que les caractères A à Z, a à z, 0 à 9, _, \$, et #
 - Ne doit pas porter le nom d'un autre objet appartenant au même utilisateur
 - Ne doit pas être un mot réservé Oracle
- 

L'ordre **CREATE TABLE**

- Vous devez posséder :
 - Un privilège CREATE TABLE
 - Un espace de stockage

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

- Spécifiez :
 - Un nom de table
 - Le nom, le type de données et la taille des colonnes.

Références aux Tables d'un autre utilisateur

- Les tables appartenant à d'autres utilisateurs ne sont pas dans le schéma utilisateur.
- Le nom du propriétaire doit précéder le nom de la table.

L'option **DEFAULT**

- Spécifie la valeur par défaut d'une colonne.

```
... hiredate DATE DEFAULT SYSDATE, ...
```

- Valeurs autorisées : littéraux, 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.

Création de Tables

- Créer la table.

```
SQL> CREATE TABLE dept
      2          (deptno  NUMBER(2) ,
      3            dname   VARCHAR2(14) ,
      4            loc     VARCHAR2(13)) ;
Table created.
```

- Vérifier la création de la table.

```
SQL> DESCRIBE dept
```

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

Interrogation du dictionnaire de données

- Décrire les tables appartenant à l'utilisateur.

```
SQL> SELECT *  
2 FROM user_tables;
```

- Afficher les différents types d'objets appartenant à l'utilisateur.

```
SQL> SELECT DISTINCT object_type  
2 FROM user_objects;
```

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

```
SQL> SELECT *  
2 FROM user_catalog;
```


Types de Données

Types de données	Description
VARCHAR2(taille)	Données caractères de longueur variable
CHAR(<i>taille</i>)	Données caractères de longueur fixe
NUMBER(<i>p,s</i>)	Numérique de longueur variable
DATE	Valeurs de date et d'heure
LONG	Données caractères de longueur variable, jusqu'à 2 giga-octets
CLOB	Données caractères mono-octet, jusqu'à 4 giga-octets
RAW et LONG RAW	Binaire
BLOB	Binaire, jusqu'à 4 giga-octets
BFILE	Binaire, stocké dans un fichier externe, jusqu'à 4 giga-octets

Création d'une table au moyen d'une sous-interrogation

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

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

- Le nombre de colonnes spécifiées doit correspondre au nombre de colonnes de la sous-interrogation.
- Définissez des colonnes avec des noms de colonne et des valeurs par défaut.

Création d'une table au moyen d'une sous-interrogation

```
SQL> CREATE TABLE      dept30
  2  AS
  3      SELECT      empno,  ename,  sal*12 ANNSAL,  hiredate
  4      FROM        emp
  5      WHERE       deptno = 30;
Table created.
```

```
SQL> DESCRIBE dept30
```

Name	NULL?	Type
-----	-----	-----
EMPNO		NUMBER (4)
ENAME		VARCHAR2 (10)
ANNSAL		NUMBER
HIREDATE		DATE

L'ordre **ALTER TABLE**

- ▶ Utilisez l'ordre **ALTER TABLE** pour :
 - Ajouter une colonne
 - Modifier une colonne existante
 - Définir une valeur par défaut pour une nouvelle colonne

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

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

Ajout de Colonnes

- Utilisez la clause ADD pour ajouter des colonnes.

```
SQL> ALTER TABLE dept30
      2 ADD          (job VARCHAR2(9)) ;
Table altered.
```

- La nouvelle colonne est placée à la fin.

EMPNO	ENAME	ANNSAL	HIREDATE	JOB
7698	BLAKE	34200	01-MAY-81	
7654	MARTIN	15000	28-SEP-81	
7499	ALLEN	19200	20-FEB-81	
7844	TURNER	18000	08-SEP-81	
...				

6 rows selected.

Modification de Colonnes

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

```
ALTER TABLE dept30  
MODIFY (ename VARCHAR2(15));  
Table altered.
```

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

Modification de colonnes

Vous pouvez :

- Augmenter la largeur ou la précision d'une colonne numérique.
- Réduire la largeur d'une colonne si elle ne contient pas de valeurs NULL ou si la table est vide.
- Modifier le type de données d'une colonne si elle contient des valeurs NULL ou si la table est vide.
- Convertir de CHAR à VARCHAR ou inversement si la colonne contient des valeurs NULL, si vous ne réduisez pas sa taille ou si la table est vide.

Suppression de Tables

- La structure et toutes les données de la table sont supprimées.
- Tous les index sont supprimés.
- La transaction en instance est validée.
- Une suppression de table ne peut être annulée.

```
SQL> DROP TABLE dept30;  
Table dropped.
```


Modification du Nom d'un Objet

- Pour modifier le nom d'une table, d'une vue, d'une séquence ou d'un synonyme, utilisez l'ordre RENAME.

```
SQL> RENAME dept TO department;  
Table renamed.
```

- Vous devez être propriétaire de l'objet.

Vider une Table

- L'ordre TRUNCATE TABLE :
 - Supprime toutes les lignes d'une table
 - Libère l'espace de stockage utilisé par la table

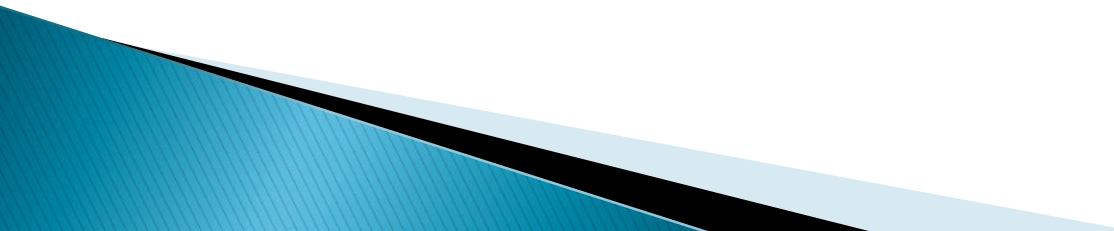
```
SQL> TRUNCATE TABLE department;  
Table truncated.
```

- Vous ne pouvez pas annuler un ordre TRUNCATE
- Vous pouvez aussi utiliser l'ordre DELETE pour supprimer des lignes

Les Contraintes

- Les contraintes contrôlent des règles de gestion au niveau d'une table.
- Les contraintes empêchent la suppression d'une table lorsqu'il existe des dépendances.
- Types de contraintes valides dans Oracle :
 - **NOT NULL**
 - **UNIQUE**
 - **PRIMARY KEY**
 - **FOREIGN KEY**
 - **CHECK**

Conventions Applicables aux Contraintes

- Si vous ne nommez pas une contrainte, Oracle8 Server créera un nom au format `SYS_Cn`.
 - Vous pouvez créer une contrainte :
 - En même temps que la création de la table
 - Une fois que la table est créée
 - Définissez une contrainte peut être définie au niveau table ou colonne.
 - Consulter le dictionnaire de données pour retrouver une contrainte.
- 

Les Contraintes

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

```
CREATE TABLE emp (  
    empno    NUMBER(4) ,  
    ename    VARCHAR2(10) ,  
    ...  
    deptno   NUMBER(2) NOT NULL,  
    CONSTRAINT emp_empno_pk  
                PRIMARY KEY (EMPNO)) ;
```

Les Contraintes

- Contrainte au niveau colonne

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

- Contrainte au niveau table

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

La Contrainte **NOT NULL**

- ▶ Se définit au niveau colonne

```
SQL> CREATE TABLE emp (  
2      empno      NUMBER(4) ,  
3      ename      VARCHAR2(10) NOT NULL,  
4      job        VARCHAR2(9) ,  
5      mgr        NUMBER(4) ,  
6      hiredate   DATE ,  
7      sal        NUMBER(7,2) ,  
8      comm       NUMBER(7,2) ,  
9      deptno     NUMBER(2) NOT NULL) ;
```

La Contrainte de Clé UNIQUE

DEPT

contrainte de clé UNIQUE

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



A insérer

50	SALES	DETROIT
60		BOSTON

Interdit
(DNAME—SALES
existe déjà)

Autorisé

La Contrainte de Clé UNIQUE

- ▶ Se définit au niveau table ou colonne

```
SQL> CREATE TABLE dept(  
2      deptno    NUMBER(2) ,  
3      dname     VARCHAR2(14) ,  
4      loc       VARCHAR2(13) ,  
5      CONSTRAINT dept_dname_uk UNIQUE(dname) ) ;
```

La Contrainte PRIMARY KEY

DEPT ← clé PRIMAIRE

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

↑ A insérer

20	MARKETING	DALLAS
	FINANCE	NEW YORK

interdit (DEPTNO-20
existe déjà)

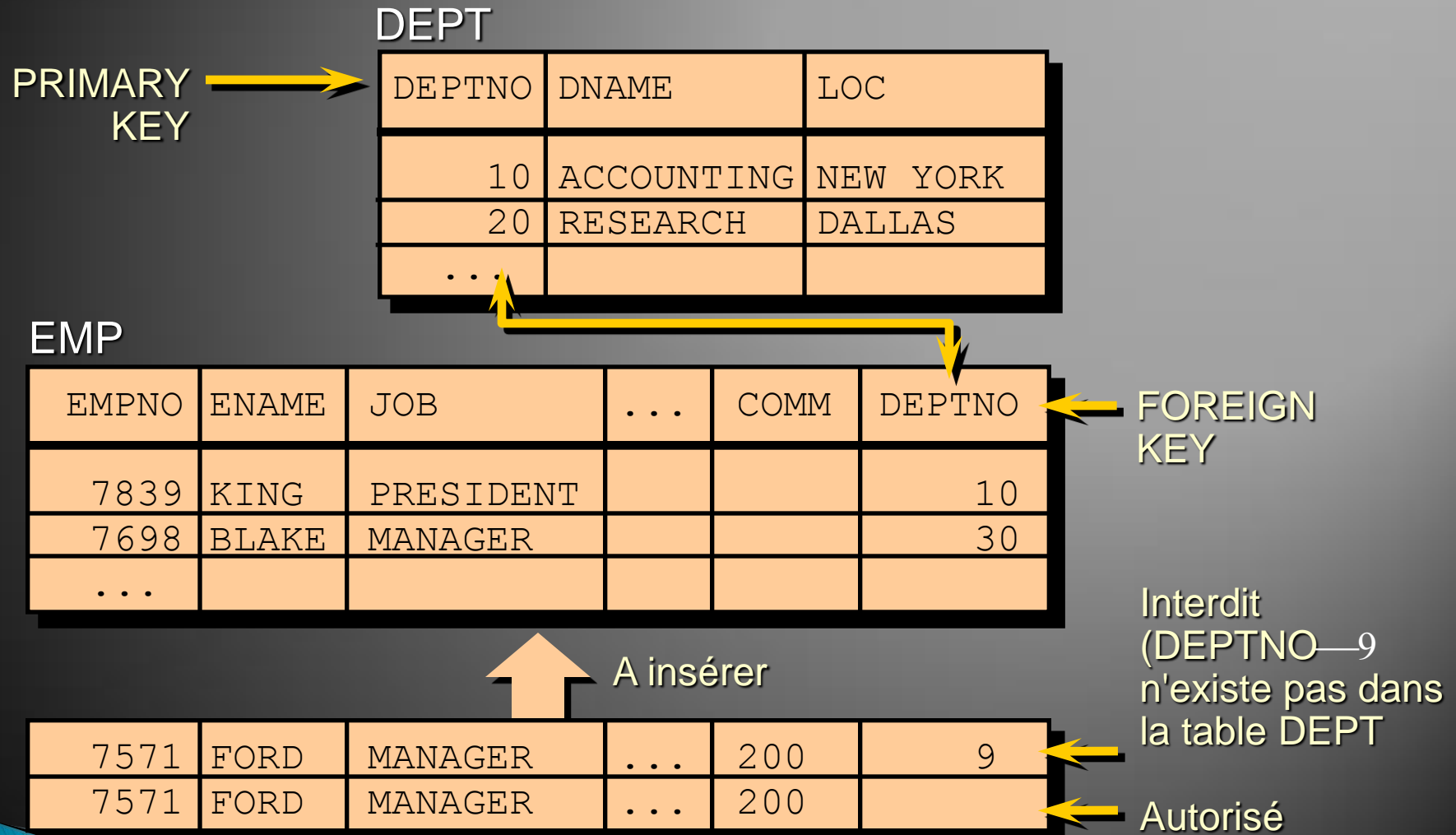
interdit
(DEPTNO est NULL)

La Contrainte **PRIMARY KEY**

- ▶ Se définit au niveau table ou colonne

```
SQL> CREATE TABLE dept(  
2     deptno      NUMBER(2) ,  
3     dname       VARCHAR2(14) ,  
4     loc         VARCHAR2(13) ,  
5     CONSTRAINT dept_dname_uk UNIQUE (dname) ,  
6     CONSTRAINT dept_deptno_pk PRIMARY KEY(deptno)) ;
```

La Contrainte FOREIGN KEY



La Contrainte **FOREIGN KEY**

- ▶ Se définit au niveau table ou colonne

```
SQL> CREATE TABLE emp (  
  2      empno      NUMBER(4) ,  
  3      ename      VARCHAR2(10) NOT NULL ,  
  4      job        VARCHAR2(9) ,  
  5      mgr        NUMBER(4) ,  
  6      hiredate   DATE ,  
  7      sal        NUMBER(7,2) ,  
  8      comm       NUMBER(7,2) ,  
  9      deptno     NUMBER(2) NOT NULL ,  
 10      CONSTRAINT emp_deptno_fk FOREIGN KEY (deptno)  
 11      REFERENCES dept (deptno)) ;
```

Mots-clés Associés à la Contrainte FOREIGN KEY

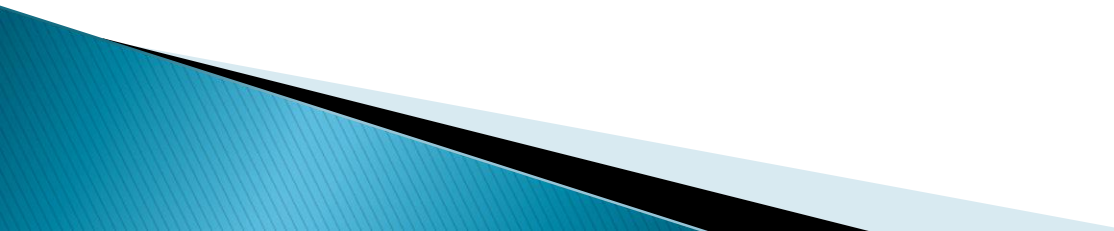
- **FOREIGN KEY**

- Définit la colonne dans la table détail dans une contrainte de niveau table

- **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 dépendantes dans la table détail
- 

La Contrainte **CHECK**

- Définit une condition que chaque ligne doit obligatoirement satisfaire
- Expressions interdites :
 - Références aux pseudo-colonnes CURRVAL, NEXTVAL, LEVEL et ROWNUM
 - Appels aux fonctions SYSDATE, UID, USER et USERENV
 - Requêtes faisant référence à d'autres valeurs dans d'autres lignes

```
..., deptno  NUMBER(2),  
        CONSTRAINT emp_deptno_ck  
        CHECK (DEPTNO BETWEEN 10 AND 99), ...
```

Ajout d'une Contrainte

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

- Vous pouvez ajouter ou supprimer une contrainte, mais pas la modifier
- Vous pouvez activer ou désactiver des contraintes
- Pour ajouter une contrainte NOT NULL, utilisez la clause MODIFY

Ajout d'une Contrainte

- ▶ Ajouter une contrainte FOREIGN KEY à la table EMP précisant qu'un manager doit déjà exister dans la table EMP en tant qu'employé valide.

```
SQL> ALTER TABLE      emp
      2  ADD CONSTRAINT emp_mgr_fk
      3          FOREIGN KEY (mgr) REFERENCES emp (empno) ;
Table altered.
```

Suppression d'une Contrainte

- Supprimer de la table EMP la contrainte concernant le manager.

```
SQL> ALTER TABLE          emp
      2  DROP CONSTRAINT emp_mgr_fk;
Table altered.
```

- Supprimer la contrainte PRIMARY KEY de la table DEPT, ainsi que la contrainte FOREIGN KEY associée définie sur la colonne EMP.DEPTNO.

```
SQL> ALTER TABLE          dept
      2  DROP PRIMARY KEY CASCADE;
Table altered.
```

Désactivation de Contraintes

- Pour désactiver une contrainte d'intégrité, utiliser la clause DISABLE de l'ordre ALTER TABLE.
- Pour désactiver les contraintes d'intégrité dépendantes, ajouter l'option CASCADE.

```
SQL> ALTER TABLE emp  
2 DISABLE CONSTRAINT emp_empno_pk CASCADE;  
Table altered.
```

Activation de Contraintes

- Pour activer une contrainte d'intégrité actuellement désactivée dans la définition de la table, utiliser la clause **ENABLE**.

```
SQL> ALTER TABLE      emp
      2  ENABLE CONSTRAINT emp_empno_pk ;
Table altered.
```

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

Vérification des Contraintes

- Pour afficher les définitions et noms de toutes les contraintes, interrogez la table USER_CONSTRAINTS.

```
SQL> SELECT constraint_name, constraint_type,  
2         search_condition  
3 FROM   user_constraints  
4 WHERE  table_name = 'EMP';
```

CONSTRAINT_NAME	C SEARCH_CONDITION
-----	- - - - -
SYS_C00674	C EMPNO IS NOT NULL
SYS_C00675	C DEPTNO IS NOT NULL
EMP_EMPNO_PK	P
...	

Affichage des Colonnes Associées aux Contraintes

- ▶ Affichez les colonnes associées aux noms de contraintes au moyen de la vue USER_CONS_COLUMNS view

```
SQL> SELECT constraint_name, column_name  
2 FROM user_cons_columns  
3 WHERE table_name = 'EMP';
```

CONSTRAINT_NAME	COLUMN_NAME
-----	-----
EMP_DEPTNO_FK	DEPTNO
EMP_EMPNO_PK	EMPNO
EMP_MGR_FK	MGR
SYS_C00674	EMPNO
SYS_C00675	DEPTNO