

## **TABLES SYSTEMES**

**Stockent le dictionnaire de données de la BD**

**USER\_TABLES (TABLE\_NAME, TABLESPACE\_NAME, ...)**

**USER\_TAB\_COLUMNS(TABLE\_NAME, COLUMN\_NAME, DATA\_TYPE, DATA\_LENGTH)**

**USER\_CONS\_COLUMNS (TABLE\_NAME, COLUMN\_NAME, CONSTRAINT\_NAME,...)**

**USER\_CONSTRAINTS (TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE, SEARCH\_CONDITION, ...)**

**USER\_OBJECTS (OBJECT\_NAME, OBJECT\_TYPE...)**

**Les valeurs de l'attribut DATA\_TYPE et DATA\_LENGTH correspondent aux types de données Oracle (VARCHAR2, NUMBER, DATE, ...) et à la taille du type.**

**La valeur de l'attribut CONSTRAINT\_TYPE peut être 'P' (Primary), 'R' (References) ou 'C'(Check).**

**La valeur de l'attribut OBJECT\_TYPE peut être TABLE, VIEW, FUNCTION, PROCEDURE, TRIGGER, ...**

### **Exemples :**

**Listez les tables (nom) de la base**

```
SELECT OBJECT_NAME  
FROM USER_OBJECTS  
WHERE OBJECT_TYPE = 'TABLE' ;
```

**Listez les attributs (nom, type et taille) de la table ETUDIANT**

```
SELECT COLUMN_NAME, DATA_TYPE, DATA_LENGTH  
FROM USER_TAB_COLUMNS  
WHERE TABLE_NAME = 'ETUDIANT' ;
```

## VUE (VIEW)

Table virtuelle c.a.d. les données associées ne sont pas stockées dans la BD mais accessibles par requête.

### VUE de CONSULTATION

**CREATE VIEW nom\_View[(col1, col2, ..)] AS**

**Sous-Requête ;**

**Contrainte : une seule table dans le FROM de la requête.**

### Exemples :

#### **Vu matières suivies par les étudiants Marseillais**

**CREATE VIEW MATIERE(MAT, LIBELLE) AS**

**SELECT EN.CODE, LIBELLE**

**FROM ENSEIGNT EN, MODULE M, ETUDIANT E**

**WHERE EN.CODE = M.CODE AND EN.NUM\_ET = E. NUM\_ET**

**AND VILLE\_ET = 'MARSEILLE' ;**

#### **Consultations**

**SELECT \* FROM MATIERE ;**

**SELECT COUNT(DISTINCT CODE)**

**FROM MATIERE ;**

## VUE de MISE A JOUR

CREATE VIEW nom\_View[(col1, col2, ..)] AS

Sous Requête

WITH CHECK Option ;

Contrainte : une seule table dans le FROM de la Sous Requête.

## Exemples :

### Vu des étudiants Marseillais

CREATE VIEW ETUDMARS AS

SELECT \* FROM ETUDIANT WHERE VILLE = 'MARSEILLE'

WITH CHECK OPTION ;

Ajout valide dans ETUDIANT par l'intermédiaire de la table ETUDMARS :

INSERT INTO ETUDMARS

INTO (2205, 'DUPONT', 'SERGE', '13001', 'MARSEILLE', 1, 2) ;

Ajout non valide Ajout valide dans ETUDIANT par l'intermédiaire de la table ETUDMARS :

INSERT INTO ETUDMARS

INTO (2205, 'DUPONT', 'SERGE', '13001', 'AIX-EN-PROVENCE', 1, 2) ;

## PRIVILEGES D'ACCES

### Octroie et retrait des accès à des tables d'autres utilisateurs

```
GRANTE {SELECT | INSERT | UPDATE | DELETE | ALTER | ALL}  
ON Nom_Table  
TO { User | Public}  
[WITH GRANT OPTION] ;
```

#### Exemple :

```
GRANT SELECT ON ETUDIANT TO ET101
```

#### Consultation :

```
SELECT * FROM ET101.ETUDIANT ;
```

```
REVOKE {SELECT | INSERT | UPDATE | DELETE | ALTER | ALL}  
ON Nom_Table  
FROM { User | Public} ;
```

#### Exemple :

```
REVOKE SELECT ON ETUDIANT FROM ET101 ;
```

## **CLASSEMENT D'OBJETS**

```
SELECT attribut1, ...,  
RANK() OVER (ORDER BY attributi, ASC|DESC) MyRank  
FROM,.....
```

```
SELECT IDV, DATEDEP, TARIF,  
RANK() (ORDER BY TARIF) RANG  
FROM PLANNING;
```

```
SELECT VILLE, NB,  
RANK() OVER (ORDER BY NB) RANG  
FROM (SELECT VILLE, COUNT(*) NB  
      FROM CLIENT  
      GROUP BY VILLE);
```

```
CREATE VIEW NB_VILLES AS  
SELECT VILLE, COUNT(*) NB  
FROM CLIENT  
GROUP BY VILLE);
```

```
SELECT VILLE, NB,  
RANK() OVER (ORDER BY NB) RANG  
FROM NB_VILLES;
```