

## **Les vues**

Une vue est une fenêtre à travers laquelle l'utilisateur voit la BD.

Les vues correspondent au niveau externe de la BD

Une vue est définie à partir d'une requête.

une vue joue un rôle fondamental dans la confidentialité.

- Peut être considérée comme une table virtuelle
- pas d'existence propre : seule sa structure est stockée dans le dictionnaire de données
- évaluée à chaque consultation : se recharge à chaque fois qu'elle est interrogée

### **Quelques avantages des vues :**

- d'éviter de taper une requête très longue : la vue sert à donner un nom à la requête pour l'utiliser souvent, (les requêtes simples)
- de masquer certaines données à certains utilisateurs. En [SQL](#), les protections d'une vue ne sont pas forcément les mêmes que celles des tables sous-jacentes.( restreignent l'accès aux données)

### **Les étapes de création des vues objets**

- Détermination de la requête de la vue (contenu de la vue)
- Création du type objet (Objet de base de la vue) ayant de même structure que la requête.
- Création de la vue objet avec la commande create view.

### **Création d'une vue**

-Pour pouvoir créer une vue dans votre schéma vous devez posséder le privilège CREATE VIEW.

## Commande de création de vue (simplifiée)

```
create [ or replace] view nom_vue  
  
[(nom_coll,...)]  
      AS SELECT ...  
WITH with read only | with check option;
```

La spécification des noms de colonnes de la vue est facultative.

Par défaut, les noms des colonnes de la vue sont les mêmes que les noms des colonnes résultat du SELECT.

### Exemple :

Un exemple, où les employés d'une entreprise sont dans une table, les départements de l'entreprise dans une autre et où on doit faire une jointure pour afficher le nom du département à côté de celui de l'employé :

```
CREATE TABLE Employes (id SERIAL, nom TEXT, departement INTEGER);  
CREATE TABLE Departements (id SERIAL, nom TEXT);  
SELECT e.nom AS Employe, d.nom AS Departement  
      FROM Employes e,Departements d WHERE e.departement = d.id;
```

on crée une vue :

```
CREATE VIEW ToutLeMonde AS  
      SELECT e.nom AS Employe, d.nom AS Departement  
      FROM Employes e,Departements d  
      WHERE e.departement = d.id;
```

On pourra alors écrire la requête SELECT ci-dessus bien plus simplement, la jointure ne sera plus visible :

```
SELECT * FROM ToutLeMonde ;
```

Les **vues** s'utilisent pratiquement comme des tables, elles peuvent être dans une clause FROM d'un SELECT, dans UPDATE, etc, avec quelques restrictions.

**Remarque :** Description de la vue : idem qu'une table à utiliser describe ou desc

```
SQL> desc ToutLeMonde;
```

**Remarque :** Pour insérer un nouveau enregistrement dans une vue simple (monotable) il faut que la définition de la vue comporte le champ clé primaire de la table sources.

**Exemple :** impossible d'insérer un nouveau enregistrement dans la vue empv1 car la clé primaire de la table source ne serait pas renseignée.

```
SQL> create or replace view empv1 as select ename, sal from emp ;
Vue créée.

SQL> insert into empv1 values('tittt12',1000);
insert into empv1 values('tittt12',1000)
*
ERREUR ó la ligne 1 :
ORA-01400: impossible d'insérer NULL dans ("SCOTT"."EMP"."EMPNO")
```

**with read only** : interdit toute tentative de modification de la vue (insert, update, delete) :  
Vues en lecture seule

**with CHECK OPTION** Permet de spécifier que les requêtes INSERT et UPDATE ne peuvent pas créer des enregistrements ne pouvant être sélectionnés par la vue.

**Exemple : (à tester)**

```
SQL> create or replace view empv1 as select empno, ename, sal from emp where sal
>4000 with check option ;
Vue créée.
```

```
SQL> select * from empv1;

  EMPNO ENAME      SAL
-----
  7839  KING          5000
```

```
SQL> insert into empv1 values(8000,'Toto', 1000);
insert into empv1 values(8000,'Toto', 1000)
*
ERREUR ó la ligne 1 :
ORA-01402: vue WITH CHECK OPTION - violation de clause WHERE
```

**Exemple :**Création d'une vue constituant une restriction de la table **emp** aux employés du département 10.

```
CREATE VIEW emp10 AS  
SELECT *  
FROM emp  
WHERE n_dept = 10 ;
```

Ainsi donc, si la vue **emp10** a été créée avec **CHECK OPTION** on ne pourra à travers cette vue ni modifier, ni insérer des employés ne faisant pas partie du département 10.

Il est possible d'effectuer des INSERT et des UPDATE à travers des vues, sous deux conditions :

- le SELECT définissant la vue ne doit pas comporter de jointure,
- les colonnes résultat du SELECT doivent être des colonnes réelles et non pas des expressions.

Condition de mise à jour d'une vue :

- La requête de définition doit respecter les critères suivants :

- \* pas de directive DISTINCT,
- \* pas de fonction (AVG, COUNT, MAX, MIN, STDDEV, SUM, ou VARIANCE),
- \* pas d'expression ou de pseudo-colonne (ROWNUM, ROWID) dans le SELECT.
- \* pas de GROUP BY, ORDER BY, HAVING

### **Supprimer une vue**

Une vue peut être détruite par la commande :

- DROP VIEW nom\_vue;

### **Renommer une vue**

- On peut renommer une vue par la commande :
  - RENAME ancien\_nom TO nouveau\_nom;