

Contexte d'information	<i>cours SQL</i>
Titre du document	Les vues
Nom de fichier	sql08.pdf

1. Principes

Les vues permettent d'assurer l'objectif d'**indépendance logique**. Grâce à elles, chaque utilisateur pourra avoir sa vision propre des données. Une vue est une table virtuelle, c'est-à-dire dont les données ne sont pas stockées dans une table de la base de données, et dans laquelle il est possible de rassembler des informations provenant de plusieurs tables. On parle de "vue" car il s'agit simplement d'une représentation des données dans le but d'une exploitation visuelle. Les données présentes dans une vue sont définies grâce à une clause `SELECT`

Les utilisateurs pourront consulter la base, ou modifier la base (avec certaines restrictions) à travers la vue, c'est-à-dire manipuler la table résultat du `select` comme si c'était une table réelle.

2. Création

La commande `CREATE VIEW` permet de créer une vue en spécifiant le `select` constituant la définition de la vue :

```
CREATE VIEW nom_vue [(nom_coll, ...)]
AS SELECT ...
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` (si certaines colonnes résultat du `select` sont des expressions, il faut renommer ces colonnes dans le `select`, ou spécifier les noms de colonne de la vue). Une fois créée, la vue est stockée dans la base et s'utilise comme une table.

Le `CHECK OPTION` permet de vérifier que la mise à jour ou l'insertion faite à travers la vue ne produisent que des lignes qui font partie de la sélection de la vue.

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 ;
```

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.

Exemple : Modification des salaires du département 10 à travers la vue `emp10`.

```
UPDATE emp10
SET sal = sal *1.1;
```

Alsi

Toutes les lignes de la table emp, telles que le contenu de la colonne n_dept est égal à 10 seront modifiées.

3. Renommer

On peut renommer une vue par la commande :

```
RENAME ancien_nom TO nouveau_nom;
```

4. Supprimer

Une vue peut être détruite par la commande :

```
DROP VIEW nom_vue;
```