

Leçon 3 : Les vues

AYIKPA KACOUTCHY JEAN : Enseignant -
Chercheur



Table des matières

I - 1- Généralité sur les vues	3
II - Application 1 :	4
III - 2- Traitement sur les vues	5
IV - Application 2 :	7
Solutions des exercices	8

1- Généralité sur les vues

I

■ Définition : 1.1- Définition

Une vue est une table virtuelle à partir des tables d'une base de données existante, 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`.

La vue représente de cette façon une sorte d'intermédiaire entre la base de données et l'utilisateur. Cela a de nombreuses conséquences :

- Une sélection des données à afficher
- Une restriction d'accès à la table pour l'utilisateur, c'est-à-dire une sécurité des données accrue
- Un regroupement d'informations au sein d'une entité

1.2- Intérêt des vues

Les vues peuvent être utilisées pour différentes raisons. Elles permettent de :

- Contrôler l'intégrité en restreignant l'accès aux données pour améliorer la confidentialité.
- Partitionnement vertical et/ou horizontal pour cacher des champs aux utilisateurs, ce qui permet de personnaliser l'affichage des informations suivant le type d'utilisateur.
- Masquer la complexité du schéma.
- Indépendance logique des données, utile pour donner aux utilisateurs l'accès à un ensemble de relations représentées sous la forme d'une table. Les données de la vue sont alors des champs de différentes tables regroupées, ou des résultats d'opérations sur ces champs.
- Modifier automatiquement des données sélectionnées (`sum()`, `avg()`, `max()`,...).
- Manipuler des valeurs calculées à partir d'autres valeurs du schéma.
- Conserver la structure d'une table si elle doit être modifiée.

Le schéma peut ainsi être modifié sans qu'il ne soit nécessaire de changer les requêtes du côté applicatif.

Application 1 :



Exercice

[solution n°1 p.8]

Un vue permet :

- ☐ d'accélérer les requêtes de recherche
- ☐ une exploitation visuelle des données
- ☐ accélérer la mise à jour des données

2- Traitement sur les vues

III

2.1- Création d'une vue

CREATE VIEW nom_vue

AS requete_select;

ou

CREATE VIEW nom_vue(liste_colonnes)

AS requete_select;

requete_select : liste des champs que doit avoir la vue

liste_colonnes : liste des différentes colonnes que doivent avoir la vue si les colonnes ne sont pas spécifiées alors les colonnes du select sont par défaut les colonnes de la vue

Exemple :

Soit la table suivant :

Client(idcli,nomcli,prencli,sexecli,datenaiscli)

Créer une vue qui affiche le nom, prénoms et sexe des client

Créer une vue qui affiche le nom, prénoms et sexe des clients nés avant l'année 2000

```
1 1)
2 CREATE VIEW liste_client
3 AS Select nomcli,prencli,sexecli from client;
4 //***** Affichage de la vue *****/
5 Select * from liste_client;
6
7 2) CREATE VIEW client_majeur
8 AS Select nomcli,prencli,sexecli from client
9 where datenaiscli<'2000/01/01';
10 //***** Affichage de la vue *****/
11 Select * from client_majeur;
```

2.2- Modification d'une vue

ALTER VIEW nom_vue [(liste_colonnes)]

AS requete_select ;

NB : La vue doit bel et bien existé.

Exemple :

Soit la table suivant :

Client(idcli,nomcli,prencli,sexecli,datenaiscli)

Modifier la vue client_majeur pour qu'elle n'affiche que les clients nés avant 2002

```
1 CREATE VIEW client_majeur
2 AS Select nomcli,prencli,sexecli from client
3 where datenaiscli<'2002/01/01';
4 //***** Affichage de la vue *****/
5 Select * from client_majeur;
6
```

2.4- Suppression d'une vue

DROP VIEW nom_vue ;

NB : La vue doit bel et bien existé.

Exemple :

Soit la table suivant :

Client(idcli,nomcli,prencli,sexecli,datenaiscli)

Supprimer la vue liste_client

```
1 DROP VIEW liste_client;
```

Application 2 :

IV

Exercice

[solution n°2 p.8]

Énoncé :

Soit la table suivante :

LIVRE(NUMLIV,NOMAUT ,TITRELIV,GENRELIV,PRIXLIV)

- Créer la table livre en créant index sur le champ nom auteur
- Créer une vue listelivresup1500 qui permettra de donner la liste des livre qui on un montant de plus de 1500

NB : Respecter l'ordre de création des propriétés

Solution :

CREATE TABLE LIVRE (

NUMLIV INT(10) not null primary key ,

TITRELIV VARCHAR(10),

GENRELIV VARCHAR(50),

PRIXLIV INT(50),

PRIXLIV Integer ,

);

CREATE VIEW listelivresup1500

AS Select NUMLIV , TITRELIV , GENRELIV , PRIXLIV from livre

where PRIXLIV > 1500

Solutions des exercices



> **Solution** n°1

Exercice p. 4

Un vue permet :

- ☐ d'accélérer les requêtes de recherche
- ☒ une exploitation visuelle des données
- ☐ accélérer la mise à jour des données

> **Solution n°2**

Exercice p. 7

Énoncé :

Soit la table suivante :

LIVRE(NUMLIV,NOMAUT ,TITRELIV,GENRELIV,PRIXLIV)

- Créer la table livre en créant index sur le champ nom auteur
- Créer une vue listelivresup1500 qui permettra de donner la liste des livre qui on un montant de plus de 1500

NB : Respecter l'ordre de création des propriétés

Solution :

```
CREATE TABLE LIVRE (
NUMLIV VARCHAR(10) not null primary key ,
NOMAUT VARCHAR(10),
TITRELIV VARCHAR(50),
GENRELIV VARCHAR(50),
PRIXLIV Integer ,
);

CREATE VIEW listelivresup1500
AS Select NOMAUT,NOMAUT,TITRELIV , GENRELIV from livre
where PRIXLIV>1500 ;
```