

Les vues

Définition

Une vue c'est une table dont les données ne sont pas physiquement stockées mais se réfèrent à des données stockées dans d'autres tables. C'est une fenêtre sur la base de données permettant à chacun de voir les données comme il le souhaite.

On peut ainsi définir plusieurs vues à partir d'une seule table ou créer une vue à partir de plusieurs tables. Une vue est interprétée dynamiquement à chaque exécution d'une requête qui y fait référence.

Avantages :

- Protéger les données.
- Simplification de requêtes.
- Données éparpillées dans un même objet.
- Mises à jour dynamiques

Les vues permettent de protéger l'accès aux tables en fonction de chacun des utilisateurs. On utilise une vue sur une table et on interdit l'accès aux tables. C'est donc un moyen efficace de protéger les données

Les vues permettent de simplifier la commande SELECT avec les sous requêtes complexes. On peut créer une vue pour chaque sous-requête complexe, ce qui facilite sa compréhension

Il est possible de rassembler dans un seul objet (vue) les données éparpillées.

Une vue se comporte dans la plupart des cas comme une table. On peut utiliser une vue comme source d'information dans les commandes SELECT, INSERT, UPDATE ou DELETE. Une vue est créée à l'aide d'une sous-requête.

Syntaxe

```
CREATE [OR REPLACE ][FORCE] VIEW <nom_de_la_vue> AS <sou_requête> [WITH CHECK OPTION]
```

OR REPLACE : commande optionnelle qui indique lors de la création de la vue de modifier la définition de celle-ci ou de la créer si elle n'existe pas

FORCE : permet de créer la vue même si les sources de données n'existent pas.

WITH CHECK OPTION : cette option permet de contrôler l'accès à la vue et par conséquent à la table dont elle est issue.

Exemples1 :

```
create view etudiantsinfo as
select * from etudiants where codep =420;
```

L'instruction suivante provoque la mise à jour de la vue etudiantsinfo:

```
Insert into etudiants values (101,'Lenouveau','Etudiant',420);
```

L'instruction suivante provoque la mise à jour de la table Etudiants.

```
insert into etudiantsinfo values (103,'PasEnInfo','PasenInfo',410);
```

Les contraintes sur les tables au moment de l'insertion s'appliquent sur la vue également.

On peut également supprimer ou modifier des enregistrements par la vue ou par la table

L'instruction suivante ne va pas renvoyer une erreur. Même si le code programme n'est pas 420 (ne correspond pas à la clause WHERE)

```
insert into etudiantsinfo values (103,'PasEnInfo','PasenInfo',410);
```

- Exemple 2:

```
create view InfoResultats as
select Et.numad,nom, prenom, titre Cours, note
from ((etudiants Et inner join Resultats Re on Et.numad = Re.numad)
inner join Cours Co on Re.Codecours =Co.codecours);
```

L'instruction suivante permet de modifier la note dans la table Resultat. La note dans la Vue est également change.

```
update Resultats set note =65 where codecours='KEE' and numad =10;
```

Attention : On ne peut pas faire de update sur la vue InfoResultats

Contraintes d'utilisation

Vous ne pouvez pas :

- Insérer dans une table à partir d'une vue, si la table contient des champs NOT NULL et qui n'apparaissent dans la vue il y'aura violation de contraintes d'intégrité
- Insérer ou mettre à jour dans la vue si la colonne en question est un résultat calculé.
- Insérer ou mettre à jour (INSERT, UPDATE et DELETE) si la vue contient les clauses GROUP BY ou DISTINCT.
- Utiliser une vue comme source de données pour INSERT, UPDATE et DELETE si elle définit avec :
 - Une jointure
 - Une opération d'ensemble
 - Une clause GROUP BY, CONNECT BY, DISTINCT de l'ordre SELECT
 - Une fonction de groupe (SUM, MAX...)

De manière générale, on peut dire que les commandes SQL INSERT, UPDATE et DELETE ne peuvent s'appliquer qu'à une vue utilisant une table avec restriction et sélection.

L'option WITH CHECK OPTION

L'option WITH CHECK OPTION : Lors d'une instruction INSERT INTO, UPDATE, DELETE par la vue, si la clause WHERE n'est pas respectée alors le SGBD renvoie une erreur.

```
Create view etudiantsinfo2 as select * from etudiants where  
codep =420 with check option;
```

L'instruction suivante va renvoyer une erreur car 410 n'est pas 420.

```
insert into etudiantsinfo2 values (104,'PasEnInfo','PasenIfo',410);
```

Détruire une VUE :

`DROP VIEW nom_de_vue` permet de supprimer la vue

`RENAME ancien_nom TO nouveau_NOM` renomme une vue.