

Les vues (view)

- 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. C'est un objet de la base de données.

- Rôles:

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

- Syntaxe, → avec une sous requête, comme la création de tables.

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

Les views

- **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.

Les views

Exemple 1:

```
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.

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

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

Les views

- Remarque:

L'instruction suivante ne va pas renvoyé 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;
```

Les views

- Attention ! 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 (pas toujours)
 - 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 INSERT, UPDATE et DELETE ne peuvent s'appliquer qu'à une vue utilisant une table avec restriction et sélection

Les views

- 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','Pasenlfo',410);
```