

Les vues : Conclusion et exemples

Ce qu'il faut absolument retenir (après avoir lu les notes de cours)

1. Les vues sont des objets de la base de données. On utilise la commande CREATE pour créer une vue. On utilise la commande DROP pour détruire une vue.
2. Une vue se crée à l'aide d'une sous-requête.
3. Les vues sont dynamiques : ce qui veut dire que la vue se met à jour automatiquement lorsque la table (les tables) dont elle issue est mise à jour.
4. Les vues servent à simplifier l'écriture de requêtes.
5. Les vues servent à mettre ensemble des données éparpillées.
6. Les vues servent à protéger les données de la BD (cours de session 3).

Exemple1 :

Dans une question du laboratoire 6, on vous demandait : Qui sont les **deux** meilleurs étudiants en CONCEPTION DE BASES DE DONNÉES ?

Évidemment, la requête peut être considérée comme complexe. Mais si vos données étaient regroupées quelque part ... ce serait simple.

Si vous avez répondu correctement à la question, vous avez écrit ceci :

```
SELECT * FROM (  
SELECT et.numad,nom, prenom, note FROM  
((etudiants et INNER JOIN resultats rs ON et.numad=rs.numad)  
INNER JOIN cours cr ON rs.codecours =cr.codecours)  
WHERE cr.titrecours='CONCEPTION DE BASES DE DONNÉES'  
ORDER BY note DESC  
)  
WHERE ROWNUM <=2;
```

En examinant la requête, on pourrait se poser la question : Peut-on garder les données de la requête la plus interne (requête en jaune) quelque part ? La réponse est : OUI dans une VUE.

Dans cette requête, les données éparpillées sont mises ensemble.

On va utiliser une VUE pour garder des valeurs renvoyées par une sous-requête. Ce qui veut dire que l'on va stocker des résultats intermédiaires.

Si je fais :

```
CREATE VIEW VresultatsKED AS
SELECT et.numad,nom, prenom, note FROM
((etudiants et INNER JOIN resultats rs ON et.numad=rs.numad)
INNER JOIN cours cr ON rs.codecours =cr.codecours)
WHERE cr.titrecours='CONCEPTION DES BASES DE DONNÉES'
ORDER BY note DESC ;
```

1. Exécuter le code pour créer la VUE
2. Examiner le contenu de la VUE VresultatsKED par un simple SELECT *

VresultatsKED

| | NUMAD | NOM | PRENOM | NOTE |
|---|-------|---------|---------|------|
| 1 | 11 | POIRIER | JUTEUX | 85 |
| 2 | 10 | PATOCHE | ALAIN | 65 |
| 3 | 12 | FAFAR | CHANTAL | 60 |

À partir de là, vous pouvez extraire les données que vous voulez. Les deux meilleurs étudiants, le meilleur étudiant, etc...

La requête est : (on va l'appeler R1)

```
SELECT * FROM VresultatsKED WHERE ROWNUM<=2;
```

Facile. N'est ce pas ?

Question :Et si j'ajoute des données dans la table Resultats ou je modifie la note d'un étudiants en KED, la requête R1 est-elle toujours valable ?

Réponse : Aucun problème, la requête R1 est toujours valable puisque la VUE est DYNAMIQUE.

On teste ?

Étape1 : D'abord, on commence par ajouter deux étudiants en informatique (car dans la table Etudiants, il n'y que trois étudiants. Si on avait plus d'étudiants, cette étape serait inutile)

Exécuter ceci :

```
INSERT INTO ETUDIANTS VALUES(20,'Saquet','Frodon',420);
INSERT INTO ETUDIANTS VALUES(21,'LeMalin','Simba',420);
```

Étape 2 : on attribut des notes en KED pour les étudiants que l'on vient d'ajouter.

1. Par curiosité, faites un SELECT * FROM Resultats (la table) juste pour voir le contenu.
2. Maintenant, exécuter les instructions suivantes :

```
INSERT INTO resultats (numad,codecours,note) VALUES(20,'KED',90);
INSERT INTO resultats (numad,codecours,note) VALUES(21,'KED',80);
COMMIT;
```

3. Puis faites un `SELECT * FROM VresultatsKED;` (on utilise la vue)

En principe vous avez ceci. On voit bien que les étudiants dont NUMAD =20 et NUMAD =21 ont été ajoutés dans la VUE. Les **VUES sont DYNAMIQUES**.

| NUMAD | NOM | PRENOM | NOTE |
|-------|---------|---------|------|
| 20 | Saquet | Frodon | 90 |
| 11 | POIRIER | JUTEUX | 85 |
| 21 | LeMalin | Simba | 80 |
| 10 | PATOCHE | ALAIN | 65 |
| 12 | FAFAR | CHANTAL | 60 |

Évidemment si vous faites un `SELECT * from Resultats,` vous aurez la même chose.

Application (exercice) :

1. Exécuter l'instruction suivante pour ajouter un étudiant :
`INSERT INTO ETUDIANTS VALUES (22,'Lenouveau','Etudiant',420);`
2. Exécuter l'instruction suivante pour ajouter une note à l'étudiant qu'on vient de créer
`INSERT INTO resultats (numad,codecours,note) VALUES(22,'KED',90);`
3. Écrire la requête qui ramène le nom et le prénom des étudiants qui ont la plus haute note en KED. Attention, il y a deux étudiants qui ont la note maximale. Votre requête doit marcher qu'il y ait un, deux ou N étudiants. Ce qui veut dire que le ROWNUM ne marchera pas ici.
 - a. Utiliser la VUE **VresultatsKED**
 - b. Écrire la requête sans utiliser la VUE.
4. Conclusion