

Introduction aux bases de données

Les vues, pour simplifier les requêtes

Les vues

Plan de la séance

- Retour sur la dernière séance:
 - Point de vue de l'étudiant
 - Point de vue de l'enseignant.
- Rappels:
 - Les sous-requêtes
- Les vues:
 - Définition
 - Avantages
 - Syntaxe
 - Exemples
 - Supprimer/renommer une vue
 - Conclusion

Les vues

Définition

- Une vue est objet de la base de donnée. Nous pouvons la voir comme 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.

Les vues

Avantages

- 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
- 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
- Une vue se comporte dans la plupart des cas comme une table. On peut utiliser une vue comme source de données dans les commandes SELECT, INSERT, UPDATE ou DELETE.

Les vues

Syntaxe:

```
CREATE [OR REPLACE ][FORCE] VIEW  
<nom_de_la_vue> AS <sous_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. (Sécurité des données)

Les vues

Exemple 1, regrouper des données éparpillées.

La vue suivante regroupe des données issues de trois tables: etudiants, resultats et cours.

- *numad, nom, prenom* sont dans la table etudiants
- *Titrecours* est dans la table cours.
- *Note* est dans la table resultats

```
CREATE VIEW vResultats as
SELECT etudiants.numad,nom, prenom, titrecours,note
FROM ((etudiants
INNER JOIN resultats ON etudiants.numad=resultats.numad)
INNER JOIN cours ON cours.codecours =resultats.codecours);
```

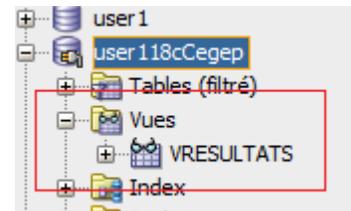
Les vues

Exemple 1 (suite), regrouper des données éparpillées.

Lorsque je fais `SELECT * FROM vResultats;` on obtient le résultats de la figure ci-après.

Lorsque vous allez dérouler l'onglet Vues, vous remarquerez que votre vue est créée

NUMAD	NOM	PRENOM	TITRECOURS	NOTE
10	Patoche	Alain	Programmation orientée objet	85
10	Patoche	Alain	Introduction aux bases de données	65
10	Patoche	Alain	Projet dirigé	60
11	Poirier	Juteux	Programmation orientée objet	70
11	Poirier	Juteux	Programmation Web	75
11	Poirier	Juteux	Introduction aux bases de données	85
12	Fafar	Anick	Programmation orientée objet	60
12	Fafar	Anick	Programmation Web	75
12	Fafar	Anick	Introduction aux bases de données	77
16	Saturne	Francis	Programmation Web	66
16	Saturne	Francis	Projet dirigé	88



Les vues

Exemple 2, la mise à jour est dynamique

Lorsque je fais `SELECT * FROM vResultats;` on obtient le résultats de la figure suivante.

On remarque qu'il y a 11 enregistrements. Et remarquez la note de l'étudiant numéro 10 dans le cours de Projet dirigé

NUMAD	NOM	PRENOM	TITRECOURS	NOTE
10	Patoche	Alain	Programmation orientée objet	85
10	Patoche	Alain	Introduction aux bases de données	65
10	Patoche	Alain	Projet dirigé	60
11	Poirier	Juteux	Programmation orientée objet	70
11	Poirier	Juteux	Programmation Web	75
11	Poirier	Juteux	Introduction aux bases de données	85
12	Fafar	Anick	Programmation orientée objet	60
12	Fafar	Anick	Programmation Web	75
12	Fafar	Anick	Introduction aux bases de données	77
16	Saturne	Francis	Programmation Web	66
16	Saturne	Francis	Projet dirigé	88

Les vues

Exemple 2, (suite)

Si on fait les instructions suivantes:

```
insert into resultats values (16,'KB6',85);
```

```
insert into resultats values (12,'KBE',90);
```

```
update resultats set note =62
```

```
where numad =10 and codecours ='KBE';
```

Puis on fait un `SELECT * FROM vResultats` on obtient la figure ci-après

On remarque que la vue s'est mise jour automatiquement: Les deux enregistrements ont été ajoutés et la note de l'étudiant 10 dans le cours de Projet dirigé est passée à 62.

NUMAD	NOM	PRENOM	TITRECOURS	NOTE
10	Patoche	Alain	Programmation orientée objet	85
10	Patoche	Alain	Introduction aux bases de données	65
10	Patoche	Alain	Projet dirigé	62
11	Poirier	Juteux	Programmation orientée objet	70
11	Poirier	Juteux	Programmation Web	75
11	Poirier	Juteux	Introduction aux bases de données	85
12	Fafar	Anick	Programmation orientée objet	60
12	Fafar	Anick	Programmation Web	75
12	Fafar	Anick	Introduction aux bases de données	77
12	Fafar	Anick	Projet dirigé	90
16	Saturne	Francis	Programmation Web	66
16	Saturne	Francis	Introduction aux bases de données	85
16	Saturne	Francis	Projet dirigé	88

Conclusion: Les vues sont dynamiques. Si les données des tables dont la vue est issues sont modifiées alors les données de la vue sont modifiées en conséquence.

Les vues

Exemple 3, simplification de requêtes.

On cherche:

Q1 - Les 3 meilleurs étudiants (les 3 meilleurs notes) dans le cours de Introduction aux bases de données.

Q2- Les étudiants avec la meilleurs note (la plus haute note)

Pour répondre à la question Q1 on utilise une sous requête dans la clause FROM.

```
SELECT * FROM (  
    SELECT etudiants.numad,nom, prenom, titre Cours,note  
    FROM ((etudiants  
        INNER JOIN resultats ON etudiants.numad=resultats.numad)  
        INNER JOIN Cours ON Cours.code Cours =resultats.code Cours)  
    WHERE titre Cours='Introduction aux bases de données'  
    ORDER BY note desc  
    )  
WHERE ROWNUM<=3;
```

Les vues

Exemple 3, (suite)

On voit bien que la requête précédente est très longue.

Le résultat de la sous-requête est le suivant:

	NUMAD	NOM	PRENOM	TITRECOURS	NOTE
1	11	Poirier	Juteux	Introduction aux bases de données	85
2	16	Saturne	Francis	Introduction aux bases de données	85
3	12	Fafar	Anick	Introduction aux bases de données	77
4	10	Patoche	Alain	Introduction aux bases de données	65

La question: Peut-on mettre la requête en rouge (la sous-requête) dans une vue afin de simplifier la requête et de pouvoir réutiliser le résultat pour d'autres requêtes? Comme par exemple la question Q2.

La réponse est oui.

Les vues

Exemple 3, (suite)

Pour simplifier la requête, on va d'abord créer la vue des résultats des étudiants dans le cours d'introduction aux bases de données: Cette vue sera ordonnée par ordre décroissant des notes

```
CREATE VIEW VresultatsKB6 AS
    SELECT etudiants.numad,nom, prenom, titre Cours,note
    FROM ((etudiants
            INNER JOIN resultats ON etudiants.numad=resultats.numad)
            INNER JOIN Cours ON Cours.codeCours =resultats.codeCours)
    WHERE titre Cours='Introduction aux bases de données'
    ORDER BY note desc;
```

Puis, on répond aux questions Q1 et Q2 (voir la page suivante)

Les vues

Exemple 3, (suite)

Pour répondre à Q1 - les 3 meilleurs étudiants (les 3 meilleurs notes) dans le cours de Introduction aux bases de données

```
SELECT * FROM VresultatsKB6 WHERE ROWNUM <=3;
```

Pour répondre à Q2: Les étudiants avec la meilleurs note (la plus haute note)

```
SELECT * FROM VresultatsKB6 WHERE note =  
    (  
        SELECT MAX(note) FROM VresultatsKB6  
    );
```

Les vues

Supprimer et renommer une vue.

Pour détruire une vue: `DROP VIEW nom_de_la_Vue`

Exemple : **`DROP VIEW VresultatsKB6;`**

Pour renommer une vue: `RENAME ancien_nom TO nouveau`

Exemple:

`RENAME VresultatsKB6 to ResultatsIntroBD;`

Les vues

Conclusion

- 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.
- Une vue se crée à l'aide d'une sous-requête.
- 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.
- Les vues servent à simplifier l'écriture de requêtes.
- Les vues servent à mettre ensemble des données éparpillées.
- Les vues servent à protéger les données de la BD (cours de session 3).

Les vues



Conclusion



Questions