

Aharouite Ahmed

PABD / TP03 Oracle

Table des matières

1. Introduction.....	3
2. Manipulation de la base de données.....	3
2.1. Vue mono-table sans contrainte.....	3
2.2. Vue mono-table sans contrainte avec requête imbriquée IN / NOT IN.....	5
3. Conclusion.....	7

1. Introduction

L'objet de ce TP est d'étudier le mécanisme de vues mono-table qui supportent ainsi les mises à jour.

```
CREATE OR REPLACE VIEW <nomvue>
AS SELECT ...
FROM <nomtable>
WHERE ...;
```

Les vues sont utilisées principalement pour :

- modifier la présentation de la base de données à certains utilisateurs. On parlera alors de "Schéma Externe" en opposition du "Schéma Physique" correspondant à l'implémentation des tables.
- simplifier la construction de certaines requêtes complexes. Les vues sont alors créées et utilisées comme des résultats intermédiaires. Une vue pouvant être utilisée dans la création d'une autre vue.
- assurer la confidentialité de la base de données en construisant des sous-schémas adaptés à chaque utilisateur (ou famille d'utilisateur). Dans un premier temps on réduit le champ de vision de certaines tables (restriction horizontale, verticale ou mixte) en construisant des vues puis, ensuite, on réduit les actions aux vues avec l'ordre GRANT.

2. Manipulation de la base de données

2.1. Vue mono-table sans contrainte

(1) Créez une vue V_Joueur_FRA (nj, prenom, nom, genre, date_ns) à partir de la table Joueur, ne comportant que les joueurs et joueuses de nationalité française.

```
CREATE VIEW V_Joueur_FRA
(nj, prenom, nom, genre, date_ns) AS
SELECT nj, pre, nom, gen, dns FROM joueur
WHERE nat = 'FRA';
```

(2) Vérifiez le contenu de la vue par une interrogation de cette dernière.

```
SELECT * FROM V_Joueur_FRA;
```

	NJ	PRENOM	NOM	GENRE	DATE_NS
1	235	DIANE	PARRY	F	01/09/02
2	244	AUDREY	ALBIE	F	24/10/94
3	251	FIONA	FERRO	F	12/03/97
4	252	KRISTINA	MLADENOVIC	F	14/05/93
5	257	FABRICE	MARTIN	H	11/09/86
6	276	AMANDINE	HESSE	F	16/01/93
7	277	BENJAMIN	BONZI	H	09/06/96
8	8	GILLES	SIMON	H	27/12/84
9	28	ELLIOT	BENCHETRIT	H	02/10/98
10	37	MAXIME	JANVIER	H	18/10/96

22	109	BENOIT	PAIRE	H	08/05/89
23	111	PIERRE-HUGUES	HERBERT	H	18/03/91
24	116	RICHARD	GASQUET	H	18/06/86
25	118	CORENTIN	MOUTET	H	19/04/99
26	137	CAROLINE	GARCIA	F	16/10/93
27	163	CHLOÉ	PAQUET	F	01/07/94
28	174	SELENA	JANICIJEVIC	F	23/07/02
29	180	HARMONY	TAN	F	11/09/97
30	210	JESSIKA	PONCHET	F	26/09/96
31	222	ALIZÉ	CORNET	F	22/01/90
32	223	PAULINE	PARMENIER	F	31/01/86

(3) Augmentez le paramètre d'environnement LONG avec la commande :

```
SET LONG 1000 ;
```

Puis vérifiez dans le dictionnaire des données USER_VIEWS le nom et la définition de la vue V_Joueur_FRA

```
SELECT * FROM USER_VIEWS ;
```

VIEW_NAME	TEXT_LENGTH	TEXT
V_JOUEUR_FRA	60	SELECT nj, pre, nom, gen, dns FROM joueur WHERE nat = 'FRA'

(4) Insérez-vous en tant que nouveau joueur au travers de la vue.

```
INSERT INTO V_Joueur_FRA  
VALUES (999, 'Ahmed', 'Aharouite', 'H', '20/09/01'); 1 ligne inséré.
```

(5) Vérifiez le contenu de la vue puis de la table. Que constatez-vous ? Expliquez ?

On remarque l'absence du joueur ajouté précédemment. Cela s'explique par le fait que ce nouveau joueur n'admet pas de nationalité et donc comme sa nationalité est nulle, ce joueur ne peut s'afficher dans la vue puisque la condition de la vue n'est pas respectée (WHERE nat = 'FRA'). La nationalité ne doit pas être nulle...

```
SELECT * FROM V_Joueur_FRA; Toutes les lignes extraites : 32 en 0,004 secondes
```

(6) Tentez de vous supprimer au travers de la vue. Que constatez-vous ? Expliquez ?

Cela ne fonctionne pas car ce joueur n'existe pas dans la vue et donc supprimer quelque chose qui n'existe revient à ne rien supprimer.

```
DELETE FROM V_Joueur_FRA WHERE nj=999; 0 lignes supprimé.
```

(7) Tentez de supprimer le joueur TSONGA au travers de la vue V_Joueur_FRA. Que constatez-vous ? Expliquez ?

Cela ne fonctionne pas car la suppression de ce joueur viole des contraintes. Ce joueur est référencé dans d'autres tables en tant que clé étrangère (Match).

```
DELETE FROM V_Joueur_FRA WHERE nom='NADAL';
```

(8) Annulez votre transaction par ROLLBACK pour revenir à l'état initial, sans les deux clients.

```
ROLLBACK; Annulation (rollback) terminée.
```

(9) Vérifiez que les contenus de la vue et de la table soient redevenus ceux initialement présents dans la base.

```
SELECT * FROM V_Joueur_FRA; Toutes les lignes extraites : 32 en 0,004 secondes
```

```
SELECT * FROM Joueur; Toutes les lignes extraites : 285 en 0,034 secondes
```

2.2. Vue mono-table sans contrainte avec requête imbriquée IN / NOT IN

(1) Créez une vue V_Finaliste (nj, prenom, nom, genre, date_ns) à partir des tables Joueur et Match, ne comportant que les joueurs et joueuses ayant joué une finale.

```
CREATE VIEW V_Finaliste
(nj, prenom, nom, genre, date_ns) AS
SELECT nj, pre, nom, gen, dns FROM joueur
WHERE nj IN (
SELECT nj1 FROM match WHERE tour='finale'
UNION
SELECT nj2 FROM match WHERE tour='finale'
UNION
SELECT nj1b FROM match WHERE tour='finale'
UNION
SELECT nj2b FROM match WHERE tour='finale');

```

View V_FINALISTE créé(e).

(2) Vérifiez le contenu de la vue par une interrogation de cette dernière.

```
SELECT * FROM V_Finaliste;
```

	NJ	PRENOM	NOM	GENRE	DATE_NS
4	263	KEVIN	KRAWIETZ	H	24/01/92
5	264	ANDREAS	MIES	H	21/08/90
6	269	GABRIELA	DABROWSKI	F	01/04/92
7	271	YINGYING	DUAN	F	03/07/89
8	278	LATISHA	CHAN	F	17/08/89
9	279	IVAN	DODIG	H	02/01/85
10	285	MATE	PAVIC	H	04/07/93
11	33	DOMINIC	THIEM	H	03/09/93
12	39	JEREMY	CHARDY	H	12/02/87
13	128	RAFAEL	NADAL	H	03/06/86
14	141	TIMEA	BABOS	F	10/05/93
15	160	ASHLEIGH	BARTY	F	24/04/96
16	175	SAISAI	ZHENG	F	05/02/94

(3) Vérifiez dans USER_VIEWS le nom et la définition de la vue.

```
SELECT * FROM USER_VIEWS;
```

VIEW_NAME	TEXT_LENGTH	TEXT
1 V_JOUEUR_FRA	60	SELECT nj, pre, nom, gen, dns FROM joueur WHERE nat = 'FRA'
2 V_FINALISTE	244	SELECT nj, pre, nom, gen, dns FROM joueur WHERE nj IN (SELECT nj1 FROM match WHERE tour='

(4) Insérez-vous en tant que finaliste au travers de la vue.

```
INSERT INTO V_Finaliste
VALUES (999, 'Ahmed', 'Aharouite', 'H', '20/09/01');
```

1 ligne inséré.

(5) Vérifiez le contenu des deux vues, celui de la table Joueur et de la table Match en vous limitant aux tuples des finales. Que constatez-vous ? Expliquez ?

```
SELECT * FROM V_Finaliste;

```

Toutes les lignes extraites : 16 en 0,005 secondes

```
SELECT * FROM V_Joueur_FRA;

```

Toutes les lignes extraites : 32 en 0,004 secondes

```
SELECT * FROM match
```

Toutes les lignes extraites : 286 en 0,021 secondes

NJ	PRE	NOM	GEN	CLA	NAT	VNS	PNS	DNS	MAIN	TAILLE	NBD	NBS
999	Ahmed	Aharouite	H	(null)	(null)	(null)	(null)	20/09/01	(null)	(null)	(null)	(null)

```
SELECT * FROM match  
WHERE tour='finale';
```

Toutes les lignes extraites : 5 en 0,008 secondes

On remarque que s'insérer en tant que nouveau joueur au travers de la vue revient à s'insérer dans la table joueur et non dans la vue. Cela s'explique par le fait que la vue est une vue mono-table sans contrainte avec requête imbriquée et donc comme ce nouveau joueur n'appartient pas à la table match, il ne peut être affiché dans la vue **V_Finaliste** puisque la condition de la vue n'est pas respectée mais le joueur créé s'insère dans la table **joueur**.

(6) Annulez la transaction avec ROLLBACK.

```
ROLLBACK; Annulation (rollback) terminée.
```

(7) Modifiez la vue précédente en utilisant une jointure relationnelle.

```
CREATE OR REPLACE VIEW V_Finaliste  
(nj, prenom, nom, genre, date_ns)  
AS SELECT nj, pre, nom, gen, dns FROM joueur, match  
WHERE nj IN (nj1,nj2,nj1b,nj2b) AND tour = 'finale';
```

View V_FINALISTE créé(e).

(8) Tentez de vous insérer en tant que nouveau joueur au travers de la vue. Que constatez-vous ? Expliquez ?

```
INSERT INTO V_Finaliste  
VALUES (555, 'AHMED', 'AHAROUITE', 'H', '20/09/01');
```

Rapport d'erreur -

Erreur SQL : ORA-01779: impossible de modifier une colonne correspondant à une table non protégée par clé

On remarque la présence d'une erreur, s'insérer en tant que nouveau joueur au travers de la vue ne fonctionne pas. Cela s'explique par le fait que la vue a été créée voir modifiée par le biais d'une jointure relationnelle et donc comme ce nouveau joueur n'appartient pas à la table match, il ne peut être insérer/affiché dans la vue **V_Finaliste** puisque la condition de la vue n'est pas respectée :

(WHERE nj IN (nj1 ,nj2 ,nj1b , nj2b) AND tour = 'finale')

3. Conclusion

Ce TP03 a pour but de mettre en application ce que nous avons déjà vu au cours des TD précédents en IBD, c'est à dire la manipulation de base de données grâce au langage SQL (Structured Query Language).

Une vue est une table virtuelle, c'est-à-dire dont les données ne sont pas stockées dans une table de la base de données, et 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 création d'une vue se fait grâce à la clause `CREATE VIEW` suivie du nom que l'on donne à la vue, puis du nom des colonnes dont on désire agrémenter cette vue (il faut autant de redéfinitions de colonne qu'il y en aura en sortie), puis enfin d'une clause `AS` précédant la sélection. La syntaxe d'une vue ressemble donc à ceci :

```
CREATE VIEW Nom_de_la_Vue  
(colonnes)  
AS 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