

Aharouite Ahmed

PABD / TP04 Oracle

Table des matières

| | |
|--------------------------------------------|---|
| 1. Introduction..... | 3 |
| 2. Manipulation de la base de données..... | 3 |
| 2.1. Vue multi-table avec jointure..... | 3 |
| 2.2. Vue multi-table avec agrégation..... | 4 |
| 3. Conclusion..... | 6 |

1. Introduction

L'objet de ce TP est d'étudier le mécanisme de vues multi-tables.

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

2. Manipulation de la base de données

2.1. Vue multi-table avec jointure

(1) Re-créez la vue V_Finaliste (nj, prenom, nom, genre, date_ns, tableau) à partir des tables Joueur, Match et Tableau comportant les joueurs/joueuses finalistes du tournoi dans un des tableaux.

```
CREATE OR REPLACE VIEW V_Finaliste
(nj, prenom, nom, genre, date_ns, tableau) AS
SELECT nj, pre, nom, gen, dns, lib FROM joueur, match M, tableau T
WHERE nj IN (nj1,nj2,nj1b,nj2b) AND tour='finale' AND M.nt=T.nt ;
```

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

```
SELECT * FROM V_Finaliste;
```

| | NJ | PRENOM | NOM | GENRE | DATE_NS | TABLEAU |
|----|-----|----------|-------------|-------|----------|------------------|
| 1 | 252 | KRISTINA | MLADENOVIC | F | 14/05/93 | Double Dames |
| 2 | 271 | YINGYING | DUAN | F | 03/07/89 | Double Dames |
| 3 | 141 | TIMEA | BABOS | F | 10/05/93 | Double Dames |
| 4 | 175 | SAISAI | ZHENG | F | 05/02/94 | Double Dames |
| 5 | 257 | FABRICE | MARTIN | H | 11/09/86 | Double Messieurs |
| 6 | 263 | KEVIN | KRAWIETZ | H | 24/01/92 | Double Messieurs |
| 7 | 264 | ANDREAS | MIES | H | 21/08/90 | Double Messieurs |
| 8 | 39 | JEREMY | CHARDY | H | 12/02/87 | Double Messieurs |
| 9 | 269 | GABRIELA | DABROWSKI | F | 01/04/92 | Double Mixte |
| 10 | 278 | LATISHA | CHAN | F | 17/08/89 | Double Mixte |
| 11 | 279 | IVAN | DODIG | H | 02/01/85 | Double Mixte |
| 12 | 285 | MATE | PAVIC | H | 04/07/93 | Double Mixte |
| 13 | 228 | MARKETA | VONDROUSOVA | F | 28/06/99 | Simple Dames |
| 14 | 160 | ASHLEIGH | BARTY | F | 24/04/96 | Simple Dames |
| 15 | 33 | DOMINIC | THIEM | H | 03/09/93 | Simple Messieurs |
| 16 | 128 | RAFAEL | NADAL | H | 03/06/86 | Simple Messieurs |

(3) Vérifiez dans le dictionnaire des données USER_VIEWS le nom et la définition de la vue V_Finaliste.

```
SELECT * FROM USER_VIEWS;
```

| | VIEW_NAME | TEXT_LENGTH | TEXT |
|---|--------------|-------------|-----------------------------------------------------|
| 1 | V_JOUEUR_FRA | 60 | SELECT nj, pre, nom, gen, dns FROM joueurWHERE nat |
| 2 | V_FINALISTE | 130 | SELECT nj, pre, nom, gen, dns, lib FROM joueur, mat |

(4) Tentez de vous insérer en tant que nouveau joueur au travers de la vue comme finaliste du Simple Messieurs. Que constatez-vous ? Expliquez ?

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

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 pour être 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 inséré/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')

(5) Tentez de supprimer les finalistes du tableau Simple Messieurs. Que constatez-vous ? Expliquez ?

```
DELETE FROM V_Finaliste
WHERE tableau = 'Simple Messieurs';
```

Il est impossible d'effectuer une suppression des finalistes du tableau Simple Messieurs à partir de la vue. Cela ne fonctionne pas car la suppression de ces joueurs viole des contraintes et nous indique une erreur. Ces joueurs sont référencés dans d'autres tables en tant que clé étrangère (Match) et donc sont protégés par ces clés.

(6) Modifiez cette vue afin de la rendre mono-table.

```
CREATE OR REPLACE VIEW V_Finaliste
(nj, prenom, nom, genre, date_ns, tableau) AS
SELECT nj, pre, nom, gen, dns,
CASE
WHEN null is null THEN (SELECT lib FROM Match M, Tableau T
WHERE J.nj IN (M. nj1, M.nj2, M.nj1b,M.nj2b)
AND tour = 'finale' AND M.nt = T.nt)
END
FROM joueur J;
```

View V_FINALISTE créé(e).

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

```
INSERT INTO V_Finaliste (nj, prenom, nom, genre, date_ns)
VALUES ( SEQ_JOUEUR.nextval, 'Ahmed', 'Aharouite', 'H', '20/09/01');
|1 ligne inséré.
```

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

```
SELECT * FROM V_Finaliste;
```

| | | | | | | |
|----|-----|-------|-----------|---|----------|--------|
| 62 | 303 | Ahmed | Aharouite | H | 20/09/01 | (null) |
|----|-----|-------|-----------|---|----------|--------|

(9) Tentez de supprimer les finalistes du tableau Simple Messieurs. Que constatez-vous ? Expliquez ?

```
DELETE FROM V_Finaliste
WHERE tableau = 'Simple Messieurs';
```

Il est impossible d'effectuer une suppression des finalistes du tableau Simple Messieurs à partir de la vue. Cela ne fonctionne pas car la suppression de ces joueurs viole des contraintes et nous indique une erreur. Ces joueurs sont référencés dans d'autres tables en tant que clé étrangère (Match) et donc sont protégés par ces clés.

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

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

2.2. Vue multi-table avec agrégation

(1) Créez une vue V_Pays (cio, nom, nbh, nbf) qui contient pour chaque pays le nombre de joueurs (nbh) et de joueuses (nbf).

```
CREATE VIEW V_Pays
(cio, nom, nbh, nbf) AS
SELECT cio, nom, SUM(nbh), SUM(nbf) FROM (
SELECT cio, P.nom, COUNT(J.nat) AS nbh, 0 AS nbf
FROM pays P, joueur J
WHERE P.cio = J.nat AND gen = 'H'
GROUP BY cio, P.nom, J.nat
UNION
SELECT cio, P.nom, 0 AS nbh, COUNT(J.nat) AS nbf
FROM pays P, joueur J
WHERE P.cio = J.nat AND gen = 'F'
GROUP BY cio, P.nom, J.nat)
GROUP BY cio, nom;
```

View V_PAYS créé(e).

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

```
SELECT * FROM V_Pays;
```

| CIO | NOM | NBH | NBF | | | | |
|--------|------------|-----|-----|--------|------------------------|---|----|
| 1 GEO | Géorgie | 1 | 0 | 45 LUX | Luxembourg | 0 | 1 |
| 2 PUR | Porto Rico | 0 | 1 | 46 BEL | Belgique | 3 | 3 |
| 3 SUI | Suisse | 3 | 2 | 47 CHN | Chine (Rép. populaire) | 0 | 7 |
| 4 UKR | Ukraine | 1 | 6 | 48 HUN | Hongrie | 1 | 1 |
| 5 CAN | Canada | 1 | 3 | 49 POL | Pologne | 1 | 3 |
| 6 GER | Allemagne | 13 | 7 | 50 ROU | Roumanie | 2 | 4 |
| 7 ITA | Italie | 9 | 2 | 51 RUS | Russie (Fédération de) | 3 | 13 |
| 8 JPN | Japon | 3 | 3 | 52 SVK | République Slovaque | 2 | 5 |
| 9 KAZ | Kazakhstan | 2 | 3 | | | | |
| 10 POR | Portugal | 1 | 0 | | | | |
| 11 SWE | Suède | 1 | 2 | | | | |

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

```
SELECT * FROM USER_VIEWS;
```

| VIEW_NAME | TEXT_LENGTH | TEXT |
|-----------|-------------|-----------------------------------------------|
| V_PAYS | 336 | SELECT cio, nom, SUM(nbh), SUM(nbf) FROM (SEL |

(4) Tentez d'insérer un nouveau pays de code IUT et nommé IUT de Blagnac. Que constatez-vous ? Expliquez ?

```
INSERT INTO V_Pays (cio, nom)
VALUES ( 'IUT', 'IUT de Blagnac');
```

On remarque la présence d'une erreur, insérer un nouveau pays de code IUT et le nommer IUT de Blagnac au travers de la vue ne fonctionne pas. Cela s'explique par le fait que la vue a été créée en tant que Vue multi-table avec agrégation et donc comme ce nouveau pays n'admet aucun joueur, il ne peut être insérer/afficher dans la vue **V_Pays** puisque la condition de la vue n'est pas respectée :

(WHERE P.cio = J.nat)

(5) Annulez la transaction avec ROLLBACK.

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

(6) En utilisant la commande CASE, rendre la vue V_Pays mono-table.

```
CREATE OR REPLACE VIEW V_Pays                                |View V_PAYS créé(e) .
(cio, nom, nbh, nbf) AS
SELECT cio, P.nom,
CASE
WHEN null is null THEN (SELECT COUNT (J.nj) AS nbh
FROM joueur J
WHERE P.cio = J.nat AND gen = 'H'
GROUP BY cio, P.nom)
END
,
CASE
WHEN null is null THEN (SELECT COUNT (J.nj) AS nbf
FROM joueur J
WHERE P.cio = J.nat AND gen = 'F'
GROUP BY cio, P.nom)
END
FROM pays P
GROUP BY cio, P.nom;
```

(7) Insérer un nouveau pays de code IUT et nommé IUT de Blagnac.

```
INSERT INTO V_Pays (cio, nom)                                |1 ligne inséré.
VALUES ( 'IUT', 'IUT de Blagnac');
```

(8) Vérifiez sa présence dans la vue.

```
SELECT * FROM V Pays;
```

| | CIO | NOM | NBH | NBF |
|---|-----|----------------|--------|--------|
| 1 | GEO | Géorgie | 1 | (null) |
| 2 | UZB | Ouzbékistan | (null) | (null) |
| 3 | PUR | Porto Rico | (null) | 1 |
| 4 | IUT | IUT de Blagnac | (null) | (null) |

(9) Annulez la transaction avec ROLLBACK.

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

3. Conclusion

Ce TP04 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).

Durant ce TP, on a pu remarquer que la présence de certaines erreurs lors d'insertion (d'un nouveau joueur/pays) et de suppression (de joueurs). Ces insertions qui nous indique une erreur sont dus à la violation de contraintes liées aux clés. On ne peut insérer une nouvelle ligne (un joueur/pays) au travers d'une vue si certains de ces attributs (les joueurs/pays) admettent des clés étrangères et que la condition de la vue n'est pas respectée. La suppression de joueurs à partir d'une car cela viole des contraintes, ces joueurs sont référencés dans d'autres tables en tant que clé étrangère (Match) et donc sont protégés par ces clés.