

Morgado-Samagaio Jonathan

PABD / TP3 Oracle

Table des matières

Table des matières	2
1. Introduction.....	3
2. Vue mono-table sans contrainte	3
3. Vue mono-table sans contrainte avec requête imbriquée IN / NOT IN	7
4. Conclusion	10

1. Introduction

Dans ce TP, nous allons voir comment utiliser les vue pour simplifier des requêtes complexes et assurer la confidentialité de la base de données.

2. Vue mono-table sans contrainte

- 1) On commence par créer une vue ne comportant que les pays d'Europe. On utilise la commande suivante :

```
--1.1  
CREATE VIEW V_Pays_EU (nom, iso3, iso2,  
AS SELECT nom, iso3, iso2, capitale  
FROM Pays
```

Nous pouvons voir comment se construit la création d'une vue. Nous avons CREATE VIEW pour annoncer sa création. On donne ensuite le nom de la vue et les différents champs entre parenthèse. On effectue ensuite la requête qui sera associé à cette vue.

- 2) On interroge maintenant cette vue :

```
--1.2  
SELECT *  
FROM V_Pays_EU
```

On peut voir qu'on interroge une vue comme n'importe quelle table. On obtient :

NOM	ISO	IS	C
-----	---	--	-
Pays-Bas	NLD	NL	A
Serbie	SRB	RS	B
Allemagne	DEU	DE	B
Slovaquie	SVK	SK	B
Belgique	BEL	BE	B
Roumanie	ROU	RO	B
Danemark	DNK	DK	C
Irlande	IRL	IE	D
Slovénie	SVN	SI	L
Royaume-Uni	GBR	GB	L
Luxembourg	LUX	LU	L
Biélorussie	BLR	BY	M
Chypre	CYP	CY	N
France	FRA	FR	P
Islande	ISL	IS	R
Lettonie	LVA	LV	R
Italie	ITA	IT	R
Bosnie-Herzégovine	BIH	BA	S
Bulgarie	BGR	BG	S
Suède	SWE	SE	S
Estonie	EST	EE	T
Albanie	ALB	AL	T
Liechtenstein	LIE	LI	V
Malte	MLT	MT	L
Pologne	POL	PL	V
Vatican	VAT	VA	C
Autriche	AUT	AT	V
Lituanie	LTU	LT	V
Croatie	HRV	HR	Z
Andorre	AND	AD	A
Espagne	ESP	ES	M
Grèce	GRC	GR	A
Kosovo		XK	P
Monaco	MCO	M	M
Portugal	PRT	PT	L
Saint-Marin	SMR	SM	S
Ukraine	UKR	UA	K
Suisse	CHE	CH	B

- 3) On veut vérifier le nom et la définition de notre vue. Pour cela on utilise la commande suivante :

```
--1.3
SET LONG 1000;
SELECT VIEW_NAME, TEXT
FROM USER_VIEWS
```

On obtient :

VIEW_NAME	TEXT
V_PAYS_EU	SELECT nom, iso3, iso2, capitale, nomc FROM Pays

La vue est bien correcte.

- 4) On essaye d'insérer un nouveau pays dans notre vue. On utilise la commande suivante :

```
--1.4
INSERT INTO V_Pays_EU
VALUES ('IUT Blagnac', 'IUT', 'UT', 'Dept Informatique')
```

Cependant, nous obtenons une erreur :

```
Erreur commençant à la ligne: 1 de la commande -
INSERT INTO V_Pays_EU
VALUES ('IUT Blagnac', 'IUT', 'UT', 'Dept Informatique')
Rapport d'erreur -
```

En effet, il y a une contrainte sur la table pays qui interdit d'avoir un nom de continent null.

- 5) Pour pouvoir insérer ce pays, nous allons retirer la contrainte. On utilise la commande :

```
--1.5
CREATE OR REPLACE VIEW V_Pays_EU (nom, iso3, iso2, capitale, nomc)
AS SELECT nom, iso3, iso2, capitale, nomc
FROM Pays
```

- 6) On essaye maintenant d'ajouter le pays. On utilise la commande :

```
--1.6
INSERT INTO V_Pays_EU
VALUES ('IUT Blagnac', 'IUT', 'UT', 'Dept Informatique')
```

Nous avons bien ajouté le nom du continent et le pays à été insérer.

- 7) On vérifie à nouveau le contenu de la vue :

```
--1.7
SELECT *
FROM V Pays E
```

Et on obtient :

NOM	ISO	IS	CAPITALE
Pays-Bas	NLD	NL	Amsterdam
Serbie	SRB	RS	Belgrade
Allemagne	DEU	DE	Berlin
Slovaquie	SVK	SK	Bratislava
Belgique	BEL	BE	Bruxelles
Roumanie	ROU	RO	Bucarest
Danemark	DNK	DK	Copenhague
Irlande	IRL	IE	Dublin
Slovénie	SVN	SI	Ljubljana
Royaume-Uni	GBR	GB	Londres
Luxembourg	LUX	LU	Luxembourg
Biélorussie	BLR	BY	Minsk
Chypre	CYP	CY	Nicosie
France	FRA	FR	Paris
Islande	ISL	IS	Reykjavik
Lettonie	LVA	LV	Riga
Italie	ITA	IT	Rome
Bosnie-Herzégovine	BIH	BA	Sarajevo
Bulgarie	BGR	BG	Sofia
Suède	SWE	SE	Stockholm
Estonie	EST	EE	Tallinn
Albanie	ALB	AL	Tirana
Liechtenstein	LIE	LI	Vaduz
Malte	MLT	MT	La Valette
Pologne	POL	PL	Varsovie
Vatican	VAT	VA	Cité du Vatican
Autriche	AUT	AT	Vienne
Lituanie	LTU	LT	Vilnius
Croatie	HRV	HR	Zagreb
Andorre	AND	AD	Andorre-la-Vieille
Espagne	ESP	ES	Madrid
Grèce	GRC	GR	Athènes
Kosovo		XK	Pristina
Monaco	MCO	M	Monaco
Portugal	PRT	PT	Lisbonne
Saint-Marin	SMR	SM	Saint-Marin
Ukraine	UKR	UA	Kiev
Suisse	CHE	CH	Berne
Hongrie	HUN	HU	Budapest

8) On va maintenant supprimer ce pays :

```
--1.8
DELETE FROM V_Pays_EU
WHERE nom = 'IUT Blac
```

Le pays à bien été supprimé.

- 9) On essaye de supprimer la France à partir de la vue :

```
--1.9
DELETE FROM V_Pays
WHERE nom = 'France'
```

Nous obtenons l'erreur suivante :

```
Erreur commençant à la ligne: 1 de la commande -
DELETE FROM V_Pays_EU
WHERE nom = 'France'
Rapport d'erreur -
ORA-02292: violation de contrainte (SCOTT.PAYS) du référentiel FOREIGN KEY
```

Il y a une contrainte sur la table frontière qui empêche la suppression de la France.

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

- 1) On va maintenant créer une vue contenant des requêtes imbriquées. On veut créer une vue comprenant les montagnes d'Europe. On utilise la commande suivante :

```
--2.1
CREATE VIEW V_Montagne_EU(numero, nom, altitu
AS SELECT nm, nom, altitude, chaine
FROM Montagne
WHERE nm IN (SELECT nm
FROM Localiser
WHERE nomp IN (SELECT nom
FROM Pays
```

La vue a bien été créée.

- 2) On va donc vérifier son contenu en l'interrogeant :

NUMERO	NOM	AL'
150	Elbrouz	
151	Dykh-Taou	
152	Chkhara	
153	Kochtan-Taou	
155	Mont Blanc	
156	Djimara	
157	Ouilpata	
158	Pointe Dufour	
159	Tikhtengen	
161	Dom des Mischabel	
162	Jailik	
163	Salinan	
164	Weisshorn	
165	Mont Teboulo	
166	Cervin	
167	Bazardüzü	
168	Mont Shani	
169	Tepli	
170	Dent Blanche	
171	Grand Combin	
172	Mont Diklo	
173	Finsteraarhorn	
175	Zinalrothorn	
176	Grandes Jorasses	
177	Rimpfischhorn	
178	Aletschhorn	
179	Strahlhorn	
180	Dent d'Hérens	
181	Breithorn	
182	Jungfrau	
183	Mont Addala-Shukhgelmeer	
184	Mont Dyultydag	
185	Aiguille Verte	
186	Mönch	

3) Comme pour la dernière vue, nous allons vérifier sa définitions :


```
--2.3
SET LONG 1000;
SELECT VIEW_NAME, TEXT
FROM USER_VIEWS
```

On obtient :

VIEW_NAME	TEXT
V_MONTAGNE_EU	SELECT nm, nom, altitude, chaine FROM Montagne WHERE nm IN (SELECT nm FROM Localiser WHERE nomp IN (SELECT n FROM Pays

- 4) On ajoute maintenant une nouvelle montagne :

```
--2.4
INSERT INTO V_Montagne_EU
VALUES (seq_montagne.NEXTVAL, 'Pic d'Aneto', 3404,
```

La ligne s'est insérée sans problèmes.

- 5) On étudie la table montagne et notre vue. Nous pouvons voir que dans la vue, notre ajout n'apparaît pas. Cependant, il est bien présent dans la table Montagne. Cela est dû au fait que, pour qu'une montagne soit considérée en Europe, il faut que l'on associe son numéro avec un pays dans la table localiser. Nous ne l'avons pas fait donc elle n'est pas présente dans la vue.

- 6) On annule notre dernier ajout avec un rollback :

```
--2.6
ROLLBACK
```

- 7) Nous devons modifier notre vue pour pouvoir ajouter le pays. Cependant, les IN empêchent de changer les éléments des tables à l'intérieur. Nous devons donc passer par une requête multi-table. On utilise la commande :

```
--2.7
CREATE OR REPLACE VIEW V_Montagne_EU(nom, nom, altitude, chaine, P.nom)
AS SELECT M.nm, M.nom, M.altitude, M.chaine, P.nom
FROM Montagne M, Localiser L, Pays P
WHERE M.nm = L.nm AND L.nomp = P.nomp AND P.pays = 'Europe'
```

La table est donc bien créée.

- 8) On essaie d'ajouter notre montagne avec le pays :

```
--2.8
INSERT INTO V_Montagne_EU
VALUES (seq_montagne.NEXTVAL, 'Pic d'Aneto', 3404, 'Pyréné
```

Nous obtenons une erreur :

```

Erreur commençant à la ligne: 1 de la commande -
INSERT INTO V_Montagne_EU
VALUES (seq_montagne.NEXTVAL, 'Pic d''Aneto', 3404, 'Pyrénées', 'France')
Erreur à la ligne de commande: 1 Colonne: 1
Rapport d'erreur -
Erreur SQL : ORA-01779: impossible de modifier une colonne correspondant à une table
01779. 00000 - "cannot modify a column which maps to a non key-preserved table"
*Cause:      An attempt was made to insert or update columns of a join view which
              maps to a non key-preserved table

```

Nous obtenons cette erreur car nous ne pouvons pas modifier les vues multi tables.

4. Conclusion

Dans ce TP, nous avons pu voir comment créer et modifier des vues. Cela nous permet d'enregistrer des requêtes sous un nom et de faciliter des requêtes complexes.