

Morgado-Samagaio Jonathan

IBD / TP10 Oracle

Table des matières

Table des matières	2
1. Introduction.....	3
2. Requêtes.....	3
3. Conclusion	8

1. Introduction

Dans ce TP, nous allons à nouveau faire des requêtes multi-table. Nous allons voir comment écrire les jointures en SQL.

2. Requêtes

- 1) Pour commencer le TP, nous allons en premier faire une jointure des tables Localiser et Montagne pour obtenir les montagnes localisé en France. Pour cela, nous allons utiliser la commande suivante :

```
SELECT nom
FROM Localiser, Montagne
WHERE Localiser.nom = Montagne.nom AND localiser.ne
```

Nous pouvons voir que le FROM prend deux tables, c'est donc bien une requête multi-table. Il y a aussi la condition de jointure dans le WHERE avec l'égalité des deux colonnes. Nous obtenons donc :

NOM
Mont Blanc
Grandes Jorasses
Aiguille Verte

- 2) Pour cette requête, nous allons ajouter plus de conditions dans le WHERE pour obtenir les pays européen frontaliers à la France. Nous utilisons donc la commande suivante :

```
SELECT P.*
FROM Pays P, Frontiere F
WHERE F.pomp = 'France' AND f.nomf = P.nom AND P.pomc
```

La condition de jointure est toujours présente avec l'égalité entre deux tables et nous avons ajouté une nouvelle condition. Nous obtenons donc :

NOM	CAPITALE	SUPERFICIE P
Pays-Bas	Amsterdam	41530
Allemagne	Berlin	357386
Suisse	Berne	41285
Belgique	Bruxelles	30688
Luxembourg	Luxembourg	2586
Italie	Rome	301336
Andorre	Andorre-la-Vieille	468
Espagne	Madrid	505911
Monaco	Monaco	?

- 3) Cette fois-ci, l'information demandée est presque similaire à la précédente. En effet, nous devons obtenir les pays non européen frontaliers à la France. Pour cela, nous allons juste prendre l'inverse d'une des conditions de la commande précédente :

```
SELECT P.*
FROM Pays P, Frontiere F
WHERE F.pomp = 'France' AND f.nomf = P.nom AND P.pomc
```

Nous pouvons voir qu'il y a un '!' pour la condition du continent qui nous permet de prendre tous les pays qui n'appartiennent pas au continent Europe. Nous obtenons donc :

NOM	CAPITALE	SUPERFICIE	POF
Brésil	Brasilia	8514876	21
Guinéma	Bayamantou	163270	

- 4) Nous allons commencer à joindre trois tables dans cette requête. Nous cherchons les montagnes localisées en Europe. Le résultat des requêtes commençant à être long, nous allons utiliser la commande SET PAGESIZE pour avoir un meilleur affichage. La commande est donc :

```
SET PAGESIZE 1000;
SELECT DISTINCT M.*
FROM Montagne M, Localiser L, Pays P
WHERE L.nm = M.nm AND L.nomp = P.nom AND P.nomc
```

Nous pouvons que, cette fois-ci, il y a deux conditions de jointure pour joindre les trois tables ensemble. Il y a aussi l'ajout du ORDER BY Pour trié le résultat. Nous obtenons :

NM NOM	ALTITUDE	CHAINE
185 Aiguille Verte	4122	Alpes
178 Aletschhorn	4195	Alpes
187 Barre des Écrins	4102	Alpes
167 Bazardüzü	4466	Caucase
181 Breithorn	4164	Alpes
166 Cervin	4478	Alpes
152 Chkhara	5193	Caucase
170 Dent Blanche	4357	Alpes
180 Dent d'Hérens	4171	Alpes
156 Djimara	4780	Caucase
161 Dom des Mischabel	4545	Alpes
151 Dykh-Taou	5205	Caucase
150 Elbrouz	5642	Caucase
173 Finsteraarhorn	4274	Alpes
171 Grand Combin	4314	Alpes
190 Grand Paradis	4061	Alpes
176 Grandes Jorasses	4208	Alpes
162 Jailik	4533	Caucase
182 Jungfrau	4158	Alpes
153 Kochtan-Taou	5150	Caucase
193 Lagginhorn	4010	Alpes
186 Mönch	4107	Alpes
183 Mont Addala-Shukhgelmeer	4152	Caucase
155 Mont Blanc	4809	Alpes
172 Mont Diklo	4285	Caucase
184 Mont Dyultydag	4127	Caucase
168 Mont Shani	4451	Caucase
165 Mont Teboulo	4493	Caucase
189 Ober Gabelhorn	4063	Alpes
157 Ouilpata	4649	Caucase
191 Piz Bernina	4049	Alpes
158 Pointe Dufour	4634	Alpes
177 Rimpfischhorn	4199	Alpes
163 Salinan	4508	Caucase

- 5) Nous restons dans le même principe de requête avec l'interrogation de trois tables. Nous cherchons les montagnes localisées dans les pays frontaliers de la Suisse. Nous utilisons donc :

```
SELECT F.nomp, M.nom
FROM Localiser L, Montagne M, Frontiere F
WHERE L.nm = M.nm AND L.nomp = F.nomp AND F.nom
```

La commande est similaire à la précédente. Nous obtenons :

NOMP	NOM
France	Grandes Jorasses
France	Barre des Écrins
France	Aiguille Verte
France	Mont Blanc
Italie	Grand Paradis
Italie	Dent d'Hérens
Italie	Grandes Jorasses
Italie	Cervin
Italie	Breithorn

- 6) Pour cette requête, nous allons utiliser le OR dans les conditions. Le OR n'étant pas prioritaire, il faut faire attention lors de son utilisation à ce que les conditions testées soient celles que l'on veut. Nous utilisons la commande :

```
SELECT DISTINCT M.nom
FROM Montagne M, Localiser L
WHERE M.nm = L.nm AND (L.nomp='France' OR L.nomp
```

Nous pouvons donc voir les parenthèses pour le OR pour que la condition correcte soit respectée. Nous obtenons :

NOM
Grandes Jorasses
Grand Paradis
Aiguille Verte
Barre des Écrins
Cervin
Dent d'Hérens
Mont Blanc
Breithorn

- 7) Nous compliquons encore plus les choses avec une requête utilisant quatre tables. Nous cherchons les fleuves qui traversent les pays frontaliers à la Russie mais en affichant les continents, le nom du pays et le nom du fleuve. Nous utilisons la commande suivante :

```
SELECT P.nomc, P.nom, F.nom
FROM Pays P, Fleuve F, Frontiere FR, Traverser T
WHERE F.nf = T.nf AND T.nomp = FR.nomp AND FR.nomf = 'Russie' AND
```

La commande ressemble à la question 5 car c'est le même principe sur des tables différentes. Il y a juste une condition de jointure de plus. Nous obtenons :

NOMC	NOM	NOM
Asie	Azerbaïdjan	La Koura
Asie	Chine	Le Tarim
Asie	Chine	Le fleuve Jaune
Asie	Chine	Le fleuve Rouge
Asie	Chine	Le Hai He
Asie	Chine	Le Yang Tsé Kiang
Asie	Chine	Le Yalu
Asie	Chine	L'Ili
Asie	Chine	L'Indus
Asie	Chine	Le Xi Jiang
Asie	Chine	La Liao
Asie	Chine	Le Mékong
Asie	Chine	Le Tumen
Asie	Chine	Le Salouen
Asie	Chine	L'Amour
Asie	Chine	Le Brahmapoutre
Asie	Corée du Nord	Le Yalu
Asie	Corée du Nord	Le Tumen
Asie	Géorgie	Le Rioni
Asie	Géorgie	La Koura
Asie	Géorgie	le Terek
Asie	Kazakhstan	L'Ili
Asie	Kazakhstan	l'Oural
Asie	Kazakhstan	Le Syr-Daria
Asie	Mongolie	L'Ienisseï
Europe	Biélorussie	le Dniepr
Europe	Biélorussie	le Niémen
Europe	Biélorussie	la Daugava
Europe	Estonie	la Narva
Europe	Finlande	le Torne
Europe	Finlande	le Kemijoki
Europe	Lettonie	la Daugava
Europe	Lettonie	la Lielupe
Europe	Lituanie	le Niémen

- 8) Nous allons pour la première fois utiliser COUNT dans des requêtes multi-table. Nous cherchons le nombre de fleuve qui traversent chaque pays européen. Nous utilisons donc la commande :

```
SELECT P.nom, COUNT(F.nom) AS NBF
FROM Pays P, Fleuve F, Traverser T
WHERE F.nf = T.nf AND T.nomp = P.nom AND P.nomp = F.nomp
GROUP BY P.nom
```

Nous pouvons voir que son utilisation ne change pas. Nous obtenons donc :

NOM
Russie
France
Espagne
Italie
Allemagne
Portugal
Ukraine
Suède
Biélorussie
Belgique
Royaume-Uni
Grèce
Pays-Bas
Bulgarie
Irlande
Finlande
République Tchèque
Lettonie
Suisse
Pologne
Autriche
Moldavie
Norvège
Roumanie
Macédoine du Nord
Serbie
Liechtenstein
Estonie

- 9) Nous allons utiliser plusieurs COUNT et un SUM dans le SELECT. Nous cherchons le nombre de fleuve, le nombre de pays et la longueur cumulée des fleuves par continent.

Nous utilisons la commande :

```
SELECT P.nomc, COUNT(F.nom) AS NBF, COUNT(DISTINCT P.nom) AS NBP, SUM(F.l) AS NL
FROM Pays P, Fleuve F, Traverser T
WHERE F.nf = T.nf AND T.nomp = P.nom
```

Nous pouvons voir l'utilisation du DISTINCT dans le COUNT pour ne pas compter plusieurs fois les pays. Nous obtenons donc :

NOMC	NBF	NB
Océanie	12	
Asie	78	2
Amérique	53	1
Afrique	83	4

- 10) Cette question nous demande de récupérer le pays le plus montagneux. Cependant, j'ai quelques difficultés à y répondre. Ma commande est :


```

SELECT P.nomc
FROM Pays P, Montagne M, Localiser L
WHERE M.nm = L.nm AND L.nomp = P.nom
HAVING MAX(COUNT(M.nom)) = (SELECT MAX(CO
FROM Montagne

```

SQL Developpeur renvoie l'erreur :

```

Erreur commençant à la ligne: 1 de la commande -
SELECT P.nomc
FROM Pays P, Montagne M, Localiser L
WHERE M.nm = L.nm AND L.nomp = P.nom
HAVING MAX(COUNT(M.nom)) = (SELECT MAX(COUNT(nom))
FROM Montagne
GROUP BY nom)
Erreur à la ligne de commande: 4 Colonne: 12
Rapport d'erreur -
Erreur SQL : ORA-00935: imbrication de fonction de groupe
ORA-00935 - "group function is nested too deeply"

```

Je n'arrive pas à comprendre d'où vient mon erreur.

3. Conclusion

Dans ce TP, nous avons pu voir comment écrire les opérations de jointure. Nous avons pu interroger plusieurs tables en même temps avec des conditions de jointure dans le WHERE. Nous avons aussi vu l'utilisation du AND et du OR pour les opérations logiques. Je n'ai malheureusement pas pu finir la dernière question.