

Morgado-Samagaio Jonathan

## **IBD / TP11 Oracle**

## Table des matières

Table des matières .....	2
1. Introduction.....	3
2. Auto-jointures .....	3
3. Thêta-Jointure .....	5
4. Jointure externe .....	7
5. Conclusion .....	9

## 1. Introduction

Dans ce TP, nous allons à nouveau faire des requêtes multi-table. Nous allons voir comment écrire différents types de jointures en SQL. Nous allons voir les auto-jointures, les thêta-jointures et les jointures externes.

## 2. Auto-jointures

Les autos jointures sont des jointures qui nécessitent l'utilisation de deux même tables.

- 1) Nous allons commencer par obtenir les montagnes se situant en France et en Italie. Pour cela, nous allons utiliser la commande suivante :

```
SELECT M.nom
FROM Montagne M, Localiser L1, Localiser L2
WHERE L1.nomp = 'France'
AND L2.nomp = 'Italie'
AND L1.nm = L2.nm
```

Nous pouvons donc voir l'apparition de deux Localiser dans le FROM et que les conditions de jointure utilisent ces deux tables. Nous obtenons le résultat suivant :

NOM
Mont Blanc
Grandes Jambes

- 2) Nous continuons avec une requête similaire puisqu'il faut cette fois ci obtenir les fleuves qui traversent la France et L'Allemagne.

```
SELECT F.nom
FROM Fleuve F, Traverser T1, Traverser T2
WHERE T1.nomp = 'France'
AND T2.nomp = 'Allemagne'
AND T1.nf = T2.nf
```

La commande est très similaire. En effet, nous avons juste changé les tables utilisées et le nom des pays. Nous obtenons :

NOM
le Rhin

- 3) Nous allons compliquer encore plus avec l'utilisation de deux versions de deux tables différentes. Nous cherchons donc les pays d'Asie et d'Europe qui ont au moins une montagne commune. Nous utilisons la commande :

```

SELECT DISTINCT P1.nom, P2.nom
FROM Pays P1, Pays P2, Localiser L1, Localiser L2
WHERE P1.nomc = 'Asie'
AND P2.nomc = 'Europe'
AND L1.nomp = P1.nom
AND L2.nomp = P2.nom
AND M.nomp = L1.nomp

```

Nous voyons bien dans le FROM les deux Localiser et les deux Pays avec pour chacun des conditions de jointures. On obtient :

NOM	NOM
Géorgie	Russie
Azerbaïdjan	Russie

- 4) Une nouvelle fois une requête similaire. On cherche les pays d'Amérique et d'Europe qui ont une frontière commune. On utilise la commande :

```

SELECT P1.nom
FROM Pays P1, Pays P2, Frontiere F1, Frontiere F2
WHERE P1.nomc = 'Amérique'
AND P2.nomc = 'Europe'
AND F1.nomp = P1.nom
AND F2.nomp = P2.nom

```

Et nous renvoie :

NOM
Brésil
Guyana

- 5) Cette question est la plus compliquée des auto-jointures. On a besoin d'utiliser deux versions de trois tables différentes. On cherche les pays ayant au moins une montagne en commun et un fleuve en commun avec la Russie. On utilise la commande suivante :

```

SELECT DISTINCT P2.nom
FROM Pays P1, Pays P2, Localiser L1, Localiser L2, Traverser T1, Traverser T2
WHERE P1.nom = 'Russie'
AND L1.nomp = 'Russie'
AND T1.nomp = 'Russie'
AND L2.nomp = P2.nom
AND T2.nomp = P2.nom

```

Le nombre de table ayant augmenté, le nombre de conditions augmente aussi. On obtient :

NOM
Russie
Géorgie

### 3. Thêta-Jointure

Les thêta-jointures sont des jointures qui utilisent des opérateurs différents comme les <>, <, > etc.

- 1) On commence en cherchant les pays qui ont au moins une montagne en commun avec la France en retirant la France du résultat. On utilise la commande suivante :

```
SELECT DISTINCT L2.nomp
FROM Localiser L1, Localiser L2
WHERE L1.nomp = 'France'
and L1.nm = L2.nm
```

Nous pouvons voir l'opérateur <> qui permet de retirer France du résultat. On obtient :

NOMP
-----
Italie

- 2) Cette fois-ci, on cherche les pays non Asiatiques qui ont au moins une montagne en commun avec un pays d'Asie. Cela nécessite l'utilisation de deux versions de la table Pays. On utilise la commande :

```
SELECT DISTINCT P2.nom
FROM Localiser L1, Localiser L2, Pays P1
WHERE P1.nomc = 'Asie'
AND L1.nm = L2.nm
AND L1.nomp = P1.nom
AND L2.nomp = P2.nom
```

On a encore une fois l'opérateur pour enlever les pays du continent Asie. On obtient :

NOM
-----
Russie

- 3) On veut effectuer une requête similaire mais cette fois si avec les fleuves et en affichant le nom des deux pays. On cherche donc les pays non européens qui ont un fleuve en commun avec un pays d'Europe. On utilise la commande :

```
SELECT DISTINCT P1.nom, P2.nom
FROM Pays P1, Pays P2, Traverser T1, Traverser T2
WHERE P1.nomc = 'Europe'
AND T1.nf = T2.nf
AND T1.nomp = P1.nom
AND T2.nomp = P2.nom
```

La commande est donc très similaire. On obtient :

NOM	NOM
France	Brésil
Russie	Chine
Russie	Kazakhstan
France	Suriname
Russie	Corée du Nord
Russie	Mongolie
Bulgarie	Turquie
Russie	Géorgie
Grèce	Turquie

- 4) Cette fois-ci, on cherche les pays de continent différents ayant une frontière commune. On utilise la commande :

```
SELECT DISTINCT P1.nom, P2.nom
FROM Pays P1, Pays P2, Front
WHERE F.nomp = P1.nom
AND F.nomf = P2.nom
```

Et on obtient :

NOM	NOM
Turquie	Grèce
Palestine	Égypte
Espagne	Maroc
Géorgie	Russie
Turquie	Bulgarie
Égypte	Israël
Chine	Russie
Russie	Chine
Russie	Kazakhstan
France	Brésil
Israël	Égypte
Azerbaïdjan	Russie
Brésil	France
Mongolie	Russie
Russie	Azerbaïdjan
France	Suriname
Suriname	France
Russie	Corée du Nord
Bulgarie	Turquie
Russie	Géorgie
Russie	Mongolie
Corée du Nord	Russie
Maroc	Espagne

On a utilisé un SET PAGESIZE Pour modifié l'affichage.

#### 4. Jointure externe

Les jointures externes permettent d'ajouter des tuples qui n'ont pas de correspondant dans l'autre table.

- 1) On commence par chercher les continents ne possédant pas de pays. On utilise la commande suivante :

```
SELECT C.nom FROM Pays P, Cont
WHERE C.nom = P.nomc (+)
```

On voit donc l'apparition du (+) qui permet d'ajouter les éléments qui sont normalement ignorés lors de la jointure. Le IS NULL permet de garder seulement l'élément normalement ignoré. On obtient :

NOM
Antarctique

- 2) On cherche maintenant les pays d'Amérique qui n'ont pas de montagne. On utilise la commande suivante :

```
SELECT P.nom
FROM Pays P, Localis
WHERE P.nomc = 'Amér
AND P.nom = L.nomp (+)
```

La commande est donc similaire. On obtient :



NOM
Costa Rica
Belize
Suriname
Paraguay
Trinité-et-Tobago
Colombie
Saint-Vincent-et-les-Grenadines
Panama
République dominicaine
Antigua-et-Barbuda
Sainte-Lucie
Haïti
Salvador
Venezuela
Jamaïque
Barbade
Bahamas
Guatemala
Brésil
Grenade
Saint-Christophe-et-Niévès
Cuba
Équateur
Uruguay

- 3) Cette fois-ci, la commande va être plus compliquée. En effet, on va utiliser un COUNT en plus de ce que l'on a vu. On cherche pour chaque pays d'Amérique le nombre de fleuve même pour ceux qui n'en ont pas. On utilise la commande :

```
SELECT P.nom, COUNT(T.nf)
FROM Pays P, Traverser T
WHERE P.nomc = 'Amérique'
AND T.nomp(+) = P.nom
GROUP BY P.nom
```

Nous pouvons voir l'apparition du COUNT dans le SELECT. Nous n'avons aussi pas besoin du IS NULL vu qu'on veut tout garder. Nous avons aussi besoin d'un GROUP BY. On obtient :



NOM
États-Unis
Brésil
Canada
Argentine
Colombie
Pérou
Suriname
Mexique
Guyana
Paraguay
Venezuela
Honduras
Guatemala
Uruguay
Salvador
République dominicaine
Sainte-Lucie
Barbade
Cuba
Nicaragua
Saint-Vincent-et-les-Grenadines
Bolivie
Haïti
Bahamas
Panama
Antigua-et-Barbuda
Équateur
Dominique
Costa Rica
Saint-Christophe-et-Niévès

## 5. Conclusion

Dans ce TP, nous avons pu voir comment écrire les de nouvelles opérations de jointure. Nous avons vu les auto-jointures qui ont besoin de deux versions d'une même table, nous avons aussi vu les thêta-jointures qui utilisent de nouveaux opérateurs comme  $\ltimes$  et enfin nous avons vu les jointures externes qui permettent de récupérer des éléments normalement ignorés lors des jointures avec  $(+)$ .