Morgado-Samagaio Jonathan

**IBD / TP11 Oracle**

Table des matières

[Table des matières 2](#_Toc59032194)

[1. Introduction 3](#_Toc59032195)

[2. Auto-jointures 3](#_Toc59032196)

[3. Thêta-Jointure 5](#_Toc59032197)

[4. Jointure externe 7](#_Toc59032198)

[5. Conclusion 9](#_Toc59032199)

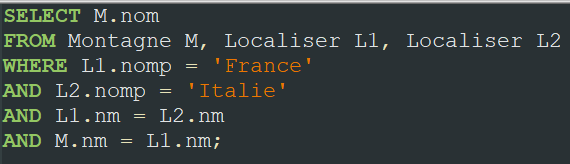
# Introduction

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.

# Auto-jointures

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

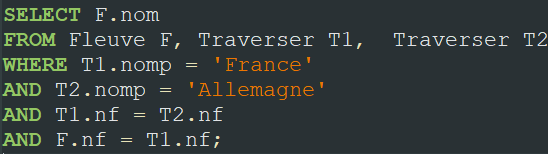
1. Nous allons commencer par obtenir les montagnes se situant en France et en Italie. Pour cela, nous allons utiliser la commande suivante :



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 :



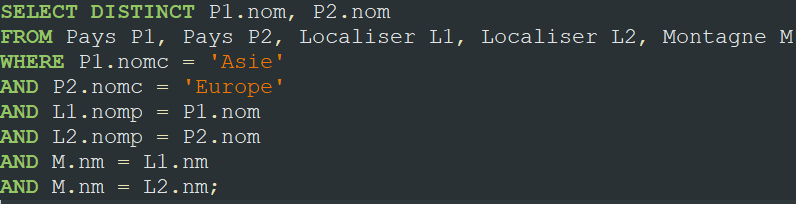
1. Nous continuons avec une requête similaire puisqu’il faut cette fois ci obtenir les fleuves qui traversent la France et L’Allemagne.



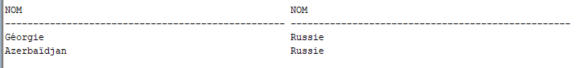
La commande est très similaire. En effet, nous avons juste changé les tables utilisées et le nom des pays. Nous obtenons :



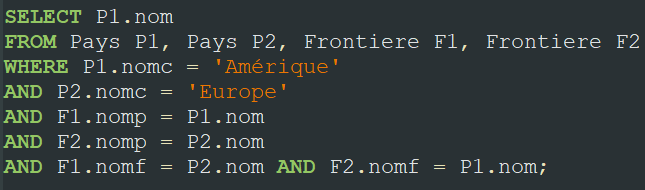
1. Nous allons compliquer encore plus avec l’utilisation de deux versions de deux tables différentes. Nous cherchons donc les pays d’Asie et d’Europe qui ont au moins une montagne commune. Nous utilisons la commande :



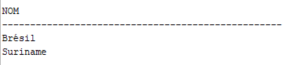
Nous voyons bien dans le FROM les deux Localiser et les deux Pays avec pour chacun des conditions de jointures. On obtient :



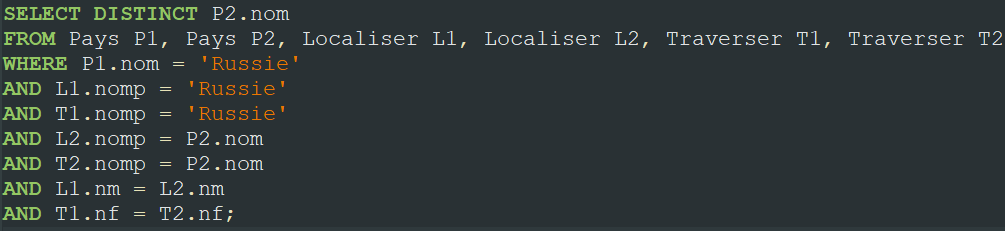
1. Une nouvelle fois une requête similaire. On cherche les pays d’Amérique et d’Europe qui ont une frontière commune. On utilise la commande :



Et nous renvoie :



1. Cette question est la plus compliqué des auto-jointures. On a besoins 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 :



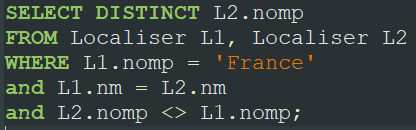
Le nombre de table ayant augmenté, le nombre de conditions augmente aussi. On obtient :



# Thêta-Jointure

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

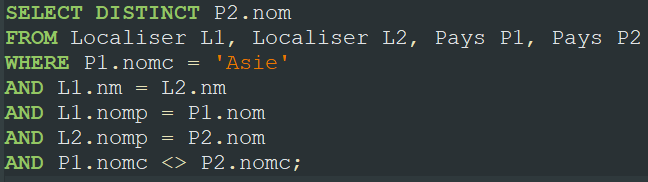
1. On commence en cherchant les pays qui ont au moins une montagne en commun avec la France en retirant la France du résultat. On utilise la commande suivante :



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



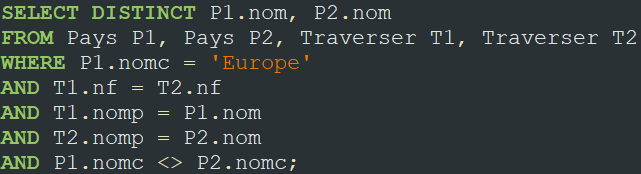
1. Cette fois-ci, on cherche les pays non Asiatiques qui ont au moins une montagne en commun avec un pays d’Asie. Cela nécessite l’utilisation de deux versions de la table Pays. On utilise la commande :



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



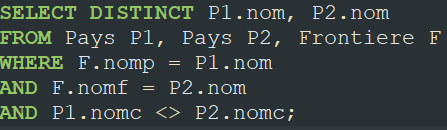
1. On veut effectuer une requête similaire mais cette fois si avec les fleuves et en affichant le nom des deux pays. On cherche donc les pays non européens qui ont un fleuve en commun avec un pays d’Europe. On utilisa la commande :



La commande est donc très similaire. On obtient :



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



Et on obtient :

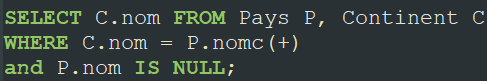


On a utilisé un SET PAGESIZE Pour modifié l’affichage.

# Jointure externe

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

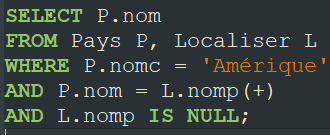
1. On commence par chercher les continents ne possédant pas de pays. On utilise la commande suivante :



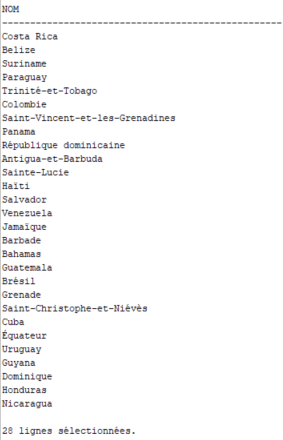
On voit donc l’apparition du (+) qui permet d’ajouter les éléments qui sont normalement ignoré lors de la jointure. Le IS NULL permet de garder seulement l’élément normalement ignoré. On obtient :



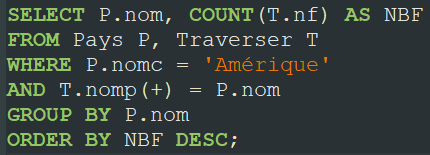
1. On cherche maintenant les pays d’Amérique qui n’ont pas de montagne. On utilise la commande suivante :



La commande est donc similaire. On obtient :



1. 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 :



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 :



# 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 <> et enfin nous avons vu les jointures externes qui permettent de récupérer des éléments normalement ignorés lors des jointures avec (+).