Morgado-Samagaio Jonathan

**IBD / TP14 Oracle**

Table des matières

[Table des matières 2](#_Toc61259935)

[1. Introduction 3](#_Toc61259936)

[2. Requêtes 3](#_Toc61259937)

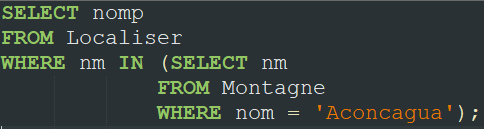
[3. Conclusion 9](#_Toc61259938)

# Introduction

Dans ce TP, nous allons continuer d’utiliser les jointures procédurale avec de nouveaux opérateur : ALL, ANY.

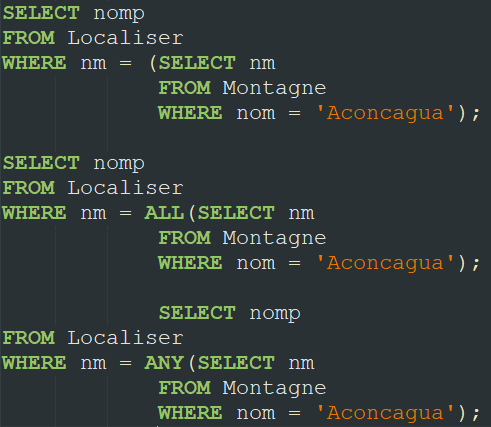
# Requêtes

1. Nous commençons facilement en cherchant le nom du pays ou est localisée la montagne « Aconcagua ». Nous commençons en faisant une version avec un IN.



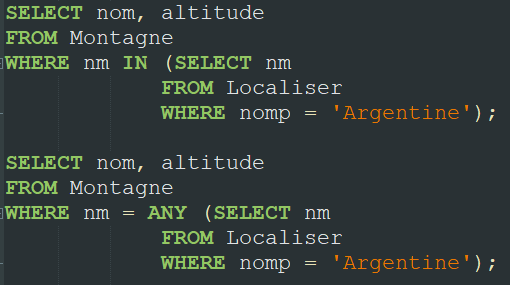
Nous pouvons voir le même type de requête qu’au TP précédent. On obtient : 

Nous allons maintenant faire cette même requête mais avec plusieurs opérateurs.

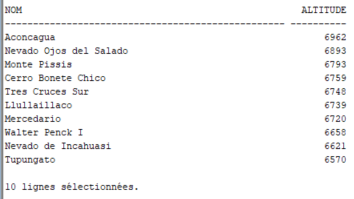


Nous pouvons voir l’utilisation du = mais aussi de deux nouveaux opérateurs : ALL et ANY. On obtient exactement le même résultat.

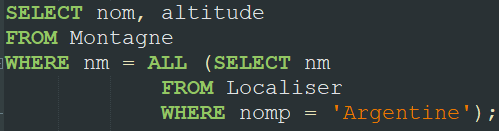
1. On continue en cherchant cette fois-ci l’altitude et le nom des montagnes localisées en Argentine. On va a nouveau faire 4 versions de cette requête.



Ces deux versions nous donnent un résultat correct qui est :

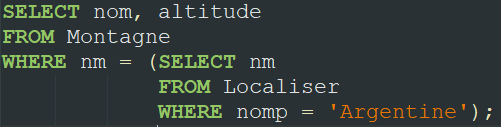


Nous pouvons donc dire que =ANY équivaut a IN. Cependant, les deux autres versions ne nous donnent pas de bon résultat.

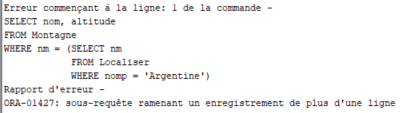


Cette version ne nous renvoie aucun résultat : 

En effet, plusieurs résultat son retourner par la requête imbriquer mais un « nm » ne peut pas être égal à plusieurs nm en même temps. Le ALL ne fonctionne donc pas.

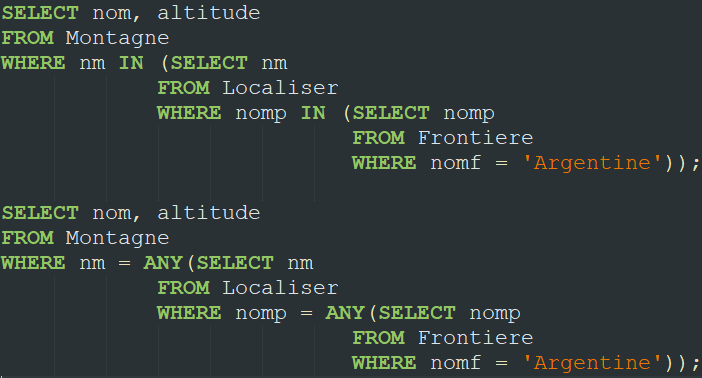


Même problème pour cette version qui nous renvoie, elle, une erreur :

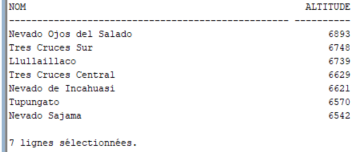


L’erreur nous dit que la sous-requête renvoie un résultat de plusieurs lignes. En effet, avec l’utilisation du =, on ne peut pas avoir plusieurs résultats.

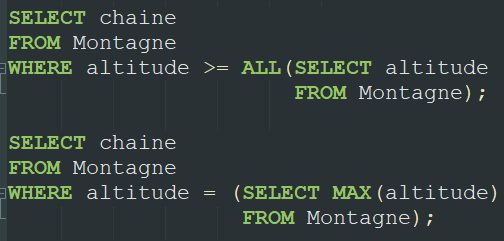
1. On complique cette fois-ci en utilisant une requête imbriqué en plus. On cherche le nom et l’altitude des montagnes localisées dans un pays frontalier de l’Argentine. On va faire deux versions avec IN et =ANY :



Le IN et =ANY étant équivalent, les deux requête s’écrivent de la même manière. On obtient :



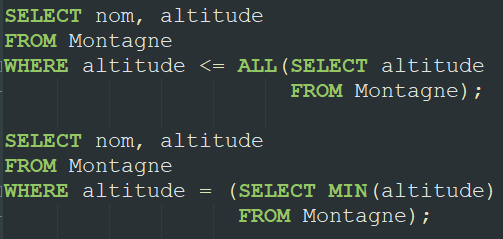
1. Nous cherchons la chaine de montagnes qui comporte la montagne la plus haute. On utilise les commandes suivantes :



Le >= ALL nécessite seulement une requête imbriqué sans conditions. Le = à besoin du MAX. On obtient :



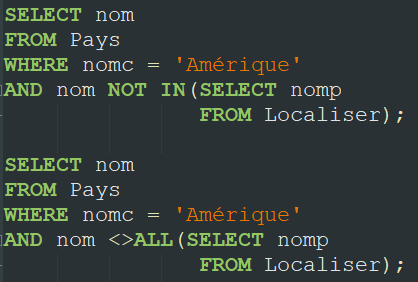
1. Nous faisons une requête similaire mais on cherche cette fois ci la montagne la moins élevée. On utilise les commandes :



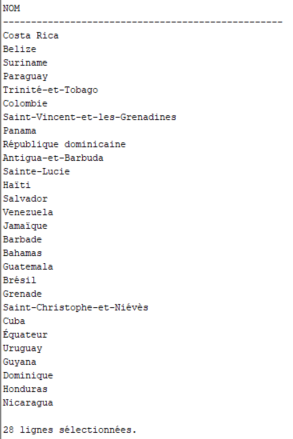
Encore une fois, les requêtes sont similaires. On obtient :



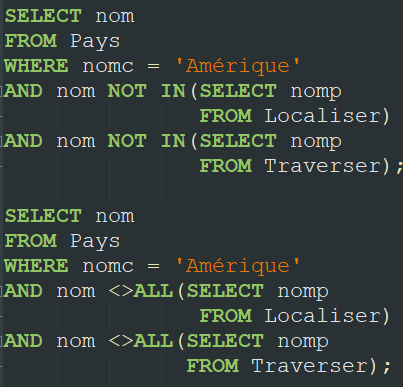
1. Nous allons utiliser d’autres opérateurs pour chercher les pays d’Amérique qui n’ont pas de montagnes. On utilise les commandes :



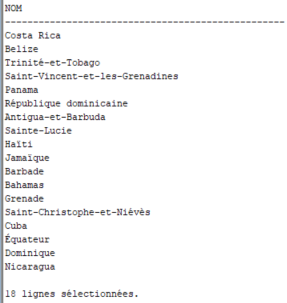
Nous pouvons voir l’apparition du NOT pour le IN et le <> avec le ALL. Ces deux opérateurs sont équivalents. On obtient :



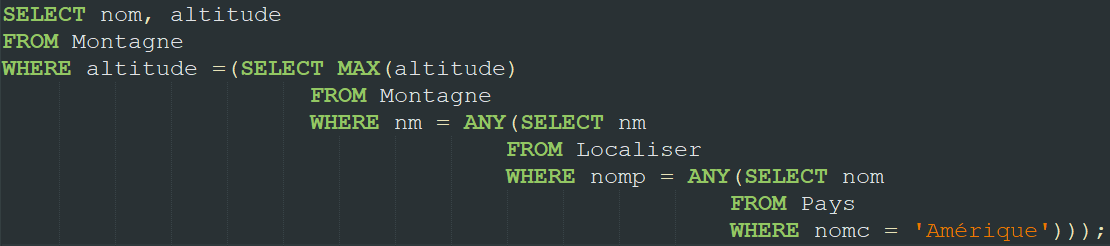
1. Nous allons faire une version améliorer de la dernière requête car nous cherchons les pays d’Amérique qui n’ont ni montagnes ni fleuves. On utilise les commandes suivantes :



La requête est donc composée de la dernière requête avec une condition en plus. On obtient :



1. Nous allons maintenant faire la requête la plus compliqué. Nous n’avons pas le droit d’utiliser IN ou NOT IN. On cherche la montagne la plus élevée sur le continent Amérique. On utilise la commande :



Nous pouvons voir qu’il y a pas mal de requête imbriquer. On obtient :



# Conclusion

Dans ce TP, nous avons vu l’utilisation de nouveaux opérateurs pour les requêtes imbriquées comme le ALL et le ANY. Nous avons vu que ces opérateurs pouvait être équivalent à d’autres et que l’on peut els combiner à d’autres.