

Morgado-Samagaio Jonathan

IBD / TP14 Oracle

Table des matières

Table des matières	2
1. Introduction.....	3
2. Requêtes.....	3
3. Conclusion	9

1. Introduction

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

2. Requêtes

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

```
SELECT nomp
FROM Localiser
WHERE nm IN (SELECT nm
             FROM Montagne
```

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

NOMP
Argentina

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

```
SELECT nomp
FROM Localiser
WHERE nm = (SELECT nm
            FROM Montagne
            WHERE nom = 'Aconcagua')

SELECT nomp
FROM Localiser
WHERE nm = ALL(SELECT nm
               FROM Montagne
               WHERE nom = 'Aconcagua')

SELECT nomp
FROM Localiser
```

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

- 2) On continue en cherchant cette fois-ci l'altitude et le nom des montagnes localisées en Argentine. On va à nouveau faire 4 versions de cette requête.

```

SELECT nom, altitude
FROM Montagne
WHERE nm IN (SELECT nm
              FROM Localiser
              WHERE nomp = 'Argen

SELECT nom, altitude
FROM Montagne
WHERE nm = ANY (SELECT nm

```

Ces deux versions nous donnent un résultat correct qui est :

```

NOM
-----
Aconcagua
Nevado Ojos del Salado
Monte Pissis
Cerro Bonete Chico
Tres Cruces Sur
Llullaillaco
Mercedario
Walter Penck I
Nevado de Incahuasi

```

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

```

SELECT nom, altitude
FROM Montagne
WHERE nm = ALL (SELECT nm
                FROM Localiser

```

Cette version ne nous renvoie aucun résultat : aucune ligne sélectionnée

En effet, plusieurs résultats sont retournés par la requête imbriquée mais un « nm » ne peut pas être égal à plusieurs nm en même temps. Le ALL ne fonctionne donc pas.

```

SELECT nom, altitude
FROM Montagne
WHERE nm = (SELECT nm
            FROM Localiser

```

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

```

Erreur commençant à la ligne: 1 de la commande -
SELECT nom, altitude
FROM Montagne
WHERE nm = (SELECT nm
            FROM Localiser
            WHERE nomp = 'Argentine')
          Dernier message -

```

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.

- 3) On complique cette fois-ci en utilisant une requête imbriquée 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 :

```

SELECT nom, altitude
FROM Montagne
WHERE nm IN (SELECT nm
            FROM Localiser
            WHERE nomp IN (SELECT nomp
                          FROM Frontiere
                          WHERE nomf = 'Ar

SELECT nom, altitude
FROM Montagne
WHERE nm = ANY(SELECT nm
              FROM Localiser

```

Le IN et =ANY étant équivalents, les deux requêtes s'écrivent de la même manière. On obtient :

```

NOM
-----
Nevado Ojos del Salado
Tres Cruces Sur
Llullaillaco
Tres Cruces Central
Nevado de Incahuasi
Tupungato
Nevado Sajama

```

- 4) Nous cherchons la chaîne de montagnes qui comporte la montagne la plus haute. On utilise les commandes suivantes :

```

SELECT chaine
FROM Montagne
WHERE altitude >= ALL(SELECT altitude
                      FROM Montagne)

SELECT chaine
FROM Montagne
WHERE altitude = (SELECT MAX(altitude)
                  FROM Montagne)

```

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

```

CHAINE
-----
Himalaya

```

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

```

SELECT nom, altitude
FROM Montagne
WHERE altitude <= ALL(SELECT altitude
                     FROM Montagne)

SELECT nom, altitude
FROM Montagne
WHERE altitude = (SELECT MIN(altitude)
                  FROM Montagne)

```

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

```

NOM
-----
Mont. Piora

```

- 6) Nous allons utiliser d'autres opérateurs pour chercher les pays d'Amérique qui n'ont pas de montagnes. On utilise les commandes :

```

SELECT nom
FROM Pays
WHERE nomc = 'Amérique'
AND nom NOT IN (SELECT nomc
                 FROM Localite)

SELECT nom
FROM Pays
WHERE nomc = 'Amérique'

```

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

```

- 7) Nous allons faire une version améliorée de la dernière requête car nous cherchons les pays d'Amérique qui n'ont ni montagnes ni fleuves. On utilise les commandes suivantes :


```

SELECT nom
FROM Pays
WHERE nomc = 'Amérique'
AND nom NOT IN (SELECT nom
                  FROM Local
                  WHERE nomc = 'Amérique')
AND nom NOT IN (SELECT nom
                  FROM Trave
                  WHERE nomc = 'Amérique')

SELECT nom
FROM Pays
WHERE nomc = 'Amérique'
AND nom <> ALL (SELECT nom

```

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

```

NOM
-----
Costa Rica
Belize
Trinité-et-Tobago
Saint-Vincent-et-les-Grenadines
Panama
République dominicaine
Antigua-et-Barbuda
Sainte-Lucie
Haïti
Jamaïque
Barbade
Bahamas
Grenade
Saint-Christophe-et-Niévès
Cuba
Équateur

```

- 8) 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 :


```

SELECT nom, altitude
FROM Montagne
WHERE altitude =(SELECT MAX(altitude)
                  FROM Montagne
                  WHERE nm = ANY(SELECT nm
                                FROM Localiser
                                WHERE nomp = ANY(SELECT nom
                                                  FROM Montagne
                                                  WHERE altitude <= altitude)))

```

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

NOM
Aconcagua

3. 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.