

Morgado-Samagaio Jonathan

**PABD / TP1 Oracle**

## Table des matières

Table des matières .....	2
1. Introduction.....	3
2. Requêtes de mise à jour synchronisée.....	3
3. Requêtes d'interrogation avec jointure procédurale synchronisé.....	5
4. Conclusion .....	10

## 1. Introduction

Dans ce TP, nous allons voir comment utiliser des jointure procédurale synchronisée donc avec EXISTS et NOT EXISTS.

## 2. Requêtes de mise à jour synchronisée

- 1) On commence par utiliser le point de reprise pour commencer avec une bonne base de données
- 2) On ajoute une nouvelle colonne dans la table Continent :

```
--1.2
ALTER TABLE Conti
ADD (NBP DECIMAL(
```

En utilisant la commande DESC Continent, on obtient :

Nom	NULL ?	Type
NOM	NOT NULL	VARCHA
SUPERFICIE		NUMBER
POPULATION		NUMBER
CROISSANCE		NUMBER

Nous pouvons voir que la colonne a bien été ajoutée.

- 3) Nous commençons les jointures procédurales synchronisées en ajoutant pour chaque continent le nombre de pays qu'ils possèdent :

```
--1.3
UPDATE Continent C
SET C.NBP = (SELECT COUNT(*)
FROM Pays P
```

Nous voyons donc un UPDATE avec un SELECT imbriqué.

- 4) Si on affiche la table Continent, on obtient :

```
--1.4
SELECT * FROM Conti
```

NOM	SUPERFICIE	POPULATION	CROISSAI
Asie	44579000	4601371000	,97
Afrique	30065000	1308064000	2,
Amérique	42189120	1014722000	,79
Antarctique	13209000	0	
Europe	9938000	747193000	,14
Océanie	7687000	42128000	:

- 5) On ajoute trois nouvelles colonnes dans la table Pays :

```
--1.5
ALTER TABLE Pays
ADD (NBP DECIMAL(5,2)
     NBF DECIMAL(5,2)
     NBM DECIMAL(5,2))
```

- 6) Nous voulons maintenant mettre à jour ces nouvelles colonnes. On utilise la commande suivante :

```
--1.6
UPDATE Pays P
SET P.NBP = (SELECT COUNT(*)
             FROM Frontiere
             WHERE F.nomp = P.nomp)
     P.NBF = (SELECT COUNT(*)
             FROM Traverser
             WHERE T.nomp = P.nomp)
     P.NBM = (SELECT COUNT(*)
             FROM MONTAGNE
             WHERE M.nomp = P.nomp)
```

Nous pouvons voir une requête imbriquée pour chaque modification de colonne.

7) On obtient :

NOM	CAPITALE	NBF
Antigua-et-Barbuda	Saint John's	0
Argentine	Buenos Aires	5
Bahamas	Nassau	0
Barbade	Bridgetown	0
Belize	Belmopan	2
Bolivie	La Paz	5
Brésil	Brasilia	10
Canada	Ottawa	1
Chili	Santiago	3
Colombie	Bogota	5
Costa Rica	San José	2
Cuba	La Havane	0
Dominique	Roseau	0
Équateur	Quito	2
États-Unis	Washington D.C.	2
Grenade	Saint-Georges	0
Guatemala	Guatemala	4
Guyana	Georgetown	3
Haïti	Port-au-Prince	1
Honduras	Tegucigalpa	3
Jamaïque	Kingston	0
Mexique	Mexico	3
Nicaragua	Managua	2
Panama	Panama	2
Paraguay	Asuncion	3
Pérou	Lima	5
République dominicaine	Saint-Domingue	1
Saint-Christophe-et-Niévès	Basseterre	0
Sainte-Lucie	Castries	0
Saint-Vincent-et-les-Grenadines	Kingstown	0

(Limitation aux pays d'Amérique.

### 3. Requêtes d'interrogation avec jointure procédurale synchronisé

1) On cherche maintenant les fleuves qui traversent l'Inde. Nous allons commencer par une version de la requête que nous avons déjà pu avoir avant :

```
--2.1.a
SELECT *
FROM Fleuve
WHERE nf IN (SELECT nf
              FROM Traverser
```

Nous pouvons voir la requête imbriquée avec IN.

On obtient :

NF	NOM	LONGUEUR
81	Le Brahmapoutre	2896
83	Le Cauvery	950
89	Le Gange	2510
90	La Godavari	1500
96	L'Indus	3180
104	La Krishnâ	1290
107	Le Mahânadi	851
111	La Narmadâ	1289
116	Le Pennar	597
124	Le Tapti	724

Nous allons maintenant faire la même requête mais cette fois ci avec une jointure procédurale synchronisée :

```
--2.1.b
SELECT *
FROM Fleuve F
WHERE EXISTS (SELECT T.nf
               FROM Traverser
               WHERE T.nomp =
```

Nous obtenons donc exactement le même résultat.

- 2) Nous cherchons maintenant les pays frontaliers de l'Inde avec les deux même type de requêtes. La première :

```
SELECT nom, capitale
FROM Pays
WHERE nom IN (SELECT nomf FROM Frontiere WHERE nomf = 'Inde')
```

Et la seconde :

```
--2.2.b
SELECT nom, capitale
FROM Pays P
WHERE EXISTS (SELECT F.nomf
               FROM Frontiere
               WHERE F.nomp =
```

Nous obtenons avec les deux requêtes :

NOM	Cl
Bangladesh	Da
Pakistan	Is
Népal	Ka
Birmanie	Na
Chine	Pe
Bhoutan	Th

- 3) On cherche maintenant les fleuves qui traversent l'Inde et la Chine. On utilise la commande suivante pour la première requête :

```
--2.3.a
SELECT *
FROM Fleuve
WHERE nf IN (SELECT nf
              FROM Traverser
              WHERE nomp = 'Inde'
              AND nf IN (SELECT nf
                        FROM Traverser
                        WHERE nomp = 'Chine'))
```

Pour la seconde requête :

```
--2.3.b
SELECT *
FROM Fleuve F
WHERE EXISTS (SELECT nf
              FROM Traverser
              WHERE T.nomp = 'Inde'
              AND T.nf = F.nf)
           INTERSECT
           (SELECT nf
            FROM Traverser
            WHERE nomp = 'Chine'
            AND nf IN (SELECT nf
                      FROM Fleuve
                      WHERE EXISTS (SELECT nf
                                FROM Traverser
                                WHERE T.nomp = 'Inde'
                                AND T.nf = F.nf)))
```



On obtient donc :

NF	NOM	LONGUEUR	EMBOUCHURE
81	Le Brahmapoutre	2896	golfe du Be
96	L'Indus	3180	mer d'Arab

- 4) On cherche maintenant les pays traversé par les fleuves qui traversent l'Inde. On utilise cette commande pour la première requête :

```
--2.4.a
SELECT nom, capitale
FROM Pays
WHERE nom IN (SELECT nomp
               FROM Traverser
               WHERE nf IN (SELECT nf
                           FROM Traverser
                           WHERE nomp = 'Inde'))
```

Et celle la pour la seconde :

```
--2.4.b
SELECT P.nom, P.capitale
FROM Pays P
WHERE EXISTS (SELECT T.nf
              FROM Traverser
              WHERE T.nomp = 'Inde'
              INTERSECT
              SELECT T.nf
              FROM Traverser
              WHERE T.nomp = 'Inde'))
```

On obtient donc :



NOM

---

Bangladesh

Pakistan

Chine

- 5) On cherche le continent qui comporte le plus de pays de deux manières différentes. On utilise cette commande pour la première requête :

```
--2.5.a
SELECT nom
FROM Continent
WHERE NPD = (SELECT MAX(NPD) FROM Co
```

Et celle-ci pour la seconde :

```
--2.5.b
SELECT nomc
FROM Pays
GROUP BY nomc
HAVING COUNT(nom) = (SELECT MAX(CO
FROM Pays
```

Nous avons déjà effectué cette requête dans un ancien TP. On obtient :

NOMC

---

Afrique

- 6) Pour finir, on cherche le pays de chaque continent comportant le plus de montagne de deux manières différentes. La première requête :

```
--2.6.a
SELECT nomc, nom
FROM Pays P1
WHERE NBM = (SELECT MAX(NBM)
FROM Pays P2
```

La seconde :

```
--2.6.b
SELECT nomc, nom
FROM Pays P
WHERE nom IN (SELECT nomp
              FROM Localiser
              GROUP BY nomp
              HAVING COUNT(nomp) = (SELECT MAX(COUNT(nomp)
              FROM Localiser
              WHERE nomp IN (SELECT
              FROM E
```

On obtient :

NOMC	NOM
Afrique	Éthiopie
Asie	Indonésie
Amérique	Canada
Océanie	Papouasie-Nouvel

#### 4. Conclusion

Dans ce TP, nous avons pu voir comment utiliser les jointures procédurales synchronisées en utilisant les EXISTS avec des INTERSECT à l'intérieur.