

TP n°2

Requêtes réparties

L'objectif est de comprendre l'évaluation d'une requête de jointure entre 2 relations qui sont situées sur 2 sites distincts.

- Définir le schéma global qui offre un accès transparent à des données de plusieurs bases,
- Formuler une requête répartie.
- Comprendre l'ordre et l'emplacement des opérations permettant d'évaluer une requête répartie (quel site traite quelles opérations?).

Scénario

Deux SGBD hébergés sur des sites différents

- Le site 1 s'appelle base1:

vérifier que la connexion est correcte et afficher le nom du SGBD :

```
SELECT sys_context('USERENV', 'INSTANCE_NAME') FROM dual;
```

- Le site 2 s'appelle base2:

De même vérifier que la connexion est correcte et afficher le nom du SGBD

Données

- Le site 1 contient les hôpitaux dans la table Hopital(numHopital, nomHopital, adresse)
- Le site 2 contient les Services dans la table Service(numService, nomService, tel, nbLits, numHopital)

La couche BDR est implémentée sur le site 1

Relier les sites : La couche BDR doit pouvoir se connecter au site 2 en utilisant le lien appelé site2

Vérifier le bon fonctionnement du lien

```
CONNECT ... @site1  
DESC Service@site2
```

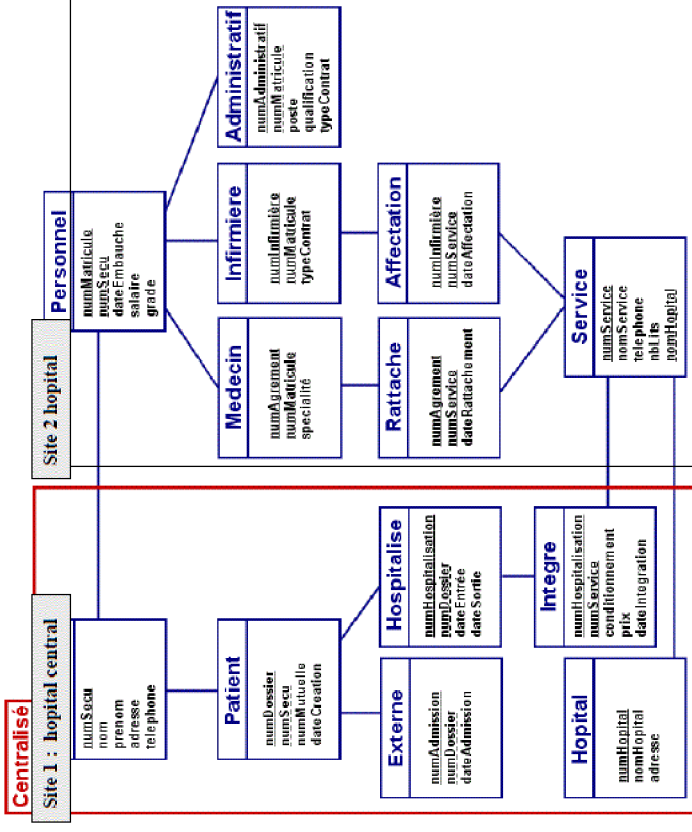


Figure 1 : Modèle conceptuel de la BDD hôpitaux

2) Construire le schéma global

```
CONNECT ... @site1  
CREATE VIEW Service AS  
SELECT *  
FROM Service@site2;
```

3) Requêtes réparties

Pour chaque requête, répondre aux questions

- Où est traitée chaque opération (sélection, projection, jointure, ...)?
- Quelles sont les données transférées entre les sites pendant l'évaluation de la requête ?

Activer le mode de visualisation des plans et le chronométrage

set timing on

set autotrace trace explain stat

R1 : Jointure seule avec un transfert volumineux

Afficher les Services avec leur Hôpital

```
CONNECT ... @base1

SELECT s.prenom, s.descriptif, h.division
FROM Service s, Hôpital h
WHERE s.numHôpital = h.numHôpital;
```

R2 : Jointure avec sélection

```
SELECT s.prenom, s.descriptif, h.division
FROM Service s, Hôpital h
WHERE s.numHôpital = h.numHôpital
AND s.nbLits > 20;
```

La sélection est-elle poussée sur le site 2 ?

R3 : Jointure très sélective

Ajouter un hôpital dans une nouvelle ville, cet hôpital n'aura que 10 services, ce qui permettra, par la suite, de poser une requête de jointure très sélective.

insert into hôpital values(100, "nouveau hôpital", 'adresse100', 'ville100');

- R3a : Jointure très sélective et avec un transfert volumineux

```
SELECT s.prenom, s.descriptif, h.division
FROM Service s, Hôpital h
WHERE s.numHôpital = h.numHôpital
AND h.ville = 'ville100';
```

- R3b : jointure très sélective et avec un transfert faible.

La directive `driving_site` prend en argument le nom de la variable associée à la table Service stockée sur le site dans lequel Oracle doit traiter la jointure. Autrement dit, Oracle doit traiter la jointure sur le site 2 contenant Service.

```
SELECT /*+ driving_site(s) */ s.prenom, s.descriptif, h.division
FROM Service s, Hôpital h
WHERE s.numHôpital = h.numHôpital
AND h.ville = 'ville100';
```

Note : pour obtenir des résultats significatifs il faudrait peupler les bases de données avec un grand nombre de services. Par exemple : 2000 services et 50 hôpitaux.

R4 : jointure et sélection avec index

Dans le site 2, créer un index sur `Service(numHôpital)`. Afficher le plan de R2 (sélection sur le nbLits). Vérifier que l'index `Service(numHôpital)` est utilisé. Expliquer le plan.

R5: Proposer une requête avec 2 jointures entre 3 tables

La première jointure traitée le site 2, la 2ème jointure sur le site 1.

R6: Proposer une requête avec 1 jointure entre deux tables traitée par semi-jointure

- transférer les valeurs d'un attribut du site 1 vers le site 2. Ne pas transférer les attributs
- transférer les n-uplets satisfaisant la requête du site 2 vers le site 1
- finir le calcul de la requête sur le site 1

4) Durée des transferts

Chronométrer les transferts en répétant n fois (n=10) l'exécution d'une requête afin d'obtenir des durées significatives mesurables.

Pour mesurer principalement les transferts de données entre les sites et non la durée d'affichage du résultat, on modifie légèrement les requêtes : le résultat d'une requête est agrégé (par exemple avec un max). Ainsi, le résultat d'une requête sera "petit", il aura un seul nuplet. En conséquence, le temps d'affichage devient négligeable face à la durée des transferts de données entre les sites. Le temps total que l'on chronomètre (avec `set timing on`) correspond principalement aux transferts de données.

Comparer les deux exécutions suivantes a) et b). Laquelle est la plus rapide ? Est-elle beaucoup plus rapide ?

- a) Avec transfert de tous les Services

```
SET timing ON
DECLARE
res NUMBER;
BEGIN
FOR i IN 1 .. 10 LOOP
SELECT MAX(LENGTH(s.descriptif))
INTO res
FROM Service s, Hôpital h
WHERE s.numHôpital = h.numHôpital
AND h.ville = 'ville100';
END LOOP;
END;
/
SET timing off
```

- b) Avec transfert des Hopitals de la ville100

```

SET timing ON
DECLARE
res NUMBER;
BEGIN
FOR i IN 1 .. 10 LOOP
    SELECT /*+ driving_site(s) */ MAX(LENGTH(s.descriptif))
    INTO res
    FROM Service s, Hopital h
    WHERE s.numHopital = h.numHopital
    AND h.ville = 'ville100';
END LOOP;
END;
/
SET timing off

```

5) Fragmentation

- Fragmenter les Hopital par division : placer les Hopitals de la division U sur le site 1 (table Hopital1) et ceux de la division H sur le site 2 (table Hopital2).
 - Définir la vue Hopital réunissant tous les Hopitals des deux divisions.
 - Est-ce que les 2 fragments sont accédés pour une requête affichant seulement les Hopitals de 1ère division ? Pourquoi ?
- ```

SELECT *
FROM Hopital h
WHERE h.division=1

```