Université des Sciences et de La Technologie d'Oran - Mohammed Boudiaf
Faculté des Mathématiques et Informatique
Département d'Informatique
Master I Bases de données avancées

Mini Projet "Base de données répartie"

Implémentation et gestion
d'une base de données répartie Oracle (Approche descendante)

1 Introduction et objectifs

Concevoir et implémenter une base de données répartie sur trois sites (Site1, Site2, Site3) selon une approche descendante, dans le but de :

- Comprendre l'architecture d'une base de données distribuée et ses enjeux.
- Maîtriser la modélisation orientée objet et l'implémentation en environnement distribué.
- Expérimenter la distribution top-down, la fragmentation, les requêtes distribuées et la reconstitution.
- liens entre sites, synonymes, triggers, sécurité,
- Automatiser la gestion avec un scheduler.
- Optimiser le schéma et les requêtes en contexte réel.

Étape	Thème	Objectif clé
1	Schéma global et modélisation orientée objet	Conception globale de la base
2	Fragmentation et distribution top-down	Répartition logique des données
3	Reconstitution / transparence de fragmentation	Vérification des propriétés de reconstruction
4	Création des utilisateurs et privilèges locaux	Sécurisation et accès
5	Création des liens inter-sites ('DBLINK')	Communication entre bases
6	Synonymes	Transparence d'accès
7	Triggers 'INSTEAD OF'	Transparence de mise à jour
8	Sécurité globale et rôles	Contrôle d'accès distribué
9	Automatisation et tâches planifiées	Gestion du système réparti
10	Optimisation et évaluation	Performance et coût réseau

2 Répartition dans Oracle

Oracle Net Manager La configuration des services oracle net facilite les complexités de configuration et de gestion du réseau, optimise les performances et améliore la sécurité du réseau et les capacités de diagnostic. Il fournit des solutions de connectivité dans des environnement distribués. Oracle fournit un assistant de configuration qui est composé de:

- **Processus d'écoute:** *Permet de créer, et de configurer des processus d'écoute chargé de recevoir les connexions clients.*
- **Méthode de résolution de nom**: *Permet de configurer les différents manières de convertir des noms simples en descripteurs de connexions.*
- **Nom de service réseau local**:*Permet de définir des noms simples pour identifier l'emplacement d'un service, tel qu'une base de données. ces noms simples sont associés aux descripteurs de connexion contenant l'emplacement réseau et l'identificateur du service.*
- Utilisation d'un annuaire.

Le programme correspondant "**Assistant Configuration Oracle Net**" exécuter en mode **administrateur**.
On peut aussi le lancer à partir de l'invité de commande en tant qu'administrateur et exécuter la commande "**netca**".

3 Accéder à différents utilisateurs de différents sites

On peut se connecter aux différentes bases de données des différents sites.

CONN[ECT] <*user_name*>[/<*user_password*>][@ <*identificateur_connexion*>] | /[AS SYSDBA | SYSOPER]

où identificateur_connexion: @Ip de la machine:Port/Service
Exemple: SQL> connect system/system@192.168.52.120:1521/rando

4 Les services Réseau

Nom service réseau L'attribution d'un *nom service réseau* permettra une écriture plus simplifier de l'identificateur de connexion.
Passer par *l'Assistant configuration oracle net* et utiliser l'option *Configuration d'un nom de service réseau locale*.

Nom service Indiquer le nom du service de la base de donnée ou du service auquel vous voulez accéder.

Protocole Sélectionner le protocole associé à la base de données (TCP)

nom d'hôte Indiquer le nom d'hôte de l'ordinateur de la base de données, dans notre cas on précise l'@IP de la machine.

Port On doit aussi préciser le n° du port TCP/IP.

Test Effectuer le test de connexion. Changer de connexion pour modifier le nom d'utilisateur.

On peut aussi le faire sans l'assistant:
'Oracle Net Manager' -> 'Oracle Net Configuration' -> 'Local' -> 'Résolution de noms de service' -> Cliquer sur le bouton de création '+' et remplir les champs.

5 Création d'un lien entre les bases de données

Pour pouvoir nommer librement les database links, exécuter à chaque fois que vous lancez SQL*Plus Worksheet :
ALTER system SET global_names = false ;

5.1 Créer un lien

Le **DB_link** ou database link est un objet d'une base de données permettant d'exécuter des requêtes sur une autre base de données, qu'elle se trouve physiquement sur la même machine ou qu'elle soit distante. Cela va notamment permettre, par exemple d'effectuer des jointures entres des tables qui ne sont pas sur la même base.
La création s'effectue via la requête suivante :

```
CREATE [PUBLIC] DATABASE LINK <nom_database_link>
        CONNECT TO <oracle_user> IDENTIFIED BY "<password_user>"
        USING '<SID>' ;
```

L'option PUBLIC peut être utilisée pour préciser que le DB_link peut être utilisé (accessible) par tous les utilisateurs.

SID Spécifie le nom du service de la base de donnée distante

5.2 Répertorier un lien

Tables: User_db_Links, All_DB_Links, DBA_DB_Links
Commande pour répertorier toutes les liaisons de base de données
SELECT * FROM *All_DB_Links*
SELECT * FROM *V$DBLink*

Si le lien ne fonctionne pas, vérifiez la connexion entre les 2 bases de données (ping des machines, tnsping, etc...)

5.3 Utiliser le lien

Pour utiliser notre DB_link, il suffira de rajouter le paramètre ***database_link*** après le nom de la table à interroger, par exemple : SELECT * FROM *nom_table* @*nom_lien*

5.4 Supprimer un lien

Commande pour supprimer une liaison de base de données:

DROP [PUBLIC] DATABASE LINK <nom_database_link>;

6 Création de tables à partir d'autres tables

Il est possible de créer des tables en sélectionnant les données d'autres tables comme suit:

```
CREATE TABLE nom_table AS ( SELECT AT t1 , . . . AT tn FROM t1 , . . . Tl WHERE cond1 , . . . condm)
```

7 Les synonymes

Un synonyme est un alias ou un nom convivial pour les objets de base de données (tels que les tables, les vues, les procédures stockées, les fonctions et les packages). Les synonymes ont été mis en place afin de masquer le nom des liens vers les bases de données distantes. Ainsi, il est possible d'écrire des requêtes de manière transparente vis-à-vis de la localisation.

7.1 Créer un synonyme

```
CREATE OR REPLACE [PUBLIC ] SYNONYM nom_synonyme FOR nom_table@nom_lien ;
```

7.2 Supprimer un synonyme

```
DROP SYNONYM nom_synonyme ;
```

8 TRIGGERS INSTEAD OF

Afin de permettre à un utilisateur de mettre à jour (insertion, suppression, modification) indépendamment des fragments, il faut utiliser les **triggers INSTEAD OF**. Ces derniers prennent la main et font les mises à jour sur les fragments distants.

```
CREATE TRIGGER nom_trigger
INSTEAD OFF INSERT | UPDATE | DELETE ON nom_vue
FOR EACH ROW
BEGIN
. . .
END;
```

9 Préparation machine - Utilisation de Oracle net Manager

1. Reconfigurer en utilisant *Oracle net Manager* le Listner de chaque site.
2. Vérifier les connexions entre les sites.
3. Configurer les noms de service réseau: Attribuer des noms de service pour les différents sites. De préférence et dans le but de simplifier l'utilisation, maintenir le nom du SID local= nom SID réseau.
    * SID réseau des machines vers site1/orcl (pour la création des tables)
4. Jeter un coup d'œil sur les fichiers de configuration réseau de chaque site *tnsnames.ora* et *listener.ora* Que remarquez vous?
   Les fichiers se trouvent dans \oracle\product\ 10.2.0\db_1\network\ADMIN.
5. Tester les noms que vous avez utiliser.

Pour une meilleure gestion et une facilité de manipulation, je vous suggère d'utiliser **Oracle SQL Developer** sur votre machine physique.

10 Conception d'une base de données répartie

La modélisation d'une base de données répartie n'a pas de formule magique pour trouver la solution optimale. l'administrateur prendra une décision de modélisation en fonction de son étude de l'existant(critères technique et organisationnelles) avec pour objectif de minimiser le nombre de transferts entre sites, les temps de transfert, le volume de données transférées, les temps moyens de traitement des requêtes, le nombre de copies de fragments, etc...
Deux types de conceptions connus: top down et bottom up design. On a choisi pour notre TP la conception **Top down design** : Intéressante dans notre cas puisqu'on part du néant.

Atelier Pratique

11 Schéma,vue Étudiant

Le schéma globale Étudiant : Soit la base de données relationnelle Étudiant modélisant l'organisation des facultés et l'administration d'une université. ***Créer la base de donner Étudiant sur la machine physique.***

- Etudiant(**EtID**, EtNom, EtPrenom, EtDatenais, EtVillenais, EtPrenom_Pere, EtMere, EtAdr, EtTel, Etemail, ETNss, EtDateInsc, EtBac, EtStatut, EtBourse ♯ DepID) *// EtBac est l'année du Bac et ETNss est le numéro de sécurité social, EtStatut 'Actif', 'Diplômé', 'Abandonné'.*

- Departement(**DepID**, DepDesig, EnsID) *// EnsID est l'identificateur du chef du département, un enseignant désigné par son numéro.*

- Enseignant(**EnsID**, EnsNom, EnsPrenom, EnsDatenais, Ensadr, Enstel, EnsMail, EnsNss, EnsEtatCivil, EnsDaterect, EnsSpec, EnsTitre ♯DepID) *// EnsSpec est la spécialité de l'enseignant*

- Salle(**SID, ♯ DepID**, SType, Snom, SNbPlaces, Setage, Sbloc) *// Il peut y avoir des salles avec le même identifiant dans des départements différents d'une même faculté. SType est le type de la salle (Emphi, Classe, Salle de TP)* .

- Seance(**♯EnsID, ♯EtID, ScType, ScJour, ScCreneau**, Descrip,♯SID, ♯ DepID)//ScType Type de la séance(Cour, Td où Tp)

12 Modélisation des données (orienté objet)

- Dessiner le diagramme UML des entités (Etudiant, Enseignant, etc.), relations, héritages éventuels;
- Identifier les entités principales: Etudiant, Enseignant, Matiere, ...
- Définir les relations et héritages (par ex. Personne → Etudiant, Enseignant);
- Implémenter les types objets Oracle;
- Créer les tables objet;

Question de réflexion: Quels avantages offre la modélisation orientée objet dans un contexte distribué ?

12.1 Méthodes à developper

Les bases de données orientées objet (OO), il est essentiel d'indiquer non seulement la structure des classes/types, mais aussi les méthodes à développer pour pratiquer réellement la modélisation objet.
Méthodes à développer pour chaque classe (type objet) qq exemples:

1. Pour la classe/type Etudiant
    - inscrire() : permet l'inscription à un cours/séance.
    - changerNiveau() : permet d'évoluer dans le cursus.
    - afficherInfos() : retourne toutes les infos de l'étudiant.
    - payerFrais() (optionnel) : simule le paiement des frais.
2. Pour la classe/type Enseignant
    - ajouterSeance() : ajouter une nouvelle séance au planning.
    - modifierGrade() : changer le grade (assist./prof).
    - afficherPlanning() : liste des séances attribuées.
3. Pour la classe/type Salle

- afficherOccupation() : indique les plages horaires occupées.
- ajouterReservation() : réserve la salle pour une séance.

4. Pour la classe/type Seance
   - changerHoraire() : modifie l'horaire de la séance.
   - ajouterEtudiant() : ajoute un étudiant à la séance.
   - afficherParticipants() : liste les inscrits à la séance.
5. Pour la classe/type Département
   - afficherMembres() : liste les enseignants/étudiants du département.
   - ajouterMembre() : ajoute un membre au département.

13 Schéma global et distribution top-down

- Définir le schéma global sur le site 1 (base de référence).
- Concevoir différents scénarios de fragmentation (horizontale, verticale ou hybride) pour répartir les données sur les sites.

1. Proposer une décomposition de la base sur ces trois sites en respectant les règles de la fragmentation (la complétude et la reconstitution). Proposer suivant votre compréhension de l'existant, une répartition des occurrences (fragmentation horizontale), ou une répartition des attributs (fragmentation verticale) où bien une répartition des valeurs (hybride) ainsi que la réplication des données, en se basant sur les hypothèses suivantes:
   - on dispose de trois sites:
     (a) premier site: Département d'informatique sur VM
     (b) deuxième site: Département de Mathématique sur VM
     (c) troisième site: l'administration de l'université(tour d'USTO) sur la machine physique.
   - Les sites Informatique et mathématique ne gèrent que les départements correspondants.
   - Les enseignants peuvent travailler dans plusieurs départements.
   - La gestion des informations d'état civil des personnes (enseignant, étudiant) (nom, adresse, téléphone,...) au site 'administration' (physique).
   - Les administrateurs des cellules informatiques de chaque département de l'université se trouve dans le site global (Administration).
   - La gestion des salles est locale à chaque département.
   - La gestion de l'emploie du temps est locale à chaque département.
2. La répartition d'une base de donnée intervient dans les trois niveaux de son architecture en plus de la répartition physique des données.
   a- niveau interne
   b- niveau conceptuel
   c- niveau externe: Proposer les vues distribuées sur les différents sites des utilisateurs.
3. l'administrateur doit créer les schémas pour les différentes bases de données des 03 sites. Écrire les scripts de création et les exécuter.
4. Prendre en compte les contraintes d'intégrité.
5. proposer des views (avec mises à jour) pour chaque site, pour avoir des informations générales de la base de données globales distante.
6. Rafraîchissement des vues matérialisées.
7. Indiquer comment se calcule chaque relation de la base globale à partir de ses fragments.
8. Justifier chaque choix de fragmentation/répartition.
9. Proposer un plan d'exécution réparti pour la requête SQL vue en Question 1, sachant maintenant que les données sont réparties sur les trois sites selon la décomposition proposée à la Question 2.

14 Reconstitution et transparence de fragmentation

S'assurer que la base globale peut être recomposée sans perte.
- Créer une vue globale sur le site principal;
- Vérifier les propriétés : Complétude, Disjonction, Reconstruction;
- Faire un test exp SELECT COUNT(*) FROM Etudiant_Global;

15 Les utilisateurs

- Créer les administrateurs locaux et gérer les accès.
- Créer un utilisateur pour chaque site.

- Créer un utilisateur avec le privilège de droit de lecture dans les départements pour les bases de données crées. l'utilisateur aura comme nom **AgentInfo** (*Département Informatique*), **AgentMaths** (*Département Mathématique*),
- créer l'utilisateur **Agent** (*Administration de la tour*) avec le privilège Mise à jour Pour les tables Etudiant et Enseignant
- Pour éviter les conflits et les oublies donner comme mot de passe le nom de l'utilisateur.

16 Création des liens inter-sites (DBLINK), Synonymes, Triggers

Permettre la communication entre les bases Oracle.

1. Créer les liens pour vos bases de données.
    * Créer les liens des différents sites pour le schéma globale
    * Interroger la vue 'User_DB_Links' pour sélectionner les propriétés des liens.

- Masquer la complexité des liens en utilisant les synonymes;
- Permettre les mises à jour sur la vue globale en utilisant les Triggers INSTEAD OF;

17 Sécurité et gestion des accès

Centraliser la gestion des privilèges inter-sites.

- Mise en place de privilèges, rôles et contraintes d'accès distinctes sur chaque site.
- Simuler l'accès concurrent et le contrôle des droits sur des fragments répartis.

18 Automatisation et scheduler

Programmation d'un scheduler afin d'automatiser les sauvegardes, la synchronisation des fragments ou le rafraîchissement des vues matérialisées.
chaque fin de journée, notre administrateur devra faire les opérations de sauvegarde des bases de données des différents sites.
a- proposer un script, qui s'exécute, tous les jours de travail à 17h, pour l'opération de sauvegarde.
b- proposer un script pour l'administrateur , pour la reconstitution (recomposition) des bases de données.
Mise en place sous Oracle (DBMS_SCHEDULER) ou avec des scripts.

19 Axe optimisation

Évaluer et améliorer la performance.
en combinant SET TIMING ON, EXPLAIN PLAN, un monitoring réseau (ping/outil), et une synthèse par tableau, tu obtiens une analyse complète du coût/réponse réseau.

19.1 Mesurer et comparer les temps de réponse

1. Comparer le temps de réponse des requêtes locales vs distribuées en chronométrant les requêtes.
   - Active le chronomètre en lançant **SET TIMING ON** avant ta requête.
   - **Comparer sur différents sites :**
   - Lance la même requête en local puis via un DB Link (accès distant).
   - Note les temps de réponse dans les deux cas.

2. Analyser le coût réseau en utilisant **EXPLAIN PLAN** et observe la colonne BYTES, IO COST dans le plan généré. (En option) Utilise un outil externe (Wireshark, tcpdump) sur les VM pour mesurer la quantité réelle de données envoyées/recues pendant la requête. Note le volume échangé en Mo/Ko pour chaque test.

3. Mesurer la latence réseau
   - Test ping.
   Si le temps de la même requête locale et distante diffère fortement, la différence provient majoritairement de la latence réseau et du volume de données transféré.
   La latence réseau s'observe sur de petites requêtes (SELECT COUNT(*) sans transfert volumineux).

4. Synthèse comparative
   Pour chaque cas d'usage, consigne dans un tableau :

test	temps de réponse	volume réseau	latence réseau
Requête locale Site 1 Requête sur Site 2(DBL)	•	•	•

- Indexation et performances:
Créer des index sur les attributs recherchés fréquemment (matricule, nom, code_dept).
Vérifier l'impact via des benchmarks ([temps de scan sans index, avec index]).
- Optimiser les requêtes par création d'index, réorganisation de la fragmentation ou usage de vues matérialisées localement.
- Proposer un cahier des charges d'optimisation : indexation, paramétrage mémoire, choix des stratégies de refresh/synchro, choix du plan de join pour requêtes distribuées.

20 Questions de synthèse

- Quels sont les avantages et les limites de l'approche descendante ?
- Comment garantir la cohérence entre fragments ?
- Quelle stratégie d'optimisation est la plus efficace dans un contexte distribué ?

20.1 Rendu attendu

- Scripts commentés (création types, tables, DB Links, jobs scheduler. . .)
- Justification de chaque choix (modèle, plan de distribution, méthode d'optimisation. . .)
- Captures écran/rapports démontrant fonctionnement et optimisation.
- Tableau de synthèse/retour d'expérience (relationnel vs objet, fragmentation, optimisation).