

TD 4 : Bases de Données Distribuées

Exercice 1 — Choix du type de fragmentation

On vous fournit une base centralisée contenant les tables :

- **CLIENT(id, nom, pays, catégorie, solde)**
- **COMMANDE(id, id_client, date, montant, pays_livraison)**
- **PRODUIT(id, nom, prix, catégorie)**

Une entreprise ouvre 3 centres de distribution (France–Algérie–Canada), chacun ne livrant que sa région.

1. Quel type de fragmentation (horizontale, verticale, hybride) est le plus adapté pour chaque table ? Pourquoi ?
2. Donner un schéma de fragmentation pour COMMANDE prenant en compte la localisation des clients.
3. Expliquer comment garantir la cohérence si un client change de pays.
4. Proposer un mode d'allocation des fragments sur les 3 sites.

Exercice 2 — Fragmentation ou réplication ?

Une application e-commerce possède :

- 10 millions d'utilisateurs,
- 200 000 produits,
- 5 entrepôts dans différents pays.

Les **recherches de produits** sont globales (tous les produits), mais la **mise à jour des stocks** est locale à chaque entrepôt.

1. Quelle stratégie est optimale :
 - fragmentation ?
 - réplication complète ?
 - réplication partielle ?
 - combinaison ?
2. Que se passe-t-il si tous les produits sont répliqués partout ?

3. Que se passe-t-il si les stocks sont fragmentés ?
4. Proposer une architecture réaliste conciliant performance et cohérence.

Exercice 3 — Cohérence faible vs forte

Vous gérez une application mobile mondiale avec 30 millions d'utilisateurs.
La recherche de produits peut supporter un léger retard des données (1–5 secondes).
Le paiement, lui, exige une cohérence stricte.

1. Quel type de synchronisation convient pour les recherches ?
2. Quel type pour les paiements ?
3. Comment combiner les deux au sein d'une même base distribuée ?
4. Donner un exemple réel où la cohérence faible serait dangereuse.

Exercice 4 — Conflits en réplication multi-maître

Un même client existe sur deux sites A et B.
Deux mises à jour simultanées se produisent :

- Sur A : $\text{solde} = \text{solde} + 50$
- Sur B : $\text{solde} = \text{solde} - 40$

Les deux mettent à jour le même enregistrement au même moment.

1. Pourquoi un conflit se produit ?
2. Quelle valeur doit être retenue ?
3. Proposer trois stratégies de résolution.
4. Quelle stratégie Oracle propose pour les conflits UPDATE ?
5. Quelle stratégie serait la plus juste dans un système bancaire ? Pourquoi ?

Exercice 5 — Materialized View ou base locale ?

Un site distant doit accéder à la table **PRODUIT**, mais seulement pour lecture.
Vous hésitez entre :

- une Materialized View (FAST REFRESH),
 - une réplication complète,
 - une simple requête distante via dblink.
1. Dans quelles situations utiliseriez-vous chacun des trois choix ?
 2. Quelles contraintes imposent les MV Logs ?
 3. Quel est l'impact si le réseau devient instable ?
 4. Que se passe-t-il si la MV FAST REFRESH échoue régulièrement ?
 5. Proposer une solution robuste.

Exercice 6 — Two-Phase Commit dans un environnement instable

Une transaction distribuée doit écrire dans :

- Site1 : CLIENT
- Site2 : COMMANDE
- Site3 : FACTURATION

Le réseau est instable et Site2 tombe en panne juste après la phase PREPARE.

1. Que fait Oracle dans ce cas ?
2. Que deviennent les verrous sur les sites participants ?
3. Comment un administrateur peut-il débloquer la situation ?
4. Quelles sont les limites du protocole 2PC ?
5. Quelle alternative (3PC, Paxos, etc.) serait plus robuste ?

Exercice 7 — Architecture distribuée complète

Vous devez concevoir une base distribuée pour une plateforme de transport international :

- Réservations
- Facturation
- Suivi des colis

Chaque continent possède un datacenter.

Les mises à jour sont nombreuses.

Les recherches doivent être globales.

1. Proposer un modèle de fragmentation pour chaque table.
2. Quels fragments doivent être répliqués ?
3. Quels fragments ne doivent **jamais** être répliqués ?
4. Quel mécanisme de synchronisation utiliser pour le suivi des colis (temps réel) ?
5. Comment gérer les pannes inter-continents ?

Exercice 8 — Allocation optimale

On dispose de 4 sites :

- **Site A** : forte lecture
- **Site B** : forte écriture
- **Site C** : lecture/écriture équilibrée
- **Site D** : backup longue durée

1. Où placer les fragments horizontaux ?
2. Où placer les fragments verticaux ?
3. Quelle stratégie de réplication partielle pour minimiser le trafic ?

4. Quels sites doivent être maîtres ?
5. Quel site doit contenir une copie complète ? Pourquoi ?

Exercice 9 — Discussion sur la transparence

Oracle vise 6 types de transparence dans les BDD distribuées :

- distribution
 - fragmentation
 - réplication
 - localisation
 - exécution
 - concurrence
1. Pourquoi la transparence est-elle indispensable ?
 2. Donner un cas où elle n'est pas souhaitable.
 3. Quel type de transparence est le plus difficile à atteindre ? Pourquoi ?
 4. Oracle assure-t-il une transparence totale ?
 5. Donner un exemple de transparence violée dans une jointure distribuée.

Exercice 10 — Problème ouvert : architecture optimale

Une entreprise veut :

- une base haute disponibilité (24/7),
 - forte lecture,
 - mises à jour modérées,
 - données dans 3 continents,
 - latence réseau variable,
 - besoin de cohérence forte uniquement pour la facturation.
1. Quel modèle de fragmentation recommandez-vous ?
 2. Quels types de réplication utiliser ?
 3. Quels mécanismes de synchronisation ?
 4. Où placer la facturation ?
 5. Comment gérer les MV en cas de défaillance réseau ?
 6. Proposer une architecture complète et justifiée.