

1. Contexte de l'exercice

On s'intéresse au système d'information d'une entreprise qui a trois sites en France.

- Le site de production et de vente à Lille
- Un site de vente à Toulouse
- Le siège social à Lyon

Pour simplifier, on ne considérera que :

les clients : nom, prenom, adresse

les produits : id, descriptif, prix unitaire et stock

les salariés : nom, prenom, evaluation, dernière promotion

Les factures par produit et par client : id, quantite et montant

Le siège social doit pouvoir sélectionner toutes les informations afin établir des tableaux de bord.

Pour un salarié, les promotions sont gérées par le siège social et l'évaluation par le site sur lequel il travaille. Les factures sont établies par le site concerné.

2. Bases de données réparties

Il faut créer trois bases de données : une sur un serveur à Lille, une autre sur un serveur à Toulouse et enfin une sur un serveur à Lyon. Sur chaque base de données, il faut créer s'il y a lieu des liens de bases de données vers les autres bases

Exemple sur la base de Lyon :

- create database link *db_toulouse* connect to *user_toulouse* identified by *password* using '*db_toulouse*'
- create database link *db_lille* connect to *user_lille* identified by *password* using '*db_lille*'

user_toulouse et *user_lille* sont les comptes dans lesquels on a créé les tables des bases de données de Toulouse et de Lille. Les noms *db_lille* et *db_toulouse* sont les noms des bases de données de lille sur le serveur de Lille et de Toulouse sur le serveur de Toulouse.

2.1 Les produits

La table réelle est créé sur la base de données Lille. Comme les deux autres bases de données auront fréquemment besoin de consulter cette table, il faudra créer sur chacune de ces bases, une vue matérialisée (donc à distance) de cette table.

Exemple sur la BDD de Toulouse **create view materialized vueProduit as select * from produit@db_lille.**

2.2 Les clients

Il faut ajouter un attribut *idCli* pour les clients

On fragmente horizontalement la table client sur les bases de données de Lille et de Toulouse.

Compte-tenu de la demande du siège social, il faudra dans la base de données de Lyon, créer deux vues matérialisées : une qui est une réplique de la table client de Lille et une autre qui est une réplique de la table client de Toulouse.

2.3 Les factures

Dans la table facture, il faut ajouter deux attribut une premier *idCli* clé étrangère avec la table client et un deuxième *idPro* avec la table produit.

ING2-SIE : BASES DE DONNEES AVANCEES : EXAMEN DE 1ERE SESSION S3

On fragmente horizontalement la table facture sur les bases de données de Lille et de Toulouse.

Compte-tenu de la demande du siège social, il faudra dans la base de données de Lyon, créer deux vues matérialisées : une qui est une réplication des factures de Lille et une autre qui est une réplication des factures client de Toulouse.

Pour optimiser, ces deux vues matérialisées doivent être la jointure des tables client, facture et vm_produit.

2.4 Les salariés

Il faut ajouter un attribut idSal pour les salariés.

On fait une fragmentation mixte (nommée salarie) de la table salarié sur les trois sites en ne retenant que les quatre attributs idSal, nom, prenom, evaluation. Chacune de ces tables ne contiennent que les salariés du site correspondant.

On fait une fragmentation verticale (nommée salariePromotion) de cette table sur le site de Lyon en tant que siège social qui gère les promotions. Cette fragmentation sera constituée des attributs id, nom, prenom, promotion. Elle contiendra les salariés des trois sites. Comme il y a redondance avec les tables ci-dessus, il faudra définir un système de trigger que son contenu et les contenus des tables ci-dessus. Pour être clair, toute insertion dans l'une des tables ci-dessus doit entraîner automatiquement une insertion d'une ligne dans cette table avec pour le champ promotion une valeur indéfinie.

Si le siège social a souvent besoin de consulter l'ensemble des informations de tous les salariés (les trois sites) alors il faut créer dans la base de données de Lyon deux vues matérialisées : une qui est une réplication à distance de la table salarie de Lille et une autre une réplication de la table salarie de Toulouse.

Pour finir, on donne ci-dessous la vue reconstituante de la table salarie (sans les promotions) sur le site de Lyon.

```
Create view vue_tous_les_salaries as
  select  * from vm_salarie_lille
  UNION
  Select * from vm_salarie_toulouse
  UNION
  Select * from salarie
```