

Mini- projet – GLPI (Gestion du parc informatique de CY Tech)
ING2 – Bases de données avancées
Année 2024 –2025

Ce mini projet consiste à repenser une partie de la BDD de GLPI pour améliorer les performances du parc informatique de CY Tech et prendre en compte l'aspect multi-sites (Cergy, Pau). Ce mini projet se fait par équipe de 3 ou 4 élèves. Pour réaliser le projet, vous devez faire du reverse engineering avec la doc de GLPI (<https://github.com/glpi-project/glpi>)

Le périmètre du projet couvre les aspects qui sont en rapport avec la gestion des matériels informatiques, des utilisateurs et des informations sur la structure des réseaux.

Le projet consiste à :

1. Définir une nouvelle base de données en utilisant toutes les notions vues en cours : users, rôles, tablespaces, cluster, index, vues, PLSQL (triggers, curseurs, procédures, fonctions...) bases de données réparties (BDDR), plan de requêtes.
2. Valider votre structure en faisant des tests de performances sur votre base à l'aide d'un jeu de test conséquent généré en PL/SQL.

Livrables et étapes du projet

Étape 1 : Modélisation et création d'une nouvelle base de données

L'objectif de cette étape est de définir et de créer une nouvelle architecture de base de données, en exploitant les notions vues en cours. Vous devrez :

- Concevoir une nouvelle structure qui améliore les performances et qui est adaptée aux besoins multi-sites,
- Implémenter les éléments suivants :
 - Utilisateurs et rôles : définir les droits et privilèges pour assurer une gestion fine des accès,
 - Tablespaces : organiser les données pour une meilleure gestion du stockage,
 - Clusters et index : optimiser les temps de requête et le traitement des données,
 - Vues : simplifier l'accès à des informations complexes et pertinentes pour les utilisateurs,
 - PL/SQL : développer des triggers, curseurs, procédures et fonctions pour automatiser des processus et garantir l'intégrité des données,
 - Bases de données réparties (BDDR) : gérer la distribution des données entre les sites de Cergy et Pau,
 - Plan de requêtes : analyser les plans d'exécution pour optimiser les performances des requêtes.

Étape 2 : Validation de la structure par des tests de performance

Une fois la base de données créée, vous devrez valider sa structure en réalisant des tests de performance. Cette phase inclut :

- La génération d'un jeu de test conséquent à l'aide de PL/SQL, en simulant des données représentatives du contexte réel (utilisateurs, matériels, réseaux, etc.),
- La réalisation de requêtes complexes pour vérifier les temps de réponse,
- La comparaison des résultats avec l'ancienne base pour mesurer les gains en performance.

Livrables attendus

À l'issue du projet, vous devrez remettre les éléments suivants :

1. Un rapport détaillé comprenant :
 - La démarche de reverse engineering (analyses et constatations),
 - La nouvelle modélisation de la base de données (diagrammes UML, schémas relationnels, etc.),
 - Les résultats des tests de performance (avec graphiques et analyses comparatives),
 - Les explications sur l'implémentation des concepts avancés (PL/SQL, BDDR, etc.).
2. Le code source de votre base de données et des scripts PL/SQL associés.
3. Une présentation orale (10-15 minutes) pour expliquer votre démarche, vos choix techniques et vos résultats.
4. Date limite des rendus jeudi 16 mars 2025
5. Date présentations vendredi 17 mars 2024

Évaluation

Votre projet sera évalué selon les critères suivants :

- Qualité de l'analyse initiale (reverse engineering),
- Pertinence et robustesse de la nouvelle structure de la base,
- Utilisation correcte des concepts avancés (PL/SQL, indexation, BDDR...),
- Performances mesurées et améliorations démontrées,
- Clarté et professionnalisme des livrables (rapport et présentation).

Remarques

- Le projet est à faire en groupe, mais votre note sera individuelle. Le travail doit donc être réparti équitablement entre les membres. Une (rapide) interrogation orale sur votre projet aura lieu lors de la soutenance.
- En ce qui concerne le plagiat : il est évident que vous devez avoir écrit vous-même votre code et vous devez donc bien sûr être capable de l'expliquer entièrement. Les similarités sont faciles à détecter automatiquement et seront sanctionnées.