



M1DWM - TP BD

Manal Ez-zaher
Anthony Menghi
Mehdi Ghoulam

Objectif général :

Créer une base de données pour gérer les formations universitaires (licences et masters) et les résultats des étudiants. Cette base de données sera utilisée par trois types d'utilisateurs : directeurs d'études, enseignants et étudiants, chacun ayant des niveaux d'accès différents.

Travail demandé :

Le travail se divise en plusieurs parties, avec des livrables spécifiques pour chaque étape. Vous devrez utiliser PostgreSQL pour implémenter la base de données, Python Faker pour générer des données fictives, et explorer des techniques avancées de gestion des utilisateurs, optimisation des requêtes, triggers, et procédures stockées.

Parties principales du projet :

1. Conception assistée de la base de données :

- Concevoir un schéma relationnel répondant aux besoins de gestion des formations et des résultats étudiants.
- Utiliser un outil d'IA générative pour vous aider dans cette conception, tout en évaluant de manière critique ses recommandations.
- Création d'un script SQL pour la création des tables, clés primaires, clés étrangères et autres contraintes (ex : `CHECK`).

2. Peuplement de la base de données :

- Utiliser Python Faker pour générer des données fictives réalistes avec une gestion stricte des clés étrangères pour maintenir l'intégrité référentielle.

- Générer des données sur une période de 2005 à 2024 pour au moins 1000 étudiants fictifs. Vous devrez tenir compte de contraintes spécifiques sur les inscriptions des étudiants (comme les redoublements et l'ordre chronologique des inscriptions).

3. Gestion des utilisateurs et création de vues :

- Mettre en place une gestion des accès via des rôles (étudiant, enseignant, directeur des études), avec des droits différenciés.
- Créer au moins cinq vues différentes selon les besoins des utilisateurs, par exemple pour la consultation des maquettes de formation ou des notes.
- Attribuer les privilèges nécessaires à chaque rôle, en respectant les contraintes de sécurité.

4. Requêtes et optimisation :

- Proposer au moins deux versions de requêtes SQL complexes et les optimiser.
- Les requêtes concernent des statistiques sur les étudiants, les notes, et les filières, et vous devrez comparer leurs performances à l'aide des outils de diagnostic de PostgreSQL (plans d'exécution).
- Créer des index pertinents pour améliorer les performances des requêtes.

5. Triggers et procédures stockées :

- Créer des triggers pour gérer l'inscription des étudiants (par exemple, vérifier le nombre d'inscrits ou empêcher une inscription sans validation de l'année précédente).
- Créer des procédures stockées pour des opérations courantes comme calculer la moyenne d'un étudiant, préparer les délibérations, générer les fiches de notes, et archiver les résultats anciens.

Livrables :

1. Rapport synthétique :

- Description de la conception de la base.
- Méthodologie sur l'utilisation d'une IA générative et Python Faker.

- Explication des scripts, optimisation des requêtes avec captures d'écran des outils de diagnostic.
2. **Scripts SQL :**
 - Création de la base de données.
 - Gestion des utilisateurs, vues et privilèges.
 - Requêtes complexes.
 - Triggers et procédures stockées.
 3. **Scripts Python :**
 - Utilisés pour générer les données fictives avec Python Faker.
 4. **Soutenance orale :**
 - Une présentation de 20 minutes décrivant la démarche, le schéma, les données, les requêtes et l'optimisation.

Critères d'évaluation :

1. **Conception de la base de données :** Pertinence, intégrité, cohérence des données et originalité.
2. **Maîtrise des solutions proposées :** Efficacité des vues, des requêtes, des triggers et des procédures stockées.
3. **Documentation et méthodologie :** Clarté de la documentation et analyse critique de l'usage de l'IA et de Python Faker.
4. **Qualité de la soutenance orale.**