

Cas 1

L'entreprise Clevity fournit des prestations techniques à ses clients : audits, déploiements, formations, assistance. Chaque client peut avoir plusieurs projets, suivis par des intervenants internes. Ces projets sont souvent multi-sites et impliquent plusieurs collaborateurs de différentes agences.

Actuellement, chaque agence saisit ces informations dans ses propres fichiers Excel ou CSV, sans format normalisé ni consolidation centrale. Cette situation entraîne des doublons, des incohérences, des pertes d'information, et complique fortement le suivi global de l'activité.

La direction a validé un projet de centralisation des données historiques dans une base unique hébergée sur MariaDB. Ce projet doit permettre à la fois de :

- Migrer les données existantes de manière fiable ;
- Mettre en place des contrôles de cohérence automatisés ;
- Garantir la traçabilité des opérations de migration ;
- Simuler une reprise partielle de données après détection d'erreurs ;
- Documenter l'ensemble du processus dans une logique de qualité et de conformité.

Votre rôle est de prendre en charge la conception, la migration, le contrôle et la correction de ces données à travers une série d'étapes techniques automatisées.

Étapes du travail

1. Analyse et modélisation

- Analyser les besoins métier.
- Identifier les entités et leurs relations (clients, projets, intervenants, affectations).
- Définir les règles d'intégrité fonctionnelle et créer la bdd en fonction.

2. Génération de données de test

- Développer un script Python `generate_data.py` qui crée des jeux de données de test dans les fichiers :
 - `clients.csv`
 - `intervenants.csv`
 - `projets.csv`
 - `affectations.csv`
- Les données doivent être réalistes, respectant les contraintes métier, notamment :

- Un projet doit avoir un client.
- Une note de satisfaction ne doit exister que pour les projets terminés.
- Les emails doivent être uniques.

3. Migration vers MariaDB

- Écrire un script migration.py qui :
 - Lit les CSV générées.
 - Effectue les transformations nécessaires.
 - Insère les données fournies (dossier data_1) en base MariaDB.
 - Journalise toutes les opérations dans log.txt.
 - Produit un rapport rapport_reprise.txt résumant les insertions, erreurs, ignorés.
 - Importe les données fournies pour réaliser la migration réelle en production

4. Validation post-migration

- Développer un script validate.py qui :
 - Contrôle les volumes d'enregistrements.
 - Vérifie les contraintes d'unicité, non nullité, cohérence inter-tables.
 - Génère un fichier rapport_validation.txt horodaté.

5. Reprise partielle

- Un fichier projets_corrompus.csv vous sera fourni contenant des erreurs.
- Écrire reprise.py qui :
 - Identifie les lignes erronées
 - Corrige ou isole ces lignes.
 - Réinjecte uniquement les données valides.
 - Loggue les actions dans log_reprise.txt.

6. Rapport final

Vous ferez votre rapport sous forme de compte rendu oral