



Rapport d'Exploration et d'Analyse des Données

1. Objectif de la démarche

Dans le cadre de la refonte de l'application **MYDental BI**, la première étape consistait à auditer l'existant pour comprendre pourquoi les indicateurs actuels étaient défaillants et préparer l'ajout du simulateur de rentabilité.

2. Méthodologie et Outils

Nous avons mis en place une chaîne d'analyse rigoureuse pour explorer la base de données `dental_data.db` (initialement en MariaDB) :

Rétro-ingénierie et Modélisation (DBeaver)

- Connexion de l'outil **DBeaver** à la base de données pour générer un diagramme Entité-Association (ERD).
- Visualisation des relations complexes entre les tables clés comme `user`, `doctor`, `consultation` et `payment`.

Scripts de Conversion et d'Analyse

- Développement de scripts personnalisés pour convertir la base **MariaDB** en **SQLite**, facilitant l'analyse locale et l'exécution de tests rapides.
- Création d'un script d'analyse automatisé générant un rapport statistique complet sur les **80 tables** de la base :
 - Volumétrie
 - Taux de valeurs NULL
 - Types de données

Analyse Assistée par IA

- Génération d'une version "réduite" de la base de données (`import_mini.sql`) contenant un échantillon représentatif des données.
 - Soumission de la structure à des outils d'IA pour valider la cohérence logique du schéma sans traiter le volume massif de production.
-

3. Constats et Résultats

L'analyse approfondie a révélé des disparités majeures entre la structure théorique et les données réelles :

Tables "Fantômes"

- De nombreuses tables sont structurellement présentes mais totalement vides (**0 lignes**).
- Cela concerne notamment :
 - Les modules de spécialités (`Ophthalmologie`, `Gynécologie`, `Cardiologie`)

- Les tables de facturation formelle (`invoice`, `invoice_prothese`)

Flux Financier

- Une incohérence critique a été identifiée :
 - Les tables de factures sont vides.
 - La table `payment` contient un volume important de données (**plus de 217 000 enregistrements**).

Données Paramétriques Manquantes

- Les colonnes nécessaires au calcul de rentabilité dans la table `doctor` (ex: `consult_percent`) sont présentes mais non renseignées (valeurs à **0** ou **NULL**).

4. Conclusion et Orientation Technique

Cette phase d'exploration a été décisive. Elle a permis de comprendre que les "erreurs" perçues dans l'application ne provenaient pas d'une corruption de la base de données, mais d'une mauvaise interrogation de celle-ci par le code existant.

Décision validée

Conformément aux contraintes du projet, **la structure de la base de données ne sera pas modifiée.**

L'effort de correction se concentrera sur le Backend **Node.js** :

- Redirection des requêtes financières vers la table `payment` au lieu de `invoice`.
- Intégration de la logique de calcul de rentabilité directement dans le code JavaScript (**Frontend/Backend**) pour pallier l'absence de paramètres en base.