Exercice : système de gestion d’un clinique

Pour implémenter ces fonctionnalités, vous auriez généralement besoin d'un système backend avec des API appropriées pour chaque action. Voici un aperçu de comment vous pourriez le structurer :

1. **Médecins :**
    - **Obtenir des médecins par filtre :** Point d'API pour récupérer des médecins en fonction de filtres spécifiés tels que la spécialité, l'emplacement, etc.
    - **Obtenir un médecin par ID :** Point d'API pour récupérer les détails d'un médecin spécifique par leur identifiant unique.
    - **Ajouter / Mettre à jour / Supprimer un médecin :** APIs pour ajouter un nouveau médecin, mettre à jour les détails du médecin existant, et supprimer un médecin du système.
2. **Patients :**
    - **Obtenir des patients par filtre :** Point d'API pour récupérer des patients en fonction de critères spécifiés tels que l'âge, le diagnostic, etc.
    - **Obtenir un patient par ID :** Point d'API pour récupérer les détails d'un patient spécifique par leur identifiant unique.
    - **Ajouter / Supprimer des patients :** APIs pour ajouter de nouveaux patients au système et supprimer les enregistrements de patients existants.
    - **Mettre à jour le diagnostic du patient :** API pour mettre à jour le diagnostic ou l'état médical d'un patient.
3. **Maladies :**
    - **Obtenir des maladies par filtre :** Point d'API pour récupérer des maladies en fonction de filtres spécifiés tels que la catégorie, la gravité, etc.

- **Obtenir une maladie par ID :** Point d'API pour récupérer les détails d'une maladie spécifique par son identifiant unique.
- **Ajouter / Supprimer des maladies :** APIs pour ajouter de nouvelles maladies au système et supprimer les maladies existantes.
- **Assigner une maladie aux patients :** API pour associer une maladie à un ou plusieurs patients dans le système.

4. **Chambres :**
    - **Obtenir des chambres :** Point d'API pour récupérer les détails de toutes les chambres disponibles dans l'établissement de santé.
    - **Assigner des chambres aux patients :** API pour attribuer des chambres aux patients lors de leur admission.
    - **Ajouter / Supprimer des chambres :** APIs pour ajouter de nouvelles chambres au système ou supprimer des chambres existantes.

Ce schéma décrit la structure de base de vos tables de base de données. Vous pouvez l'élargir en fonction des exigences spécifiques ou des fonctionnalités supplémentaires que vous souhaitez implémenter. Les contraintes de clé étrangère garantissent l'intégrité référentielle en liant les enregistrements dans différentes tables. De plus, vous voudrez peut-être ajouter des index pour des requêtes plus rapides, ainsi que des contraintes ou des déclencheurs pour assurer l'intégrité et la cohérence des données.

1. **Médecins :**
    - id (Clé primaire)
    - nom
    - spécialité
    - coordonnées
    - autres champs pertinents
2. **Patients :**
    - id (Clé primaire)

- nom
- âge
- genre
- diagnostic
- coordonnées
- autres champs pertinents

3. **Maladies :**
   - id (Clé primaire)
   - nom
   - catégorie
   - gravité
   - autres champs pertinents
4. **Chambres :**
   - id (Clé primaire)
   - numéro de chambre
   - type de chambre
   - statut de disponibilité
   - autres champs pertinents
5. **Assignations :**
   - id (Clé primaire)
   - doctor_id (Clé étrangère - Médecins)
   - patient_id (Clé étrangère - Patients)
   - room_id (Clé étrangère - Chambres)
   - date d'assignation
   - date de sortie
   - autres champs pertinents