**Méthode d’ordre par type**

1. Patient

- Méthode : `trierPatientsParNom()`

- Description : Trie les patients par leur nom de famille.

2. Medecin

- Méthode : `trierMedecinsParSpecialite()`

- Description : Trie les médecins par leur spécialité.

3. RendezVous

- Méthode : `trierRendezVousParDate()`

- Description : Trie les rendez-vous par la date du rendez-vous.

4. Consultation

- Méthode : `trierConsultationsParDate()`

- Description : Trie les consultations par la date de consultation.

5. Facture

- Méthode : `trierFacturesParMontant()`

- Description : Trie les factures par le montant total.

6. Examen

- Méthode : `trierExamensParDate()`

- Description : Trie les examens par la date de l'examen.

7. Prescription

- Méthode : `trierPrescriptionsParDate()`

- Description : Trie les prescriptions par la date de prescription.

**Méthodes de gestion des liens N dans le schéma**

1. Association Patient - RendezVous

Description : Un patient peut avoir plusieurs rendez-vous.

- Méthode : `ajouterRendezVous(Patient, RendezVous)`: Ajoute un rendez-vous pour un patient.

- Méthode : `supprimerRendezVous(Patient, RendezVous)`: Supprime un rendez-vous d'un patient.

- Méthode : `listerRendezVous(Patient)`: Liste tous les rendez-vous d'un patient.

2. Association Medecin - RendezVous

Description : Un médecin peut avoir plusieurs rendez-vous.

- Méthode : `ajouterRendezVous(Medecin, RendezVous)`: Ajoute un rendez-vous pour un médecin.

- Méthode : `supprimerRendezVous(Medecin, RendezVous)`: Supprime un rendez-vous d'un médecin.

- Méthode : `listerRendezVous(Medecin)`: Liste tous les rendez-vous d'un médecin.

3. Association Patient - Consultation

Description : Un patient peut passer plusieurs consultations.

- Méthode : `ajouterConsultation(Patient, Consultation)`: Ajoute une consultation pour un patient.

- Méthode : `supprimerConsultation(Patient, Consultation)`: Supprime une consultation d'un patient.

- Méthode : `listerConsultations(Patient)`: Liste toutes les consultations d'un patient.

4. Association Medecin - Consultation

Description : Un médecin peut effectuer plusieurs consultations.

- Méthode : `ajouterConsultation(Medecin, Consultation)`: Ajoute une consultation pour un médecin.

- Méthode : `supprimerConsultation(Medecin, Consultation)`: Supprime une consultation d'un médecin.

- Méthode : `listerConsultations(Medecin)`: Liste toutes les consultations d'un médecin.

5. Association Consultation - Examen

Description : Une consultation peut nécessiter plusieurs examens.

- Méthode : `ajouterExamen(Consultation, Examen)`: Ajoute un examen pour une consultation.

- Méthode : `supprimerExamen(Consultation, Examen)`: Supprime un examen d'une consultation.

- Méthode : `listerExamens(Consultation)`: Liste tous les examens d'une consultation.

6. Association Consultation - Prescription

Description : Une consultation peut contenir plusieurs prescriptions.

- Méthode : `ajouterPrescription(Consultation, Prescription)`: Ajoute une prescription pour une consultation.

- Méthode : `supprimerPrescription(Consultation, Prescription)`: Supprime une prescription d'une consultation.

- Méthode : `listerPrescriptions(Consultation)`: Liste toutes les prescriptions d'une consultation.

7. Association Patient - Facture

Description : Un patient peut recevoir plusieurs factures.

- Méthode : `ajouterFacture(Patient, Facture)`: Ajoute une facture pour un patient.

- Méthode : `supprimerFacture(Patient, Facture)`: Supprime une facture d'un patient.

- Méthode : `listerFactures(Patient)`: Liste toutes les factures d'un patient.

**Méthodes applicatives de consultation par type**

1. Patient

Méthode 1 : `listerPatients()`

- Description : Retourne la liste de tous les patients.

- Requête SQL : `SELECT \* FROM Patient;`

Méthode 2 : `rechercherPatientParNom(nom)`

- Description : Recherche et retourne les patients dont le nom correspond au paramètre.

- Requête SQL : `SELECT \* FROM Patient WHERE Nom = :nom;`

2. Medecin

Méthode 1 : `listerMedecins()`

- Description : Retourne la liste de tous les médecins.

- Requête SQL : `SELECT \* FROM Medecin;`

Méthode 2 : `rechercherMedecinParSpecialite(specialite)`

- Description : Recherche et retourne les médecins dont la spécialité correspond au paramètre.

- Requête SQL : `SELECT \* FROM Medecin WHERE Specialite = :specialite;`

3. RendezVous

Méthode 1 : `listerRendezVous()`

- Description : Retourne la liste de tous les rendez-vous.

- Requête SQL : `SELECT \* FROM RendezVous;`

Méthode 2 : `rechercherRendezVousParDate(date)`

- Description : Recherche et retourne les rendez-vous prévus à une date spécifique.

- Requête SQL : `SELECT \* FROM RendezVous WHERE DateRendezVous = :date;`

4. Consultation

Méthode 1 : `listerConsultations()`

- Description : Retourne la liste de toutes les consultations.

- Requête SQL : `SELECT \* FROM Consultation;`

Méthode 2 : `rechercherConsultationParPatient(patientId)`

- Description : Recherche et retourne les consultations pour un patient spécifique.

- Requête SQL : `SELECT \* FROM Consultation WHERE PatientId = :patientId;`

5. Facture

Méthode 1 : `listerFactures()`

- Description : Retourne la liste de toutes les factures.

- Requête SQL : `SELECT \* FROM Facture;`

Méthode 2 : `rechercherFactureParMontant(montant)`

- Description : Recherche et retourne les factures dont le montant est supérieur à un certain montant.

- Requête SQL : `SELECT \* FROM Facture WHERE Montant > :montant;`

6. Examen

Méthode 1 : `listerExamens()`

- Description : Retourne la liste de tous les examens.

- Requête SQL : `SELECT \* FROM Examen;`

Méthode 2 : `rechercherExamenParDate(date)`

- Description : Recherche et retourne les examens prévus à une date spécifique.

- Requête SQL : `SELECT \* FROM Examen WHERE DateExamen = :date;`

7. Prescription

Méthode 1 : `listerPrescriptions()`

- Description : Retourne la liste de toutes les prescriptions.

- Requête SQL : `SELECT \* FROM Prescription;`

Méthode 2 : `rechercherPrescriptionParConsultation(consultationId)`

- Description : Recherche et retourne les prescriptions pour une consultation spécifique.

- Requête SQL : `SELECT \* FROM Prescription WHERE ConsultationId = :consultationId;`

Méthodes CRUD avec valeurs remplacées

1. Patient

Create : `ajouterPatient(Patient patient)`

- Description : Ajoute un nouveau patient.

- Requête SQL : `INSERT INTO Patient (Nom, Prenoms, Adresse, Email, Telephones, DateNaissance) VALUES ('Dupont', 'Jean', '123 Rue Example, Paris', 'jean.dupont@example.com', '0123456789', '1980-01-01');`

Read : `lirePatient(int patientId)`

- Description : Retourne les détails d'un patient par son ID.

- Requête SQL : `SELECT \* FROM Patient WHERE IdPatient = 1;`

Update : `modifierPatient(Patient patient)`

- Description : Met à jour les informations d'un patient.

- Requête SQL : `UPDATE Patient SET Nom = 'Martin', Prenoms = 'Paul', Adresse = '456 Rue Exemple, Paris', Email = 'paul.martin@example.com', Telephones = '0987654321', DateNaissance = '1975-05-05' WHERE IdPatient = 1;`

Delete : `supprimerPatient(int patientId)`

- Description : Supprime un patient par son ID.

- Requête SQL : `DELETE FROM Patient WHERE IdPatient = 1;`

2. Medecin

Create : `ajouterMedecin(Medecin medecin)`

- Description : Ajoute un nouveau médecin.

- Requête SQL : `INSERT INTO Medecin (Nom, Prenoms, Specialite, Email, CV) VALUES ('Durand', 'Marie', 'Cardiologue', 'marie.durand@example.com', 'CV.pdf');`

Read : `lireMedecin(int medecinId)`

- Description : Retourne les détails d'un médecin par son ID.

- Requête SQL : `SELECT \* FROM Medecin WHERE IdMedecin = 1;`

Update : `modifierMedecin(Medecin medecin)`

- Description : Met à jour les informations d'un médecin.

- Requête SQL : `UPDATE Medecin SET Nom = 'Lefevre', Prenoms = 'Luc', Specialite = 'Dermatologue', Email = 'luc.lefevre@example.com', CV = 'new\_cv.pdf' WHERE IdMedecin = 1;`

Delete : `supprimerMedecin(int medecinId)`

- Description : Supprime un médecin par son ID.

- Requête SQL : `DELETE FROM Medecin WHERE IdMedecin = 1;`

3. RendezVous

Create : `ajouterRendezVous(RendezVous rendezVous)`

- Description : Ajoute un nouveau rendez-vous.

- Requête SQL : `INSERT INTO RendezVous (PatientId, MedecinId, DateRendezVous, Motif) VALUES (1, 2, '2024-08-10', 'Consultation annuelle');`

Read : `lireRendezVous(int rendezVousId)`

- Description : Retourne les détails d'un rendez-vous par son ID.

- Requête SQL : `SELECT \* FROM RendezVous WHERE IdRendezVous = 1;`

Update : `modifierRendezVous(RendezVous rendezVous)`

- Description : Met à jour les informations d'un rendez-vous.

- Requête SQL : `UPDATE RendezVous SET PatientId = 2, MedecinId = 1, DateRendezVous = '2024-09-01', Motif = 'Suivi post-opératoire' WHERE IdRendezVous = 1;`

Delete : `supprimerRendezVous(int rendezVousId)`

- Description : Supprime un rendez-vous par son ID.

- Requête SQL : `DELETE FROM RendezVous WHERE IdRendezVous = 1;`

4. Consultation

Create : `ajouterConsultation(Consultation consultation)`

- Description : Ajoute une nouvelle consultation.

- Requête SQL : `INSERT INTO Consultation (MedecinId, PatientId, DateConsultation, Raison, Diagnostic) VALUES (1, 1, '2024-08-15', 'Douleur thoracique', 'Angine de poitrine');`

Read : `lireConsultation(int consultationId)`

- Description : Retourne les détails d'une consultation par son ID.

- Requête SQL : `SELECT \* FROM Consultation WHERE IdConsultation = 1;`

Update : `modifierConsultation(Consultation consultation)`

- Description : Met à jour les informations d'une consultation.

- Requête SQL : `UPDATE Consultation SET MedecinId = 2, PatientId = 1, DateConsultation = '2024-09-05', Raison = 'Suivi annuel', Diagnostic = 'Aucune anomalie détectée' WHERE IdConsultation = 1;`

Delete : `supprimerConsultation(int consultationId)`

- Description : Supprime une consultation par son ID.

- Requête SQL : `DELETE FROM Consultation WHERE IdConsultation = 1;`

5. Facture

Create : `ajouterFacture(Facture facture)`

- Description : Ajoute une nouvelle facture.

- Requête SQL : `INSERT INTO Facture (Montant, DateFacture, PatientId, ConsultationId) VALUES (150.0, '2024-08-16', 1, 1);`

Read : `lireFacture(int factureId)`

- Description : Retourne les détails d'une facture par son ID.

- Requête SQL : `SELECT \* FROM Facture WHERE IdFacture = 1;`

Update : `modifierFacture(Facture facture)`

- Description : Met à jour les informations d'une facture.

- Requête SQL : `UPDATE Facture SET Montant = 200.0, DateFacture = '2024-09-06', PatientId = 2, ConsultationId = 1 WHERE IdFacture = 1;`

Delete : `supprimerFacture(int factureId)`

- Description : Supprime une facture par son ID.

- Requête SQL : `DELETE FROM Facture WHERE IdFacture = 1;`

6. Examen

Create : `ajouterExamen(Examen examen)`

- Description : Ajoute un nouvel examen.

- Requête SQL : `INSERT INTO Examen (Details, DateExamen, ConsultationId) VALUES ('Radiographie thoracique', '2024-08-17', 1);`

Read : `lireExamen(int examenId)`

- Description : Retourne les détails d'un examen par son ID.

- Requête SQL : `SELECT \* FROM Examen WHERE IdExamen = 1;`

Update : `modifierExamen(Examen examen)`

- Description : Met à jour les informations d'un examen.

- Requête SQL : `UPDATE Examen SET Details = 'Scanner thoracique', DateExamen = '2024-09-07', ConsultationId = 1 WHERE IdExamen = 1;`

Delete : `supprimerExamen(int examenId)`

- Description : Supprime un examen par son ID.

- Requête SQL : `DELETE FROM Examen WHERE IdExamen = 1;`

7. Prescription

Create : `ajouterPrescription(Prescription prescription)`

- Description : Ajoute une nouvelle prescription.

- Requête SQL : `INSERT INTO Prescription (Details, DatePrescription, ConsultationId) VALUES ('Paracétamol 500mg', '2024-08-18', 1);`

Read : `lirePrescription(int prescriptionId)`

- Description : Retourne les détails d'une prescription par son ID.

- Requête SQL : `SELECT \* FROM Prescription WHERE IdPrescription = 1;`

Update : `modifierPrescription(Prescription prescription)`

- Description : Met à jour les informations d'une prescription.

- Requête SQL : `UPDATE Prescription SET Details = 'Ibuprofène 400mg', DatePrescription = '2024-09-08', ConsultationId = 1 WHERE IdPrescription = 1;`

Delete : `supprimerPrescription(int prescriptionId)`

- Description : Supprime une prescription par son ID.

- Requête SQL : `DELETE FROM Prescription WHERE IdPrescription = 1;`