



Université Jean Monnet, Saint-Étienne

Faculté des Sciences et Techniques

# Rapport de Bases de Données

**Encadrant :** Madame SAMUEL Emilie

**Étudiant 1 :** GHODBANE Rachid

**Étudiant 2 :** FAIQ Daryan

**Étudiant 3 :** LAICHE Khayr Eddine

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Répartition du travail</b>	<b>2</b>
<b>3</b>	<b>Définition des Données</b>	<b>2</b>
3.1	Présentation du Sujet . . . . .	2
3.2	Schéma de la Base de Données . . . . .	2
3.3	Contraintes sur les Données . . . . .	4
3.3.1	Contraintes Intra-Tables (CHECK) . . . . .	4
3.3.2	Contraintes Inter-Tables (FOREIGN KEY) . . . . .	4
<b>4</b>	<b>Utilisateurs et Rôles</b>	<b>4</b>
4.1	Administrateur . . . . .	4
4.2	Médecin . . . . .	5
4.3	Infirmier . . . . .	5
4.4	Administratif . . . . .	5
4.5	Exemple de Code SQL pour Assigner les Privilèges . . . . .	5
<b>5</b>	<b>Objets Catalogués Définis</b>	<b>6</b>
5.1	Gestion du Peuplement . . . . .	6
5.2	Gestion des Triggers . . . . .	6
5.3	Programmation et Planification . . . . .	6
5.4	Gestion des Factures et Rapports . . . . .	6
5.5	Événements Automatisés . . . . .	7
<b>6</b>	<b>Perspectives d'Évolution</b>	<b>7</b>
<b>7</b>	<b>Conclusion</b>	<b>7</b>

## 1. Introduction

Dans le cadre de notre projet de bases de données, nous avons développé une base de données pour la gestion des informations dans un hôpital. Ce projet a pour objectif de modéliser et de structurer les différentes entités et interactions présentes dans un environnement hospitalier.

La base de données permet de gérer les informations relatives aux patients, au personnel, aux consultations, aux interventions chirurgicales, aux admissions, aux urgences, aux chambres, aux historiques médicaux et à la facturation.

La conception de cette base de données vise à assurer l'intégrité et la cohérence des données, tout en offrant des fonctionnalités avancées telles que la génération de rapports et la gestion des rôles et des privilèges des utilisateurs.

Ce rapport présente en détail la structure de la base de données, les rôles et privilèges des utilisateurs, les objets catalogués définis, ainsi que la répartition du travail entre les membres du trinôme. Nous aborderons également les contraintes sur les données et les points pertinents liés à la réalisation de ce projet.

L'objectif principal de ce projet est de fournir une solution efficace et fiable pour la gestion des données hospitalières, en facilitant le travail du personnel médical et administratif, et en améliorant la qualité des soins aux patients.

## 2. Répartition du travail

Tâches	Rachid	Khayr Eddine	Daryan
Modélisation et schéma	✓	✓	
Création des tables et contraintes		✓	✓
Rapport	✓	✓	✓
Gestion des utilisateurs et privilèges	✓		
Procédures	✓	✓	✓
Fonctions	✓	✓	✓
Déclencheurs	✓	✓	✓
Événements		✓	
Tests	✓	✓	✓

## 3. Définition des Données

### 3.1. Présentation du Sujet

Dans le cadre de notre projet de bases de données, nous avons développé une base de données pour la gestion des informations dans un hôpital. Cette base de données a pour objectif de modéliser et de structurer les différentes entités et interactions présentes dans un environnement hospitalier.

La base de données permet de gérer les informations relatives aux patients, au personnel, aux consultations, aux interventions chirurgicales, aux admissions, aux urgences, aux chambres, aux historiques médicaux et à la facturation. Elle vise à assurer l'intégrité et la cohérence des données, tout en offrant des fonctionnalités avancées telles que la génération de rapports et la gestion des rôles et des privilèges des utilisateurs.

### 3.2. Schéma de la Base de Données

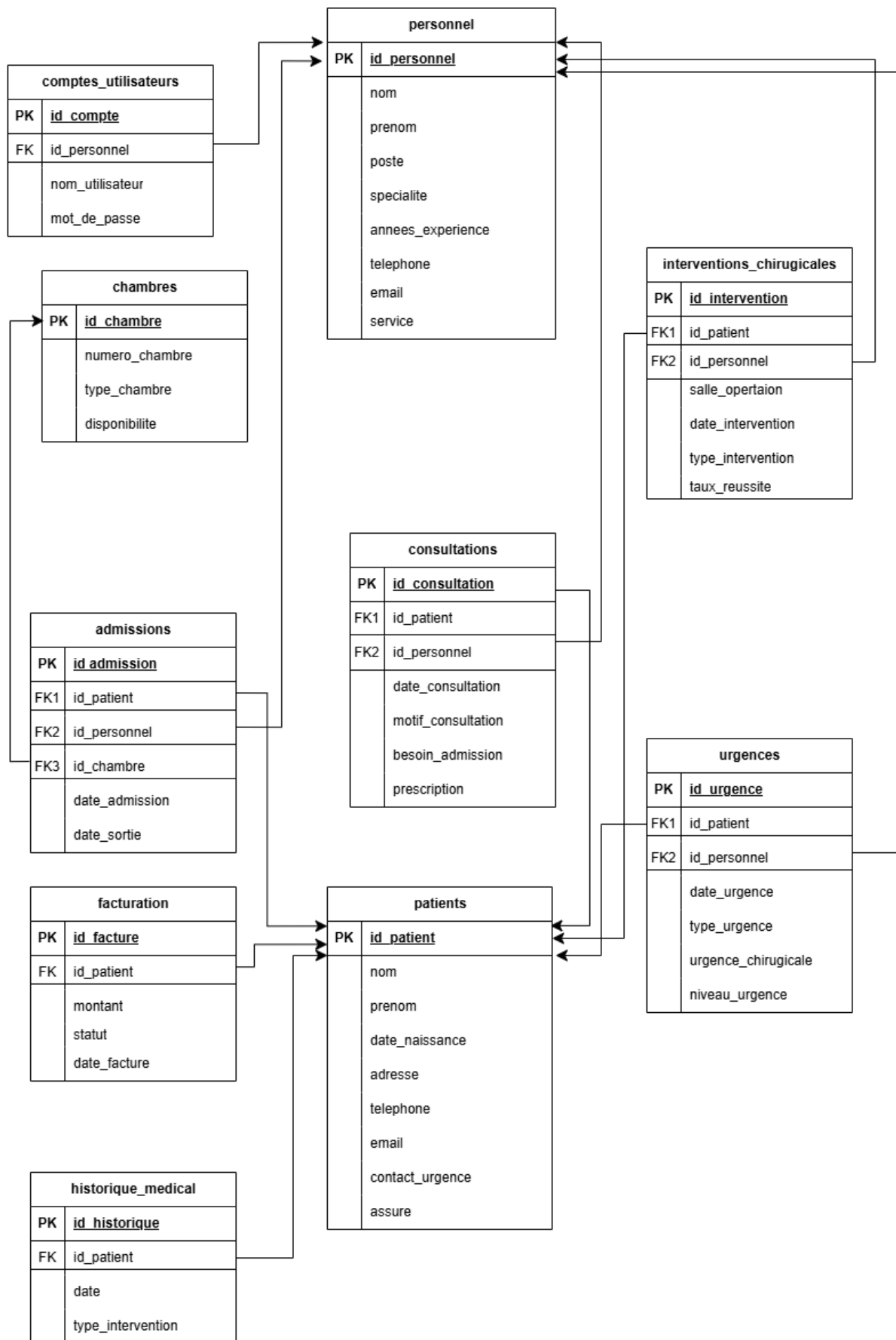


FIGURE 1 – Schéma de la Base de Données

### 3.3. Contraintes sur les Données

Afin d'assurer l'intégrité de notre base de données, nous avons appliqué des contraintes judicieuses lors de la création des tables. Parmi celles-ci, certaines retiennent particulièrement notre attention :

#### 3.3.1 Contraintes Intra-Tables (CHECK)

Nous avons mis en place des contraintes intra-tables pour garantir la validité des données au sein de chaque table. Par exemple :

- La date de naissance des patients doit être antérieure ou égale à la date actuelle.
- Les adresses email doivent respecter un format valide.
- Les années d'expérience du personnel doivent être comprises entre 0 et 50.
- La date de sortie des admissions doit être postérieure ou égale à la date d'admission.
- Le niveau d'urgence doit être compris entre 1 et 5.
- Le taux de réussite des interventions chirurgicales doit être compris entre 0 et 100 ... etc

#### 3.3.2 Contraintes Inter-Tables (FOREIGN KEY)

Pour assurer l'intégrité référentielle entre les différentes tables de la base de données, des contraintes inter-tables (FOREIGN KEY) ont été mises en place. Ces contraintes garantissent que les relations entre les entités sont cohérentes et valides. Voici un résumé des principales contraintes :

- **Admissions, Consultations, Urgences et Interventions Chirurgicales** : Ces tables doivent référencer des patients et des personnels existants. Cela garantit que chaque enregistrement est associé à un patient et un personnel valides.
- **Admissions** : Les admissions doivent également référencer des chambres existantes, assurant qu'elles sont correctement assignées.
- **Facturation** : Les factures doivent référencer des patients existants, ce qui garantit qu'elles sont associées à des patients valides.

Pour implémenter cette logique, nous avons ajouté des contraintes de clé étrangère avec des actions spécifiques comme DELETE ON CASCADE et ON UPDATE CASCADE. Par exemple, la suppression d'un enregistrement dans une table principale se répercute automatiquement sur toutes les références à cet enregistrement dans d'autres tables, assurant une gestion efficace des relations entre les données.

Ces contraintes, bien que fondamentales, jouent un rôle crucial dans la gestion de l'intégrité des données dans notre base de données.

## 4. Utilisateurs et Rôles

Dans notre base de données hospitalière, plusieurs rôles ont été définis pour garantir une gestion efficace et sécurisée des données. Chaque rôle dispose de privilèges spécifiques en fonction de ses responsabilités.

### 4.1. Administrateur

L'administrateur dispose des privilèges les plus élevés, lui permettant de superviser l'ensemble de la base de données.

**Privilèges :**

- Créer, modifier et supprimer des utilisateurs.
- Accéder et modifier toutes les tables.
- Gérer les rôles et privilèges.
- Générer des rapports.

## 4.2. Médecin

Le médecin a accès aux informations médicales des patients et peut gérer les consultations, interventions chirurgicales, et historiques médicaux.

**Privilèges :**

- Accéder et modifier les informations des patients.
- Gérer les consultations et interventions chirurgicales.
- Consulter les historiques médicaux.

## 4.3. Infirmier

L'infirmier gère les admissions et urgences tout en ayant un accès limité aux données des patients.

**Privilèges :**

- Accéder et modifier les informations des patients.
- Gérer les admissions et urgences.
- Consulter les historiques médicaux.

## 4.4. Administratif

Le personnel administratif s'occupe de la gestion des informations administratives et financières des patients.

**Privilèges :**

- Accéder et modifier les informations des patients.
- Gérer les factures et les comptes utilisateurs.
- Générer des rapports financiers.

Ces rôles et privilèges assurent que chaque utilisateur peut accomplir ses tâches tout en maintenant la sécurité et la confidentialité des données.

## 4.5. Exemple de Code SQL pour Assigner les Privilèges

Voici un exemple de code SQL permettant de créer des utilisateurs et d'assigner les privilèges appropriés :

```
-- Création des utilisateurs
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'mot_de_passe';
CREATE USER 'medecin'@'localhost' IDENTIFIED BY 'mot_de_passe';

-- Assignment des privilèges pour l'administrateur
GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost' WITH GRANT OPTION;

-- Assignment des privilèges pour le médecin
GRANT SELECT, INSERT, UPDATE ON hopital.patients TO 'medecin'@'localhost';
GRANT SELECT, INSERT, UPDATE ON hopital.consultations TO 'medecin'@'localhost';
GRANT SELECT, INSERT, UPDATE ON hopital.interventions_chirurgicales TO
'medecin'@'localhost';
GRANT SELECT ON hopital.historique_medical TO 'medecin'@'localhost';
```

## 5. Objets Catalogués Définis

Cette section présente les objets clés de notre base de données, tels que les procédures, triggers, et autres outils essentiels à la gestion hospitalière. Nous détaillons ici les fonctionnalités principales et leur impact sur le système.

### 5.1. Gestion du Peuplement

Le peuplement des tables initialise les données tout en respectant les contraintes métier :

- **Peuplement des admissions** : La procédure **p\_peuplement\_admission** attribue des chambres disponibles aux patients, en tenant compte des disponibilités des infirmiers et des ressources hospitalières.
- **Création des comptes utilisateurs** : La procédure **p\_peupler\_comptes\_utilisateurs** génère des comptes uniques et sécurisés pour le personnel, en utilisant des noms d'utilisateur distincts et des mots de passe générés par **f\_generer\_mot\_de\_passe**.

Ces procédures automatisent les tâches critiques tout en assurant l'intégrité des données.

### 5.2. Gestion des Triggers

Les triggers garantissent la cohérence des données lors des modifications :

- **Vérification des contraintes** : **t\_bf\_ins\_personnel** contrôle la validité des postes, spécialités, formats de téléphone et email avant insertion.
- **Mise à jour des chambres** : **t\_af\_ins\_admission** ajuste la disponibilité des chambres après chaque admission.
- **Archivage des données médicales** : **t\_af\_ins\_historique\_medical** transfère automatiquement les consultations et interventions vers l'historique pour éviter la redondance.
- **Validation des factures** : **t\_bf\_ins\_facturation** vérifie la cohérence des montants et des dates de facturation.

Ces mécanismes renforcent la fiabilité et la sécurité du système.

### 5.3. Programmation et Planification

Les procédures de gestion des flux hospitaliers optimisent les ressources :

- **Programmation des interventions** : **p\_programmation\_intervention\_chirurgicale** priorise les cas urgents en tenant compte des disponibilités et spécialités des chirurgiens.
- **Prise de rendez-vous** : **p\_prise\_rendez\_vous** planifie une consultation et ajoute le patient à la base si nécessaire.
- **Détails des patients** : **p\_details\_patient** centralise toutes les informations médicales d'un patient pour un accès simplifié.

Ces outils améliorent la gestion des soins et des ressources.

### 5.4. Gestion des Factures et Rapports

Les fonctions et procédures suivantes assurent une gestion efficace des données financières et analytiques :

- **Calcul des factures** : **f\_calcul\_facture** applique des réductions pour les patients assurés et génère le montant total dû.
- **Recommandation des médecins** : **f\_recommandation** évalue les performances des médecins en fonction de leurs taux de réussite.

- **Rapports statistiques** : `p_generer_rapport` compile des données clés comme le taux d'occupation des chambres et les statistiques financières pour une période donnée. Ces outils fournissent des analyses détaillées et des indicateurs de performance.

### 5.5. Événements Automatisés

Des tâches périodiques automatisées augmentent la fiabilité du système :

- **Sauvegarde de la base** : `p_sauvegarde_base` génère des sauvegardes régulières pour prévenir les pertes de données.
- **Archivage des données** : `p_peuplement_historique_medical` transfère les données obsolètes vers l'historique, maintenant la base active performante.

Ces événements automatisés simplifient la gestion et renforcent la sécurité des données.

## 6. Perspectives d'Évolution

Pour améliorer et enrichir notre base de données hospitalière, plusieurs pistes d'évolution peuvent être envisagées. Par exemple, la création d'une table dédiée à la gestion des équipements médicaux pourrait permettre un suivi précis de l'utilisation, de l'entretien et des disponibilités des appareils. Une autre idée serait d'ajouter une table pour le suivi des prescriptions médicales, afin d'inclure la gestion des ordonnances et des traitements associés à chaque patient. De plus, l'intégration d'un système de journalisation des modifications, via une table d'historique des actions utilisateurs, offrirait une traçabilité complète des opérations effectuées sur la base. Ces évolutions pourraient renforcer la gestion globale des données et mieux répondre aux besoins d'un environnement hospitalier dynamique.

## 7. Conclusion

Dans ce rapport, nous avons présenté la conception et la mise en œuvre d'une base de données pour la gestion des informations dans un hôpital. Nous avons détaillé les différentes entités et interactions présentes dans cet environnement, ainsi que les contraintes et les relations entre les tables pour garantir l'intégrité et la cohérence des données.

Cette base de données offre une solution robuste et efficace pour la gestion des informations hospitalières, permettant d'améliorer la qualité des soins et l'efficacité des opérations. Les perspectives d'amélioration incluent l'intégration de nouvelles fonctionnalités, telles que la gestion des rendez-vous en ligne et l'analyse prédictive des données médicales pour anticiper les besoins des patients.

Nous tenons à exprimer notre gratitude à notre professeur pour son soutien et ses conseils tout au long de ce projet. Son expertise et son dévouement ont été essentiels à la réussite de ce travail.