# SAE 2-08. Rapport : Contrôle d'accès RBAC sur une BD


Réalisé par :

Pap Moctar Gaye
El Hadj Abdoulaye Gangue
Serigne Fall

Enseignant :

Kevin Atighehchi


19/05/2025

Année universitaire 2024/2025

# Introduction

Imaginez une infirmière en oncologie qui, pour préparer une chimiothérapie urgente, doit consulter en urgence le dossier d'un patient. Mais elle n'a besoin que d'une information : ses allergies médicamenteuses pas son historique psychiatrique. Ce genre de situation résume bien le dilemme auquel sont confrontés les établissements de santé : comment garantir un accès rapide aux informations sensibles, tout en protégeant la vie privée des patients ?

Les cyberattaques dans le milieu hospitalier ne sont plus des exceptions. En 2020, le CHU de Rouen a vu ses services paralysés, causant l'annulation de centaines d'opérations. À l'heure où le RGPD impose l'anonymisation des données tout en permettant leur utilisation à des fins de soin ou de recherche, la mise en œuvre d'une sécurité granulaire devient essentielle.

Dans ce projet, nous avons conçu un système de gestion des droits d'accès basé sur PostgreSQL, en mettant en œuvre le modèle RBAC (Role-Based Access Control) dans le contexte d'une base de données hospitalière. Notre objectif : protéger les données sensibles sans bloquer les soins. Notre solution repose sur l'idée d'**anonymisation contextuelle**. Un chercheur verra des tranches d'âge et des codes de diagnostic, jamais des noms. Un médecin en urgence pourra, grâce à une authentification renforcée, accéder temporairement à davantage de données. Toutes les actions sont tracées pour garantir la responsabilité.

# Organisation du Groupe

Notre groupe est composé de trois étudiants, chacun avec un rôle défini :

**Fall Serigne :** Responsable technique : installation et paramétrage de PostgreSQL

**Gaye Pap** Moctar : Responsable sécurité : modélisation des rôles et privilèges

**Gangue El hadj Abdoulaye** : Responsable documentation : rédaction du rapport et tests

# Développement

## 1.Schéma de la Base de Données

### *Description des Tables*

| Table | Description | Clés & Relations |
|---|---|---|
| Physician | Médecins (ID employé, nom, poste, SSN). | EmployeeID (PK). |
| Département | Services hospitaliers (ID, nom, responsable). | Head (FK vers Physician). |
| Affiliated_With | Affiliation des médecins aux départements. | Clé composite (Physician, Department), FK vers Physician et Department. |
| Medica Procédure | Procédures médicales (code, nom, coût). | Code (PK). |
| Trained_In | Certifications des médecins pour des procédures. | Clé composite (Physician, Treatment), FK vers Physician et MedicalProcedure. |
| Patient | Patients (SSN, nom, adresse, médecin traitant). | PCP (FK vers Physician). |
| Appointment | Rendez-vous médicaux (ID, patient, infirmier, médecin, créneau, salle). | FK vers Patient, Nurse, et Physician. |
| Prescribes | Prescriptions (médecin, patient, médicament, dose, date). | Clé composite (Physician, Patient, Medication, Date), FK multiples. |
| Stay | Séjours hospitaliers (ID, patient, chambre, dates). | FK vers Patient et Room. |
| Undergoes | Historique des procédures subies par les patients. | Clé composite (Patient, Procedure, Stay, Date), FK multiples. |

### *Relations Clés*

- Un **médecin** (Physician) peut être responsable d'un **département** (Department).
- Un **patient** (Patient) est lié à un médecin traitant (PCP).
- Une **prescription** (Prescribes) associe un médecin, un patient, et un médicament.

## 2. Sécurité RBAC (Role-Based Access Control)

***Structure des Rôles***

| Rôle | Utilisateur Associé | Privilèges |
|---|---|---|
| admin | user_admin | Accès complet à toutes les tables (GRANT ALL). |
| medecin | user_medecin | Lecture/écriture sur Patient, Appointment, Prescribes. |
| infirmier | user_infirmier | Lecture/écriture sur Patient, On_Call, Stay. |
| secretaire | user_secretaire | Lecture/écriture sur Patient, Appointment; Lecture sur Physician. |
| chercheur | user_chercheur | Lecture des vues anonymisées (vw_undergoes_anonymized, vw_stay_stats). |

***Hiérarchie des Rôles***

```
admin
 ├── medecin
 │    └── infirmier
 ├── secretaire
 └── chercheur
```

***Scripts d'Attribution***

```sql
-- Attribution des rôles
GRANT medecin TO user_medecin;
GRANT infirmier TO user_infirmier;

-- Héritage des privilèges
GRANT infirmier TO medecin;
GRANT medecin, secretaire, chercheur TO admin;
```

## 3. Anonymisation des Données pour la Recherche

***Problème Initial***

- Le rôle chercheur avait accès direct aux tables Undergoes, MedicalProcedure, et Stay, exposant les SSN des patients.

***Solution : Vues Sécurisées***

```sql
-- Vue anonymisée des procédures médicales
CREATE VIEW vw_undergoes_anonymized AS
SELECT
  u.Procedure AS CodeProcedure,
```

```
    mp.Name AS NomProcedure,
    mp.Cost AS Cout,
    s.Room AS Chambre,
    s.Start AS DebutSejour,
    s."End" AS FinSejour
FROM Undergoes u
JOIN MedicalProcedure mp ON u.Procedure = mp.Code
JOIN Stay s ON u.Stay = s.StayID;

-- Vue statistique sur les séjours
CREATE VIEW vw_stay_stats AS
SELECT
    Room AS Chambre,
    COUNT(*) AS NombreSejours,
    ROUND(AVG(EXTRACT(EPOCH FROM ("End" - Start)) / 86400, 1) AS DureeMoyenneJours
FROM Stay
GROUP BY Room;

-- Attribution au chercheur
GRANT SELECT ON vw_undergoes_anonymized, vw_stay_stats TO chercheur;
```

**4. Optimisation et Validation**

***Indexation***

Pour accélérer les requêtes sur les vues anonymisées :

sql

```
CREATE INDEX idx_undergoes_procedure ON Undergoes(Procedure);
CREATE INDEX idx_stay_room ON Stay(Room);
```

***Scénarios de Test***

1. **Accès Médecin** :

sql

```
-- user_medecin peut lire/mettre à jour les dossiers patients
SELECT * FROM Patient WHERE SSN = 123456789;
UPDATE Patient SET Address = 'Nouvelle Adresse' WHERE SSN = 123456789;
```

2. **Accès Chercheur** :

sql

```
-- user_chercheur ne voit que les données anonymisées
SELECT * FROM vw_undergoes_anonymized WHERE CodeProcedure = 100;
SELECT * FROM vw_stay_stats WHERE DureeMoyenneJours > 5;
```

3. **Contrôle d'Accès** :

sql

```
-- Tentative d'accès non autorisé (rejetée)
SELECT * FROM Undergoes;  -- Erreur : Permission denied pour le rôle "chercheur"
```

## 5. scénario d'accès à la BD

```
Hopital=# \q

C:\Program Files\PostgreSQL\17\pgAdmin 4\runtime>psql  -U user_infirmier -d Hopital -h localhost -W
Password:

psql (17.4)
WARNING: Console code page (850) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.
Hopital=> INSERT INTO Stay (StayID, Patient, Room, Start, "End")VALUES (99, 100000001, 101, '2025-05-01 10:00', '2025-05-05 12:00');
INSERT 0 1
Hopital=> SELECT * FROM Stay;
 stayid |  patient  | room |        start        |         End
--------+-----------+------+---------------------+---------------------
   3215 | 100000001 |  111 | 2008-05-01 00:00:00 | 2008-05-04 00:00:00
   3216 | 100000003 |  123 | 2008-05-03 00:00:00 | 2008-05-14 00:00:00
   3217 | 100000004 |  112 | 2008-05-02 00:00:00 | 2008-05-03 00:00:00
     99 | 100000001 |  101 | 2025-05-01 10:00:00 | 2025-05-05 12:00:00
(4 rows)
Hopital=> INSERT INTO Stay (StayID, Patient, Room, Start, "End")VALUES (100, 999999999, 101, '2025-05-01 10:00', '2025-05-05 12:00');
ERREUR:  une instruction insert ou update sur la table « stay » viole la contrainte de clé
étrangère « fk_stay_patient »
DETAIL:  La clé (patient)=(999999999) n'est pas présente dans la table « patient ».
Hopital=>
```

**Conclusion 1 :**

En tant qu'utilisateur user_infirmier, j'ai pu insérer un séjour pour un patient déjà existant dans la base, conformément aux droits accordés. La tentative d'ajout d'un séjour pour un patient inexistant a échoué à cause de la contrainte de clé étrangère sur le champ Patient de la table Stay.

Ce test montre que les droits sont correctement restreints : l'infirmier peut effectuer des actions autorisées (comme insérer un séjour), mais ne peut pas contourner les règles d'intégrité ou accéder à des données non permises

```
Hopital-> \q
C:\Program Files\PostgreSQL\17\pgAdmin 4\runtime>psql  -U user_chercheur -d Hopital -h localhost -W
Password:

psql (17.4)
WARNING: Console code page (850) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

Hopital=> SELECT * FROM vw_undergoes_anonymized LIMIT 5;
 procedure | stay |        date         | physician | assistingnurse |         procedurename          | cost | room |      staystart      |       stayend
-----------+------+---------------------+-----------+----------------+--------------------------------+------+------+---------------------+---------------------
         6 | 3215 | 2008-05-02 00:00:00 |         3 |            101 | Reversible Pancreomyoplasty    | 5600 |  111 | 2008-05-01 00:00:00 | 2008-05-04 00:00:00
         2 | 3215 | 2008-05-03 00:00:00 |         7 |            101 | Obtuse Pyloric Recombobulation | 3750 |  111 | 2008-05-01 00:00:00 | 2008-05-04 00:00:00
         1 | 3217 | 2008-05-07 00:00:00 |         3 |            102 | Reverse Rhinopodoplasty        | 1500 |  112 | 2008-05-02 00:00:00 | 2008-05-03 00:00:00
         5 | 3217 | 2008-05-09 00:00:00 |         6 |                | Obfuscated Dermogastrotomy     | 4899 |  112 | 2008-05-02 00:00:00 | 2008-05-03 00:00:00
         7 | 3217 | 2008-05-10 00:00:00 |         7 |            101 | Follicular Demiectomy          |   25 |  112 | 2008-05-02 00:00:00 | 2008-05-03 00:00:00
(5 rows)


Hopital=> SELECT * FROM vw_stay_stats;
 room | nbstays | avgstaydays
------+---------+-------------
  101 |       1 |         4.1
  112 |       1 |         1.0
  111 |       1 |         3.0
  123 |       1 |        11.0
(4 rows)


Hopital=> SELECT * FROM Patient LIMIT 5;
ERREUR:  droit refusé pour la table patient
Hopital=>
```

**Conclusion 2 :**

En tant qu'utilisateur chercheur, j'ai pu consulter les vues anonymisées vw_undergoes_anonymized et vw_stay_stats, conformément aux droits de lecture qui me sont accordés. Toute tentative d'accès direct à des tables sensibles comme Patient a échoué, en raison de l'absence de privilèges suffisants.

Ce test montre que les droits sont correctement restreints : le chercheur peut accéder uniquement à des données anonymisées prévues pour l'analyse, sans possibilité d'accéder à des informations personnelles ou d'effectuer des modifications dans la base

# Conclusion

Imaginez une salle d'urgence à 2 h du matin. Un médecin ouvre son poste, l'écran s'allume et affiche en quelques secondes l'historique médical d'un patient en arrêt cardiaque. Ce moment critique montre l'enjeu de notre projet : fournir des informations rapides et fiables sans bloquer le soin.

Nous avons mis en place un modèle RBAC simple et clair : chaque rôle (infirmier, médecin, secrétaire, chercheur, administrateur) dispose uniquement des droits nécessaires. Grâce aux vues anonymisées, les chercheurs accèdent à des données agrégées sans jamais voir le SSN ou le nom des patients. Les logs et les index assurent une surveillance discrète, détectent les accès suspects et optimisent les performances.

Ce travail nous a appris que la sécurité est un équilibre : trop de barrières, les soignants perdent du temps ; pas assez, la confidentialité est compromise. L'essentiel est de rester à l'écoute des besoins, d'ajuster les droits et de tester en conditions réelles.

En conclusion, ce projet est avant tout un engagement pour la confiance et le respect des patients.