



**POLYTECHNIQUE
MONTRÉAL**

**UNIVERSITÉ
D'INGÉNIERIE**

INF3710 –Bases de données

Automne 2023

TP No. 5

**Groupe vendredi B2
2174916– Arman Lidder
2183376 – Hamza Boukaftane**

Soumis à : Joe Abdo

2 décembre 2023

1. Requêtes SQL

1) Affichez la spécialité du médecin avec l'identifiant 3?

```
SELECT specialite  
FROM medecins  
WHERE idmedecin = 3
```

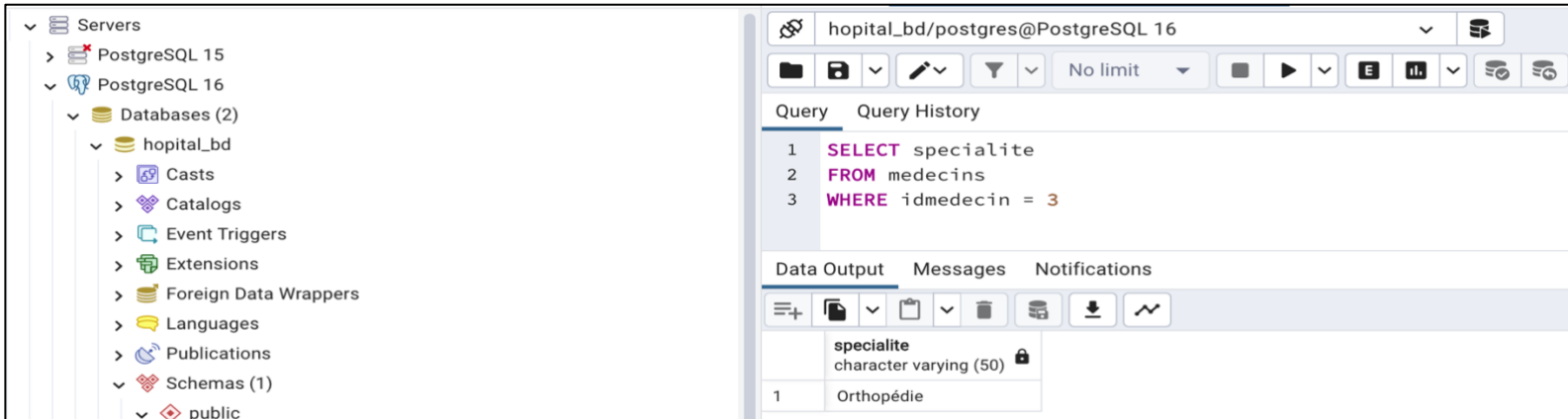


Figure 1. Spécialité du médecin avec l'identifiant 3.

2) Affichez les noms et prénoms des médecins qui ont plus de 10 ans d'expérience?

```
SELECT nom, prenom  
FROM medecins  
WHERE anneesexperience > 10
```

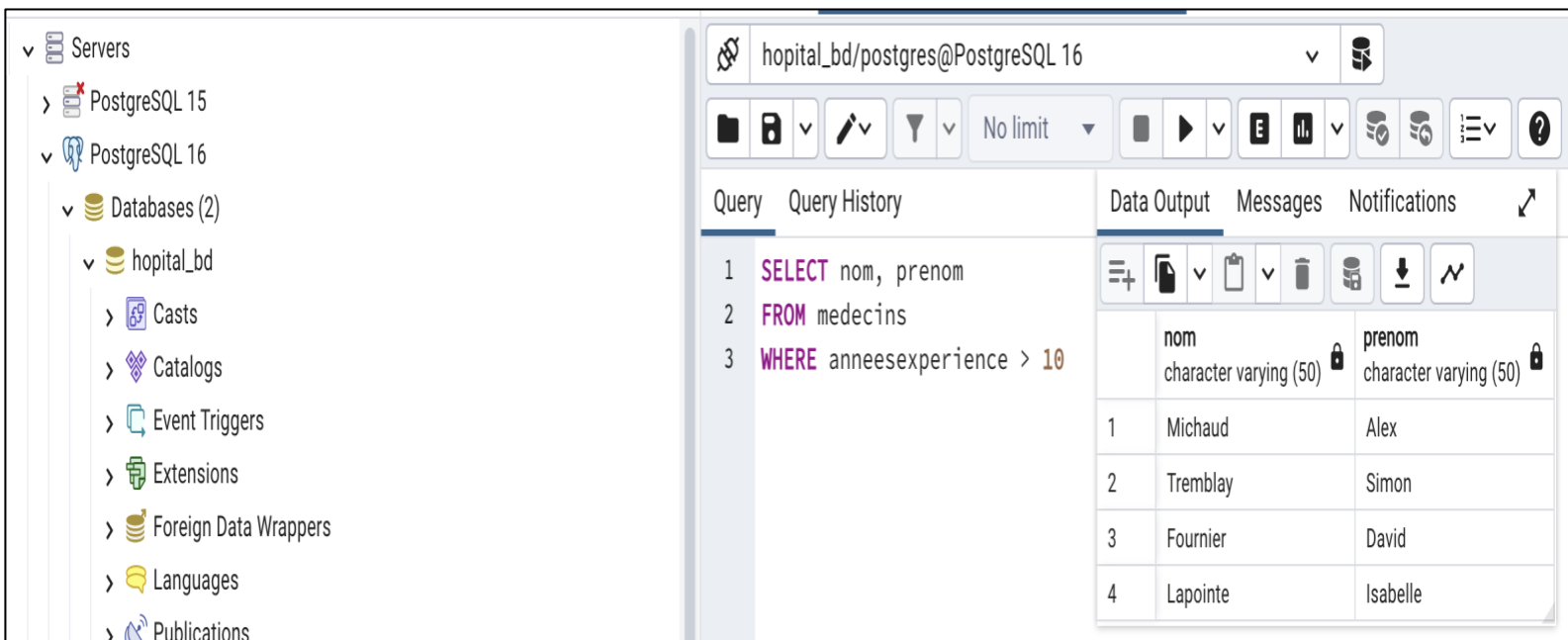


Figure 2. Nom et prénom de médecins avec plus de 10 ans expérience.

3) Affichez le nombre total d'examens qui ont été enregistrés dans la base de données?

```
SELECT COUNT(*) as nombretotalexamen
```

```
FROM examens
```

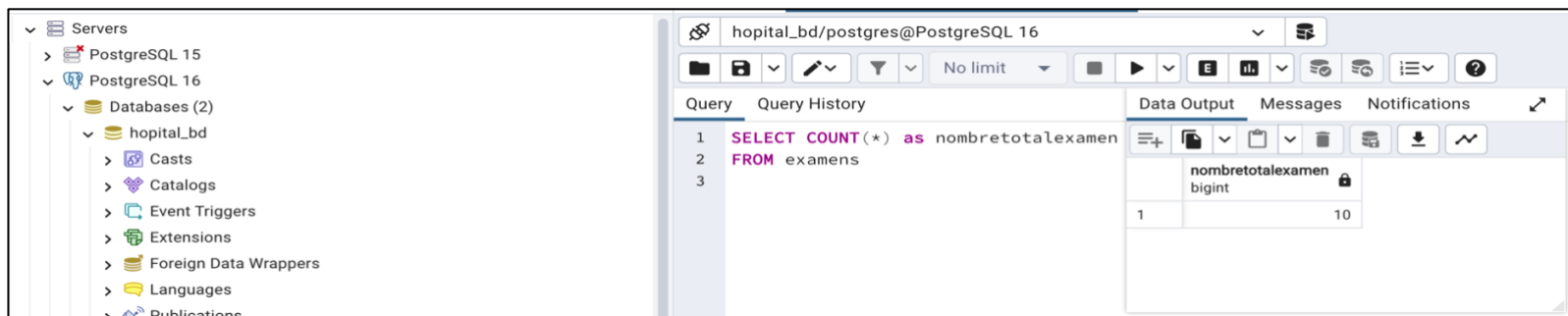


Figure 3. Le nombre total d'examen enregistrée dans hopital_bd.

4) Afficher le nom de l'examen le plus coûteux, son prix ainsi que la dernière date à laquelle il a été effectué?

```
WITH maxCoutExamen AS (
```

```
    SELECT e.nomexamen, e.cout, pe.dateexamen
```

```
    FROM examens e
```

```
    JOIN patientexamens pe ON e.idexamen = pe.idexamen
```

```
    WHERE e.cout = (SELECT MAX(cout) FROM examens))
```

```
SELECT *
```

```
FROM maxCoutExamen
```

```
WHERE dateexamen = (SELECT MAX(dateexamen) FROM maxCoutExamen)
```

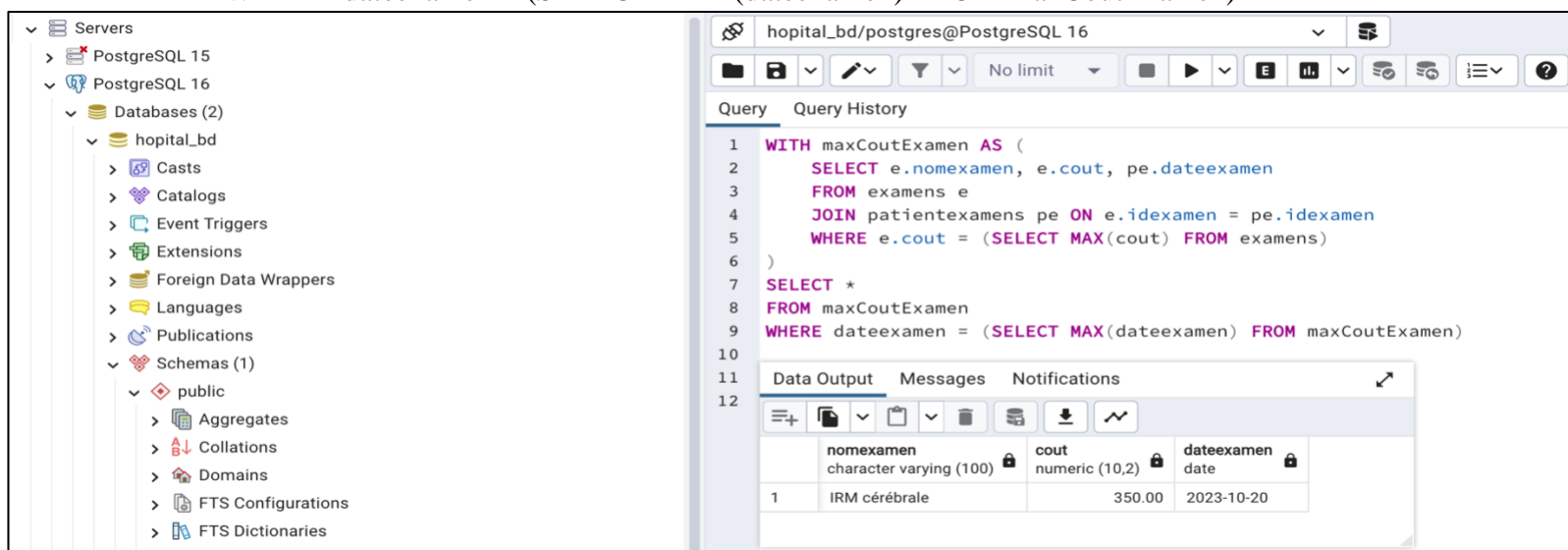
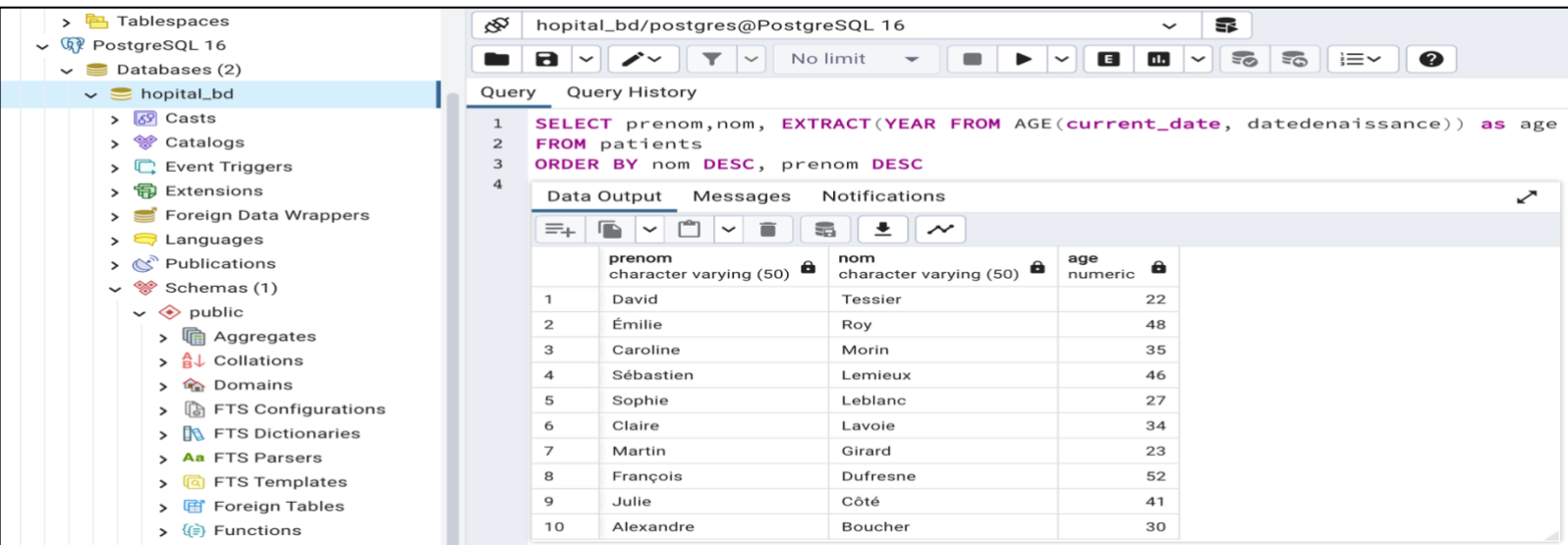


Figure 4. Le nom, le coût et la dernière date de l'examen le plus coûteux.

5) Affichez les prénoms et noms des patients avec leur âge, en ordre décroissant de l'alphabet?

```
SELECT
    prenom,
    nom,
    EXTRACT(YEAR FROM AGE(current_date, datedenaissance)) as age
FROM patients
ORDER BY nom DESC, prenom DESC
```



The screenshot shows the PostgreSQL 16 interface with the 'hopital_bd' database selected. The query editor contains the following SQL:

```
1 SELECT prenom,nom, EXTRACT(YEAR FROM AGE(current_date, datedenaissance)) as age
2 FROM patients
3 ORDER BY nom DESC, prenom DESC
4
```

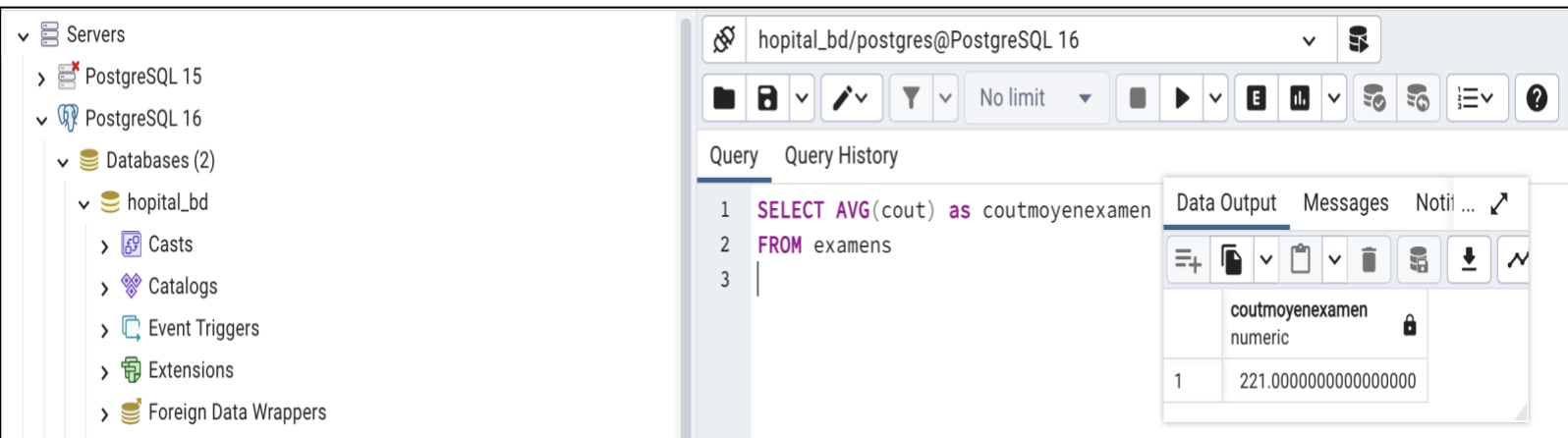
The 'Data Output' tab shows the results of the query, sorted by last name (nom) in descending order, then by first name (prenom) in descending order. The results are as follows:

	prenom character varying (50)	nom character varying (50)	age numeric
1	David	Tessier	22
2	Émilie	Roy	48
3	Caroline	Morin	35
4	Sébastien	Lemieux	46
5	Sophie	Leblanc	27
6	Claire	Lavoie	34
7	Martin	Girard	23
8	François	Dufresne	52
9	Julie	Côté	41
10	Alexandre	Boucher	30

Figure 5. Prénoms, noms et âges des patients triés en ordre décroissant de nom.

6) Affichez le prix moyen du coût des examens médicaux?

```
SELECT AVG(cout) as coutmoyenexamen
FROM examens
```



The screenshot shows the PostgreSQL 16 interface with the 'hopital_bd' database selected. The query editor contains the following SQL:

```
1 SELECT AVG(cout) as coutmoyenexamen
2 FROM examens
3
```

The 'Data Output' tab shows the results of the query, which is a single row representing the average cost of medical exams:

	coutmoyenexamen numeric
1	221.0000000000000000

Figure 6. Prix moyen du coût des examens médicaux.

7) Affichez les noms et prénoms de chaque médecin avec le nombre de rendez-vous qu'ils ont eus chacun depuis le 21 septembre 2023?

```
WITH rdvDepuisDate AS (
    SELECT idmedecin
    FROM rendezvous
    WHERE daterendezvous >= '2023-09-21'
)
SELECT nom, prenom, COUNT(rdv.idmedecin) as nbrendezvous
FROM medecins md
LEFT JOIN rdvDepuisDate rdv ON md.idmedecin = rdv.idmedecin
GROUP BY nom, prenom
```

The screenshot shows a PostgreSQL client interface with a query editor and a results pane. The query editor contains the following SQL code:

```
1 WITH rdvDepuisDate AS (
2     SELECT idmedecin
3     FROM rendezvous
4     WHERE daterendezvous >= '2023-09-21'
5 )
6 SELECT nom, prenom, COUNT(rdv.idmedecin) as nbrendezvous
7 FROM medecins md
8 LEFT JOIN rdvDepuisDate rdv ON md.idmedecin = rdv.idmedecin
9 GROUP BY nom, prenom
```

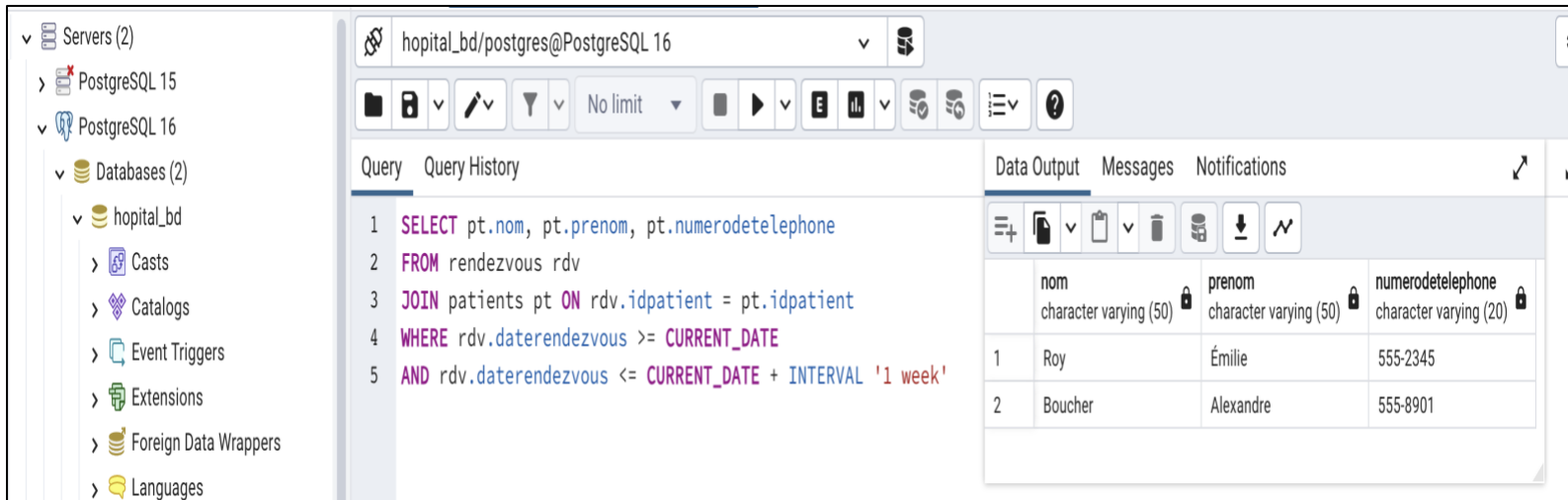
The results pane displays the following data:

	nom	prenom	nbrendezvous
	character varying (50)	character varying (50)	bigint
1	Beaulieu	Audrey	2
2	Martel	François	2
3	Rousseau	Marie	1
4	Lapointe	Isabelle	2
5	Lemelin	Philippe	1
6	Michaud	Alex	2
7	Fournier	David	2
8	Bélanger	Valérie	2
9	Tremblay	Simon	2
10	Gagné	Nathalie	2

Figure 7. Prénoms, nom et nombre de rendez-vous depuis le 21 septembre 2023.

8) Affichez les noms, prénoms et numéro de téléphone des patients qui ont des rendez-vous la semaine prochaine?

```
SELECT pt.nom, pt.prenom, pt.numerodetelephone
FROM rendezvous rdv
JOIN patients pt ON rdv.idpatient = pt.idpatient
WHERE rdv.daterendezvous >= CURRENT_DATE
AND rdv.daterendezvous <= CURRENT_DATE + INTERVAL '1 week'
```



The screenshot shows a PostgreSQL client interface with a query editor on the left and a results pane on the right. The query editor contains the following SQL query:

```
1 SELECT pt.nom, pt.prenom, pt.numerodetelephone
2 FROM rendezvous rdv
3 JOIN patients pt ON rdv.idpatient = pt.idpatient
4 WHERE rdv.daterendezvous >= CURRENT_DATE
5 AND rdv.daterendezvous <= CURRENT_DATE + INTERVAL '1 week'
```

The results pane displays the output of the query as a table with three columns: nom, prenom, and numerodetelephone. The table contains two rows of data:

	nom	prenom	numerodetelephone
1	Roy	Émilie	555-2345
2	Boucher	Alexandre	555-8901

Figure 8. Nom, prénom et numéro de téléphone des patients ayant rendez-vous la semaine prochaine (Requête exécuté le 19 novembre 2023).

9) Affichez le nom, prénom ainsi que le coût total des examens passés par le patient avec l'identifiant 4 dans un service dont le nom débute par la lettre 'C'?

```
WITH examenServiceC AS (
    SELECT *
    FROM examens e
    JOIN patientexamens pe ON e.idexamen = pe.idexamen
    JOIN services sv ON e.idservice = sv.idservice
    WHERE sv.nomservice LIKE 'C%'),
specificPatient AS (
    SELECT *
    FROM patients
    WHERE idpatient = 4)
```

```

SELECT nom, prenom, COALESCE(SUM(esc.cout),0) as montantserviceC
FROM specificPatient pt
LEFT JOIN examenServiceC esc ON pt.idpatient = esc.idpatient
GROUP BY nom, prenom

```

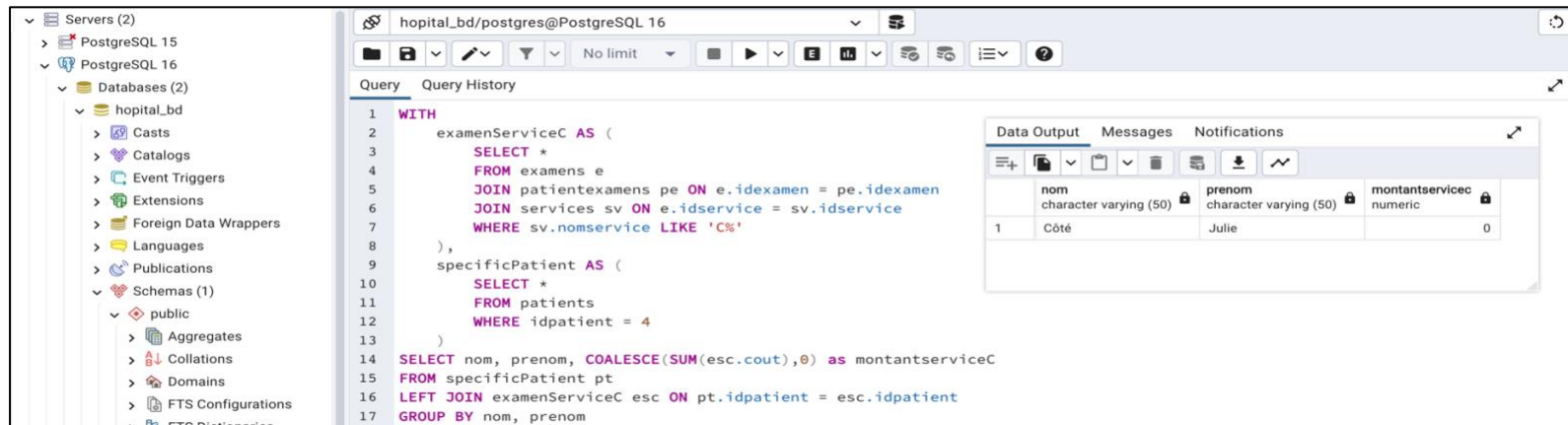


Figure 9. Nom, prénom et montant dépensé pour les examens des services commençant par C du patient ayant l'identifiant 4.

10) Afficher les nom et prénoms des patients qui ont consulté le médecin avec le plus d'expérience?

```

SELECT pt.nom, pt.prenom
FROM rendezvous rdv
JOIN patients pt ON rdv.idpatient = pt.idpatient
JOIN medecins md ON rdv.idmedecin = md.idmedecin
WHERE md.anneesexperience = (SELECT MAX(anneesexperience) FROM medecins)
AND rdv.daterendezvous <= CURRENT_DATE

```

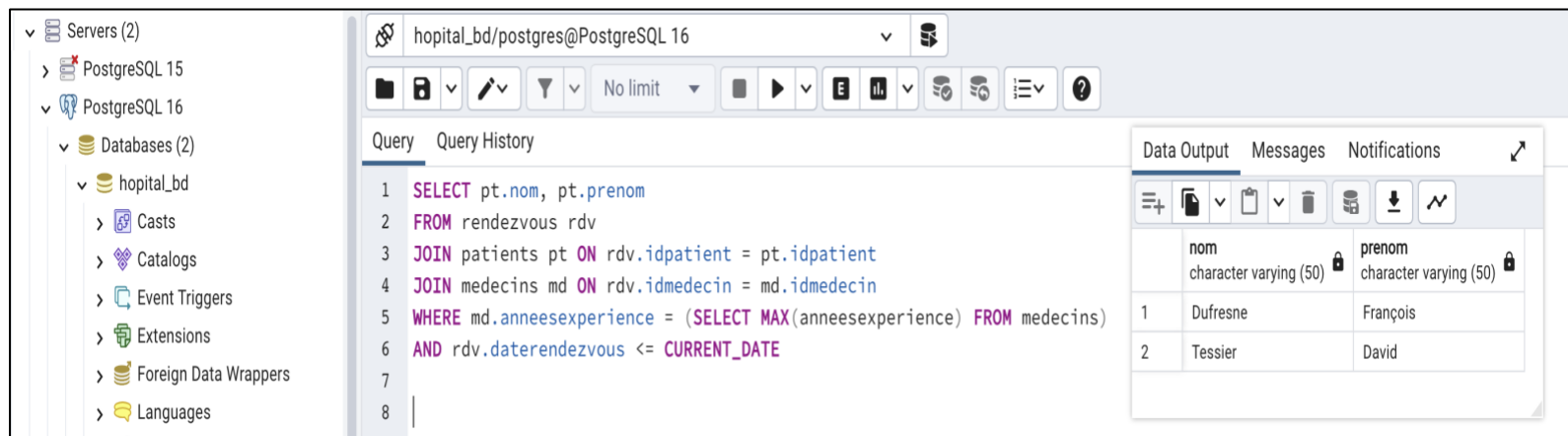


Figure 10. Nom et prénom des patients ayant consulté le médecin le plus expérimenté.

11) Affichez les noms et prénoms des patients qui ont passé des examens médicaux avec un coût supérieur à 200\$ par examen et un cout total de tous les examens dépassant 2000\$?

```
SELECT pt.nom, pt.prenom
FROM patients pt
JOIN patientexamens pe ON pt.idpatient = pe.idpatient
JOIN examens e ON pe.idexamen = e.idexamen
GROUP BY nom, prenom
HAVING MIN(e.cout) > 200 AND SUM(e.cout) > 2000
```

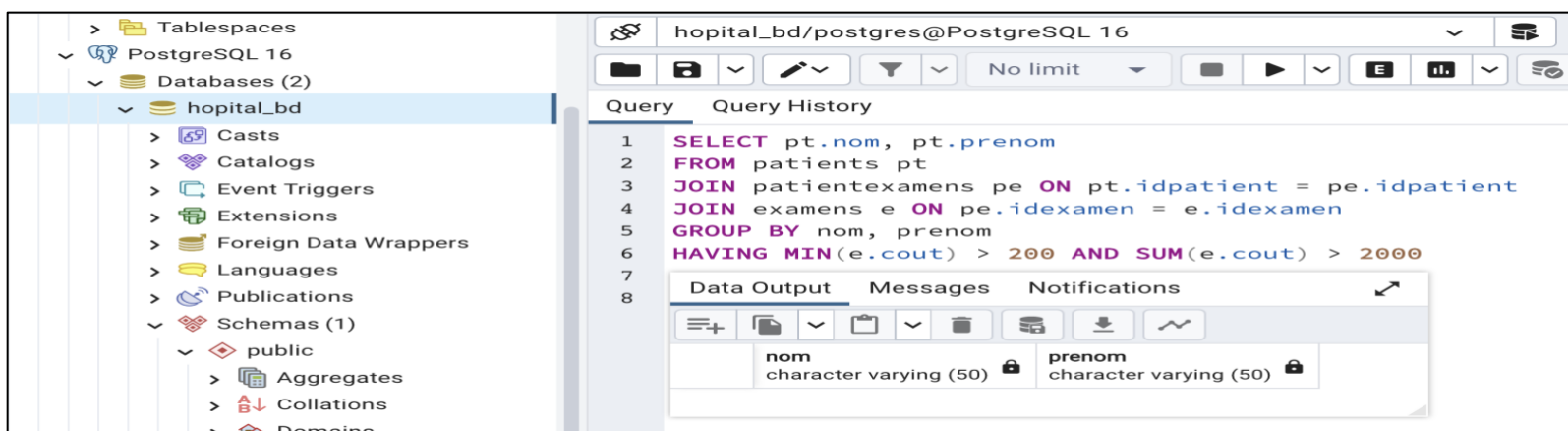


Figure 11. Nom et prénom des patients payant plus que 200\$ par examen et plus de 2000\$ au total.

12) Afficher le nom et prénom du médecin ayant le plus grand nombre de patients différents consultés durant les 6 derniers mois?

WITH

rdvDernierSixMoix as (

SELECT *

FROM rendezvous rdv

WHERE rdv.daterendezvous <= CURRENT_DATE

AND rdv.daterendezvous >= CURRENT_DATE - INTERVAL '6 months'),

patientParMedecin as (

SELECT idmedecin,

COUNT (DISTINCT idpatient) AS nbPatients

FROM rdvDernierSixMoix


```

GROUP BY idmedecin),
maxPatientParMedecin as (
    SELECT MAX(nbPatients) as nbPatients
    FROM patientParMedecin),
maxIdMedecin as (
    SELECT *
    FROM patientParMedecin ppm
    JOIN maxPatientParMedecin mppm ON ppm.nbPatients = mppm.nbPatients)
SELECT md.nom, md.prenom
FROM medecins md
JOIN maxIdMedecin maxmed ON md.idmedecin = maxmed.idmedecin

```

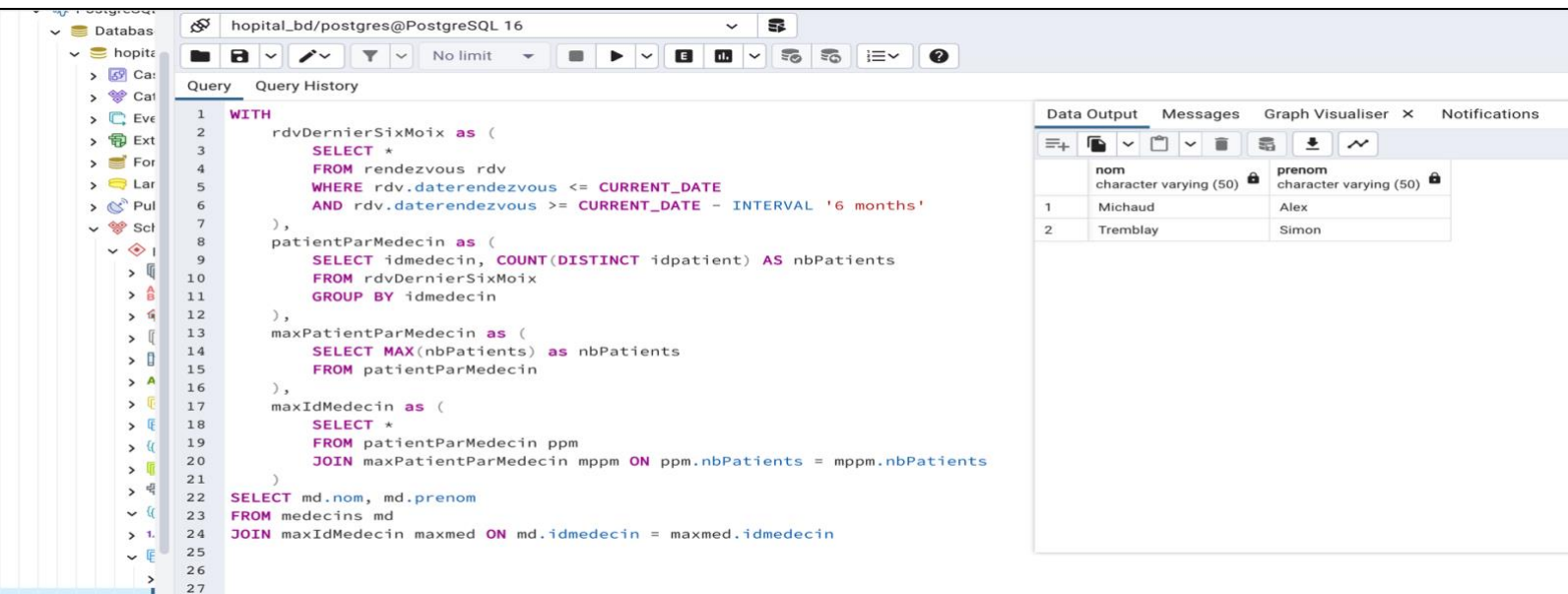


Figure 12. Nom et prénom des médecins ayant le plus grand nombre de patients différents les 6 derniers mois (requête effectué le 19 novembre 2023).

13) Affichez le prénom et le nom des médecins qui n'ont jamais eu de rendez-vous et n'ont effectué aucun examen (Pas de relation entre idMedecin et idExamen, donc je vais simplement chercher les medecins qui n'ont jamais eu de rendez-vous)?

```

SELECT med.prenom, med.nom
FROM medecins med
LEFT JOIN rendezvous rdv ON med.idmedecin = rdv.idmedecin
WHERE rdv.idrendezvous IS NULL AND med.idmedecin IS NULL;

```

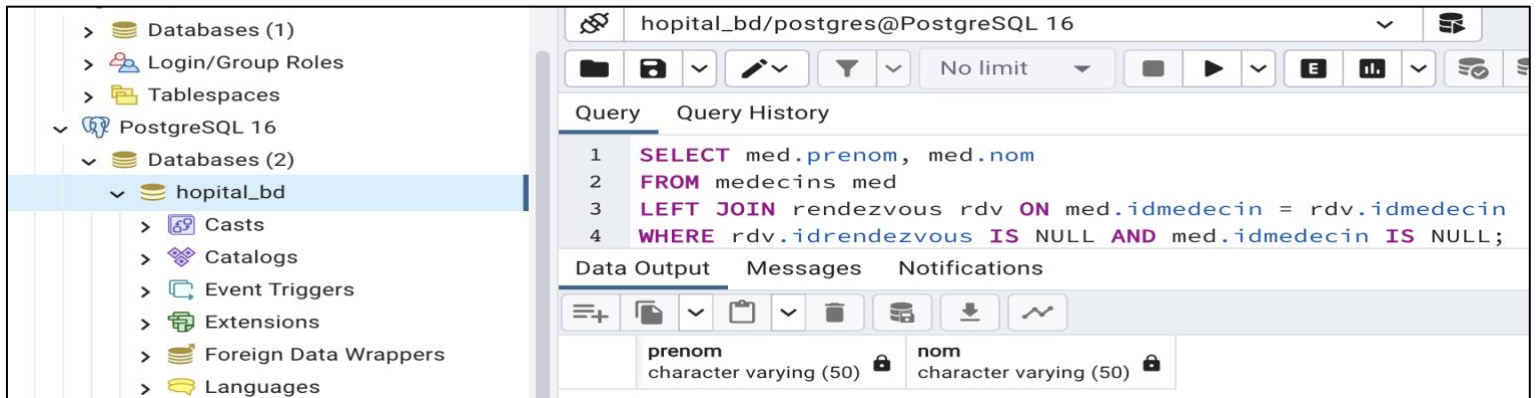


Figure 13. Prénom et nom des médecins n'ayant jamais eu de rendez-vous.

14) Affichez le nom et prénom du patient qui a passé le plus grand nombre d'examens sans obtenir de résultat "Résultats normaux"? (Si le nombre max est le même pour plusieurs patients on les affiche tous)

WITH

examensNonNormaux as (

SELECT *, COUNT(idpatient) as nbexamennonnormal

FROM patientexamens

WHERE resultat NOT LIKE 'Résultats normaux'

GROUP BY idpatientexamen),

maxNonNormaux as (

SELECT MAX (nbexamennonnormal) as nbexamennonnormal

FROM examensNonNormaux) ,

maxIdPatient as (

SELECT enn.idpatient

FROM examensNonNormaux enn

JOIN maxNonNormaux mnn

ON enn.nbexamennonnormal = mnn.nbexamennonnormal

)

SELECT pt.nom, pt.prenom

FROM patients pt

JOIN maxIdPatient mip ON pt.idpatient = mip.idpatient

The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```

1 WITH
2   examensNonNormaux as (
3     SELECT *, COUNT(idpatient) as nbexamennonnormal
4     FROM patientexamens
5     WHERE resultat NOT LIKE 'Résultats normaux'
6     GROUP BY idpatientexamen
7   ),
8   maxNonNormaux as (
9     SELECT MAX(nbexamennonnormal) as nbexamennonnormal
10    FROM examensNonNormaux
11  ),
12  maxIdPatient as (
13    SELECT enn.idpatient
14    FROM examensNonNormaux enn
15    JOIN maxNonNormaux mnn ON enn.nbexamennonnormal = mnn.nbexamennonnormal
16  )
17  SELECT pt.nom, pt.prenom
18  FROM patients pt
19  JOIN maxIdPatient mip ON pt.idpatient = mip.idpatient
20

```

The results pane on the right shows a table with two columns: **nom** and **prenom**. The data is as follows:

	nom	prenom
1	Dufresne	François
2	Côté	Julie
3	Leblanc	Sophie
4	Roy	Émilie
5	Girard	Martin
6	Tessier	David
7	Morin	Caroline
8	Lemieux	Sébastien
9	Boucher	Alexandre

Figure 14. Nom et prénom des patients ayant passé le plus grand temps sans obtenir de résultat.

15) Affichez les noms et prénoms des patients qui ont consulté un médecin de chaque service?

```

WITH patientService as (
    SELECT DISTINCT
        pt. idPatient,
        pt. prenom,
        pt. nom,
        med. idService
    FROM rendezvous rdv
    JOIN medecins med ON rdv.idMedecin = med.idMedecin
    JOIN patients pt ON rdv.idPatient = pt.idPatient
    JOIN services sv ON med.idService = sv.idService)

SELECT ps.nom, ps.prenom
FROM patientservice ps
GROUP BY ps. nom, ps. prenom
HAVING COUNT(DISTINCT ps.idService) = (SELECT COUNT(*) FROM services)

```

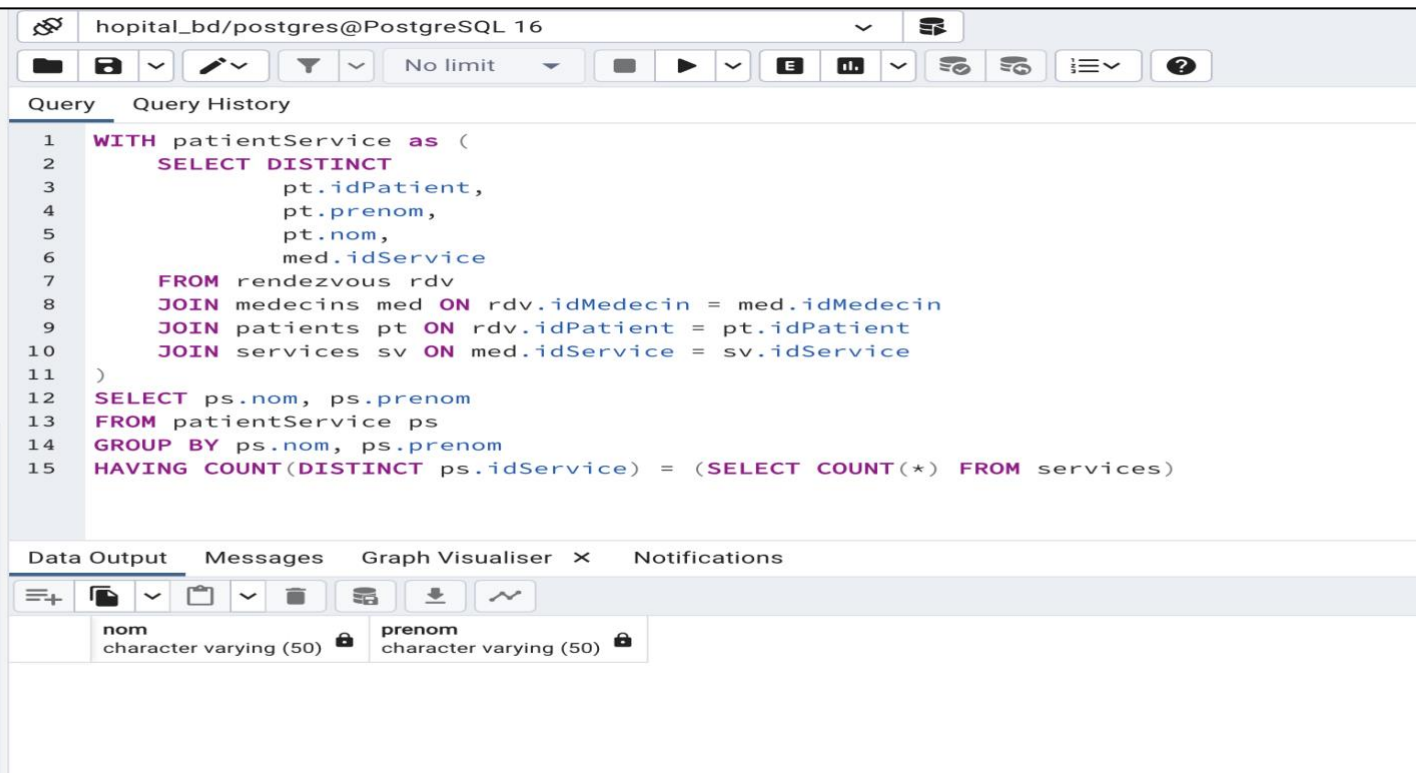


Figure 15. Nom et prénom des patients ayant consulté un médecin de chaque service.

16) Créez une vue 'V_TotalExamens' respectant les critères requis

CREATE VIEW V_TotalExamens AS

SELECT

pt.nom AS V_nom,

pt.prenom AS V_prenom,

COUNT(pe.idpatient) AS V_tot

FROM patientexamens pe

JOIN examens e ON pe.idexamen = e.idexamen

JOIN patients pt ON pe.idpatient = pt.idpatient

WHERE pt.adresse LIKE '%e__' OR pt.adresse LIKE '%o__'

GROUP BY pt.nom, pt.prenom

HAVING SUM(e.cout) > 500

ORDER BY pt.nom, COUNT(pe.idpatient) DESC

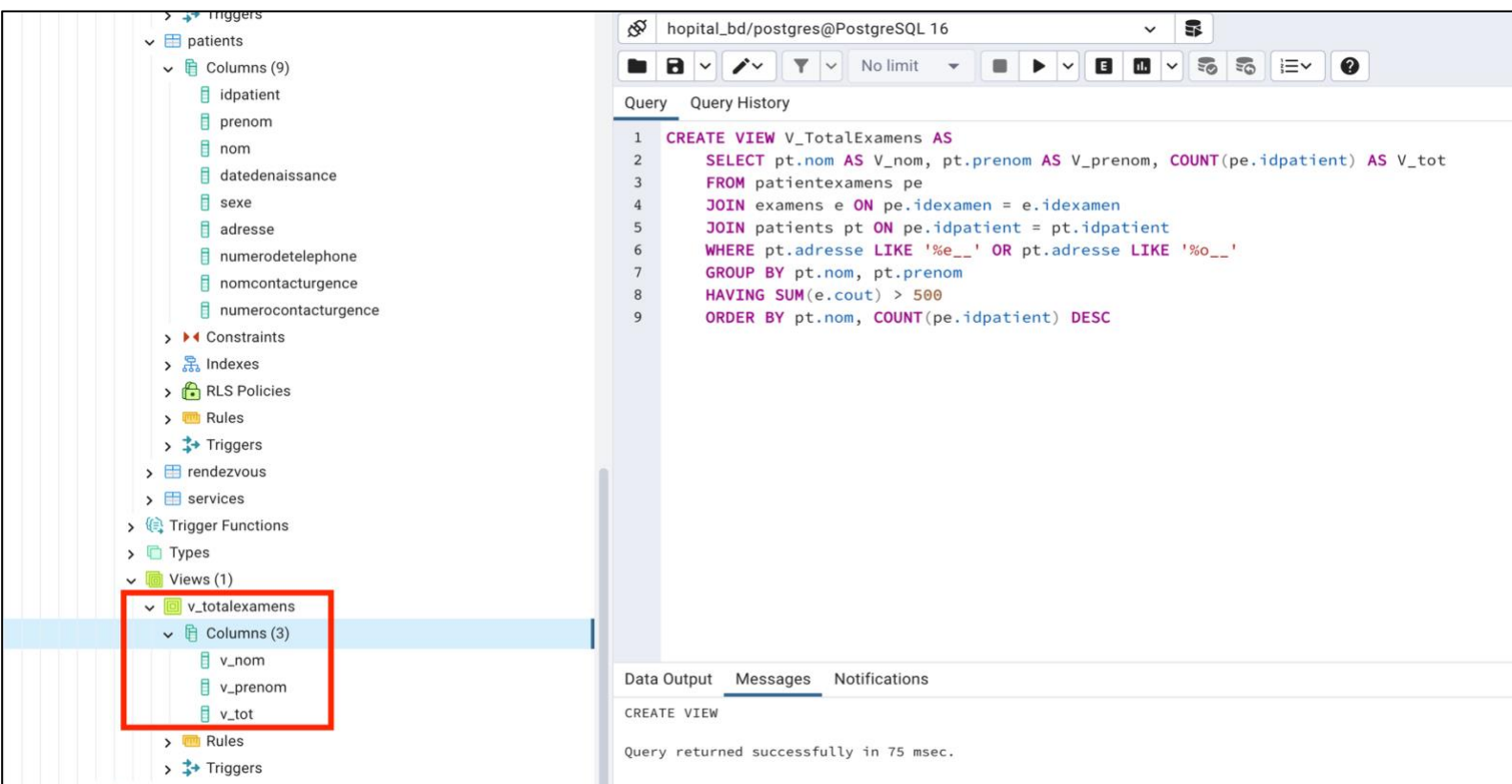


Figure 16. Création réussie de la vue V_TotalExamens à l'exécution de la requête SQL.

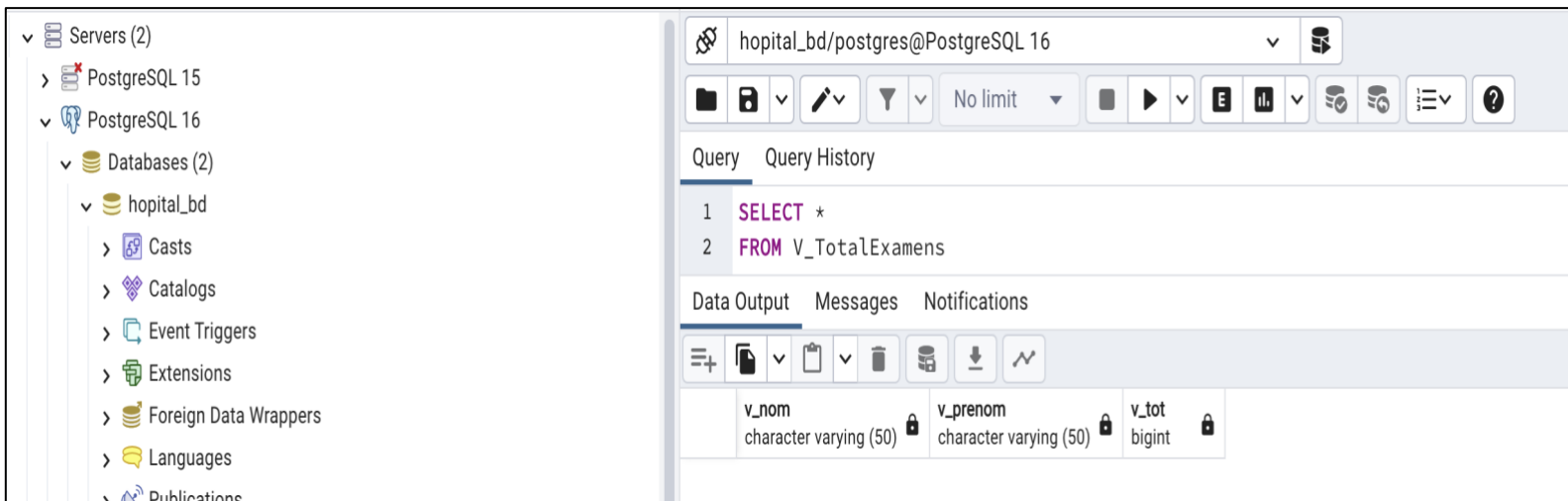


Figure 17. Affichage de la vue V_TotalExamens.

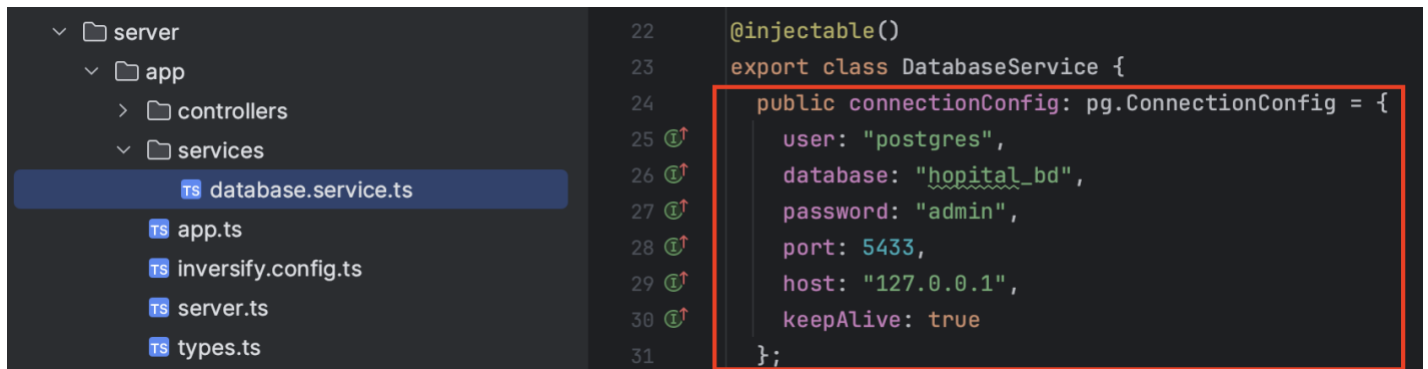
2. Application Web

***** Très important le tableau du site possède un scroll bar et npm start crée et popule hopital_db si besoin.**

Avant de lancer l'application web, veuillez suivre les instructions suivantes :

1- Veuillez d'abord implémenter un serveur postgres contenant une base de données nommé hopital_bd dans pg4admin.

2- Modifier les configurations du service de connexion à la base de données en fonction de vos propres configurations. (location : database.service.ts)



```
22 @injectable()
23 export class DatabaseService {
24     public connectionConfig: pg.ConnectionConfig = {
25         user: "postgres",
26         database: "hopital_bd",
27         password: "admin",
28         port: 5433,
29         host: "127.0.0.1",
30         keepAlive: true
31     };
}
```

Figure 18. Configuration actuelle de la connexion avec la base de données.

3- Exécuter les commandes npm install sur les dossiers sources clients et serveur.

4- Exécuter les commandes npm start sur les dossiers sources clients et serveur. Lors de l'exécution de npm start sur le serveur, le code créera les tables si elles n'existent pas et populera la base de données (sans conflit de clé).

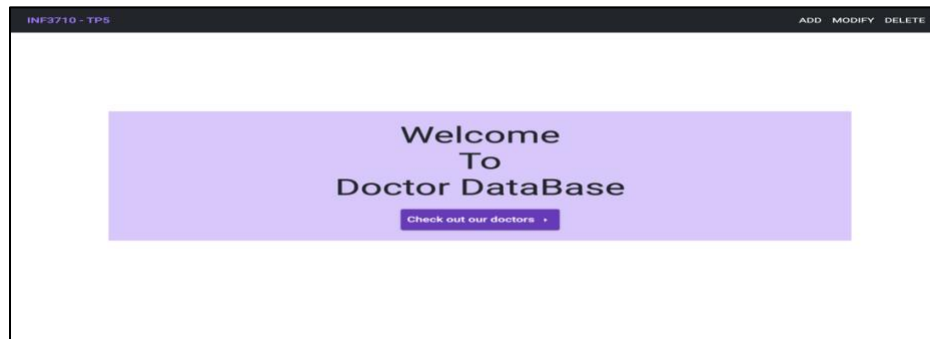


Figure 19. Page d'accueil.

INF3710 - TP5						ADD	MODIFY	DELETE
Doctor id	First name	Last name	Specialty	Service id	Years of experience			
1	Philippe	Lemelin	Neurologie	4	6			
2	Valérie	Bélanger	Ophtalmologie	1	10			
3	Alex	Michaud	Orthopédie	2	13			
4	Nathalie	Gagné	Psychiatrie	3	9			
5	Simon	Tremblay	Cardiologie	4	11			
6	Audrey	Beaulieu	Pédiatrie	1	7			
7	David	Fournier	Chirurgie	2	11			
8	Isabelle	Lapointe	Gynécologie	3	14			
9	Hamza	Boukafane	Neurologie	1	15			
10	Arman	Lidder	Psychiatrie	7	2			
11	Karim	Benzema	Dermatologie	0	1			

Figure 20. Page de présentation des médecins.

INF3710 - TP5						ADD	MODIFY	DELETE
First name	Last name	Specialty	Service id	Years of experience				
Marie	Rousseau	Dermatologie	3	8				
Philippe	Lemelin	Neurologie	4	6				
Valérie	Bélanger	Ophtalmologie	1	10				
Alex	Michaud	Orthopédie	2	13				
Nathalie	Gagné	Psychiatrie	3	9				
Simon	Tremblay	Cardiologie	4	11				
Audrey	Beaulieu	Pédiatrie	1	7				
David	Fournier	Chirurgie	2	11				
Isabelle	Lapointe	Gynécologie	3	14				
Hamza	Boukafane	Neurologie	1	15				
Arman	Lidder	Psychiatrie	7	2				

Figure 21. Page de d'ajout de médecins.

INF3710 - TP5

ADD

MODIFY

DELETE

Add New Doctor

First name:

Karim

Last name:

Benzema

Specialty:

Dermatologie

Service id:

0: Dermatologie

Years of experience:

1

Add New Doctor

Cancel

Figure 22. Formulaire d'ajout de médecins.

INF3710 - TP5						ADD	MODIFY	DELETE
First name	Last name	Specialty	Service id	Years of experience	Action			
Philippe	Lemelin	Neurologie	4	6				
Valérie	Bélanger	Ophtalmologie	1	10				
Alex	Michaud	Orthopédie	2	13				
Nathalie	Gagné	Psychiatrie	3	9				
Simon	Tremblay	Cardiologie	4	11				
Audrey	Beaulieu	Pédiatrie	1	7				
David	Fournier	Chirurgie	2	11				
Isabelle	Lapointe	Gynécologie	3	14				

Figure 23. Page de modifications de médecins.

INF3710 - TP5
ADD MODIFY DELETE

Modify Doctor

First name:
Simon

Last name:
Tremblay

Specialty:
Cardiologie

Service id:
4: Psychiatrie

Years of experience:
11

Save this doctor

Cancel

Figure 24. Formulaire de modifications de médecins.

INF3710 - TP5
ADD MODIFY DELETE

First name <small>Prénom</small>	Last name <small>Nom de famille</small>	Specialty <small>Spécialité</small>	Service id <small>Service</small>	Years of experience <small>Années d'expérience</small>	Action
Nathalie	Gagné	Psychiatrie	3	9	
Simon	Tremblay	Cardiologie	4	11	
Audrey	Beaulieu	Pédiatrie	1	7	
David	Fournier	Chirurgie	2	11	
Isabelle	Lapointe	Gynécologie	3	14	
Hamza	Boukaftane	Neurologie	1	15	
Arman	Lidder	Psychiatrie	7	2	
Karim	Benzema	Dermatologie	0	1	

Figure 25. Page de suppression de médecins.