



INF3710 –Bases de données

Automne 2023

TP No. 5

Groupe VB2_20

2130873 – Melliz Medina, Vivian

2147319 – Le Clainche, Lillo

Soumis à : Abdo, Joe

2023-12-02

4. Requête SQL (4.5pts)

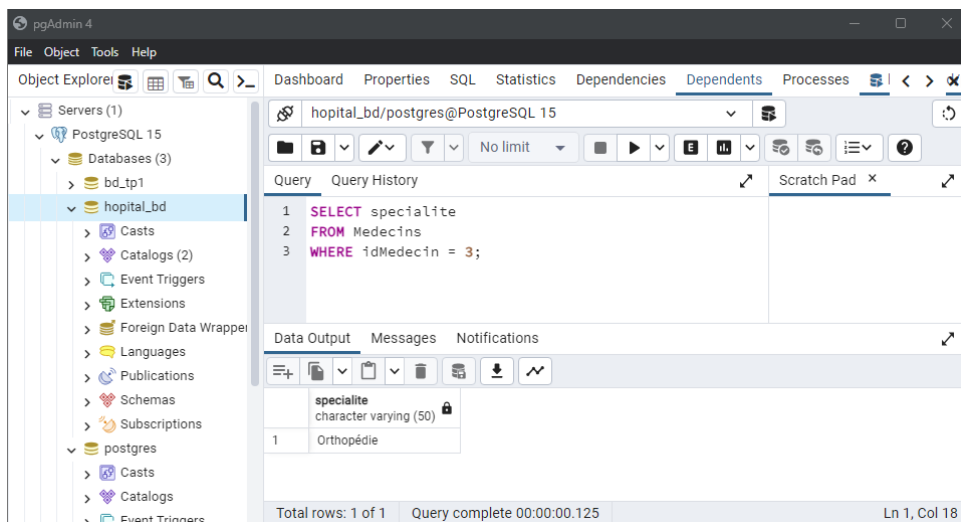
Pour chacune des questions suivantes, donnez la requête SQL en texte que vous avez effectuée et fournissez une capture d'écran du résultat de cette dernière. Avant de commencer, récupérez le code contenu dans le fichier [BD-TP5-schema.sql](#) et servez-vous en pour générer une base de données que vous nommerez `hopital_bd`. Ensuite, utilisez `BD-TP5-populate.sql` pour peupler votre base de données. Les requêtes suivantes devront être effectuées sur cette base de données et en utilisant le système de gestion de base de données PostgreSQL.

1) Affichez la spécialité du médecin avec l'identifiant 3. 0.25pt

SELECT specialite

FROM Medecins

WHERE idMedecin = 3;



Data Output Messages Notifications

Successfully run. Total query runtime: 125 msec.
1 rows affected.

2) Affichez les noms et prénoms des médecins qui ont plus de 10 ans d'expérience. 0.25pt

SELECT nom, prenom

FROM Medecins

WHERE anneesExperience > 10;

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with 'hopital_bd' selected. The main pane shows a SQL query in the 'Query' tab:

```
1 SELECT nom, prenom
2 FROM Medecins
3 WHERE anneesExperience > 10;
```

The 'Data Output' tab shows the results of the query:

	nom	prenom
	character varying (50)	character varying (50)
1	Michaud	Alex
2	Tremblay	Simon
3	Fournier	David
4	Lapointe	Isabelle

Total rows: 4 of 4 Query complete 00:00:00.053 Ln 1, Col 19

Data Output Messages Notifications

Successfully run. Total query runtime: 53 msec.
4 rows affected.

3) Affichez le nombre total d'examens qui ont été enregistrés dans la base de données. 0.25pt

SELECT COUNT(*)

FROM Examens;

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with 'hopital_bd' selected. The main pane shows a SQL query in the 'Query' tab:

```
1 SELECT COUNT(*)
2 FROM Examens;
```

The 'Data Output' tab shows the results of the query:

	count
	bigint
1	10

Total rows: 1 of 1 Query complete 00:00:00.072 Ln 2, Col 14

Data Output Messages Notifications

Successfully run. Total query runtime: 72 msec.
1 rows affected.

4) Afficher le nom de l'examen le plus coûteux, son prix ainsi que la dernière date à laquelle il a été effectué. 0.50pt

```
SELECT e.nomExamen, e.cout, MAX(pe.dateExamen)
```

```
FROM Examens e
```

```
JOIN PatientExamens pe on e.idExamen = pe.idExamen
```

```
GROUP BY e.nomExamen, e.cout
```

```
ORDER BY e.cout DESC
```

```
LIMIT 1;
```

The screenshot shows a PostgreSQL query editor interface. The left sidebar displays a tree view of the database structure, including 'Servers (1)', 'PostgreSQL 15', 'Databases (3)', 'bd_tp1', and 'hopital_bd'. The 'hopital_bd' database is selected, showing its contents like 'Casts', 'Catalogs (2)', 'Event Triggers', 'Extensions', 'Foreign Data Wrappers', 'Languages', 'Publications', 'Schemas', and 'Subscriptions'. The main query editor displays the following SQL query:

```
1 SELECT e.nomExamen, e.cout, MAX(pe.dateExamen)
2
3 FROM Examens e
4
5 JOIN PatientExamens pe on e.idExamen = pe.idExamen
6
7 GROUP BY e.nomExamen, e.cout
8
9 ORDER BY e.cout DESC
10
11 LIMIT 1;
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query in a table:

	nomexamen character varying (100)	cout numeric (10,2)	max date
1	IRM cérébrale	350.00	2023-10-20

The 'Messages' tab at the bottom shows the execution status: 'Successfully run. Total query runtime: 171 msec. 1 rows affected.'

5) Affichez les prénoms et noms des patients avec leur âge, en ordre décroissant de l'alphabet. 0.50pt

```
SELECT prenom, nom, EXTRACT(YEAR FROM AGE(NOW(), dateDeNaissance)) as age
```

```
FROM Patients
```

```
ORDER BY prenom DESC, nom DESC;
```

pgAdmin 4

File Object Tools Help

Dashboard Properties SQL Statistics Dependencies Dependents Processes hospital_bd/

hospital_bd/postgres@PostgreSQL 15

Query Query History

```

1 SELECT prenom, nom, EXTRACT(YEAR FROM AGE(NOW(), dateDeNaissance)) as age
2 FROM Patients
3 ORDER BY prenom DESC, nom DESC;

```

Data Output Messages Notifications

	prenom character varying (50)	nom character varying (50)	age numeric
1	Sophie	Leblanc	27
2	Sébastien	Lemieux	46
3	Martin	Girard	23
4	Julie	Côté	41
5	François	Dufresne	52
6	Émilie	Roy	48
7	David	Tessier	22
8	Claire	Lavoie	34
9	Caroline	Morin	35
10	Alexandre	Boucher	29

Total rows: 10 of 10 Query complete 00:00:00.060 Ln 2, Col 14

Data Output Messages Notifications

Successfully run. Total query runtime: 60 msec.
10 rows affected.

6) Affichez le prix moyen du coût des examens médicaux. 0.25pt

SELECT round(avg(cout),2) as "Prix moyen du coût des examens médicaux"

FROM Examens;

pgAdmin 4

File Object Tools Help

Dashboard Properties SQL Statistics Dependencies Dependents Processes hospital_bd/

hospital_bd/postgres@PostgreSQL 15

Query Query History

```

1 SELECT round(avg(cout),2) as "Prix moyen du coût des examens médicaux"
2 FROM Examens;

```

Data Output Messages Notifications

	Prix moyen du coût des examens médicaux numeric
1	221.00

Total rows: 1 of 1 Query complete 00:00:00.079 Ln 1, Col 26

Data Output Messages Notifications

Successfully run. Total query runtime: 47 msec.
1 rows affected.

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. 0.50pt

```
SELECT m.nom, m.prenom, count(rv.idRendezVous)
```

```
FROM Medecins m
```

```
JOIN Rendezvous rv on m.idMedecin = rv.idMedecin
```

```
WHERE rv.dateRendezVous >= TO_DATE('21-09-2023', 'DD-MM-YYYY')
```

```
GROUP BY m.idMedecin;
```

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including the 'hopital_bd' database. The central pane shows a SQL query:

```
1 SELECT m.nom, m.prenom, count(rv.idRendezVous)
2 FROM Medecins m
3 JOIN Rendezvous rv on m.idMedecin = rv.idMedecin
4 WHERE rv.dateRendezVous >= TO_DATE('21-09-2023', 'DD-MM-YYYY')
5 GROUP BY m.idMedecin, m.prenom, m.nom;
```

 The bottom pane shows the 'Data Output' tab with a table of results. The table has three columns: 'nom', 'prenom', and 'count'. The results are as follows:

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

The status bar at the bottom indicates 'Total rows: 10 of 10' and 'Query complete 00:00:00.054'.

Data Output Messages Notifications

Successfully run. Total query runtime: 54 msec.
10 rows affected.

8) Affichez les noms, prénoms et numéro de téléphone des patients qui ont des rendez-vous la semaine prochaine. 0.50pt

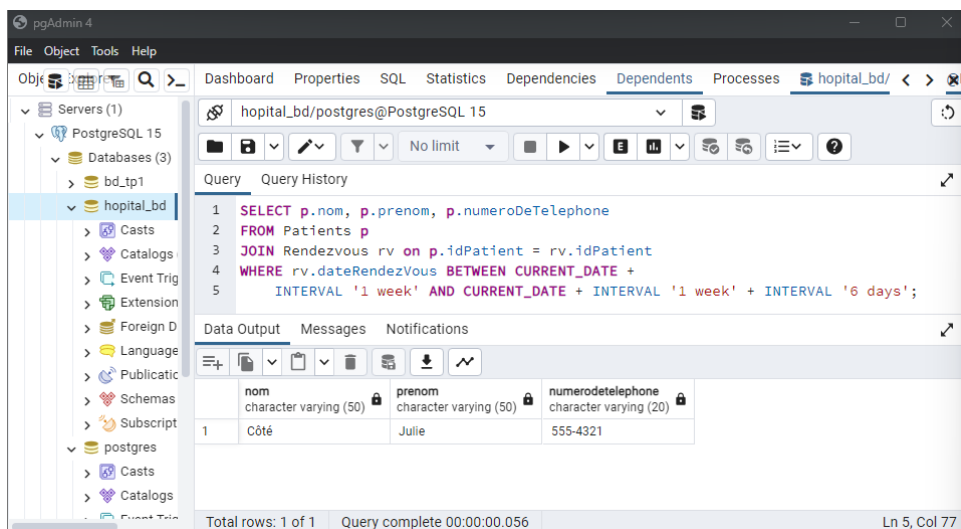
```
SELECT p.nom, p.prenom, p.numeroDeTelephone
```

```
FROM Patients p
```

```
JOIN Rendezvous rv on p.idPatient = rv.idPatient
```

```
WHERE rv.dateRendezVous BETWEEN CURRENT_DATE +
```

```
INTERVAL '1 week' AND CURRENT_DATE + INTERVAL '1 week' + INTERVAL '6 days';
```



Data Output Messages Notifications

Successfully run. Total query runtime: 56 msec.
1 rows affected.

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'? 0.25pt

```
SELECT p.nom, p.prenom, sum(e.cout)
```

```
FROM Patients p
```

```
JOIN Rendezvous rv on p.idPatient = rv.idPatient
```

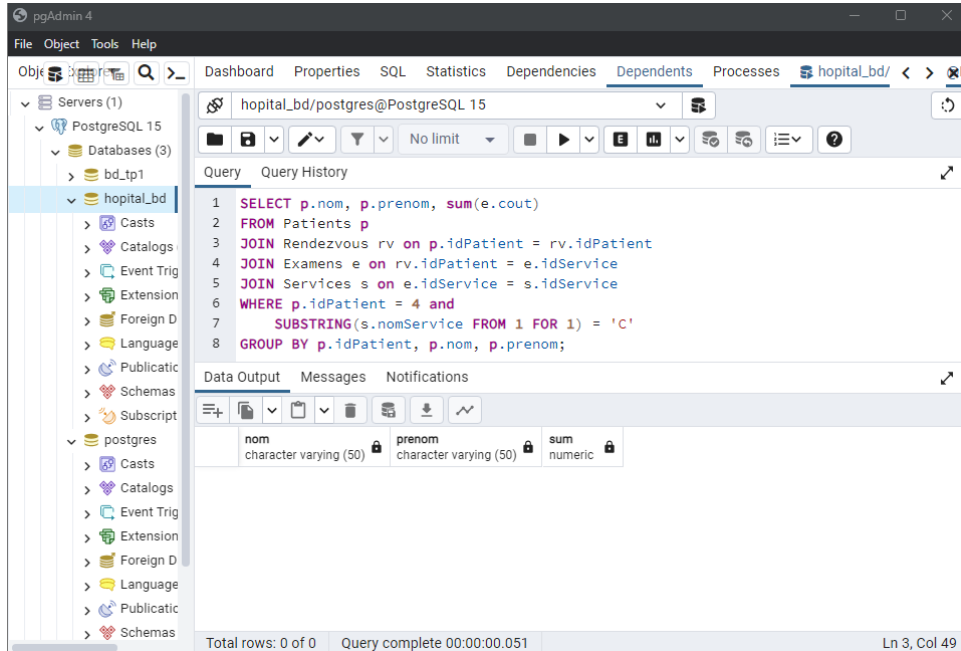
```
JOIN Examens e on rv.idPatient = e.idService
```

```
JOIN Services s on e.idService = s.idService
```

WHERE p.idPatient = 4 and

SUBSTRING(s.nomService FROM 1 FOR 1) = 'C'

GROUP BY p.idPatient, p.nom, p.prenom;



Data Output Messages Notifications

Successfully run. Total query runtime: 51 msec.
0 rows affected.

10) Afficher les nom et prénoms des patients qui ont consulté le médecin avec le plus d'expérience.
0.5pt

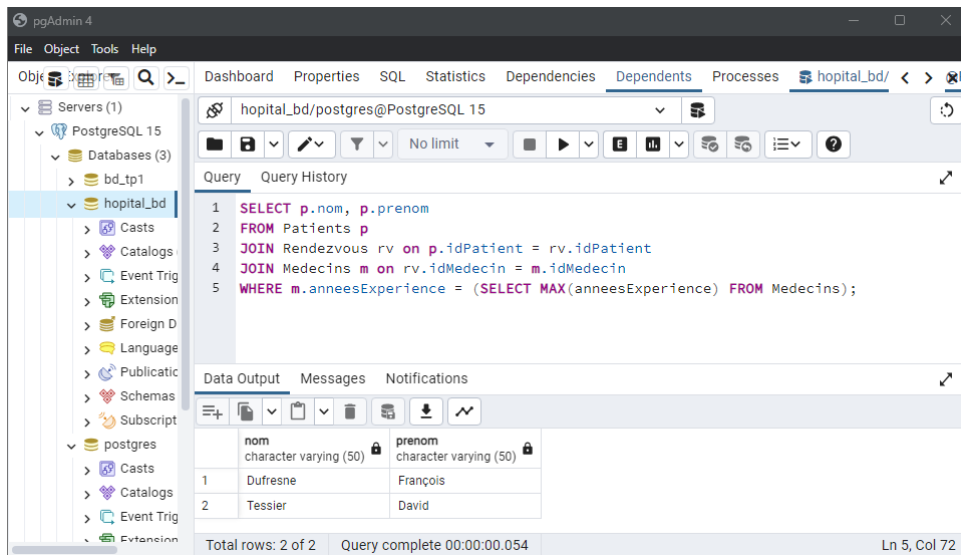
SELECT p.nom, p.prenom

FROM Patients p

JOIN Rendezvous rv on p.idPatient = rv.idPatient

JOIN Medecins m on rv.idMedecin = m.idMedecin

WHERE m.anneesExperience = (SELECT MAX(anneesExperience) FROM Medecins);



Data Output Messages Notifications

Successfully run. Total query runtime: 54 msec.
2 rows affected.

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\$. 0.25pt

SELECT p.nom, p.prenom

FROM Patients p

JOIN PatientExamens pe ON p.idPatient = pe.idPatient

JOIN Examens e on pe.idExamen = e.idExamen

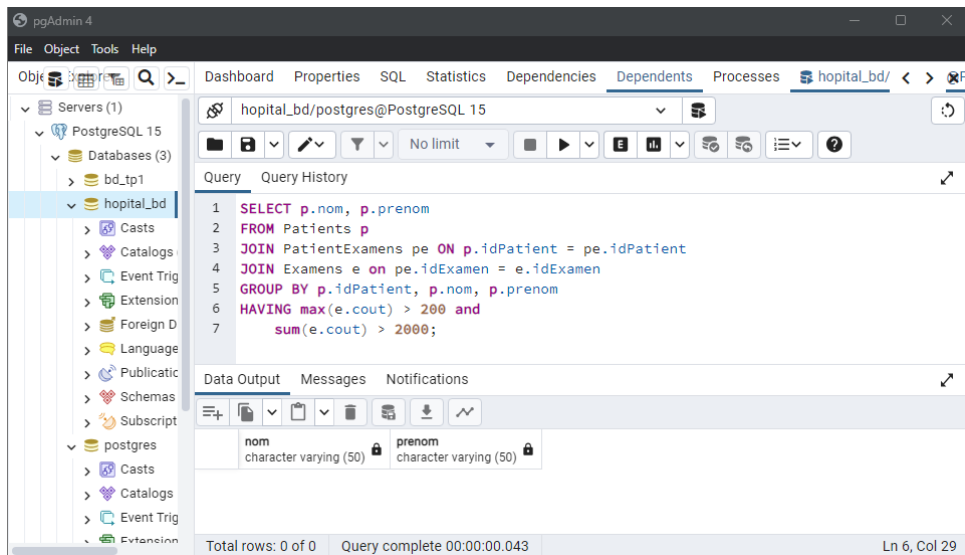
GROUP BY p.idPatient, p.nom, p.prenom

HAVING max(e.cout) > 200 and

sum(e.cout) > 2000;

HAVING e.cout > 200 and

sum(e.cout) > 2000;



Data Output Messages Notifications

Successfully run. Total query runtime: 43 msec.
0 rows affected.

12) Afficher le nom et prénom du médecin ayant le plus grand nombre de patients différents consultés durant les 6 derniers mois. 0.25pt

SELECT m.nom, m.prenom

FROM Medecins m

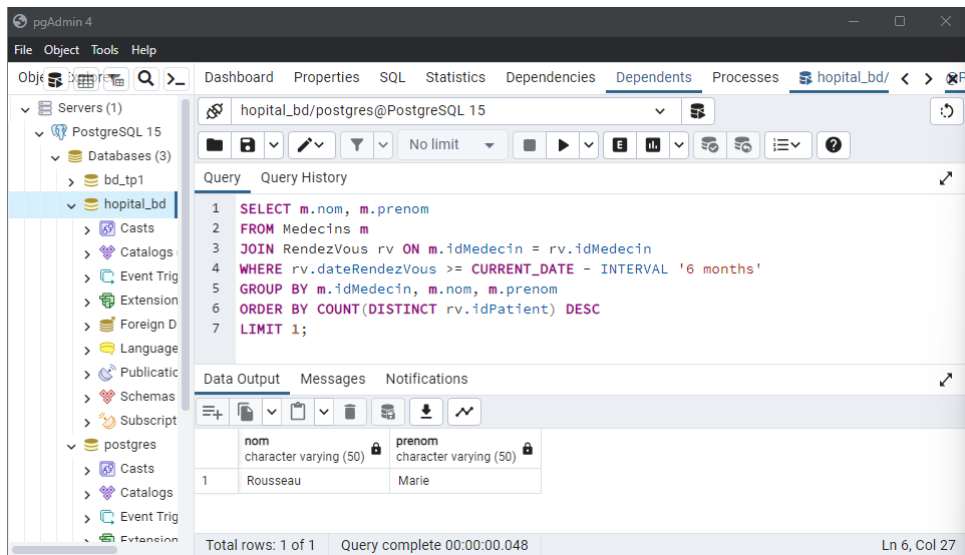
JOIN RendezVous rv ON m.idMedecin = rv.idMedecin

WHERE rv.dateRendezVous >= CURRENT_DATE - INTERVAL '6 months'

GROUP BY m.idMedecin, m.nom, m.prenom

ORDER BY COUNT(DISTINCT rv.idPatient) DESC

LIMIT 1;



Data Output Messages Notifications

Successfully run. Total query runtime: 48 msec.
1 rows affected.

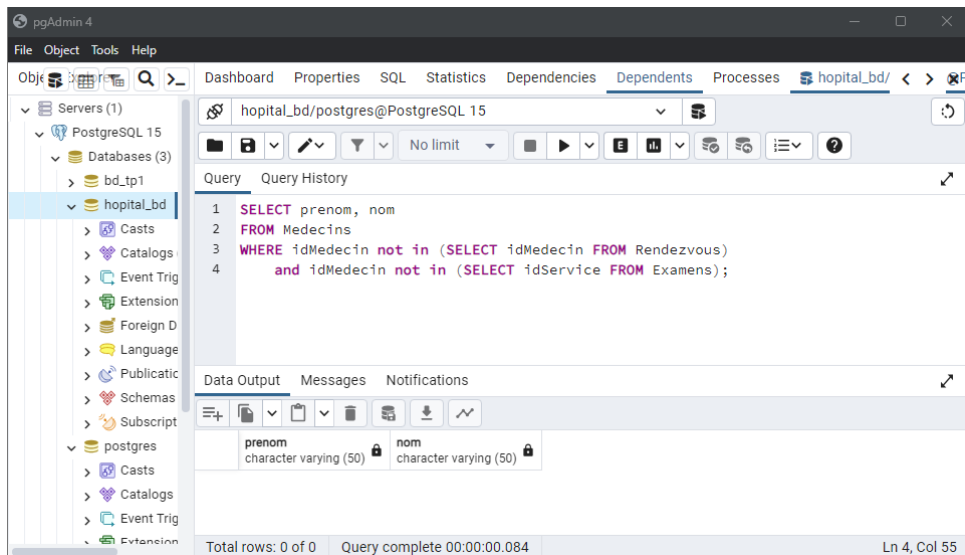
13) Affichez le prénom et le nom des médecins qui n'ont jamais eu de rendez-vous et n'ont effectué aucun examen. 0.50pt

SELECT prenom, nom

FROM Medecins

WHERE idMedecin not in (SELECT idMedecin FROM Rendezvous)

and idMedecin not in (SELECT idService FROM Examens);



Data Output Messages Notifications

Successfully run. Total query runtime: 84 msec.
0 rows affected.

14) Affichez le nom et prénom du patient qui a passé le plus grand nombre d'examens sans obtenir de résultat "Résultats normaux". 0.50pt

SELECT p.nom, p.prenom

FROM Patients p

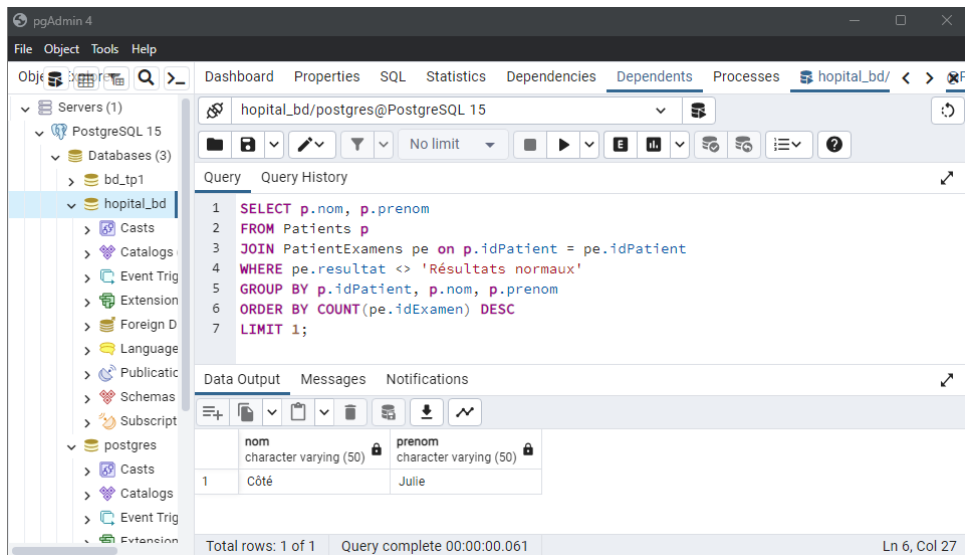
JOIN PatientExamens pe on p.idPatient = pe.idPatient

WHERE pe.resultat <> 'Résultats normaux'

GROUP BY p.idPatient, p.nom, p.prenom

ORDER BY COUNT(pe.idExamen) DESC

LIMIT 1;



Data Output Messages Notifications

Successfully run. Total query runtime: 61 msec.
1 rows affected.

15) Affichez les noms et prénoms des patients qui ont consulté un médecin de chaque service. 0.5pt

SELECT p.nom, p.prenom

FROM Patients p

JOIN Rendezvous rv on p.idPatient = rv.idPatient

JOIN Medecins m on rv.idMedecin = m.idMedecin

JOIN Services s on m.idService = s.idService

WHERE EXISTS (

SELECT idService

FROM Services

WHERE idService not in (

SELECT m.idService

FROM Medecins m

JOIN Rendezvous r on m.idMedecin = r.idMedecin

WHERE p.idPatient = r.idPatient

)
)

GROUP BY p.idPatient, p.nom, p.prenom;

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including the 'hopital_bd' database. The main window shows a SQL query in the 'Query' tab. The query is as follows:

```
1 SELECT p.nom, p.prenom
2 FROM Patients p
3 JOIN Rendezvous rv on p.idPatient = rv.idPatient
4 JOIN Medecins m on rv.idMedecin = m.idMedecin
5 JOIN Services s on m.idService = s.idService
6 WHERE EXISTS (
7     SELECT idService
8     FROM Services
9     WHERE idService not in (
10        SELECT m.idService
11        FROM Medecins m
12        JOIN Rendezvous r on m.idMedecin = r.idMedecin
13        WHERE p.idPatient = r.idPatient
14    )
15 )
16 GROUP BY p.idPatient, p.nom, p.prenom;
```

The 'Data Output' tab shows the results of the query, which are 10 rows of patient data:

	nom_patient character varying (50)	prenom_patient character varying (50)
1	Lavoie	Claire
2	Tessier	David
3	Roy	Émilie
4	Boucher	Alexandre
5	Côté	Julie
6	Dufresne	François
7	Leblanc	Sophie
8	Girard	Martin
9	Morin	Caroline
10	Lemieux	Sébastien

The status bar at the bottom indicates 'Total rows: 10 of 10' and 'Query complete 00:00:00.080'.

Data Output Messages Notifications
Successfully run. Total query runtime: 80 msec.
10 rows affected.

16) Créez une vue 'V_TotalExamens' contenant le nom du patient 'V_nom', le prénom du patient 'V_prenom' et le nombre total d'examens passés par chaque patient 'V_tot'. Cette vue doit uniquement contenir les patients dont V_tot est supérieur à 500\$ et dont l'adresse contient la lettre 'e' ou la lettre 'o' à la troisième position de la fin ; triés par ordre croissant selon le nom et par ordre décroissant selon V_tot. Finalement, afficher le résultat de cette vue. 0.50pt

CREATE VIEW V_TotalExamens (V_nom, V_prenom, V_tot) as (

SELECT p.nom, p.prenom, count(pe.idExamen)

FROM Patients p

JOIN PatientExamens pe on p.idPatient = pe.idPatient

JOIN Examens e on pe.idExamen = e.idExamen

WHERE e.cout > 500 and

SUBSTRING(p.adresse FROM LENGTH(p.adresse)-2 FOR 1) in ('e', 'o')

GROUP BY p.idPatient, p.nom, p.prenom

ORDER BY p.nom ASC, count(pe.idExamen) DESC

);

The screenshot shows the pgAdmin 4 web interface. On the left, the 'Servers' tree is expanded to show the 'hopital_bd' database. The main panel displays a SQL query in the 'Query' tab. The query is a CREATE VIEW statement for 'V_TotalExamens'. The query text is as follows:

```
1 CREATE VIEW V_TotalExamens (V_nom, V_prenom, V_tot) as (  
2   SELECT p.nom, p.prenom, count(pe.idExamen)  
3   FROM Patients p  
4   JOIN PatientExamens pe on p.idPatient = pe.idPatient  
5   JOIN Examens e on pe.idExamen = e.idExamen  
6   WHERE e.cout > 500 and  
7     SUBSTRING(p.adresse FROM LENGTH(p.adresse)-2 FOR 1) in ('e', 'o')  
8   GROUP BY p.idPatient, p.nom, p.prenom  
9   ORDER BY p.nom ASC, count(pe.idExamen) DESC  
10 );
```

Below the query editor, the 'Messages' tab shows the execution result: 'CREATE VIEW' and 'Query returned successfully in 41 msec.' The status bar at the bottom indicates 'Total rows: 10 of 10', 'Query complete 00:00:00.041', and 'Ln 10, Col 3'.

pgAdmin 4

File Object Tools Help

Dashboard Properties SQL Statistics Dependencies **Dependents** Processes hopita < > < > < >

hopital_bd/postgres@PostgreSQL 15

Query Query History

```
1 SELECT * FROM V_TotalExamens;
```

Data Output Messages Notifications

v_nom	v_prenom	v_tot
character varying (50)	character varying (50)	bigint

Total rows: 0 of 0 Query complete 00:00:00.056 Ln 1, Col 30

5. Application Web (5pts)

En se basant sur la base de données [hopital_bd](#), vous êtes confiés à créer une application Web comme suit. À noter qu'il faut référer au guide d'application disponible sur Moodle sur ce lien (Tutoriel de l'application Web : <https://moodle4.polymtl.ca/mod/folder/view.php?id=540662>).

5.1- Votre application Web doit contenir une page qui affiche tous les champs et toutes les entrées de la table Medecin. (1pt)

INF3710 - TP4 Ajouter Modifier Supprimer

Voici la liste des médecins:

ID	Prénom	Nom	Spécialité	Années d'expérience	ID du Service		
0	Marie	Rousseau	Dermatologie	8	3	Modifier	Supprimer
1	Philippe	Lemelin	Neurologie	6	4	Modifier	Supprimer
2	Valérie	Bélanger	Ophthalmologie	10	1	Modifier	Supprimer
3	Alex	Michaud	Orthopédie	12	2	Modifier	Supprimer
4	Nathalie	Gaoné	Psychiatrie	9	3	Modifier	Supprimer

INF3710 - TP4 Ajouter Modifier Supprimer

Voici la liste des médecins:

5	Simon	Tremblay	Cardiologie	15	4	Modifier	Supprimer
6	Audrey	Beaulieu	Pédiatrie	7	1	Modifier	Supprimer
7	David	Fournier	Chirurgie	11	2	Modifier	Supprimer
8	Isabelle	Lapointe	Gynécologie	14	3	Modifier	Supprimer
9	François	Martel	Radiologie	5	4	Modifier	Supprimer

5.2- Nous voulons pouvoir ajouter (INSERT), modifier (MODIFY) et supprimer (DELETE) n'importe quel Medecin présent au sein de votre base de données par l'intermédiaire d'une modale de votre application Web. Votre application Web devrait gérer toutes les erreurs possibles, comme les erreurs de référencement de clefs, en offrant à l'utilisateur une modale avec une description appropriée de l'erreur survenue. L'ajout, la modification et la suppression doivent avoir leur propre page. Dans la page d'ajout,

on inclut un formulaire avec les informations à saisir pour le Medecin. Pensez à proposer des valeurs par défaut pour chaque champ de saisie, afin de ne pas avoir à remplir tous les champs du formulaire lors des tests. Quand c'est nécessaire, on utilisera des listes pour sélectionner une valeur (pour la spécialité, l'identifiant du service avec son nom) obtenue de la base de données. Dans la page de modification, on s'attend à charger toutes les informations reliées au Medecin à partir de la base de données et on permet leur modification, de même pour la suppression. (3pts)

Modifier le médecin :

ID	Prénom	Nom	Spécialité	Années d'expérience	ID du Service
9	<input type="text" value="François"/>	<input type="text" value="Martel"/>	<input type="text" value="Radiologie"/>	<input type="text" value="5"/>	<input type="text" value="4"/>

Modifier

Retour à la liste des médecins

Ajouter un médecin :

ID	Prénom	Nom	Spécialité	Années d'expérience	ID du Service
10	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>

Ajouter

Retour à la liste des médecins

Êtes-vous certain de vouloir supprimer ce Médecin?

ID	Prénom	Nom	Spécialité	Années d'expérience	ID du Service
9	François	Martel	Radiologie	5	4

Supprimer

Retour à la liste des médecins

5.3- Votre application Web devrait contenir une page d'accueil qui contient des liens directs vers des dites pages à implémenter énumérées dans les deux questions 5.1 et 5.2. (1pt)

INF3710 - TP4

Ajouter Modifier Supprimer

Centre de gestion des Médecins

Afficher les médecins »

Nous vous conseillons fortement de vous inspirer du guide d'application vu en laboratoire et disponible sur Moodle sur ce lien (Tutoriel de l'application Web :

<https://moodle4.polymtl.ca/mod/folder/view.php?id=540662>). Enfin, nous recommandons grandement l'utilisation d'Angular Material pour accélérer votre programmation.