



## **INF3710 –Bases de données**

**Automne 2023**

**TP No. 5**

**Groupe 2**

**1904589 – Miguel Gascon**

**2183409 – Xavier Schlegel**

**Soumis à : M. Rachad Chazbek**

**2 décembre 2023**

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

SELECT specialite FROM Medecins WHERE idMedecin = 3;

Query		Query History
1	SELECT	specialite FROM Medecins WHERE idMedecin = 3;
2		

Data Output		Messages	Notifications
	specialite character varying (50)		
1	Orthopédie		

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;

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

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

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

SELECT COUNT(\*) FROM Examens;

1	SELECT	COUNT(*) FROM Examens;
2		

Data Output		Messages	Graph Visualiser	Notifications
	count bigint			
1	10			

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
WHERE E.cout = (SELECT MAX(cout) FROM Examens) GROUP BY E.nomExamen, E.cout;
```

Query		Query History	
1	SELECT	E.nomExamen, E.cout, MAX(PE.dateExamen)	FROM
2	Examens E JOIN	PatientExamens PE ON E.idExamen = PE.idExamen	
3	WHERE	E.cout = (SELECT MAX(cout) FROM Examens) GROUP BY E.nomExamen, E.cout;	
4			

Data Output		Messages	Graph Visualiser	×	Notifications
	nomexamen	cout	max		
	character varying (100)	numeric (10,2)	date		
1	IRM cérébrale	350.00	2023-10-20		

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(CURRENT_DATE, dateDeNaissance)) AS age
FROM Patients
ORDER BY prenom DESC, nom DESC;
```

1	SELECT	prenom, nom, EXTRACT (YEAR FROM age(CURRENT_DATE, dateDeNaissance)) AS age	
2	FROM	Patients	
3	ORDER BY	prenom DESC, nom DESC;	

Data Output		Messages	Graph Visualiser	×	Notifications
	prenom	nom	age		
	character varying (50)	character varying (50)	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		

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

```
SELECT AVG(cout) FROM Examens;
```

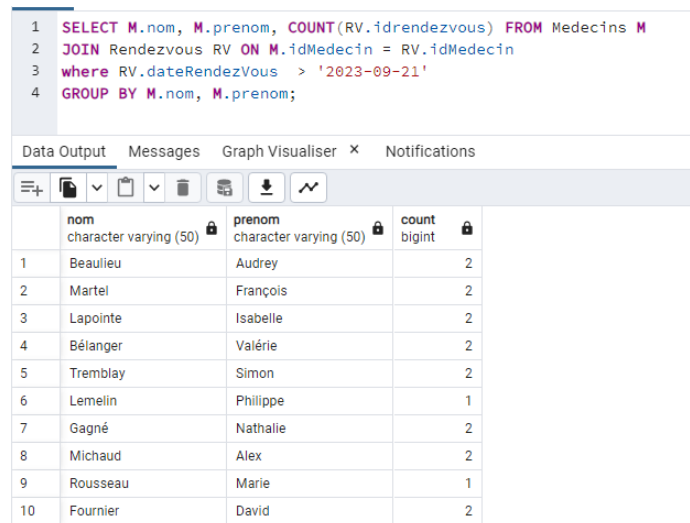
1	SELECT	AVG(cout) FROM Examens;	
2			
3			

Data Output		Messages	Graph Visualiser	×	Notifications
	avg				
	numeric				
1	221.0000000000000000				

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 > '2023-09-21'
GROUP BY M.nom, M.prenom;
```



The screenshot shows a SQL query editor with a query and its results. The query is:

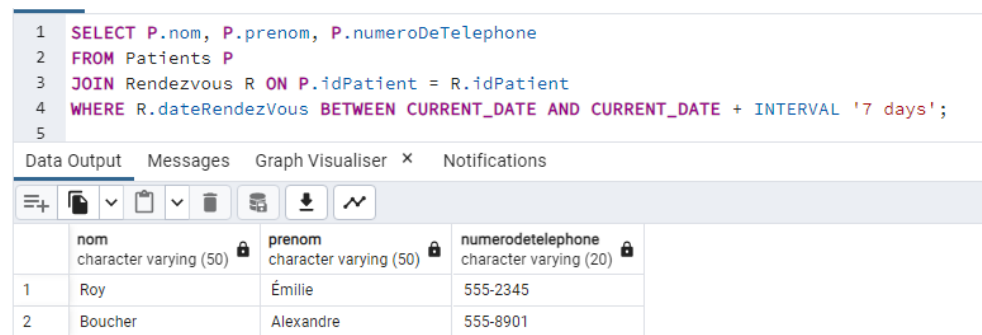
```
1 SELECT M.nom, M.prenom, COUNT(RV.idrendezvous) FROM Medecins M
2 JOIN Rendezvous RV ON M.idMedecin = RV.idMedecin
3 where RV.dateRendezVous > '2023-09-21'
4 GROUP BY M.nom, M.prenom;
```

The results are displayed in a table with the following columns: nom, prenom, and count. The table contains 10 rows of data.

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

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 R ON P.idPatient = R.idPatient
WHERE R.dateRendezVous BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '7 days';
```



The screenshot shows a SQL query editor with a query and its results. The query is:

```
1 SELECT P.nom, P.prenom, P.numeroDeTelephone
2 FROM Patients P
3 JOIN Rendezvous R ON P.idPatient = R.idPatient
4 WHERE R.dateRendezVous BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '7 days';
5
```

The results are displayed in a table with the following columns: nom, prenom, and numerodetelephone. The table contains 2 rows of data.

	nom character varying (50)	prenom character varying (50)	numerodetelephone character varying (20)
1	Roy	Émilie	555-2345
2	Boucher	Alexandre	555-8901

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 PatientExamens PE ON P.idPatient = PE.idPatient
JOIN Examens E ON PE.idExamen = E.idExamen
JOIN Services S ON E.idService = S.idService
WHERE P.idPatient = 4 AND S.nomService LIKE 'C%'
GROUP BY P.nom, P.prenom;
```

The screenshot shows a SQL query editor with the following query:

```
1 SELECT P.nom, P.prenom, SUM(E.cout)
2 FROM Patients P
3 JOIN PatientExamens PE ON P.idPatient = PE.idPatient
4 JOIN Examens E ON PE.idExamen = E.idExamen
5 JOIN Services S ON E.idService = S.idService
6 WHERE P.idPatient = 4 AND S.nomService LIKE 'C%'
7 GROUP BY P.nom, P.prenom;
```

Below the query editor, there is a toolbar with icons for undo, redo, save, and other functions. Below the toolbar, there is a table with the following columns:

	nom character varying (50)	prenom character varying (50)	sum numeric
--	-------------------------------	----------------------------------	----------------

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

```
SELECT DISTINCT Pat.nom, Pat.prenom
FROM Patients Pat
JOIN Rendezvous R ON Pat.idPatient = R.idPatient
JOIN Medecins M ON R.idMedecin = M.idMedecin
WHERE M.anneesExperience = (SELECT MAX(anneesExperience) FROM Medecins);
```

The screenshot shows a SQL query editor with the following query:

```
1 SELECT DISTINCT Pat.nom, Pat.prenom
2 FROM Patients Pat
3 JOIN Rendezvous R ON Pat.idPatient = R.idPatient
4 JOIN Medecins M ON R.idMedecin = M.idMedecin
5 WHERE M.anneesExperience = (SELECT MAX(anneesExperience) FROM Medecins);
```

Below the query editor, there is a toolbar with icons for undo, redo, save, and other functions. Below the toolbar, there is a table with the following columns:

	nom character varying (50)	prenom character varying (50)
1	Dufresne	François
2	Tessier	David

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 DISTINCT P.nom, P.prenom
FROM Patients P
JOIN PatientExamens PE ON P.idPatient = PE.idPatient
JOIN Examens E ON E.idExamen = PE.idExamen
WHERE E.cout > 200
GROUP BY P.nom, P.prenom Having SUM(E.cout) > 2000;
```

The screenshot shows a SQL query editor with two tabs: 'Query' and 'Query History'. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT DISTINCT P.nom, P.prenom
2 FROM Patients P
3 JOIN PatientExamens PE ON P.idPatient = PE.idPatient
4 JOIN Examens E ON E.idExamen = PE.idExamen
5 WHERE E.cout > 200
6 GROUP BY P.nom, P.prenom Having SUM(E.cout) > 2000;
```

Below the query editor, there are three tabs: 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with two columns: 'nom' and 'prenom'. Both columns are described as 'character varying (50)'.

nom	prenom
character varying (50)	character varying (50)

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 R ON M.idMedecin = R.idMedecin
WHERE R.dateRendezVous BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '6 months'
GROUP BY M.nom, M.prenom
ORDER BY COUNT(DISTINCT R.idPatient) DESC
LIMIT 1;
```

The screenshot shows a SQL query editor with two tabs: 'Query' and 'Query History'. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT M.nom, M.prenom
2 FROM Medecins M
3 JOIN Rendezvous R ON M.idMedecin = R.idMedecin
4 WHERE R.dateRendezVous BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '6 months'
5 GROUP BY M.nom, M.prenom
6 ORDER BY COUNT(DISTINCT R.idPatient) DESC
7 LIMIT 1;
```

Below the query editor, there are three tabs: 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with two columns: 'nom' and 'prenom'. Both columns are described as 'character varying (50)'. The first row of data shows 'Beaulieu' for 'nom' and 'Audrey' for 'prenom'.

nom	prenom
character varying (50)	character varying (50)
Beaulieu	Audrey

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 M.prenom, M.nom
FROM Medecins M
LEFT JOIN Rendezvous R ON M.idMedecin = R.idMedecin
WHERE R.idMedecin IS NULL;
```

Query	Query History
1	SELECT M.prenom, M.nom
2	FROM Medecins M
3	LEFT JOIN Rendezvous R ON M.idMedecin = R.idMedecin
4	WHERE R.idMedecin IS NULL;

Data Output	Messages	Notifications
	prenom character varying (50)	nom character varying (50)

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.nom, P.prenom
ORDER BY COUNT(PE.idExamen) DESC
LIMIT 1;
```

Query	Query History
1	SELECT P.nom, P.prenom
2	FROM Patients P
3	JOIN PatientExamens PE ON P.idPatient = PE.idPatient
4	WHERE PE.resultat != 'Résultats normaux'
5	GROUP BY P.nom, P.prenom
6	ORDER BY COUNT(PE.idExamen) DESC
7	LIMIT 1;

Data Output	Messages	Notifications
	nom character varying (50)	prenom character varying (50)
1	Morin	Caroline

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

```
SELECT P.prenom, P.nom
FROM Patients P
JOIN Rendezvous R ON P.idPatient = R.idPatient
JOIN Medecins M ON R.idMedecin = M.idMedecin
JOIN Services S ON M.idService = S.idService
GROUP BY P.idPatient
HAVING COUNT(DISTINCT S.idService) = (SELECT COUNT(*) FROM Services);
```

The screenshot shows a SQL query editor with two tabs: 'Query' and 'Query History'. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT P.prenom, P.nom
2 FROM Patients P
3 JOIN Rendezvous R ON P.idPatient = R.idPatient
4 JOIN Medecins M ON R.idMedecin = M.idMedecin
5 JOIN Services S ON M.idService = S.idService
6 GROUP BY P.idPatient
7 HAVING COUNT(DISTINCT S.idService) = (SELECT COUNT(*) FROM Services);
```

Below the query editor, there are three tabs: 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with two columns: 'prenom' and 'nom'. Both columns are of type 'character varying (50)' and have a lock icon.

prenom	nom
--------	-----



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 AS
SELECT
    P.nom AS V_nom,
    P.prenom AS V_prenom,
    SUM(E.cout) AS V_tot
FROM Patients P
JOIN PatientExamens PE ON P.idPatient = PE.idPatient
JOIN Examens E ON PE.idExamen = E.idExamen
WHERE (P.adresse LIKE '___e%' OR P.adresse LIKE '___o%')
GROUP BY P.idPatient
HAVING SUM(E.cout) > 500
ORDER BY P.nom ASC, V_tot DESC;
```

Query Query History

```

1 CREATE VIEW V_TotalExamens AS
2 SELECT
3     P.nom AS V_nom,
4     P.prenom AS V_prenom,
5     SUM(E.cout) AS V_tot
6 FROM Patients P
7 JOIN PatientExamens PE ON P.idPatient = PE.idPatient
8 JOIN Examens E ON PE.idExamen = E.idExamen
9 WHERE (P.adresse LIKE '___e%' OR P.adresse LIKE '___o%')
10 GROUP BY P.idPatient
11 HAVING SUM(E.cout) > 500
12 ORDER BY P.nom ASC, V_tot DESC;
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 69 msec.

select \* from V\_TotalExamens;

Query Query History

```

1 select * from V_TotalExamens;
```

Data Output Messages Notifications

≡

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	v_nom character varying (50) 🔒	v_prenom character varying (50) 🔒	v_tot numeric 🔒
--	-----------------------------------	--------------------------------------	--------------------