

PHP

Partie 4 - Le SQL

...

Mathieu Nibas
B3 - ECV 2023



Le programme de cette partie

Etat des lieux

Base de données relationnelles

Langage de manipulation de données



Etat des lieux

Langage informatique normalisé servant à exploiter des **bases de données relationnelles**

Créé en 1974, il est depuis devenu incontournable

L'immense majorité des systèmes de gestion de bases de données relationnelles (abrégé SGBD-R) du marché le supporte, **à quelques variations près**

** MySQL, Oracle Database, Microsoft Access, PostgreSQL, SQLite.*

Les choses bougent avec la tendance « NoSQL » (Not Only SQL)



Base de données relationnelles

L'information est organisée dans des tableaux à deux dimensions appelés des **relations** ou **tables**

- Les lignes de ces relations sont appelées des n-uplets ou enregistrements
- Les colonnes sont appelées des attributs



Base de données relationnelles

Une relation (au sens du modèle de Codd) est constituée de deux choses :

- Un Schéma : Le format de la table (incluant types et contraintes).
Le schéma est fixé ;
- Une Extension : Le contenu de la table, qui est un ensemble de n-uplets dont l'ordre n'a pas d'importance, et sans doublons

Le qualificatif « relationnelle » d'une BDD ne fait ainsi aucunement référence au fait que les tables peuvent être reliées entre elles.



Base de données relationnelles

Dossard	Nom	Prenom	Club
503	Afond	Jocelyne	AP
123	Véloce	Marc	NULL
229	Le Goaër	Olivier	TDB
303	Speedy	Nadine	AP
271	Pressé	Marc	E64
404	Asphalte	Didier	NULL

Relation « Coureur »

Code	Intitule	Pro?
AP	Aigles de Pau	No
E64	Endurance 64	Yes
ASM	ASM Mourenx	No
TDB	Touristes du Béarn	No

Relation « Club »



Langage de manipulation de données



Langage de manipulation de données

Sous-ensemble du SQL utilisé pour ajouter, consulter, modifier, et supprimer des données des tables existantes

→ Correspond aux commandes SQL: INSERT, SELECT, UPDATE, DELETE

Syntaxe générale d'une **requête** pour consulter des données :

<code>SELECT</code> attributs	-- ce que l'on cherche
<code>FROM</code> relations	-- où on cherche
[<code>WHERE</code> condition]	-- avec quelles conditions
[<code>GROUP BY</code> attributs]	-- en utilisant
[<code>HAVING</code> condition]	-- des sous-ensembles
[<code>ORDER BY</code> attributs]	-- résultat dans quel ordre



Sélection d'attributs

Le mot clé SELECT vous permet de ne conserver que les attributs qui vous intéressent

```
SELECT Dossard, Nom --ou * si vous les voulez tous
```

Vous pouvez en profiter pour les renommer à l'affichage : les « alias »

```
SELECT Dossard, Nom as NomPatronimique
```

Vous pouvez appliquer des fonctions scalaires et des calculs sur les attributs

```
SELECT (Dossard * 3) + 1, LEFT (Nom, 3) as PetitNomPatronimique
```

Vous pouvez forcer l'élimination des doublons apparus potentiellement

```
SELECT DISTINCT Prenom
```



Table(s) Source(s)

Il faut évidemment définir d'où proviennent vos attributs avec le mot clé FROM

```
SELECT Dossard, Nom FROM Coureur
```

```
SELECT * FROM Club
```

Il faudra lever les ambiguïtés quand il y aura plusieurs tables en jeu

→ Vous pouvez donner un alias à la table et s'y référer au niveau des attributs

```
SELECT c1.Dossard, c1.Nom FROM Coureur as c1
```

→ Ou vous pouvez utiliser une notation explicite en préfixant l'attribut par sa table d'appartenance

```
SELECT Coureur.Dossard, Coureur.Nom FROM Coureur
```



Tri des résultats

Vous pouvez utiliser les attributs pour trier les résultats avec le mot clé ORDER BY

```
SELECT Dossard, Nom FROM Coureur ORDER BY Dossard
```

Vous pouvez choisir de trier dans l'ordre croissant (par défaut), ou décroissant

```
SELECT Dossard, Nom FROM Coureur ORDER BY Dossard ASC      --tri croissant
```

```
SELECT Dossard, Nom FROM Coureur ORDER BY Dossard DESC     --tri décroissant
```

Vous pouvez opérer le tri à plusieurs niveaux, en cas de valeurs identiques

```
SELECT Dossard, Nom FROM Coureur ORDER BY LEFT (Nom, 3), Dossard ASC
```



Filtrage des résultats

Vous pouvez éliminer les n-uplets qui ne vous intéressent pas avec le mot clé WHERE

→ Dis autrement, vous cherchez à conserver ceux qui satisfont une condition logique déterminée

```
SELECT Dossard, Nom FROM Coureur WHERE Dossard > 102
```

Vous pouvez exprimer des conditions d'inégalité, de ressemblance ou de nullité

```
SELECT Dossard, Nom FROM Coureur WHERE Nom = "Le Goaër"
```

```
SELECT Dossard, Nom FROM Coureur WHERE Nom like "*Go??r" --Cf. RegExp
```

```
SELECT Dossard, Nom FROM Coureur WHERE Club is NULL
```

Vous pouvez construire des conditions complexes à l'aide de connecteurs logiques

```
SELECT Dossard, Nom FROM Coureur WHERE (not (LEFT (Nom, 3) = "Le ") or (Dossard >= 50 and Dossard <= 300))
```



Jointure (interne)

Sert à reconstituer l'information éclatée entre deux tables, à l'aide du mot clé JOIN

- Produit cartésien : combinaison de chaque n-uplet d'une table avec ceux de l'autre table
- Un critère spécial doit être satisfait pour les apparier deux à deux : la condition de jointure

```
SELECT * FROM Coureur INNER JOIN Club ON Coureur.club = Club.code
```

La condition de jointure après le mot clé ON n'est pas forcément basique ni naturelle

```
SELECT * FROM Coureur INNER JOIN Club ON LEFT (Coureur .nom, 3) = Club. code
```

Ce mécanisme est généralisable à n tables, mais exige une rigueur syntaxique

```
SELECT * FROM (((T1 INNER JOIN T2 ON Cond1)
                INNER JOIN T3 ON Cond2)
                INNER JOIN T4 ON Cond3)
```



Jointure (Externe)

Parfois, ne pas réussir à reconstituer l'information éclatée entre deux tables, est une information en elle-même ! *

La non-satisfaction du critère de jointure fait apparaître des valeurs nulles (NULL) sur des attributs appariés, qui pourront par exemple servir à filtrer

Une table parmi les 2 est considérée comme table de référence lors de cette jointure, selon qu'elle se trouve à gauche ou à droite dans le sens d'écriture de votre requête

```
SELECT * FROM Coureur RIGHT OUTER JOIN Club ON Coureur.Club = Club.code WHERE Coureur. Dossard is NULL
```

```
SELECT * FROM Club LEFT OUTER JOIN Coureur ON Coureur. Club = Club.code WHERE Coureur. Dossard is NULL
```

--100% équivalent à ci-dessus



Groupes et fonctions de groupes

L'ensemble des n-uplets forme un groupe par défaut, sur lequel il est possible d'appliquer des « fonctions de groupes »

- Il s'agit de fonctions spéciales d'agrégation statistiques
- Fonctions natives : AVG(), COUNT(), MIN(), MAX(), SUM(), STDEV(), VAR()

```
SELECT MAX(Dossard) FROM Coureur
```

Vous pouvez explicitement former des sous-groupes, à plusieurs niveaux, avec GROUP BY

```
SELECT Club, COUNT(Dossard) FROM Coureur GROUP BY Club
```

La sélection peut contenir tout au plus les attributs ayant servis à grouper, mais autant de fonctions de groupes que vous le souhaitez.



Groupes et fonctions de groupes

Si un filtrage doit porter sur une fonction de groupe, elle ne peut figurer derrière WHERE.

Elle est introduite par la clause HAVING

```
SELECT Club, COUNT(Dossard) as Inscrits FROM Coureur GROUP BY Club HAVING COUNT(Dossard) > 2
```

Il est fréquent d'appliquer des alias aux fonctions de groupe.

Toutefois, ces alias ne peuvent pas être référencés depuis le HAVING.



Sous-requête

Les expressions à l'intérieur d'une requête peuvent être remplacée par...une requête

```
SELECT * FROM Coureur WHERE Dossard < (SELECT MAX(Dossard) FROM Coureur)
```

Si la sous-requête ne renvoie pas une valeur unique mais des n-uplets, alors il faut utiliser les opérateurs IN, EXISTS, ANY, ALL

```
SELECT * FROM Coureur WHERE Club IN (SELECT Code FROM Club WHERE Pro?=Yes)
```

Parfois, la sous-requête peut être corrélée : elle ne s'exécute pas indépendamment

- Ses valeurs dépendent alors de celles de la requête hôte,
- Les alias de tables sont incontournables pour lever les ambiguïtés.



Exécution des requêtes

SQL est un langage **déclaratif**: il dit ce qu'il faut faire, mais pas comment le faire !

C'est le SGBD-R qui choisira la façon optimale d'exécuter votre requête. Néanmoins, l'ordre logique de traitement est grosso modo le suivant :

1. FROM

6. HAVING

2. ON

7. SELECT

3. JOIN

8. DISTINCT

4. WHERE

9. ORDER BY

5. GROUP BY



Exercices



Partie 1 - Livres

id_livre	titre	isbn_10	auteur	prix
1	Forteresse digitale	2709626306	Dan Brown	20.5
2	La jeune fille et la nuit	2253237620	Guillaume Musso	21.9
3	T'choupi se brosse les dents	2092589547	Thierry Courtin	5.7
4	La Dernière Chasse	2226439412	Jean-Christophe Grangé	22.9
5	Le Signal	2226319484	Maxime Chattam	23.9

Table : livres



Exercice 1

Partie 1 - Livres



Quelle requête utiliser pour afficher l'ensemble des enregistrements de la table **livres** ?

Exercice 2

Partie 1 - Livres



Quelle requête utiliser pour sélectionner uniquement les livres qui ont un prix strictement supérieur à 20 de la table livres ?

Exercice 3

Partie 1 - Livres



Quelle requête utiliser pour trier les enregistrements de la table **livres** du prix le plus élevé aux prix le plus bas ?

Exercice 4

Partie 1 - Livres

Quelle requête utiliser pour récupérer le prix du livre le plus élevé de la table **livres** ?



Exercice 5

Partie 1 - Livres



Quelle requête utiliser pour récupérer les livres de la table **livres** qui ont un prix compris entre 20 et 22 ?

Exercice 6

Partie 1 - Livres



Quelle requête utiliser pour récupérer tous les livres de la table **livres** à l'exception de celui portant la valeur pour la colonne isbn_10 : 2092589547 ?

Exercice 7

Partie 1 - Livres

Quelle requête utiliser pour récupérer le prix du livre le moins élevé de la table **livres** en renommant la colonne dans les résultats par minus ?



Exercice 8

Partie 1 - Livres

Quelle requête utiliser pour sélectionner uniquement les 3 premiers résultats sans le tout premier de la table [livres](#) ?



Partie 2 - Examens

id_etudiant	prenom	nom	id	id_examen	id_etudiant	matiere
30	Joseph	Biblo	788	45	30	Histoire-Geographie
31	Paul	Bismuth	789	87	33	Mathématiques
32	Jean	Michel	790	87	34	Mathématiques
33	Ted	Bundy	791	45	31	Histoire-Geographie
34	Caroline	Martine	792	45	32	Histoire-Geographie
35	Joséphine	Henry	793	87	31	Mathématiques
			794	45	33	Histoire-Geographie
			795	45	36	Histoire-Geographie
			796	45	34	Histoire-Geographie
			797	87	30	Mathématiques

Table : etudiants

Table : examens



Exercice 1

Partie 2 - Examens

Quelle requête utiliser pour afficher
l'id des étudiants qui ont participé à
au moins un examen ?



Exercice 2

Partie 2 - Examens



Quelle requête utiliser pour compter le nombre d'étudiants qui ont participé à au moins un examen ?

Exercice 3

Partie 2 - Examens



Quelle requête utiliser pour calculer la moyenne de l'examen portant l'id : 45 ?

Exercice 4

Partie 2 - Examens

Quelle requête utiliser pour récupérer la meilleure note de l'examen portant l'id : 87 ?



Exercice 5

Partie 2 - Examens



Quelle requête utiliser pour afficher
l'id des étudiants qui ont eu plus de 11
à l'examen 45 ou plus de 12 à l'examen
87 ?

Exercice 6

Partie 2 - Examens



Quelle requête utiliser pour afficher tous les enregistrements de la table `examens` avec en plus, si c'est possible, le prénom et le nom de l'étudiant ?

Exercice 7

Partie 2 - Examens



Quelle requête utiliser pour afficher les enregistrements de la table `examens` avec le prénom et le nom de l'étudiant, uniquement quand les étudiants sont présents dans la table `etudiants` ?

Exercice 8

Partie 2 - Examens

Quelle requête utiliser pour afficher uniquement le nom et le prénom de l'étudiant avec l'id : 30 avec la moyenne de ses deux examens dans une colonne moyenne ?



Exercice 9

Partie 2 - Examens

Quelle requête utiliser pour afficher les 3 meilleurs examens, du meilleur au moins bon, avec le prénom et le nom de l'étudiant associé ?



Partie 3 - Films

id	nom	prenom	sexe	nation
16	Scott	Ridley	0	uk
22	Aronofsky	Darren	0	us
47	Jenkins	Patty	1	us
66	Ritchie	Guy	0	uk

Table : realisateurs

id	nom	id_realisateur	id	id_film	note
121	Requiem for a Dream	22	1	546	4.5
546	Gladiator	16	2	546	2.5
666	Fight Club	61	3	775	5
775	Blade Runner	16	4	984	3.5
984	Seul sur Mars	16	5	987	3.1
986	Black Swan	22	6	666	4.2
987	Wonder Woman	47	7	986	3
988	The Tomorrow Man	85	8	986	4.3
			9	121	1

Table : films

Table : films_notes



Exercice 1

Partie 3 - Films



Quel est le résultat de la requête ci-dessous ?

```
SELECT id, prenom, nom  
FROM realisateurs  
WHERE nation = "us"  
AND sexe = 1;
```

Exercice 2

Partie 3 - Films



Quel est le résultat de la requête ci-dessous ?

```
SELECT *  
FROM realisateurs  
WHERE sexe = "0"  
ORDER BY nom DESC  
LIMIT 1;
```

Exercice 3

Partie 3 - Films



Quel est le résultat de la requête ci-dessous ?

```
SELECT f.id, f.nom AS film,  
r.prenom, r.nom  
FROM films f  
INNER JOIN realisateurs r ON  
f.id_realisateur = r.id  
ORDER BY f.id ASC;
```

Exercice 4

Partie 3 - Films



Quel est le résultat de la requête ci-dessous ?

```
SELECT f.id, f.nom AS film,  
r.prenom, r.nom  
FROM films f  
LEFT JOIN realisateurs r ON  
f.id_realisateur = r.id  
ORDER BY f.id ASC;
```

Exercice 5

Partie 3 - Films



Quel est le résultat de la requête ci-dessous ?

```
SELECT f.id, f.nom, fn.note  
FROM films f  
LEFT JOIN films_notes fn ON f.id =  
fn.id_film  
ORDER BY f.id ASC;
```

Exercice 6

Partie 3 - Films



Quel est le résultat de la requête ci-dessous ?

```
SELECT f.nom, r.prenom AS  
realisateur_prenom, r.nom AS  
realisateur_nom, AVG(fn.note) AS  
moyenne_note  
FROM films f  
INNER JOIN realisateurs r ON  
f.id_realisateur = r.id  
INNER JOIN films_notes fn ON f.id  
= fn.id_film  
WHERE f.id = 546;
```

Exercice 8

Partie 3 - Films



Quel est le résultat de la requête ci-dessous ?

```
SELECT r.nation, MAX(fn.note) AS  
max_note  
FROM films f  
INNER JOIN realisateurs r ON  
f.id_realisateur = r.id  
INNER JOIN films_notes fn ON f.id  
= fn.id_film  
WHERE r.nation = "uk";
```

Partie 4 - Géographie

id	department_code	insee_code	zip_code	name	gps_lat
30834	77	77205	77165	Gesvres-le-Chapitre	49.03256314814814
30966	77	77341	77114	Noyen-sur-Seine	48.45316787878787
35499	95	95270	95420	Genainville	49.12055698924732
35563	95	95500	95000	Pontoise	49.03316710000000
31267	78	78317	78440	Jambville	49.0495396296296

Table : cities

id	region_code	code	name	slug
41	75	40	Landes	landes
101	06	976	Mayotte	mayotte
15	84	15	Cantal	cantal
7	84	07	Ardèche	ardeche
70	84	69	Rhône	rhone

Table : departments

id	code	name	slug
16	84	Auvergne-Rhône-Alpes	auvergne rhone alpes
14	75	Nouvelle-Aquitaine	nouvelle aquitaine
10	32	Hauts-de-France	hauts de france
11	44	Grand Est	grand est
6	11	Île-de-France	ile de france

Table : regions



Exercice 1

Partie 4 - Géographie

Quelle requête utiliser pour retrouver la ville qui possède les coordonnées GPS suivantes : 48.66913724637683, 1.87586057971015 ?



Exercice 2

Partie 4 - Géographie



Sans jointure, quelle requête utiliser pour calculer le nombre de villes que compte le département de l'Essonne ?

Exercice 3

Partie 4 - Géographie



Sans jointure, quelle requête utiliser pour calculer le nombre de villes en Île-de-France se terminant par **"-le-Roi"** ?

Exercice 4

Partie 4 - Géographie



Combien de villes possèdent le code postal (`zip_code`) 77320 ? Renommez la colonne de résultat `n_cities`.

Exercice 5

Partie 4 - Géographie



Sans jointure, quelle requête utiliser pour calculer le nombre de villes commençant par "**Saint-**" en Seine-et-Marne ?

Exercice 6

Partie 4 - Géographie



Quelles villes possèdent un code postal ([zip_code](#)) compris entre 77210 et 77810 ?

Exercice 7

Partie 4 - Géographie



Sans jointure, quelles sont les deux villes de Seine-et-Marne à avoir le code postal ([zip_code](#)) le plus grand ?

Exercice 8

Partie 4 - Géographie



Quel est le code postal ([zip_code](#)) le plus grand de la table lpecom_cities ?

Exercice 9

Partie 4 - Géographie



Avec un seul **WHERE** et aucun **OR**, quelle est la requête permettant d'afficher les départements des régions ayant le code suivant : 75, 27, 53, 84 et 93 ?

Le résultat doit afficher le nom du département ainsi que le nom et le slug de la région associée.

Exercice 10

Partie 4 - Géographie

Quelle requête utiliser pour obtenir en résultat, les noms de la région, du département et de chaque ville du département ayant pour `code` 77 ?



Partie 5 - Covid19

id	id_region	jour	n_dose1	n_dose2	n_cum_dose1	n_cum_dose2	coun
404	04	2021-04-06	676	633	40066	22082	4.70
202	02	2021-04-06	615	104	18330	5236	5.10
303	03	2021-04-06	301	300	10572	5199	3.60
505	06	2021-04-06	125	184	10236	4781	3.70
101	01	2021-04-06	227	166	10503	4027	2.80

Table : covid

id	code	name	slug
9	28	Normandie	normandie
6	11	Île-de-France	ile de france
4	04	La Réunion	la reunion
15	76	Occitanie	occitanie
12	52	Pays de la Loire	pays de la loire

Table : regions



Exercice 1

Partie 5 - Covid19



Quelle requête utiliser pour afficher
toutes les données de vaccination
uniquement pour le 1er avril 2021 ?

Exercice 2

Partie 5 - Covid19

Quelle requête utiliser pour afficher toutes les données de vaccination uniquement pour le 1er avril 2021 avec le nom de la région concernée ?



Exercice 3

Partie 5 - Covid19



Quelle requête utiliser pour afficher le nombre au cumulé de vaccination première dose toutes régions en 2020 ?

Proposez également une solution pour les vaccination deuxième dose.

Exercice 4

Partie 5 - Covid19

Quelle requête SQL utiliser pour afficher le nombre au cumulé de vaccination première dose pour la région avec le code 93 uniquement pour le mois de mars 2021 ?



Exercice 5

Partie 5 - Covid19



Quelle requête utiliser pour afficher le nombre au cumulé de vaccination deuxième dose pour la région avec le code **11** uniquement pour le mois de mars 2021 ?

Exercice 6

Partie 5 - Covid19



Quelle requête utiliser pour afficher le record de vaccination première dose en une seule journée ?

Avec une deuxième requête, afficher les informations de la région concernée, dont son nom, ainsi que le jour du record.

Exercice 7

Partie 5 - Covid19



Quelle requête utiliser pour afficher le record de vaccination deuxième dose en une seule journée ?

Avec une deuxième requête, afficher les informations de la région concernée, dont son nom, ainsi que le jour du record.

Exercice 8

Partie 5 - Covid19



Quelles requêtes permettent de connaître quelle région possède la plus grande couverture de vaccination avec une dose et deux doses ?

Vous aurez besoin de 4 requêtes pour répondre aux deux questions.

Vous aurez besoin du résultat de la première requête pour la deuxième.

Exercice 9

Partie 5 - Covid19



Quelle requête utiliser pour afficher le nom de la région qui a le plus faible taux de couverture de vaccination avec une dose ?

Vous aurez besoin de 2 requêtes pour répondre à la question.

Exercice 10

Partie 5 - Covid19



Quelle requête utiliser pour calculer la couverture moyenne entre les différentes régions à la date la plus récente, pour les vaccinations une et deux doses ?

Vous renommez les colonnes de résultats : `couverture_dose1_avg` et `couverture_dose2_avg`.

Exercice 11

Partie 5 - Covid19

Quelle requête utiliser pour afficher les données de vaccination des régions (avec leur nom) qui possèdent une couverture vaccinale supérieure à 15 % pour la première dose et supérieure à 5 % pour la deuxième dose ?



Partie 6 - Vaccins

id	dep_code	vaccin	jour	n_dose1	n_dose2	n_cum_dose1	n_cum_do
2526	75	0	2021-04-06	5273	3457	370829	116607
102	75	1	2021-04-06	4114	3384	240541	109520
203	75	2	2021-04-06	3	70	7579	6946
304	75	3	2021-04-06	1156	3	122709	141
2627	77	0	2021-04-06	2626	1915	142547	44880

Table : covid_vaccin

id	nom
0	Tous vaccins
1	COMIRNATY Pfizer/BioNTech
2	Moderna
3	AstraZeneka

Table : covid_vaccin_type

id	region_code	code	name	slug
105	COM	984	Terres australes et antarctiques françaises	terres australes et antarctiques francaises
67	76	66	Pyrénées-Orientales	pyrenees orientales
20	27	21	Côte-d'Or	cote dor
30	94	2B	Haute-Corse	haute corse
58	44	57	Moselle	moselle

Table : departments



Exercice 1

Partie 6 - Vaccins



Sans jointure, quelle requête SQL utiliser pour afficher toutes les données de vaccination du 14 février 2021 uniquement, pour le département de Seine-et-Marne (77) ?

Exercice 2

Partie 6 - Vaccins

Sans jointure, quelle requête SQL utiliser pour afficher le cumul de toutes les données de vaccination pour tous les vaccins du 14 février 2021 uniquement, pour les départements de l'Essonne (91) et de la Seine-et-Marne (77) ?



Exercice 3

Partie 6 - Vaccins



Sans jointure, quelle requête utiliser pour afficher la somme des vaccinations première dose réalisées uniquement avec le vaccin [AstraZeneka](#) pour le mois de [février 2021](#) pour le département de la Seine-et-Marne ([77](#)) ?

Exercice 4

Partie 6 - Vaccins



Sans jointure, quelle requête utiliser pour afficher la somme des vaccinations deuxième dose réalisées avec le vaccin [AstraZeneka](#) ou [Moderna](#) pour le mois de [mars 2021](#) pour le département de la Seine-et-Marne ([77](#)) ?

Exercice 5

Partie 6 - Vaccins



Sans jointure, quelle requête utiliser pour afficher **le record de vaccination première dose** avec un type de vaccin en **une seule journée** ?

Avec une deuxième requête qui exploitera une jointure, afficher toutes les informations possibles pour cette journée record et sur le type de vaccin.

Exercice 6

Partie 6 - Vaccins



Sans jointure, quelle requête utiliser pour afficher **le record de vaccination deuxième dose** avec un type de vaccin en **une seule journée** ?

Avec une deuxième requête qui exploitera deux jointures, afficher toutes les informations possibles pour cette journée record, sur le type de vaccin et sur le département.

Exercice 7

Partie 6 - Vaccins



Quelle requête permet de savoir quel département possède le plus grand nombre d'injections première dose pour le vaccin AstraZeneka ?

Avec une deuxième requête, afficher uniquement les colonnes suivantes :

- le nom du vaccin
- le jour
- le nom et le code du département
- → le nombre cumulé d'injections.

Exercice 8

Partie 6 - Vaccins



Quelle requête permet de savoir quel département a eu **le moins de vaccinations première dose** avec le vaccin **COMIRNATY Pfizer/BioNTech** ?

Avec une deuxième requête, afficher uniquement les colonnes suivantes :

- le nom du vaccin
- le jour
- le nom et le code du département
- le nombre cumulé d'injections.

Exercice 9

Partie 6 - Vaccins



Quelle requête permet de connaître la moyenne de vaccinations `première dose` dans tous les départements pour le vaccin `Moderna` ?

Renommer la colonne de résultat avec `avg_moderna`.

Exercice 10

Partie 6 - Vaccins



Quelle requête utiliser pour afficher les départements (avec leur nom) qui possèdent un nombre d'injections deuxième dose avec le vaccin

Moderna supérieur à **9000** ou un nombre d'injections première dose avec le vaccin **COMIRNATY** **Pfizer/BioNTech** supérieur à **120000** ?

Vous aurez besoin de deux jointures.

Partie 7 - Répertoire Partagé des Professionnels de Santé

id	id_pp_nat	nom	prenom	code_profession	lib_profession
4636	810006134398	REYNAUD	MARINE	80	Pédicure- Podologue
4419	810003708426	ROUILLARD	PHILIPPE	10	Médecin
2638	810001277358	NODARIAN	MARILYNE	10	Médecin
467	810100059467	BENZIANE	KENZA	10	Médecin
1899	810102313565	DERBEL	HAYTHAM	10	Médecin

Table : rpps



Exercice 1

Partie 7 - RPPS



Quelle requête utiliser pour compter, sans doublons, le nombre de professionnels de santé en Seine-et-Marne (77) ?

Exercice 1

Partie 7 - RPPS



Quelle requête utiliser pour afficher pour tous les professionnels de santé avec le code postal 77300 les colonnes suivantes : `id_pp_nat`, `prenom`, `nom`, `code_postal`, `ville`, `departement` et `région`.

Vous aurez besoin de plusieurs jointures.
