

# Annales de concours sur les Bases de Données

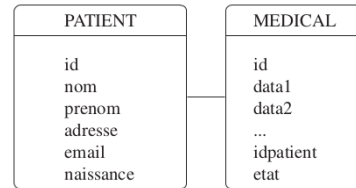
## 1) CCINP 2019, filière PSI

Il s'agit d'un problème d'application d'une intelligence artificielle à la médecine, plus particulièrement pour détecter les affections du rachis (la colonne vertébrale). Les questions sur les BDD sont au début du sujet.

La base de données médicale contient des informations administratives sur les patients et des informations médicales. Pour simplifier le problème, on considère deux tables : PATIENT et MEDICAL.

La table PATIENT contient les attributs suivants :

- id : identifiant d'un individu (entier), clé primaire ;
- nom : nom du patient (chaîne de caractères) ;
- prenom : prénom du patient (chaîne de caractères) ;
- adresse : adresse du patient (chaîne de caractères) ;
- email : (chaîne de caractères) ;
- naissance : année de naissance (entier).



La table MEDICAL contient les attributs suivants :

- id : identifiant d'un ensemble de propriétés médicales (entier), clé primaire ;
- data1 : donnée (flottant) ;
- data2 : donnée (flottant) ;
- ... ;
- idpatient : identifiant du patient représenté par l'attribut id de la table PATIENT (entier) ;
- etat : description de l'état du patient (chaîne de caractères).

Les attributs data1, data2... sont des données relatives à l'analyse médicale souhaitée (dans notre cas des données biomécaniques). L'attribut « etat » permet d'affecter un label à un ensemble de données médicales : « normal », « hernie discale », « spondylolisthésis ».

- Q1.** Écrire une requête SQL permettant d'extraire les identifiants des patients ayant une « hernie discale ».
- Q2.** Écrire une requête SQL permettant d'extraire les noms et prénoms des patients atteints de « spondylolisthésis ».
- Q3.** Écrire une requête SQL permettant d'extraire chaque état et le nombre de patients pour chaque état.

## 2) Mines-Ponts, 2019, épreuve commune MP, PC, PSI

Le sujet aborde des questions relatives aux nombres premiers. La partie sur les bases de données est la dernière.

Au cours du développement des fonctions nécessaires à la manipulation des nombres premiers on s'aperçoit que le choix des algorithmes pour évaluer chaque fonction est primordial pour garantir des performances acceptables. On souhaite donc mener des tests à grande échelle pour évaluer les performances réelles du code qui a été développé. Pour ce faire on effectue un grand nombre de tests sur une multitude d'ordinateurs. Les données sont ensuite centralisées dans une base de données composée de deux tables.

La première table est **ordinateurs** et permet de stocker des informations sur les ordinateurs utilisés pour les tests. Ses attributs sont :

- **nom** TEXT, clé primaire, le nom de l'ordinateur.
- **gflops** INTEGER la puissance de l'ordinateur en milliards d'opérations flottantes par seconde.
- **ram** INTEGER la quantité de mémoire vive de l'ordinateur en Go.

Exemple du contenu de cette table :

nom	gflops	ram
nyarlathotep114	69	32
nyarlathotep119	137	32
...		
shubniggurath42	133	16
azathoth137	85	8

La seconde table est **fonctions** et stocke les informations sur les tests effectués pour différentes fonctions en cours de développement. Ses attributs sont :

- **id** INTEGER l'identifiant du test effectué.
- **nom** TEXT le nom de la fonction testée (par exemple li, Ei, etc).
- **algorithme** TEXT le nom de l'algorithme qui permet le calcul de la fonction testée (par exemple BBS si on teste une fonction de génération de nombres aléatoires).
- **teste\_sur** TEXT le nom du PC sur lequel le test a été effectué.
- **temps\_exec** INTEGER le temps d'exécution du test en millisecondes.

Exemple du contenu de cette table :

id	nom	algorithme	teste_sur	temps_exec
1	li	rectangles	nyarlathotep165	2638
2	li	rectangles	shubniggurath28	736
3	li	trapezes	nyarlathotep165	4842
...				
2154	Ei	puiseux	nyarlathotep145	2766
2155	aleatoire	BBS	azathoth145	524

**Q25** – Expliquer pourquoi il n'est pas possible d'utiliser l'attribut **nom** comme clé primaire de la table **fonctions**.

**Q26** – Écrire des requêtes SQL permettant de :

1. Connaître le nombre d'ordinateurs disponibles et leur quantité moyenne de mémoire vive.
2. Extraire les noms des PC sur lesquels l'algorithme **rectangles** n'a pas été testé pour la fonction nommée **li**.
3. Pour la fonction nommée **Ei**, trier les résultats des tests du plus lent au plus rapide. Pour chaque test retenir le nom de l'algorithme utilisé, le nom du pc sur lequel il a été effectué et la puissance du PC.

## Correction

### 1) CCINP 2019

Remarque : les tables ont des noms en majuscule, ce qu'on respecte dans le corrigé, mais ça prête à confusion avec les mots-clé du langage qui sont aussi en majuscule. Sur une copie, il serait judicieux de mettre les mots-clé en couleur.

Q1)

```
SELECT DISTINCT idpatient
FROM MEDICAL
WHERE etat = " hernie discale "
```

Le DISTINCT est facultatif : il vise à éviter les redondances si un patient apparaît plusieurs fois dans la table MEDICAL

Q2)

```
SELECT PATIENT . nom , PATIENT . prenom
FROM MEDICAL JOIN PATIENT
ON PATIENT . id = MEDICAL . idpatient
WHERE etat = " spondylolisthésis "
```

Q3)

```
SELECT etat , COUNT (*)
FROM MEDICAL
GROUP BY etat
```

Dans cette réponse, on a considéré qu'il n'y a qu'un enregistrement par patient dans la table MEDICAL.

### 2) CCMP 2019

Remarque : cette partie arrive en fin de sujet alors que c'est la plus simple du sujet. De plus, elle est déconnectée des autres : les candidats avaient eu tout intérêt à la traiter en premier. Cela arrive régulièrement, soyez vigilants de toujours avoir le temps de traiter les BDD.

Q25 ) Plusieurs enregistrements peuvent avoir le même attribut nom dans la table fonctions, il ne peut donc pas servir de clé primaire.

Q26.1)

```
SELECT COUNT (*) AS nb_orði , AVG ( RAM ) AS RAM_moyenne
FROM ordinateurs
```

Q26.2) Deux propositions de correction :

```
SELECT nom
FROM ordinateurs
WHERE nom NOT IN (
    SELECT teste_sur
    FROM fonctions
    WHERE algorithmme = " rectangles " and nom = " li "
)
```

On peut aussi faire utiliser l'opérateur de différence ensembliste qui correspond à la commande EXCEPT :

```
SELECT nom
FROM ordinateurs EXCEPT (
    SELECT teste_sur
    FROM fonction
    WHERE algorithmme = " rectangles " and nom = " li "
)
```

Q26.3)

```
SELECT algorithme , teste_sur , ram , gflops  
FROM fonctions JOIN ordinateurs  
ON fonctions . teste_sur = ordinateurs . nom  
WHERE fonctions . nom = " Ei "  
ORDER BY temps_exec DESC
```