

## Exercice 2 (5 points)

*Cet exercice porte sur les bases de données, la représentation des données*

Cet exercice utilise certains des mots-clés du langage SQL suivants : DELETE, FROM, INSERT, INTO, JOIN, ON, SELECT, SET, UPDATE, VALUES, WHERE.

Les vacances d'été se rapprochent et le propriétaire d'une pension pour animaux gère les places dont il dispose à l'aide d'une base de données dont voici le schéma relationnel :

```
client(num_client, nom_client, prenom_client, mail_client, tel_client)
animal(num_animal, nom_animal, categorie_animal, taille_animal, num_client)
cage(num_cage, taille_cage, secteur_cage)
reservation(num_reservation, date_debut_reservation, date_fin_reservation,
num_client, num_animal, num_cage)
```

Ci-dessous, on donne des extraits des tables `client`, `animal`, `cage` et `reservation`.

Extrait de la table `client` :

num_client	nom_client	prenom_client	mail_client	tel_client
16	Dupont	Marc	marc.dupont@mail.com	0604050401
345	Morel	Fabien	fabien.morel@mail.com	0700051020

Extrait de la table `animal` :

num_animal	nom_animal	categorie_animal	taille_animal	num_client
22	Yuki	souris	petit	16
112	Balou	chat	moyen	141
320	Api	chien	grand	237
423	Rex	chien	moyen	259
491	Rex	chien	petit	345

Extrait de la table `cage` :

num_cage	taille_cage	secteur_cage
4	grand	chien
12	petit	chien
23	moyen	chien
31	moyen	chien
32	petit	rongeur
33	grand	chat

Extrait de la table `reservation` :

<code>num_reservation</code>	<code>date_debut_reservation</code>	<code>date_fin_reservation</code>	<code>num_client</code>	<code>num_animal</code>	<code>num_cage</code>
44	2022-08-23	2022-08-25	26	12	12
45	2022-07-11	2022-07-22	345	491	23
46	2022-08-11	2022-08-22	345	491	23
47	2022-08-23	2022-09-10	345	491	23
48	2022-10-11	2022-10-22	345	491	23

## 1. Étude du schéma relationnel

- a. Pour chaque attribut de la relation `cage`, spécifier son type, en utilisant le tableau des types suivant :

<code>CHAR (t)</code>	Texte de longueur fixe de t caractères.
<code>VARCHAR (t)</code>	Texte de longueur variable de t caractères au maximum.
<code>INT</code>	Nombre entier de $-2^{31}$ à $2^{31}-1$ (signé) ou de 0 à $2^{32}-1$ (non signé).
<code>FLOAT</code>	Réel à virgule flottante.
<code>DATE</code>	Date format AAAA-MM-JJ.
<code>DATETIME</code>	Date et heure format AAAA-MM-JJ HH:MI:SS.

- b. Préciser, pour la relation `reservation`, le nom de la clé primaire pouvant être utilisée.
- c. Indiquer, pour la relation `reservation`, la ou les clés étrangères (ou secondaires) et en indiquer l'utilité.

## 2. Requêtes

- a. Indiquer le résultat de l'exécution de la requête suivante :

```
SELECT nom_animal
FROM animal
WHERE categorie_animal = 'chien';
```

- b. Écrire une requête SQL permettant d'afficher les noms de tous les clients dont l'animal a occupé la cage numéro 23.
- c. Un nouvel animal doit être enregistré dans la base de données qui contient actuellement 491 animaux. Il s'appelle Suki, c'est un chat de petite taille dont le propriétaire a déjà été enregistré sous le numéro 342.  
Écrire la requête SQL permettant d'insérer ces nouvelles données dans la base de données.

### 3. Programmation Python

Suite à une panne, le responsable de la pension n'a plus accès à sa base de données. Heureusement, il avait fait une sauvegarde de ses tables au format csv. Il les a importées à l'aide d'un programme Python, chacune sous la forme d'une liste de dictionnaires.

Pour simplifier, on considérera que la table `reservation` est la liste de dictionnaires suivante :

```
reservation = [  
    {'num_reservation' : 44, 'date_debut_reservation' : '2022-08-23',  
     'date_fin_reservation' : '2022-08-25', 'num_client' : 26,  
     'num_animal' : 12, 'num_cage' : 12},  
    {'num_reservation' : 45, 'date_debut_reservation' : '2022-07-11',  
     'date_fin_reservation' : '2022-07-22', 'num_client' : 345,  
     'num_animal' : 491, 'num_cage' : 23},  
    {'num_reservation' : 46, 'date_debut_reservation' : '2022-08-11',  
     'date_fin_reservation' : '2022-08-22', 'num_client' : 345,  
     'num_animal' : 491, 'num_cage' : 23},  
    {'num_reservation' : 47, 'date_debut_reservation' : '2022-08-23',  
     'date_fin_reservation' : '2022-09-10', 'num_client' : 345,  
     'num_animal' : 491, 'num_cage' : 23},  
    {'num_reservation' : 48, 'date_debut_reservation' : '2022-10-11',  
     'date_fin_reservation' : '2022-10-22', 'num_client' : 345,  
     'num_animal' : 491, 'num_cage' : 23}]
```

a. On donne ci-dessous, le code Python d'une fonction `mystere`.

Numéro de lignes	Fonction mystere
1	<code>def mystere(table, date):</code>
2	<code>    liste = []</code>
3	<code>    for ligne in table:</code>
4	<code>        if ligne['date_debut_reservation'] == date:</code>
5	<code>            liste.append(ligne['num_client'])</code>
6	<code>    return liste</code>

On rappelle que l'appel `L.append(x)` ajoute l'élément `x` à la fin de la liste `L`.

Indiquer l'affichage produit par l'exécution de la ligne de code suivante :

```
print(mystere(reservation, '2022-08-23'))
```

b. Le responsable de la pension veut obtenir le nombre de réservations qui ont été effectuées pour un numéro de client donné.

Écrire les lignes de code après la ligne 7 de la fonction `nombre_reservation` afin de respecter la spécification donnée ci-dessous

Numéro de lignes	Fonction <code>nombre_reservation</code>
1	<code>def nombre_reservation(table, numero_client):</code>
2	<code>    """Paramètres :</code>
3	<code>    table : liste de dictionnaires, représentant les réservations</code>
4	<code>    numero_client : un entier, représentant le numéro du client</code>
5	<code>    concerné</code>
6	<code>    Valeur renvoyée : un entier donnant le nombre d'occurrences</code>
7	<code>    du numéro du client concerné. """</code>
...	<i>à compléter</i>

