

Cette situation d'évaluation comporte ce document ainsi que des fichiers de codes et de données présents sur l'ordinateur à la disposition du candidat. Le candidat doit restituer ce document avant de sortir de la salle d'examen. Le candidat doit agir en autonomie et faire preuve d'initiative tout au long de l'épreuve.

En cas de difficulté, le candidat peut solliciter l'examineur afin de lui permettre de continuer la tâche. Des moments privilégiés pour solliciter l'examineur sont indiqués dans le document sous la forme d'appels professeur.

L'examineur peut intervenir à tout moment, s'il le juge utile.

Cadre

Un cabinet vétérinaire utilise une base de données pour assurer le suivi et la gestion des clients et animaux qui y sont suivis, dans le but d'améliorer la qualité des soins et de fournir des services utiles, que ce soit pour le vétérinaire ou pour ses clients.

Objectif

La vaccination régulière des chats (une fois par an) étant importante mais souvent oubliée par les propriétaires, le vétérinaire souhaiterait pouvoir envoyer des messages de rappel de vaccination par SMS, deux mois avant l'échéance (la date à laquelle la vaccination devrait avoir lieu).

Il dispose pour cela d'une plateforme d'envoi de SMS, mais il est nécessaire de préparer les données et les messages à envoyer.

Description de la base de données

Le logiciel mis en œuvre par le cabinet utilise 3 tables pour enregistrer les animaux, leur propriétaire et les consultations.

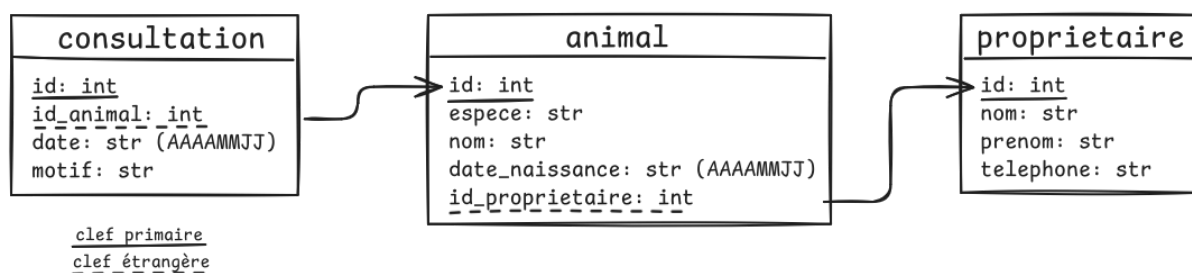


Schéma de la base de données

Extraits de la table proprietaire

| id | nom | prenom | telephone |
|----|---------|----------|-----------------|
| 1 | Duhamel | François | 06-57-84-52-41 |
| 2 | Dumont | Marcelle | 0648747223 |
| 3 | Joubert | Anouk | 0 4 76 33 78 15 |

Extraits de la table consultation

| id | id_animal | date | motif |
|----|-----------|----------|-------------|
| 1 | 1 | 20171229 | vaccination |
| 25 | 4 | 20050621 | bilan |
| 34 | 5 | 20131106 | maladie |

Extraits de la table animal

| id | espece | nom | date_naissance | id_proprietaire |
|----|--------|-----------|----------------|-----------------|
| 1 | oiseau | Flocon | 20171111 | 1 |
| 2 | lapin | Roudoudou | 20210901 | 2 |
| 3 | chat | Mistral | 20251206 | 2 |

Extraits de la base de données

Le système de gestion de base de données utilisé est SQLite, et il est possible d'interagir avec la base de données en utilisant le module sqlite, disponible dans la bibliothèque standard python.

Rappels techniques

La connexion à la base de données se fait en précisant l'emplacement du fichier de la base de données. Par exemple, si le fichier cabinet.sqlite se trouve dans le répertoire courant, on écrit :

```
DB_PATH = "cabinet.sqlite"
conn = sqlite3.connect(DB_PATH)
```

Une fois que l'on dispose d'une connexion (variable conn), on utilise un curseur (variable cursor) pour exécuter des requêtes sur la base de données et éventuellement récupérer les résultats de la requête.

```
cursor = conn.cursor()
resultats = cursor.execute(
    "SELECT id, nom FROM animal ORDER BY nom LIMIT 10"
)
# resultats permet de récupérer les lignes de manière
# progressive, sous forme d'une liste itérable de tuples
# correspondant aux champs demandés
# Pour accéder aux champs sélectionnés
for r in resultats:
# les valeurs renvoyées par le SELECT sont
# accessibles par leur position
    print("id:", r[0], ", nom:", r[1])
```

Lorsqu'un paramètre doit être donné dans une requête SQL, il est fourni sous forme d'un ? dans le texte de la requête SQL, et passé en argument à la fonction execute sous forme d'un tuple (chaque ? correspond à un élément du tuple, dans l'ordre). Par exemple, pour récupérer les chats du propriétaire d'id 2 :

```
resultat = cursor.execute(
    """SELECT nom FROM animal
    WHERE espece = 'chat' AND id_proprietaire = ?""",
    (2,)
)
```

Les clauses de langage SQL à utiliser sont : SELECT, FROM, JOIN, ON, WHERE, ORDER BY

Numéros de téléphone

La plateforme d'envoi de SMS requiert un numéro de téléphone au format 06XXXXXXXX ou 07XXXXXXXX, 10 chiffres au total, sans espace, ni point, ni autre caractère. Malheureusement, les téléphones ont été enregistrés dans la base sans distinguer les téléphones mobiles des téléphones fixes, et en utilisant différents styles comme "(0)6 12 34 56 78" ou "01.23.45.67.89".

Rappel : on peut utiliser la méthode `isdigit` de la classe `str` pour déterminer si un caractère est un chiffre

```
>>> '2'.isdigit()
True
>>> 'C'.isdigit()
False
```

Question 1

Écrire une fonction python `normalisation_tel` qui prend un numéro de téléphone `tel` de type `str` en argument et qui renvoie le numéro de téléphone nettoyé en ne gardant que les chiffres. Exemples :

```
normalisation_tel("(0)2 12 99 90 12") # renvoie "0212999012"  
normalisation_tel("0.6.12.99.90.12") # renvoie "0612999012"  
normalisation_tel("0-6-12-9-90-1") # renvoie "06129901"
```

Une fonction `test_normalisation_tel` est fournie.



Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

Validation des numéros de téléphone

La fonction `validation_tel(tel)` prend comme argument un numéro de téléphone nettoyé, et renvoie un booléen selon que ce numéro est un numéro de téléphone portable valide ou non.

Question 2

Écrire un jeu de tests permettant de vérifier le bon fonctionnement de la fonction.



Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

Détermination de la liste des chats à vacciner

La vaccination se faisant tous les ans, le vétérinaire souhaite envoyer trois rappels aux propriétaires. Chaque rappel sera donc envoyé aux propriétaires des chats ayant été vaccinés pour la **dernière** fois il y a entre 10 et 13 mois.

Pour cela on commencera par déterminer quels chats ont été vaccinés il y a **moins de 13 mois**, à l'aide de la fonction `consultation_vaccination_chat(date)`.

Puis à l'aide de la fonction `derniere_vaccination(consultations)`, on déterminera la date de **dernière** vaccination de ces chats.

Enfin, dans la dernière étape du projet, se fera le filtrage des chats ayant été vaccinés pour la **dernière** fois il y a plus de 10 mois.

Interrogation de la base de données

Pour interroger la base de données, on utilise la fonction `consultation_vaccination_chat(date)` qui doit récupérer toutes les consultations de motif vaccination des chats (espece : chat) du cabinet, pour des dates supérieures à une date, `date`, exprimée au format AAAAMMJJ (ce format permet de comparer des dates en utilisant l'ordre alphabétique).

On souhaite récupérer les champs suivants :

- id de l'animal ;
- nom de l'animal ;

- telephone du propriétaire ;
- date de la consultation.

triés par le champ id d'animal puis date de consultation croissants.

Question 3

En vous inspirant de la fonction `proprietaires_animaux_nes_apres(date)`, écrire la fonction `consultation_vaccination_chat(date)`.
Des tests sont fournis dans la fonction `test_consultation_vaccination_chat`, votre fonction doit les passer.



Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

Détermination de la date de dernière vaccination

Pour déterminer la date de dernière vaccination, on utilise la fonction `derniere_vaccination` qui prend en paramètre une liste de consultations avec les champs précédents (id de l'animal, nom de l'animal, id_proprietaire de l'animal, date de la consultation). Elle renvoie un dictionnaire associant comme clef l'id de chaque chat avec les informations de sa **dernière** (plus récente) consultation.

Exemple : pour Plume et Gollum

```
(16, 'Plume', '0.6.36.96.89.83', '20241024'),  
(16, 'Plume', '0.6.36.96.89.83', '20251125'),  
(17, 'Gollum', '0.6.36.96.89.83', '20250113'),
```

La dernière consultation de Plume est le 20251125, et pour Gollum, c'est le 20250113 donc `derniere_vaccination` doit donc renvoyer le dictionnaire :

```
{  
16: (16, 'Plume', '0.6.36.96.89.83', '20251125'),  
17: (17, 'Gollum', '0.6.36.96.89.83', '20250113')  
}
```

Cependant, la fonction ne produit pas les résultats attendus : les tests fournis dans la fonction `test_derniere_vaccination` échouent.

Question 4

Expliquer le problème rencontré, puis comment corriger cette fonction pour qu'elle passe les tests avec succès.



Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

Description du dossier

Le dossier fourni au candidat sur l'ordinateur comporte les éléments suivants :

- une version PDF de l'énoncé ;
- un code source de départ `veto.py`.
- une base de données `cabinet.sqlite`

Préparation de l'environnement

Environnement de gestion des bases de données : SQLite

Pour faire fonctionner le code fourni dans le dossier, les bibliothèques suivantes doivent être présentes : `sqlite3`.