

Asie – 2025 – sujet1 - Correction

Exercice 3 (8 points)

Partie A – Base de données et SQL

1. Numéros des kimonos en cours de location

👉 Un kimono est **en cours de location** si l'attribut `fin` est vide ('').

```
SELECT id-kimono
FROM location
WHERE fin = '';
```

Commentaire : On exploite ici la convention donnée dans l'énoncé :

fin peut être égal à la chaîne vide pour les locations en cours.

2. Nombre de kimonos de taille 130 cm

```
SELECT COUNT(id-kimono)
FROM kimono
WHERE taille-kimono = 130;
```

Commentaire : La fonction d'agrégation COUNT permet de compter le nombre d'enregistrements correspondant à une condition donnée.

3. Nom et prénom de l'adhérent qui possède le kimono 42 (si location en cours)

```
SELECT nom, prenom
FROM adherent
JOIN location ON adherent.numero-licence = location.numero-licence
WHERE location.id-kimono = 42
AND location.fin = '';
```

Commentaire :

- Le JOIN permet de relier les tables `adherent` et `location`
- Le test `fin = ''` garantit qu'il s'agit de la **location actuelle**

- On ne renvoie **que le nom et le prénom**, comme demandé
-

4. Mise à jour de la taille des adhérents mesurant moins de 160 cm

```
UPDATE adherent
SET taille-adherent = taille-adherent + 10
WHERE taille-adherent < 160;
```

Commentaire :

- On utilise une **mise à jour conditionnelle**
 - L'expression $a = a + 10$ est explicitement autorisée par l'énoncé
-

5. Suppression du kimono numéro 25

⚠ Le kimono peut être référencé dans la table `location`.
Il faut donc supprimer **dans le bon ordre**.

```
DELETE FROM location
WHERE id-kimono = 25;
```

```
DELETE FROM kimono
WHERE id-kimono = 25;
```

Commentaire :

- On supprime d'abord les références (clé étrangère)
 - Puis l'enregistrement dans la table `kimono`
-

Partie B – Dictionnaires et programmation Python

6. Numéro de licence possible pour Eddie Nirrer

Données :

- Prénom : Eddie
- Nom : Nirrer → **NIRRE**
- Sexe : supposé masculin → **M**
- Date de naissance : 12/10/2021 → **12102021**
- Numéro de différenciation : par exemple **01**

Réponse possible : M12102021NIRRE01

Commentaire : Plusieurs réponses sont possibles tant que la structure est respectée.

7. Exploitation du numéro M23091974MARTI01

- Sexe : **M**
- Date de naissance : **23/09/1974**
- Nom possible : **MARTIN, MARTINEZ, MARTIAL, etc.**

Réponse :

- Date de naissance : **23 septembre 1974**
 - Nom de famille possible : **MARTIN**
-

8. Valeur de `tab_adherents[1]['prenom']`

'STEPHANIE'

Commentaire :

- `tab_adherents[1]` désigne le **deuxième dictionnaire**
 - On accède ensuite à la clé `'prenom'`
-

9. Instruction permettant d'obtenir 'F03071997DUPON01'

```
tab_adherents[0]['numero-licence']
```

Commentaire : C'est le numéro de licence du **premier adhérent** du tableau.

10. Fonction `nombre_adherents`

```
def nombre_adherents(table, annee):  
    compteur = 0  
    for adherent in table:  
        if adherent['annee'] == annee:  
            compteur = compteur + 1  
    return compteur
```

Commentaire :

- *Parcours complet de la table*

- *Test sur l'année de naissance*
 - *Incrément d'un compteur*
-

11. Fonction `adherent_plus_age`

```
def adherent_plus_age(table):
    annee_min = table[0]['annee']
    for adherent in table:
        if adherent['annee'] < annee_min:
            annee_min = adherent['annee']

    liste = []
    for adherent in table:
        if adherent['annee'] == annee_min:
            liste.append(adherent)

    return liste
```

Commentaire :

- *On cherche d'abord l'année la plus ancienne*
 - *Puis on construit la liste des adhérents correspondants*
 - *Deux parcours → solution claire et robuste*
-

12. Fonction `verification_licence`

```
def verification_licence(adherent):
    licence = adherent['numero-licence']

    sexe = extraire(licence, 0, 1)
    jour = extraire(licence, 1, 3)
    mois = extraire(licence, 3, 5)
    annee = extraire(licence, 5, 9)
    nom = extraire(licence, 9, 14)

    if sexe != adherent['sexe']:
        return False
    if jour != adherent['jour']:
        return False
    if mois != adherent['mois']:
        return False
    if annee != adherent['annee']:
        return False
    if nom != adherent['nom'][:5]:
        return False

    return True
```

Commentaire :

- *On extrait chaque composante du numéro de licence*
- *On compare avec les informations de l'adhérent*
- *La fonction renvoie un booléen, comme demandé*