

# SUJET 0 – 2024 – sujet2 - Correction

## Exercice 3 (8 points)

---

### Partie A – Dictionnaire Python

#### 1. Valeurs de a et b

```
a = dico_livres['note']  
b = dico_livres['titre'][2]
```

- **a** vaut : [10, 8, 9, 9, 8, 10, 6, 7, 8, 8]
- **b** vaut : 'Fondation'

👉 **Commentaire :** `dico_livres['note']` renvoie toute la liste des notes. L'indice 2 correspond au troisième élément de la liste.

---

#### 2. Compléter la fonction titre\_livre

```
def titre_livre(dico, id_livre):  
    for i in range(len(dico['id'])):  
        if dico['id'][i] == id_livre:  
            return dico['titre'][i]  
    return None
```

👉 **Commentaire :** On cherche l'indice correspondant à l'identifiant, puis on renvoie le titre à ce même indice.

---

#### 3. Fonction note\_maxi

```
def note_maxi(dico):  
    return max(dico['note'])
```

👉 **Commentaire :** La fonction `max` permet d'obtenir directement la note maximale.

---

#### 4. Fonction livres\_note

```
def livres_note(dico, n):  
    liste = []  
    for i in range(len(dico['note'])):  
        if dico['note'][i] == n:  
            liste.append(dico['titre'][i])
```

👉 **Commentaire :** *On parcourt les notes et on récupère les titres correspondants.*

---

#### 5. Fonction livre\_note\_maxi

```
def livre_note_maxi(dico):  
    maxi = note_maxi(dico)  
    return livres_note(dico, maxi)
```

👉 **Commentaire :** *On réutilise les fonctions précédentes : bonne modularité du code.*

---

## Partie B – Programmation Orientée Objet

#### 6. Un attribut et une méthode de la classe Livre

- Attribut : titre
- Méthode : get\_titre()

👉 **Commentaire :** *Un attribut stocke une donnée, une méthode réalise une action.*

---

#### 7. Méthode get\_note

```
def get_note(self):  
    return self.note
```

👉 **Commentaire :** *Une méthode d'accès (getter) renvoie la valeur d'un attribut.*

---

#### 8. Ajouter Blade Runner dans la bibliothèque

```
biblio = Bibliotheque()
livre_blade = Livre(8, "Blade Runner", "K.Dick", 1968, 8)
biblio.ajout_livre(livre_blade)
```

👉 **Commentaire :** *On crée d'abord l'objet Livre, puis on l'ajoute à la liste de la bibliothèque.*

---

### 9. Compléter la méthode titre\_livre

```
def titre_livre(self, id_livre):
    for livre in self.liste_livre:
        if livre.get_id() == id_livre:
            return livre.get_titre()
    return None
```

👉 **Commentaire :** *On parcourt la liste d'objets Livre et on utilise les getters.*

---

## Partie C – Base de données relationnelle

### 10. Pourquoi auteur ne peut pas être clé primaire ?

Plusieurs livres peuvent avoir le même auteur (ex : Asimov).

👉 **Commentaire :** *Une clé primaire doit être unique.*

---

### 11. Résultat de la requête

```
SELECT titre
FROM livres
WHERE auteur = 'K.Dick';
```

Résultat :

- Ubik
- Blade Runner

👉 **Commentaire :** *Deux livres correspondent à cet auteur.*

---

## 12. Livres d'Asimov publiés après 1950

```
SELECT titre
FROM livres
WHERE auteur = 'Asimov' AND ann_pub > 1950;
```

👉 **Commentaire** : *On combine deux conditions avec AND.*

---

## 13. Modifier la note de Ubik

```
UPDATE livres
SET note = 10
WHERE titre = 'Ubik';
```

👉 **Commentaire** : *UPDATE permet de modifier une valeur existante. Ne pas oublier la clause SET.*

---

## Nouvelle structure avec deux tables

### 14. Intérêt d'utiliser deux tables

- évite la redondance des données
- meilleure organisation
- respecte la normalisation
- facilite les mises à jour

👉 **Commentaire** : *Un auteur n'est stocké qu'une seule fois.*

---

### 15. Rôle de id\_auteur

C'est une **clé étrangère** vers auteurs(id).

👉 **Commentaire** : *Elle permet de relier chaque livre à son auteur.*

---

## 16. Auteurs des livres publiés après 1960

```
SELECT nom, prenom
FROM auteurs
JOIN livres ON livres.id_auteur = auteurs.id
WHERE ann_pub > 1960;
```

👉 **Commentaire :** *Une jointure permet de relier les deux tables.*

---

## 17. Interpréter la requête

```
SELECT titre
FROM livres
JOIN auteurs ON id_auteur = auteurs.id
WHERE ann_pub - annee_naissance < 30;
```

Phrase en français : Cette requête renvoie les titres des livres écrits par des auteurs qui avaient moins de 30 ans au moment de la publication du livre.

👉 **Commentaire :** *On calcule l'âge de l'auteur à la publication.*

---

## 18. Projet d'annuaire : pourquoi problématique ?

Problème :

- données personnelles sensibles
- respect du RGPD
- besoin de consentement
- sécurisation des données obligatoire

👉 **Commentaire :** *Un tel projet nécessite une vigilance juridique importante.*

---