

# Polynésie – 2025 – sujet2 - Correction

## Exercice 2 (6 points)

---

### Partie A – Base de données champignons

#### 1. Champignon avec lamelles et couleur orange

```
SELECT nom
FROM champignon
WHERE lamelle = 'oui'
AND couleur = 'orange';
```

👉 *Commentaire :*

- *On sélectionne uniquement l'attribut `nom`.*
  - *Les deux conditions sont combinées avec `AND`.*
  - *On suppose ici que la présence de lamelles est codée par `'oui'`.*
- 

#### 2. Champignon sans pied et chapeau de 15 cm

```
SELECT nom
FROM champignon
WHERE longueur_pied = 0
AND taille_chapeau = 15;
```

👉 *Commentaire :*

- *"Sans pied" → longueur du pied égale à 0.*
  - *On combine avec la taille du chapeau.*
- 

#### 3. Clé étrangère de la table `champignon`

La clé étrangère est : `id_ordre`

👉 *Commentaire :*

- Elle référence la clé primaire `id` de la table `ordre`.
  - Elle permet de relier chaque champignon à un ordre précis.
- 

#### 4. Champignons de la classe des agaricomycètes

```
SELECT champignon.nom
FROM champignon
JOIN ordre ON champignon.id_ordre = ordre.id
WHERE ordre.classe = 'agaricomycètes';
```

👉 *Commentaire :*

- On utilise une jointure (`JOIN`).
  - On relie les tables grâce à la clé étrangère.
  - On filtre avec la condition sur la classe.
- 

#### 5. Ajouter l'amanite solitaire (id = 56)

```
INSERT INTO champignon
VALUES (56, 'amanite solitaire', 'oui', 'blanc', 6, 20, 4, 10, 4);
```

👉 *Commentaire :*

- 56 est l'identifiant.
  - L'ordre amanitales correspond à `id = 4` dans la table `ordre`.
  - On insère toutes les valeurs dans l'ordre des attributs.
- 

#### 6. Schéma relationnel

ordre(🔑 id , nom, classe )

toxicite(🔑 id\_tox , type, effets )

champignon(🔑 id , nom, lamelle, couleur, taille\_chapeau, longueur\_pied,  
#id\_ordre, #id\_toxicite )

👉 *Commentaire :*

- Clés primaires soulignées.
  - Clés étrangères précédées de #.
  - Deux relations vers *ordre* et *toxicite*.
- 

## 7. Corriger la toxicité de l'amanite citrine

```
UPDATE champignon
SET id_toxicite = 1
WHERE nom = 'amanite citrine';
```

👉 *Commentaire :*

- 1 correspond à "très toxique".
  - *UPDATE* permet de modifier une valeur existante.
- 

## 8. Amanitales très toxiques

```
SELECT champignon.nom
FROM champignon
JOIN ordre ON champignon.id_ordre = ordre.id
JOIN toxicite ON champignon.id_toxicite = toxicite.id_tox
WHERE ordre.nom = 'amanitales' AND toxicite.type = 'très toxique';
```

👉 *Commentaire :*

- Double jointure.
  - Filtre sur l'ordre et sur la toxicité.
  - Très classique au bac : requête multi-tables.
- 

## Partie B – Programmation Python

### 9. Afficher les champignons d'été

```
for e in liste_champi:
    if e.saison == 'été':
        print(e.nom)
```

👉 *Commentaire :*

- *On parcourt la liste.*
  - *On teste l'attribut `saison`.*
  - *On affiche le nom correspondant.*
- 

## 10. Pourquoi le programme renvoie False ?

Programme donné :

```
for c in liste_champi:
    if c.nom == 'Lactaire délicieux':
        return c.cuisson == 'feu moyen'
```

Explication : L'attribut `cuisson` contient : '12 minutes à feu moyen' et non : 'feu moyen' donc la comparaison stricte avec `==` échoue.

👉 *Il n'y a pas d'erreur dans la liste, mais une erreur de comparaison : on compare une phrase complète avec une chaîne partielle.*

---

## 11. Correction avec `recherche_textuelle`

```
for c in liste_champi:
    if c.nom == 'Lactaire délicieux':
        return recherche_textuelle(c.cuisson, 'feu moyen')
```

👉 *Commentaire :*

- *On cherche si "feu moyen" est contenu dans le texte complet.*
  - *On obtient alors `True`.*
  - *On évite les comparaisons strictes inadaptées.*
-