DM 3 Correction

2026 sujet 03 - Correction exercice 1

Partie A

1.

```
INSERT INTO choregraphie (nomchore, annee, choregraphe)
VALUES ('Tout autour', 2015, 9);
```

2.

```
SELECT COUNT(*)
FROM choregraphie
WHERE annee = 1983;
```

3.

La requête affiche, pour chaque chorégraphie, le nom de la chorégraphie et le nom de la personne qui l'a créée.

4.

```
SELECT DISTINCT personne.nom
FROM personne
JOIN participe
ON personne.id = participe.idpersonne
JOIN spectacle
ON spectacle.id = participe.idspectacle
WHERE spectacle.titre = 'Tout autour';
```

5.

Avant de supprimer un spectacle, il faut vérifier s'il est référencé dans la table `participe`. Il faut d'abord supprimer les lignes correspondantes dans `participe`, ou avoir prévu une suppression en cascade, afin de ne pas rompre les contraintes de clé étrangère.

Partie B

6.

```
n = 0
for clef in dmateriel:
    n += dmateriel[clef]['nb']
```

7.

```
def tab_clefs_tab_volumes(dmatos):
    tab_clefs = []
    tab_volumes = []
    for clef, dico in dmatos.items():
        tab_clefs.append(clef)
        tab_volumes.append(dico['volume'])
    return tab_clefs, tab_volumes
```

8.

Cette stratégie correspond à un algorithme glouton.

9.

On peut citer le tri par insertion. Son coût temporel dans le pire des cas est de l'ordre de n^2.

10.

```
def remplissage(dmatos, vmax):
    vol = 0
    clefs_triees = clefs_triees_selon_volume(dmatos)
    i = len(clefs_triees) - 1
    elements = []
    while i >= 0 and vol < vmax:
```

```
        clef = clefs_triees[i]
        if dmatos[clef]['nb'] == 0:
            i -= 1
        elif vol + dmatos[clef]['volume'] <= vmax:
            vol += dmatos[clef]['volume']
            elements.append(clef)
            dmatos[clef]['nb'] -= 1
        else:
            i -= 1
    return elements
```

11.

```
def nb_voitures(dmatos, vmax):
    nb = 0
    chargement = remplissage(dmatos, vmax)
    while chargement != []:
        nb += 1
        chargement = remplissage(dmatos, vmax)
    return nb
```

Programme complet pour la partie B

```
dmateriel = {
    'costume_A': {'nb': 3, 'volume': 2, 'poids': 1},
    'costume_B': {'nb': 1, 'volume': 6, 'poids': 2},
    'chaise_bois': {'nb': 40, 'volume': 50, 'poids': 4},
    'chapeau_A': {'nb': 2, 'volume': 3, 'poids': 1},
    'chapeau_B': {'nb': 4, 'volume': 7, 'poids': 1}
}

def tab_clefs_tab_volumes(dmatos):
    tab_clefs = []
    tab_volumes = []
    for clef, dico in dmatos.items():
        tab_clefs.append(clef)
        tab_volumes.append(dico['volume'])
    return tab_clefs, tab_volumes

def clefs_triees_selon_volume(dmatos):
    clefs = list(dmatos.keys())
    for i in range(1, len(clefs)):
        j = i
        while j > 0 and dmatos[clefs[j]]['volume'] < dmatos[clefs[j - 1]]['volume']:
            clefs[j], clefs[j - 1] = clefs[j - 1], clefs[j]
            j -= 1
    return clefs

def remplissage(dmatos, vmax):
    vol = 0
    clefs_triees = clefs_triees_selon_volume(dmatos)
    i = len(clefs_triees) - 1
    elements = []
    while i >= 0 and vol < vmax:
        clef = clefs_triees[i]
        if dmatos[clef]['nb'] == 0:
            i -= 1
        elif vol + dmatos[clef]['volume'] <= vmax:
            vol += dmatos[clef]['volume']
            elements.append(clef)
            dmatos[clef]['nb'] -= 1
        else:
            i -= 1
    return elements

def nb_voitures(dmatos, vmax):
    nb = 0
    chargement = remplissage(dmatos, vmax)
```

```
    while chargement != []:
        nb += 1
        chargement = remplissage(dmatos, vmax)
    return nb

print(tab_clefs_tab_volumes(dmateriel))
print(remplissage(dmateriel, 160))
print(dmateriel)
```