Antilles-Guyane – 2026 – sujet2 - Correction

Exercice 1 (6 points)

Partie A – Base de données

1. Clés étrangères de reservation

id_client est une clé étrangère qui référence la clé primaire **id_client** de la relation **client**, et **id_salle** est une clé étrangère qui référence la clé primaire **id_salle** de la relation **salle**.

Elles permettent de relier chaque réservation au client qui l'a faite et à la salle réservée : leurs valeurs doivent exister dans les tables référencées.

2. Résultat de la requête

intitule
RB
RA
Ravel

👉*Commentaire : les salles de plus de 5 places sont Ravel (2300 €), RA (350 €) et RB (300 €), triées par tarif croissant.*

3. Salles réservées depuis janvier 2024

```sql
SELECT DISTINCT id_salle
FROM reservation
WHERE date > '2023-12-31';
```

👉*Commentaire : les dates au format AAAA-MM-JJ se comparent comme des chaînes, dans l'ordre chronologique. On peut aussi écrire date >= '2024-01-01'. Sur l'extrait, on obtient 3 et 4.*

4. Ajout du client et de sa réservation

```sql
INSERT INTO client VALUES (3, 'EN', '0144624055');
INSERT INTO reservation VALUES (5, 3, 4, '2024-06-12', 1);
```

Le client doit être ajouté **en premier** : la clé étrangère id_client de la réservation doit référencer un client existant.

5. Nouvelle capacité de la salle Ravel

```sql
UPDATE salle
SET nb_places = 95
```

```sql
WHERE id_salle = 2;
```

👉 *Commentaire : on peut écrire WHERE intitule = 'Ravel', mais attention à la casse : dans la table, l'intitulé est écrit « Ravel » et non « RAVEL », et la comparaison de chaînes distingue majuscules et minuscules. Utiliser la clé primaire évite ce piège.*

6. Temps total de réservation de la salle 6

```sql
SELECT SUM(duree)
FROM reservation
WHERE id_salle = 6;
```

👉 *Commentaire : la salle 6 n'apparaît pas dans l'extrait : sur ces données, SUM renvoie NULL (aucune ligne).*

7. Sociétés ayant réservé une salle de réunion

```sql
SELECT DISTINCT client.societe
FROM client
JOIN reservation ON client.id_client = reservation.id_client
JOIN salle ON reservation.id_salle = salle.id_salle
WHERE salle.type = 'Réunion';
```

👉 *Commentaire : sur l'extrait, on obtient FleursJ (deux réservations de la salle RA, d'où l'intérêt de DISTINCT).*

Partie B – Programmation objet

8. Un attribut et une méthode

Attribut : par exemple **nb_places** (ou intitule, type_salle, tarif, dispo, occupant).

Méthode : **reserver** (ou le constructeur `__init__`).

9. Méthode liberer

```python
    def liberer(self):
        self.dispo = True
        self.occupant = ""
```

10. Fonction salles_dispos

```python
def salles_dispos(liste_salles):
    res = []
    for salle in liste_salles:          # ligne 3
        if salle.dispo:                 # ligne 4
            res.append(salle.intitule)  # ligne 5
    return res
```

👉 *Commentaire : après le code de l'énoncé, S1 et S2 sont réservées : salles_dispos(liste_salles) renvoie ['RA', 'B2', 'RB'].*

11. Méthode ajouter_reservation

```
    def ajouter_reservation(self, salle):
        if salle.dispo:
            self.liste_salles.append(salle)
            self.prix_total = self.prix_total + salle.reserver(self.client)
            return True
        return False
```

👉*Commentaire : la méthode reserver de la salle la réserve au nom du client du séminaire et renvoie son tarif, qu'on ajoute au prix total. Comme on a vérifié que la salle est disponible, reserver ne renvoie pas False.*