

Centres étrangers – 2025 – sujet1 - Correction

Exercice 3 (8 points)

Partie A

1. Deux avantages d'une base de données relationnelle

- Réduction des redondances grâce à la normalisation.
- Garantie de cohérence grâce aux clés primaires et étrangères.

👉 *Commentaire : une base relationnelle assure l'intégrité des données et facilite les requêtes complexes.*

2. Caractéristique d'une clé primaire

Un attribut utilisé comme clé primaire doit être unique et non nul pour chaque enregistrement.

👉 *Commentaire : chaque ligne doit être identifiable sans ambiguïté.*

3. Rôle des clés étrangères dans Reservation

- `id_client` relie une réservation à un client.
- `id_emplacement` relie une réservation à un emplacement.

👉 *Commentaire : elles assurent le lien entre les tables et l'intégrité référentielle.*

4. Schéma relationnel de Emplacement

Emplacement (`id_emplacement` : INT,
nom : VARCHAR,
localisation : VARCHAR,
tarif_journalier : FLOAT)

👉 *Commentaire : l'identifiant est un entier, le tarif peut être décimal.*

5. Résultat de la requête

```
SELECT id_emplacement, nom, localisation
FROM Emplacement
WHERE tarif_journalier = 25;
```

Résultat :

id_emplacement	nom	localisation
1	myrtille	A4
4	mandarine	B1
6	melon	A2

👉 *Commentaire : on sélectionne uniquement les emplacements à 25 €.*

6. Requête pour les clients habitant Strasbourg

```
SELECT nom, prenom
FROM Client
WHERE ville = 'Strasbourg';
```

👉 *Commentaire : la condition porte sur l'attribut ville.*

7. Requête pour ajouter le client

```
INSERT INTO Client
VALUES (42, 'CODD', 'Edgar', '28 rue des Capucines',
       'Lyon', 'France', '0555555555');
```

👉 *Commentaire : on respecte l'ordre des attributs du schéma.*

8. Requête pour obtenir les informations de la réservation 18

```
SELECT Client.nom, Client.prenom,
       Reservation.nombre_personne,
       Reservation.date_arrivee,
       Reservation.date_depart,
       Emplacement.tarif_journalier
FROM Reservation
JOIN Client ON Reservation.id_client = Client.id_client
```

```
JOIN Emplacement ON Reservation.id_emplacement =  
Emplacement.id_emplacement  
WHERE Reservation.id_reservation = 18;
```

👉 *Commentaire : deux jointures sont nécessaires pour relier les trois tables.*

Partie B

9. Rôle du paramètre self

Le paramètre `self` représente l'instance de la classe sur laquelle la méthode est appelée.

👉 *Commentaire : il permet d'accéder aux attributs de l'objet.*

10. Instanciation de client01

Exemple possible :

```
client01 = Client(42, "CODD", "Edgar",  
                 "28 rue des Capucines",  
                 "Lyon", "France",  
                 "0555555555")
```

👉 *Commentaire : on passe les attributs dans l'ordre défini par le constructeur.*

11. Compléter montant_a_regler

La formule est :

$\text{tarif} * \text{durée} + \text{nombre_personne} * 2.20 * \text{durée}$

```
def montant_a_regler(triplet):  
    """ renvoie le montant en euros  
    à régler pour cette réservation """  
    client, reservation, emplacement = triplet  
    return (emplacement.tarif_journalier *  
            (reservation.date_depart - reservation.date_arrivee).days  
            + reservation.nombre_personne * 2.20 *  
            (reservation.date_depart - reservation.date_arrivee).days)
```

👉 *Commentaire : on multiplie par le nombre de jours du séjour.*

(Si les dates sont déjà un nombre de jours, on utilise simplement cette valeur.)

12. Pourquoi une erreur se produit dans `facture_est_valide`

L'erreur se produit si la chaîne ne contient pas exactement deux tirets.

Dans ce cas, `split('-')` ne renvoie pas trois éléments et l'accès aux indices provoque une erreur.

👉 *Commentaire : il faut vérifier la longueur de la liste obtenue après `split`.*

13. Correction pour éviter l'erreur

Ajouter une vérification :

```
parties = numero.split('-')
if len(parties) != 3:
    return False
```

👉 *Commentaire : on sécurise l'accès aux éléments.*

14. Vérification du mois et du numéro

Exemple de complétion :

```
calendrier = ["JAN", "FEB", "MAR", "APR", "MAY", "JUN",
              "JUL", "AUG", "SEP", "OCT", "NOV", "DEC"]

if mois not in calendrier:
    return False

if len(numero_part) != 3 or not que_des_chiffres(numero):
    return False
```

👉 *Commentaire : le mois doit appartenir à une liste prédéfinie et le numéro doit contenir exactement trois chiffres.*
