

Éléments de correction sujet 07 (2023)

Exercice 1

1.
 - a. L'attribut doit être unique afin de pouvoir distinguer 2 entrées de la table.
 - b. L'attribut idClient permet de lier la table Commandes et la table Clients
L'attribut idMeuble permet de lier la table Commandes et la table Meubles
 - c. Meubles (id : INT, intitule : TEXT, prix : FLOAT, stock : INT, description : TEXT)

2. On obtient :

62	2	'Armoire blanche 3 portes'
63	3	'Armoire noire 3 portes'

3.

```
SELECT nom, prenom
FROM Clients
WHERE ville = 'Paris'
```
4.

```
UPDATE Meubles
SET stock = 50
WHERE id = 98
```
5.

```
INSERT INTO Meubles
VALUES
(65, 'matta', 95.99, 25, 'Tapis vert à pois rouges')
```
6.

```
SELECT nom, prenom
FROM Clients
JOIN Commandes ON Clients.id = Commandes.idClient
WHERE date = '30/04/2002'
```

Exercice 2

1. Il y a plusieurs chemins possibles pour aller d'un site vers un autre. Par exemple, pour aller de A vers C, il est possible de passer par R1 - R3 - R2 - R5 ou R1 - R3 - R4 - R5. si la liaison 5 tombe en panne, il sera toujours possible de passer par R1 - R3 - R2 - R5.
2. Site B - R2 - R3 - R4 - R5 - Site C

3.

Routeur R1 (RIP)		
Destination	Suivant	Nombre de sauts
SiteA	Local	0
SiteB	R2	1
SiteC	R2	2
SiteD	R3	2

4.

D'après le protocole RIP, pour aller du site B au site C, on devrait parcourir R2 - R5 (et donc la liaison 2). La liaison 2 étant très lente, il serait donc plus judicieux de parcourir R2 - R3 - R4 - R5 qui évite la liaison 2.

5.

a. le débit le plus faible correspond au coût le plus important. C'est donc la liaison 2 (coût = 1000000) qui a le débit le plus faible.

b.

SiteA - R1 - R2 - R5 - SiteC => 1 100 000

SiteA - R1 - R3 - R2 - R5 - SiteC => 1 050 005

SiteA - R1 - R3 - R4 - R5 - SiteC => 50 015

SiteA - R1 - R2 - R3 - R4 - R5 - SiteC => 100 020

c.

Routeur R1 (OSPF)		
Destination	Suivant	Coût total du chemin
SiteA	Local	0
SiteB	R3	50 005
SiteC	R3	50 015
SiteD	R3	50 005

Exercice 3

Partie 1

1.

nom, tab_voisines, tab_couleurs_disponibles et couleur_attribuee sont des attributs de la classe Region.

2.

nom_region est de type string

3.

ge = Region("Grand Est")

4.

```
def renvoie_premiere_couleur_disponible(self):  
    return self.tab_couleurs_disponibles[0]
```

5.


```
def renvoie_nb_voisines(self) :
    return len(self.tab_voisines)
```
6.


```
def est_coloriee(self):
    return self.couleur_attribuee != None
```
7.


```
def retire_couleur(self, couleur):
    if couleur in self.tab_couleurs_disponibles:
        self.tab_couleurs_disponibles.remove(couleur)
```
8.


```
def est_voisine(self, region):
    for r in self.tab_voisines:
        if r == region:
            return True
    return False
```

Partie 2

9.


```
def renvoie_tab_regions_non_coloriees(self):
    tab_r = []
    for r in self.tab_regions:
        if not r.est_coloriee():
            tab_r.append(r)
    return tab_r
```
10.
 - a.

La méthode renvoie None dans le cas où toutes les régions sont coloriées
 - b.

La région renvoyée est non coloriée et possède le plus grand nombre de voisins
11.


```
def colorie(self):
    r_max = self.renvie_max()
    while r_max != None:
        couleur = r_max.renvie_premiere_couleur_disponible()
        r_max.couleur_attribuee = couleur
        for r in r_max.tab_voisines :
            r.retire_couleur(couleur)
        r_max = self.renvie_max()
```