

Exercice 3 : Partie A :

1. `personneA = Personne(112, "LESIEUR", "Isabelle", 1982, 2005)`

2. `personneA.num_badge`

3. `# Méthode de la classe Personne`
`def annee_anciennete(self):`
 `return 2024 - self.annee_entree`

4. `# Méthode de la classe Personnel`
`def ajouter(self, personne):`
 `self.liste.append(personne)`

5. `# Méthode de la classe Personnel`
`def effectif(self):`
 `return len(self.liste)`

6. `# Méthode de la classe Personnel`
`def donne_nom(self, num):`
 `for elt in self.liste:`
 `if elt.num_badge == num:`
 `return elt.nom`
 `return None`

7.
`def nb_personne_honneur(self, annee):`
 `total = 0`
 `for elt in self.liste:`
 `if elt.annee_anciennete() == 10:`
 `total += 1`
 `return total`

ou

`def nb_personne_honneur(self, annee):`
 `return sum([1 for elt in self.liste if elt.annee_anciennete() == 10])`

8. En considérant que l'on ne retient que ceux ayant la même ancienneté maximale :

Solution 1 :

`def plus_anciens(self):`
 `maxi = 0`
 `for p in self.liste:`
 `if p.annee_anciennete() >= maxi:`
 `maxi = p.annee_anciennete()`
 `liste = []`
 `for p in self.liste:`
 `if p.annee_anciennete() == maxi:`
 `liste.append(p.num_badge)`
 `return liste`

Solution 2 :

`def plus_anciens(self):`
 `liste = []`
 `maxi = 0`
 `for p in self.liste:`
 `if p.annee_anciennete() >= maxi:`
 `if p.annee_anciennete() > maxi:`
 `maxi = p.annee_anciennete()`
 `liste = []`
 `liste.append(p.num_badge)`
 `return liste`

Exercice 3 : Partie B :

9. La requête donner le nom et le prénom de tout le personnel travaillant dans le centre n° 2.

10.

```
UPDATE Personnel SET num_centre = 3 WHERE num_badge = 135
```

11. Utiliser deux tables permet d'éviter la redondance d'information et facilite les mises à jour.

12. Ces deux tables sont mises en relation par la clé étrangère num_centre de la table Personnel qui fait référence à la clé primaire num de la table Centre.

13.

```
SELECT Personnel.nom FROM Personnel  
JOIN Centre ON Personnel.num_centre = Centre.num  
WHERE Centre.nom = 'Lille' AND Personnel.annee_debut BETWEEN 2015 AND 2020 ;
```

14. Cet requête renvoie un erreur car elle viole la contrainte d'intégrité de référence de la base de donnée. En effet, il est impossible de supprimer un enregistrement auquel d'autres font référence.

(Erreur de l'énoncé : La requête devrait être :

```
DELETE FROM Centre WHERE nom = 'Normandie';)
```