

EXERCICE 2 (Bases de données relationnelles) :

1.a)

Les prénoms et noms des membres du site qui proposent la location d'un appareil à raclette sont :

Mohamed Ali  
Fernando Alonso

1.b)

Harry Kane ne propose pas d'objet à la location.

2.a)

Cette requête donne comme résultat :

Antoine Dupont  
Harry Kane

2.b)

```
SELECT tarif FROM Objet WHERE id_objet=5
```

2.c)

```
UPDATE Objet SET tarif=20 WHERE id_objet=1
```

2.d)

```
INSERT INTO Membre VALUES (6, "Renard", "Wendy", "69100")
```

3.a)

Si l'on utilisait le couple de clés étrangères (`id_objet`, `id_membre`) en tant que clé primaire de la relation `Reservation` un membre ne pourrait pas réserver un objet puis plus tard ce même objet.

3.b)

Cette requête produit une erreur car ce membre (`id_membre=1`) est référencé dans la table `Possede` (il possède un appareil à raclette et un appareil à gauffre) et dans la table `Reservation`. Une clé étrangère ne peut pas pointer sur une référence inexistante, donc ce membre ne peut pas être supprimé

3.c)

```
DELETE FROM Membre WHERE id_membre=1
```

```
DELETE FROM Reservation WHERE id_membre=1
```

3.b)

```
INSERT INTO Match VALUES(10,3,73,6,78,23/10/2021)
```

4.a)

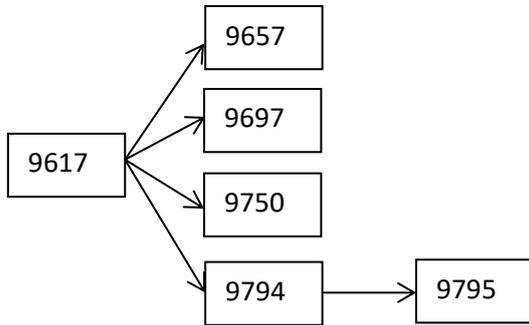
```
SELECT COUNT(*) FROM Reservation  
JOIN Membre ON Reservation.id_membre=Membre.id_membre  
WHERE Membre.prenom="Fernando" and Membre.nom="Alonso"
```

4.b)

```
SELECT nom,prenom FROM Membre  
JOIN Possede ON Possede.id_membre=Membre.id_membre  
JOIN Objet ON Possede.id_objet=Objet.id_objet  
WHERE Objet.description="Appareil à raclette"
```

EXERCICE 3 (Processus et ordonnancement) :

1.a)



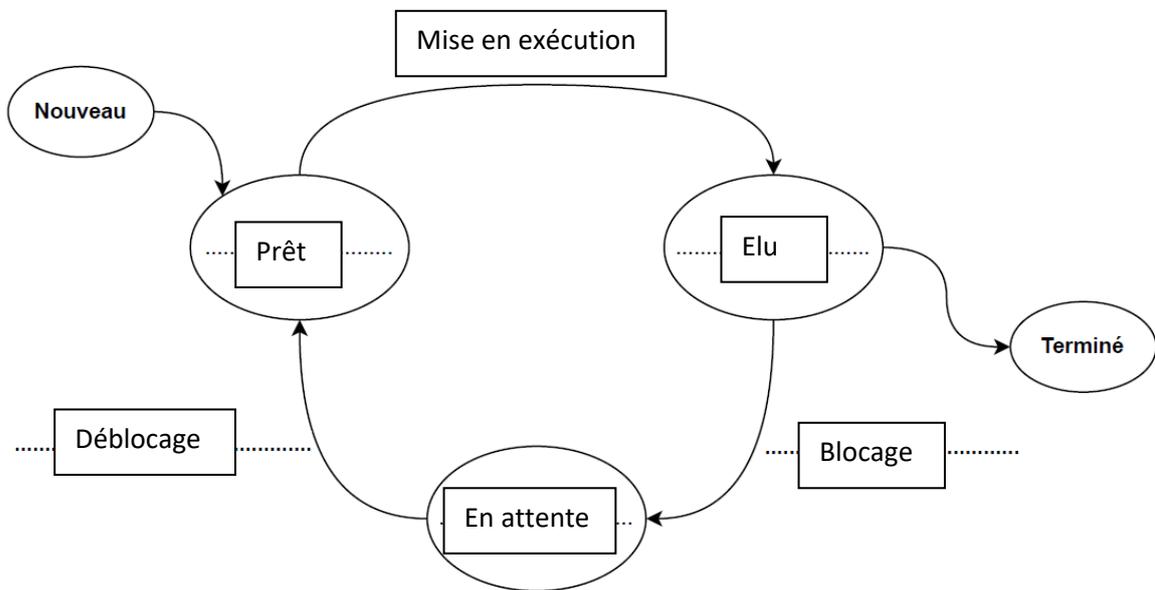
1.b)

Le premier processus de Firefox a été lancé par la commande `bash` (processus 8887).

1.c)

La commande `kill 9617` permet de supprimer tous les processus liés à Firefox

2.a)



2.b)

Temps d'exécution de P1 =  $12 - 0 = 12$

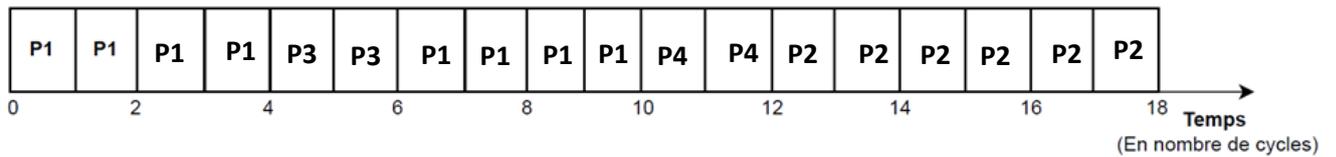
Temps d'exécution de P2 =  $18 - 2 = 16$

Temps d'exécution de P3 =  $5 - 3 = 2$

Temps d'exécution de P4 =  $9 - 7 = 2$

Moyenne des temps d'exécution =  $(12 + 16 + 2 + 2) / 4 = 32 / 4 = 8$ .

2.c)



2.d)

Temps d'exécution de P1 = 10-0 = 10

Temps d'exécution de P2 = 18-2=16

Temps d'exécution de P3 = 6-3=3

Temps d'exécution de P4 = 12-7=5

Moyenne des temps d'exécution =  $(10+16+3+5)/4 = 34/4 = 8,5$ .

Pour traiter ces 4 processus le nouvel ordonnancement est moins performant que le premier.

3.a)

```
for i in range(1, len(liste_attente)) :
    d = 0
    for j in liste_attente[i] :
        if j != "" :
            d = d+1 # compte la durée du process d'indice i
    if d < mini :
        mini = d
        indice = i
```

3.b)

```
# Ajoute indice dans la liste d'exécution des processus
execution.append(indice)
# Retire le premier élément du processus,
# pour diminuer de 1 sa durée.
attente[indice].pop(0)
```