

1. attributs de la table groupes : idgrp, nom, style, nb\_pers
2. Le même nom peut apparaître plusieurs fois (par exemple "Parker"), ce qui n'est pas possible avec une clé primaire
3. Cette requête renvoie : 'Weather Report' et 'Return to Forever'
4. UPDATE concerts  
SET heure\_fin = '22h30'  
WHERE idconc = 36
5.  
SELECT nom  
FROM groupes  
JOIN concerts ON concerts.idgrp = groupes.idgrp  
WHERE scene = 1
6.  
INSERT INTO groupes  
VALUES  
(15, 'Smooth Jazz Fourplay', 'Free Jazz', 4)
7.  
def recherche\_nom(tab):  
 t = []  
 for d in tab:  
 if d['nb\_concerts'] >= 4 and d['nom'] not in t :  
 t.append(d['nom'])  
 return t

1. panier1.enfiler((31002, "café noir", 1.50, 50525))

2.

```
def remplir(self, panier_temp):  
    while not panier_temp.est_vide() :  
        article = panier_temp.defiler()  
        self.enfiler(article)
```

3.

```
def prix_total(self):  
    p_temp = Panier()  
    montant = 0  
    while not self.est_vide() :  
        article = self.defiler()  
        montant = montant + article[2]  
        p_temp.enfiler(article)  
    while not p_temp.est_vide() :  
        article = p_temp.defiler()  
        self.enfiler(article)  
    return montant
```

4.

```
def horaire_scan(self):  
    if self.est_vide():  
        return 0  
    premier_article = self.defiler()[3]  
    dernier_article = premier_article  
    while not self.est_vide() :  
        dernier_article = self.defiler()[3]  
    return dernier_article - premier_article
```

1.
  - a. Une fonction est dite récursive si cette fonction s'appelle elle-même
  - b. Cette fonction s'arrête quand n est égal à -1. En effet, quand  $n = -1$ ,  $n \geq 0$  devient False, on ne "rentre plus dans le if", les appels récursifs cessent.

2.

```
def fact(n) :  
    if n == 0:  
        return 1  
    else :  
        return n*fact(n-1)
```

3.

- a. Après l'exécution de `res = somme_entiers_rec(3)` dans la console, on obtient l'affichage suivant :

```
3  
2  
1
```

- b.

La valeur affectée à la variable res est 6 ( $3+2+1 = 6$ )

4.

```
def somme_entiers(n) :  
    somme = 0  
    while n > 0:  
        somme = somme + n  
        n = n - 1  
    return somme
```