EXERCICE 1 (Bases de données relationnelles) :

1.a)

Cette requête `INSERT INTO` provoque une erreur car la relation `Equipe` contient déjà un enregistrement ayant comme attribut `id_equipe` la valeur 11. Comme `id_equipe` est une clef primaire, il ne peut pas y avoir plusieurs équipes avec ce même attribut.

1.b)

L'attribut `telephone` de la relation `Equipe` est de type `VARCHAR(20)` c'est-à-dire une chaîne de caractères de longueur maximum 20. C'est un bon choix car un numéro de téléphone peut commencer par le caractère 0 ou le caractère +, ce qui empêche d'utiliser un type `INT`.

1.c)

Cette requête donne comme résultat :

Lyon 451 cours Emile Zola, 69100 Villeurbanne 04 05 06 07 08

1.d)

Cette requête compte le nombre d'enregistrements dans la relation Equipe, soit 12.

1.e)

La requête qui affiche les noms des équipes triés par ordre alphabétique est :

```
SELECT nom FROM Equipe ORDER BY nom
```

1.f)

Pour corriger le nom de l'équipe dont l'`id_equipe` est égal à 4, on écrit la requête SQL :

```
UPDATE Equipe SET nom="Tarbes" WHERE id_equipe=4
```

2.a)

L'attribut `id_equipe` a été déclaré clé étrangère de la relation `Joueuse`, car toute joueuse doit appartenir à une équipe. Ainsi on ne pourra pas créer un enregistrement dans la relation `Joueuse` qui ne référence pas une équipe existante dans la relation `Equipe`.

2.b)

On ne peut pas supprimer une équipe dans la relation `Equipe` qui aurait comme `id_equipe` une valeur référencée dans la relation `Joueuse`. Cela reviendrait à avoir une joueuse qui n'aurait pas d'équipe : c'est impossible, cela viole les contraintes de référence d'une base de données relationnelle.

2.c)

On peut écrire la requête SQL suivante :

```
SELECT Joueuse.nom, Joueuse.prenom FROM Joueuse JOIN Equipe ON
Joueuse.id_equipe=Equipe.id_equipe WHERE Equipe.nom="Angers"
ORDER BY Joueuse.nom
```

3.a)

Pour la relation `Match` on peut avoir le schéma relation suivant :

attribut	domaine	commentaire
id_match	INT	Identifiant du match. Clef primaire.
#id_equipe_dom	INT	Identifiant de l'équipe jouant à domicile. Référence id_equipe de la relation Equipe.
score_dom	INT	Score de l'équipe jouant à domicile.
#id_equipe_dep	INT	Identifiant de l'équipe jouant en déplacement. Référence id_equipe de la relation Equipe.
score_dep	INT	Score de l'équipe jouant à domicile.
Date	DATE	Date du match

3.b)

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

4.a)

Pour stocker les statistiques des joueuses, on peut créer une nouvelle relation `Stats` ayant le schéma relation suivant :

attribut	domaine	commentaire
#id_match	INT	Identifiant d'un match. Référence id_match de la relation Match.
#id_joueuse	INT	Identifiant d'une joueuse. Référence id_joueuse de la relation Match.
points	INT	Nombre de points marqués par la joueuse.
rebonds	INT	Nombre de rebonds récupérés par la joueuse.

passes	INT	Nombre de passes décisives réalisées par la joueuse.

La clé primaire de cette relation est le couple `(id_match , id_joueuse)`.

4.b)

Pour afficher l'extrait des statistiques on peut écrire la requête SQL :

```
SELECT Equipe.nom,Joueuse.nom,Joueuse.prenom,Stats.points,
Stats.rebonds,Stats.passes FROM Stats
JOIN Match ON Stats.id_match=Match.id_match
JOIN Joueuse ON Stats.id_joueuse=Joueuse.id_joueuse
JOIN Equipe ON Match.id_equipe_dom=Joueuse.id_equipe
JOIN Equipe ON Match.id_equipe_rep=Joueuse.id_equipe
WHERE Match.id_match=53
```

EXERCICE 2 (Processus et POO) :

1.a)

La suite des PID des processus dans l'ordre de leur exécution est :

11 , 20 , 32 , 11 , 20 , 32 , 11 , 32 , 11

1.b)

La suite des PID des processus dans l'ordre de leur exécution est :

11 , 11 , 20 , 20 , 32 , 32 , 11 , 11 , 32

2.a)

```
liste_attente = [Processus(11,4),Processus(20,2),Processus(32,3)]
```

2.b)

```
def execute_un_cycle(self):
     self.reste_a_faire = self.reste_a_faire - 1

def change_etat(self, nouvel_etat):
     self.etat = nouvel_etat

def est_termine(self):
     if self.reste_a_faire==0 :
          return True
     else :
          return False
```

2.c)

```
while compteur_tournique<quantum and
      not processus.est_termine() :

if not processus.est_termine() :

processus.change_etat("Terminé")
```