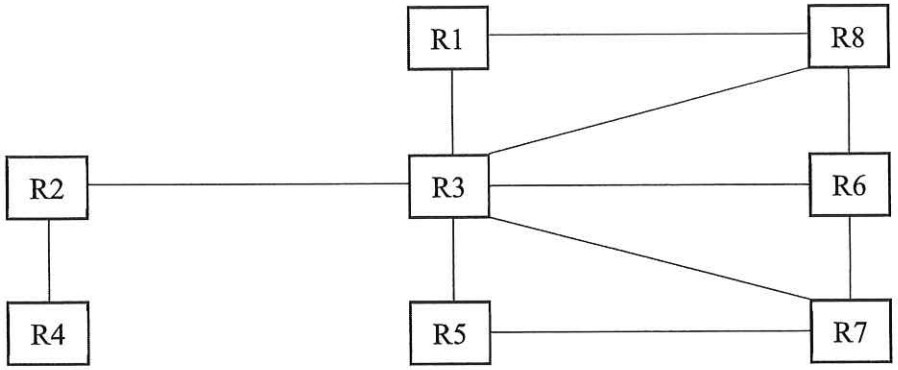
EXERCICE 1 (5 points)

Les deux parties sont indépendantes.

Partie A

La responsable informatique doit gérer le réseau informatique de son entreprise représenté ci-dessous dans lequel R1, R2, R3, R4, R5, R6, R7 et R8 sont des routeurs. Elle décide d'utiliser le protocole RIP pour configurer les tables de routages.

Ce protocole est un protocole à vecteurs de distance. La métrique permettant de décider du meilleur chemin vers un routeur distant est le nombre de sauts.



1. La table de routage de R1 débute ainsi :

Destination	Passerelle	Métrique
R1	R1	0
R2	R3	2
R3	R3	1

Recopier puis compléter la table de routage de R1 selon le protocole RIP.

On donne le constructeur de la classe `Routeur`

```
class Routeur :
    def __init__(self, name) :
        self.nom = name
        self.table_routage = { self : (self, 0) }
        # le routeur est lié à lui-même (self)
```

L'attribut `table_routage` est un dictionnaire dont les clés sont des objets de type `Routeur` et la valeur est le couple `(pasr, m)` où `pasr` est un objet de type `Routeur` et `m` est la métrique selon le protocole RIP entre le routeur de la clé et le routeur `pasr`.

2. La méthode `ajout_destination(self, dest, pasr, m)` de la classe `Routeur`, prend en paramètres un routeur de destination `dest`, un routeur passerelle `pasr` et un entier `m`. Elle ajoute à la table de routage de `self` le routeur `dest` associé à la passerelle `pasr` et à la métrique `m`.

   Par exemple, on crée les routeurs R1, R2 et R3 puis on ajoute les destinations R2 et R3 à la table de routage de R1 à l'aide des commandes suivantes :

```
r1 = Routeur("R1")
r2 = Routeur("R2")
r3 = Routeur("R3")
r1.ajout_destination(r2, r3, 2)
r1.ajout_destination(r3, r3, 1)
```

   On dispose d'une méthode `afficher(self)` qui permet d'obtenir la table de routage du routeur `self`.

   Par exemple, la commande `r1.afficher()` affiche :

```
      ROUTEUR R1 : table de routage
dest.     | pass.    | metrique
R1        | R1       | 0
R2        | R3       | 2
R3        | R3       | 1
```

   a. Proposer les commandes nécessaires permettant d'ajouter les routeurs R4, R5 à la table de routage de R1.

   b. Écrire la méthode `ajout_destination(self, dest, pasr, m)`.

3. Écrire la méthode `voisins(self)` qui renvoie la liste des routeurs directement connectés au routeur `self`.

   Par exemple, `r1.voisins()` renverra la liste `[r3, r8]`, où `r3` et `r8` sont des objets de types `Routeur`.

4. La méthode `calcul_route(self, dest)` de la classe `Routeur` prend en paramètre un routeur de destination `dest` et renvoie la liste des routeurs parcourus lors d'une communication entre les routeurs `self` et `dest`.

Par exemple, `r1.calcul_route(r7)` renverra la liste `[r1, r3, r7]` où `r1`, `r3` et `r7` sont des objets de types `Routeur`.

Recopier et compléter le code de la méthode `calcul_route(self, dest)` ci-dessous

```
def calcul_route(self, dest) :
    route = [self]
    routeur_courant = route[-1]
    while ... :

        ...          # plusieurs lignes

    return route
```

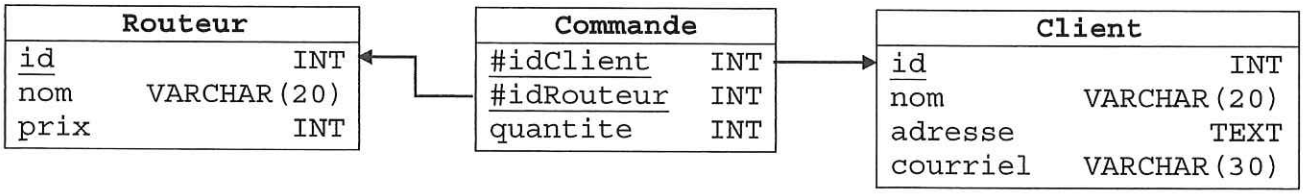
Partie B

Dans cette partie, on pourra utiliser les mots clés suivants du langage SQL.

`SELECT, INSERT INTO, WHERE, UPDATE, JOIN, ORDER BY`

La commande `ORDER BY propriété` permet de trier dans l'ordre croissant les résultats d'une requête selon l'attribut `propriété`.

Pour gérer son réseau informatique, la responsable achète son matériel à la société BeauReseau qui dispose d'une base de données dont le schéma relationnel est ci-dessous.



Les clés primaires sont soulignées et les clés étrangères sont précédées du caractère #.
Ainsi l'attribut `idRouteur` de la relation `Commande` est une clé étrangère qui fait référence à l'attribut `id` de la relation `Routeur` et l'attribut `idClient` de la relation `Commande` est une clé étrangère qui fait référence à l'attribut `id` de la relation `Client`.

Pour les questions où on demande des résultats de requêtes, on considèrera les extraits des tables remplies ainsi :

Table Routeur

id	nom	prix
1	C6Po-1000	1200
2	Coq6-300	6000
3	Al-200	6000
4	C6Po-9000	9000

Table Client

id	nom	adresse	courriel
1	Knuth	rue Donald, Tampa	dknuth@usa.org
2	Hooper	rue Grace, Boston	ghooper@usa.org
3	Torvalds	rue Linus, Helsinki	ltorvalds@finland.org
4	Pouzin	rue Louis, Paris	lpouzin@france.fr

Table Commande

idClient	idRouteur	quantite
1	1	4
1	3	1
2	1	1
2	1	5
3	2	3
4	4	1
4	1	5

5. On considère la requête d'insertion suivante

```
INSERT INTO Routeur(id, nom, prix) VALUES(3, 'Dali-32', 4000)
```

Expliquer pourquoi cette requête renvoie une erreur.

6. Proposer une requête qui renvoie le nom de tous les routeurs dont le prix est compris entre 2500 (inclus) et 7000 euro (inclus).

7. On considère la requête suivante :

```
SELECT nom FROM Client
JOIN Commande ON Commande.idClient = Client.id
WHERE Commande.quantite = 1
```

En considérant les extraits des tables fournies, préciser ce que renvoie cette requête.

8. Proposer une requête qui renvoie les noms triés dans l'ordre croissant des routeurs achetés par le client n°2.