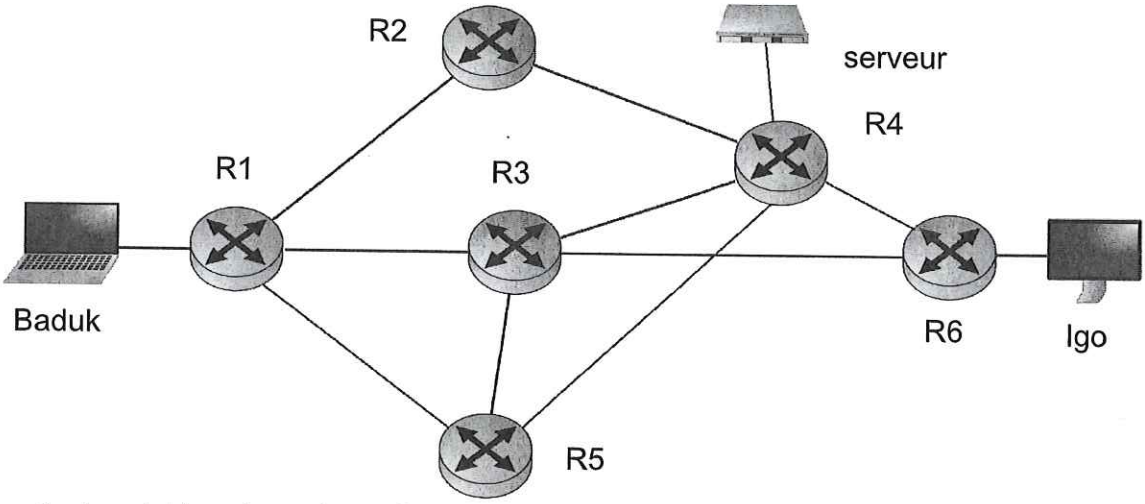
Exercice 2 (4 points)

Les deux parties de cet exercice sont indépendantes.

Le jeu de go est un jeu de société originaire de Chine. Il oppose deux adversaires qui placent à tour de rôle des pierres, respectivement noires et blanches, sur un plateau.

Partie A : Réseau

Baduk et Igo jouent une partie de go grâce à une application en ligne, hébergée sur un serveur. La portion de réseau au voisinage des ordinateurs de Baduk, d'Igo et du serveur de jeu est représentée ci-dessous.



1. Les tables de routage des routeurs sont les suivantes. La métrique permettant ici de décider du meilleur chemin vers un routeur distant est le nombre de sauts.

R1		
Destination	**Passerelle**	**Métrique**
R2, R3, R5	-	1
R4	R2	2
R6	R3	2

R2		
Destination	**Passerelle**	**Métrique**
R1, R4	-	1
R3, R5, R6	R4	2

R3		
Destination	**Passerelle**	**Métrique**
R1, R4, R5, R6	-	1
R2	R4	2

R4		
Destination	**Passerelle**	**Métrique**
R2, R3, R5, R6	-	1
R1	R5	2

R5		
Destination	**Passerelle**	**Métrique**
R1, R3, R4	-	1
R2	R1	2
R6	R4	2

R6		
Destination	**Passerelle**	**Métrique**
R3, R4	-	1
R1, R5	R3	2
R2	R4	2

Igo effectue un coup grâce à son application puis Baduk joue à son tour. Les données partent de la machine d'Igo vers le serveur puis sont transmises à la machine de Baduk. Les données du coup de Baduk sont envoyées de sa machine vers le serveur puis transmises à la machine d'Igo.
En utilisant les tables de routage fournies, indiquer un chemin emprunté par les données lors de ces échanges.

2. Recopier la table de routage du routeur R3 avec une mise à jour possible suite à la rupture de la liaison entre les routeurs R3 et R4.

Partie B : Base de données

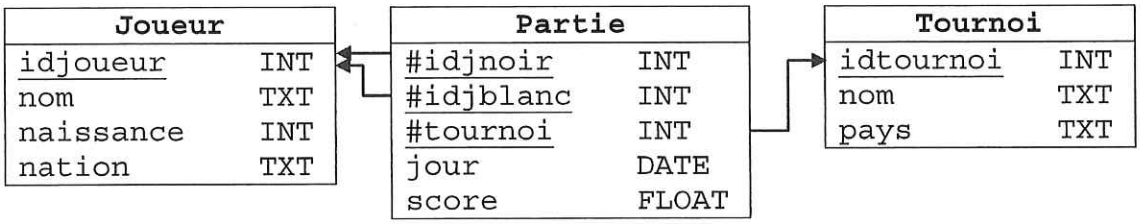
Dans cette partie, on pourra utiliser les mots clés suivants du langage SQL.

`SELECT, INSERT INTO, DISTINCT, WHERE, UPDATE, JOIN, COUNT, MIN, MAX, ORDER BY`

La fonction d'agrégation `COUNT(*)` renvoie le nombre d'enregistrements de la requête. Les fonctions d'agrégation `MIN(propriete)` et `MAX(propriete)` renvoient respectivement la plus petite et la plus grande valeur de l'attribut `propriete` pour les enregistrements de la requête. La commande `ORDER BY propriété` permet de trier les résultats d'une requête selon l'attribut `propriété`.

Le responsable de la fédération internationale de jeu de go enregistre dans une base de données les résultats de parties historiques.

Il définit pour cela des relations `Joueur`, `Partie` et `Tournoi` qui suivent le schéma relationnel suivant (les clés primaires sont soulignées et les clés étrangères sont précédées du caractère #).



Les clés primaires sont soulignées et les clés étrangères sont précédées du caractère #.
Ainsi l'attribut `idjnoir` de la relation `Partie` est une clé étrangère qui fait référence à l'attribut `idjoueur` de la relation `Joueur`.
`idjnoir` et `idjbanc` identifient les joueurs ayant respectivement les pierres noires et les pierres blanches.

3. On suppose que ce schéma relationnel a été implémenté dans un système de gestion de base de données. La base de données est vide et on souhaite enregistrer les résultats d'un premier tournoi à l'aide des commandes SQL suivantes.

```
1  INSERT INTO Joueur (idjoueur, nom, naissance, nation)
2  VALUES  (1 ,'Dosaku', 1645, 'Japon'),
3          (2, 'Genan Inseki', 1798, 'Japon'),
4          (3, 'Shusaku', 1829, 'Japon');
5
6  INSERT INTO Partie (idjnoir, idjblanc, tournoi, jour, score)
7  VALUES (2, 3, 1, '1846-09-12', -2);
8
9  INSERT INTO Tournoi (idtournoi, nom, pays)
10 VALUES (1, 'Osaka', 'Japon');
```

a. Quelle est la nature de l'erreur produite par l'exécution de cette succession de commandes ? Justifier.

b. Comment corriger la succession des commandes SQL données pour que le traitement s'effectue sans erreur ?

On donne ci-dessous un extrait des enregistrements contenus dans la base de données.

Joueur			
idjoueur	nom	naissance	nation
1	Dosaku	1645	Japon
2	Genan Inseki	1798	Japon
3	Shusaku	1829	Japon
4	Kitani Minuro	1909	Japon
5	Go Seigen	1914	Chine
6	Sakata Eio	1920	Japon
7	Rin Kaiho	1942	Taiwan
8	Cho Chikun	1953	Corée
9	Rui Naiwei	1963	Chine
10	Lee Changho	1975	Corée

Partie				
idjnoir	idjblanc	tournoi	jour	score
2	3	1	1846-09-12	-2
4	5	0	1933-11-12	0
5	4	2	1939-09-28	2
5	6	0	1953-11-19	999
5	4	3	1961-06-28	999
6	5	3	1962-08-05	0
7	6	3	1967-08-09	2
7	8	4	1983-05-16	-999
10	8	6	1993-04-24	0.5
9	10	5	2000-01-04	999

Tournoi		
idtournoi	nom	pays
0	Inconnu	Autre
1	Osaka	Japon
2	Kamakura games	Japon
3	Meijin	Japon
4	Honinbo	Japon
5	Guksu	Corée
6	Ton Yang Cup	Corée

4. On considère la requête SQL ci-dessous.

```
SELECT COUNT(*) FROM Partie
WHERE idjnoir = 4 OR idjblanc = 4;
```

   a. Quel serait l'affichage produit par cette requête, appliquée aux seuls extraits des enregistrements donnés dans les tableaux précédents ?
   b. Expliquer avec une phrase ce que renvoie cette requête.

5. Proposer une requête qui renvoie, dans l'ordre alphabétique, les noms des tournois ayant eu lieu au Japon.

6. Proposer une requête qui renvoie le nom des joueurs qui ont joué avec les pierres noires le 15 mars 2016.

7. Expliquer avec une phrase ce que renvoie la requête ci-dessous.

```
SELECT DISTINCT nom
FROM Joueur
JOIN Partie
     ON (Joueur.idjoueur = Partie.idjnoir
         OR Joueur.idjoueur = Partie.idjblanc)
WHERE Partie.tournoi = 3;
```