Exercice 4 : Bases de données relationnelles et le langage SQL

L'énoncé de cet exercice utilise les mots du langage SQL suivant : SELECT, FROM, WHERE, JOIN, INSERT INTO, VALUES, COUNT, ORDER BY.
Pour les besoins de l'organisation du lycée, le chef d'établissement exploite la base de données par des requêtes en langage SQL. Il a pour cela créé une table (ou relation) SQL dénommée seconde dans son système de gestion de bases de données dont la structure est la suivante :

seconde	Type
num_eleve (clef primaire)	entier (??)
langue1	CHAR
langue2	CHAR
option	CHAR
classe	CHAR

Question 1

1. **Dans le modèle relationnel, quel est l'intérêt de l'attribut num_eleve.**
   La clé primaire d'une relation est un attribut qui permet de désigner d'une façon unique un uplet.
   Par exemple l'attribut *num_eleve* permet d'identifier de façon unique les uplets de la table (ou relation) *seconde*. La seule connaissance de la clé primaire permet d'identifier toute ligne de la table.
   Cet identifiant (souvent auto géré) unique est ici associé à un élève, par essence unique dans l'établissement.
2. **Écrire une requête SQL d'insertion permettant d'enregistrer l'élève ACHIR Mussa dans la table seconde. Les informations relatives à cet élève sont données dans la ligne 1 du fichier seconde_lyc.csv.**

num_eleve	nom	prenom	datenaissance	langue1	langue2	option	classe
133310FE	ACHIR	Mussa	01/01/2005	anglais	espagnol		2A
156929JJ	ALTMEYER	Yohan	05/05/2005	allemand	anglais	théâtre	2D

Remarque

Un problème ici car la clé primaire proposée n'est pas un entier. Généralement cette clé est automatiquement attribuée par la bdd. On ne va prendre en compte que la partie de la clé composée de chiffres.

On ne complète pas la colonne **option** car l'élève n'en a pas. On suppose évidemment que cet attribut peut être NULL. cela doit être spécifié lors de la création de la table.

```
INSERT INTO seconde(num_eleve, langue1, langue2, classe)
VALUES (133310,'anglais','espagnol','2A') ;
```

3. **Lors de l'insertion de l'élève ALTMEYER Yohan (ligne 2 du fichier seconde_lyc.csv), une erreur de saisie a été commise sur la première langue, qui devrait être allemand. Écrire une requête SQL de mise à jour corrigeant les données de cet élève.**

```
UPDATE seconde
SET langue1 = 'allemand'
WHERE num_eleve = 156929 ;
```

Question 2

On suppose maintenant que la table seconde contient les informations issues de la figure 1 (ni plus, ni moins, même si la figure 1 n'est qu'un extrait du fichier seconde_lyc.csv).

1. **Quel est le résultat de la requête SELECT num_eleve FROM seconde;?**

```
SELECT num_eleve FROM seconde ;
```

   Le résultat de la requête sera la colonne des clefs primaires *num_eleve* :

num_eleve (clef primaire)
133310
156929
...
666702

2. **On rappelle qu'en SQL, la fonction d'agrégation COUNT() permet de compter le nombre d'enregistrements dans une table. Quel est le résultat de la requête SELECT COUNT(num_eleve) FROM seconde;?**

```
SELECT COUNT(num_eleve) FROM seconde ;
```

   Le résultat de la requête sera le nombre de lignes (de clefs primaires) de la table *seconde* donc ici 30.

3. **Écrire la requête permettant de connaître le nombre d'élèves qui font allemand en langue1 ou langue2.**

```
SELECT COUNT(num_eleve) FROM seconde
WHERE langue1='allemand' OR langue2='allemand' ;
```

Question 3

Le chef d'établissement souhaite faire évoluer la structure de sa base de données. Pour ce faire, il créé une nouvelle table *eleve* dont la structure est la suivante :

eleve	Type
num_eleve (clef primaire, clef étrangère de la table seconde)	entier (??)
nom	CHAR
prenom	CHAR
datenaissance	CHAR

Là encore, l'attribut num_eleve est un entier, les autres sont des chaînes de caractère (le type CHAR).

1. **Expliquer ce qu'apporte l'information clef étrangère pour l'attribut num_eleve de cette table en termes d'intégrité et de cohérence.**
   Les clés étrangères permettent de gérer des relations entre plusieurs tables, et garantissent la cohérence des données. On peut ainsi modifier des données d'un élève sans avoir à modifier plusieurs tables.

2. **On suppose la table eleve correctement créée et complétée. Le chef d'établissement aimerait lister les élèves (nom, prénom, date de naissance) de la classe 2A. Écrire la commande qui permet d'établir cette liste à l'aide d'une jointure entre *eleve* et *seconde*.**

seconde	Type
num_eleve (clef primaire)	entier (??)
langue1	CHAR
langue2	CHAR
option	CHAR
classe	CHAR

eleve	Type
num_eleve (clef primaire, clef étrangère de la table seconde)	entier (??)
nom	CHAR
prenom	CHAR
datenaissance	CHAR

```
SELECT nom, prenom ,datenaissance
        FROM eleve
        INNER JOIN seconde
        ON seconde.num_eleve = eleve.num_eleve
        WHERE seconde.classe='2A' ;
```

Question 4

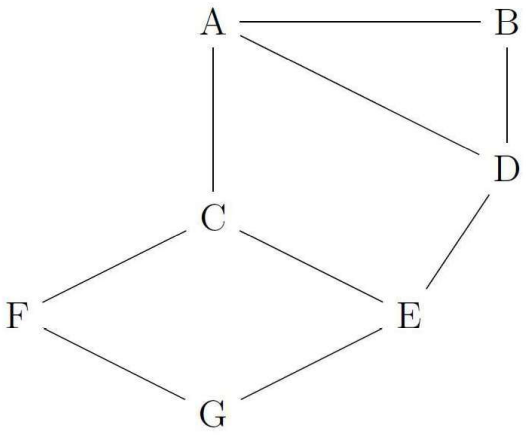
Proposer la structure d'une table *coordonnees* dans laquelle on pourra indiquer, pour chaque élève, son adresse, son code postal, sa ville, son adresse mail. Préciser la clef primaire et/ou la clé étrangère en vue de la mise en relation avec les autres tables.

coordonnees	Type
num_eleve (clef primaire, clef étrangère de la table seconde)	entier (??)
adresse	CHAR ou TEXT (pour des données plus longues)
codepostal	INT
ville	CHAR
email	CHAR

Exercice 5 : Réseaux en général et les protocoles RIP et OSPF en particulier

Le protocole RIP

Le protocole RIP permet de construire les tables de routage des différents routeurs, en indiquant pour chaque routeur la distance, en nombre de sauts, qui le sépare d'un autre routeur.



Question 1

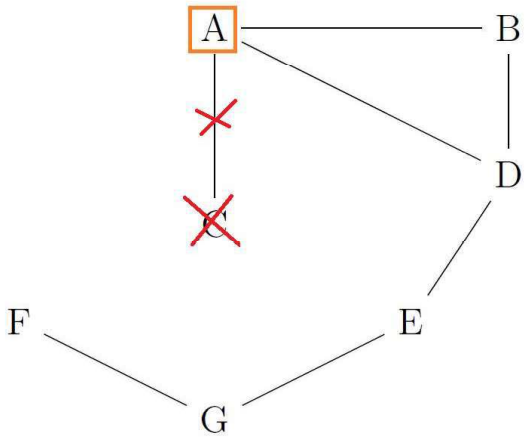
1. **Le routeur A doit transmettre un message au routeur G, en effectuant un nombre minimal de sauts. Déterminer le trajet parcouru.**
   Il y a deux trajets possible ACFG et ACEG. La distance est de 3.
2. **Déterminer une table de routage possible pour le routeur G obtenu à l'aide du protocole RIP.**

Table de routage de G		
Destination	Routeur Suivant	Distance
A	E (ou F)	3
B	E	3
C	E (ou F)	2
D	E	2
E	E	1
F	F	1

Question 2

Le routeur C tombe en panne. Reconstruire la table de routage du routeur A en suivant le protocole RIP.

Table de routage de A		
Destination	Routeur Suivant	Distance
B	B	1
D	D	1
E	D	2
F	D	4
G	D	3

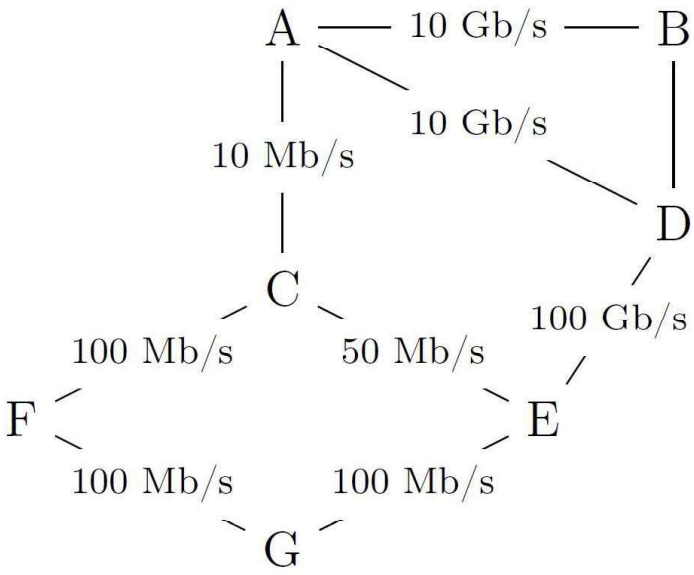


Le protocole OSPF

Contrairement au protocole RIP, l'objectif n'est plus de minimiser le nombre de routeurs traversés par un paquet. La notion de distance utilisée dans le protocole OSPF est uniquement liée aux coûts des liaisons. L'objectif est alors de minimiser la somme des coûts des liaisons traversées. Le coût d'une liaison est donné par la formule suivante :

$$coût = \frac{10^8}{d}$$

où d est la bande passante en bits/s entre les deux routeurs. On a rajouté sur le graphe représentant le réseau précédent les différents débits des liaisons. On rappelle que 1 Gb/s = 1 000 Mb/s = 10^9 bits/s.



Question 3

1. **Vérifier que le coût de la liaison entre les routeurs A et B est 0,01.**
   Entre A et B la distance est de $d = 10Gb/s = 10^{10}$ bits/s. Donc le coût est :

$$coût = \frac{10^8}{10^{10}} = 10^{-2} = \underline{0,01}$$

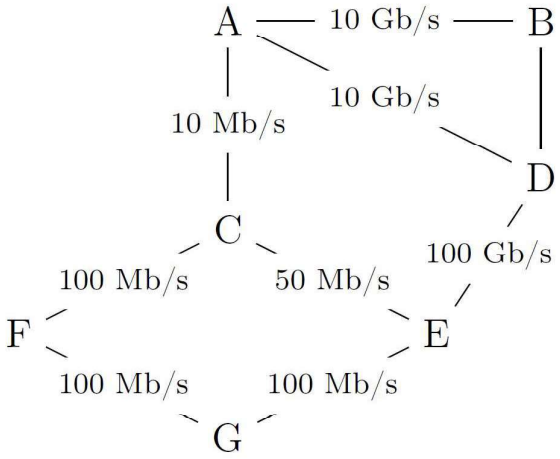
2. **La liaison entre le routeur B et D a un coût de 5. Quel est le débit de cette liaison ?**
   On a :

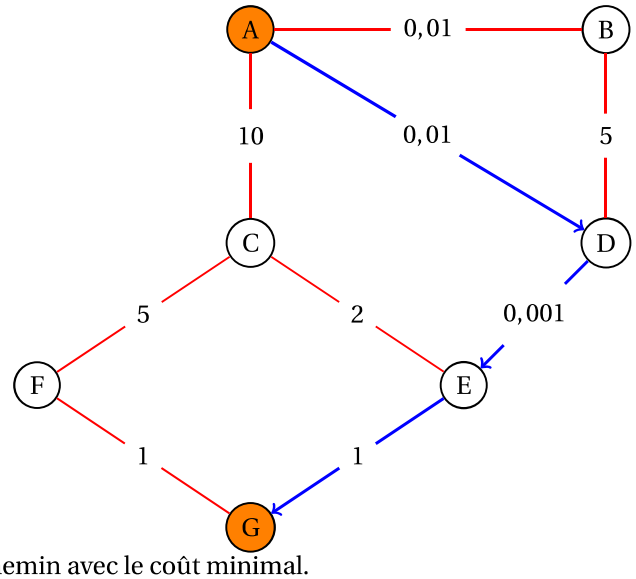
$$coût = \frac{10^8}{d} = 5 \Longleftrightarrow d = \frac{10^8}{5} = 2.10^7 \text{ bits /s} = \underline{20 \text{ Mb/s}}.$$

Question 4

Le routeur A doit transmettre un message au routeur G, en empruntant le chemin dont la somme des coûts sera la plus petite possible. Déterminer le chemin parcouru. On indiquera le raisonnement utilisé.
On construit le graphe en indiquant les coûts sur chaque arête.

Débit d bits/s	Coût $= \frac{10^8}{d}$
$d = 10Gb/s = 10^{10} bits/s$	coût $= 10^8 \times 10^{-10} = 10^{-2} = 0,01$
$d = 100Gb/s = 10^{11} bits/s$	coût $= 10^8 \times 10^{-11} = 10^{-3} = 0,001$
$d = 100Mb/s = 10^8 bits/s$	coût $= 10^8 \times 10^{-8} = 1$
$d = 50Mb/s = 5 \times 10^7 bits/s$	coût $= 0,2 \times 10^8 \times 10^{-7} = 2$
$d = 10Mb/s = 10^7 bits/s$	coût $= 10^8 \times 10^{-7} = 10$





On utilise l'algorithme de **Dijkstra** pour trouver le chemin avec le coût minimal.

Départ	A	B	C	D	E	F	G	Sommet choisi
	0	∞	∞	∞	∞	∞	∞	A
A		0,01 (A)	10 (A)	0,01(A)	∞	∞	∞	B (A)
B			10 (A)	~~10,01 (B)~~ 0,01(A)	∞	∞	∞	D (A)
D			10 (A)		0,011(D)	∞	∞	E(D)
E			~~10 (A)~~ 2,011(E)			∞	1,011(E)	G (E)
G						2,011(G)		F (G)
F			~~7,011 (F)~~ 2,011(E)					C (E)

Le parcourt avec un coût minimal pour aller de A à G est donc ADEG dont le coût est 1,011.

$$A \xrightarrow{0,01} D \xrightarrow{0,001} E \xrightarrow{1} G$$