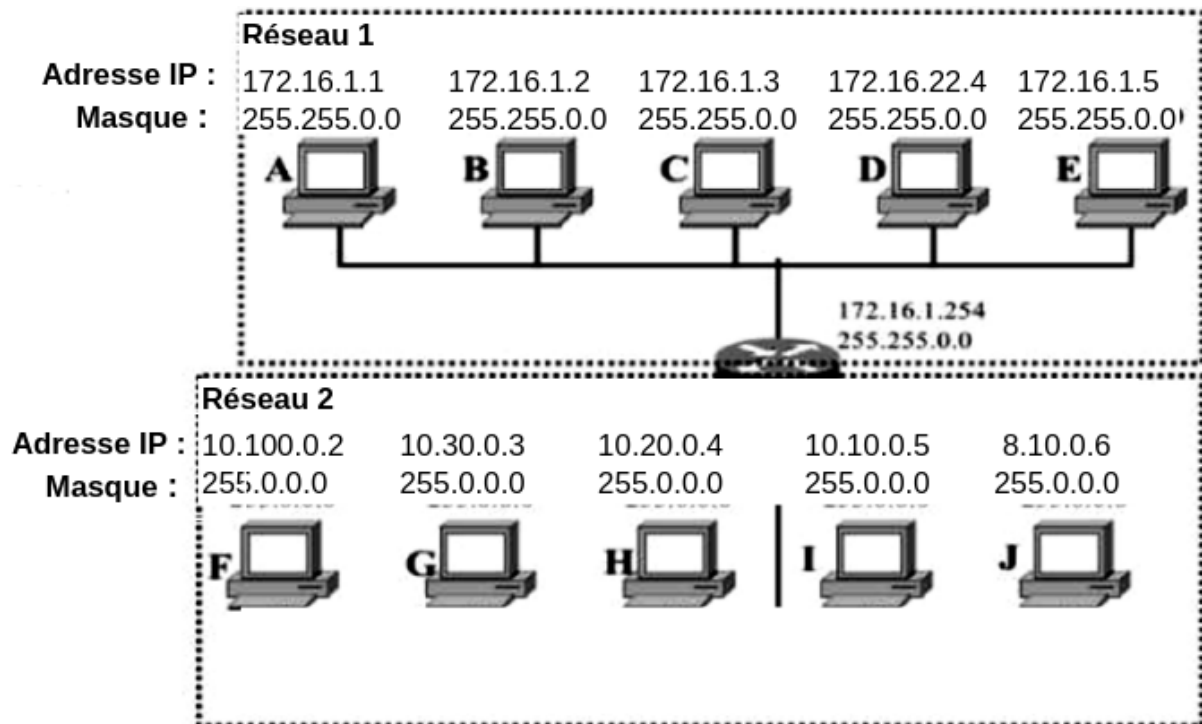


# DS NSI : architecture réseau et bases de données : SQL - corrigé

## Exercice 1 : Architecture réseau

Les parties A et B sont indépendantes.

### Partie A (5 points)



- (1 pt) Donner l'identifiant (adresse) réseau de la machine A et celui de la machine F.

✓ **Corrigé :** Adresse du réseau de la machine A : 172.16.0.0.  
Adresse du réseau de la machine F : 10.0.0.0.

- (2 pts) Un utilisateur signale que sa machine ne parvient pas à communiquer avec les autres machines de son réseau. De quelle machine s'agit-il ? Proposer une solution qui peut être mise en place par l'ingénieur réseau.

✓ **Corrigé :** Il s'agit de la machine J, dont l'identifiant réseau est 8.0.0.0 et pas 10.0.0.0.  
L'ingénieur réseau peut lui affecter l'adresse 10.10.0.6 (d'autres adresses sont possibles).

- (2 pt) Donner le nombre d'hôtes (machines) pouvant être adressés, au maximum, sur le réseau 1. Justifier le calcul.

✓ **Corrigé :**  $256^2 - 2 = 65\,534$  (de 0 à 255, on a 256 valeurs ; 2 choix de 256 valeurs :  $256^2$  ; on retire l'adresse réseau et broadcast : -2).

### Partie B (10 points)

- Un réseau local est relié à internet à l'aide d'une box faisant office de routeur. Un utilisateur connecte un nouvel ordinateur à ce réseau et veut tester son fonctionnement.

Il utilise en premier la commande linux `ifconfig` qui correspond à `ipconfig` sous environnement Windows. Cela lui donne le résultat suivant :

```
1 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
2   inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
3   inet6 fe80::761a:3e85:cc97:6491 prefixlen 64 scopeid 0x20<link>
4   ether 08:00:27:8b:c3:91 txqueuelen 1000 (Ethernet)
5   RX packets 136 bytes 13703 (13.3 KiB)
6   RX errors 0 dropped 0 overruns 0 frame 0
7   TX packets 180 bytes 17472 (17.0 KiB)
8   TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
9   device interrupt 9 base 0xd020
```

✍️ **Corrigé :**

- Adresse IPv4 de l'ordinateur : 10.0.2.15
- Masque de sous-réseau : 255.255.255.0
- Adresse IPv6 de l'ordinateur : fe80::761a:3e85:cc97:6491
- Adresse de diffusion (broadcast) du réseau : 10.0.2.255
- Adresse MAC de l'interface réseau l'ordinateur : 08:00:27:8b:c3:91

- a. (2 pts) Extraire les informations et les écrire ci-dessous :
- b. (2 pt) Pour tester sa connection internet, l'utilisateur fait alors un `tracert` vers un site distant.  
La première ligne sortie est la ligne suivante :

1 gateway (10.0.2.2) 0.328 ms 0.275 ms 0.267 ms

Quel nom précis porte le matériel correspondant l'adresse 10.0.2.2 ? Fait-il partie du même sous-réseau que l'utilisateur (pas de justification demandée) ?

✍️ **Corrigé :** Il s'agit de la **passerelle** (gateway), 1er routeur permettant un accès vers l'extérieur. Oui, une de ses interfaces est dans le même sous-réseau que le poste de l'utilisateur.

2. Cinq routeurs R1, R2, R3, R4, R5 sont connectés dans un réseau avec les caractéristiques suivantes :

Routeur R1			
Destination	Direction	Saut	Débit (Mbits/s)
R2	R2	1	10
R3	R3	1	100
R4	R2	2	
R5	R5	1	10

Routeur R2			
Destination	Direction	Saut	Débit (Mbits/s)
R1	R1	1	10
R3	R3	1	100
R4	R4	1	10
R5	R1	2	

Routeur R3			
Destination	Direction	Saut	Débit (Mbits/s)
R1	R1	1	100
R2	R2	1	100
R4	R2	2	
R5	R1	2	

Routeur R4			
Destination	Direction	Saut	Débit (Mbits/s)
R1	R2	2	
R2	R2	1	10
R3	R2	2	
R5	R2	3	

Routeur R5			
Destination	Direction	Saut	Débit (Mbits/s)
R1	R1	1	10
R2	R1	2	
R3	R1	2	
R4	R1	3	

Dans cette question, on utilise le protocole de routage RIP, qui cherche à minimiser le nombre de sauts.

- a. (2 pts) Compléter le schéma ci-dessous qui représente le réseau : indiquer les numéros des routeurs et tracer les connexions entre eux.

✍️ **Corrigé :**

```
R1 - R5
/ |
R3 |
\ |
R2 - R4
```

- b. (2 pts) Quelle route faut-il prendre pour aller de R4 à R5 ?

✍️ **Corrigé :** R4 → R2 → R1 → R5

3. (2 pts) Les cinq routeurs précédents sont connectés dans la même configuration que précédemment. Toutefois le protocole de routage appliqué est désormais le protocole OSPF qui prend en compte le débit (Mbits/s) pour minimiser le coût total de la transmission.

Le coût  $C$  pour passer d'un routeur à un autre est donné par la formule :  $C = \frac{100}{\text{débit}}$

Quelle route faut-il prendre pour aller de R4 à R5 en respectant le protocole OSPF ?

✍️ **Corrigé :** Le chemin doit minimiser le coût total, on a donc la route : R4 → R2 → R3 → R1 → R5 (coût = 22).

## Exercice 2 : Bases de données et langage SQL (15 points)

Cet exercice traite du thème « base de données », et principalement du modèle relationnel et du langage SQL.

L'énoncé de cet exercice peut utiliser les mots du langage SQL suivants : CREATE TABLE, SELECT, FROM, WHERE, JOIN ON, INSERT INTO, VALUES, UPDATE, SET, DELETE, COUNT, DISTINCT, AND, OR, AS, ORDER BY, ASC, DESC

Un site web recueille des données de navigation dans une base de données afin d'étudier les profils de ses visiteurs. Chaque requête d'interrogation d'une page de ce site est enregistrée dans une première table dénommée `Visites` sous la forme d'un 5-uplet :

(identifiant, adresse IP, date et heure de visite, nom de la page, navigateur) Lorsqu'un client change de page sur le site, un nouvel identifiant lui est attribué (il restera reconnu par son adresse ip).

Le chargement de la page `index.html` par 192.168.1.91 le 12 juillet 2023 à 22h48 aura par exemple été enregistré de la façon suivante :

(1534, "192.168.1.91", "2023-07-12 22:48:00", "index.html", "Internet explorer 10").

1. Le script SQL ayant permis de créer cette table est le suivant :

```
1 CREATE TABLE Visites (
2     identifiant INTEGER NOT NULL UNIQUE,
3     ip VARCHAR(15),
4     dateheure DATETIME,
5     nompage TEXT,
```

```
navigateur TEXT  
);
```

- a. (1 pt) Donner une commande d'interrogation en langage SQL permettant d'obtenir l'ensemble, dans cette table, des 2-uplets :

adresse IP, nom de la page **Corrigé :** `SELECT ip, nompage FROM Visites;`

- b. (1 pt) Donner une commande en langage SQL permettant d'obtenir l'ensemble des adresses IP ayant interrogé le site, sans doublon. **Corrigé :** `SELECT DISTINCT ip FROM Visites;`

- c. (1 pt) Donner une commande en langage SQL permettant d'obtenir la liste des noms des pages visitées par l'adresse IP

192.168.1.91. **Corrigé :** `SELECT nompage FROM Visites WHERE ip='192.168.1.91';`

2. Ce site web met en place, sur chacune de ses pages, un programme en javascript qui envoie au serveur, à intervalle régulier de 15 secondes, le temps en secondes de présence sur la page.

Ces envois contiennent tous la valeur de identifiant correspondant au chargement initial de la page.

Par exemple, si le visiteur du 12 juillet 1998 est resté 65 secondes sur la page, celle-ci a envoyé au serveur les 4 doublets (1534, 15), (1534, 30), (1534, 45) et (1534, 60).

Ces données sont enregistrées dans une table nommée `Pings` créée avec la commande ci-dessous :

```
1 CREATE TABLE Pings (  
2     identifiant INTEGER,  
3     duree INTEGER  
4     FOREIGN KEY (identifiant) REFERENCES Visites(identifiant)  
5 );
```

En plus de l'inscription d'une ligne dans la table `Visites`, chaque chargement d'une nouvelle page provoque l'insertion d'une ligne dans la table `Pings` comprenant l'identifiant de ce chargement et une durée de 0.

Les attributs identifiant des tables `Visites` et `Pings` partagent les mêmes valeurs.

- a. (1 pt) De quelle table l'attribut identifiant est-il la clé primaire ?

**Corrigé :** L'attribut `identifiant` est la clé primaire de la table `Visites` (un nouvel identifiant est fourni si le client change de page, d'après l'énoncé).

- b. (1 pt) De quelle table l'attribut identifiant est-il une clé étrangère ?

**Corrigé :** L'attribut `identifiant` est la clé étrangère de la table `Pings`.

- c. (2 pts) Par conséquent, quelles vérifications, concernant ces contraintes (clé primaire et clé étrangère), sont automatiquement effectuées par le système de gestion de base de données ?

**Corrigé :** Le SGBD va vérifier que l'attribut `identifiant` est unique pour chaque p-uplet de la table `Visites`. Le SGBD va aussi vérifier que l'attribut `identifiant` de la table `Pings` correspond bien à un attribut identifiant de la table `Visites` pour chaque p-uplet de la table `Pings`.

3. (2 pts) Le serveur reçoit le doublet (`identifiant, duree`) suivant : (1534, 105).

**Corrigé :**

```
1 INSERT INTO Pings(identifiant, duree)  
2 VALUES (1534, 105);
```

Écrire la commande SQL d'insertion qui permet d'ajouter cet enregistrement à la table `Pings`.

4. On envisage ensuite d'optimiser la table en se contentant d'une seule ligne par identifiant dans la table `Pings` : les valeurs de l'attribut `duree` devraient alors être mises à jour (UPDATE) à chaque réception d'un nouveau doublet (`identifiant, duree`).

- a. (2 pts) Écrire la requête de mise à jour permettant de fixer à 120 la valeur de l'attribut `duree` associée à l'identifiant 1534 dans

la table `Pings`. **Corrigé :**

```
1 UPDATE Pings  
2 SET duree = 120  
3 WHERE identifiant = 1534;
```

- b. (2 pts) Lorsque les données envoyées depuis le navigateur d'un client, via plusieurs requêtes formulées en javascript, arrivent au serveur, elles ne sont pas forcément dans l'ordre. Expliquer quel problème cela peut générer de se contenter de mettre à jour une seule ligne par identifiant dans la table `Pings` en utilisant UPDATE.

**Corrigé :** Si le message annonçant 15 secondes de connexion à une nouvelle page arrive avant le message de connexion initiale à cette page (le seul qui crée une entrée), le serveur peut demander au SGBD de modifier une entrée alors qu'elle n'est pas encore créée. Avec une durée de 15s, ce type d'incident serait rare mais pas impossible.

5. (2 pts) Écrire une requête SQL utilisant le mot-clé `JOIN` et une clause `WHERE`, permettant de trouver les noms de toutes les pages qui ont

été consultées plus d'une minute par au moins un utilisateur.

**Corrigé :**

```
1 SELECT nompage  
2 FROM Visites  
3 JOIN Pings ON Visites.identifiant = Pings.identifiant  
4 WHERE Pings.duree > 60;
```