EXERCICE 3 (4 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)`.

Le chargement de la page index.html par 192.168.1.91 le 12 juillet 1998 à 22h48 aura par exemple été enregistré de la façon suivante :
`(1534, "192.168.1.91", "1998-07-12 22:48:00", "index.html", "Internet explorer 4.1")`.

La commande SQL ayant permis de créer cette table est la suivante:

```
CREATE TABLE Visites (
    identifiant INTEGER NOT NULL UNIQUE,
    ip VARCHAR(15),
    dateheure DATETIME,
    nompage TEXT,
    navigateur TEXT
);
```

1. a. Donner une commande d'interrogation en langage SQL permettant d'obtenir l'ensemble des 2-uplets (adresse IP, nom de la page) de cette table.

   b. Donner une commande en langage SQL permettant d'obtenir l'ensemble des adresses IP ayant interrogé le site, sans doublon.

   c. Donner une commande en langage SQL permettant d'obtenir la liste des noms des pages visitées par l'adresse IP 192.168.1.91

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 :

```
CREATE TABLE Pings (
    identifiant INTEGER,
    duree INTEGER
);
```

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.

2. a. De quelle table l'attribut `identifiant` est-il la clé primaire ?

   b. De quelle table l'attribut `identifiant` est-il une clé étrangère ?

   c. Par conséquent, quelles vérifications sont automatiquement effectuées par le système de gestion de base de données ?

3. Le serveur reçoit le doublet `(identifiant, duree)` suivant : (1534, 105). Écrire la commande SQL d'insertion qui permet d'ajouter cet enregistrement à la table `Pings`.

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 à chaque réception d'un nouveau doublet `(identifiant, duree)`.

4. a. 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`.

   b. Expliquer pourquoi on ne peut pas être certain que les données envoyées par une page web, depuis le navigateur d'un client, via plusieurs requêtes formulées en javascript, arrivent au serveur dans l'ordre dans lequel elles ont été émises.

   c. En déduire qu'il est préférable d'utiliser une requête d'insertion plutôt qu'une requête de mise à jour pour ajouter des données à la table `Pings`.

5. Écrire une requête SQL utilisant le mot-clef `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.