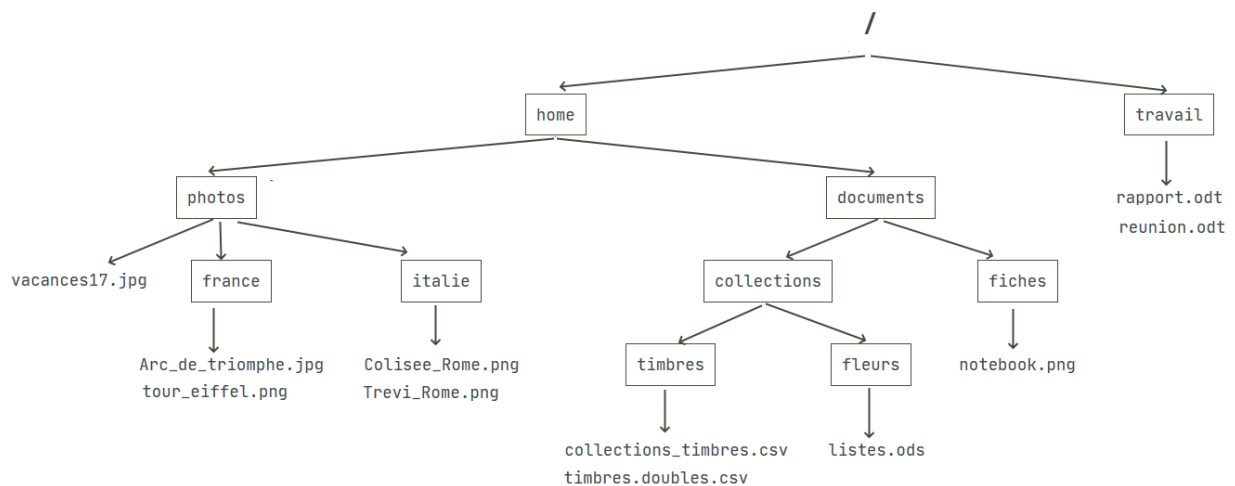


EXERCICE 1 (4 points)

Cet exercice est consacré aux commandes de base en lignes de commande sous Linux, au traitement des données en tables et aux bases de données (modèle relationnel, base de données relationnelle, langage SQL).

1. Sur une machine équipée du système d'exploitation GNU/Linux, les informations sont enregistrées dans un fichier du répertoire `collections`.

Dans le schéma ci-dessous, on trouve des répertoires (encadrés par un rectangle, exemple : `travail`) et des fichiers (les noms non encadrés, comme `rapport.odt`).



- a. Sachant que le répertoire courant est le répertoire `fiches`, indiquer sur la copie les numéros de toutes les commandes parmi celles proposées ci-dessous qui permettent de se positionner dans le répertoire `timbres`:

Commande 1 : `cd /home/documents/collections/timbres`

Commande 2 : `cd ../collections/timbres`

Commande 3 : `cd /timbres`

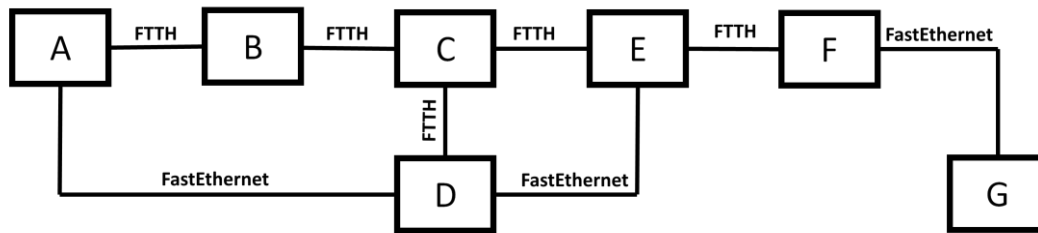
Commande 4 : `cd documents/collections/timbres`

Commande 5 : `cd ../collections/timbres`

Commande 6 : `cd timbres`

- b. Donner une commande qui permet d'accéder au répertoire `timbres` à partir de la racine.

2. On considère le réseau ci-dessous dans lequel :
 - les nœuds A, B, C, D, E, F et G sont des routeurs,
 - le type de liaison est précisé entre chaque routeur.



On rappelle que la bande passante des liaisons FTTH (fibre optique : Fiber To The Home) est de 10 Gbit/s et celle des liaisons FastEthernet de 100 Mbit/s.

On s'intéresse au protocole de routage OSPF. Le protocole OSPF est un protocole de routage qui cherche à minimiser la somme des coûts des liaisons entre les routeurs empruntés par un paquet. Le coût C d'une liaison est donné par :

$$C = \frac{10^8}{d}, \text{ où } d \text{ est la bande passante en bit/s de la liaison.}$$

- Calculer le coût d'une liaison de communication par la technologie FastEthernet.
 - Le fichier `collections_timbres.csv` contenu dans une machine reliée au routeur A doit être envoyé à une machine reliée au routeur G. Déterminer la route permettant de relier le routeur A au routeur G et minimisant la somme des coûts selon le protocole OSPF.
3. Un extrait des informations stockées dans le fichier `collection_timbres.csv` au format CSV est donné ci-dessous :

```

nom_timbre;annee_fabrication;nom_collectionneur
Gustave Eiffel;1950;Dupont
Marianne;1989;Durand
Alan Turing;2012;Dupont

```

Donner les différents descripteurs de ce fichier CSV. Pour chacun de ces descripteurs, donner les valeurs associées.

4. On cherche maintenant à stocker une partie de ces informations dans une base de données relationnelle. La relation suivante a été proposée :

timbres

nom	annee_fabrication
Gustave Eiffel	1950
Marianne	1989
Alan Turing	2012
Gustave Eiffel	1989
Ada Lovelace	1951

- Définir la notion de clé primaire d'une relation.
- L'attribut `nom` peut-il jouer le rôle de clé primaire ?
- Même question pour l'attribut `annee_fabrication`.

- d. Dans le cas d'une réponse par la négative aux deux questions ci-dessus, proposer une solution permettant d'avoir une clé primaire dans la relation timbres.

5. On considère maintenant la relation suivante :

collectionneurs

ref_licence	nom	prenom	annee_naissance	nbre_timbres
Hqdfapo	Dupuis	Daniel	1953	53
Dfacqpe	Dupond	Jean-Pierre	1961	157
Qdfqnay	Zaoui	Jamel	1973	200
Aerazri	Pierre	Jean	1967	130
Nzxoeqg	Dupond	Alexandra	1960	61

- a. Décrire le résultat de la requête SQL ci-dessous :

```
UPDATE collectionneurs
SET ref_licence = 'Ythpswz'
WHERE nom = 'Dupond';
```

- b. Expliquer pourquoi, suite à cette requête, l'attribut `ref_licence` ne peut plus être une clé primaire de la relation `collectionneurs`.

6. Écrire une requête SQL qui affiche, pour chacune des personnes nées en 1963 ou après, le nom, le prénom et le nombre de timbres qu'elles possèdent dans leur collection, enregistrées dans la relation `collectionneurs`. On pourra utiliser certaines des clauses usuelles suivantes : `SELECT`, `FROM`, `WHERE`, `JOIN`, `UPDATE`, `INSERT`, `DELETE`.

EXERCICE 1 (3 points)

L'exercice porte sur les bases de données et les types construits de données.

On pourra utiliser les mots clés SQL suivants : AND, FROM, INSERT, INTO, JOIN, OR, ON, SELECT, SET, UPDATE, VALUES, WHERE.

On étudie une base de données permettant la gestion de l'organisation d'un festival de musique de jazz, dont voici le schéma relationnel comportant trois relations :

- la relation `groupes` (`idgrp`, `nom`, `style`, `nb_pers`)
- la relation `musiciens` (`idmus`, `nom`, `prenom`, `instru`, `#idgrp`)
- la relation `concerts` (`idconc`, `scene`, `heure_debut`, `heure_fin`, `#idgrp`)

Dans ce schéma relationnel :

- les clés primaires sont soulignées ;
- les clés étrangères sont précédées d'un #.

Ainsi `concerts.idgrp` est une clé étrangère faisant référence à `groupes.idgrp`.

Voici un extrait des tables `groupes`, `musiciens` et `concerts` :

extrait de groupes			
idgrp	nom	style	nb_pers
12	'Weather Report'	'Latin Jazz'	5
25	'Breckers Brothers'	'Swing Jazz'	4
87	'Return to Forever'	'Latin Jazz'	8
96	'The Jazz Messenger'	'Free Jazz'	3

extrait de musiciens				
idmus	nom	prenom	instru	idgrp
12	'Parker'	'Charlie'	'trompette'	96
13	'Parker'	'Charlie'	'trombone'	25
58	'Dufler'	'Candy'	'saxophone'	96
97	'Miles'	'Davis'	'saxophone'	87

extrait de concerts				
idconc	scene	heure_debut	heure_fin	idgrp
10	1	'20h00'	'20h45'	12
24	2	'20h00'	'20h45'	15
36	1	'21h00'	'22h00'	96
45	3	'18h00'	'18h30'	87

Figure 1 : Extrait des tables `groupes`, `musiciens` et `concerts`

1. Citer les attributs de la table `groupes`.
2. Justifier que l'attribut `nom` de la table `musiciens` ne peut pas être une clé primaire.
3. En s'appuyant uniquement sur l'extrait des tables fourni dans la figure 1 écrire ce que renvoie la requête :

```
SELECT nom
FROM groupes
WHERE style = 'Latin Jazz';
```

4. Le concert dont l'idconc est 36 finira à 22h30 au lieu de 22h00. Recopier sur la copie et compléter la requête SQL ci-dessous permettant de mettre à jour la relation concerts pour modifier l'horaire de fin de ce concert.

```
UPDATE concerts
SET ...
WHERE ... ;
```

5. Donner une requête SQL permettant de récupérer le nom de tous les groupes qui jouent sur la scène 1.

6. Fournir une requête SQL permettant d'ajouter dans la relation groupes le groupe 'Smooth Jazz Fourplay', de style 'Free Jazz', composé de 4 membres. Ce groupe aura un idgrp de 15.

Les données sont ensuite récupérées pour être analysées par la société qui produit les festivals de musique. Pour ce faire, elle utilise la programmation en Python afin d'effectuer certaines opérations plus complexes.

Elle stocke les données relatives aux musiciens sous forme d'un tableau de dictionnaires dans laquelle a été ajouté le nombre de concerts effectués par chaque musicien :

```
>>> print(musiciens)
[{'idmus': 12, 'nom': 'Parker', 'prenom': 'Charlie',
  'instru': 'trompette', 'idgrp' : 96, 'nb_concerts': 5},
 {'idmus': 13, 'nom': 'Parker', 'prenom': 'Charlie',
  'instru': 'trombone', 'idgrp' : 25, 'nb_concerts': 9},
 {'idmus': 58, 'nom': 'Dufler', 'prenom': 'Candy',
  'instru': 'saxophone', 'idgrp' : 96, 'nb_concerts': 4},
 {'idmus': 97, 'nom': 'Miles', 'prenom': 'Davis',
  'instru': 'saxophone', 'idgrp' : 87, 'nb_concerts': 2},
 ...
]
```

7. Écrire la fonction recherche_nom ayant pour unique paramètre un tableau de dictionnaires (comme musiciens présenté précédemment) renvoyant un tableau contenant le nom de tous les musiciens ayant participé à au moins 4 concerts.

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.