

## 0.1

### Les bases de données relationnelles

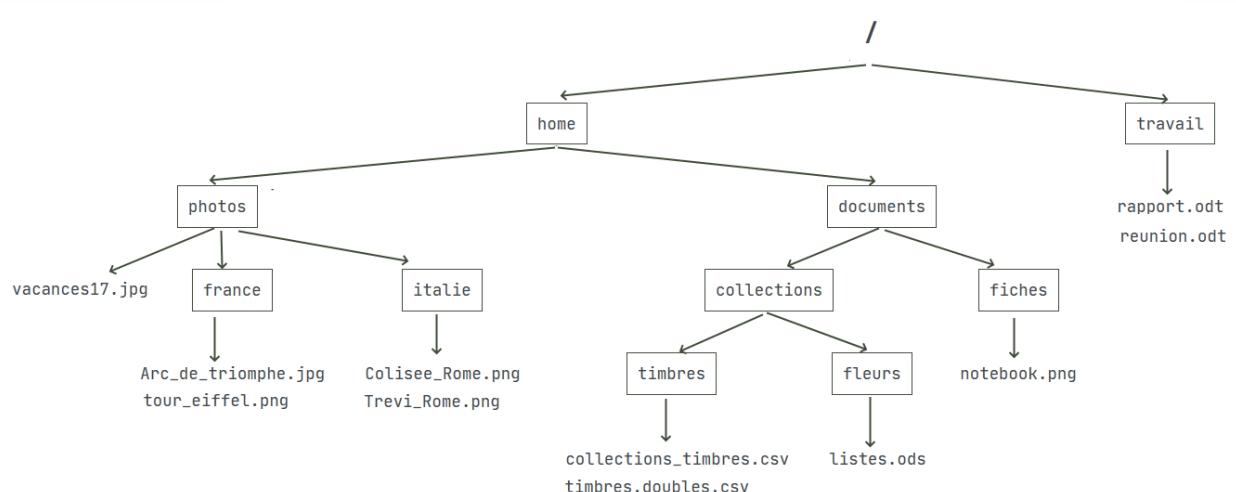
NSI TERMINALE - JB DUTHOIT

## 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).



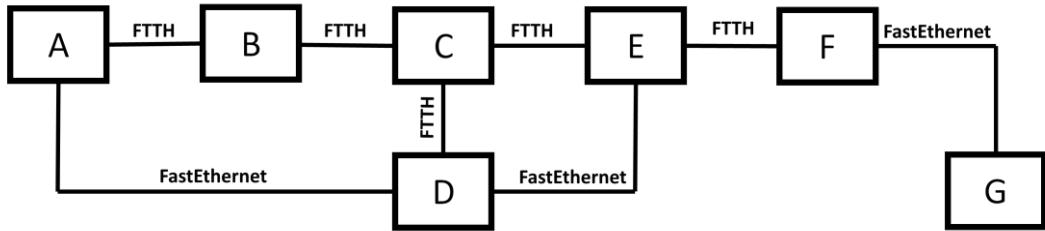
- a. Sachant que le répertoire courant est le répertoire fiches, indiquer sur la copie les numéros de toutes les commandes parmi celles proposées ci-dessous qui permettent de se positionner dans le répertoire timbres :

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

<b>nom</b>	<b>annee_fabrication</b>
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				extrait de musiciens				
idgrp	nom	style	nb_pers	idmus	nom	prenom	instru	idgrp
12	'Weather Report'	'Latin Jazz'	5	12	'Parker'	'Charlie'	'trompette'	96
25	'Breckers Brothers'	'Swing Jazz'	4	13	'Parker'	'Charlie'	'trombone'	25
87	'Return to Forever'	'Latin Jazz'	8	58	'Dufler'	'Candy'	'saxophone'	96
96	'The Jazz Messenger'	'Free Jazz'	3	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.

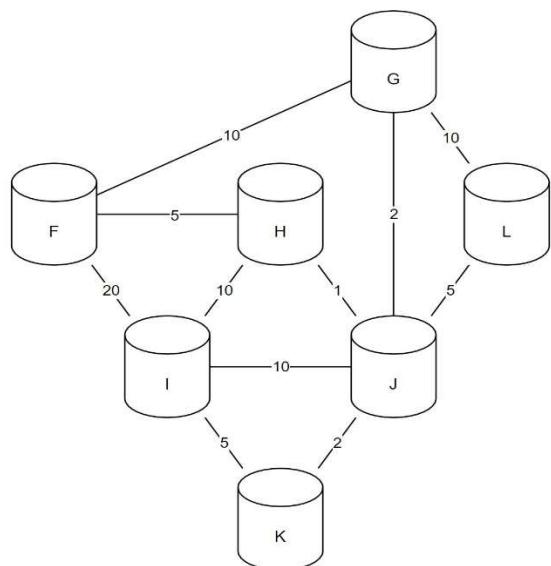
- Donner la liste des routeurs avec lesquels le routeur A est directement relié.
  - Représenter graphiquement et de manière sommaire les 5 routeurs ainsi que les liaisons existantes entre ceux-ci.
3. Le protocole OSPF est un protocole de routage qui cherche à minimiser la somme des métriques des liaisons entre routeurs.

Dans le protocole de routage OSPF le débit des liaisons entre routeurs agit sur la métrique via la relation :  $métrique = \frac{10^8}{débit}$  dans laquelle le débit est exprimé en bit par seconde (bps).

On rappelle qu'un kbps est égal à  $10^3$  bps et qu'un Mbps est égal à  $10^6$  bps.  
Recopier sur votre copie et compléter le tableau suivant :

Débit	100 kbps	500 kbps	?	100 Mbps
Métrique associée	1 000	?	10	1

4. Voici la représentation d'un réseau et la table de routage incomplète du routeur F obtenue avec le protocole OSPF :



Routeur F

Destination	Métrique
F	0
G	8
H	5
I	
J	
K	
L	

Les nombres présents sur les liaisons représentent les coûts des routes avec le protocole OSPF.

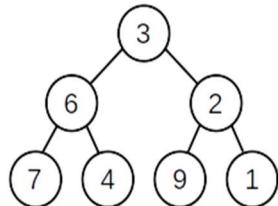
- Indiquer le chemin emprunté par un message d'un ordinateur du réseau F à destination d'un ordinateur du réseau I.  
Justifier votre réponse.
- Recopier et compléter la table de routage du routeur F.
- Citer une unique panne qui suffirait à ce que toutes les données des échanges de tout autre réseau à destination du réseau F transitent par le routeur G. Expliquer en détail votre réponse.

## EXERCICE 4 (4 points)

Cet exercice, composé de deux parties A et B, porte sur le parcours des arbres binaires, le principe “diviser pour régner” et la récursivité.

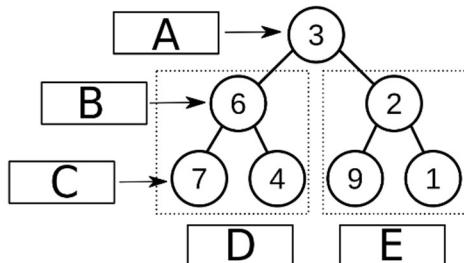
Cet exercice traite du calcul de la somme d'un arbre binaire. Cette somme consiste à additionner toutes les valeurs numériques contenues dans les nœuds de l'arbre.

L'arbre utilisé dans les parties A et B est le suivant :



### Partie A : Parcours d'un arbre

1. Donner la somme de l'arbre précédent. Justifier la réponse en explicitant le calcul qui a permis de l'obtenir.
2. Indiquer la lettre correspondante aux noms ‘racine’, ‘feuille’, ‘nœud’, ‘SAG’ (Sous Arbre Gauche) et ‘SAD’ (Sous Arbre Droit). Chaque lettre **A**, **B**, **C**, **D** et **E** devra être utilisée une seule fois.



Arbre avec les lettres à associer

3. Parmi les quatre propositions A, B, C et D ci-dessous, donnant un parcours en largeur d'abord de l'arbre, une seule est correcte. Indiquer laquelle.  
**Proposition A** : 7 - 6 - 4 - 3 - 9 - 2 - 1  
**Proposition B** : 3 - 6 - 7 - 4 - 2 - 9 - 1  
**Proposition C** : 3 - 6 - 2 - 7 - 4 - 9 - 1  
**Proposition D** : 7 - 4 - 6 - 9 - 1 - 2 - 3
4. Écrire en langage Python la fonction `somme` qui prend en paramètre une liste de nombres et qui renvoie la somme de ses éléments.  
Exemple : `somme([1, 2, 3, 4])` est égale à 10.

## EXERCICE 4 (4 points)

Cet exercice porte sur les bases de données relationnelles et le langage SQL.

L'énoncé de cet exercice utilise les mots clefs du langage SQL suivants : SELECT, FROM, WHERE, JOIN ON, UPDATE, SET, INSERT INTO VALUES, COUNT, ORDER BY.

- La clause ORDER BY suivie d'un attribut permet de trier les résultats par ordre croissant de l'attribut ;
- COUNT (\*) renvoie le nombre de lignes d'une requête ;

Un musicien souhaite créer une base de données relationnelle contenant ses morceaux et interprètes préférés. Pour cela il utilise le langage SQL. Il crée une table morceaux qui contient entre autres les titres des morceaux et leur année de sortie :

id_morceau	titre	annee	id_interprete
1	Like a Rolling Stone	1965	1
2	Respect	1967	2
3	Imagine	1970	3
4	Hey Jude	1968	4
5	Smells Like Teen Spirit	1991	5
6	I Want To hold Your Hand	1963	4

Il crée la table interpretes qui contient les interprètes et leur pays d'origine :

id_interprete	nom	pays
1	Bob Dylan	États-Unis
2	Aretha Franklin	États-Unis
3	John Lennon	Angleterre
4	The Beatles	Angleterre
5	Nirvana	États-Unis

id\_morceau de la table morceaux et id\_interprete de la table interpretes sont des clés primaires.

L'attribut id\_interprete de la table morceaux fait directement référence à la clé primaire de la table interpretes.

1.

- a. Écrire le résultat de la requête suivante :

```
SELECT titre FROM morceaux WHERE id_interprete = 4;
```

- b. Écrire une requête permettant d'afficher les noms des interprètes originaires d'Angleterre.

- c. Écrire le résultat de la requête suivante :

```
SELECT titre, annee FROM morceaux ORDER BY annee;
```

- d. Écrire une requête permettant de calculer le nombre de morceaux dans la table morceaux.

- e. Écrire une requête affichant les titres des morceaux par ordre alphabétique.

2.

- a. Citer, en justifiant, la clé étrangère de la table morceaux.

- b. Écrire un schéma relationnel des tables interpretes et morceaux.

- c. Expliquer pourquoi la requête suivante produit une erreur :

```
INSERT INTO interpretes VALUES (1, 'Trust', 'France');
```

3.

- a. Une erreur de saisie a été faite. Écrire une requête SQL permettant de changer l'année du titre « Imagine » en 1971.

- b. Écrire une requête SQL permettant d'ajouter l'interprète « The Who » venant d'Angleterre à la table interpretes. On lui donnera un id\_interprete égal à 6.

- c. Écrire une requête SQL permettant d'ajouter le titre « My Generation » de « The Who » à la table morceaux. Ce titre est sorti en 1965 et on lui donnera un id\_morceau de 7 ainsi que l'id\_interprete qui conviendra.

4. Écrire une requête permettant de lister les titres des interprètes venant des États-Unis.

## Exercice 1 (4 points).

*Cet exercice porte sur les bases de données relationnelles et le langage SQL.*

L'énoncé de cet exercice utilise les mots du langage SQL suivants :

SELECT, FROM, WHERE, JOIN, INSERT INTO, VALUES, UPDATE, SET, COUNT

On rappelle qu'en SQL, la fonction d'agrégation COUNT permet de compter le nombre d'enregistrements dans une table.

L'entreprise "Vacances autrement" propose des séjours d'une durée d'une semaine dans différentes stations situées en France. Les séjours sont organisés autour d'une seule activité sportive. Par exemple, la station "La Tramontane Catalane" située à Leucate propose une formule autour de la planche à voile. Les clients pourront durant toute la durée du séjour pratiquer ce sport : le matériel est fourni et deux sessions encadrées par des moniteurs diplômés sont organisées chaque journée. Cette même station propose aussi des séjours pour les amateurs de kitesurf.

Les réservations se font par l'intermédiaire d'un site internet : chaque client complète un formulaire avec ses données personnelles (nom, prénom...) et choisit le séjour souhaité.

Ce site s'appuie sur une base de données relationnelle. Elle contient notamment les relations Station, Sport, Client et Sejour.

Voici le schéma relationnel où les clefs primaires sont soulignées et les clefs étrangères sont suivies du symbole # :

```
Station (nomStation, ville, region)
Sport (nomSport, nomStation#, prix)
Client (cID, nom, prenom, mail)
Sejour (cID, semaine, annee, nomStation#,nomSport#)
```

- Les différentes stations sont stockées dans la relation Station. Chaque station est identifiée par un nom caractéristique (nomStation). Les attributs ville et region permettent de décrire la localisation de chacune des stations.
- La relation Sport fait le lien entre les stations et les activités sportives qui y sont proposées. Les trois attributs sont nomSport, nomStation et prix. L'attribut prix est un nombre entier : il correspond au montant plein tarif (en euros) du séjour d'une semaine dans la station avec le sport associé.
- La relation Client caractérise les clients de l'entreprise : ils sont identifiés par un nombre entier (cID) et décrits par leur nom, prénom et adresse mail.
- La relation Sejour permet de lister les séjours auxquels les clients ont participé. Les deux attributs semaine et annee sont des nombres entiers : ils permettent d'identifier le moment où a été effectué chaque séjour (semaine est le numéro de la semaine pendant laquelle le séjour a été réalisé).

**Exemple :** un séjour ayant été effectué la semaine numéro 12 de l'année 2020 correspond à semaine = 12 et à annee = 2020.

On donne aussi un extrait des trois premières lignes de chacune de ces relations :

**Station**

nomStation	ville	region
La tramontane catalane	Leucate	Occitanie
La baie sauvage	La Torche	Bretagne
La pinède	Calvi	Corse

**Sport**

nomSport	nomStation	prix
planche à voile	La tramontane catalane	1200
kitesurf	La tramontane catalane	1100
plongée	La baie sauvage	950

**Client**

cID	nom	prenom	mail
1	GENEREUX	Eric	eric.genereux@mail.fr
2	PIERRE	Daniel	daniel.pierre@mail.fr
3	JOLY	Emilie	emilie.joly@mail.fr

**Sejour**

cID	semaine	annee	nomStation	nomSport
1	26	2020	La tramontane catalane	planche à voile
1	38	2020	La tramontane catalane	planche à voile
2	33	2020	La baie sauvage	plongée

1. (a) Donner la clé primaire et les éventuelles clés étrangères de la relation Sport.  
(b) Citer une contrainte d'intégrité de domaine puis une contrainte d'intégrité de relation et enfin une contrainte d'intégrité de référence que doivent respecter les données de la relation Sport.
2. (a) Le tarif du séjour à la station "La tramontane catalane" basé sur la planche à voile passe de 1 200 euros à 1 350 euros.  
La requête SQL suivante a été utilisée pour la mise à jour du tarif :

```
INSERT INTO Sport VALUES ("planche à voile","La tramontane catalane",1350);
```

Cette requête a été rejetée et la mise à jour n'a pas été effectuée. Après avoir expliqué pourquoi cette requête a été refusée, proposer une requête SQL qui permettra de modifier le tarif de ce séjour.

- (b) Une nouvelle station vient d'être référencée. Son nom est "Soleil Rouge". Elle est située à Bastia en Corse. Des séjours d'une semaine y seront organisés. Le sport pratiqué sera la plongée au tarif de 900 euros.

Écrire les requêtes SQL permettant d'insérer ces nouvelles données dans la base.

3. (a) L'agence souhaite envoyer un mail d'information à ses clients pour leur présenter cette nouvelle possibilité de séjour mise en place à Bastia. Écrire une requête SQL permettant d'obtenir l'adresse mail de tous les clients.
- (b) Un client qui pratique la plongée souhaite réserver un séjour. Écrire une requête SQL permettant d'obtenir le nom de toutes les stations où l'on peut pratiquer la plongée.

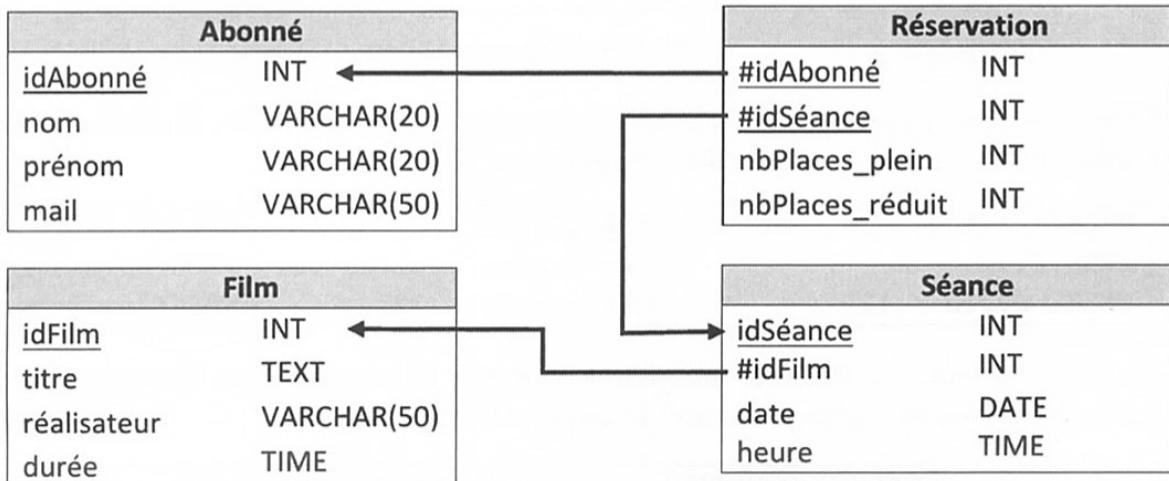
4. (a) Pour faire son choix, le client souhaiterait connaître les villes où sont situées les stations dans lesquelles il pourra pratiquer la plongée.  
Écrire une requête SQL permettant d'obtenir le nom des villes ainsi que le nom des stations où l'on peut pratiquer la plongée.
- (b) Écrire une requête SQL permettant de déterminer le nombre total de séjours effectués en Corse durant l'année 2020.

### Exercice 3 (4 points).

Cet exercice porte sur les bases de données relationnelles et le langage SQL.

Cet exercice utilise les mots du langage SQL suivants : SELECT, FROM, WHERE, JOIN ON, UPDATE, SET, DELETE, COUNT, AND, OR.

Une salle de cinéma propose un site Web à ses abonnés afin d'effectuer des réservations de séances en ligne. Deux tarifs sont proposés : plein et réduit (-16 ans, senior +65 ans, étudiants, ...). Le site est associé à une base de données dont le modèle relationnel contient les quatre relations décrites ci-dessous :



Un attribut souligné correspond à une clé primaire et un attribut précédé du symbole # à une clé étrangère.

Voici un extrait de quelques enregistrements des relations Film, Séance et Abonné :

<u>idFilm</u>	titre	réalisateur	durée
1	Le sens de la famille	Jean-Patrick Benes	90
2	Les croods 2	Joel Crawford	95
8	Black widow	Cate Shortland	134
...			

Extrait de la relation Film, les durées sont en minutes.

<u>idSéance</u>	<u>idFilm</u>	date	heure
35	1	2021-10-11	21:00
737	8	2021-10-11	21:00
738	8	2021-10-13	16:15
...			

Extrait de la relation Séance.

<u>idAbonné</u>	nom	prénom	mail
1	Henry	Jean	jean.henry@envoi.fr
2	Jacquin	Morgane	jacquin.morgane@mail.com
13	Dupont	Charles	charles.dupont@envoi.fr
...			

Extrait de la relation Abonné.

- (a) Définir le rôle d'une clé primaire.  
 (b) Définir le rôle d'une clé étrangère.  
 (c) Déterminer, en justifiant, si un abonné peut réserver plusieurs fois une même séance.  
 (d) M. Charles Dupont réserve trois places au tarif plein et deux places au tarif réduit pour assister à la projection du film "Black widow" le 11 octobre 2021 à 21:00. À l'aide des extraits des relations donnés précédemment, recopier et compléter l'enregistrement correspondant dans la relation Réservation ci-dessous :

idAbonné	idSéance	nbPlaces_plein	nbPlaces_réduit

Relation Reservation.

- (a) Parmi les trois requêtes SQL suivantes, recopier celle qui permet d'afficher le titre et le réalisateur des films de moins de 120 minutes.

SELECT titre, réalisateur FROM Film WHERE durée < 120 ;	SELECT Film FROM titre, réalisateur WHERE durée < 120 ;	SELECT titre FROM Film WHERE durée < 120 ;
---	---	--

- (b) En SQL, la fonction COUNT() permet de compter le nombre d'enregistrements dans une table. Exprimer en langage naturel la requête SQL suivante :

```
SELECT COUNT(*)
FROM Séance
WHERE date="2021-10-22" OR date="2021-10-23" ;
```

On remarquera que les dates apparaissent sous la forme "aaaa-mm-jj".

Par exemple, le 11 octobre 2021 apparaît sous la forme "2021-10-11".

- Écrire en SQL les requêtes permettant d'effectuer les tâches suivantes :
  - Afficher le nom et le prénom de tous les abonnés.
  - Afficher le titre et la durée des films projetés le 12 octobre 2021 à 21:00.  
On remarquera que les heures apparaissent sous la forme "hh:mm".
- (a) Écrire une requête en SQL permettant de modifier la durée du film Jungle Cruise (initialement enregistré avec 90) à 127.  
  
 (b) On souhaite écrire une requête SQL permettant de supprimer la séance dont l'attribut idSéance vaut 135.  
Déterminer la contrainte d'intégrité que pourrait violer cette requête.  
  
 (c) Écrire la requête précédente en SQL.