

Exercice 1 (E1S8)

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 suivant :

SELECT, FROM, WHERE, JOIN, INSERT INTO, VALUES, ORDER BY

On rappelle qu'en SQL la clause **ORDER BY**, suivie d'un attribut, permet de classer les résultats par ordre croissant de l'attribut.

Dans un lycée, le parc informatique est constitué d'ordinateurs, d'imprimantes, de vidéoprojecteurs et de TNI (Tableau Numérique Interactif).

Tous les ordinateurs et toutes les imprimantes sont connectés au réseau de l'établissement.

Chaque salle de cours est identifiée par un numéro unique et contient :

- un ou plusieurs ordinateurs reliés au réseau de l'établissement ;
- aucun ou un seul vidéoprojecteur ;
- s'il y a un vidéoprojecteur, aucun ou un seul TNI ;
- une ou plusieurs imprimantes réseau.

Un ordinateur peut être connecté via le réseau à une ou plusieurs imprimantes (situées éventuellement dans une ou plusieurs autres salles) et à un vidéoprojecteur avec TNI ou non.

Les ordinateurs et les imprimantes possèdent un nom unique sur le réseau de l'établissement.

Les vidéoprojecteurs ne sont pas connectés au réseau, ils ne possèdent pas de nom unique. Ils sont donc identifiés par le numéro de la salle où ils sont installés.

Tous ces matériels sont gérés à l'aide d'une base de données relationnelle qui comprend 3 relations (ou tables) nommées : **Ordinateur**, **Videoprojecteur**, **Imprimante**.

On donne ci dessous le schéma relationnel de la relation **Ordinateur**, suivi d'un extrait de la relation **Ordinateur**. La clé primaire est soulignée.

Ordinateur(nom_ordi : String, salle : String, marque_ordi : String, modele_ordi : String, annee : Int, video : Boolean)

On distingue deux types d'ordinateurs :

- les ordinateurs multimédias pour les logiciels généraux avec un nom commençant par le groupe de lettres Gen.
- les ordinateurs techniques pour les logiciels qui demandent plus de ressources avec un nom commençant par le groupe de lettres Tech.

Ce groupe de lettres est suivi d'un tiret et du numéro unique de l'ordinateur pour chaque type.

Les 5 premières lignes de la relation **Ordinateur**

nom_ordi	salle	marque_ordi	modele_ordi	annee	video
Gen-24	012	HP	compaq pro 6300	2012	true
Tech-62	114	Lenovo	p300	2015	true
Gen-132	223	Dell	Inspiron Compact	2019	true
Gen-133	223	Dell	Inspiron Compact	2019	false
Gen-134	223	Dell	Inspiron Compact	2019	false

1. (a) À l'aide d'un système de gestion de base de données, on envoie au serveur la requête SQL suivante :

```
SELECT salle, marque_ordi FROM Ordinateur;
```

Quel résultat produit cette requête sur l'extrait de la relation **Ordinateur** donné ci-dessus ?

- (b) Quel résultat produit la requête suivante sur l'extrait de la relation **Ordinateur** donné ci-dessus ?

```
SELECT nom_ordi, salle FROM Ordinateur WHERE video = true;
```

2. Écrire une requête SQL donnant tous les attributs des ordinateurs correspondant aux années supérieures ou égales à 2017 ordonnées par dates croissantes.
3. (a) Pour quelle raison l'attribut **salle** ne peut-il pas être une clé primaire pour la relation **Ordinateur** ?
- (b) On donne ci-dessous un extrait de la relation **Imprimante** :

Les 5 premières lignes de la relation **Imprimante**

nom_imprimante	marque_imp	modele_imp	salle	nom_ordi
imp_BTS_NB	HP	Laserjet pro M15w	114	Tech-62
imp_BTS_Couleur	Canon	Megatank Pixma G5050	114	Tech-62
imp_salle-info1	Brother	2360DN	223	Gen-132
imp_salle-info1	Brother	2360DN	223	Gen-133
imp_salle-info1	Brother	2360DN	223	Gen-134

On prend (**nom_imprimante**, **nom_ordi**) comme clé primaire. Écrire le schéma relationnel de la relation **Imprimante** en précisant les éventuelles clés étrangères pour les autres relations.

4. On donne ci-dessous un extrait de la relation **Videoprojecteur** :

Les 4 premières lignes de la relation **Videoprojecteur**

salle	marque_video	modele_video	tni
012	Epson	xb27	true
114	Sanyo	PLV-Z3	false
223	Optoma	HD143X	false
225	Optoma	HD143X	true

- (a) Écrire une requête SQL pour ajouter à la relation **Videoprojecteur** le vidéoprojecteur nouvellement installé en salle 315 de marque NEC, modèle ME402X et non relié à un TNI.
- (b) Écrire une requête SQL permettant de récupérer les attributs **salle**, **nom_ordi**, **marque_video** des ordinateurs connectés à un vidéoprojecteur équipé d'un TNI.

Exercice 2 (E4S1)

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

SELECT, FROM, WHERE, JOIN, INSERT INTO, VALUES, COUNT, ORDER BY.

Dans un lycée imaginaire, les données relatives aux élèves de secondes sont regroupées dans un fichier nommé `seconde_lyc.csv`. Un extrait de son contenu est représenté figure 1.

num_eleve	nom	prenom	datenaissance	langue1	langue2	option	classe
133310FE	ACHIR	Mussa	01/01/2005	anglais	espagnol		2A
156929JJ	ALTMAYER	Yohan	05/05/2005	allemand	anglais	théâtre	2D
500633KH	BELEY	Thibaut	05/05/2005	anglais	espagnol		2A
911887GA	BELEY	Marie	05/05/2005	anglais	espagnol		2A
906089JJ	BELEY	Manon	10/01/2005	anglais	allemand		2E
488697GA	CAILLE	Marie	30/03/2004	italien	anglais		2D
193514FB	CHARPENTIER	Jules	26/12/2005	espagnol	anglais		2C
321188FA	CLAUDEL	Benjamin	09/09/2005	espagnol	anglais		2E
081282GF	EISEN	Carla	23/06/2004	anglais	allemand		2A
026946KB	EL AYAR	Amir	11/09/2005	anglais	arabe	cinéma	2D
108303KG	GEHIN	Arthur	26/02/2005	allemand	anglais		2D
057934BK	GROSJEAN	Alexandre	09/11/2005	anglais	espagnol		2C
571113KE	HENRY	Paul	12/03/2005	allemand	anglais		2E
488820DE	JACQUEY	Marc	13/11/2005	anglais	italien		2D
024810CE	JULIANO	Alberto	21/04/2005	anglais	espagnol		2C
249992EJ	KLEIBER	Gusti	20/02/2005	anglais	espagnol	cinéma	2E
492698AF	LACOUR	Julie	06/04/2005	italien	anglais		2D
026454FA	LARBI	Nourdine	14/07/2005	espagnol	anglais		2C
309341GD	LEFZA	Yasmina	26/11/2005	espagnol	anglais		2E
076725HD	MARTIN	Victor	13/03/2005	anglais	espagnol		2A
815183CB	NGUYEN	Ngong	16/03/2005	anglais	espagnol		2D
094002FC	PELTIER	Romane	14/06/2005	allemand	anglais		2D
321262HD	RENAULT	Zoé	06/08/2005	anglais	espagnol	latin	2E
075421AK	ROTH	Ursule	03/01/2005	anglais	allemand		2A
121001CK	SERHANI	Sabrina	01/09/2005	italien	anglais		2D
538965DJ	TUDJANE	Yourk	31/01/2005	espagnol	anglais		2D
389873GC	VIALET	Priscille	28/02/2005	espagnol	anglais		2C
980306CA	WADE	Marcelin	03/05/2005	allemand	anglais		2E
807158DH	WENGER	Alexandre	20/08/2005	allemand	anglais		2A
666702FA	YAMAN	Elamine	23/04/2005	anglais	arabe		2D

Extrait du fichier `seconde_lyc.csv`

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
num_eleve (clef primaire)
langue1
langue2
option
classe

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

Question 1

1. Dans le modèle relationnel, quel est l'intérêt de l'attribut `num_eleve`.
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`.
3. Lors de l'insertion de l'élève ALTMAYER 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.

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 ;?
```

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 ;?
```

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

Question 3 Le chef d'établissement souhaite faire évoluer la structure de sa base de données. Pour ce faire, il crée une nouvelle table eleve dont la structure est la suivante :

eleve
num_eleve (clef primaire, clef étrangère de la table seconde)
nom
prenom
datenaissance

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.

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.

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.

Exercice 3 (E1S4)

Dans cet exercice, on pourra utiliser les mots clés suivants du langage SQL :

SELECT, FROM, WHERE, JOIN, ON, INSERT INTO, VALUES, MIN, MAX, OR, AND.

Les fonctions d'agrégation MIN(propriete) et MAX(propriete) renvoient, respectivement, la plus petite et la plus grande valeur de l'attribut propriete pour les enregistrements sélectionnés.

Des acteurs ayant joué dans différentes pièces de théâtre sont recensés dans une base de données Theatre dont le schéma relationnel est donné ci-dessous :

- Piece (idPiece, titre, langue)
- Acteur (idActeur, nom, prenom, anneeNaiss)
- Role (#idPiece, #idActeur, nomRole)

Dans ce schéma, les clés primaires sont soulignées et les clés étrangères sont précédées du symbole #. L'attribut idPiece de la relation Role est une clé étrangère faisant référence à l'attribut idPiece de la relation Piece.

L'attribut idActeur de la relation Role est une clé étrangère faisant référence à l'attribut idActeur de la relation Acteur.

Tous les attributs dont le nom est préfixé par id sont des nombres entiers ainsi que l'attribut anneeNaiss. Les autres attributs sont des chaînes de caractères.

1. Expliquer pourquoi il n'est pas possible d'insérer une entrée dans la relation Role si les relations Piece et Acteur sont vides.
2. Dans la pièce « Le Tartuffe », l'acteur Micha Lescot a joué le rôle de Tartuffe.
L'identifiant de Micha Lescot est 389761 et celui de cette pièce est 46721.
Écrire une requête SQL permettant d'ajouter ce rôle dans la table (ou relation) Role.
3. Expliquer ce que fait la requête SQL suivante.

```
UPDATE Piece
  SET langue = "Anglais"
  WHERE langue = "Américain" OR langue = "Britannique";
```

4. Pour chacun des quatre items suivants, écrire une requête SQL permettant d'extraire les informations demandées.
 - a. Le nom et prénom des artistes nés après 1990.
 - b. L'année de naissance du plus jeune artiste.
 - c. Le nom des rôles joués par l'acteur Vincent Macaigne.
 - d. Le titre des pièces écrites en Russe dans lesquelles l'actrice Jeanne Balibar a joué.

Exercice 4 (E4S7)

Gestion d'un club de handball

Un club de handball souhaite regrouper efficacement toutes ses informations. Il utilise pour cela des bases de données relationnelles afin d'avoir accès aux informations classiques sur les licenciés du club ainsi que sur les matchs du championnat. Le langage SQL a été retenu.

On suppose dans l'exercice que tous les joueurs d'une équipe jouent à chaque match de l'équipe.

La structure de la base de données est composée des deux tables (ou relations) suivantes:

Table licenciés	
Attributs	Types
id_licence	INT
premier nom	VARCHAR
nom	VARCHAR
année_naissance	INT
équipe	VARCHAR

Table matchs	
Attributs	Types
id_matchs	INT
équipe	VARCHAR
adversaire	VARCHAR
lieu	VARCHAR
date	DATE

Ci-dessous un exemple de ce que l'on peut trouver dans la base de données :

Exemple **non exhaustif** d'entrées de la table **licenciés**

id_licence	premier nom	nom	année_naissance	équipe
63	Jean-Pierre	Masclef	1965	Vétérans
102	Eva	Cujon	1992	Femmes 1
125	Emile	Alinio	2000	Hommes 2
247	Ulysse	Trentain	2008	-12 ans

Exemple **non exhaustif** d'entrées de la table **matchs**

id_match	équipe	adversaire	lieu	date
746	-16 ans	PHC	Domicile	2021-06-19
780	Vétérans	PHC	Exterieur	2021-06-26
936	Hommes 3	LSC	Exterieur	2021-06-20
1032	-19 ans	LOH	Exterieur	2021-05-22
1485	Femmes 2	CHM	Domicile	2021-05-02
1512	Vétérans	ATC	Domicile	2021-04-12

1.

- L'attribut **nom** de la table **licenciés** pourrait-il servir de clé primaire ? **Justifier**.
- Citer** un autre attribut de cette table qui pourrait servir de clé primaire.

- 2.
- 2.a. **Expliquer** ce que renvoie la requête SQL suivante :
- SELECT prenom,nom FROM licencies WHERE equipe ="-12ans"**
- 2.b. **Que renvoie** la requête précédente si **prenom,nom** est remplacé par une étoile (*) ?
- 2.c. **Écrire** la requête qui permet l'affichage des dates de tous les matchs joués à domicile de l'équipe *Vétérans*.
4. **Écrire** la requête qui permet d'inscrire dans la table **licencies**, *Jean Lavenu* né en 2001 de l'équipe *Hommes 2* et qui aura comme numéro de licence 287 dans ce club.
5. On souhaite mettre à jour les données de la table **licencies** du joueur *Joseph Cuviller*, déjà inscrit. Il était en équipe *Hommes 2* et il est maintenant en équipe *Vétérans*. Afin de modifier la table dans ce sens, **proposer** la requête adéquate.
6. Pour obtenir le nom de tous les licenciés qui jouent contre le LSC le 19 juin 2021, **recopier et compléter** la requête suivante :

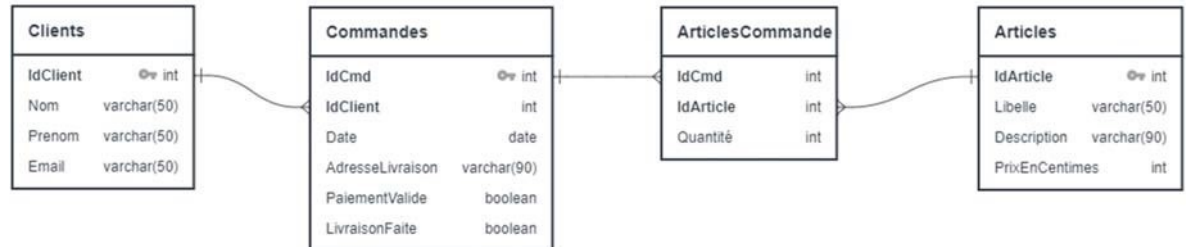
SELECT nom FROM licencies
JOIN Matches ON licencies.equipe = matches.equipe WHERE
..... ;

Exercice 5 (E2S9)

Cet exercice traite principalement du thème « traitement de données en tables et bases de données ». Nous nous interrogerons dans cet exercice sur la modélisation et l'utilisation des données nécessaires aux fonctionnements de sites de vente en ligne.

Partie A : Modèle relationnel

Voici le modèle relationnel qui sera utilisé dans cet exercice :



1. Donner les clés primaires des relations Clients et Articles.
2. Les commandes ci-dessous ont été utilisées pour créer ces deux relations. Donner le domaine (c'est-à-dire le type) des attributs Email et Quantité.

```

CREATE TABLE Clients (
  IdClient INT PRIMARY KEY,
  Nom VARCHAR(50),
  Prenom VARCHAR(50),
  Email VARCHAR(50)
);

CREATE TABLE ArticlesCommande (
  IdCmd INT,
  IdArticle INT,
  Quantite INT,
  PRIMARY KEY (IdCmd, IdArticle)
  FOREIGN KEY (IdArticle) REFERENCES Articles(IdArticle) );

```

3. En vous inspirant des commandes ci-dessus, recopier et compléter la commande suivante qui permet de créer la relation Commandes en précisant sa clé étrangère.

```

CREATE TABLE Commandes (
  IdCmd INT PRIMARY KEY,
  IdClient INT,
  Date DATE,
  AdresseLivraison VARCHAR(90),
  PaiementValide BOOLEAN, LivraisonFaite BOOLEAN,
  FOREIGN KEY(.....) REFERENCES .....
  (.....)
);

```


Partie B : Site web

La plateforme de vente en ligne possède un site web pour ses clients qui passent des commandes en remplissant un formulaire.

1. Expliquer pourquoi, lorsqu'un formulaire contient une quantité importante de données, il est préférable d'utiliser la méthode POST plutôt que GET.
2. Pour la validation du paiement, est-il préférable d'utiliser le protocole HTTP ou HTTPS ? Pourquoi ?
3. Expliquer l'intérêt de vérifier, avant la validation du formulaire, le format des informations saisies (par exemple qu'il n'y a pas de chiffre dans le nom ou qu'il y a bien un @ dans l'adresse de courriel).

Partie C : Requêtes SQL

7. Écrire une requête SQL permettant de récupérer l'identifiant et le libellé de tous les articles coûtant moins de 15 euros.
8. Expliquer ce que fait la requête SQL suivante.

```
SELECT u.IdClient, u.Email, v.IdCmd, v.AdresseLivraison  
FROM Clients as u JOIN Commandes as v  
ON u.IdClient = v.IdClient  
WHERE v.PaiementValide = False;
```

9. Écrire une requête SQL permettant de récupérer le libellé des articles de la commande 1345.
10. On suppose que l'attribut IdArticle de la table Articles est auto-incrémenté et ne doit donc pas être précisé lors de l'ajout d'un nouvel article. Écrire une requête SQL permettant d'ajouter dans la base ce nouvel article.



« Cet imperméable se replie en forme de pochette. »

Prix : 9,99 euros

Partie D : Adaptation du modèle relationnel

Le propriétaire du site souhaite une adaptation du modèle relationnel afin de :

- comptabiliser le stock pour chaque article
- pouvoir mémoriser, pour chaque client, une adresse de livraison par défaut de ses commandes.
- Préciser, pour chaque relation que vous jugez nécessaire de modifier, les attributs ajoutés ainsi que leurs domaines.

À l'arrivée d'une nouvelle commande d'un client, l'algorithme de mise à jour de la base de données ci-dessous est exécuté.

Indiquer l'erreur présente dans cet algorithme.

Algorithme nouvelle_commande (commande)

Début

Pour chaque article de la commande **Faire**

Si Quantité Stock **Alors**

 Stock \leftarrow Stock – 1

Sinon

 annuler l'achat de cet article

FinSi

FinPour

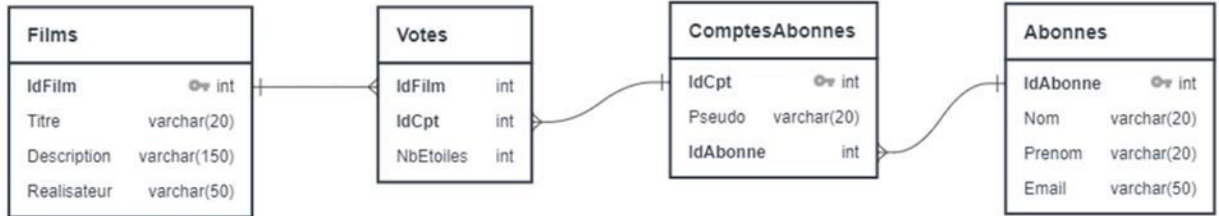
Fin

Exercice 6 (E5S9)

Les fournisseurs de VOD (vidéos à la demande) permettent à leurs abonnés d'avoir plusieurs comptes (afin que chaque membre de la famille puisse avoir son espace client rattaché à un compte unique pour la famille). Ils proposent également un service de votes et de conseils.

Partie A : Modèle relationnel

Voici le modèle relationnel qui sera utilisé dans cet exercice :



1. Donner les clés primaires des relations Films et Abonnes.
2. Le code SQL suivant a été utilisé pour créer la relation Films. Donner le domaine (c'est-à-dire le type) des attributs IdFilm et Description.

```
CREATE TABLE Films (  
  IdFilm INT AUTO_INCREMENT PRIMARY KEY,  
  Titre VARCHAR(20),  
  Description VARCHAR(150),  
  Realisateur VARCHAR(50)  
);
```

3. Préciser la clé primaire et la clé étrangère de la relation ComptesAbonnes.

L'entreprise gérant le service de VOD demande une adaptation du modèle relationnel.

4. Elle souhaite pouvoir stocker les acteurs principaux de chaque film. Préciser les modifications à apporter.
5. Elle souhaite pouvoir associer une tranche d'âge (-12 ans, 13-15 ans, 16-18 ans, +18 ans) à chacun des comptes d'un abonné et à attribuer, à chaque film, un âge minimum à avoir avant de le visionner. Préciser les modifications à apporter.

Partie B : Requêtes SQL

1. Écrire une requête SQL permettant de récupérer l'identifiant et le pseudo de tous les comptes rattachés à l'abonné 237
2. Expliquer ce que fait la requête SQL suivante.

```
SELECT AVG(NbEtoiles)
FROM Votes
WHERE IdFilm = 1542;
```

3. Expliquer ce que fait la requête SQL suivante.

```
SELECT u.IdFilm, u.Titre, v.NbEtoiles
FROM Films as u JOIN Votes as v
ON u.IdFilm = v.IdFilm
WHERE v.IdCpt = 508
ORDER BY v.NbEtoiles DESC;
```

4. Écrire une requête SQL permettant de modifier le pseudo du compte 508 pour lui attribuer la valeur « Champion ».

Partie C : Conseils de films

Dans cette partie, nous utiliserons un module Python dont l'interface est fournie cidessous.

Module ConsultationBaseVOD.py	
podiumCompte(IdCpt)	Prend en paramètre un identifiant de compte et renvoie la liste des films les mieux notés par l'utilisateur de ce compte (<i>au maximum 10 films</i>) par ordre décroissant de nombre d'étoiles.
spectateurs(listeFilms)	Prend en paramètre une liste de films et renvoie une liste contenant les identifiants des comptes sur lesquels tous ces films ont été visionnés.
nbEtoiles(IdFilm, IdCpt)	Prend en paramètre un identifiant de film et un identifiant de compte. Renvoie le nombre d'étoiles attribuées à ce film par cet utilisateur.

1. Expliquer ce que fait la fonction ci-dessous.

```
def distance(IdCpt1, IdCpt2, listeFilms):
    assert (type(listeFilms) is list) and (len(listeFilms) != 0)
    somme_ecarts = 0
    for film in listeFilms:
        somme_ecarts += abs(nbEtoiles(film, IdCpt1) - \
                           nbEtoiles(film, IdCpt2))
    return somme_ecarts / len(listeFilms)
```

2. Traduire en Python la fonction conseilsFilms suivante qui permet, à partir d'un identifiant de compte, de conseiller des films proches des goûts de l'utilisateur de ce compte.

Fonction conseilsFilms (IdCpt)

Créer une liste vide nommée « conseils »

Récupérer la liste des films les mieux notés par l'utilisateur du compte passé en paramètre

Récupérer la liste des spectateurs ayant visionné ces films

Pour chacun de ces spectateurs

Si la distance avec l'utilisateur du compte passé en paramètre est inférieure à 10

Ajouter à la liste « conseils » les films préférés de ce spectateur (maximum 3)

Renvoyer la liste « conseils »

Remarque : Les répétitions sont autorisées dans la liste « conseils ».