

EXAMEN SEMESTRE 2

Matière : INFORMATIQUE

Classes : MP2, PC2 et PT2

Durée : 2 h

Exercice 1 (4.5 pts)

Soient les tables livres et adhérents suivantes :

id	titre	auteur	editeur	nb_pages	annee	emprunteur	date_retour
1	Notre-Dame de Paris	Victor Hugo	Gosselin	636	1831	1	05/06/2014
2	Les misérables	Victor Hugo	Lacroix	1662	1862		
3	Voyage au centre de la Terre	Jules Verne	Hetzel	372	1864	1	10/06/2014
4	Le tour du monde en 80 jours	Jules Verne	Hetzel	223	1872	2	10/07/2014

Table 1. Table des livres.

id	nom	prenom	email
1	DUPONT	Jean	jean.dupont@gmail.com
2	MARTIN	Paul	paul.martin@gmail.com

Table 2. Table des adhérents.

1) Traduire les expressions relationnelles suivantes en requête SQL :

- a) $\sigma_{annee < 1850}(\text{livres})$
- b) $\Pi_{titre, auteur}(\text{livres})$,
- c) $\sigma_{annee > 1850}(\text{livres}) \cap \sigma_{nb_pages > 1000}(\text{livres})$
- d) $\sigma_{editeur = "Lacroix"}(\text{livres}) \cup \sigma_{editeur = "Hetzel"}(\text{livres})$
- e) $\Pi_{livres.titre, adherents.nom, adherents.prenom}(\text{livres} \bowtie_{emprunteur=id} \text{adherents})$

2) Répondre par vrai ou faux aux requêtes suivantes :

- a) La requête SQL : `SELECT titre FROM livres ORDER BY titre ASC` renvoie la liste : ["Notre-Dame de Paris", "Les misérables", "Voyage au centre de la Terre", "Le tour du monde en 80 jours"].
- b) La requête SQL : `SELECT count(nb_pages), sum(nb_pages) FROM livres` renvoie les entiers 3 et 1613.
- c) La requête SQL: `SELECT * FROM livres WHERE (emprunteur = "" AND titre = "Les misérables")` permet de savoir si le livre Les misérables est disponible.
- d) La requête SQL: `SELECT DISTINCT titre FROM (SELECT * FROM livres UNION SELECT * FROM livres) WHERE auteur = "Victor Hugo"` renvoie la liste ["Notre-Dame de Paris", "Les misérables"].

Exercice 2 (4.5 pts)

Question 1 :

Une image négative est une image dont les couleurs ont été inversées par rapport à l'originale; par exemple le rouge devient cyan, le vert devient magenta, le bleu devient jaune et inversement. Les régions sombres deviennent claires, le noir devient blanc. Pour cela il suffit d'inverser les niveaux de chacune des couleurs primaires : un pixel initialement à [120, 10, 250] deviendra [135, 245, 5]. C'est-à-dire que le ton d'un pixel de la nouvelle image est $255 - v$, où v est le ton du pixel correspondant de l'image d'origine.

Écrire une fonction ***negatif(img)*** retournant l'image négative de l'image passée en paramètre.

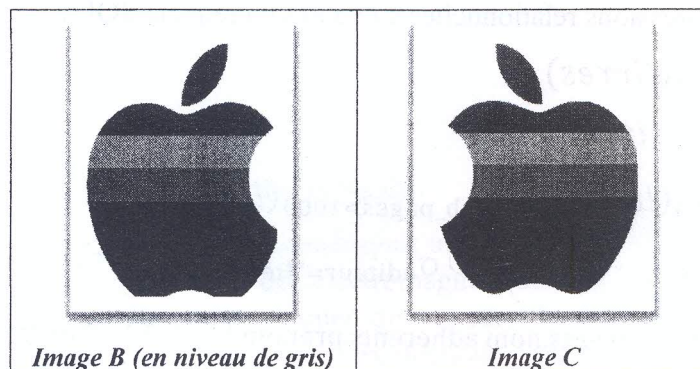
Question 2 :

Dans une image en niveaux de gris, les trois composantes R, V, B de chaque pixel ont la même valeur. Celle-ci vaut 0 pour un pixel noir, 255 pour un pixel blanc.

L'œil est plus sensible à certaines couleurs qu'à d'autres. Le vert (pur), par exemple, paraît plus clair que le bleu (pur). Pour tenir compte de cette sensibilité dans la transformation d'une image couleur en une image en niveaux de gris, on ne prend généralement pas la moyenne arithmétique des intensités de couleurs fondamentales, mais une moyenne pondérée. La formule standard donnant le niveau de gris en fonction des trois composantes est :

$$\text{gris} = \text{partie entière de } (0.299 * \text{rouge} + 0.587 * \text{vert} + 0.114 * \text{bleu})$$

Écrire une fonction ***niveau_gris(im)*** renvoyant une nouvelle image, en niveau de gris. Les trois niveaux R, V, B d'un pixel sont égaux au niveau de gris donné par la formule ci-dessus. On aura besoin de la fonction `round()` pour calculer la partie entière.



Question 3 :

Dans cette question on traitera une image en niveau de gris, décrite par une matrice dont chaque pixel est représenté seulement par un entier décrivant le ton de gris.

Écrire une fonction ***flipH(img)*** qui renvoie la transformée de l'image `img` par la symétrie d'axe vertical. Par exemple, l'image C de la figure 2 résulte de l'application de *flipH* à l'image B.

Rappel :

commandes	Rôle
<code>import numpy as np</code>	Importation du module numpy
<code>from PIL import Image</code>	Importation du module Image de Pillow
<code>appl=Image.open("apple.png")</code>	Ouverture de l'image apple.png
<code>T= np.array(appl)</code>	Transformation de l'image "appl" en matrice
<code>Image.fromarray(T)</code>	Transformation de la matrice T en image

Exercice 3 (6 pts)

On dispose d'une base de données concernant les astres et planètes de notre galaxie qui contient cinq tables. Son schéma de base de données relationnelle est le suivant :

- astre(**nomAstre**, Diametre) :
- planete(nomAstre, **nomPlanete**, diametre, masse, tempsRevolution)
- astrophysicien(nom, prenom, pays)
- asteroide(**nomAsteroide**, diametre, nom, prenom)
- collision(nomAstre, nomPlanete, nomAsteroide, date)

Un astre (par exemple le soleil) est identifié par un nom et possède un diamètre.

Une planète (par exemple la terre) est identifiée par un nom et le nom de l'astre autour duquel elle tourne. Elle possède un diamètre, une masse et un temps de révolution autour de son astre (la terre met 365 jours environ pour faire le tour du soleil).

Un astrophysicien est identifié par un nom et un prénom

Un astéroïde est identifié par un nom et caractérisé par un diamètre et le nom et le prénom de l'astrophysicien qui l'a découvert. Nom et prénom constituent une clé étrangère sur la relation astrophysicien.

Enfin, une collision est identifiée par le nom d'une planète, le nom de l'astre autour duquel la planète tourne (nomAstre) et le nom d'un astéroïde (nomAsteroide). Elle est caractérisée par la date de collision.

asteroide

nomAsteroide	diametre	nom	prenom
Ceres	0.076354	Piazzi	Giuseppe
Pluton	0.1807	Tombaugh	Clyde
Fictive120	0.0081	Tournesol	Tryphon
Fictive3456	0.004	Tournesol	Tryphon
(4769) Castalia	0.00003	Helin	Eleano

astre

nomAstre	diametre
Soleil	1.0
Alpha Tauri	44.2
Alpha Canis Majoris	1.711
Fictive	3.0

collision

nomAstre	nomPlanete	nomasteroide	date
Fictive	FictiveA	Fictive120	30/05/2014
Fictive	FictiveB	Fictive3456	07/09/2021
Soleil	Terre	(4769) Castalia	00/00/0000

planete

nomAstre	nomPlanete	diametre	masse	tempsRevolution
Soleil	Mercure	0.384	0.055	58
Soleil	Terre	12456.3	1	365.257
Soleil	Uranus	4.007	14.536	30799.1
Soleil	Mars	0.533	0.107	669
Soleil	Venus	0.949	0.815	224.701
Soleil	Jupiter	11.209	317.8	4335.35
Soleil	Saturne	9.4492	95.152	10757.7
Fictive	FictiveA	1.8	7	678
Fictive	FictiveB	2	6	456

astrophysicien

Nom	prenom	pays
Piazzi	Giuseppe	Italie
Herschel	William	Britannique
Tombaugh	Clyde	USA
Helin	Eleanor	USA
Tournesol	Tryphon	Belgique

Exprimer chacune des questions 1) et 2) qui suivent en algèbre relationnelle puis en SQL :

- 1) Quels sont les astres dont le diamètre dépasse 10^6 km sachant que les diamètres proposés sont donnés par rapport au diamètre du soleil qui vaut $1.3927 \cdot 10^6$ km?
- 2) Quelles sont les Planètes qui tournent autour du soleil et dont le temps de révolution dépasse les 500 jours ?

Exprimer les requêtes suivantes en SQL :

- 3) Quels sont les astrophysiciens (nom et prenom) qui ont découverts un astéroïde au moins ?
- 4) Quels sont les astrophysiciens (nom, prenom et pays) qui ont découverts un astéroïde au moins ?
- 5) Quels sont les astrophysiciens qui n'ont découverts aucun astéroïde ?
- 6) Quels sont les astres (nomAstre) dont toutes les planètes ne sont pas concernées par des collisions.
- 7) Pour chaque astrophysicien, donner son prénom, son nom et le nombre d'astéroïdes qu'il a découvert.
- 8) Quels sont les astres possédant le maximum de planètes ?

astre	نجم
planète	كوكب
astrophysicien	عالم الفيزياء الفلكية
astéroïde	نيزك
collision	اصطدام

Exercice 4 (6 pts)

Les checksums et la correction des erreurs

Les données échangées entre systèmes numériques au travers des différents supports physiques (câbles électriques, fibres optiques, ondes électromagnétiques, etc.) peuvent être perturbées par toutes sortes de facteurs pouvant provoquer des erreurs dans les données : autres signaux électromagnétiques, masses métalliques, imperfections du matériel électronique, etc. En pratique, il est donc indispensable de pouvoir détecter ces erreurs et, dans la mesure du possible, de les corriger sans que cela ne nécessite une nouvelle transmission. L'objet de cet exercice est de mettre en place quelques fonctions dans ce but.

Bit de parité

Une technique, simple et très répandue, pour s'assurer qu'une donnée transmise sous la forme d'un mot binaire sera lue correctement par son récepteur est de lui adjoindre un bit de parité, égal par définition à :

- 0, si la donnée contient un nombre pair de 1 (et donc si ses bits sont de somme paire) ;
- 1, si la donnée contient un nombre impair de 1 (et donc si ses bits sont de somme impaire).

Après réception de la donnée, le récepteur recalcule le bit de parité et le compare à celui que l'émetteur lui a adressé. Si la donnée n'a pas été altérée lors de la transmission, alors les deux bits de parité sont forcément identiques.

- 1) Donner les valeurs binaires et les bits de parité associés aux entiers 5, 16 et 37 en complétant le suivant :

Entier	valeur binaire	Bit de parité
5		
16		
37		

- 2) Écrire une fonction `parite(bits)` prenant pour argument une liste de bits constituée d'entiers valant 0 ou 1 et retournant l'entier 0 ou 1 correspondant à son bit de parité.

Exemple : si `bits = [1,0,0,1,0,1]` alors `print(parite(bits))` affiche 1

Code de Hamming

Le code de Hamming est un exemple d'utilisation des bits de parité pour détecter et corriger des erreurs. Nous nous intéressons ici au code dit (7,4), ainsi appelé parce qu'il consiste à joindre trois bits de parité à quatre bits de données, ce qui donne un message d'une longueur totale de sept bits. Ces trois bits de parité sont définis ainsi : si la donnée s'écrit (d_1, d_2, d_3, d_4) avec $d_1 = 0$ ou 1, alors :

- p_1 est le bit de parité du triplet (d_1, d_2, d_4) ;
- p_2 est le bit de parité du triplet (d_1, d_3, d_4) ;
- p_3 est le bit de parité du triplet (d_2, d_3, d_4) .

Le message encodé, que l'on transmet, s'écrit alors comme suit : $(p_1, p_2, d_1, p_3, d_2, d_3, d_4)$.

- 3) Ecrire une fonction `encode_hamming(d)` prenant pour argument une liste d de quatre bits (représentés par des entiers valant 0 ou 1) et retournant une liste de bits contenant le message encodé. On pourra appeler la fonction `parite(bits)` précédemment définie.

La technique proposée par Hamming, pour le contrôle après réception d'un message ainsi encodé, est de calculer les trois bits de contrôle suivants, notés (c_1, c_2, c_3) , à partir du message complet (données et bits supplémentaires), noté (m_1, \dots, m_7) :

- c_1 est le bit de parité de l'ensemble (m_4, m_5, m_6, m_7) ;
- c_2 est le bit de parité de l'ensemble (m_2, m_3, m_6, m_7) ;
- c_3 est le bit de parité de l'ensemble (m_1, m_3, m_5, m_7) .

On montre que, si le message a bien été encodé selon les règles précédentes et n'a pas été altéré, alors les trois bits de contrôle doivent être à 0. Si ce n'est pas le cas, alors il y a eu une erreur; l'intérêt de la technique de Hamming est que, dans le cas particulier où l'erreur est unique, le mot de contrôle donne la représentation binaire de la position de cette erreur en numérotant à partir de 1. Par exemple, si $(c_1, c_2, c_3) = (0,1,1)$, alors l'erreur porte sur le troisième bit du message. Il suffit ainsi d'inverser ce bit (le mettre à 1 s'il est à 0, et inversement) pour corriger l'erreur.

La donnée décodée est alors constituée des quatre bits (d_4, d_2, d_3, d_4) qui se trouvent respectivement en positions 3, 5, 6 et 7 (toujours en numérotant à partir de 1), conformément à la description de l'encodage donnée ci-dessus.

- 4) Ecrire une fonction `decode_hamming(m)` prenant pour argument une liste m de sept bits et retournant une liste de quatre bits contenant la donnée décodée. En cas d'erreur, on affichera à l'écran un avertissement indiquant la position du bit affecté et on effectuera la correction. On supposera dans cette question que, s'il y a une erreur, alors elle est unique.