



Concours Biologie et Géologie Epreuve d'Informatique.

Date : Mercredi 22 Juillet 2020

Heure : 8 H

Durée : 2 H

Nombre de pages : 6

Barème : PROBLEME 1 : 11 points

PROBLEME 2 : 9 points

DOCUMENTS NON AUTORISES
L'USAGE DES CALCULATRICES EST INTERDIT
IL FAUT RESPECTER IMPERATIVEMENT LES NOTATIONS DE L'ENONCE
VOUS POUVEZ EVENTUELLEMENT UTILISER LES FONCTIONS PYTHON
DECRIRES A L'ANNEXE (PAGE 6)

Un concours national d'entrée aux cycles de formation d'ingénieurs réunit chaque année des milliers de candidats qui passent un ensemble d'épreuves spécifiques aux sections MP (Mathématiques et Physique), PC (Physique et Chimie), T (Technologie) et BG (Biologie et Géologie). On propose de concevoir en partie une application de gestion de certains résultats des épreuves d'un concours.

Dans la suite, les deux problèmes sont indépendants mais se rapportent au même contexte.

PROBLEME 1

On considère une partie d'une base de données schématisée par les relations décrites ci-dessous. Bien que se rapprochant de la réalité, il est à noter que le schéma a été élaboré pour les besoins de l'énoncé.

▪ **Etablissement** (*idEtab*, nom, adresse, gouv)

La table **Etablissement** décrit tous les établissements universitaires assurant un cycle préparatoire ainsi qu'un établissement fictif (service concours) pour les candidats libres.

- *idEtab* : identifiant de l'établissement contenant le nom abrégé (chaîne de caractères), *clé primaire* (l'établissement fictif a comme identifiant la valeur 'Libre') ;
- *nom* : nom complet de l'établissement (chaîne de caractères) ;
- *adresse* : adresse de l'établissement (chaîne de caractères) ;
- *gouv* : nom du gouvernorat où l'établissement est situé (chaîne de caractères).

▪ **Candidat** (*idC*, nom, date_naissance, sexe, nationalite, adresse, section, *idEtab*)

La table **Candidat** décrit tous les candidats inscrits au concours.

- *idC* : identifiant du candidat (entier), *clé primaire* ;
- *nom* : nom et prénom du candidat (chaîne de caractères) ;
- *date_naissance* : date de naissance du candidat (chaîne de caractères en format 'aaaa/mm/jj' avec 'aaaa' : année, 'mm' : mois et 'jj' : jour) ;
- *sexe* : sexe du candidat (un caractère, 'F' pour féminin et 'M' pour masculin) ;
- *nationalite* : nationalité du candidat (chaîne de caractères) ;
- *adresse* : adresse du domicile du candidat (chaîne de caractères) ;
- *section* : section à laquelle le candidat appartient (chaîne de caractères, 'MP', 'PC', 'T', 'BG') ;

- idEtab : identifiant de l'établissement d'origine du candidat (chaîne de caractères), *clé étrangère* qui fait référence à la table **Etablissement**.

Tous les candidats non inscrits à un cycle préparatoire sont considérés comme candidats libres et affectés à l'établissement fictif dont l'attribut idEtab est 'Libre'.

▪ **Epreuve** (idEpr, nomEpr, section, dateEpr, heure, duree, coeff)

La table **Epreuve** décrit toutes les épreuves du concours.

- idEpr : identifiant de l'épreuve (entier), *clé primaire* ;
- nomEpr : nom de l'épreuve (chaîne de caractère) ;
- section : section à laquelle l'épreuve est associée (chaîne de caractères, 'MP', 'PC', 'T', 'BG') ;
- dateEpr : date de l'épreuve (chaîne de caractères en format 'aaaa/mm/jj' avec 'aaaa' : année, 'mm' : mois et 'jj' : jour) ;
- heure : horaire de passage de l'épreuve (chaîne de caractères en format 'hh : mm' avec 'hh' : heures et 'mm' : minutes) ;
- duree : durée de l'épreuve en nombre d'heures (entier) ;
- coeff : coefficient de l'épreuve (entier).

▪ **Evaluation** (idC, idEpr, note)

La table **Evaluation** décrit les notes des candidats aux épreuves passées.

- idC : identifiant du candidat (entier), *clé étrangère* qui fait référence à la table **Candidat** ;
- idEpr : identifiant de l'épreuve passée (entier), *clé étrangère* qui fait référence à la table **Epreuve** ;
- note : la note du candidat à une épreuve donnée (un réel compris entre 0 et 20).

Le couple (idC, idEpr) représente la *clé primaire* de la table Evaluation.

Partie 1 : Algèbre relationnelle

Exprimer en algèbre relationnelle les requêtes permettant de donner :

1. Les noms des candidats libres de la section 'PC'.
2. Les identifiants des candidats qui n'ont passé aucune épreuve du concours.

Partie 2 : SQL

Dans la suite, en supposant que toutes les tables ont été créées et remplies en respectant les règles d'intégrités, exprimer en SQL les requêtes suivantes permettant de :

3. Déterminer les noms des établissements assurant un cycle préparatoire pour la section 'BG'.
4. Modifier à 2 heures la durée de l'épreuve ayant le nom 'Informatique' de la section 'T'.
5. Calculer, pour chaque section, le total des coefficients des épreuves.
6. Déterminer, par ordre décroissant suivant le nom, les noms des candidats ainsi que leurs notes aux épreuves passées le 10 juin 2019.
7. Déterminer pour chaque épreuve le nombre de candidats ayant obtenu des notes supérieures à 10.
8. Déterminer les identifiants des candidats ayant obtenu des notes supérieures à 15, au moins à 3 épreuves.
9. Déterminer le nom et la date de naissance de la plus jeune candidate au concours.

Partie 3 : SQLite

Dans la suite, les fonctions demandées doivent être écrites en Python en désignant par **cur**, le curseur d'exécution de requêtes.

10. Ecrire la fonction **Score_candidat** qui prend en paramètres **cur** et l'identifiant **id** d'un candidat, puis calcule et retourne la valeur du score du candidat.

On considère que le score d'un candidat est la somme de chacune de ses notes obtenues aux épreuves, multipliées par le coefficient associé.

11. Ecrire la fonction **Maj_candidats** qui, à partir du paramètre **cur**, ajoute une colonne nommée **score** à la table **Candidat** et la met à jour en calculant le score de chaque candidat.

12. Ecrire la fonction **Admis** qui prend en paramètres **cur** et un dictionnaire **d_nbplaces** où chaque clé est le nom d'une section et la valeur est le nombre total de places disponibles associé. Cette fonction retourne un dictionnaire des admis **d_admis** où chaque clé est le nom d'une section et la valeur est la liste des identifiants des candidats admis, triée par leurs scores en ordre décroissant.

NB : Pour une section donnée, si le nombre de places disponibles dans les écoles d'ingénieurs est n , alors les candidats ayant les n meilleurs scores sont admis.

PROBLEME 2

Dans ce problème, on s'intéresse à une étude statistique des résultats partiels du concours pour un établissement donné disposant du classement de ses candidats à l'échelle nationale.

Pour cela, on définit quelques notions de statistique, telles que la médiane, le quartile, etc.

Une médiane, pour une série de valeurs triées, est la valeur x qui coupe la série en deux sous-séries de tailles égales telles que la première contient les valeurs inférieures à x et la deuxième contient celles supérieures à x . Ainsi, pour une série de taille paire, la médiane est la moyenne des deux valeurs au milieu et pour une série de taille impaire, la médiane est la valeur du milieu. La médiane est une valeur qui peut ne pas faire partie de la série étudiée.

Exemple :

- Pour $S1 = 98, 103, 105, 106, 113, 114, 117, 138$ de taille 8, la valeur médiane est $(106+113)/2=109.5$
- Pour $S2 = 83, 98, 103, 105, 106, 113, 114$ de taille 7, la médiane est 105

Pour une série de valeurs triées, sa distribution est dite *symétrique*, *asymétrique à gauche* ou *asymétrique à droite* selon la position de la valeur moyenne par rapport à la valeur médiane. En effet, si on désigne par Mo et Me respectivement la valeur moyenne et la valeur médiane, on a une distribution :

- symétrique si $Mo=Me$,
- asymétrique à gauche si $Mo>Me$,
- asymétrique à droite si $Mo<Me$.

Exemple :

- Pour $S1$, la moyenne étant 111.75 et la médiane 109.5, la distribution est asymétrique à gauche.
- Pour $S2$, la moyenne étant 103.14 et la médiane 105, la distribution est asymétrique à droite.

Pour avoir une meilleure idée sur la répartition des valeurs de la série triée, on peut la diviser en 4 sous-séries de tailles égales, séparées par 3 valeurs nommées *quartiles* :

- Le premier quartile, noté $Q1$, est la valeur telle que les premières 25 % des valeurs de la série lui sont inférieures ;
- Le deuxième quartile, noté $Q2$, est la valeur telle que les premières 50 % des valeurs de la série lui sont inférieures, ce qui correspond à la valeur médiane ;
- Le troisième quartile, noté $Q3$, est la valeur telle que les premières 75 % des valeurs de la série lui sont inférieures.

Les quartiles sont des valeurs qui peuvent ne pas faire partie de la série étudiée.

Pour une série de N valeurs triées par ordre croissant :

- Q_1 est calculée à partir de r tel que $r = (N - 1) / 4$;
- Q_3 est calculée à partir de r tel que $r = 3(N - 1) / 4$.

Si r est un entier, la valeur du quartile est la valeur de la série de rang r .

Si r n'est pas entier, on définit :

- dr : la différence entre r et sa partie entière ;
- a : la valeur de la série de rang égal à la partie entière de r ;
- b : la valeur de la série de rang égal à la partie entière de $(r + 1)$.

Selon la valeur de dr , la valeur d'un quartile est égale à :

- $(3a + b) / 4$, si $dr = 0.25$;
- $(a + b) / 2$, si $dr = 0.5$;
- $(a + 3b) / 4$, si $dr = 0.75$.

Exemple :

Dans le cas de Q_1 pour S_1 : $r = 1.75$, $dr = 0.75$, $a = 103$, $b = 105$, $Q_1 = (103 + 3 * 105) / 4$.

$$S_1 = 98 \quad \underbrace{103 \quad 105}_{Q_1=104.5} \quad \underbrace{106 \quad 113}_{Q_2=109.5} \quad \underbrace{114 \quad 117}_{Q_3=114.75} \quad 138$$

L'écart interquartile noté E , représentant le critère de dispersion des valeurs de la série, est la différence entre Q_3 et Q_1 ($E = Q_3 - Q_1$). Une valeur **aberrante** est une valeur qui s'écarte fortement des autres valeurs de la série, anormalement faible ou élevée. En effet, les valeurs aberrantes sont les valeurs de la série qui n'appartiennent pas à l'intervalle $[Q_1 - k E, Q_3 + k E]$ avec k un réel positif.

Exemple : Pour $k = \frac{3}{2}$, la valeur aberrante de S_1 est 138 et celle de S_2 est 83.

Travail demandé

On suppose que le classement, à l'échelle nationale des candidats, est enregistré dans un dictionnaire nommé **d_Rangs** où chaque :

- clé est le nom de la section (chaîne de caractères ayant pour valeur 'MP', 'PC', 'T', 'BG') ;
 - valeur est la liste des tuples contenant chacun l'identifiant (entier) et le rang (entier) d'un candidat.
1. Ecrire une fonction **IDRang** qui prend en paramètres le dictionnaire **d_Rangs**, le nom d'une section **s** et un entier **rg**, puis retourne l'identifiant du candidat de la section **s** ayant le rang **rg**.
 2. Ecrire une fonction **ListRangSection** qui prend en paramètres le dictionnaire **d_Rangs** et le nom d'une section **s**, puis retourne dans une liste **L**, les rangs des candidats de **s** triés par ordre croissant.
 3. Ecrire la fonction **MoyenneRang** qui prend en paramètre une liste d'entiers **L** et retourne la valeur moyenne de ses éléments.
 4. Ecrire la fonction **MedianeRang** qui prend en paramètre une liste d'entiers **L** triée et retourne la valeur médiane associée.
 5. Ecrire la fonction **DistributionRang** qui prend en paramètre une liste d'entiers **L** triée et retourne la chaîne représentant la distribution des rangs des différents candidats ('symétrique', 'asymétrique à gauche', 'asymétrique à droite').

6. Ecrire la fonction **LQuartiles** qui prend en paramètres une liste d'entiers **L** triée et un entier **q** ayant comme valeur par défaut 1, puis calcule et retourne la valeur du quartile selon la valeur de **q**. La fonction retourne la valeur Q1 si **q=1**, la valeur médiane si **q=2** et la valeur Q3 si **q=3**.
7. Ecrire la fonction **ListAberrantes** qui prend en paramètres une liste d'entiers **L** triée et un réel **k** ayant comme valeur par défaut 3/2, puis détermine et retourne une liste contenant les valeurs aberrantes de **L**.
8. Ecrire la fonction **Rapport** qui prend en paramètre **d_Rangs** et permet d'écrire dans un fichier en format texte nommé " rapport.txt " un rapport détaillé des résultats pour toutes les sections. Chaque ligne du fichier relative aux résultats d'une section, contient les informations séparées par le caractère '*' selon le format suivant :

section*nombre de candidats*identifiant du meilleur candidat*valeur moyenne des rangs*valeur médiane des rangs*le type de distribution*le nombre de valeurs aberrantes

ANNEXE – Quelques fonctions Python

Sans aucune obligation, les fonctions suivantes pourraient vous être utiles.

Module math

- **floor**(r) retourne la partie entière d'un réel r.
- **ceil**(r) retourne la partie entière d'un réel r +1.

Opérations sur les itérables (str, tuple, list, dict, etc.)

- **len**(it) retourne le nombre d'éléments de l'itérable it.
- **range**(d,f,p) retourne la séquence des valeurs entières successives comprises entre d et f, f exclu, par pas=p.
- **sum**(it) retourne la somme des éléments de l'itérable it.
- **x in it** vérifie si x appartient à it.
- **sorted**(it) retourne une liste contenant les éléments de it dans l'ordre croissant.
- **lst.sort()** trie la liste lst dans l'ordre croissant.
- **lst.count**(val) retourne le nombre d'occurrences de val dans la liste lst.
- **lst.append**(val) ajoute val à la fin de la liste lst.
- **ch.format**(paramètres) retourne une chaîne de caractères obtenue en substituant dans l'ordre chaque '{ }' dans ch par un objet de paramètres.
- **d.values**() retourne un itérable formé par les valeurs du dictionnaire d.
- **d.items**() retourne un itérable de couples (k,v) ou k est une clé du dictionnaire d et v est la valeur associée.

Opérations sur les fichiers

- **f=open** (nomF, m) permet d'ouvrir le fichier nomF en mode m où m='r', 'w' ou 'a'.
- **f.close**() permet de fermer un fichier.
- **f.read**() permet de lire et retourner le contenu d'un fichier dans une chaîne de caractères.
- **f.readline**() permet de lire et retourner le contenu de la ligne courante d'un fichier dans une chaîne de caractères.
- **f.write**(...) et **f.writelines** (...) permettent d'écrire dans le fichier.