

PROTOTYPE 2023

Épreuve pratique d'informatique

Section : Économie et Gestion

Durée : 1 heure

Coefficient : 0.5

Le sujet comporte 4 pages numérotées de 1/4 à 4/4

Important :

- Toutes les ressources à utiliser se trouvent dans le répertoire "**Ressources**" situé sur la racine du disque C.
- Il est demandé au candidat :
 - de créer, dans le répertoire **Bac2023** situé sur la racine du disque C, un dossier de travail portant son numéro d'inscription (6 chiffres) et dans lequel il doit enregistrer, au fur et à mesure, tous les fichiers solutions de ce sujet,
 - de copier, dans son dossier de travail, le fichier "**EG_Séance1.rar**" situé dans "**C:\Ressources**", puis d'extraire son contenu, en utilisant le mot de passe "**123456**", dans ce même dossier de travail,
 - de vérifier à la fin de l'épreuve que tous les fichiers créés sont enregistrés dans son dossier de travail.

Afin de mieux gérer les différents services dans un restaurant, on se propose d'utiliser la base de données simplifiée intitulée "**Gestion_Restaurant**" décrite par le schéma suivant :

TABLE (NumTable, NbrePlaces)

SERVEUR (IdServeur, NomPrenom, Genre, Tel)

PLAT (CodePlat, NomPlat, PrixPlat)

COMMANDE (NumTable#, IdServeur#, CodePlat#, DateCom, HeureCom, NbrePlats)

Soit la description des colonnes des tables de cette base de données :

Nom	Description	Type	Taille	Format	Contrainte
NumTable	Le numéro de la table	Numérique	Entier		Null interdit
NbrePlaces	Le nombre de places d'une table		Entier		Null interdit
IdServeur	L'identifiant d'un serveur	Texte (Texte court)	3		Null interdit
NomPrenom	Le nom d'un serveur		50		Null interdit
Genre	Le genre d'un serveur		1		
Tel	Le numéro du téléphone d'un serveur		8		
CodePlat	Le code d'un plat		5		Null interdit
NomPlat	Le nom d'un plat		50		Null interdit
PrixPlat	Le prix d'un plat		Numérique	Réel	3 chiffres après la virgule
DateCom	La date d'une commande	Date/Heure		Date abrégé	Null interdit
HeureCom	L'heure d'une commande			Heure abrégé	Null interdit
NbrePlats	Le nombre de plats commandés	Numérique	Entier		> 0

A) À l'aide du logiciel de gestion de base de données disponible :

- 1) Ouvrir la base de données incomplète intitulée "**Gestion_Restaurant**" située dans votre dossier de travail.
- 2) Ajouter à la base de données la table "**SERVEUR**" décrite précédemment tout en respectant les types et les propriétés cités dans la description ci-dessus.

3) Insérer les données ci-dessous dans la table "SERVEUR".

Serveur			
IdServeur	NomPrenom	Genre	Tel
S01	SALHI Jamel	H	33222333
S02	JEMNI Nadia	F	66000333
S03	SELMI Kamilia	F	66000332
S04	SOUSSI Wafa	F	33111999

4) Ajouter les contraintes mentionnées dans la description des colonnes des tables ci-dessus sur le champs "NbrePlats" de la table "COMMANDE".

5) Créer les relations entre les tables de cette base de données.

6) Créer les requêtes suivantes :

R1 : Afficher les noms et les prix des plats.

R2 : Afficher le nom et le nombre total de chaque plat servi le "03/01/2022" pour la table numéro 1.

R3 : Afficher la liste des commandes (**NumTable**, **NomPrenom**, **NomPlat**, **PrixPlat**, **NbrePlats**) servies pendant l'année 2022, triée par ordre chronologique des dates de commandes.

R4 : Augmenter de 2 DT le prix du plat pour un code donné.

7) Exporter, dans votre dossier de travail et sous le nom "req3.csv", le résultat de la requête **R3** au format csv (point-virgule en séparateur de champ, point en séparateur décimal, page de codes : Unicode (UTF-8) et inclure les noms des champs sur la première ligne).

B) À l'aide de l'environnement de développement disponible et en exploitant la bibliothèque Pandas :

1) Créer un fichier et l'enregistrer sous le nom "commande" dans votre dossier de travail.

2) Choisir les méthodes et les propriétés adéquates, à partir de l'annexe (pages 3 et 4), pour écrire dans le fichier créé les scripts permettant :

a) d'importer, à partir de votre dossier de travail, le contenu du fichier "commandes.csv" et de le stocker dans un DataFrame à nommer "com",

b) d'afficher le contenu de ce DataFrame,

c) d'afficher les dimensions de ce DataFrame,

d) de calculer et d'afficher pour chaque commande le prix total à payer dans une nouvelle colonne à nommer "PrixTotal", sachant que :

$$\text{PrixTotal} = \text{PrixPlat} * \text{NbrePlats}$$

e) d'afficher le montant total des commandes,

f) de trier le DataFrame "com" dans un nouveau DataFrame à nommer "com_tri" selon l'ordre croissant des numéros des tables et l'ordre décroissant des noms des plats servis puis d'afficher le DataFrame trié.

Remarque : Ce sujet n'est qu'un exemple de prototype, d'autres variantes peuvent être proposées dans lesquelles on pourra demander :

- d'importer une base de données qui peut être :
 - complète (la structure seulement ou la structure avec les données),
 - incomplète ou erronée,
- d'exploiter un ou plusieurs formats de ressources (xlsx, csv, py, etc.).

ANNEXE

Les méthodes et les propriétés à utiliser avec la bibliothèque pandas

Catégorie	Syntaxe
<i>Création d'un DataFrame</i>	<code>IdDataFrame = pandas.DataFrame ({"Id_Colonne1":["Val1",..., "ValN"],..., "Id_ColonneM":[Val1,...,ValN]})</code>
<i>Importation des données dans un DataFrame</i>	<code>IdDataFrame = pandas.read_excel ("Chemin/ Nom_Fichier.extension", "Nom_Feuille")</code> <code>IdDataFrame = pandas.read_csv ("Chemin/Nom_Fichier.extension", sep = "séparateur")</code>
<i>Manipulation d'un DataFrame</i>	<ul style="list-style-type: none"> ▪ Affichage des dimensions d'un DataFrame : <code>IdDataFrame.shape</code> ou bien <code>print (IdDataFrame.shape)</code> ▪ Affichage du nombre d'éléments d'un DataFrame : <code>IdDataFrame.size</code> ou bien <code>print (IdDataFrame.size)</code> ▪ Affichage des informations concernant un DataFrame : <code>IdDataFrame.info ()</code> ou bien <code>print (IdDataFrame.info ())</code> ▪ Affichage du résumé rapide d'un DataFrame : <code>IdDataFrame.describe ()</code> ou bien <code>print (IdDataFrame.describe ())</code> ▪ Affichage des noms des colonnes d'un DataFrame : <code>IdDataFrame.columns</code> ou bien <code>print (IdDataFrame.columns)</code> ▪ Renommage des libellés d'une ou de plusieurs colonnes : <code>Resultat = IdDataFrame.rename (columns = {"NomColonne1": "NouveauNomColonne1", "NomColonne2": "NouveauNomColonne2", ... })</code> ▪ Suppression des lignes : <code>Resultat = IdDataFrame.drop ([N° Ligne1, ... , N° LigneN])</code> ▪ Suppression de lignes consécutives : <code>Resultat = IdDataFrame.drop (IdDataFrame.index [N° LigneInitial : N° LigneFinal])</code> ▪ Suppression des colonnes : <code>Resultat = IdDataFrame.drop (columns = ["NomColonne1",..., "NomColonneN"])</code>
<i>Affichage des données d'un DataFrame</i>	<ul style="list-style-type: none"> ▪ Affichage du contenu d'un DataFrame : <code>IdDataFrame</code> ou bien <code>print (IdDataFrame)</code> ▪ Affichage du contenu d'une colonne : <code>IdDataFrame ["NomColonne"]</code> ou bien <code>print (IdDataFrame ["NomColonne"])</code> ▪ Affichage du contenu de plusieurs colonnes : <code>Id_DataFrame [["Id_Colonne1", "Id_Colonne2", ...]]</code> ou bien <code>print (Id_DataFrame [["Id_Colonne1", "Id_Colonne2", ...]])</code> ▪ Affichage du contenu d'une colonne entre une position initiale et une position finale – 1 : <code>IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal]</code> ou bien <code>print (IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal])</code> ▪ Affichage du contenu d'une ligne : <code>IdDataFrame.iloc [N° Ligne]</code> ou bien <code>print (IdDataFrame.iloc [N° Ligne])</code> ▪ Affichage des n premières lignes d'un DataFrame : <code>IdDataFrame.head (n)</code> ou bien <code>print (IdDataFrame.head (n))</code> ▪ Affichage des n dernières lignes d'un DataFrame : <code>IdDataFrame.tail (n)</code> ou bien <code>print (IdDataFrame.tail (n))</code> ▪ Affichage du contenu des lignes entre une position initiale et une position finale – 1 : <code>IdDataFrame.iloc [N° LigneInitial : N° LigneFinal]</code> ou bien <code>print (IdDataFrame.iloc [N° LigneInitial : N° LigneFinal])</code> ▪ Affichage du contenu d'une cellule : <code>IdDataFrame.iloc [N° Ligne , N° Colonne]</code> ou bien <code>print (IdDataFrame.iloc [N° Ligne , N° Colonne])</code>

Catégorie	Syntaxe																										
<i>Modification et ajout des données dans un DataFrame</i>	<ul style="list-style-type: none"> Modification du contenu d'une cellule d'un DataFrame : <code>IdDataFrame.loc [N° ligne , "NomColonne"] = Valeur (ou Formule)</code> Modification des données d'une ligne d'un DataFrame : <code>Id_DataFrame.loc [N° ligne] = [Liste_Valeur]</code> Ajout d'une colonne dans un DataFrame : <code>Id_DataFrame ["Id_Colonne"] = Valeur (ou Formule)</code> 																										
<i>Nettoyage d'un DataFrame</i>	<ul style="list-style-type: none"> Suppression des doublons dans un DataFrame : <code>Resultat = Id_DataFrame.drop_duplicates ()</code> Suppression de toutes les lignes contenant des valeurs nulles : <code>Resultat = Id_DataFrame.dropna ()</code> Suppression de toutes les colonnes contenant des valeurs nulles : <code>Resultat = Id_DtaFrame.dropna (axis = 1)</code> 																										
<i>Les fonctions statistiques</i>	<p>mean () – min () – max () – sum () – count ()</p>																										
<i>Affichage des données d'un DataFrame selon une ou plusieurs conditions</i>	<ul style="list-style-type: none"> Affichage des données d'un DataFrame selon une condition <code>IdDataFrame [IdDataFrame ["NomColonne"] OperateurComparaison Valeur]</code> <table border="1" style="margin-left: 40px;"> <thead> <tr> <th colspan="7">Les opérateurs de comparaison</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">==</td> <td style="text-align: center;"><</td> <td style="text-align: center;"><=</td> <td style="text-align: center;">></td> <td style="text-align: center;">>=</td> <td style="text-align: center;">!=</td> <td style="text-align: center;">isin</td> </tr> </tbody> </table> Affichage des données d'un DataFrame selon plusieurs conditions <code>IdDataFrame [(IdDataFrame ["NomColonne"] OperateurComp Valeur) OperateurLogique (IdDataFrame ["NomColonne"] OperateurComp Valeur)]</code> <table border="1" style="margin-left: 40px;"> <thead> <tr> <th colspan="3">Les opérateurs logiques</th> </tr> <tr> <th>Opérateur</th> <th>Nomination</th> <th>Rôle</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">OU logique</td> <td>Vérifie qu'une des conditions est réalisée</td> </tr> <tr> <td style="text-align: center;">&</td> <td style="text-align: center;">ET logique</td> <td>Vérifie que toutes les conditions sont réalisées</td> </tr> </tbody> </table> 	Les opérateurs de comparaison							==	<	<=	>	>=	!=	isin	Les opérateurs logiques			Opérateur	Nomination	Rôle		OU logique	Vérifie qu'une des conditions est réalisée	&	ET logique	Vérifie que toutes les conditions sont réalisées
Les opérateurs de comparaison																											
==	<	<=	>	>=	!=	isin																					
Les opérateurs logiques																											
Opérateur	Nomination	Rôle																									
	OU logique	Vérifie qu'une des conditions est réalisée																									
&	ET logique	Vérifie que toutes les conditions sont réalisées																									
<i>Tri des éléments d'un DataFrame</i>	<p><code>DataFrame_Trié = IdDataFrame.sort_values (by = ["NomColonne1" , "NomColonne2" , ...] , ascending = [True/False , True/False , ...])</code></p>																										
<i>Création d'un graphique</i>	<ul style="list-style-type: none"> Création d'un graphique à barres : <code>IdDataFrame.plot.bar (x = "NomColonneAbscisse" , y = "NomColonneOrdonnée" , title = "TitreGraphique" , color = "CouleurGraphique")</code> Création d'un graphique en courbe : <code>IdDataFrame.plot.line (x = "NomColonneAbscisse" , y = "NomColonneOrdonnée" , title = "TitreGraphique" , color = "CouleurGraphique")</code> Affichage d'un graphique : <code>alias_matplotlib.show ()</code> 																										