

RÉPUBLIQUE TUNISIENNE *** Ministère de l'Éducation	EXAMEN DU BACCALAURÉAT		SESSION 2024	
	ÉPREUVE PRATIQUE D'INFORMATIQUE			
	Section : Économie et Gestion			
	Coefficient de l'épreuve : 0.5		Durée : 1h	

Le sujet comporte 3 pages numérotées de 1/3 à 3/3

Important : Il est demandé au candidat de créer dans le dossier **Bac2024** 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.

Afin de mieux gérer les affectations des conducteurs pour le transport des marchandises, une société de distribution propose d'utiliser la base de données simplifiée intitulée "**Gestion_Affectations**" décrite par le schéma textuel suivant :

CAMION (MatCam, Marque, Type, PTAC)

CONDUCTEUR (CINCond, NomPrénom, Tel)

AFFECTATION (NumAff, MatCam#, CINCond#, DateDébAff, DateFinAff, Destination, PoidsMar)

Sachant que :

- **CAMION** est une table contenant les informations relatives aux camions.
- **CONDUCTEUR** est une table contenant les informations relatives aux conducteurs.
- **AFFECTATION** est une table contenant les informations relatives aux affectations des conducteurs.

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

Nom	Description	Type	Taille	Format	Contrainte
MatCam	Matricule d'un camion	Texte	11		
Marque	Marque d'un camion	Texte	20		
Type	Type d'un camion	Texte	50		
PTAC	Poids Total Autorisé en Charge (en Tonne)	Numérique	Réel		>0
CINCond	Numéro de la CIN d'un conducteur	Texte	8		
NomPrénom	Nom et prénom d'un conducteur	Texte	50		Null interdit
Tel	Numéro de téléphone d'un conducteur	Texte	8		
NumAff	Numéro d'une affectation	Numéro Auto			
DateDébAff	Date de début d'une affectation	Date/Heure		Date, général	Null interdit
DateFinAff	Date de fin d'une affectation	Date/Heure		Date, général	Null interdit
Destination	Destination d'une marchandise	Texte	50		Null interdit
PoidsMar	Poids de la marchandise à transporter (en Tonne)	Numérique	Réel		>0

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

- 1) Créer, dans le dossier de travail, la base de données nommée "**Gestion_Affectations**".
- 2) Créer les tables et les relations de cette base de données tout en respectant les types, les tailles, les formats et les contraintes cités dans la description ci-dessus.
- 3) Remplir les tables par les données représentées dans les tableaux suivants :

CAMION			
MatCam	Marque	Type	PTAC
250 TU 2222	SCANIA	Camion Semi-remorque	40
250 TU 3333	VOLVO	Camion Semi-remorque	38
251 TU 1111	IVECO	Camion Frigorifique	7,2
252 TU 8888	MAN	Camion Plateau	35

CONDUCTEUR		
CINCond	NomPrénom	Tel
25551111	Aydi Lamjed	33222000
19999000	Nsibi Aida	38777111
19944444	Nsiri Mounir	32666222
23388888	Salhi Mondher	34999555

AFFECTATION						
NumAff	MatCam	CINCond	DateDébAff	DateFinAff	Destination	PoidsMar
1	250 TU 2222	19944444	02/04/2024	03/04/2024	Sfax	38
2	250 TU 2222	23388888	04/04/2024	10/04/2024	Gabès	40
3	251 TU 1111	19999000	13/04/2024	15/04/2024	Gafsa	5
4	252 TU 8888	19944444	04/04/2024	07/04/2024	Gabès	20
5	250 TU 3333	23388888	15/04/2024	18/04/2024	Gabès	35

4) Créer les requêtes suivantes :

- R1** : Afficher la liste des camions (**MatCam, Marque, Type, PTAC**) utilisés pour transporter des marchandises vers une ville donnée,
- R2** : Afficher les informations (**NomPrénom, Tel**) relatives aux conducteurs qui ont transportés des marchandises à **Gabès** ainsi que le nombre de missions effectuées,
- R3** : Afficher toutes les affectations (**MatCam, NomPrénom, Destination, PoidsMar, PTAC**) faites pendant le mois d'**avril 2024** et dont le poids de la marchandise transportée dépasse **10 tonnes**.

5) Exporter, dans le dossier de travail, le résultat de la requête "**R3**" dans un classeur à nommer "**missions.xlsx**".

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

- Créer un fichier et l'enregistrer sous le nom "**Missions**" dans le dossier de travail.
- Utiliser les méthodes et les propriétés adéquates, à partir de **l'annexe de la page 3/3**, pour écrire dans le fichier "**Missions**" les scripts permettant de (d') :
 - importer le contenu du classeur "**missions.xlsx**" et le stocker dans un DataFrame à nommer "**dfmis**",
 - afficher le contenu du DataFrame "**dfmis**",
 - renommer la colonne "**PoidsMar**" du DataFrame "**dfmis**" par "**Poids**",
 - trier le DataFrame "**dfmis**" dans un nouveau DataFrame à nommer "**dfmis_tri**" selon l'ordre croissant des poids des marchandises transportées,
 - afficher la quantité moyenne de marchandises transportées par le conducteur "**Nsiri Mounir**",
 - calculer dans une nouvelle colonne à nommer "**Reste Charge**" et pour chaque mission, la charge restante dans le camion utilisé puis afficher le contenu du DataFrame "**dfmin**" sachant que : **Reste Charge = PTAC – Poids**.

Grille d'évaluation	Partie	A (13 points)					B (7 points)						
	Question	1	2	3	4	5	1	2.a.	2.b.	2.c.	2.d.	2.e.	2.f.
	Note	0.5	4.5	3	4	1	0.5	1	0.25	1.5	1.25	1.25	1.25

ANNEXE

Les méthodes et les propriétés à utiliser avec la bibliothèque pandas

Catégorie	Syntaxe
Création d'un DataFrame	IdDataFrame = pandas.DataFrame ({"NomColonne1":["Val1",..., "ValN"],..., "NomColonneM":["Val1,...,ValN"]})
Importation des données dans un DataFrame	IdDataFrame = pandas.read_excel ("Chemin/ Nom_Fichier.extension", "Nom_Feuille") IdDataFrame = pandas.read_csv ("Chemin/Nom_Fichier.extension", sep = "séparateur")
Manipulation d'un DataFrame	IdDataFrame.shape ou bien print (IdDataFrame.shape) IdDataFrame.size ou bien print (IdDataFrame.size) IdDataFrame.info () ou bien print (IdDataFrame.info ()) IdDataFrame.describe () ou bien print (IdDataFrame.describe ()) IdDataFrame.columns ou bien print (IdDataFrame.columns) Resultat = IdDataFrame.rename (columns = {"NomColonne1": "NouveauNomColonne1", "NomColonne2": "NouveauNomColonne2", ... }) Resultat = IdDataFrame.drop ([N° Ligne1, ... , N° LigneN]) Resultat = IdDataFrame.drop (IdDataFrame.index [N° LigneInitial : N° LigneFinal]) Resultat = IdDataFrame.drop (columns = ["NomColonne1",..., "NomColonneN"])
Affichage des données d'un DataFrame	IdDataFrame ou bien print (IdDataFrame) IdDataFrame ["NomColonne"] ou bien print (IdDataFrame ["NomColonne"]) Id_DataFrame [["NomColonne1", "NomColonne2", ...]] ou bien print (Id_DataFrame [["NomColonne1", "NomColonne2", ...]]) IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal] ou bien print (IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal]) IdDataFrame.iloc [N° Ligne] ou bien print (IdDataFrame.iloc [N° Ligne]) IdDataFrame.head (n) ou bien print (IdDataFrame.head (n)) IdDataFrame.tail (n) ou bien print (IdDataFrame.tail (n)) IdDataFrame.iloc [N° LigneInitial : N° LigneFinal] ou bien print (IdDataFrame.iloc [N° LigneInitial : N° LigneFinal]) IdDataFrame.iloc [N° Ligne , N° Colonne] ou bien print (IdDataFrame.iloc [N° Ligne , N° Colonne])
Modification et ajout des données dans un DataFrame	IdDataFrame.loc [N° ligne , "NomColonne"] = Valeur (ou Formule) Id_DataFrame.loc [N° ligne] = [Liste_Valeur] Id_DataFrame ["NomColonne"] = Valeur (ou Formule)
Nettoyage d'un DataFrame	Resultat = Id_DataFrame.drop_duplicates () Resultat = Id_DataFrame.dropna () Resultat = Id_DtaFrame.dropna (axis = 1)
Les fonctions statistiques	mean () – min () – max () – sum () – count ()
Affichage des données d'un DataFrame selon une ou plusieurs conditions	IdDataFrame [IdDataFrame ["NomColonne"] OperateurComparaison Valeur] IdDataFrame [(IdDataFrame ["NomColonne"] OperateurComp Valeur) OperateurLogique (IdDataFrame ["NomColonne"] OperateurComp Valeur)]
Tri des éléments d'un DataFrame	DataFrame_Trié = IdDataFrame.sort_values (by = ["NomColonne1" , "NomColonne2" , ...] , ascending = [True/False , True/False , ...])
Création d'un graphique	IdDataFrame.plot.bar (x = "NomColonneAbscisse", y = "NomColonneOrdonnée", title = "TitreGraphique", color = "CouleurGraphique") IdDataFrame.plot.line (x = "NomColonneAbscisse", y = "NomColonneOrdonnée", title = "TitreGraphique", color = "CouleurGraphique") alias_matplotlib.show ()