

RÉPUBLIQUE TUNISIENNE ☆☆☆ Ministère de l'Éducation	EXAMEN DU BACCALAURÉAT	SESSION 2023
	ÉPREUVE PRATIQUE D'INFORMATIQUE	
	Section : Économie et Gestion	
	Coefficient de l'épreuve : 0.5	Durée : 1h

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

Important : Dans le répertoire **Bac2023** situé sur la racine du disque C de votre poste, créez un dossier de travail portant votre numéro d'inscription (6 chiffres) et dans lequel vous devez enregistrer, au fur et à mesure, tous les fichiers solutions de ce sujet.

Afin de gérer le suivi du ravitaillement (rechargement) en carburant relatif à ses véhicules, une société se propose d'utiliser la base de données simplifiée intitulée "**GestionCarburant**" décrite par le schéma textuel suivant :

CARBURANT (CodeCarb, NomCarb)

VEHICULE (Immatriculation, Marque, Puissance, CodeCarb#)

STATION (CodeStat, NomStat, Adresse)

RAVITAILLEMENT (Immatriculation#, CodeStat#, DateRav, HeureRav, Quantité, PrixLit)

Sachant que :

- **CARBURANT** est une table contenant les informations relatives aux différents carburants (Gasoil, Essence, etc.)
- **VEHICULE** est une table contenant les informations relatives aux différents véhicules de la société.
- **STATION** est une table contenant les informations relatives aux différentes stations.
- **RAVITAILLEMENT** est une table contenant les informations relatives aux différents rechargements en carburant effectués.

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

Nom	Description	Type	Taille	Format	Contrainte
CodeCarb	Code d'un carburant	Texte	4		
NomCarb	Nom d'un carburant	Texte	100		Null interdit
Immatriculation	Immatriculation d'un véhicule	Texte	11		
Marque	Marque d'un véhicule	Texte	50		
Puissance	Puissance d'un véhicule	Numérique	Entier		
CodeStat	Code d'une station	Texte	4		
NomStat	Nom d'une station	Texte	100		Null interdit
Adresse	Adresse d'une station	Texte	100		
DateRav	Date d'un ravitaillement	Date/Heure		Date, abrégé	
HeureRav	Heure d'un ravitaillement	Date/Heure		Heure, abrégé	
Quantite	Quantité d'un ravitaillement en litres	Numérique	Entier		Null interdit et > 0
PrixLit	Prix d'un litre de carburant en millimes	Numérique	Entier		Null interdit et > 0

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

- 1) Créer, dans votre dossier de travail, la base de données à nommer "**GestionCarburant**".
- 2) Créer les tables et les relations relatives à 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 différentes tables par les données représentées dans les tableaux suivants :

CARBURANT		VEHICULE			
CodeCarb	NomCarb	Immatriculation	Marque	Puissance	CodeCarb
C001	Gasoil sans soufre	2673 TU 131	BMW	7	C001
C002	Gasoil	6762 TU 86	TOYOTA	10	C002
C003	Essence sans plomb super	7861 TU 140	VOLKSWAGEN	5	C003
C004	Essence sans plomb premier	9362 TU 177	CITROEN	4	C004

STATION		
CodeStat	NomStat	Adresse
S001	TotalEnergies	Borjine
S002	Shell	Sidi Khalifa
S003	Agil	El Jem
S004	Oilibya	M'Saken

RAVITAILLEMENT					
Immatriculation	CodeStat	DateRav	HeureRav	Quantite	PrixLit
9362 TU 177	S002	29/05/2022	3:30 PM	20	2600
6762 TU 86	S001	13/01/2023	12:00 PM	30	1790
6762 TU 86	S003	15/01/2023	4:00 PM	45	1790
7861 TU 140	S001	16/02/2022	1:00 AM	10	2525
9362 TU 177	S004	20/02/2023	3:00 AM	44	2855

4) Créer les requêtes suivantes :

R1 : Afficher la liste des véhicules (**Immatriculation, Marque, Puissance, NomCarb**).

R2 : Afficher, pour chaque carburant, son code, son nom ainsi que la quantité totale rechargée.

R3 : Afficher la liste des ravitaillements effectués durant l'année 2023, triée par ordre croissant des dates de ravitaillements (**Immatriculation, Marque, Puissance, NomCarb, NomStat, DateRav, HeureRav, Quantite, PrixLit**).

5) Exporter, dans votre dossier de travail sous le nom "**ravitaillement.csv**", le contenu de la table "**RAVITAILLEMENT**" au format **csv** (utiliser le **Point-virgule** comme séparateur de champs, le **point** comme 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 "**carburant**" dans votre dossier de travail.
- 2) Choisir les méthodes et les propriétés adéquates, à partir de l'annexe de la page 3, pour écrire dans le fichier créé les scripts permettant de/d' :
 - a) importer le contenu du fichier "**ravitaillement.csv**" et le stocker dans un DataFrame à nommer "**rav**",
 - b) afficher les noms des colonnes du DataFrame "**rav**",
 - c) supprimer la colonne "**HeureRav**",
 - d) afficher la liste des véhicules (**Immatriculation, DateRav, Quantite**) ayant rechargés des quantités de carburant supérieures strictement à 30 litres,
 - e) calculer, dans une nouvelle colonne à nommer "**Montant**", le montant à payer pour chaque ravitaillement puis afficher le DataFrame "**rav**". Sachant que : **Montant = Quantite * PrixLit**,
 - f) Afficher le montant total des ravitaillements effectués.

Grille d'évaluation	Partie	A (12 points)					B (8 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	2	4	1	0.5	1.25	0.5	1.25	2	1.25	1.25

ANNEXE

Les méthodes et les propriétés à utiliser avec la bibliothèque pandas

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