

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 d'améliorer la gestion du flux des passagers au niveau des postes frontaliers terrestres tunisiens, on se propose d'utiliser la base de données simplifiée intitulée "**Poste_Frontalier**" décrite par le schéma textuel suivant :

POSTE (CodeP, Nom, Gouvernorat, Frontière, Catégorie, Superficie)

VOYAGEUR (IdVoy, NomPrénom, Passeport, Nationalité, Genre)

PASSAGE (NumPass, CodeP#, IdVoy#, Date, Accès)

Sachant que :

- **POSTE** est une table contenant les informations à propos des postes frontaliers terrestres tunisiens.
- **VOYAGEUR** est une table contenant les informations relatives aux voyageurs.
- **PASSAGE** est une table contenant les informations relatives aux flux des passagers au niveau des postes frontaliers terrestres tunisiens.

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

Nom	Description	Type	Taille	Format	Contrainte
CodeP	Code d'un poste frontalier	Texte	3		
Nom	Nom d'un poste frontalier	Texte	50		Null interdit
Gouvernorat	Gouvernorat d'un poste frontalier	Texte	50		
Frontière	"TA" pour "Tuniso-Algérienne" ou "TL" pour "Tuniso-Libyenne"	Texte	2		"TA" par défaut
Catégorie	"C" pour "Commercial" ou "T" pour "Touristique"	Texte	1		"C" par défaut
Superficie	Superficie d'un poste frontalier en m ²	Numérique	Entier		> 0
IdVoy	Identifiant d'un voyageur	Texte	8		
NomPrénom	Nom et prénom d'un voyageur	Texte	50		Null interdit
Passeport	Numéro du passeport d'un voyageur	Texte	10		Null interdit
Nationalité	Nationalité d'un voyageur	Texte	20		
Genre	Genre d'un voyageur ("M" ou "F")	Texte	1		
NumPass	Numéro de passage	NuméroAuto			
Date	Date de passage	Date/Heure		Date, abrégé	
Accès	"E" pour "Entrée" ou "S" pour "Sortie"	Texte	1		

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 "**Poste_Frontalier**".
- 2) Créer les tables et les relations de cette base de données tout en respectant les types, les propriétés et les contraintes cités dans la description ci-dessus.

3) Remplir les tables par les données représentées dans les tableaux suivants :

POSTE					
CodeP	Nom	Gouvernorat	Frontière	Catégorie	Superficie
P01	Ras Jedir	Médenine	TL	C	308754
P02	Dhehiba	Tataouine	TL	C	200000
P03	Melloula	Jendouba	TA	T	45340

VOYAGEUR				
IdVoy	NomPrénom	Passeport	Nationalité	Genre
V0000001	Briki Ridha	W111111	Tunisienne	M
V0000002	Hamrouni Maram	224466777	Algérienne	F
V0000003	Zouari Mongi	Z445544	Tunisienne	M
V0000004	Nheri Sarra	110006666	Algérienne	F

PASSAGE				
NumPass	CodeP	IdVoy	Date	Accès
1	P01	V0000001	22/11/2023	S
2	P01	V0000002	08/01/2024	S
3	P01	V0000001	27/01/2024	E
4	P03	V0000003	10/02/2024	S

4) Créer les requêtes suivantes :

R1 : Afficher la liste des postes frontaliers commerciaux (**CodeP**, **Nom**, **Gouvernorat**).

R2 : Afficher la liste des voyageurs (**NomPrénom**, **Nationalité**, **Genre**) qui ont quitté le territoire tunisien avant le mois de **février 2024** par le poste frontalier de code **"P01"**.

R3 : Afficher, pour chaque poste, son nom ainsi que le nombre de voyageurs sortants.

5) Exporter, dans le dossier de travail et sous le nom **"poste.csv"**, le contenu de la table **"POSTE"** au format **csv** (**Point-virgule** en séparateur de champs, **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 **"Poste"** dans le dossier de travail.

2) Choisir les méthodes et les propriétés adéquates, à partir de **l'annexe de la page 3/3**, pour écrire dans le fichier **"Poste"** les scripts permettant de (d') :

a) importer le contenu du fichier **"poste.csv"** et le stocker dans un DataFrame à nommer **"dfpos"**,

b) afficher le contenu du DataFrame **"dfpos"**,

c) ajouter à la fin du DataFrame **"dfpos"** la ligne suivante :

CodeP	Nom	Gouvernorat	Frontière	Catégorie	Superficie
P04	Hazoua	Tozeur	TA	C	90376

d) modifier la superficie du poste frontalier **"Dhehiba"** par **230323**,

e) trier le DataFrame **"dfpos"** dans un nouveau DataFrame à nommer **"dfpos_tri"** selon l'ordre croissant des superficies et afficher les informations relatives au poste ayant la superficie la plus grande,

f) créer et afficher un graphique en barres bleues à partir du DataFrame **"dfpos"**, représentant pour chaque nom de poste frontalier, sa superficie et ayant comme titre **"Superficies des postes frontaliers"**.

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	1.25	0.25	1	1	1.25	1.75

Les méthodes et les propriétés à utiliser avec la bibliothèque pandas

Catégorie	Syntaxe
Création d'un DataFrame	<code>IdDataFrame = pandas.DataFrame ({"NomColonne1":["Val1",..., "ValN"],..., "NomColonneM":["Val1",...,ValN]})</code>
Importation des données dans un DataFrame	<code>IdDataFrame = pandas.read_excel ("Chemin/ Nom_Fichier.extension", "Nom_Feuille")</code> <code>IdDataFrame = pandas.read_csv ("Chemin/Nom_Fichier.extension", sep = "séparateur")</code>
Manipulation d'un DataFrame	<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>
Affichage des données d'un DataFrame	<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>
Modification et ajout des données dans un DataFrame	<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>
Nettoyage d'un DataFrame	<code>Resultat = Id_DataFrame.drop_duplicates ()</code> <code>Resultat = Id_DataFrame.dropna ()</code> <code>Resultat = Id_DtaFrame.dropna (axis = 1)</code>
Les fonctions statistiques	<code>mean ()</code> – <code>min ()</code> – <code>max ()</code> – <code>sum ()</code> – <code>count ()</code>
Affichage des données d'un DataFrame selon une ou plusieurs conditions	<code>IdDataFrame [IdDataFrame ["NomColonne"] OperateurComparaison Valeur]</code> <code>IdDataFrame [(IdDataFrame ["NomColonne"] OperateurComp Valeur) OperateurLogique (IdDataFrame ["NomColonne"] OperateurComp Valeur)]</code>
Tri des éléments d'un DataFrame	<code>DataFrame_Trié = IdDataFrame.sort_values (by = ["NomColonne1" , "NomColonne2" , ...] , ascending = [True/False , True/False , ...])</code>
Création d'un graphique	<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>