

RÉPUBLIQUE TUNISIENNE ★★★ MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT SESSION 2022		NOUVEAU RÉGIME
	ÉPREUVE PRATIQUE D'INFORMATIQUE		
	Section : Économie et Gestion		
	Coefficient de l'épreuve : 0.5		Durée : 1h

Important : Dans le répertoire **Bac2022** 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.

L'association "Faune et Flore" veut améliorer la gestion des visites aux différents parcs nationaux afin d'assurer la découverte des espèces naturelles (animale et végétale). L'informaticien responsable de ce projet au sein de cette association propose d'utiliser la base de données simplifiée intitulée "**Gestion_Parc**" décrite par le schéma textuel suivant :

Parc (CodeParc, NomParc, Ville)

Espec (CodeEspec, NomEspec, Protection)

Parc_Espec (CodeParc#, CodeEspec#)

Visite (DateVisite, CodeParc#, NbParticipants, Frais)

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

Nom	Description	Type	Taille	Format	Contrainte
CodeParc	Code d'un parc	Texte court	4		
NomParc	Nom d'un parc	Texte court	50		Null interdit
Ville	Ville où se trouve un parc	Texte court	50		
CodeEspec	Code d'une espèce	Texte court	4		
NomEspec	Nom d'une espèce	Texte court	20		Null interdit
Protection	Etat de protection d'une espèce (protégée/non protégée)	Texte court	3		Null interdit
DateVisite	Date d'une visite	Date/Heure		Date abrégé	
NbParticipants	Nombre de participants à une visite	Numérique	Entier		> 0
Frais	Frais de participation à une visite par personne exprimé en DT	Numérique	Entier		> 0

N.B. : L'association ne peut programmer qu'une seule visite par jour à un parc donné. Toutefois, elle peut programmer, pour la même journée, plusieurs visites à des parcs différents.

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 "**Gestion_Parc**".
- 2) Créer les tables et les relations de cette base de données tout en respectant les types et les propriétés cités dans la description ci-dessus.
- 3) Remplir ces tables par les données représentées dans les tableaux suivants :

Parc		
CodeParc	NomParc	Ville
P001	Chaâmbi	Kasserine
P002	Ichkeul	Bizerte
P003	El Feija	Jendouba
P004	Dghoumès	Tozeur

Espec		
CodeEspec	NomEspec	Protection
FL01	Chêne zéen	Non
FL02	Talh	Oui
FN01	Sanglier	Non
FN02	Mouflon	Oui

Parc_Espece	
CodeParc	CodeEspece
P001	FL01
P003	FL01
P003	FN01
P004	FL02
P001	FN02

Visite			
DateVisite	CodeParc	NbParticipants	Frais
27/02/2022	P001	50	30
24/03/2022	P002	40	20
25/03/2022	P002	35	20
26/03/2022	P004	50	40
03/04/2022	P002	35	20

4) Créer les requêtes suivantes :

R1 : Afficher les noms et les villes des parcs.

R2 : Afficher les noms des espèces protégées du parc "Chaâmbi".

R3 : Afficher la liste des visites (**DateVisite**, **NomParc**, **NbParticipants**, **Frais**) réalisées pendant le mois de mars de l'année 2022, triée par ordre décroissant de la date de visite.

5) Exporter, dans votre dossier de travail et sous le nom "**visites_mars.csv**", le résultat de la requête "**R3**" 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 :

- Créer un fichier et l'enregistrer sous le nom "**visite**" dans votre dossier de travail.
- Choisir les méthodes et les propriétés adéquates, à partir de l'annexe (page 3), pour écrire dans le fichier créé les scripts permettant de/d':
 - 1) Importer le contenu du fichier "**visites_mars.csv**" et le stocker dans un DataFrame à nommer "**visite**".
 - 2) Afficher le résumé de ce DataFrame.
 - 3) Afficher les dates des différentes visites de ce DataFrame.
 - 4) Calculer et afficher, pour chaque visite, le montant total des frais dans une nouvelle colonne à nommer "**TotalFrais**", sachant que :

$$\text{TotalFrais} = \text{NbParticipants} * \text{Frais}$$
 - 5) Afficher le nombre total des participants.
 - 6) Créer un graphique en lignes représentant le nombre des participants en fonction des dates des visites ayant pour titre "**Participants par journée**".

Grille d'évaluation	Partie	A (13 points)					B (7 points)					
	Question	1	2	3	4	5	1	2	3	4	5	6
	Note	1	4.5	2.5	4	1	1	0.5	1	1.5	1.5	1.5

ANNEXE

Les méthodes et les propriétés à utiliser avec la bibliothèque pandas

Catégorie	Syntaxe
Création et importation d'un DataFrame	<p>IdDataFrame = pandas.read_excel ("Chemin/ Nom_Fichier.extension", "Nom_Feuille")</p> <p>IdDataFrame = pandas.read_csv ("Chemin/Nom_Fichier.extension", sep = "séparateur")</p>
Manipulation d'un DataFrame	<p>IdDataFrame.shape ou bien print (IdDataFrame.shape)</p> <p>IdDataFrame.size ou bien print (IdDataFrame.size)</p> <p>IdDataFrame.info () ou bien print (IdDataFrame.info ())</p> <p>IdDataFrame.describe () ou bien print (IdDataFrame.describe ())</p> <p>IdDataFrame.columns ou bien print (IdDataFrame.columns)</p> <p>IdDataFrame.rename (columns = { "NomColonne1" : "NouveauNomColonne1", "NomColonne2" : "NouveauNomColonne2", ... } , inplace= True)</p> <p>Resultat = IdDataFrame.drop (columns = ["NomColonne1",..., "NomColonneN"])</p> <p>Resultat = IdDataFrame.drop ([Indice_Ligne])</p>
Affichage des données d'un DataFrame	<p>IdDataFrame ou bien print (IdDataFrame)</p> <p>IdDataFrame ["NomColonne"] ou bien print (IdDataFrame ["NomColonne"])</p> <p>IdDataFrame [["NomColonne1" , "NomColonne2" , ...]] ou bien print (IdDataFrame [["NomColonne1" , "NomColonne2" , ...]])</p> <p>IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal] ou bien print (IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal])</p> <p>IdDataFrame.head (n) ou bien print (IdDataFrame.head (n))</p> <p>IdDataFrame.tail (n) ou bien print (IdDataFrame.tail (n))</p> <p>IdDataFrame.iloc [N° Ligne] ou bien print (IdDataFrame.iloc [N° Ligne])</p> <p>IdDataFrame.iloc [N° LigneInitial : N° LigneFinal] ou bien print (IdDataFrame.iloc [N° LigneInitial : N° LigneFinal])</p> <p>IdDataFrame.iloc [N° Ligne , N° Colonne] ou bien print (IdDataFrame.iloc [N° Ligne , N° Colonne])</p> <p>IdDataFrame.loc [N° ligne , "NomColonne"] = Valeur</p> <p>IdDataFrame.loc [N° ligne] = [Liste_Valeur]</p> <p>IdDataFrame ["NomColonne"] = Valeur</p> <p>IdDataFrame [IdDataFrame ["NomColonne"] OperateurComparaison Valeur]</p> <p>IdDataFrame [(IdDataFrame ["NomColonne"] OperateurComp Valeur) OperateurLogique (IdDataFrame ["NomColonne"] OperateurComp Valeur)]</p>
Les fonctions statistiques	<p>mean () – min () – max () – sum () – count ()</p>
Tri des éléments d'un DataFrame	<p>DataFrame_Trié = IdDataFrame.sort_values (by = ["NomColonne1" , "NomColonne2" , ...] , ascending = [True/False , True/False , ...])</p>
Création d'un graphique	<p>import matplotlib.pyplot as alias_matplotlib</p> <p>IdDataFrame.plot.bar (x = "NomColonneAbscisse" , y = "NomColonneOrdonnée" , title = "TitreGraphique" , color = "CouleurGraphique")</p> <p>IdDataFrame.plot.line (x = "NomColonneAbscisse" , y = "NomColonneOrdonnée" , title = "TitreGraphique" , color = "CouleurGraphique")</p> <p>alias_matplotlib.show ()</p>