

RÉPUBLIQUE TUNISIENNE ☆☆☆ Ministère de l'Éducation	EXAMEN DU BACCALAURÉAT		SESSION 2025	
	ÉPREUVE PRATIQUE D'INFORMATIQUE			
	Section : Économie et Gestion			
	Coefficient de l'épreuve: 0.5		Durée: 1h	

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

Important : Dans le répertoire **Bac2025** 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 d'informatiser la gestion des parties de chasse, le responsable d'un club de chasse utilise la base de données simplifiée intitulée "**GestionChasse**" décrite par la représentation textuelle suivante :

CHASSEUR (IdChass, NomPrenChass, DateNaisChass, TelChass)

ESPECE (CodEsp, NomEsp, PoidsMoy)

NIVEAUTIR (CodNiv, Libelle, NbPoints)

PARTIE (IdChass#, CodEsp#, DatePart, NbChasses, CodNiv#)

Sachant que :

- **CHASSEUR** est une table contenant les informations relatives aux chasseurs.
- **ESPECE** est une table contenant les informations relatives aux différentes espèces chassables.
- **NIVEAUTIR** est une table contenant les informations des différents niveaux de tir relatifs aux types d'espèces.
- **PARTIE** est une table contenant les informations relatives aux différentes parties de chasse.

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

Nom	Description	Type	Taille	Format	Contrainte
IdChass	Numéro d'identification d'un chasseur.	Texte court	3		
NomPrenChass	Nom et prénom d'un chasseur.	Texte court	30		
DateNaisChass	Date de naissance d'un chasseur.	Date/Heure		Date, abrégé	
TelChass	Numéro de téléphone d'un chasseur.	Texte court	8		
CodEsp	Code d'une espèce chassable.	Texte court	3		
NomEsp	Nom d'une espèce.	Texte court	30		
PoidsMoy	Poids moyen d'une espèce en Kg.	Numérique	Réel simple		> 0
CodNiv	Code d'un niveau de tir.	Texte court	3		
Libelle	Libellé d'un niveau de tir.	Texte court	30		
NbPoints	Nombre de points pour un niveau de tir.	Numérique	Entier		Null interdit
DatePart	Date d'une partie de chasse.	Date/Heure		Date, abrégé	
NbChasses	Nombre de gibiers chassés par espèce.	Numérique	Entier		

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 "**GestionChasse**".
- 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 :

CHASSEUR			
IdChass	NomPrenChass	DateNaisChass	TelChass
C01	Salem TRIKI	12/02/1980	82336655
C02	Khaled MTIRI	20/03/1992	82444777
C03	Farid LATRECH	28/10/1985	82666222

ESPECE		
CodEsp	NomEsp	PoidsMoy
E01	Pigeon	0,5
E02	Lapin	1,25
E03	Caille	0,3

NIVEAUTIR		
CodNiv	Libelle	NbPoints
TR1	Tir facile	5
TR2	Tir moyen	10
TR3	Tir difficile	15
TR4	Tir très difficile	25

PARTIE				
IdChass	CodEsp	DatePart	NbChasses	CodNiv
C01	E02	05/12/2024	3	TR1
C02	E03	12/02/2025	2	TR3
C01	E01	20/03/2025	1	TR3
C03	E02	22/01/2025	0	TR2
C02	E02	05/12/2024	4	TR2
C03	E03	12/02/2025	2	TR4

4) Créer les requêtes suivantes :

R1 : Afficher les noms et les prénoms des chasseurs, la date de la partie de chasse et le nombre de gibiers chassés pour un code d'espèce donné.

R2 : Afficher la liste des parties de chasse (**NomPrenChass**, **DatePart**, **NomEsp**, **PoidsMoy**, **CodNiv**, **NbPoints**, **NbChasses**) triée par ordre croissant des dates des parties de chasse.

R3 : Afficher le nom et le poids total de chaque espèce chassée.

5) Exporter, dans votre dossier de travail sous le nom "**Partie.csv**", le contenu de la requête "**R2**" au format csv (Utiliser un **point-virgule** comme séparateur de champs, un **point** comme séparateur décimal, la 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 "**Chasse**" dans votre dossier de travail.

2) Dans ce fichier, choisir les méthodes et les propriétés adéquates à partir de l'annexe (pages 3 et 4), pour écrire les scripts permettant :

a) d'importer le contenu du fichier "**Partie.csv**" et de le stocker dans un DataFrame à nommer "**gibiers**",

b) de renommer le libellé de la colonne "**NbChasses**" par "**NbGibiers**",

c) de modifier, dans la ligne numéro 1, la valeur initiale de "**NbGibiers**" par 4 et d'afficher cette ligne,

d) de calculer, dans une nouvelle colonne à nommer "**Score**", le score de chaque partie de chasse puis d'afficher le contenu du DataFrame "**gibiers**", sachant que : **Score = NbGibiers * NbPoints**

e) d'afficher le nombre total de lapins chassés,

f) de trier le DataFrame "**gibiers**" dans un nouveau DataFrame à nommer "**gibiers_tri**" selon l'ordre décroissant des scores puis d'afficher le DataFrame trié.

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	3.75	3	4.5	1.25	0.5	1.25	0.5	0.75	1.25	1.25	1.5

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>	▪ <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.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>IdDataFrame [["Id_Colonne1", "Id_Colonne2", ...]]</code> ou bien <code>print (IdDataFrame [["Id_Colonne1", "Id_Colonne2", ...]])</code> ▪ <code>IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal]</code> ou bien <code>print (IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal])</code> ▪ <code>IdDataFrame.loc [N° Ligne]</code> ou bien <code>print (IdDataFrame.loc [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.loc [N° LigneInitial : N° LigneFinal]</code> ou bien <code>print (IdDataFrame.loc [N° LigneInitial : N° LigneFinal])</code> ▪ <code>IdDataFrame.loc [[N° Ligne1 , N° Ligne2 , ...]]</code> ou bien <code>print (IdDataFrame.loc [[N° Ligne1 , N° Ligne2 , ...]])</code> ▪ <code>IdDataFrame.loc [N° Ligne , "NomColonne"]</code> ou bien <code>print (IdDataFrame.loc [N° Ligne , "NomColonne"])</code>

Catégorie	Syntaxe
<i>Modification et ajout des données dans un DataFrame</i>	▪ <code>IdDataFrame.loc [N° ligne , "NomColonne"] = Valeur (ou Formule)</code> ▪ <code>IdDataFrame.loc [N° ligne] = [Liste_Valeur]</code> ▪ <code>IdDataFrame ["Id_Colonne"] = Valeur (ou Formule)</code>
<i>Nettoyage d'un DataFrame</i>	▪ <code>IdDataFrame.drop_duplicates ()</code> ▪ <code>IdDataFrame.dropna ()</code> ▪ <code>IdDataFrame.dropna (axis = 1)</code>
<i>Les fonctions statistiques</i>	Liste des fonctions : <code>mean ()</code> – <code>min ()</code> – <code>max ()</code> – <code>sum ()</code> – <code>count ()</code> Syntaxe : <code>IdDataFrame ["NomColonne"].NomFonction()</code> ou bien <code>print (IdDataFrame ["NomColonne"].NomFonction())</code> Utilisation des fonctions avec groupement : Regroupement des données selon une colonne : <code>IdDataFrame.groupby ("Id_Colonne_à_regrouper") ["Id_Colonne_à_aggréger"].NomFonction()</code> ou bien <code>print (IdDataFrame.groupby ("Id_Colonne_à_regrouper") ["Id_Colonne_à_aggréger"].NomFonction())</code> Regroupement des données selon plusieurs colonnes : <code>IdDataFrame [["Id_Colonne1", "Id_Colonne2", ...]].groupby ("Id_Colonne1_Critère").NomFonction()</code> ou bien <code>print (IdDataFrame [["Id_Colonne1", "Id_Colonne2", ...]].groupby ("Id_Colonne1_Critère").NomFonction())</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>