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 de gérer la planification des séances d'apprentissage de la conduite des motos, le responsable d'une école de conduite se propose d'utiliser la base de données simplifiée intitulée "**GestionMoto**" décrite par la représentation textuelle suivante :

MOTO (**Immat**, Cylindree)

MONITEUR (IdMon, NomPrenMon, TelMon)

APPRENTI (CINApp, NomPrenApp, DateNaisApp, TelApp**)**

PLANNING (**CINApp#**, **IdMon#**, **Immat#**, **DateSeance**, **HeureDebut**, HeureFin)

Sachant que :

- **MOTO** est une table contenant les informations relatives aux différentes motos de l'école de conduite.
- **MONITEUR** est une table contenant les informations relatives aux différents moniteurs de l'école.
- **APPRENTI** est une table contenant les informations relatives aux différents apprentis.
- **PLANNING** est une table contenant les informations relatives aux différentes séances de conduite.

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

Nom	Description	Type	Taille	Format	Contrainte
Immat	Numéro d'immatriculation d'une moto.	Texte court	7		
Cylindree	Mesure du volume du cylindrée d'une moto exprimée en CC (Centimètres cubes).	Numérique	Entier		>2
IdMon	Identifiant d'un moniteur.	Texte court	4		
NomPrenMon	Nom et prénom d'un moniteur.	Texte court	50		
TelMon	Numéro de téléphone d'un moniteur.	Texte court	8		Null interdit
CINApp	Numéro de la carte d'identité d'un apprenti.	Texte court	8		
NomPrenApp	Nom et prénom d'un apprenti.	Texte court	50		Null interdit
DateNaisApp	Date de naissance d'un apprenti.	Date/Heure		Date, abrégé	
TelApp	Numéro de téléphone d'un apprenti.	Texte court	8		Null interdit
DateSeance	Date d'une séance.	Date/Heure		Date, abrégé	
HeureDebut	Heure de début d'une séance.	Numérique	Entier		>=6 et <=21
HeureFin	Heure de fin d'une séance.	Numérique	Entier		>=7et <=22

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 "**GestionMoto**".

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 :

MOTO	
Immat	**Cylindree**
MC00101	125
MC00111	50
MC00121	600
MC00131	250
MC00122	600

MONITEUR		
IdMon	**NomPrenMon**	**TelMon**
M001	Ali SALHI	88001002
M002	Farid TOUNSI	88300200
M003	Mouna TRIKI	88600250

APPRENTI			
CINApp	**NomPrenApp**	**DateNaisApp**	**TelApp**
11111111	Ali GUEBSI	04/09/2001	82333333
08999888	Ahmed TOUNSI	15/03/2002	82555555
12222222	Faiza KLIBI	15/12/2000	82444666
05111111	Mondher SOUSSI	02/03/2000	82999777

PLANNING					
CINApp	**IdMon**	**Immat**	**DateSeance**	**HeureDebut**	**HeureFin**
11111111	M001	MC00101	15/02/2025	6	8
08999888	M002	MC00122	20/04/2025	8	9
08999888	M002	MC00121	02/05/2025	17	19
12222222	M003	MC00121	15/02/2025	9	10
05111111	M003	MC00131	25/03/2025	9	11
05111111	M003	MC00131	02/04/2025	14	16

4) Créer les requêtes suivantes :

R1 : Afficher, pour un identifiant donné d'un moniteur, les informations relatives au planning de ses séances de conduite (**NomPrenApp**, **TelApp**, **DateSeance**, **HeureDebut**, **HeureFin**, **Immat**).

R2 : Afficher les informations relatives au planning des séances de conduite (**NomPrenMon**, **NomPrenApp**, **DateSeance**, **HeureDebut**, **HeureFin**, **Immat**, **Cylindree**) effectuées pendant les mois de **février, mars et avril** de l'année **2025**, triées par ordre croissant des dates des séances.

R3 : Afficher, pour chaque apprenti, son nom et son prénom ainsi que le nombre total des séances de conduite planifiées.

5) Exporter, dans votre dossier de travail sous le nom "**Planning.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 "**Planning2025**" 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 "**Planning.csv**" et de le stocker dans un DataFrame à nommer "**plan**",

b) d'afficher les noms des colonnes de ce DataFrame,

c) d'afficher, pour chaque séance, la date, le nom et le prénom de l'apprenti, l'immatriculation et la cylindrée de la moto conduite,

d) d'afficher le nombre total des séances effectuées par l'apprenti "**Mondher SOUSSI**",

e) de calculer, dans une nouvelle colonne à nommer "**PrixSeance**", le prix de la séance de conduite puis d'afficher le contenu du DataFrame "**plan**", sachant que :

PrixSeance = (**HeureFin** - **HeureDebut)** * 30

f) de trier le DataFrame "**plan**" dans un nouveau DataFrame à nommer "**plan_tri**" selon l'ordre croissant des noms et des prénoms des moniteurs 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
Création d'un DataFrame	▪ **IdDataFrame = pandas.DataFrame ({"Id_Colonne1":["Val1",…,"ValN"],…, "Id_ColonneM":[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.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"])** ▪ **IdDataFrame [["Id_Colonne1","Id_Colonne2",…]]** ou bien **print (IdDataFrame [["Id_Colonne1","Id_Colonne2",…]])** ▪ **IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal]** ou bien **print (IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal])** ▪ **IdDataFrame.loc [N° Ligne]** ou bien **print (IdDataFrame.loc [N° Ligne])** ▪ **IdDataFrame.head (n)** ou bien **print (IdDataFrame.head (n))** ▪ **IdDataFrame.tail (n)** ou bien **print (IdDataFrame.tail (n))** ▪ **IdDataFrame.loc [N° LigneInitial : N° LigneFinal]** ou bien **print (IdDataFrame.loc [N° LigneInitial : N° LigneFinal])** ▪ **IdDataFrame.loc [[N° Ligne1 , N° Ligne2 , …]]** ou bien **print (IdDataFrame.loc [[N° Ligne1 , N° Ligne2 , …])** ▪ **IdDataFrame.loc [N° Ligne , "NomColonne"]** ou bien **print (IdDataFrame.loc [N° Ligne , "NomColonne"])**

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