

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.

Une société de gardiennage propose à ses clients des missions de sécurité de leurs locaux. Ces missions sont assurées à travers ses agences sises dans différentes régions et moyennant des agents de sécurité qualifiés.

Pour gérer les activités de ses agences, on se propose d'utiliser la base de données simplifiée intitulée "**GestionGardien**" décrite par le schéma textuel suivant :

AGENCE (CodeAg, AdresseAg, TelAg)

CLIENT (CodeCL, NomCL, AdresseCL, TelCL)

AGENT (CIN, NomAgent, TelAgent, SalaireHeure, CodeAg#)

MISSION (CodeMiss, CodeCL#, CIN#, DateMiss, LieuMiss, NbrHeure)

Sachant que :

- **AGENCE** est une table contenant les informations relatives aux différentes agences de gardiennage.
- **CLIENT** est une table contenant les informations relatives aux différents clients de la société.
- **AGENT** est une table contenant les informations relatives aux différents agents de sécurité.
- **MISSION** est une table contenant les missions effectuées.

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

Nom	Description	Type	Taille	Format	Contrainte
CodeAg	Code d'une agence	Texte	4		
AdresseAg	Adresse d'une agence	Texte	100		Null interdit
TelAg	Numéro de téléphone d'une agence	Texte	8		Null interdit
CodeCL	Code d'un client	Texte	4		
NomCL	Nom d'un client	Texte	100		Null interdit
AdresseCL	Adresse d'un client	Texte	100		
TelCL	Numéro de téléphone d'un client	Texte	8		Null interdit
CIN	Numéro de la carte d'identité d'un agent	Texte	8		
NomAgent	Nom et prénom d'un agent	Texte	100		Null interdit
TelAgent	Numéro de téléphone d'un agent	Texte	8		Null interdit
SalaireHeure	Salaire d'une heure de travail d'un agent en DT	Numérique	Entier		Null interdit et > 0
CodeMiss	Code d'une mission	Texte	4		
DateMiss	Date d'une mission	Date/Heure		Date, abrégé	
LieuMiss	Lieu d'une mission	Texte	100		
NbrHeure	Nombre d'heures de travail d'une mission	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 "**GestionGardien**".
- 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 :

AGENCE		
CodeAg	AdresseAg	TelAg
A001	Tunis	61200200
A002	Gafsa	66222333
A003	Béja	65950750

AGENT				
CIN	NomAgent	TelAgent	SalaireHeure	CodeAg
21034568	Ali BEN SALEM	22336655	8	A001
21569854	Nadim ELKEFI	96254185	10	A002
23560214	Anissa ELHENCHI	96145874	10	A002

CLIENT			
CodeCL	NomCL	AdresseCL	TelCL
C001	G.M.N	Rue Ghazala	37150150
C002	S.A.B	Avenue 02 mars 1934	39200200
C003	B.E.I.M	Rue des Jasmins	84300700

MISSION					
CodeMiss	CodeCL	CIN	DateMiss	LieuMiss	NbrHeure
M001	C001	21034568	01/02/2023	Magasin Centrale N° 1	12
M002	C002	21034568	16/02/2023	Salle de réunion	8
M003	C002	23560214	16/02/2023	Salle de réunion	6
M004	C003	23560214	15/03/2023	Salle des fêtes	8

4) Créer les requêtes suivantes :

R1 : Afficher la liste des agences (**CodeAg, AdresseAg, TelAg**).

R2 : Afficher, par ordre décroissant des noms et des prénoms des agents, la liste des missions effectuées à une date donnée (**NomAgent, NomCL, AdresseCL, LieuMiss, NbrHeure**).

R3 : Afficher, pour chaque agent les champs (**NomAgent, TelAgent, CodeAg, AdresseAg, SalaireHeure**) ainsi que le nombre total d'heures effectuées pendant le premier trimestre de l'année 2023 (**Janvier, Février, Mars**).

5) Exporter, dans votre dossier de travail sous le nom "**agent.csv**", le contenu de la requête "**R3**" 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 "**gardien**" 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 "**agent.csv**" et le stocker dans un DataFrame à nommer "**agent**",

b) afficher, pour chaque agent, son nom et son prénom (**NomAgent**) ainsi que son numéro de téléphone,

c) afficher les noms et les prénoms (**NomAgent**) des agents qui travaillent pour l'agence de code "**A001**",

d) renommer la colonne contenant le total des heures de travail réalisées par "**NbrHeureTotal**",

e) calculer, dans une nouvelle colonne à nommer "**Salaire**", le salaire de base de chaque agent puis afficher le DataFrame "**agent**". Sachant que : **Salaire = SalaireHeure * NbrHeureTotal**,

f) trier le DataFrame "**agent**" dans un nouveau DataFrame à nommer "**agent_tri**" selon l'ordre décroissant des codes des agences puis afficher le DataFrame trié.

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	1	1.5	1.25	1	1.5

ANNEXE

Les méthodes et les propriétés à utiliser avec la bibliothèque pandas

Catégorie	Syntaxe
<i>Création d'un DataFrame</i>	IdDataFrame = pandas.DataFrame ({"NomColonne1":["Val1",..., "ValN"],..., "NomColonneM":["Val1, ..., ValN"]})
<i>Importation des données dans un DataFrame</i>	IdDataFrame = pandas.read_excel ("Chemin/ Nom_Fichier.extension", "Nom_Feuille") IdDataFrame = pandas.read_csv ("Chemin/Nom_Fichier.extension", sep = "séparateur")
<i>Manipulation d'un DataFrame</i>	IdDataFrame.shape ou bien print (IdDataFrame.shape) IdDataFrame.size ou bien print (IdDataFrame.size) IdDataFrame.info () ou bien print (IdDataFrame.info ()) IdDataFrame.describe () ou bien print (IdDataFrame.describe ()) 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"])
<i>Affichage des données d'un DataFrame</i>	IdDataFrame ou bien print (IdDataFrame) IdDataFrame ["NomColonne"] ou bien print (IdDataFrame ["NomColonne"]) Id_DataFrame [["NomColonne1", "NomColonne2", ...]] ou bien print (Id_DataFrame [["NomColonne1", "NomColonne2", ...]]) IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal] ou bien print (IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal]) IdDataFrame.iloc [N° Ligne] ou bien print (IdDataFrame.iloc [N° Ligne]) IdDataFrame.head (n) ou bien print (IdDataFrame.head (n)) IdDataFrame.tail (n) ou bien print (IdDataFrame.tail (n)) IdDataFrame.iloc [N° LigneInitial : N° LigneFinal] ou bien print (IdDataFrame.iloc [N° LigneInitial : N° LigneFinal]) IdDataFrame.iloc [N° Ligne , N° Colonne] ou bien print (IdDataFrame.iloc [N° Ligne , N° Colonne])
<i>Modification et ajout des données dans un DataFrame</i>	IdDataFrame.loc [N° ligne , "NomColonne"] = Valeur (ou Formule) Id_DataFrame.loc [N° ligne] = [Liste_Valeur] Id_DataFrame ["NomColonne"] = Valeur (ou Formule)
<i>Nettoyage d'un DataFrame</i>	Resultat = Id_DataFrame.drop_duplicates () Resultat = Id_DataFrame.dropna () Resultat = Id_DtaFrame.dropna (axis = 1)
<i>Les fonctions statistiques</i>	mean () – min () – max () – sum () – count ()
<i>Affichage des données d'un DataFrame selon une ou plusieurs conditions</i>	IdDataFrame [IdDataFrame ["NomColonne"] OperateurComparaison Valeur] IdDataFrame [(IdDataFrame ["NomColonne"] OperateurComp Valeur) OperateurLogique (IdDataFrame ["NomColonne"] OperateurComp Valeur)]
<i>Tri des éléments d'un DataFrame</i>	DataFrame_Trié = IdDataFrame.sort_values (by = ["NomColonne1" , "NomColonne2" , ...] , ascending = [True/False , True/False , ...])
<i>Création d'un graphique</i>	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 ()