

RÉPUBLIQUE TUNISIENNE ----- MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT SESSION 2026	
	ÉPREUVE PRATIQUE D'INFORMATIQUE	
	Section : Économie et Gestion	
	Coefficient de l'épreuve : 0.5	Durée : 1 heure

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

Important : Dans le répertoire **Bac2026** 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 à ce sujet.

La compagnie des phosphates de Gafsa est une entreprise publique tunisienne leader dans l'extraction et la transformation du phosphate. Afin de gérer les données relatives à la production des mines, le responsable du service informatique utilise une base de données simplifiée intitulée "**BD_Mines**" décrite par la représentation textuelle suivante :

MINE (IdMine, NomMine, Localisation, CapaciteProd)

PRODUIT (IdProd, LibProd, QualiteProd)

PRODUCTION (IdProd#, IdMine#, DateProd, QteProduite)

Soit la description des colonnes des tables de la base de données "**BD_Mines**" :

Nom	Description	Type	Taille	Format	Contrainte
IdMine	Identifiant d'une mine.	Texte court	4		
NomMine	Nom d'une mine.	Texte court	40		
Localisation	Lieu d'une mine.	Texte court	30		
CapaciteProd	Capacité de production d'une mine exprimée en million de Tonnes.	Numérique	Entier		Supérieure ou égale à zéro
IdProd	Identifiant d'un produit.	Texte court	4		
LibProd	Libellé d'un produit.	Texte court	30		
QualiteProd	Qualité d'un produit.	Texte court	45		
DateProd	Date de production d'un produit.	Date/Heure		Date, abrégé	Null interdit
QteProduite	Quantité produite d'un produit exprimée en Tonnes.	Numérique	Réel simple	Fixe ; 3 chiffres après la virgule	Supérieure à zéro

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 "**BD_Mines**".
- 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 :

Table : MINE			
IdMine	NomMine	Localisation	CapaciteProd
M001	Redeyef	Gafsa	10
M002	Fej Lahdoum	Siliana	5
M003	Metlaoui	Gafsa	4
M004	Zriba	Zaghouan	10

Table : PRODUIT		
IdProd	LibProd	QualiteProd
P001	Zinc	20%Zn
P002	Phosphate brut	SSP
P003	Plomb	5.7%Pb
P004	Phosphate tricalcique	DAP

Table : PRODUCTION			
IdProd	IdMine	DateProd	QteProduite
P001	M002	20/04/2018	5240.325
P002	M001	10/09/2019	2780.500
P003	M003	25/08/2020	8254.320
P004	M001	24/12/2020	4788.600
P001	M004	22/07/2019	6200.400

4) Créer les requêtes suivantes :

R1 : Afficher pour chaque production, l'identifiant du produit, l'identifiant de la mine, la date de production, le nom de la mine et sa localisation ainsi que le libellé du produit et la quantité produite.

R2 : Afficher, pour chaque produit de la famille **Phosphate**, le nom de la mine source de sa production ainsi que la date et la quantité produite.

R3 : Afficher les noms des mines qui produisent plus que 6000 tonnes de Zinc.

R4 : Augmenter de 10% la quantité produite de Zinc pour la mine de Zriba.

5) Exporter, dans votre dossier de travail et sous le nom "**ExtraMines.csv**", le résultat de la requête **R1** 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 "**ProdMines**" dans votre dossier de travail.
- 2) En se référant à l'annexe, utiliser les méthodes et les propriétés adéquates, pour écrire dans le fichier **ProdMines** les scripts permettant :
 - a) d'importer le contenu du fichier **ExtraMines.csv** et de le stocker dans un DataFrame à nommer "**mines**",
 - b) d'afficher le DataFrame **mines**,
 - c) d'afficher la quantité totale produite par mine,
 - d) d'afficher le nombre de produits ayant une quantité produite supérieure à 5000 Tonnes,
 - e) d'afficher, sans redondance, les libellés des produits extraits dans les différentes mines,
 - f) de trier le DataFrame **mines** dans un nouveau DataFrame à nommer "**minestri**" par ordre alphabétique des noms des mines puis par ordre décroissant des quantités produites et 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	2.25	6	1.25	0.5	1.25	0.25	1.25	1	1	1.75

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> <code>IdDataFrame.groupby (["Id_Colonne "]).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>