

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.

Une huilerie réceptionne des lots d'olives pour en extraire de l'huile. Afin de gérer le processus d'extraction, le responsable du service informatique utilise une base de données simplifiée intitulée "**BD_Extra_Huile**" décrite par la représentation textuelle suivante :

VAROLIVE (IdVarOli, NomVarOli)

LOT (IdLot, PoidsLot, IdVarOli#)

HUILE (IdHuile, TypeH, AciditeH, PrixH)

EXTRACTION (IdLot#, DateExtract, HeureExtract, QteProduite, IdHuile#)

Sachant qu'un lot d'olives est constitué d'une seule variété d'olive.

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

Nom	Description	Type	Taille	Format	Contrainte
IdVarOli	Identifiant d'une variété d'olive.	Texte court	2		
NomVarOli	Nom d'une variété d'olive.	Texte court	20		Null Interdit
IdLot	Identifiant d'un lot d'olives.	Texte court	4		
PoidsLot	Poids d'un lot d'olives en Kg.	Numérique	Entier		Supérieur ou égal à 200
IdHuile	Identifiant d'une huile d'olive.	Texte court	4		
TypeH	Type d'une huile d'olive.	Texte court	50		Null Interdit
AciditeH	Taux d'acidité d'une huile d'olive.	Numérique	Réel simple		
PrixH	Prix d'un Kg d'huile d'olive en dinar.	Numérique	Réel simple		
DateExtract	Date d'extraction de l'huile d'un lot d'olives.	Date/Heure		Date, abrégé	
HeureExtract	Heure d'extraction de l'huile d'un lot d'olives.	Date/Heure		Heure, abrégé	
QteProduite	Quantité d'huile produite par lot d'olives exprimée en Kg.	Numérique	Réel simple		Supérieure ou égale à 20

A) À l'aide du logiciel de gestion de base de données disponible :

- Créer, dans votre dossier de travail, la base de données à nommer "**BD_Extra_Huile**".
- 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.
- Remplir les différentes tables par les données représentées dans les tableaux suivants :

Table : VAROLIVE	
IdVarOli	NomVarOli
O1	Chetoui
O2	Chemlali
O3	Oueslati

Table : LOT		
IdLot	PoidsLot	IdVarOli
L101	500	O1
L102	750	O2
L103	800	O3
L104	300	O1

Table : HUILE			
IdHuile	TypeH	AciditeH	PrixH
H001	Extra Vierge	0.3	20.0
H002	Vierge	0.8	16.0
H003	Extra Vierge	0.2	22.0
H004	Raffiné	0.3	14.0

Table : EXTRACTION				
IdLot	DateExtract	HeureExtract	QteProduite	IdHuile
L101	01/11/2025	8:00	190	H001
L104	07/11/2025	12:00	70	H003
L102	02/11/2025	15:00	100	H004
L102	03/11/2025	15:00	80	H004
L103	04/11/2025	8:00	120.5	H002
L101	01/11/2025	14:00	50	H001

4) Créer les requêtes suivantes :

R1 : Afficher les extractions d'huile réalisées (**IdLot**, **NomVarOli**, **DateExtract**, **HeureExtract**, **TypeH**, **QteProduite**, **PrixH**) par ordre croissant selon la date puis l'heure d'extraction.

R2 : Afficher, pour un identifiant d'une variété d'olive donné, la liste des identifiants des lots et leurs poids.

R3 : Augmenter de **10%** le prix d'un Kg d'huile de type "**Extra Vierge**".

5) Exporter, dans votre dossier de travail et sous le nom "**Huile.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 "**Extraction**" dans votre dossier de travail.
- 2) En se référant à l'annexe, utiliser les méthodes et les propriétés adéquates, pour écrire dans le fichier **Extraction** les scripts permettant :
 - a) d'importer le contenu du fichier **Huile.csv** et de le stocker dans un DataFrame à nommer "**hui**",
 - b) d'afficher le DataFrame **hui**,
 - c) de calculer, dans une nouvelle colonne à nommer "**Montant**", la valeur monétaire de la quantité d'huile produite, sachant que : **Montant = QteProduite * PrixH**
 - d) d'afficher, les lots d'olives et leurs variétés qui ont donné des huiles de type "**Extra Vierge**",
 - e) de trier, dans un nouveau DataFrame à nommer "**huietri**", les données du DataFrame **hui** par ordre croissant du type d'huile puis par ordre croissant du montant et d'afficher le DataFrame trié,
 - f) de supprimer, du DataFrame **huietri** les lignes numéros 2 et 3 et de l'afficher.

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	0.25	1	1.25	1.75	1

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>