ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR
Membre de HONORIS UNITED UNIVERSITIES

SUJET D'EXAMEN BLANC

MATIERE : ARCHITECTURE DES DONNÉES

NIVEAU :	**4IA-DATA**	**OPTION :**	**AI-DATA**	**SEMESTRE :**	**S8**
SESSION :	**Normale**	**DUREE :**	**2H**	**CONSIGNES :**	**Documents Non autorisés**
ANNEE UNIVERSITAIRE 2025-2026			**ENSEIGNANT (E)**	**Pr. Dr. Abdelali EL Gourari**	

!!! TOUS LES EQUIPEMENTS SMART DOIVENT ETRE ETEINTS ET RANGES !!!

EXERCICE 1 : Fondamentaux Python & Environnement Data (5 points)

Partie A - Concepts et Environnement (2,5 points)

1. **[0,5 pt]** Expliquez la différence entre pandas et polars en termes de performance et de cas d'usage. Dans quel scénario privilégieriez-vous Polars ?
2. **[0,5 pt]** Pourquoi l'utilisation d'environnements isolés (Conda, Docker) est-elle indispensable dans un projet Data Science professionnel ? Citez deux avantages concrets.
3. **[0,5 pt]** Quelle est la différence entre df.isnull().sum() et df.dropna() dans Pandas ? Dans quel contexte utiliseriez-vous chacune de ces méthodes ?
4. **[0,5 pt]** Expliquez l'intérêt du format **Parquet** par rapport au CSV pour le stockage de données dans un pipeline Big Data. Citez deux arguments techniques.
5. **[0,5 pt]** Que signifie le principe de *"Lazy Evaluation"* dans Polars ? Quel bénéfice apporte-t-il pour le traitement de gros volumes ?

Partie B - Cas pratique : Pipeline de nettoyage (2,5 points)

Un analyste reçoit un fichier transactions.csv contenant des incohérences. Voici un extrait du code de traitement :

```
import pandas as pd
import numpy as np
df = pd.read_csv("transactions.csv")
# Détection des valeurs problématiques
print(df.isnull().sum())
print(df[df["montant"] < 0])
# Nettoyage
df_clean = df[df["montant"] >= 0].copy()
df_clean["categorie"] = df_clean["categorie"].fillna("Non classé")
df_clean["montant_ttc"] = df_clean["montant"] * 1.20
# Export
df_clean.to_parquet("transactions_nettoye.parquet", index=False)
```

Questions :

1. **[1 pt]** Expliquez ligne par ligne les opérations de filtrage et d'imputation réalisées. Pourquoi utilise-t-on .copy() après le filtrage ?
2. **[1 pt]** Proposez deux vérifications supplémentaires de qualité des données que vous ajouteriez avant l'export. Justifiez chaque proposition.
3. **[0,5 pt]** Pourquoi exporter en Parquet plutôt qu'en CSV dans ce contexte professionnel ?

EXERCICE 2 : Architectures de Données & Pipelines ETL/ELT (5 points)

Partie A - Typologie et Stockage Moderne (2,5 points)

1. **[0,5 pt]** Définissez clairement les concepts de **Data Lake**, **Data Warehouse** et **Lakehouse**. Quel problème spécifique le Lakehouse résout-il pour les projets d'IA ?
2. **[0,5 pt]** Expliquez le principe de l'architecture **Bronze → Silver → Gold**. Quel type de transformation est appliqué à chaque couche ?
3. **[0,5 pt]** Qu'apporte le format **Delta Lake** par rapport à un stockage objet classique (S3) ? Citez deux fonctionnalités critiques pour l'IA.
4. **[0,5 pt]** Dans un contexte Big Data, pourquoi privilégie-t-on souvent l'approche **ELT** plutôt qu'**ETL** ? Donnez un argument technique et un argument métier.
5. **[0,5 pt]** Quel est le rôle du ***Time Travel*** dans Delta Lake ? Donnez un cas d'usage concret en environnement de production ML.

Partie B - Étude de cas : Pipeline hybride (2,5 points)

Une entreprise souhaite construire un pipeline pour alimenter un modèle de recommandation produit. Les sources sont :

- Base CRM (données clients structurées, sensibles RGPD)
- Logs d'application (JSON semi-structurés, volumineux)
- Avis clients en texte libre (non structurés)

Questions :

1. **[1 pt]** Proposez une architecture de stockage (Bronze/Silver/Gold) pour ce cas. Précisez le format de fichier et le type de traitement à chaque étape.
2. **[1 pt]** Pour l'anonymisation des données clients (conformité RGPD), à quelle étape du pipeline devez-vous l'appliquer ? Justifiez votre choix entre ETL et ELT.
3. **[0,5 pt]** Comment prépareriez-vous les avis clients textuels pour qu'ils soient exploitables par un modèle de Machine Learning ? Décrivez brièvement les étapes de prétraitement.

EXERCICE 3 : Analyse Exploratoire avec Pandas (5 points)

Un fichier ventes_maroc.csv contient les données suivantes :

commande_id	produit	categorie	prix_unitaire	quantite	ville	date
2001	Laptop	Informatique	8999.99	2	Casablanca	2024-01-15
2002	Souris	Accessoires	299.99	10	Rabat	2024-01-16
2003	Clavier	Informatique	799.99	NaN	Tanger	2024-01-16
2004	Écran	Informatique	-1999.99	1	Fès	2024-01-17

```
import pandas as pd
df = pd.read_csv("ventes_maroc.csv")
# Nettoyage initial
df = df[df["prix_unitaire"] > 0].copy()
df["quantite"] = df["quantite"].fillna(df["quantite"].median())
df["montant_ligne"] = df["prix_unitaire"] * df["quantite"]
# Analyse
ca_par_ville = df.groupby("ville")["montant_ligne"].sum()
top_produits = df.groupby("produit")["quantite"].sum().sort_values(ascending=False)
```

Questions :

1. **[1 pt]** Expliquez pourquoi on filtre d'abord sur prix_unitaire > 0 avant de remplir les valeurs manquantes de quantite.
2. **[1 pt]** Que retourne l'expression df.groupby("ville")["montant_ligne"].sum() ? Décrivez la structure du résultat.
3. **[1 pt]** Proposez une instruction Pandas pour identifier les commandes dont le montant_ligne dépasse 5000 DH.
4. **[1 pt]** Comment calculeriez-vous le prix moyen par catégorie, en excluant les valeurs aberrantes (prix > 90e percentile) ? Écrivez le code correspondant.
5. **[1 pt]** Pourquoi est-il important de documenter chaque étape de transformation dans un notebook Jupyter ? Citez deux bonnes pratiques de traçabilité.

EXERCICE 4 : Architecture RAG — Questions Ouvertes (5 points)

1. **[1,5 pt]** Expliquez, avec vos propres mots, pourquoi un système RAG (Retrieval Augmented Generation) est souvent préféré à un LLM seul pour des applications professionnelles. Illustrez votre réponse par **trois limitations des LLM** que le RAG permet de contourner.
2. **[1,5 pt]** Décrivez les quatre étapes principales d'un pipeline RAG : *Ingestion* → *Embedding* → *Retrieval* → *Génération*. Pour chaque étape, précisez :
   - L'objectif technique

- Un outil ou bibliothèque Python couramment utilisé
- Un défi potentiel et une piste de solution

3. **[2 pt]** Qu'est-ce que le *chunking* et pourquoi est-il critique dans un système RAG ? Comparez brièvement deux stratégies de découpage (ex: taille fixe vs récursif) et indiquez dans quel contexte vous privilégieriez chacune.