**Examen : Gestion d’un Centre de Formation – Base de Données SQL & NoSQL**

**Durée :** 2h | **Total :** 30 points

**🧩 Dossier 1 : MySQL / SQL (16 points)**

**Structure relationnelle :**

* **Apprenants** (idApprenant, nom, prenom, dateNaissance, email, niveau, groupe)
* **Formations** (idFormation, intitule, domaine, duree, niveau, formateur#)
* **Inscriptions** (idInscription, idApprenant#, idFormation#, dateInscription, statut)
* **Formateurs** (idFormateur, nom, prenom, specialite, contact)

**Questions SQL :**

1. **Ajouter les contraintes suivantes :** (2 pts)
   * Le champ niveau dans Apprenants ne peut être que : "Débutant", "Intermédiaire", "Avancé"
   * Le champ statut dans Inscriptions doit être : "Validée", "En attente", "Annulée"
2. **Écrire une procédure stockée qui affiche tous les apprenants inscrits dans une formation donnée (passée en paramètre) avec leur nom, prénom et date d’inscription.** (3 pts)
3. **Écrire une procédure stockée qui affiche la liste des formations qui n’ont encore aucun inscrit.** (2 pts)
4. **Créer un trigger qui empêche l’inscription d’un apprenant à une formation s’il est déjà inscrit (vérification de doublon).** (4 pts)
5. **Créer une fonction qui retourne le nombre total d’apprenants inscrits dans un groupe donné.** (2 pts)
6. **Créer un rôle Gestionnaire avec tous les droits sur les tables Apprenants et Inscriptions, et affecter l’utilisateur admin\_centre à ce rôle.** (3 pts)

**🗃️ Dossier 2 : MongoDB (14 points)**

**Collections :**

**Apprenants**

{

"\_id": ObjectId,

"nom": "Zahra",

"prenom": "Younes",

"groupe": "G1",

"niveau": "Intermédiaire",

"inscriptions": [

{

"formation": "Développement Web",

"dateInscription": ISODate("2025-02-15"),

"statut": "Validée"

}

]

}

**Formateurs**

{

"idFormateur": "F123",

"nom": "Hassan",

"prenom": "Salma",

"specialite": "Base de données",

"dateRecrutement": ISODate("2021-06-01"),

"salaire": 7000

}

**Questions MongoDB :**

a. Ajouter un nouvel apprenant avec deux inscriptions à des formations. (1 pt)  
b. Afficher les noms des apprenants du groupe "G2", triés par nom. (1 pt)  
c. Afficher les inscriptions effectuées entre le 1er février et le 31 mars 2025. (2 pts)  
d. Afficher pour chaque groupe le nombre d’apprenants inscrits. (2 pts)  
e. Supprimer tous les apprenants ayant le niveau "Débutant". (2 pts)  
f. Afficher les inscriptions validées à la formation "Cybersécurité". (2 pts)  
g. Augmenter de 10% les salaires des formateurs recrutés avant 2023. (2 pts)  
h. Écrire un script Python avec PyMongo qui affiche tous les apprenants avec la liste des formations auxquelles ils sont inscrits. (2 pts)

CREATE ROLE Gestionnaire;

GRANT SELECT, INSERT, UPDATE, DELETE ON Apprenants TO Gestionnaire;

GRANT SELECT, INSERT, UPDATE, DELETE ON Inscriptions TO Gestionnaire;

CREATE USER 'admin\_centre'@'localhost' IDENTIFIED BY 'password123';

GRANT Gestionnaire TO 'admin\_centre'@'localhost';

✅ Corrigé – Dossier MongoDB

a. Ajouter un apprenant

js

Copier

Modifier

db.Apprenants.insertOne({

nom: "Nour",

prenom: "Karim",

groupe: "G3",

niveau: "Avancé",

inscriptions: [

{

formation: "Python Avancé",

dateInscription: ISODate("2025-04-01"),

statut: "Validée"

},

{

formation: "Sécurité Réseau",

dateInscription: ISODate("2025-04-03"),

statut: "En attente"

}

]

})

b. Apprenants du groupe G2 triés

js

Copier

Modifier

db.Apprenants.find({ groupe: "G2" }, { nom: 1, \_id: 0 }).sort({ nom: 1 })

c. Inscriptions entre deux dates

js

Copier

Modifier

db.Apprenants.aggregate([

{ $unwind: "$inscriptions" },

{ $match: {

"inscriptions.dateInscription": {

$gte: ISODate("2025-02-01"),

$lte: ISODate("2025-03-31")

}

}},

{ $project: {

nom: 1, prenom: 1, formation: "$inscriptions.formation", dateInscription: "$inscriptions.dateInscription"

}}

])

d. Nombre d’apprenants par groupe

js

Copier

Modifier

db.Apprenants.aggregate([

{ $group: { \_id: "$groupe", total: { $sum: 1 } } }

])

e. Supprimer débutants

js

Copier

Modifier

db.Apprenants.deleteMany({ niveau: "Débutant" })

f. Inscriptions validées à la formation "Cybersécurité"

js

Copier

Modifier

db.Apprenants.aggregate([

{ $unwind: "$inscriptions" },

{ $match: {

"inscriptions.formation": "Cybersécurité",

"inscriptions.statut": "Validée"

}},

{ $project: {

nom: 1, prenom: 1, dateInscription: "$inscriptions.dateInscription"

}}

])

g. Augmentation des salaires

js

Copier

Modifier

db.Formateurs.updateMany(

{ dateRecrutement: { $lt: ISODate("2023-01-01") } },

{ $mul: { salaire: 1.10 } }

)

🐍 Script Python avec PyMongo (h)

python

Copier

Modifier

from pymongo import MongoClient

client = MongoClient("mongodb://localhost:27017/")

db = client["CentreFormation"]

apprenants = db.Apprenants.find()

for apprenant in apprenants:

print(f"Apprenant : {apprenant['nom']} {apprenant['prenom']}")

for inscription in apprenant.get("inscriptions", []):

print(f" ➤ Formation : {inscription['formation']} - Statut : {inscription['statut']}")

📄 Formats téléchargeables