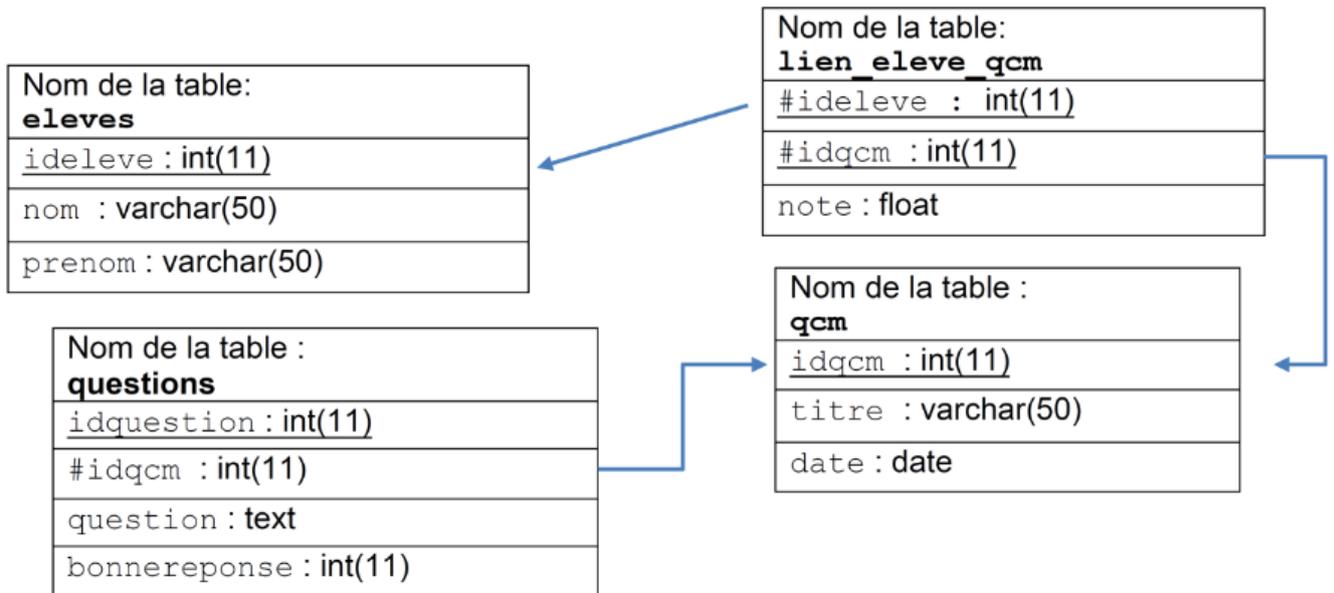


Les 2 exercices sont indépendants. Le premier porte sur les bases de données. Le second sur la programmation objet.

## 1- BASE DE DONNEES POUR UN QCM EN LIGNE :

Un enseignant a mis en place un site web qui permet à ses élèves de faire des QCM de NSI en ligne.

L'enseignant a créé une base de données nommée QCM\_NSI pour gérer ses QCM, contenant les quatre relations (appelé aussi communément "table") du schéma relationnel ci-dessous :



Dans le cas de la relation lien\_eleve\_qcm la clé primaire est composée de l'association des deux attributs *ideleve* et *idqcm*.

On donne ci-dessous le contenu exhaustif des relations *eleves*, *lien\_eleve\_qcm* et *qcm* :

**Table eleves**

ideleve	nom	prenom
2	Dubois	Thomas
3	Dupont	Cassandra
4	Marty	Mael
5	Bikila	Abebe

**Table qcm**

idqcm	titre	date
1	Base de données	2021-09-20
2	POO	2022-04-08
3	Arbre Binaire	2022-01-09
4	Arbre Parcours	2022-02-15
5	Piles-Files	2021-12-05

**Table lien\_eleve\_qcm**

ideleve	idqcm	note
2	1	12
2	3	18
2	4	13
2	5	15
3	1	20
3	2	9
3	3	18
3	5	13
4	4	15
4	5	20
5	4	15

Question 1. : Ecrire la requête SQL qui donne les notes de l'élève qui a pour identifiant 4 :

```
-- Question 1 :  
SELECT note FROM lien_eleve_qcm  
WHERE ideleve = 4;
```

Question 2. : Sachant que la clé primaire de la relation lien\_eleve\_qcm est composée de l'association des deux attributs ideleve et idqcm, expliquer pourquoi avec ce schéma relationnel, un élève ne peut pas faire deux fois le même QCM ?

Chaque clé primaire doit être **unique**. Ainsi il ne peut y avoir qu'un seul couple (ideleve , idqcm). Un élève ne peut donc pas faire deux fois le même qcm.

Question 3. : Un nouvel élève (nom : Lefevre, prénom : Kevin) est enregistré. Ecrire la requête permettant de l'insérer dans la bdd :

```
-- Question 3 :  
INSERT INTO eleves VALUES (6,'Lefevre','Kevin');  
  
INSERT INTO eleves(ideleves,nom,prenom) VALUES (6,'Lefevre','Kevin');
```

Question 4. : L'élève Dubois Thomas quitte l'établissement et toutes les références à cet élève doivent être supprimées des relations. Pour la relation lien\_eleve\_qcm, écrire la requête pour supprimer toutes les références à l'élève Dubois Thomas qui a pour identifiant 2 :

```
-- Question 4 :  
DELETE FROM lien_eleve_qcm WHERE ideleve = 2;
```

Question 5. : La date du qcm dont le titre est 'Piles-Files' doit être modifiée : '2024-11-05' au lieu de '2021-12-05'. Ecrire la requête SQL qui permet de réaliser cette mise à jour :

```
-- Question 5 :  
UPDATE qcm SET date = '2024-11-05' WHERE titre = 'Piles-Files';
```

Question 6. : Donner la requête SQL qui permet d'obtenir la moyenne des notes pour le qcm dont l'idqcm est égal à 5. On désire que l'affichage du résultat de cette requête sur le logiciel DB Browser soit celui donné ci-contre :

	noteMoy
1	14.3

```
-- Question 6 :  
SELECT AVG(note) AS noteMoy FROM lien_eleve_qcm  
WHERE idqcm = 5;
```

Question 7. : Donner la requête qui renvoie la liste des noms et prénoms des élèves ayant fait le QCM d'idqcm égal à 4 :

```
-- Question 7 :  
SELECT nom,prenom FROM eleves AS e  
JOIN lien_eleve_qcm AS lien ON e.ideleve = lien.ideleve  
WHERE lien.idqcm = 4;
```

Question 8. : Donner la requête qui renvoie le nom, le prénom et la note des élèves ayant fait le QCM *Arbre Parcours* :

```
-- Question 8 :
SELECT e.nom , e.prenom , lien.note FROM eleves AS e
JOIN lien_eleve_qcm AS lien ON e.ideleve = lien.ideleve
JOIN qcm ON lien.idqcm = qcm.idqcm
WHERE qcm.titre = 'Arbre Parcours':
```

## 2- POO POUR GERER UNE LIBRAIRIE :

On se propose de gérer les livres d'une bibliothèque, à l'aide de deux classes, la classe *Livre* et la classe *Bibliotheque*.

### a. CLASSE LIVRE :

On se propose de créer une classe nommée *Livre*.

Cette classe possèdera 1 attribut de classe nommé nbLivres dont la valeur est égale au nombre d'instances créées (*int*).

Les attributs d'instances des objets créés seront :

- titre : titre du livre (*string*)
- auteur : nom de l'auteur (*string*)
- etat : nombre entier (*int*) compris entre 0 et 5. Si l'état vaut 0, le livre est trop abîmé et ne pourra plus être prêté. L'état d'un livre ne peut pas être négatif.

Cette classe comprend une méthode *degrade()*. Elle permet de baisser de 1 la valeur de l'attribut *etat* d'un livre.

En exécutant le programme principal donné ci-contre, on retrouve l'affichage donné ci-après :

```
# Programme principal
livre1 = Livre("Les Misérables", "Victor Hugo")
livre2 = Livre("Les fleurs du mal", "Charles Baudelaire", 1)
print(livre1)
print(livre2)
livre2.degrade()
livre2.degrade()
print(livre2)
```

```
>>> (executing file "pooBibliotheque.py")
Titre : Les Misérables
Auteur : Victor Hugo
Etat : 5

Titre : Les fleurs du mal
Auteur : Charles Baudelaire
Etat : 1

Titre : Les fleurs du mal
Auteur : Charles Baudelaire
Etat : 0
```

En exécutant ensuite dans la console, la commande `>>> Livre.nbLivres` , celle-ci retourne 2 .

Question 1 : Ecrire le début du script de la classe *Livre* qui permet de définir l'attribut de classe et ceux d'instance à travers la méthode constructeur : `__init__()`

Question 2 : Ecrire le script de la méthode : `__str__()` (info : `\n` placé dans un string permet d'obtenir un retour à la ligne lorsque ce string est affiché avec la fonction `print()`)

Question 3 : Ecrire le script de la méthode : `degrade()`

```
class Livre():
    nbLivres = 0

    def __init__(self, titre, auteur, etat = 5):
        self.titre = titre
        self.auteur = auteur
        self.etat = etat
        Livre.nbLivres = Livre.nbLivres + 1

    def __str__(self):
        s = f"Titre : {self.titre} \nAuteur : {self.auteur} \nEtat : {self.etat}\n"
        return s

    def degrade(self):
        if self.etat > 0:
            self.etat = self.etat - 1
```

**CORRIGE**

## b. CLASSE BIBLIOTHEQUE :

On se propose de créer une classe nommée *Bibliothèque*. Elle utilise les ressources de la classe *Livre* mise au point dans le paragraphe précédent.

En exécutant le programme principal donné ci-contre, on retrouve l'affichage donné ci-dessous :

```
# Programme principal
livre1 = Livre("Les Misérables", "Victor Hugo")
livre2 = Livre("Les fleurs du mal", "Charles Baudelaire", 1)

ma_bibli = Bibliotheque()
ma_bibli.ajoute(livre1)
ma_bibli.ajoute(livre2)
ma_bibli.inventaire()
```

```
>>> (executing file "pooBibliotheque.py")
-----
contenu de ma bibliothèque :
-----
Titre : Les Misérables
Auteur : Victor Hugo
Etat : 5

Titre : Les fleurs du mal
Auteur : Charles Baudelaire
Etat : 1
```

Question 4 : Ecrire le script de la classe *Bibliothèque*

```
class Bibliothèque:
    def __init__(self):
        self.l = []

    def ajoute(self, livre):
        self.l.append(livre)

    def inventaire(self):
        print("-----")
        print("contenu de ma bibliothèque :")
        print("-----")
        for livre in self.l:
            print(livre , "\n")
```

**CORRIGE**