

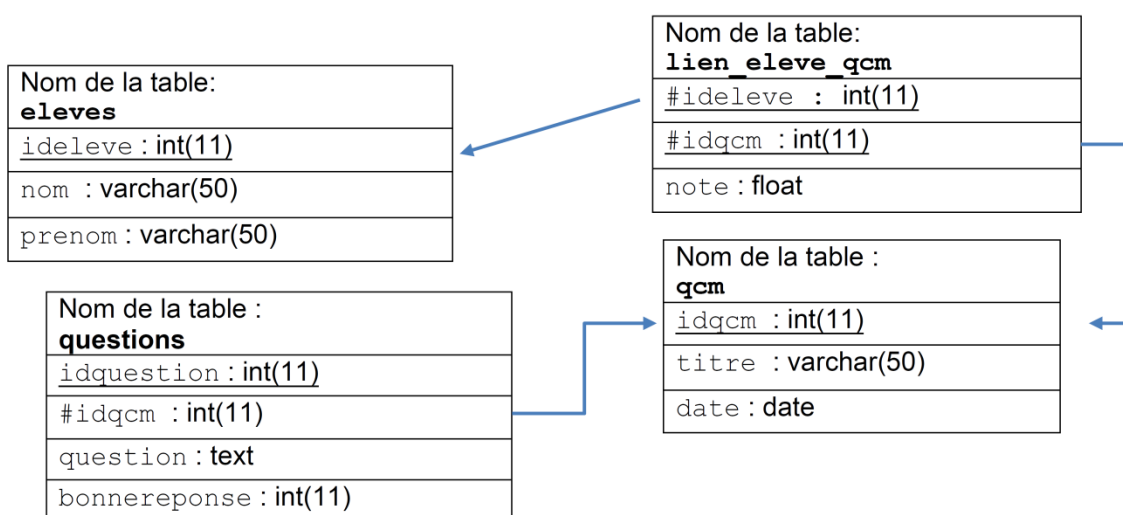
EXERCICE 3 (4 points)

Cet exercice porte sur les bases de données (bases de données relationnelles, langage SQL).

Un rappel sur la syntaxe de quelques fonctions SQL est donné en annexe 1 en fin de sujet.

Un enseignant a mis en place un site web qui permet à ses élèves de faire des QCM (questionnaire à choix multiples) 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 schéma relationnel précédent, un attribut souligné indique qu'il s'agit d'une clé primaire. Un attribut précédé du symbole # indique qu'il s'agit d'une clé étrangère et la flèche associée indique l'attribut référencé. Ainsi, par exemple, l'attribut ideleve de la relation `lien_eleve_qcm` est une clé étrangère qui fait référence à l'attribut ideleve de la relation `eleves`.

Dans le cas de la relation `lien_eleve_qcm` la clé primaire est composée de l'association des deux attributs ideleve et idqcm, eux-mêmes étant des clés étrangères.

On donne ci-dessous le contenu exhaustif des relations :

Table eleves

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

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

Table qcm

idqcm	titre	Datecreation
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

1.

- a. Que retourne la requête suivante ?

```
SELECT titre FROM `qcm` WHERE date > '2022-01-10' ;
```

- b. Ecrire une requête qui donne les notes de l'élève qui a pour identifiant 4.

2.

- a. 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.
- b. L'élève *Marty Mael* vient de faire le QCM sur la POO et a obtenu une note de 18.

Comment est/sont modifiée(s) le(s) table(s) ? Il n'est pas demandé d'écrire une requête SQL.

- c. Un nouvel élève (nom : *Lefèvre*, prenom : *Kevin*) est enregistré. Ecrire la requête permettant la mise à jour du/des relation(s).
- d. 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.

3.

- a. Recopier et compléter les de la requête suivante pour qu'elle affiche la liste des noms et prénoms des élèves ayant fait le QCM d'idqcm égal à 4.

```
SELECT  FROM eleves
JOIN lien_eleve_qcm ON eleves.ideleve = 
WHERE  ;
```

- b. Donner le résultat de la requête de la question a.

4. Ecrire une requête qui affiche le nom, le prénom et la note des élèves ayant fait le QCM Arbre Binaire. L'utilisation des trois tables dans la requête est attendue.

ANNEXE 1 – LANGAGE SQL

- **Types de données**

CHAR(t) VARCHAR(t) TEXT	Texte fixe de t caractères. Texte de t caractères variables. Texte de 65 535 caractères max.
INT	<i>Nombre entier de -2^{31} à $2^{31}-1$ (signé) ou de 0 à $2^{32}-1$ (non signé)</i>
FLOAT	Réel à virgule flottante
DATE DATETIME	Date format AAAA-MM-JJ Date et heure format AAAA-MM-JJHH:MI:SS

- **Quelques exemples de syntaxe SQL :**

- Insérer des enregistrements :

```
INSERT INTO Table (attribut1, attribut2) VALUES(valeur1 , valeur2)
```

- Modifier des enregistrements :

```
UPDATE Table SET attribut1=valeur1, attribut2=valeur2 WHERE Selecteur
```

- Supprimer des enregistrements :

```
DELETE FROM Table WHERE Selecteur
```

- Sélectionner des enregistrements :

```
SELECT attributs FROM Table WHERE Selecteur
```

- Effectuer une jointure :

```
SELECT attributs FROM TableA JOIN TableB ON TableA.cle1=TableB.cle2 WHERE  
Selecteur
```