ET_1_Ex4_Corrigé

Exercice 4 : Bases de données relationnelles et le langage SQL

L'énoncé de cet exercice utilise les mots du langage SQL suivant : SELECT, FROM, WHERE, JOIN, INSERT INTO, VALUES, COUNT, ORDER BY.
Pour les besoins de l'organisation du lycée, le chef d'établissement exploite la base de données par des requêtes en langage SQL. Il a pour cela créé une table (ou relation) SQL dénommée seconde dans son système de gestion de bases de données dont la structure est la suivante :

seconde	Type
num_eleve (clef primaire)	entier (??)
langue1	CHAR
langue2	CHAR
option	CHAR
classe	CHAR

Question 1

1. **Dans le modèle relationnel, quel est l'intérêt de l'attribut num_eleve.**
   La clé primaire d'une relation est un attribut qui permet de désigner d'une façon unique un uplet.
   Par exemple l'attribut *num_eleve* permet d'identifier de façon unique les uplets de la table (ou relation) *seconde*. La seule connaissance de la clé primaire permet d'identifier toute ligne de la table.
   Cet identifiant (souvent auto géré) unique est ici associé à un élève, par essence unique dans l'établissement.
2. **Écrire une requête SQL d'insertion permettant d'enregistrer l'élève ACHIR Mussa dans la table seconde. Les informations relatives à cet élève sont données dans la ligne 1 du fichier seconde_lyc.csv.**

num_eleve	nom	prenom	datenaissance	langue1	langue2	option	classe
133310FE	ACHIR	Mussa	01/01/2005	anglais	espagnol		2A
156929JJ	ALTMEYER	Yohan	05/05/2005	allemand	anglais	théâtre	2D

<u>Erreur d'énoncé</u> : l'attribut *num_eleve* n'est pas un entier si on regarde bien ! On choisit ici de n'écrire que les caractères numériques par cohérence. Cela ne serait évidemment pas sanctionné sur la copie d'un élève.

3. **Lors de l'insertion de l'élève ALTMEYER Yohan (ligne 2 du fichier seconde_lyc.csv), une erreur de saisie a été commise sur la première langue, qui devrait être allemand. Écrire une requête SQL de mise à jour corrigeant les données de cet élève.**

```sql
UPDATE seconde
SET langue1 = 'allemand'
WHERE num_eleve = 156929 ;
```

Question 2

On suppose maintenant que la table seconde contient les informations issues de la figure 1 (ni plus, ni moins, même si la figure 1 n'est qu'un extrait du fichier seconde_lyc.csv).

1. **Quel est le résultat de la requête SELECT num_eleve FROM seconde;?**

```
SELECT num_eleve FROM seconde ;
```

Le résultat de la requête sera la colonne des clefs primaires *num_eleve* :

num_eleve (clef primaire)
133310
156929
...
666702

2. **On rappelle qu'en SQL, la fonction d'agrégation COUNT() permet de compter le nombre d'enregistrements dans une table. Quel est le résultat de la requête SELECT COUNT(num_eleve) FROM seconde;?**

```
SELECT COUNT(num_eleve) FROM seconde ;
```

Le résultat de la requête sera le nombre de lignes (de clefs primaires) de la table *seconde* donc ici 30.

3. **Écrire la requête permettant de connaître le nombre d'élèves qui font allemand en langue1 ou langue2.**

```
SELECT COUNT(num_eleve) FROM seconde
WHERE langue1='allemand' OR langue2='allemand' ;
```

Question 3

Le chef d'établissement souhaite faire évoluer la structure de sa base de données. Pour ce faire, il créé une nouvelle table *eleve* dont la structure est la suivante :

eleve	Type
num_eleve (clef primaire, clef étrangère de la table seconde)	entier (??)
nom	CHAR
prenom	CHAR
datenaissance	CHAR

Là encore, l'attribut num_eleve est un entier, les autres sont des chaînes de caractère (le type CHAR).

1. **Expliquer ce qu'apporte l'information clef étrangère pour l'attribut num_eleve de cette table en termes d'intégrité et de cohérence.**
   Les clés étrangères permettent de gérer des relations entre plusieurs tables, et garantissent la cohérence des données. On peut ainsi modifier des données d'un élève sans avoir à modifier plusieurs tables.

2. **On suppose la table eleve correctement créée et complétée. Le chef d'établissement aimerait lister les élèves (nom, prénom, date de naissance) de la classe 2A. Écrire la commande qui permet d'établir cette liste à l'aide d'une jointure entre *eleve* et *seconde*.**

seconde	Type
num_eleve (clef primaire)	entier (??)
langue1	CHAR
langue2	CHAR
option	CHAR
classe	CHAR

eleve	Type
num_eleve (clef primaire, clef étrangère de la table seconde)	entier (??)
nom	CHAR
prenom	CHAR
datenaissance	CHAR

```
SELECT nom, prenom ,datenaissance
        FROM eleve
        INNER JOIN seconde
        ON seconde.num_eleve = eleve.num_eleve
        WHERE seconde.classe='2A' ;
```

Question 4

Proposer la structure d'une table *coordonnees* dans laquelle on pourra indiquer, pour chaque élève, son adresse, son code postal, sa ville, son adresse mail. Préciser la clef primaire et/ou la clé étrangère en vue de la mise en relation avec les autres tables.

coordonnees	Type
num_eleve (clef primaire, clef étrangère de la table seconde)	entier (??)
adresse	CHAR ou TEXT (pour des données plus longues)
codepostal	INT
ville	CHAR
email	CHAR