Evaluation sur les SGBD et la POO

Exercice 1 : Base de données

Cet exercice porte sur les bases de données.

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

Les enseignants d'un établissement imaginaire proposent des parcours d'entraînement au numérique à leurs élèves en créant des séries d'exercices appelées **Evaluations**. Les différentes informations sont stockées dans une base de données.

Les informations de chaque campagne créée sont stockées dans la table `Evaluations` dont la structure est la suivante :

attribut	type
Code_evaluation	CHAR
Nom_evaluation	CHAR
Auteur	CHAR
Date	CHAR
Code_competences	INT

Un extrait de la table `Evaluations` est donné ci-dessous :

Code_evaluation	Nom_evaluation	Auteur	Date	Code_competences
EXKVLX886	Term7	Peltier	13/10/2021	1453
AZVBYB689	Groupe3	Lacour	07/10/2021	1276
PRJUYR491	Term5	Peltier	07/10/2021	1453
RTKVLX656	campagneSTMG	Beley	03/10/2021	476
DZLYYR479	Term5	Serhani	27/09/2021	1659
XJVBTX585	grNSI2	Eisen	24/09/2021	532
CRLYYR439	1ere6	Caille	13/09/2021	532
AZVBYB789	rentreeHGGSP	Martin	13/09/2021	386
OBJUYR491	Web_2nde	Boucher	07/09/2021	452
AGTBYB689	rechercheBTS	Beley	07/09/2021	1341
DQVBTX905	2nde2	Nguyen	07/09/2021	452

Tableau 1

1.

a. Dans la table `Evaluations`, quel est le seul attribut pouvant servir de clé primaire ? Justifier votre réponse.

b. Ecrire la requête SQL d'insertion qui a permis d'enregistrer la campagne **Term7** dans la table `Evaluations`. Les informations relatives à cette campagne sont données dans la première ligne du tableau 1 précédent.

2. On suppose maintenant que la table `Evaluations` contient **uniquement** les 11 enregistrements présentés en tableau 1.

   a. Combien de lignes s'affichent après l'exécution de la requête ?

   ```
   SELECT auteur FROM Evaluations
   ```

   b. Recopier les lignes issues de la requête :

   ```
   SELECT Nom_evaluation, Date FROM Evaluations WHERE
   auteur= "Peltier"
   ```

   c. Rédiger une requête permettant de connaître le nom des campagnes prévoyant un entraînement ciblé sur le web (`Code_competences` 452).

3. Le système de gestion de bases de données dispose également d'une table `resultats` dont la structure est la suivante :

attribut	type
Code_evaluation	CHAR
Num_eleve	INT
Score	INT

Si l'élève s'est connecté à la campagne mais n'a pas cliqué sur «envoyer les résultats», son score vaut -1.

   a. Qu'imposerait le choix du couple (`Code_evaluation, Num_eleve`) comme clé primaire pour la table `resultats` ?

   Un extrait de la relation est donné ci-dessous :

Code_evaluation	Num_eleve	Scores
PRJUYR491	17	300
CRLYYR439	654	-1
PRJUYR491	1454	220
RTKVLX656	554	255
DZLYYR479	17	-1
XJVBTX585	1664	12
CRLYYR439	18703	0
PRJUYR491	1565	422
XJVBTX585	12	643
CRLYYR439	168	19
DZLYYR479	17	140
XJVBTX585	1658	647

   b. Écrire une requête permettant d'obtenir les numéros des élèves (`Num_eleve`) qui ont travaillé la compétence `532`

4.

   a. Proposer la structure d'une table `eleves` permettant d'identifier les noms, prénoms et les classes des élèves.

   b. Proposer une clef primaire pour cette table.

ANNEXE 1 – LANGAGE SQL

- **Types de données**

CHAR	Chaîne de caractère
INT	*Nombre entier de* -2^{31} à $2^{31}-1$ (signé) ou de 0 à $2^{32}-1$ (non signé)
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
```

Exercice 2 : POO

Cet exercice porte sur les structures de données (programmation objet).

Simon souhaite créer en Python le jeu de cartes « la bataille » pour deux joueurs. Les questions qui suivent demandent de reprogrammer quelques fonctions du jeu. On rappelle ici les règles du jeu de la bataille :

Préparation

- Distribuer toutes les cartes aux deux joueurs.
- Les joueurs ne prennent pas connaissance de leurs cartes et les laissent en tas face cachée devant eux.

Déroulement

- A chaque tour, chaque joueur dévoile la carte du haut de son tas.
- Le joueur qui présente la carte ayant la plus haute valeur emporte les deux cartes qu'il place sous son tas.
- **Les valeurs des cartes sont :** dans l'ordre de la plus forte à la plus faible : As, Roi, Dame, Valet, 10, 9, 8, 7, 6, 5, 4, 3 et 2 (la plus faible)

Si deux cartes sont de même valeur, il y a "bataille".

- Chaque joueur pose alors une carte face cachée, suivie d'une carte face visible sur la carte dévoilée précédemment.
- On recommence l'opération s'il y a de nouveau une bataille sinon, le joueur ayant la valeur la plus forte emporte tout le tas.

Lorsque l'un des joueurs **possède toutes les cartes du jeu**, la partie s'arrête et ce dernier gagne.

Pour cela Simon crée une classe Python `Carte`. Chaque instance de la classe a deux attributs : un pour sa valeur et un pour sa couleur. Il donne au valet la valeur 11, à la dame la valeur 12, au roi la valeur 13 et à l'as la valeur 14. La couleur est une chaîne de caractères : `"trefle"`, `"carreau"`, `"coeur"` ou `"pique"`.

1. Simon a écrit la classe Python `Carte` suivante, ayant deux attributs `valeur` et `couleur`, et dont le constructeur prend deux arguments : `val` et `coul`.
   a. Recopier et compléter les ..... des lignes 3 et 4 ci-dessous.

```
1. class Carte:
2.       def __init__(self, val, coul):
3.           .....valeur = .....
4.           ..... = coul
```

b. Parmi les propositions ci-dessous quelle instruction permet de créer l'objet « 7 de cœur » sous le nom `c7` ?

- `c7.__init__(self, 7, "coeur")`
- `c7 = Carte(self, 7, "coeur")`
- `c7 = Carte(7, "coeur")`
- `from Carte import 7, "coeur"`

2. On souhaite créer le jeu de cartes. Pour cela, on écrit une fonction `initialiser()` :

- sans paramètre
- qui renvoie une liste de 52 objets de la classe `Carte` représentant les 52 cartes du jeu.

  Voici une proposition de code. Recopier et compléter les lignes suivantes pour que la fonction réponde à la demande :

```
def initialiser() :
  jeu = []
  for c in ["coeur", "carreau", "trefle", "pique"] : # couleur carte
    for v in range(...) :                            # valeur carte
      carte_cree  = ...
      jeu.append(carte_cree)
  return jeu
```

3. On rappelle que dans une partie de bataille, les deux joueurs tirent chacun une carte du dessus de leur tas, et celui qui tire la carte la plus forte remporte les deux cartes et les place en dessous de son tas.

Parmi les structures linéaires de données suivantes : Tableau, File, Pile, quelle est celle qui modélise le mieux un tas de cartes dans ce jeu de la bataille ? Justifier votre choix.

4. Écrire une fonction `comparer(carte1, carte2)` qui prend en paramètres deux objets de la classe `Carte`. Cette fonction renvoie :

- `0` si la force des deux cartes est identique,
- `1` si la carte `carte1` est strictement plus forte que `carte2`
- `-1` si la carte `carte2` est strictement plus forte que `carte1`

Corrigé de l'évaluation

Exercice 1 :

1.

a. La clé primaire doit être unique, le seul attribut que peut être unique pour chaque entrée, est l'attribut Code_evaluation. Par conséquence, le seul attribut qui peut jouer le rôle de clé primaire est Code_evaluation.

b.

```
INSERT INTO Evaluations
VALUES
('EXKVLX886', 'Term7', 'Peltier', '13/10/2021', 1453)
```

2.

a. 11

b.

Term7, 13/10/2021
Term5, 07/10/2021

c.

```
SELECT Nom_evaluation
FROM Evaluations
WHERE Code_competences = 452
```

3.

a.

Il faut que le couple (Code_evaluation, Num_eleve) soit unique. Un élève donné ne peut donc pas faire plusieurs fois la même évaluation.

b.

```
SELECT Num_eleve
FROM resultats
JOIN Evaluations ON resultats.Code_evaluation = Evaluations.Code_evaluation
WHERE Code_competences = 532
```

4.

a.

attribut	type
Num_eleve	INT
Nom	CHAR
prenom	CHAR
classe	CHAR

b.

Num_eleve peut jouer le rôle de clé primaire

Exercice 2 :

1.

a.

```
class Carte:
    def __init__(self, val, coul):
        self.valeur = val
        self.couleur = coul
```

b.

```
c7 = Carte(7, "coeur")
```

2.

```
def initialiser() :
    jeu = []
    for c in ["coeur", "carreau", "trefle", "pique"] :
        for v in range(2,15) :
            carte_cree = Carte(v,c)
            jeu.append(carte_cree)
    return jeu
```

3.

La structure des données la plus adaptée est la file, puisque l'on a affaire à une structure de type FIFO (First IN First OUT). Le classement des cartes doit suivre la "règle FIFO", car la carte remportée (la dernière arrivée) doit être placée en dessous du tas.

4.

```
def comparer(carte1, carte2):
    if carte1.valeur > carte2.valeur :
        return 1
    elif carte1.valeur < carte2.valeur :
        return -1
    else :
        return 0
```