

On veut implanter une base de données pour la gestion des rencontres des tournois de tennis d'une saison donnée. On pourra ainsi déterminer facilement le classement des joueurs au niveau mondial. On dispose des informations suivantes.

Spécifications : Chaque joueur a un nom, un sexe et représente un pays. Deux joueurs peuvent former une équipe (de double, mixte ou non). Un tournoi est identifié par son nom et se déroule dans un pays donné à une date prévue. La dotation des gagnants varie selon les tournois et est exprimée dans la monnaie du pays d'accueil. Afin de plus tard afficher les dotations en utilisant la bonne « unité », on souhaite également stocker le nom de la monnaie de chaque pays. À la fin d'un tournoi, un joueur ou une équipe participant à ce tournoi obtient un score qui représente le nombre de tours passés dans le tournoi (1^{er} tour vaut 1 point, 2^{ème} tour vaut 2 points, etc.). On attribue à chaque tournoi un coefficient selon son importance.

Le score final d'un joueur (ou d'une équipe) est obtenu de la manière suivante : $\sum_{i=1}^n \text{score} \times \text{coefficient}$ pour les n tournois de l'année. Les joueurs (ou équipes) sont classés par ordre décroissant de leur score final.

Partie I : Modèles conceptuels (UML)

1. En considérant qu'une équipe forme une classe, proposer un modèle conceptuel UML pour cette base de données.

RAPPEL :

1. Identifier les *classes*.
2. Identifier les *associations* entre classes.
3. Identifier les *attributs* de chaque classe, leur *type* et souligner l'*identifiant*.
4. Identifier les attributs de chaque association et les placer dans une *classe-association*.
5. Évaluer les *multiplicités* des associations.

2. Comment transformer le modèle UML proposé à la question 1 si on ne souhaite pas qu'une équipe forme une classe ? Proposer dans ce cas un nouveau modèle UML.

Partie II : Modèles logiques (relationnels)

Traduire les modèles UML proposés dans la Partie I en modèles logiques relationnels.

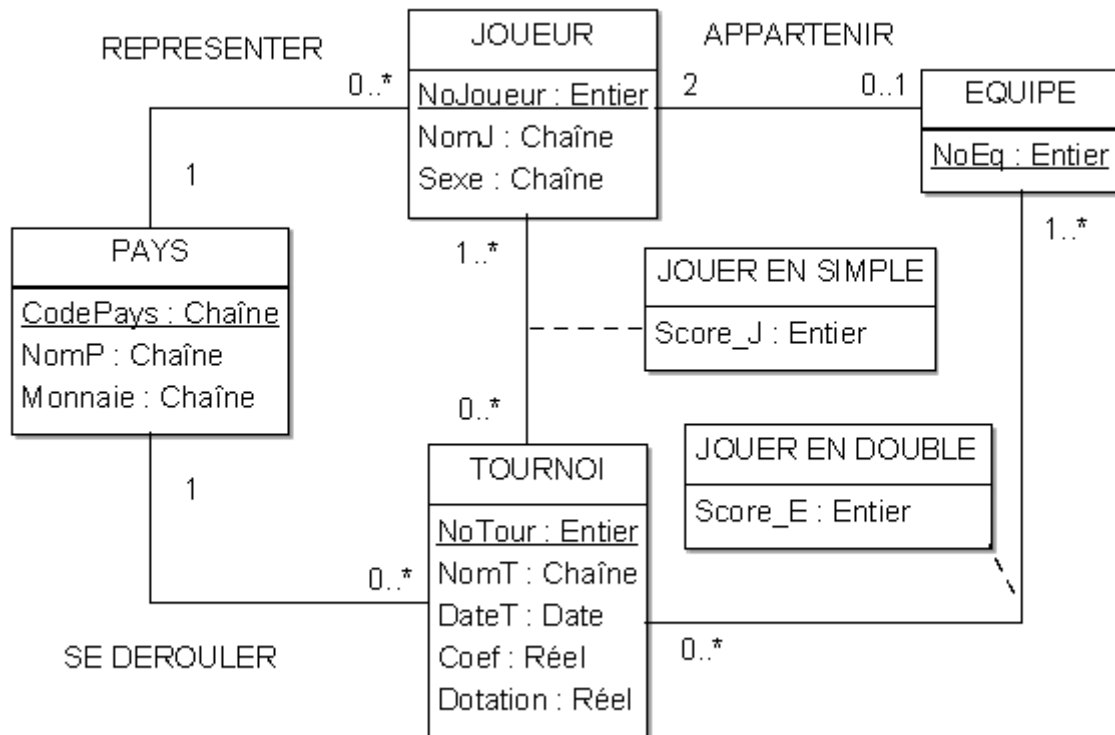
RAPPEL :

1. Chaque *classe* devient une *relation* (table). Les *attributs* de la classe deviennent *attributs* de la relation. L'*identifiant* de la classe devient *clé primaire* de la relation.
2. Chaque *association 1-N* est prise en compte en incluant la *clé primaire* de la relation dont la multiplicité maximale est 1 comme *clé étrangère* dans l'autre relation.
3. Chaque *association M-N* est prise en compte en créant une *nouvelle relation* (nouvelle table) dont la *clé primaire* est la concaténation des clés primaires des relations participantes. Les attributs de la classe-association, s'ils existent, sont insérés dans cette nouvelle relation en plus de la clé primaire.

Notation : Clés primaires
Clés étrangères#

Correction avec équipe comme classe

Modèle UML



Modèle relationnel

JOUEUR (NoJoueur, NomJ, Sexe, NoEq#, CodePays#)

EQUIPE (NoEq)

TOURNOI (NoTour, NomT, DateT, Coef, Dotation, CodePays#)

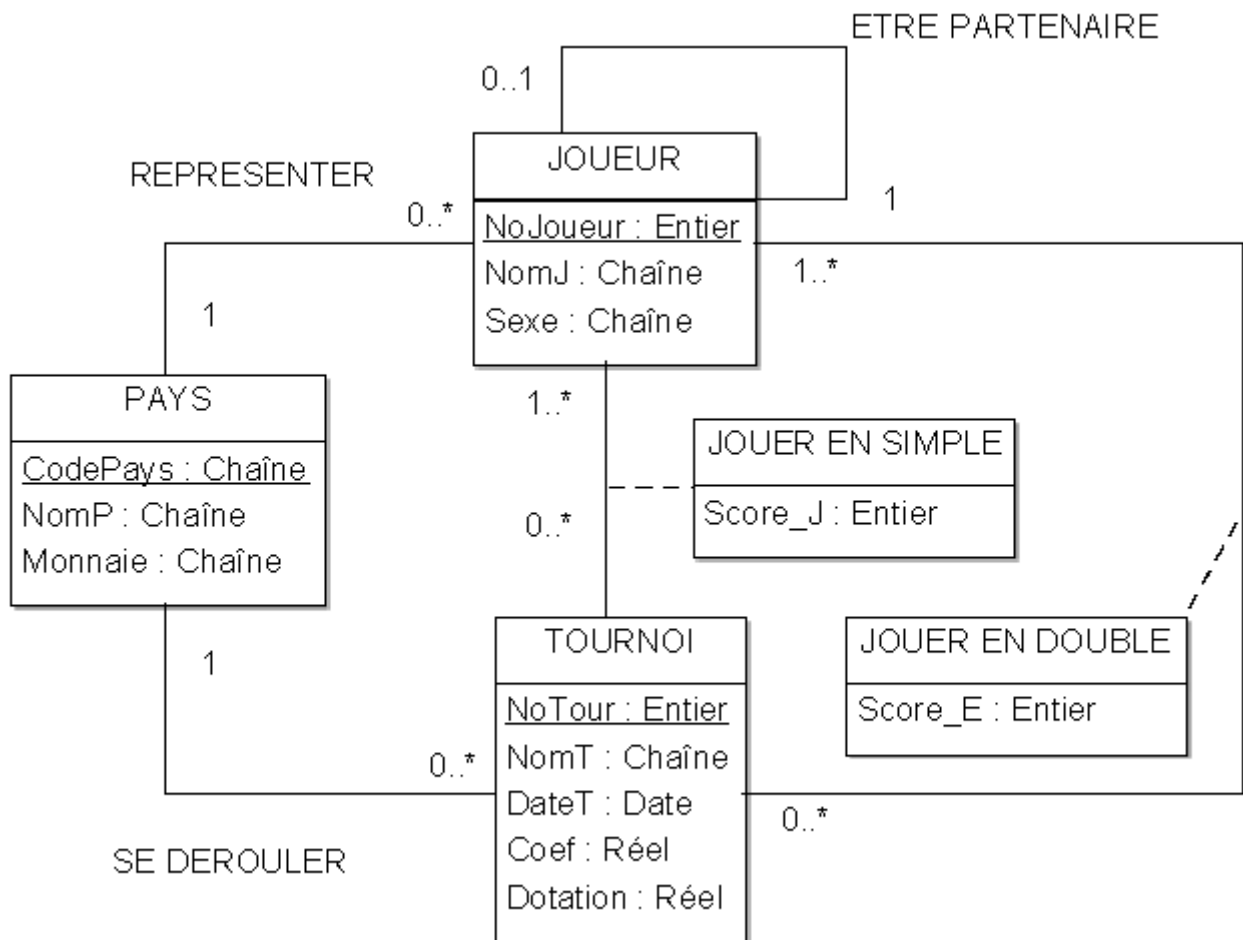
PAYS (CodePays, NomP, Monnaie)

JOUER_EN_SIMPLE (NoJoueur#, NoTour#, Score_J)

JOUER_EN_DOUBLE (NoEq#, NoTour#, Score_E)

Correction avec équipe comme association

Modèle UML



Modèle relationnel

JOUEUR (NoJoueur, NomJ, Sexe, NoPartenaire^{*}, CodePays#)

TOURNOI (NoTour, NomT, DateT, Coef, Dotation, CodePays#)

PAYS (CodePays, NomP, Monnaie)

JOUER_EN_SIMPLE (NoJoueur#, NoTour#, Score_J)

JOUER_EN_DOUBLE (NoJoueur#, NoTour#, Score_E)

^{*} Le numéro du partenaire d'un joueur (NoPartenaire) est soit le numéro de joueur (NoJoueur) d'un autre joueur, soit NULL (pas de partenaire).