

# Projet CEBD

Syrine BEN HASSINE - Thibault GAILLARD - Antony FABRE  
G1 - L3 info générale

## Question 1

### a) LesEpreuves

Relation :

$LesEpreuves(numEp, nomEp, formeEp, categorieEp, nbSportifsEp, dateEp, nomDi)$

#### Dépendances fonctionnelles trouvées :

- $numEp \rightarrow nomEp, formeEp, categorieEp, nbSportifsEp, dateEp, nomDi$
- $nomEp \rightarrow nomDi$
- $nomEp, formeEp \rightarrow nbSportifsEp$

Clé : La clé est donc **numEp**.

#### Analyse des formes normales :

- La table est en **1NF** car chaque attribut contient une seule valeur.
- La table est en **2NF** car la clé génère tous les attributs non-clés.
- La table est en **3NF** car les attributs non-clés dependent directement de la clé.
- La table n'est pas en **BCNF** car il existe les dépendances  $nomEp \rightarrow nomDi$  et  $(nomEp, formeEp) \rightarrow nbSportifsEp$  car la partie gauche de DF n'est pas une clé ( $nomEp$  n'est pas une clé).

#### Décomposition en BCNF :

- T1( $numEp \{id\}, nomEp, formeEp, categorieEp, nbSportifsEp, dateEp$ )
- T2( $nomEp \{id\}, nomDi$ )
- T3( $nomEp, formeEp \{id \text{ double}\}, nbSportifsEp$ )

### b) LesSportifsEQ

Relation :

$LesSportifsEQ(numSp, nomSp, prenomSp, pays, categorieSp, dateNaisSp, numEq)$

#### Dépendances fonctionnelles trouvées :

- $(nomSp, prenomSp, numEq) \rightarrow numSp, pays, categorieSp, dateNaisSp$
- $numEq \rightarrow pays$
- $numSp \rightarrow nomSp, prenomSp, pays, categorieSp, dateNaisSp$

Clés : Les clés de la table sont (**nomSp, numEq**).

### Analyse des formes normales :

- La table est en **1NF** car chaque attribut contient une seule valeur.
- La table n'est pas en **2NF** car (nomSp, prenomSp, categorieSp, dateNaisSp) ne dépend que de numSp.

### Décomposition en BCNF :

- T1(nomSp, prenomSp, numEq {id triple}, numSp, pays, categorieSp, dateNaisSp)
- T2(numEq {id}, pays)
- T3(numSp {id}, nomSp, prenomSp, pays, categorieSp, dateNaisSp, numEq)

### Diagramme UML et Contraintes

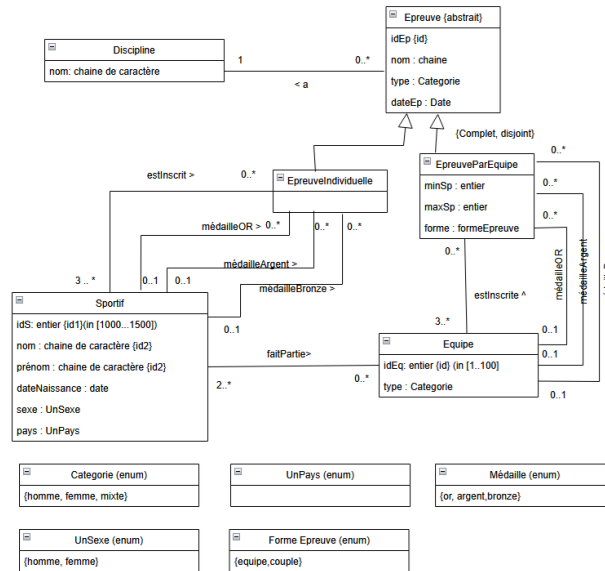


FIGURE 1 – Diagramme UML des entités principales

### Contraintes supplémentaires (non représentées dans le diagramme UML)

- La date de naissance des sportifs est antérieure à la date de l'épreuve.
- Tous les sportifs d'une même équipe appartiennent au même pays.
- Pour une épreuve individuelle, la catégorie est du même type que le sexe du sportif.
- Pour une équipe féminine, il n'y a que des sportives féminines.
- Pour une équipe masculine, il n'y a que des sportifs masculins.
- Pour une équipe mixte, il y a au moins un sportif masculin et au moins une sportive féminine.
- Si une médaille est attribuée, alors nécessairement la médaille de rang supérieur est également attribuée.
- Les médailles (or, argent, bronze) d'une même épreuve sont attribuées à des personnes différentes.
- Une personne ou une équipe qui reçoit une médaille doit être inscrite à l'épreuve correspondante.

### Question 3

On a décidé de faire de pays un type énuméré. Ce type décrit tous les pays et cela évite de faire des fautes de frappe en rentrant les données et ça évite à chaque JO de réécrire tous les pays.

On a aussi modifié Médaille. En suivant le schéma UML, on aurait dû avoir 3 tables : une pour les médailles d'or, une pour les médailles d'argent et une pour les médailles de bronze. Nous avons voulu rassembler en une grande table pour simplifier les requêtes et rassembler les données.

Une fois le schéma traduit, nous avons les relations suivantes (avec les contraintes associées).

**LesPays(nomP)** (type énuméré)

**LesTypes(nomT)** (type énuméré)

**LesFormes(nomF)** (type énuméré)

**LesSexes(sexe)** (type énuméré)

**LesDisciplines(nomD)**

**LesEpreuvesIndividuelles(idEp, nom, type, date, discipline)**

$\text{LesEpreuvesIndividuelles}[\text{discipline}] \subseteq \text{LesDisciplines}[\text{nomD}]$

$\text{LesEpreuvesIndividuelles}[\text{type}] \subseteq \text{LesTypes}[\text{nomT}]$

**LesEpreuvesParEquipe(idEp, nom, type, date, minsp, maxsp, forme, discipline)**

$\text{LesEpreuvesParEquipe}[\text{discipline}] \subseteq \text{LesDisciplines}[\text{nomD}]$

$\text{LesEpreuvesParEquipe}[\text{forme}] \subseteq \{\text{'Equipe'}, \text{'Couple'}\}$

$\text{LesEpreuvesParEquipe}[\text{type}] \subseteq \{\text{'Masculin'}, \text{'Feminin'}, \text{'Mixte'}\}$

$\text{MinSP} \leq \text{MaxSP}$

**LesSportifs(idS, nom, prenom, date de naissance, sexe, pays)**

nom, prenom clé secondaire

$\text{LesSportifs}[\text{pays}] \subseteq \text{LesPays}[\text{nomP}]$

$\text{LesSportifs}[\text{sexe}] \subseteq \{\text{'Masculin'}, \text{'Feminin'}\}$

**LesEquipes(idEq, type)**

$\text{LesEquipes}[\text{type}] \subseteq \{\text{'Masculin'}, \text{'Feminin'}, \text{'Mixte'}\}$

**LesMembresEquipes(idEq, idS)**

$\text{LesMembresEquipes}[\text{idEq}] \subseteq \text{LesEquipes}[\text{idEq}]$

$\text{LesMembresEquipes}[\text{idS}] \subseteq \text{LesSportifs}[\text{idS}]$

**LesInscriptionsEpreuvesIndividuelles(idEp, idS)**

$\text{LesInscriptionsEpreuvesIndividuelles}[\text{idEp}] \subseteq \text{LesEpreuvesIndividuelles}[\text{idEp}]$

$\text{LesInscriptionsEpreuvesIndividuelles}[\text{idS}] \subseteq \text{LesSportifs}[\text{idS}]$

**LesInscriptionsParEquipe(idEp, idEq)**

$\text{LesInscriptionsParEquipe}[\text{idEp}] \subseteq \text{LesEpreuvesParEquipe}[\text{idEp}]$

$\text{LesInscriptionsParEquipe}[\text{idEq}] \subseteq \text{LesEquipes}[\text{idEq}]$

**LesMedailles(idEp, or, argent, bronze)**

$\text{LesMedailles}[\text{idEp}] \subseteq (\text{LesInscriptionsEpreuvesIndividuelles}[\text{idEp}] \cup \text{LesEpreuvesParEquipe}[\text{idEp}])$

$\text{LesMedailles}[\text{or}] \subseteq (\text{LesMembresEquipes}[\text{idEq}] \cup \text{LesInscriptionsEpreuvesIndividuelles}[\text{idEp}])$

$\text{LesMedailles}[\text{argent}] \subseteq (\text{LesMembresEquipes}[\text{idEq}] \cup \text{LesInscriptionsEpreuvesIndividuelles}[\text{idEp}])$

$\text{LesMedailles}[\text{bronze}] \subseteq (\text{LesMembresEquipes}[\text{idEq}] \cup \text{LesInscriptionsEpreuvesIndividuelles}[\text{idEp}])$

Vues possibles :

- Pour calculer le nombre de sportif dans une équipe
- Pour calculer l'âge des sportifs

## Explication Code Trigger

### 1. Trigger : CheckInscriptionBeforeMedailleIndiv

**Objectif** : Vérifier les contraintes avant l'insertion d'une médaille individuelle.

Ce trigger est exécuté **avant** chaque insertion dans la table `LesMedaillesIndividuelles`, et son rôle est de vérifier deux contraintes principales :

- **Vérification 1 : Les sportifs doivent être inscrits à l'épreuve** Avant d'attribuer une médaille à un sportif, il faut vérifier qu'il est inscrit à l'épreuve correspondante. Le trigger fait cela pour les trois médailles : or, argent et bronze. Il effectue trois requêtes pour chaque sportif (médaille d'or, argent et bronze) en comptant le nombre d'inscriptions de ce sportif à cette épreuve. Si un des sportifs n'est pas inscrit (`COUNT(*) = 0`), une erreur est levée.
- **Vérification 2 : Pas de doublon de médaille pour un même sportif** Un sportif ne peut pas recevoir deux médailles pour la même épreuve. Le trigger vérifie si un même sportif est attribué à plusieurs médailles (or, argent, bronze). Si c'est le cas, une exception est levée.

### 2. Trigger : CheckInscriptionBeforeMedailleEquipe

**Objectif** : Vérifier les contraintes avant l'insertion d'une médaille par équipe.

Ce trigger est similaire au précédent, mais il est destiné aux médailles par équipe, en vérifiant aussi deux contraintes principales :

- **Vérification 1 : Les équipes doivent être inscrites à l'épreuve** Avant d'attribuer une médaille à une équipe, il faut s'assurer que l'équipe est inscrite à l'épreuve. Le trigger effectue trois requêtes similaires pour vérifier l'inscription des trois équipes : or, argent et bronze.
- **Vérification 2 : Pas de doublon de médaille pour une même équipe** Il faut aussi s'assurer qu'une équipe ne peut pas recevoir deux médailles dans la même épreuve. Le trigger vérifie si une équipe est déjà inscrite sous une médaille différente et lève une erreur si nécessaire.

### 3. Trigger : VerifierMemePaysEquipe

**Objectif** : Vérifier que tous les sportifs d'une même équipe sont du même pays.

Ce trigger vérifie qu'un sportif ne peut pas être ajouté à une équipe si les autres membres de cette équipe viennent d'un pays différent. Cette contrainte s'applique lors de l'insertion ou de la mise à jour d'un membre d'une équipe dans la table `LesMembresEquipes`.

- **Vérification 1 : Récupérer le pays du nouveau membre** Le trigger récupère le pays du sportif à ajouter à l'équipe (via une requête dans la table `LesSportifs`).
- **Vérification 2 : Vérifier s'il y a déjà des membres dans l'équipe** Si l'équipe a déjà des membres (`v_nombre_membres > 0`), on récupère le pays du premier membre de l'équipe.
- **Vérification 3 : Comparer le pays** Si le pays du nouveau membre est différent du pays du premier membre de l'équipe, une erreur est levée.

## Récapitulatif des Points Clés des Triggers

- `CheckInscriptionBeforeMedailleIndiv` : Vérifie que chaque sportif (or, argent, bronze) est inscrit à l'épreuve et qu'il n'y a pas de doublon de médaille pour un même sportif.
- `CheckInscriptionBeforeMedailleEquipe` : Vérifie que chaque équipe (or, argent, bronze) est inscrite à l'épreuve et qu'il n'y a pas de doublon de médaille pour une même équipe.
- `VerifierMemePaysEquipe` : Vérifie que tous les membres d'une équipe sont du même pays avant d'ajouter un nouveau membre à cette équipe.

### **Lien de GitHub**

Le lien de depot git : <https://github.com/Syrinh/ProjetCebd.git>