

# RAPPORT DE PROJET

## Gestion des Jeux Olympiques d'Hiver

---

### 1. Conception UML et Normalisation

#### Les dépendances fonctionnelles

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

Dans l'énoncé on a : Le numéro qui identifie l"épreuve.

Alors on constate qu' un épreuves est identifié par numEp unique {clé}

**numEp → nomEp, formeEp, categorieEp, nbSportifsEp, dateEp, nomDi**

**Atomique:** FN1

1 Clé Simple -> **pas de dépendances partiel** : FN2

**Pas de dépendances non cles** : FN3

**Le seul déterminant non trivial est la clé** : BCNF

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

**numSp → nomSp, prenomSp, pays, categorieSp, dateNaisSp**

Chaque sportif est aussi identifiée par un numero

**(nomSp, prenomSp) → numSp, pays, categorieSp, dateNaisSp**

On suppose que deux sportifs se distinguent par leur nom ou par leur prénom s'ils ont le même nom.

**numEq → pays**

Tous les sportifs d'une même équipe doivent ^etre du même pays

---

Cle : (numSp, numEq) ou (nomSp , prenomSp, numEq)

**Conclusion :** FN1 mais pas FN2 (pas FN3 ni BCNF , dépendances partiel)

BCNF :

**On décompose LesSportifsEQ en relations plus normales :**

. Sportifs(numSp, nomSp, prenomSp, pays, categorieSp, dateNaisSp)

**numSp → nomSp, prenomSp, pays, categorieSp, dateNaisSp**

. Equipes(numEq, paysEq )

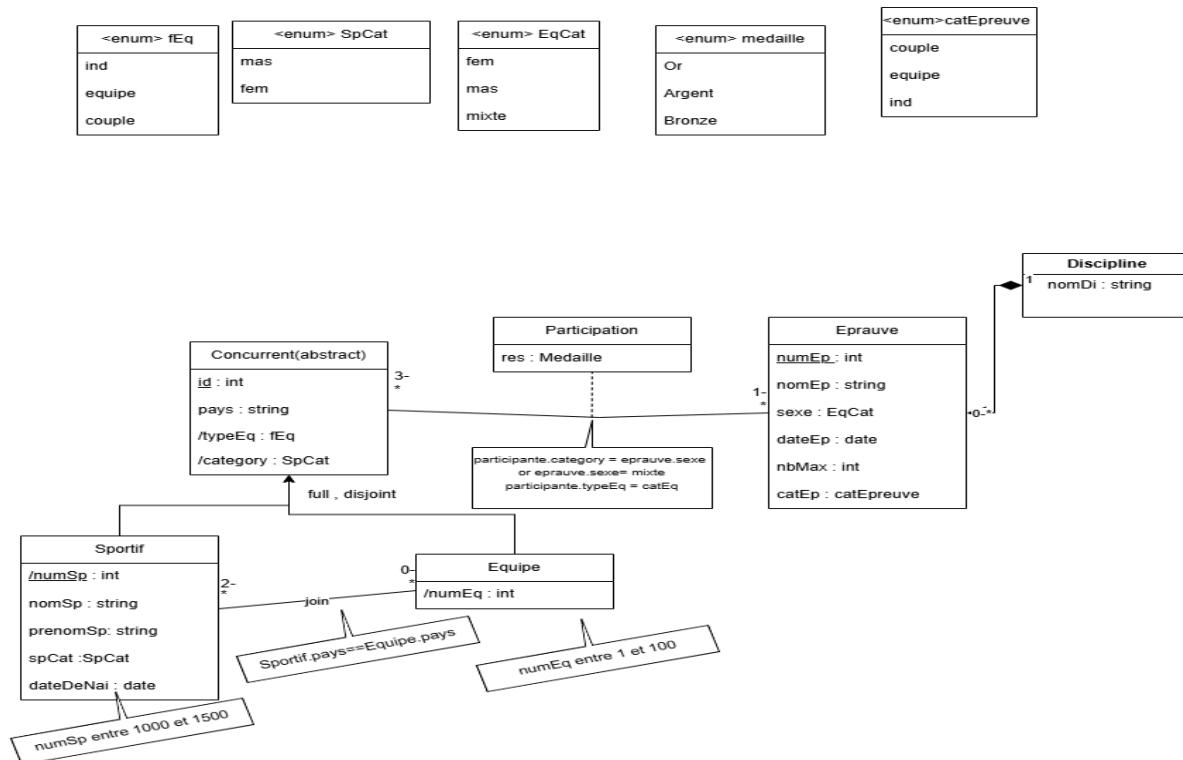
Une même équipe ou un même sportif peut être engagée dans plusieurs épreuves de la même discipline (on peut ajouter latt nomDi)

**numEq->paysEq,nomDi**

. SpDansEq(numSp , numEq)

**Toutes ces relations sont en 3NF et même en BCNF.**

## UML



---

## 2. Modèle relationnel

Sportifs(numSp, nomSp, prenomSp, dateNaisSp, spCat, pays)

Avec  $1000 \leq \text{numSp} \leq 1500$

/

Equipes(numEq, pays)

Avec  $1 \leq \text{numEq} \leq 100$

/

Membre\_Equipe(numSp, numEq)

Membre\_Equipe[numSp]  $\subseteq$  Sportifs[numSp]

Membre\_Equipe[numEq]  $\subseteq$  Equipes[numEq]

avec au moins 2 sportifs par équipe

/

Disciplines(nomDi)

/

Epreuves(numEp, numDi, nomEp, sexe, nbMax, catEp, dateEp)

Epreuves[nomDi]  $\subseteq$  Disciplines[nomDi]

où :

sexe  $\in \{\text{masculin, féminin, mixte}\}$  (type EqCat),

catEp  $\in \{\text{individuelle, équipe, couple}\}$  (type CatEpreuve).

/

Participations(P\_ID, numE, result)

---

Participations[numEp] ⊆ Epreuves[numEp]

result ∈ {Or, Argent, Bronze} (type Medaille), valeur éventuellement nulle en absence de médaille.

/

## Vue Participant

On définit une vue logique **Participant**, qui regroupe l'ensemble des entités pouvant participer à une épreuve (sportifs ou équipes) :

Participant(P\_ID,pays , forme, categorie)

Participant[P\_ID]=Sportifs[numSp] ∪ Equipes[numEq]

Les attributs forme (individuelle / par équipe / par couple) et categorie (masculin / féminin / mixte) sont **dérivés** des règles du modèle UML :

- si Participant est un Sportif → forme = individuelle, categorie = spCat,
- si Participant est une Equipe → forme et categorie sont déterminées en fonction du type d'équipe et de la composition de ses membres.

La contrainte suivante relie la participation aux épreuves :

Participations[P\_ID] ⊆ Participant[P\_ID] (*un participant n'est pas obligé de participer*)

domaine(numSp) = entier, 1000 ≤ numSp ≤ 1500

domaine(numEq) = entier, 1 ≤ numEq ≤ 100

domaine(dateNaisSp) = date

domaine(dateEp) = date

domaine(spCat) = {'masculin', 'feminin'}

domaine(categorieEp) = {'masculin', 'feminin', 'mixte'}

domaine(formeEp) = {'individuelle', 'equipe', 'couple'}

domaine(medaille) = {'Or', 'Argent', 'Bronze'}

### 3. GIT

<https://github.com/mouhieddinelakhdar-hash/CEBD.git>

The screenshot shows a GitHub repository page for 'mouhieddinelakhdar-hash/CEBD.git'. At the top, there are navigation buttons for 'main' (selected), '1 Branch', '0 Tags', 'Go to file', 'Add file', and 'Code'. Below this is a table of files and their commit history:

File	Description	Time Ago
actions	Initial commit: project structure	2 minutes ago
data	Initial commit: project structure	2 minutes ago
utils	Initial commit: project structure	2 minutes ago
.gitignore	Update main script and gitignore	2 minutes ago
README.md	Initial commit	13 minutes ago
README.txt	Initial commit: project structure	2 minutes ago
main.py	Update main script and gitignore	2 minutes ago

Below the file list is a section titled 'Commits' with a table of recent commits:

Commit	Author	Date	Actions
Add trigger for teams	Bouguerra Lakhdar	1 minute ago	02b4aab
Update main script and gitignore	Bouguerra Lakhdar	18 minutes ago	00c5094
Initial commit: project structure	Bouguerra Lakhdar	18 minutes ago	84b83ad
Initial commit	mouhieddinelakhdar-hash	28 minutes ago	f3a8c38