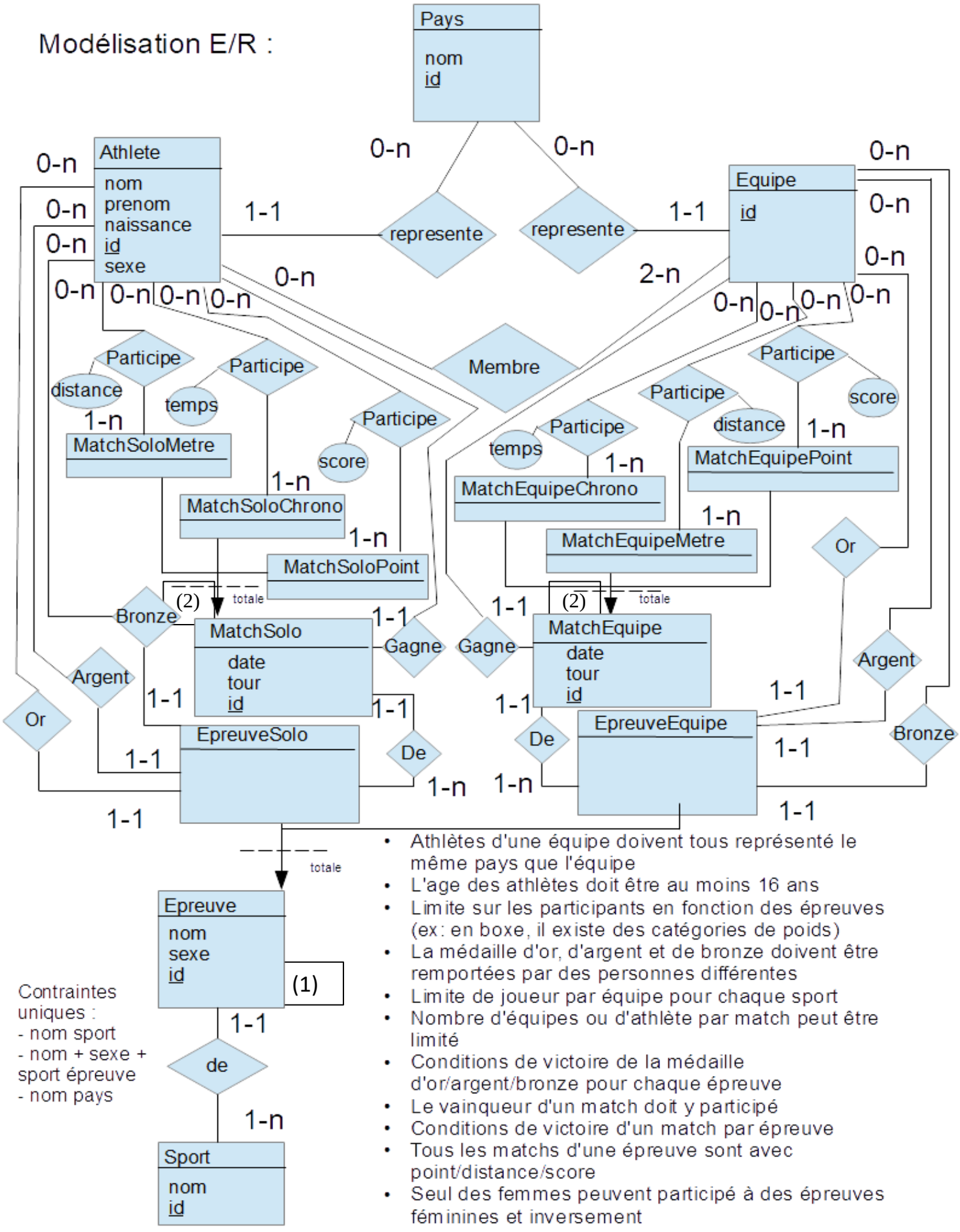
16/12/2018

Projet Base De Données L2
Rio ne répond plus

Mohammed Diawara
Dao Thauvin

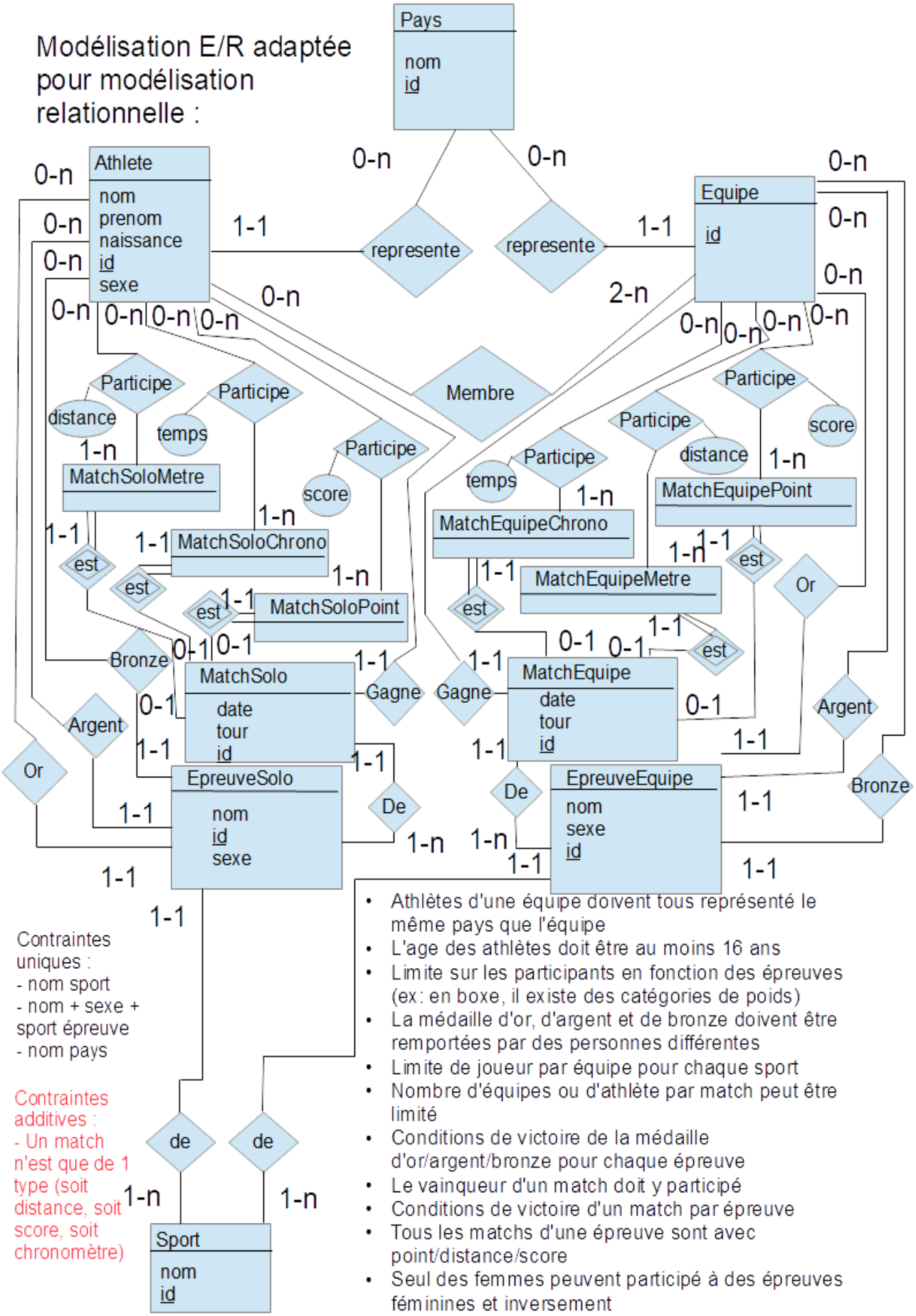
Sommaire

Modélisation E/R :



Contraintes uniques :
- nom sport
- nom + sexe + sport épreuve
- nom pays

- Athlètes d'une équipe doivent tous représenté le même pays que l'équipe
- L'age des athlètes doit être au moins 16 ans
- Limite sur les participants en fonction des épreuves (ex: en boxe, il existe des catégories de poids)
- La médaille d'or, d'argent et de bronze doivent être remportées par des personnes différentes
- Limite de joueur par équipe pour chaque sport
- Nombre d'équipes ou d'athlète par match peut être limité
- Conditions de victoire de la médaille d'or/argent/bronze pour chaque épreuve
- Le vainqueur d'un match doit y participé
- Conditions de victoire d'un match par épreuve
- Tous les matchs d'une épreuve sont avec point/distance/score
- Seul des femmes peuvent participé à des épreuves féminines et inversement


Modélisation E/R adaptée pour modélisation relationnelle :
Pays
nom
id
Athlete
nom
prenom
naissance
id
sexe
Equipe
id
represente
represente
Membre
Participe
distance
temps
score
MatchSoloMetre
MatchSoloChrono
MatchSoloPoint
MatchEquipeChrono
MatchEquipeMetre
MatchEquipePoint
est
Bronze
Argent
Or
Gagne
MatchSolo
date
tour
id
MatchEquipe
date
tour
id
De
EpreuveSolo
nom
id
sexe
EpreuveEquipe
nom
sexe
id
de
Sport
nom
id
0-n
1-1
1-n
2-n
0-1
Contraintes uniques :
- nom sport
- nom + sexe + sport épreuve
- nom pays
Contraintes additives :
- Un match n'est que de 1 type (soit distance, soit score, soit chronomètre)
• Athlètes d'une équipe doivent tous représenté le même pays que l'équipe
• L'age des athlètes doit être au moins 16 ans
• Limite sur les participants en fonction des épreuves (ex: en boxe, il existe des catégories de poids)
• La médaille d'or, d'argent et de bronze doivent être remportées par des personnes différentes
• Limite de joueur par équipe pour chaque sport
• Nombre d'équipes ou d'athlète par match peut être limité
• Conditions de victoire de la médaille d'or/argent/bronze pour chaque épreuve
• Le vainqueur d'un match doit y participé
• Conditions de victoire d'un match par épreuve
• Tous les matchs d'une épreuve sont avec point/distance/score
• Seul des femmes peuvent participé à des épreuves féminines et inversement

Modélisation relationnelle :

Athlete(idA ,prenom, nom,naissance, #idP,sexe)
Pays(idP, nom)
Equipe(idE, #idPays)
Membre(#idA,#idE)
Sport(idS,nom)
EpreuveSolo(idES,nom,#idS,#mor,#margent,#mbronze,sexe)
EpreuveEquipe(idEE,nom,#idS,#mor#margent,#mbronze,sexe)
MatchSolo(idMS,#gagnant,jour,#idES,tour)
MatchEquipe(idME,#gagnant,jour,#idEE,tour)
MatchSoloPoint(#idMS)
MatchSoloChrono(#idMS)
MatchSoloMetre(#idMS)
MatchEquipePoint(#idME)
MatchEquipeChrono(#idME)
MatchEquipeMetre(#idME)
ParticipationMatchEquipePoint(#idMEP,#idE,score)
ParticipationMatchEquipeMetre(#idMEM,#idE,distance)
ParticipationMatchEquipeChrono(#idMEC,#idE,temps)
ParticipationMatchSoloPoint(#idMSP,#idA,score)
ParticipationMatchSoloMetre(#idMSM,#idA,distance)
ParticipationMatchSoloChrono(#idMSC,#idA,temps)

Contraintes uniques :
- nom sport
- nom + sexe + sport épreuve
- nom pays

mbronze->medaille bronze (athlete ou equipe)
margent->medaille argent (athlete ou equipe)
mor->medaille or (athlete ou equipe)
IdMSM -> id de MatchSoloMetre
IdMSP -> id de MatchSoloPoint
IdMSC -> id de MatchSoloChrono
IdMEM -> id de MatchEquipeMetre
IdMEP -> id de MatchEquipePoint
IdMEC -> id de MatchEquipeChrono

Contraintes non représentées :

Représentable dans SQL :

- La médaille d'or, d'argent et de bronze doivent être remportées par des personnes différentes

Conditions à vérifier avant insertion :

- Athlètes d'une équipe doivent tous représenté le même pays que l'équipe
- Limite sur les participants en fonction des épreuves (ex: l'escrime à une catégorie homme et femme)
- Limite de joueur par équipe pour certains sports
- Nombre d'équipes ou d'athlète par match peut être limité
- Conditions de victoire de la médaille d'or/argent/bronze pour chaque épreuve
- Le vainqueur d'un match doit y participé
- Conditions de victoire d'un match par épreuve
- Tous les matchs d'une épreuve sont avec point/distance/score
- Les athlètes doivent avoir au moins 16 ans
- Un match n'est que de 1 type (soit distance, soit score, soit chronomètre)

Modélisation Basique

Tout d'abord nous avons commencé par une modélisation basique :

- Sport : ensemble des sports
- Epreuve : catégorie de sports
  Remarque : chaque épreuve à des médailles qui correspondent à des athlètes/équipes (le nombre étant fini avec seulement une médaille or, argent, bronze par épreuve).
- Match : Match d'une épreuve
- Athlète : Ensemble des athlètes
- Equipe : Ensemble des équipes
- Membre : Membres des équipes
- Participation : Participation d'une équipe/d'un athlète à un match avec son score

Les Spéciations

1 Une Epreuve peut être une EpreuveSolo ou EpreuveEquipe, la séparation est nécessaire car les médailles sont des clés étrangères d'athlète ou d'équipe, il faut donc des champs différents (créer 2 tables enfants restent le choix le plus simple pour reconnaître les épreuves en solitaire et en équipe).
  Remarque : la séparation est aussi nécessaire pour participation et match, match ayant soit un gagnant athlète, soit un gagnant équipe. De même pour participation qui correspond à la participation d'un athlète ou d'une équipe.

2 Un Match peut être :
  - Un match à point (avec champ score)
  - Un match à temps (avec champ chrono)
  - Un match à distance (avec champ distance)

  Il est nécessaire de créer des tables différentes pour différencier les champs score, chrono et distance.

Le passage à la modélisation relationnelle

- (1) Pour les épreuves, l'entité épreuve a été supprimée car elle n'était pas nécessaire, n'ayant qu'une relation avec sport (et ces deux enfants), nous l'avons donc supprimé et hérité toutes ces liaisons et champs à ces enfants.
- (2) Dans le cas des matchs nous avons décidé de garder toutes les entités, même si cela est la source d'un problème d'insertion (écrit en rouge dans le schéma relationnel), cela permet de nettement simplifié les requêtes notamment en fonction des dates :
  Ex : Quel est le nombre d'épreuves se déroulant le 2016-08-14 ?
  Cette requête aurai due avoir 6 UNION sans matchEquipe et matchSolo alors que dans avec mon implementation, elle n'en demande que 1.

Anomalies

Anomalies d'insertion	Anomalies de suppression
Elles sont indiquées page 4, il s'agit des conditions à vérifier avant insertion.	Elles sont gérées soit par CASCADE si le champ ne peut être null, soit mis à null avec SET NULL, car même après suppression d'une équipe ou d'un athlète (par exemple s'il a triché), les matchs et les épreuves restent présentes en attente d'une décision pour le nouveau gagnant/médaillé.
Anomalies de mise à jour	Redondance
Chaque table où il existe une dépendance fonctionnelle est représenté par un identifiant qui ne donne aucune information, il n'y a donc aucun intérêt à les modifiés, il n'y a donc pas d'anomalies de mise à jour.	Il existe une redondance minime, seuls les identifiants sont répétés avec les clés étrangères.

Formes normales

La plus part des tables sont représentées par un identifiant unique, on a donc :

id->reste des champs

Ce n'est pas le cas pour les tables :

- ParticipationSolo/EquipePoint/Score/Metre
- Membre
- MatchSolo/EquipePoint/Score/Metre

Or

- MatchSolo/EquipePoint/Score/Metre n'a qu'un champ, il n'y a donc aucunes dépendances intéressantes
- Les tables participationSolo/EquipePoint/Score/Metre ont une dependance fonctionnelle : idMatch,idAthlete/Epreuve->score/point/distance
- Il n'y a pas de dépendances fonctionnelles intéressantes dans la table Membre

On observe aussi les dépendances :

- nomSport -> idSport
- nomEpreuve,sexeEpreuve,SportEpreuve -> idEpreuve
  (Donc nomEpreuve,sexeEpreuveSportEpreuve->idEpreuve->reste de la table)
- nomPays -> idPays

Ainsi on observe que toutes les dépendances fonctionnelles viennent de superclés (dont le reste de leur table dépend), il s'agit donc d'une forme normale de Boyce-Codd, il y a donc absence de redondance comme indiqué dans la partie précédente. La base est donc aussi sous 3ème forme normale, cette forme étant moins forte que celle de Boyce-Codd.

Limitation de l'implémentation

- En cas d'égalité, il est possible d'avoir plusieurs gagnants dans un même match. Or avec notre représentation, cela n'est pas possible.
- Il existe 21 tables dans notre implémentation ce qui crée des requêtes avec beaucoup de jointure et donc des requêtes complexes.
  Par exemple : pour aller de la table pays à sport il faudra au moins faire une jointure avec 5 tables (Sport -> Epreuve -> EpreuveSolo -> Athlete -> Pays par exemple).
- Chaque autre type de notation pour les matchs (par exemple pour l'haltérophilie, il s'agit d'une notation par poids porté) demande 4 nouvelles tables (dans notre exemple ParticipationSoloPoids, MatchSoloPoids, ParticipationEquipePoids, MatchEquipePoids) pour pouvoir représenter ce type de notation tout en les différenciants des autres.
- Les manches ne sont pas représentées (par exemple au tennis).
- Les recherches globales sont complexes, notamment sur les participants, par exemple : Les athlètes ayant participés au JO de Rio 2016. Cela demande de faire des UNION pour toutes les tables Participation (3 tables par table Epreuve) et aussi pour les tables Epreuve (2 tables), ce n'est pas un problème ici mais cela peut vite devenir problématique pour des requêtes plus complexes.

L'indice

Pour l'indice, nous avons choisie de faire un indice assez simple à l'aide d'une vue matérialisé étant donné que l'indice récupère juste des informations des tables.
Ainsi notre indice ce calcul comme ci-dessous :
Un volontaire par athlète + 5 volontaires par évènements
Ainsi lors d'un match, chaque volontaire s'occupe de son joueur :
Préparation des matchs et du matériel…
Et on ajoute 5 personnes par évènement pour s'occuper des événements eu même ou pour aider les autres volontaires.
(Si un athlète participe à 2 événements de même catégorie, on ne compte qu'un seul volontaire qui suivra l'athlète)
L'implémentation est disponible à la fin des questions.
Points forts de l'implémentation :

- Ne nécessite pas d'avoir des informations précises sur les sports/épreuves comme le nombre de personnes nécessaires…
- Les informations sont regroupées dans une seule table, donc facile d'accès
- L'accès à cette table ne permet pas de modifications des informations.

Points faibles de l'implémentation :

- Ne considère pas les épreuves/les sports, un sport peut nécessité beaucoup plus de volontaire qu'un autre.
  ➔ Pour cela il aurait fallu créer un champ dans sport/épreuve indiquant le nombre de volontaire nécessaire.
- Ne considère pas le nombre de personnes déjà présentes.
  ➔ Il aurait fallu créer un champ Volontaires qui indique le nombre de volontaire déjà présent (à notre niveau cela n'est pas faisable avec une vue matérialisé qui ne peut pas être modifié dans notre cas).

Nos requêtes

Lors de l'implémentation de nos requêtes, la complexité de notre base de données a été l'obstacle le plus important, c'est pour cela que nous avons créé une vue nous permettant un accès plus simple aux médaillés.
Nous avons choisi d'implémenter 3 requêtes :

- Le nombre d'épreuves se déroulant le 2016-08-14
  ➔ Cette requête nous montre l'importance de créer une table MatchSolo/Equipe qui rend certaines requêtes moins complexe (voir le passage à la modélisation relationnelle)
- Les athlètes ayant participés au JO de Rio 2016
  ➔ Cette requête nous montre la complexité d'une requête paressant simple mais demandant beaucoup d'effort dans notre implémentation (voir limitation de notre implémentation)
- Les sportifs n'ayant pas participé ni à un match en équipe, ni à un match noté par point
  ➔ Cette requête nous montre l'importance de séparer les matchs solo et équipe et aussi les matchs en point/chronomètre/distance (voir Les spéciations)

Voici une autre requête intéressante :

- Les Athlètes n'ayant jamais joué avec un Athlète ayant le même âge que lui.

Pour nous, il s'agit d'une requête paressant simple mais très complexe à cause de la complexité de notre base, en effet il faut vérifié les personnes avec qui il a joué en solo donc les personnes dans le même match que lui donc dont l'id de match dans les tables participations correspondent à un de ses match. Mais il faut aussi vérifié les équipes avec lequel il a joué de la même façon mais en allant plus loin en recherchant les membres de l'équipe, il faut aussi vérifier si sa propre équipe contient une personne ayant le même âge que lui, cette requête en devient extrêmement complexe.
Pour améliorer notre base, nous aurions pu :

- Fusionner les EpreuveSolo/EpreuveEquipe en les gérants avec des booléens malgré les problèmes d'insertion.
- Créer des tables ParticipationSolo/Equipe qui permettrait de regroupé les participants et les matchs sans leurs scores.

Les annexes

- ProjetBD.sql : il s’agit de la base de données.
- Insertions.sql : il s’agit des tuples à insérer dans les tables.
- ProjetBDQuestions : Il s’agit de l’ensemble de nos requêtes, on y retrouve aussi notre indice.