

Rédigé par : Jimmy Paquereau

Fiche de révisions - MCD et schéma relationnel

1. MCD - généralités

MCD (Modèle Conceptuel des Données) : un MCD est un diagramme permettant de donner une représentation schématique de tout ou partie d'une base de données relationnelle. Lorsqu'une base de données devient consistante, on a tendance à diviser le modèle en sous-modèles. Les MCD font partie plus généralement d'une méthode de conception d'origine française appelée la méthode MERISE, laquelle définit un certain nombre de schémas et diagrammes. MCD, MOT et schéma relationnel sont certainement les plus importants.

Le MCD, en particulier, visent en outre à modéliser des tables et les relations existantes entre elles. De manière quelque peu grossière, on peut voir une table comme un « gros tableau ».

Avec l'apparition puis le développement de ce qu'on qualifie de programmation orientée objets (abrégée POO ou OOP en anglais), cette méthode de conception franco-française tend depuis bien des années à être remplacée en pratique par la notation UML (*Unified Modeling Language*). On y retrouve des concepts similaires. Cependant, celle-ci fournit, disons, une vision plus orientée programmeur.

MCD et syntaxe : le MCD possède une syntaxe et un vocabulaire qui lui est propre. On parle d'entités et d'associations. C'est pourquoi le MCD est parfois appelé modèle entités-associations.

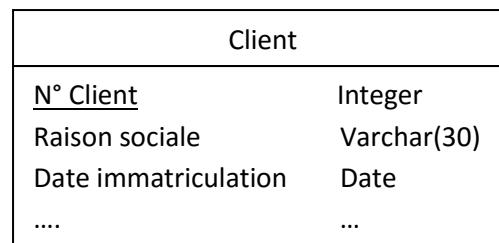
Entité : une entité une unité élémentaire qui se suffit à elle-même (exemples : client, fournisseur, produit, article...). A toute entité correspond une table (grossièrement, un tableau).

Elle possède un nom et des propriétés, encore appelés des attributs (grossièrement, les colonnes du tableau).

Optionnellement, comme en algorithmique, on précise le type, à savoir le type de chaque propriété (qu'importe !)

La ou les propriétés soulignées est ou sont appelées clef primaire.

Une entité est représentée par un rectangle.

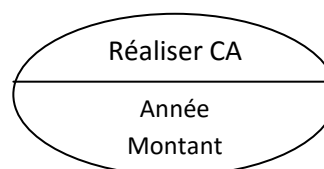


Association : une association décrit une relation existant entre plusieurs entités. Elle n'a de sens que sous réserve de l'existence des entités dont elle fait la relation. On entend par « relation » comme une forme de relation d'appartenance (exemple : un client a 0 ou plusieurs produits, sous-entendu il commercialise 0 à plusieurs produits).

Elle possède un nom et éventuellement des propriétés, appelées propriétés portées (sous-entendu portées par l'association).

Important ! A une association correspond ou non une table. Si l'association correspond à un CIF, l'association ne correspond pas à une table mais à une clef étrangère. Si l'association correspond à un CIM, elle correspond à une table.

Une association est représentée par un ovale.



Dans l'exemple ci-dessus, il y a 2 propriétés portées. Le nom d'une association est souvent un verbe, cependant que cela ne soit pas une obligation. Essentiellement, l'utilisation d'un verbe facilite la lecture du diagramme.

Clef primaire : si une table est une sorte de tableau et les propriétés des sortes de colonnes, alors une clef primaire correspond grossièrement à l'identifiant de la ligne, i.e. la ou les colonnes identifiants la ligne. Toute table possède une clef primaire, un identifiant. Cet identifiant est unique, il est propre à chaque ligne. Lorsque la clef primaire est constituée de plusieurs propriétés, on parle de clef primaire composée (sous-entendue composée de plusieurs propriétés).

Important !

La clef primaire propre à une entité doit figurer dans les propriétés de l'entité. Elle est soulignée. La clef primaire correspondant à une association est sous-entendue. Elle est constituée de la clef primaire de chacune des entités qu'elle met en relation.

Cardinalités : une association relie au moins deux entités entre elles. Si elle relie trois entités, on dira que c'est une association ternaire, ou plus simplement une ternaire. Les cardinalités précisent le lien entre les entités reliées. Elles permettent de préciser la quantité minimale et maximale qu'une entité a d'une autre entité. Dans le cas d'une ternaire, les cardinalités seront forcément des 0,n (voire 1,n).

Important !

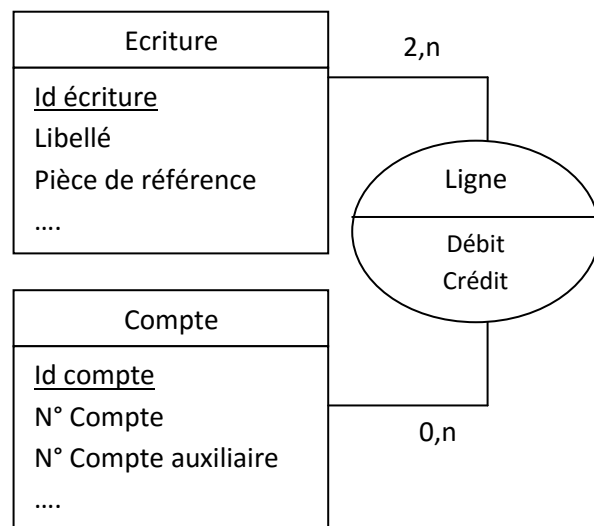
Lorsque les cardinalités sont de la forme :

- **X,n - Y,n** (exemple : 0,n - 1,n) avec X et Y des nombres, on parle de CIM (contrainte d'intégrité multiple) ;
- **X,1 - Y,n** (exemple : 1,1 - 0,n) avec X et Y des nombres, on parle de CIF (contrainte d'intégrité fonctionnelle).

Si une association représente une CIM, alors l'association correspond à une table.

Si une association représente une CIF, alors elle ne correspond pas à une table mais à une clef étrangère. Le clef étrangère apparaît alors dans la table correspondant à l'entité ayant la cardinalité de la forme X,1.

Association et entités mises en relation sont reliées par un trait, appelée patte. Sur ces pattes, on précise la cardinalité, et optionnellement le rôle de la patte, i.e. sa signification.



A droite, les cardinalités signifient :

- Une écriture comptable est constituée de 2 à plusieurs lignes ;
- Un compte peut être apparu dans aucune à plusieurs écritures comptables.
- L'association « Ligne » représente une CIM. Ligne correspond donc à une table.

Clef étrangère : dans un MCD, une clef étrangère est représentée par une CIF. Dans un schéma relationnel, la clef étrangère est représentée par un ou plusieurs champs. Lorsque la clef étrangère est constituée de plusieurs propriétés, on parle de clef étrangère composée. Grossièrement, une clef étrangère correspond, dans une table, pour chaque ligne, à une valeur pointant vers l'identifiant d'une ligne d'une autre table.

2. Schéma relationnel - généralités

Le schéma relationnel, volontiers appelé Modèle Physique des Données (MPD), est une autre représentation d'une base de données relationnelle, plus concrète (moins « conceptuelle »). Il représente concrètement ses tables, leur clef primaire, leurs clefs étrangères et leurs autres champs. On le rédige sous la forme d'une sorte de liste dans laquelle figurent les informations citées ci-avant.

La syntaxe : **MaTable(MaClefPrimaire, #UneClefEtrangere, UnChamp, ...)**

Détail :

- le ou les champs constituant la clef primaire sont soulignés ;
- les clefs étrangères éventuelles sont préfixées par un « # » ;
- les autres champs sont simplement mentionnés ;
- la clef primaire peut être constituée de plusieurs champs, tous soulignés ;
- il peut y avoir aussi bien aucune qu'une ou plusieurs clefs étrangères ;
- il peut très bien y avoir aucune, une ou plusieurs clefs étrangères dans la clef primaire ;
- une clef étrangère peut très bien être constituée de plusieurs champs, la syntaxe diffère alors légèrement :
MaTable(MaClefPrimaire, #(Champ1, Champ2), UnChamp, ...)
Le couple (Champ1, Champ2) fait ci-dessus office de clef étrangère composée.

Exemple : on notera que la CIM (« Ligne ») correspond bien à une table.

Ecriture(EcritureNum, EcritureLib, PieceRef, ...)

Ligne(#EcritureNum, #Compteld, Debit, Credit, Lettrage, ...)

Compte(Compteld, CompteNum, CompteAuxNum, CompteLib, ...)

3. Entité dépendante

Quoiqu'une entité soit une unité *a priori* indépendante, il arrive qu'on veuille la faire dépendre d'une ou plusieurs autres entités. En ce sens, une telle entité s'apparente à une association. Il y a principalement deux cas pratiques où l'on est (doit) être tenté de faire dépendre une entité d'une (ou plusieurs) autres :

o lorsque l'on souhaite que l'identifiant d'une entité dépende d'une autre entité ;

Exemple : facture n°F401CAROUF123456

On a choisi de faire dépendre notre n° de facture du compte auxiliaire propre à Carrefour (401CAROUF) et de préciser ensuite la « combienième » facture est adressée à Carrefour. Le 123456 correspond à la valeur de l'identifiant relatif, à savoir que c'est la 123456^{ème} facture adressée spécifiquement à Carrefour. Une telle situation peut être modélisée par une entité dépendante. On a alors besoin de connaître le compte auxiliaire (pour retrouver la valeur 401CAROUF) et le « n° de facture » partiel relativement à Carrefour.

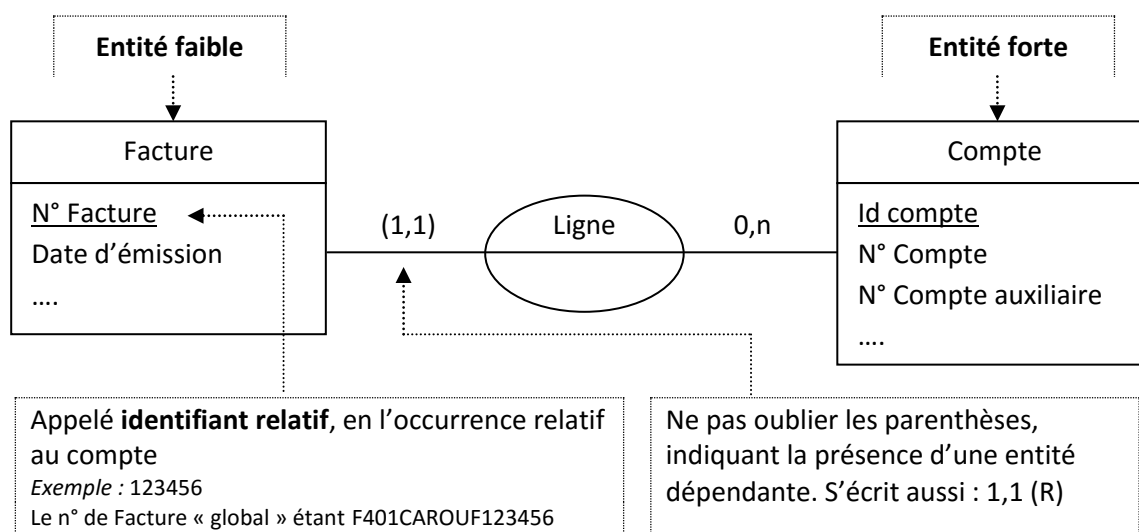


Schéma relationnel :

Facture(#IdCompte, NumFacture, DateEmission, ...)

Compte(IdCompte, NumCompte, NumCompteAux, ...)

o lorsqu'une l'existence d'une entité n'a de fait de sens que sous réserve de l'existence d'une ou plusieurs autres.

Exemple : un appartement n'existe que sous réserve de l'existence de l'immeuble. Détruisez l'immeuble, l'appartement n'existe plus. On peut alors modéliser cette dépendance par une entité dépendante.

Important ! Comprenez bien que, lorsque l'on modélise une base de données, on pense la modélisation en fonction du résultat qu'on veut obtenir. Il y a souvent plusieurs possibilités. On choisit (doit choisir) la modélisation qui répond au besoin.

4. Pseudo-entité

Le concept de pseudo-entité est assez proche de celui d'entité dépendante. Cette fois-ci, on souhaite non plus qu'une entité se comporte un peu comme une association, mais qu'une association se comporte un peu comme une entité. En fait, une association met en relation au moins deux entités. Mais il arrive que l'on souhaite pouvoir adjoindre des CIF (des clefs étrangères quoi...) à une association. Dit autrement : on peut avoir une association de type CIM, relative à deux entités (ou plus), et pouvoir lui ajouter d'autres relations (comme on le ferait pour une entité). Alors, on peut modéliser cette situation sous forme de pseudo-entité. On pourrait également le faire sous la forme d'une entité dépendante avec typiquement une entité faible et deux entités fortes (à méditer).

Comment se visualiser une pseudo-entité ? Se dire, une pseudo-entité, voilà, c'est une CIM centrale (i.e. typiquement une association relative à deux entités) + une ou plusieurs CIF associées à la CIM.

Quand est-on (doit-on) être tenté d'utiliser une pseudo-entité ? Lorsqu'une relation (association) n'a de sens que relativement à deux entités (ou plus).

Exemple : on souhaite réserver une nuit d'hôtel à une date donnée. La réservation n'a de sens que relativement à l'hôtel et à la date (i.e. réservez d'une part l'hôtel, d'autre par la date, ne fait pas sens...).

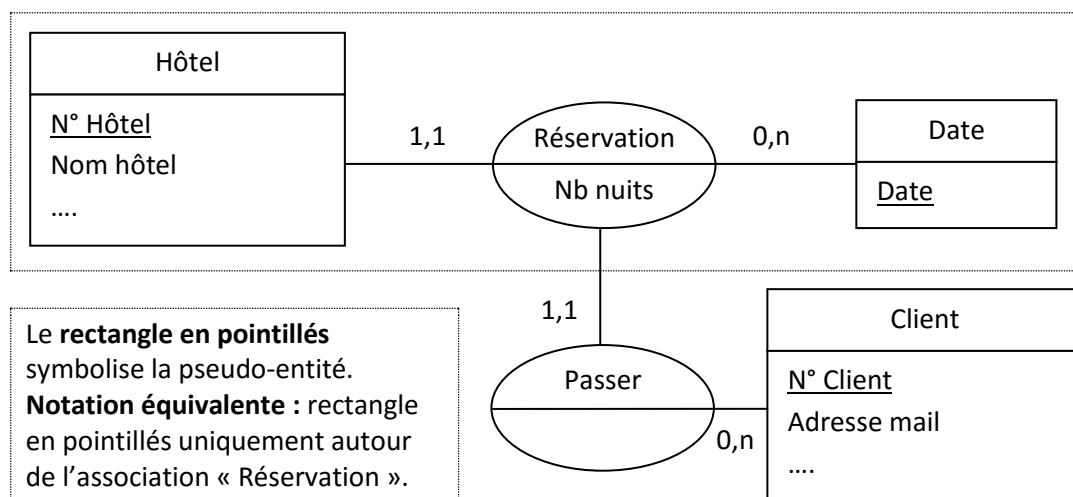


Schéma relationnel : notez que, très normalement, « Réservation » se comporte est une CIM et se comporte comme une CIM, « Passer » est une CIF et se comporte également comme une CIF.

Hôtel(NumHotel, NomHotel, ...)

Date(Date) ← en pratique, inutile de créer une table date

Client(NumClient, MailClient, ...)

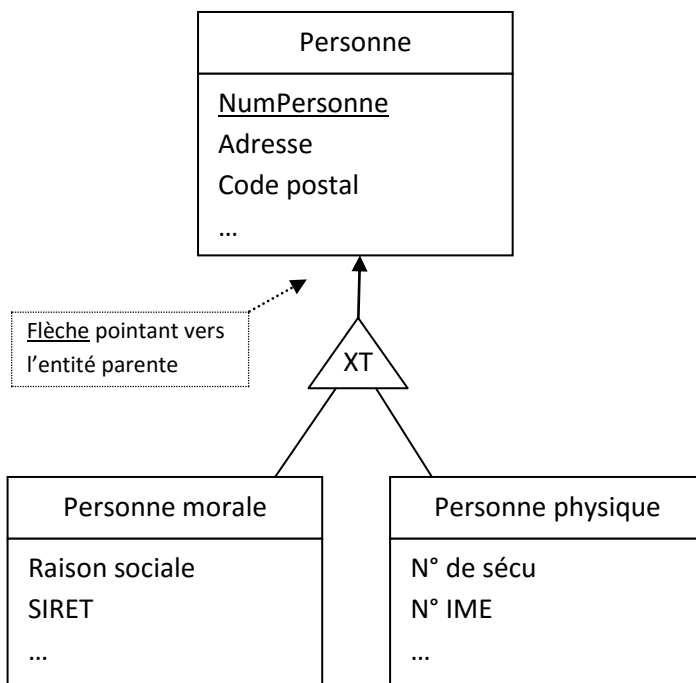
Réservation(#N°Hôtel, #Date, #N°Client, NbNuits) ← en pratique, la date n'est pas une clef étrangère

5. Spécialisation

La spécialisation est concept également appelé héritage, seconde dénomination qui nous vient entre autre de la POO et de l'UML (termes évoqués au début). En ce qui concerne les bases de données relationnelles, La spécialisation consiste à regrouper les propriétés communes d'entités semblables au sein d'une même entité, appelée entité générique (ou entité parente). Les entités semblables sont appelées entités spécialisées (ou entités filles). Il a plusieurs types de spécialisation : rien (vide), X, T, XT ou +.

Il n'y a qu'une façon de modéliser la spécialisation sous forme de MCD (au moyen d'un triangle reliant les entités filles à l'entité parente). Il y a en pratique plusieurs manières (essentiellement 4) de modéliser la spécialisation sous forme de schéma relationnel. Nous n'en retiendrons qu'une.

Quand est-on (doit-on) être tenté d'utiliser une pseudo-entité ? Lorsque l'on besoin de deux entités, lorsque ces entités sont similaires et ne diffèrent que par quelques propriétés ou ont beaucoup de propriétés communes (i.e., ces entités sont sémantiquement proches).



Les types de spécialisation :

o **X** (exclusivité) : équivaut à un OUX logique (ou exclusive), voir fiche algorithmique.

Description : on est l'une, l'autre ou aucune, mais pas les deux à la fois.

o **T** (totalité) : équivaut au OU logique.

Description : on est l'une, l'autre, les deux, mais par aucune des deux.

o **XT** ou **+** (partition) : totalité + exclusivité

Description : on est l'une, l'autre (et donc ni les deux, ni aucune des deux).

o **vide** : tout et n'importe quoi

Description : on est l'une, l'autre, les deux ou aucune des deux.

Par l'une, l'autre... on entend : l'une, l'autre... des entités filles.

Schéma relationnel : notez que, dans le MCD, les clefs primaires des entités filles sont sous-entendues. Il ne faut pas les préciser dans le MCD ! Ci-dessous, c'est une modélisation possible de la spécialisation, celle que vous devez connaître. Elle n'est guère utilisée en pratique en raison de son inefficacité.

Personne(NumPersonne, Adresse, CodePostal, ...)

PersonneMorale(#NumPersonne, RaisonSociale, SIRET, ...)

PersonnePhysique(#NumPersonne, NumSecu, NumIME, ...)

Requêtes utilisant la spécialisation :

Liste des personnes morales :

SELECT * FROM Personne NATURAL JOIN PersonneMorale

Résultat équivalent :

SELECT * FROM Personne AS P, PersonneMorale AS M WHERE P.NumPersonne = M.NumPersonne

Toutes les personnes (cas typique d'utilisation de l'opérateur ensembliste UNION) :

SELECT NumPersonne, RaisonSociale, SIRET, " ", " " FROM PersonneMorale

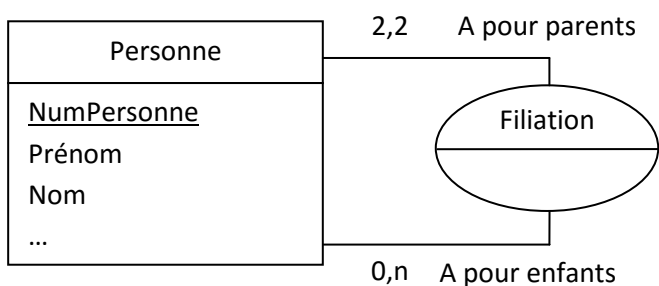
UNION

SELECT NumPersonne, " ", " ", NumSecu, NumIME FROM PersonnePhysique

6. Réflexivité

En matière de bases de données, la réflexivité est un concept nécessaire afin de produire une structure arborescente, une structure récursive. En d'autres termes, la réflexivité permet de modéliser des hiérarchies, des relations d'antécédence. Pour exemple, l'on ne veut prendre que celui de l'arbre généalogique. Un arbre généalogique représente des liens de filiations entre personnes. Alors pourquoi aurait-on besoin d'une autre entité que l'entité personne ?

Avec la réflexivité, on retrouve une fois encore les concepts de CIF et de CIM. Seulement, au lieu d'avoir deux et plus entités reliées entre elles, l'on a une entité reliée à elle-même. Cette fois-ci, on est contraint de préciser le rôle de chacun des pattes de l'association réflexive. Dans un arbre généalogique, une personne a le rôle du parent, l'autre de l'enfant. Les deux sont bien des personnes. Elles n'ont juste pas le même rôle. C'est bien comme s'il y avait deux entités, mais deux fois la même.



Comment s'imaginer ça ? On peut se dire que tout se passe comme s'il y avait deux tables : la table « Parent » (patte « A pour parents ») et la table « Enfant » (patte « A pour enfants »). Mais on a réuni les deux tables dans une seule.

Attention ! C'est une CIF avec une cardinalité 2,2. Elle se comporte donc comme deux CIF 1,1. A savoir, cela fait 2 clefs étrangères.

Schéma relationnel : on notera que « Filiation » correspond à une CIF, « Filiation » ne correspond donc pas à une table, mais, en l'occurrence à deux clefs étrangères (cardinalité 2,2 et non 0,1 ou 1,1).

Personne(NumPersonne, #Pere, #Mere, Prénom, Nom, ...)

Ci-dessus, les champs « Pere » et « Mere » sont chacun une clef étrangère pointant vers une personne. Qu'il s'agissent de parents ou d'enfants, il n'y a finalement ici que des personnes !

Requête utilisant la réflexivité : nombre d'enfants de chaque personne. On notera qu'on est obligé d'utiliser deux fois la table « Personne », qu'on est obligé de préciser des alias.

```
SELECT Parent.Prenom, Parent.Nom, count(NumPersonne) AS [ Nb enfants ] ← on souhaite le nombre de
FROM Personne AS Parent, Personne AS Enfant ← Tous les parents/enfants
WHERE Enfant.Pere = Parent.NumPersonne ← Relie le père à son enfant (s'il en a)
OR Enfant.Mere = Parent.NumPersonne ← Ou relie la mère à son enfant (si elle en a)
GROUP BY Parent.NumPersonne ← On souhaite le nombre d'enfants par personne/parent
```

N.B. : on aurait aussi bien pu mettre count().*

7. Contrainte d'association

Les CIF et les CIM ne sont pas les seules contraintes pesant sur une base de données relationnelle. Il en existe de nombreuses. On parle de contraintes, ou plus exactement de contraintes d'intégrité référentielle, car, lorsque l'on crée une base de données, on cherche à pouvoir stocker des données intègre et conforme au référentiel. Cela signifie que l'on cherche à avoir des données conformes au bon sens et aux règles de gestion imposées. Il est par exemple absurde que l'on puisse ajouter dans un plan comptable le compte « 101BIGUP » ou « 123456789123456 » car : on ne peut pas créer de compte auxiliaire (101BIGUP) sur un compte de classe 1 ; tout n° de compte comporte au maximum 13 caractères (123456789123456, 15 caractères).

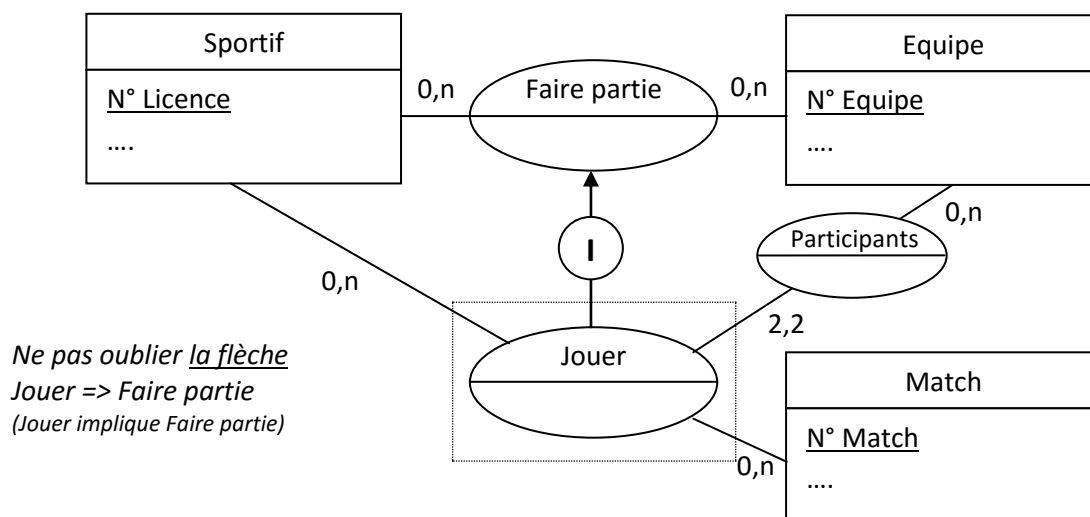
Certaines contraintes, à l'instar de celles citées précédemment, ne peuvent guère être modélisées sur un MCD ou un schéma relationnel. Néanmoins, elles peuvent néanmoins effectivement être implémentées (mises en œuvre) directement sur une base de données relationnelle pour la plupart ou peuvent toutes être implémentées sur un logiciel ayant recours à une base données (on parle de contrôle applicatif).

En revanche, il existe encore une catégorie de contrainte que l'on peut modéliser sur un MCD : les contraintes d'associations. Les contraintes d'association sont sans impact sur le schéma relationnel.

Contrainte d'inclusion : le terme « inclusion » est peu intuitif. Une telle contrainte symbolise une « implication ». Elle représente fait que, pour que l'on participe à une relation, il faille qu'on apparaisse dans une autre relation.

Autrement dit, cela représente la contrainte suivante : on souhaite qu'une ligne ne puisse apparaître dans la table correspondant à une association que si tout ou partie de sa clef primaire apparaît dans la table correspondant à une autre association.

Dans l'exemple ci-dessous, la contrainte d'inclusion représente le fait qu'un sportif ne puisse jouer un match que s'il fait partie de l'une des équipes s'affrontant. Ce qui signifie : pour qu'une couple (N°Licence, N°équipe) apparaissent dans la table « Jouer », il faut que ce même couple apparaisse dans la table « Faire partie ».



Contrainte d'exclusion « X » (exclusivité) : c'est le même « X » que celui de la spécialisation. Une contrainte d'exclusion, placée entre deux associations, signifie qu'on ne puisse pas participer aux deux relations à la fois (l'un, l'autre ou aucune).

Contrainte de totalité « T » (totalité) : c'est le même « T » que celui de la spécialisation. Une contrainte de totalité, placée entre deux associations, signifie qu'on doit absolument participer à l'une ou l'autre des deux associations (l'une, l'autre ou les deux).

Contrainte de partition « XT » ou « + » (partition) : ce sont les mêmes « XT » et « + » que ceux de la spécialisation. Une contrainte de partition, placée entre deux associations, signifie qu'on ne puisse ni participer aux deux relations à la fois ni à participer à aucune (l'une, l'autre).

Contrainte de partition « = » (simultanéité) : Une contrainte de simultanéité, placée entre deux associations, signifie qu'on doivent absolument participer à l'une et à l'autre des associations (les deux). Cela revient à un « ET » logique (voir fiche algorithmique).