

Fiche d'information

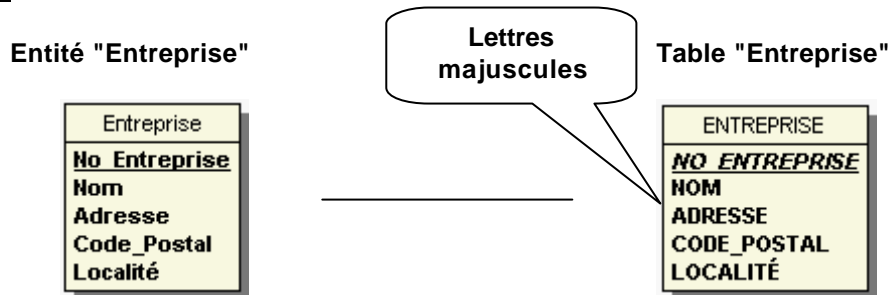
RÈGLES DE TRANSFORMATION DU MCD AU MLD (MRD)

Nous allons définir les règles de transformation pour le passage du MCD au MLD, en respectant les différents cas qui se posent.


Transformation des entités

 Toute entité est transformée en table. Les propriétés de l'entité deviennent les attributs de la table. L'identifiant de l'entité devient la clé primaire de la table.

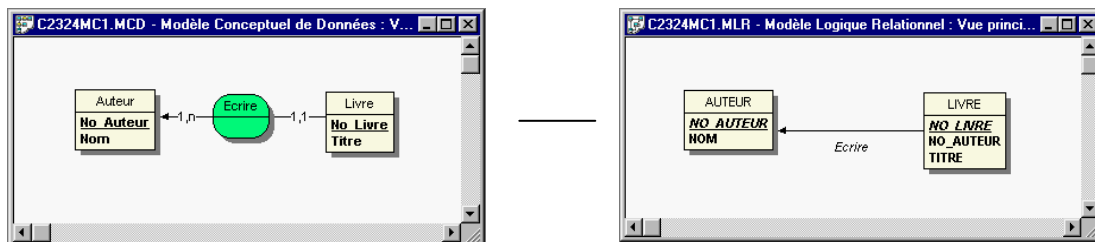
Exemple:



Transformation des relations binaires du type¹ (x,n) – (x,1)

 Afin de représenter la relation, on duplique la clé primaire de la table basée sur l'entité à cardinalité (x,n) dans la table basée sur l'entité à cardinalité (x,1). Cet attribut est appelé clé étrangère. Les deux tables sont liées par une flèche nommée selon la relation, qui pointe de la table à clé étrangère vers la table qui contient la clé primaire correspondante.

Exemple:



L'attribut *No_Auteur* qui est clé primaire de la table *Auteur*, devient clé étrangère dans la table *Livre*.

¹ x peut prendre les valeurs 0 ou 1

Fiche d'information

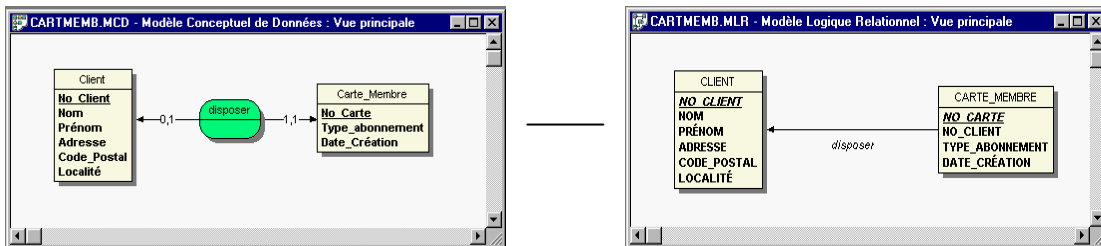
Transformation des relations binaires du type (x,1) – (x,1)

Nous devons distinguer plusieurs cas. Sachant qu'une relation binaire du type (1,1)-(1,1) ne doit pas exister il nous reste les 2 cas suivants:

Relation binaire (0,1)-(1,1)

⚠ On duplique la clé de la table basée sur l'entité à cardinalité (0,1) dans la table basée sur l'entité à cardinalité (1,1).

Exemple :

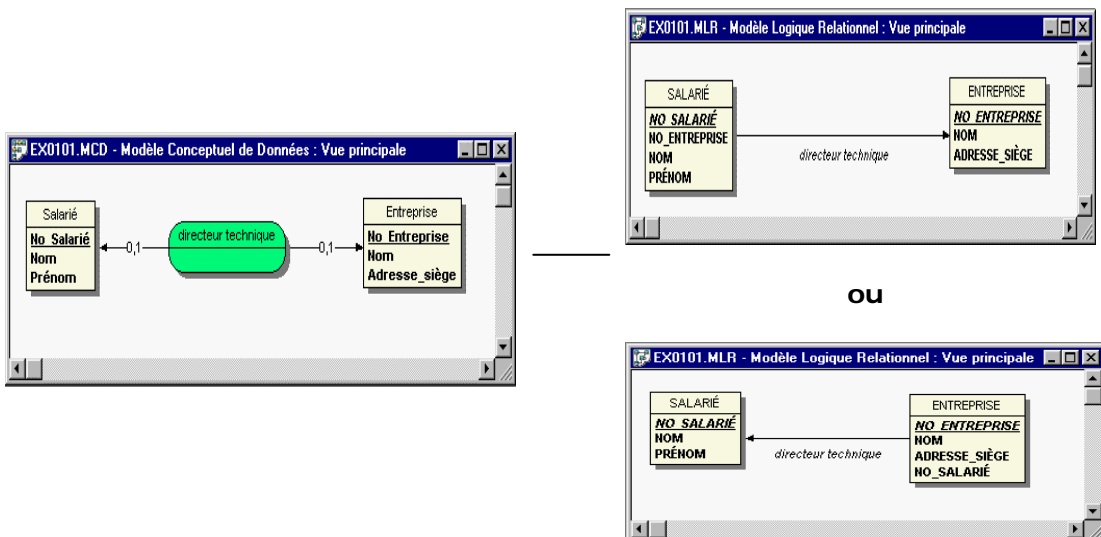


Le *No_Client*, qui est clé primaire de la table *Client*, devient clé étrangère dans la table *Carte_Membre*.

Relation binaire (0,1)-(0,1)

⚠ On duplique la clé d'une des tables dans l'autre. Lorsque la relation contient elle-même des propriétés, celles-ci deviennent également attributs de la table dans laquelle a été ajoutée la clé étrangère.

Exemple :



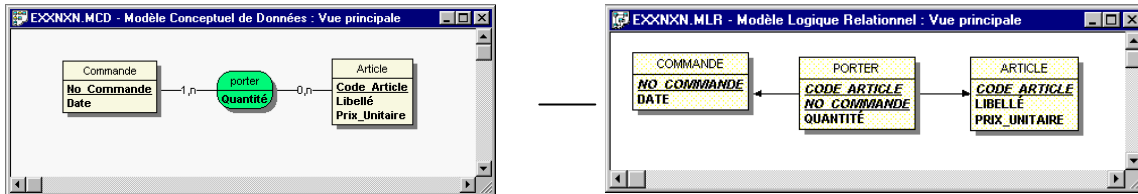
Soit on migre la clé primaire de la table *Entreprise* dans la table *Salarié*, soit on fait l'inverse.

Fiche d'information

Transformation des relations binaires du type (x,n) – (x,n)

⚠ On crée une table supplémentaire ayant comme clé primaire une clé composée des clés primaires des 2 tables. Lorsque la relation contient elle-même des propriétés, celles-ci deviennent attributs de la table supplémentaire. Une propriété de la relation qui est soulignée devra appartenir à la clé primaire composée de la table supplémentaire.

Exemple :

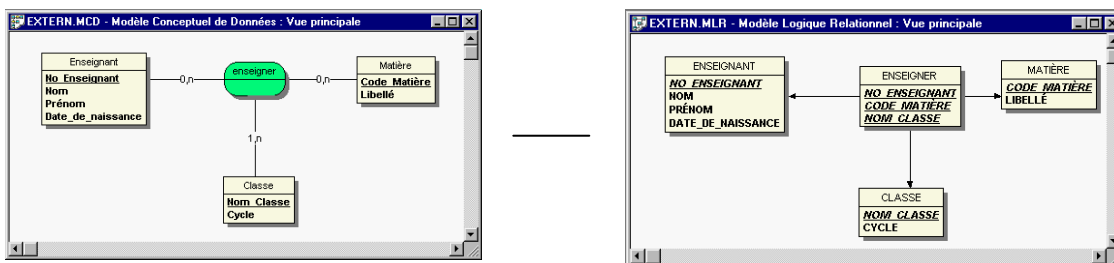


On crée une table *Porter*, qui contient comme clé primaire une clé composée de *No-Commande* et *Code_Article*. Elle contient également la propriété *Quantité* issue de la relation *Porter*.

Transformation des relations ternaires

⚠ On crée une table supplémentaire ayant comme clé primaire une clé composée des clés primaires de toutes les tables reliées. Cette règle s'applique de façon indépendante des différentes cardinalités. Lorsque la relation contient elle-même des propriétés, celles-ci deviennent attributs de la table supplémentaire. Une propriété de la relation qui est soulignée devra appartenir à la clé primaire composée de la table supplémentaire.

Exemple :

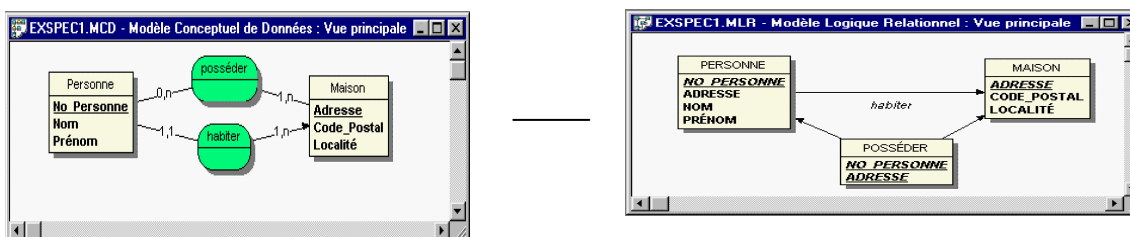


La table *Enseigner* contient une clé composée de *No_Enseignant*, *Code_Matière* et *Nom_Classe*.

Transformation de plusieurs relations entre 2 entités

⚠ Les règles générales s'appliquent

Exemple :



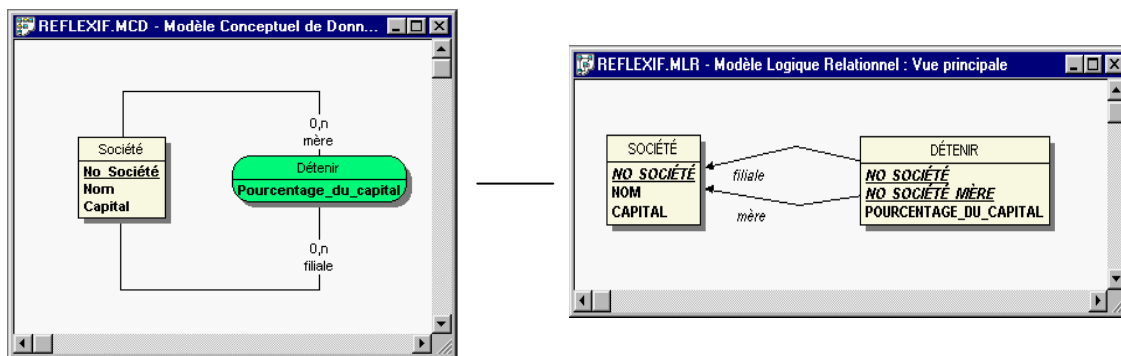
Fiche d'information

La relation *habiter* du type (x,n)-(x,1), est traduite par la migration de l'attribut *Adresse* dans la table *Personne*. La relation *posséder* du type (x,n)-(x,n) est traduite par la création d'une table supplémentaire du même nom. Cette table contient comme clé primaire composée, les clés des deux tables reliées *Personne* et *Maison*. On a donc simplement appliqué 2 fois de façon indépendante les règles de transfert MCD \rightarrow MLD.

Transformation des relations réflexives

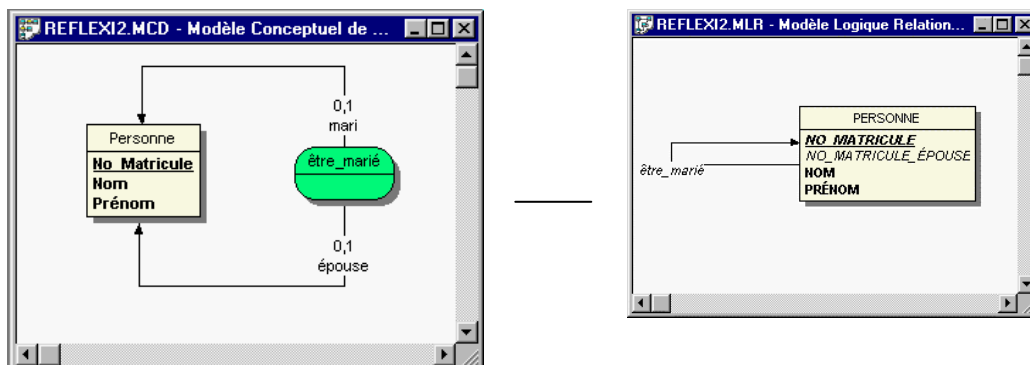
 Nous appliquons les règles générales avec la seule différence que la relation est 2 fois reliée à la même entité

Exemple 1:



Comme il s'agit d'une relation (x,n)-(x,n), une table supplémentaire est créée. Cette table contient comme clé primaire composée, la clé des "deux" entités reliées. Comme la même entité est liée 2 fois à la relation, on ne peut pas utiliser 2 fois le même nom pour la clé. Dans ce cas il convient d'utiliser des rôles dans le MCD, et d'intégrer le rôle dans le nom d'une des clés migrées dans le MLD.

Exemple 2:



Comme il s'agit d'une relation (0,1)-(0,1), nous avons en général le choix en ce qui concerne quelle entité contiendra la clé étrangère. Comme cette relation est liée deux fois à la même entité, il est évident que nous devons dupliquer la clé primaire, tout en veillant que le même nom de clé ne sera pas utilisé pour la clé primaire et la clé étrangère. Dans notre exemple, tous les hommes mariés, ont comme valeur de la clé étrangère la matricule de leur épouse actuelle. Pour les hommes non mariés et les femmes, la clé étrangère est sans valeur. On pourrait bien sûr utiliser la modélisation inverse avec une clé étrangère *NO_MATRICULE_MARI*, qui indique pour chaque femme mariée, la matricule de son mari.

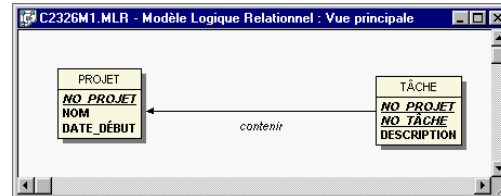
Transformation de l'identifiant relatif

Fiche d'information



Sachant que l'entité dépendante est toujours liée à la relation par les cardinalités (1,1), nous pouvons appliquer les règles générales. Dans chaque cas, la table issue de l'entité dépendante contient donc comme clé étrangère, la clé primaire de l'autre table. L'identification relative est représentée par le fait que la table issue de l'entité dépendante contient une clé primaire composée, constituée de la clé primaire transformée de l'identifiant de cette entité et de la clé étrangère.

Exemple :



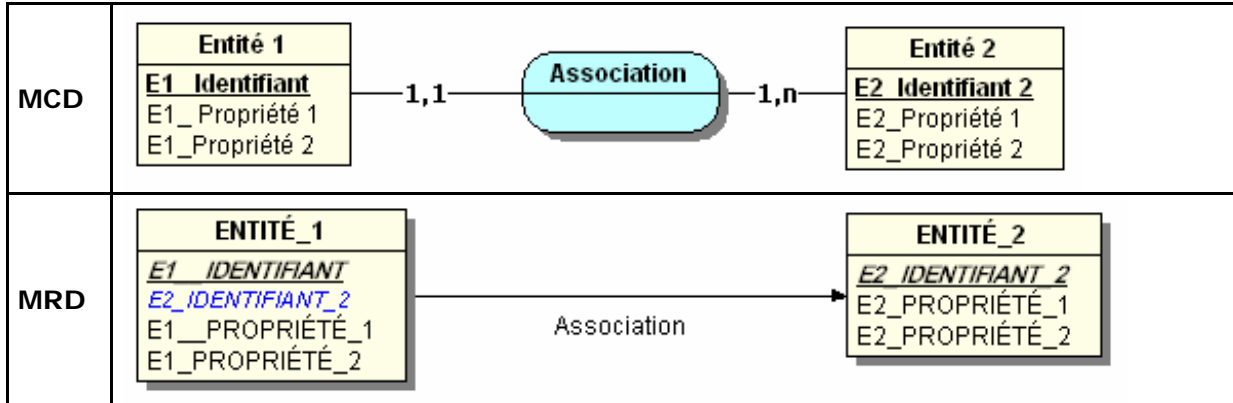
Tout en respectant les règles générales du passage MCD \rightarrow MLD, la clé primaire de la table *Projet* migre comme clé étrangère dans la table *Tâche*. L'identification relative est représentée par le fait que la table tâche contient une clé primaire composée de *No_Tache* et *No_Projet*.

Source : P. Stockreiser. Lycée du Nord Wiltz - Lux.

Fiche d'information

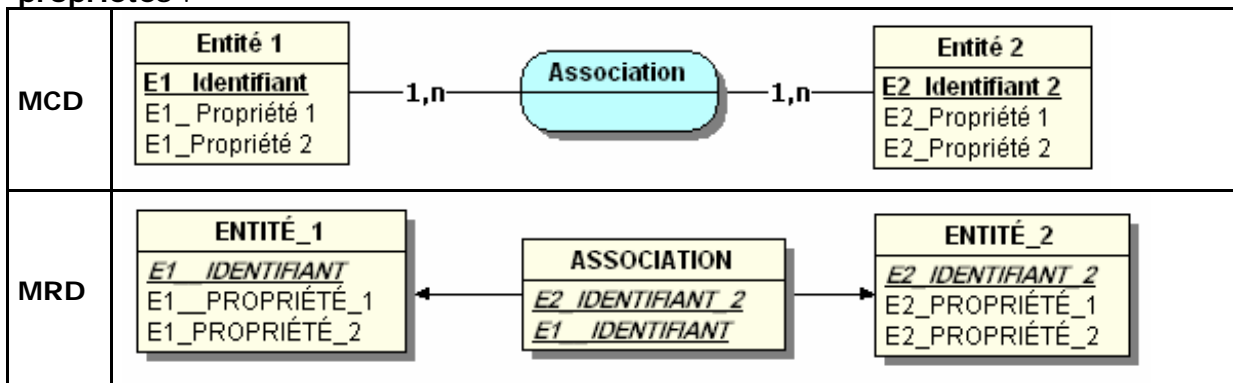
RÉCAPITULATIF :

Représenter une association binaire 1,1 - 1,n



ENTITE_1(E1 Identifiant, #E2 Identifiant, E1-Propriété_1, E1-Propriété_2)
 ENTITE_2(E2 Identifiant 2, E2_Propriété_1, E2_Propriété_2)

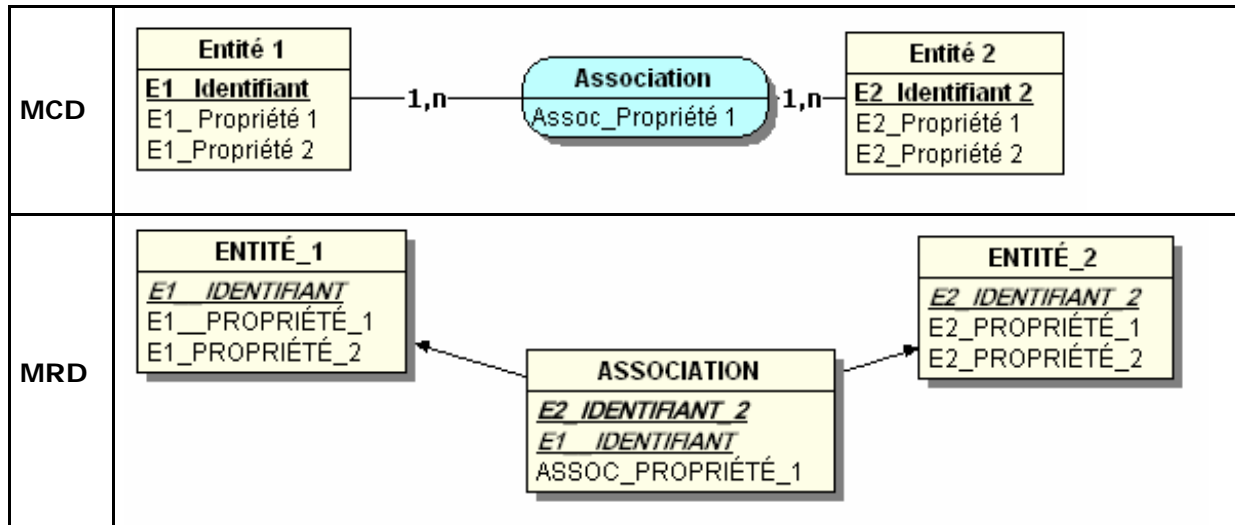
Représenter une association binaire (0 ou 1) , n - (0 ou 1), n non porteuse de propriétés :



ENTITE_1(E1 Identifiant, E1-Propriété_1, E1-Propriété_2)
 ENTITE_2(E2 Identifiant 2, E2_Propriété_1, E2_Propriété_2)
 ASSOCIATION(#E2 Identifiant 2, #E1 Identifiant 1)

Fiche d'information

Représenter une association binaire 1,n - 1,n porteuse de propriétés :

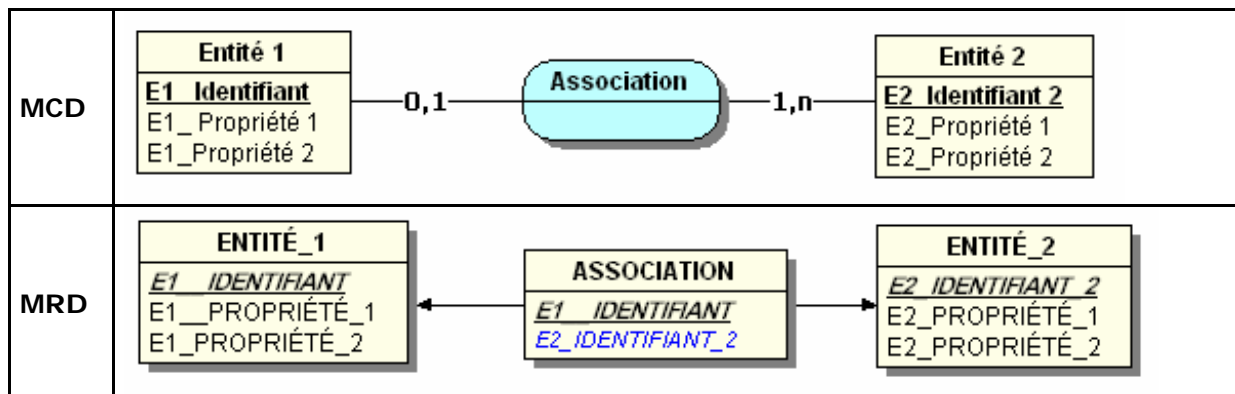


ENTITE_1(E1 Identifiant, E1-Propriété_1, E1-Propriété_2)

ENTITE_2(E2 Identifiant 2, E2_Propriété_1, E2_Propriété_2)

ASSOCIATION(#E2 Identifiant 2, #E1 Identifiant 1, Assoc Propriété 1)

Représenter une association binaire 0,1 - 1,n :



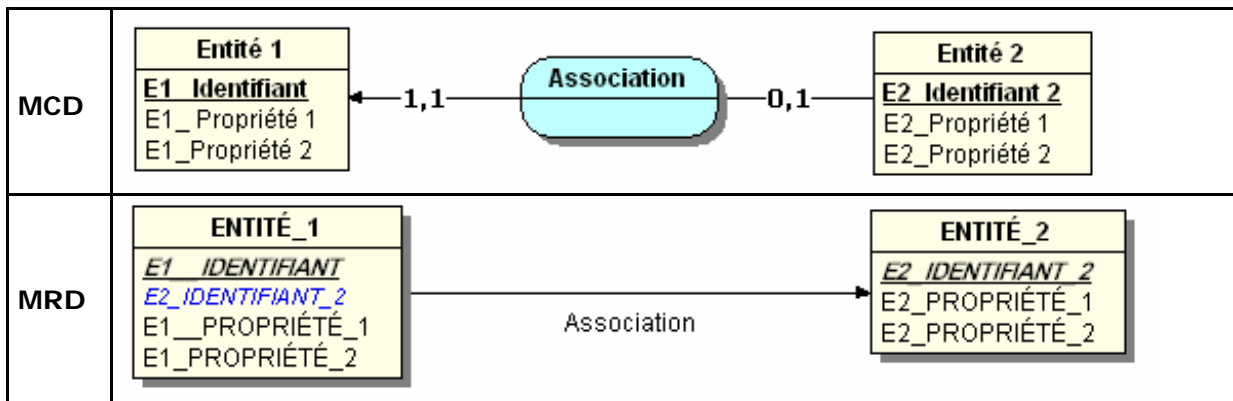
ENTITE_1(E1 Identifiant, E1-Propriété_1, E1-Propriété_2)

ENTITE_2(E2 Identifiant 2, E2_Propriété_1, E2_Propriété_2)

ASSOCIATION(#E2 Identifiant 2, #E1 Identifiant 1)

Fiche d'information

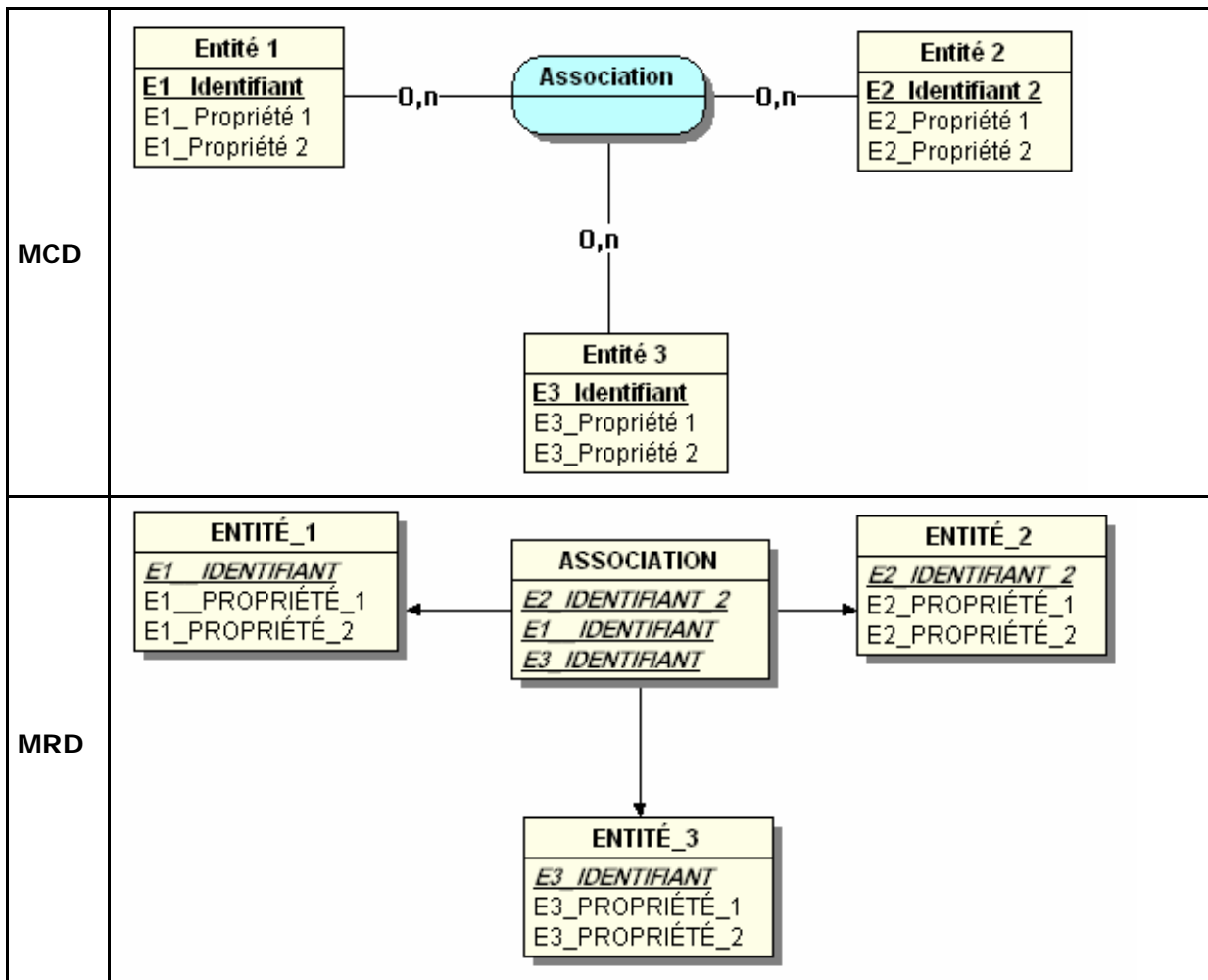
Représenter une association binaire 1,1 - 0,1 :



ENTITE_1(E1 Identifiant, #E2 Identifiant 2, E1-Propriété_1, E1-Propriété_2)

ENTITE_2(E2 Identifiant 2, E2-Propriété_1, E2-Propriété_2)

Représenter une association ternaire 0,n - 0,n - 0,n :



ENTITE_1(E1 Identifiant, E1-Propriété_1, E1-Propriété_2)

ENTITE_2(E2 Identifiant 2, E2-Propriété_1, E2-Propriété_2)

ENTITE_3(E3 Identifiant, E3-Propriété_1, E3-Propriété_2)

ASSOCIATION(#E2 Identifiant 2, #E1 Identifiant 1, #E3 Identifiant)