

Récapitulatif des Règles de vérification et de normalisation d'un MCD

R1 - Existence d'un identifiant pour chaque objet.

R2 - Pour chaque occurrence d'une entité, chaque propriété ne peut prendre qu'une seule valeur. Autrement dit, on ne peut avoir de valeurs répétitives pour une même propriété.

R3 - Toutes les propriétés doivent être élémentaires, c'est-à-dire non décomposables. (1FN)

R4 - Toutes les propriétés autres que l'identifiant doivent dépendre pleinement et directement de l'identifiant. (2FN et 3FN).

R5 - Si une entité a un identifiant concaténé, un des éléments composant cet identifiant ne doit pas dépendre d'une autre propriété.

R6 - A chaque occurrence d'une relation correspond une et une seule occurrence de chaque objet participant à la relation, d'où deux sous-règles :

- a- 2 occurrences d'un objet ne peuvent participer à une même occurrence de relation.
- b- Pour une occurrence de relation il n'y a pas de participation optionnelle d'un objet.

R7 - Pour chaque occurrence d'une relation il ne peut exister qu'une et une seule valeur pour chaque propriété de la relation.

R8 - Toutes les propriétés d'une relation doivent dépendre pleinement de l'identifiant de la relation ; ainsi, chaque propriété doit dépendre de tout l'identifiant et non pas d'une partie de cet identifiant.

R9 - Il ne doit pas y avoir de redondance par transitivité des relations, ni des propriétés.

R10 - Une relation de dimension supérieure à 2, dont une branche a des cardinalités de type (0,1) ou (1,1), doit être éclatée en relations de dimensions inférieures.

Contrainte d'intégrité fonctionnelle (CIF)

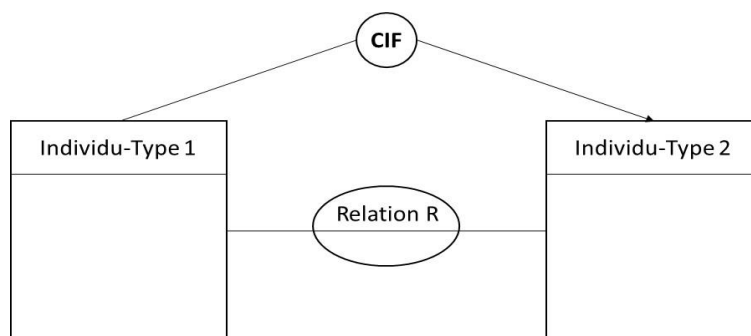
Définition :

Une Contrainte d'Intégrité Fonctionnelle (CIF) se définit par le fait que l'une des individus-types participant à la relation est complètement déterminée par la connaissance d'une ou plusieurs autres individus-types participant dans cette même relation.

Représentation :

Exprime qu'à partir d'une occurrence d'une entité, lui correspond (au plus) une seule occurrence de l'autre entité.

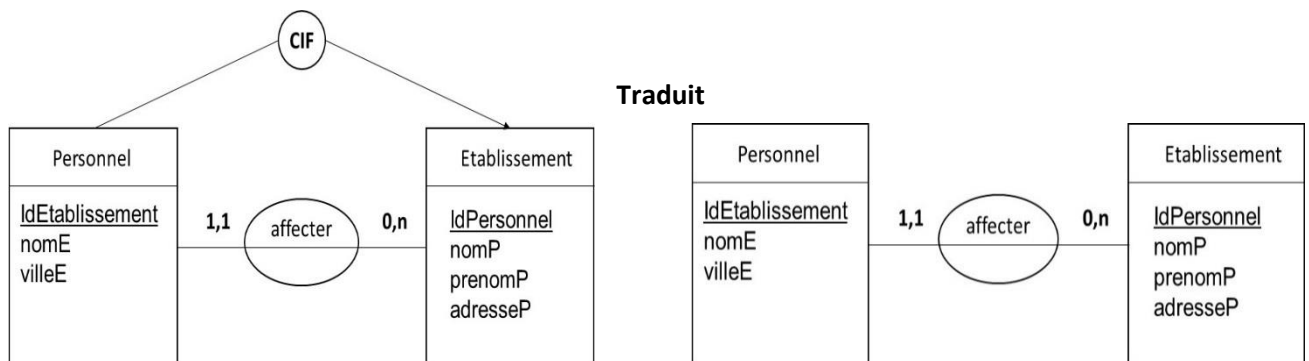
- La cardinalité maximum=1
- et l'existence d'une dépendance. Ces relations seront couramment appelées relations fonctionnelles.



3. CIF sur une relation binaire

Exemple :

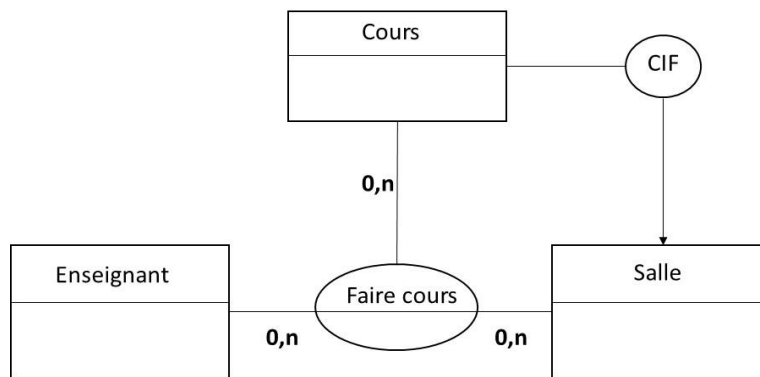
- 1 membre du personnel est affecté à 1 seul établissement (affecter ne renvoie qu'une valeur)
- Le sens de la flèche traduit une dépendance de Personnel envers Etablissement



4. CIF sur une relation n-aire

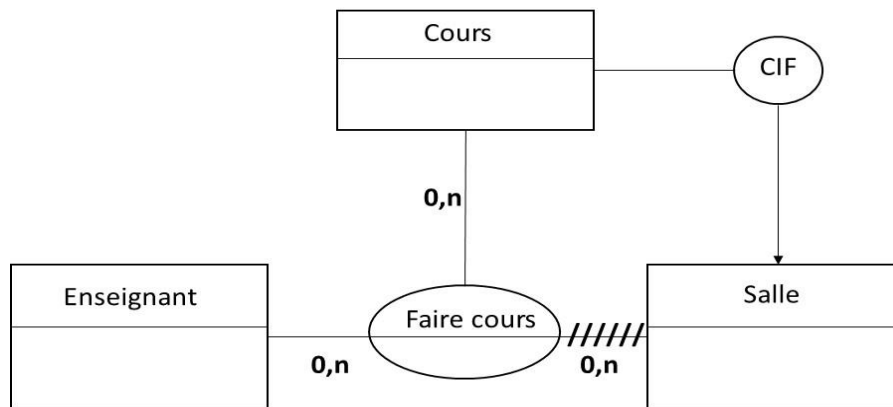
a. CIF n'englobant pas la totalité de la collection :

- Un cours ne peut avoir lieu que dans une seule salle
- Cours → Salle
- L'enseignant n'étant pas compris dans cette contrainte peut faire plusieurs cours dans la même salle...
- ... ou pas : 2 cours dans 2 salles différentes
- Mais dans tous les cas 1 seule salle pour 1 cours

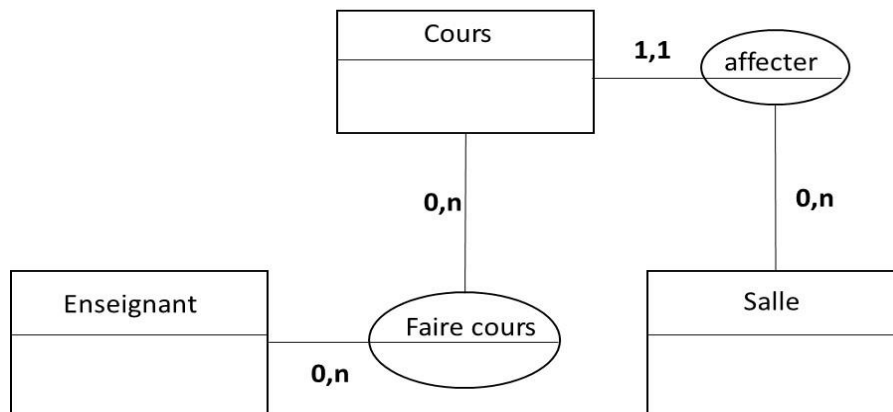


- L'affectation de la salle est indépendante de l'enseignant ...
- On peut décomposer « faire cours » en " assurer le cours " et « trouver une salle » pour le cours

➡ Réduire la dimension des relations et les rendre plus facilement interprétables.

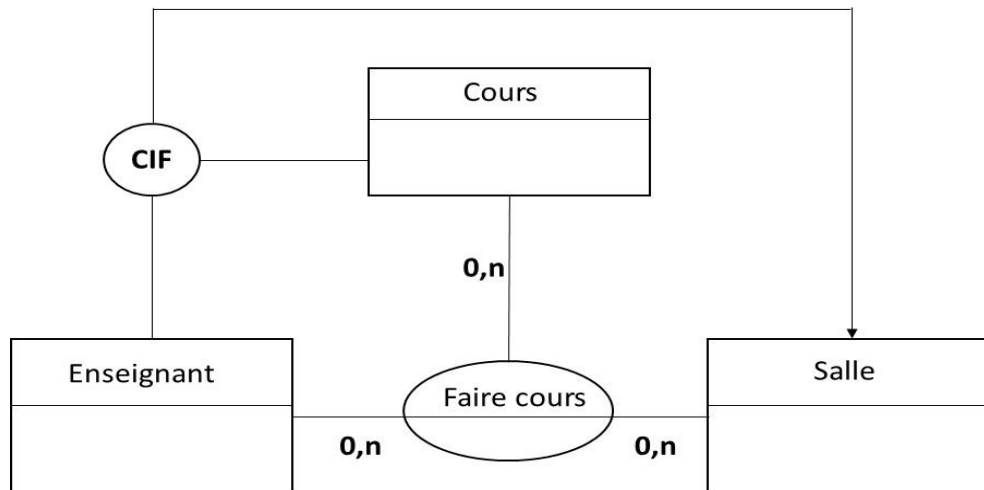


Après la décomposition



b. CIF englobant la totalité de la collection :

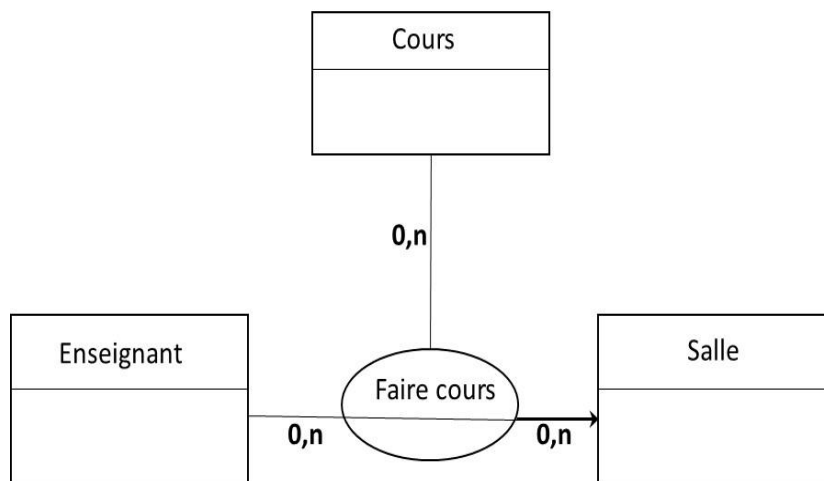
- Un enseignant ne fait 1 cours que dans une seule salle
 - En d'autres termes à 1 (enseignant, cours) ne correspond qu'une seule salle
- Mais :
- Un enseignant peut faire 2 cours différents dans des salles différentes
 - 2 enseignants peuvent faire le même cours dans 2 salles différentes



Traduction Fonctionnelle : Dans le cadre de la relation faire_cours:

- (Enseignant , Cours) \rightarrow Salle
- Donc je ne puis avoir (Enseignant , Cours, Salle) & (Enseignant, Cours, Salle') vrais simultanément
- (Enseignant , Cours) ne doit être associé qu'à 1 seul Salle.

Après la décomposition



La relation *Faire cours* ne se décompose pas, il faut mettre la flèche pour qu'on puisse garder la dépendance : Enseignant, Cours \rightarrow Salle. Et dans le MLD on va avoir :

Faire_cours (Id_enseignant, Id_Cours, Id_salle)

Modèle logique de données (MLD):

Règle de passage vers le Relationnel

Après avoir conçu le Modèle Conceptuel de Donnée (MCD), il est maintenant temps de le transposer en Modèle Logique de Données Relationnelles (MLDR). Il s'agit du passage entre le Modèle Conceptuel de Données et l'implémentation physique de la base. Le MLD est lui aussi indépendant du matériel et du logiciel, il ne fait que prendre en compte l'organisation des données. C'est d'ailleurs le point primordial de la modélisation : si l'organisation des données est relationnelle (si elles sont "liées" entre elles), alors le MLD est relationnel et devient le MLDR, ou Modèle Logique de Données Relationnelles. Nous définissons les 6 règles nécessaires pour passer d'un MCD à un MLDR.

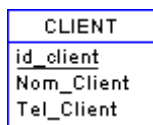
Vocabulaire :

Les données sont stockées dans des relations. Une relation est un ensemble de T-uple, et un T-uple est défini par un ou plusieurs attributs. Dans la pratique, la relation est en fait la table, un T-uple est une ligne (ou enregistrement), et les attributs sont les colonnes.

Règle1 :

Toute entité du MCD devient une relation du MLDR, et donc une table de la Base de données. Chaque propriété de l'entité devient un attribut de cette relation, et dont une colonne de la table correspondante. L'identifiant de l'entité devient la Clé Primaire de la relation (elle est donc soulignée), et donc la Clé Primaire de la table correspondante.

Exemple : Modèle Conceptuel de Données (MCD) :



Modèle Logique de Données Relationnelles (MLDR) :

CLIENT (id_client, Nom_Client, Tel_client)

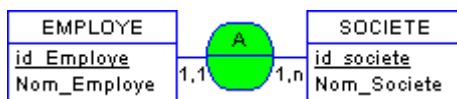
Règle 2 : Relation binaire aux cardinalités (X,1) - (X,n), X=0 ou X=1 (relation père-fils)

La **Clé Primaire** de la table à la cardinalité (X,n) (*Père*) devient une **Clé Etrangère** dans la table à la cardinalité (X,1) (*fils*). Si la relation est porteuse de données, celle-ci devient un attribut dans la table à la cardinalité (X,1) (*fils*).

Exemple de Système d'Information (SI) :

Un employé a une et une seule société. Une société a 1 ou n employés.

Modèle Conceptuel de Données (MCD) :



Modèle Logique de Données Relationnelles (MLDR) :

EMPLOYE (id_Employe, Nom_Employe, id_Societe*)

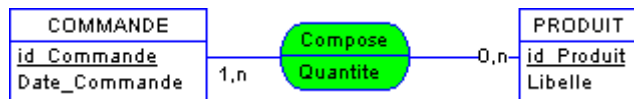
SOCIETE (id_Societe, Nom_Societe).

Règle3 : Relation binaire aux cardinalités (X,n) - (X,n), X=0 ou X=1

Il y a création d'une table supplémentaire ayant comme Clé Primaire une clé composée des *identifiant* des 2 entités. On dit que la Clé Primaire de la nouvelle table est la *concaténation* des Clés Primaires des deux autres tables. Si la relation est porteuse de données, celles-ci deviennent des attributs pour la nouvelle table.

Exemple de SI : Une commande est composée de 1 ou n produits distincts en certaine quantité. Un produit est présent dans 0 ou n commandes en certaine quantité.

MCD :



MLDR :

COMMANDE (id_Commande, Date_commande)

PRODUIT (id_Produit, libelle)

COMPOSE (id_Commande*, id_Produit*, quantité)

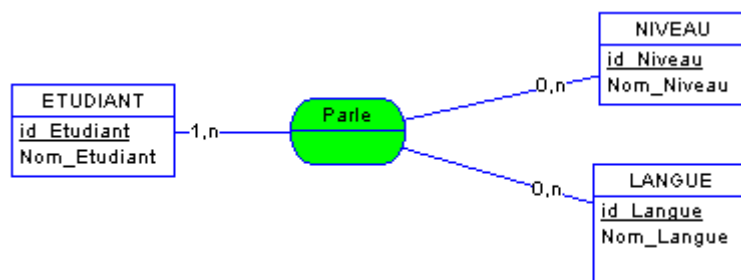
La table COMPOSE servira en pratique à retrouver tous les produits composants chaque commande. De même, pour chaque produit, on pourra retrouver toutes les commandes où il figure.

Règle 4 : Relation n-aire (quelles que soient les cardinalités)

Il y a création d'une table supplémentaire ayant comme Clé Primaire la *concaténation* des *identifiants* des *entités* participant à la *relation*. Si la relation est porteuse de données, celles-ci deviennent des *attributs* pour la nouvelle table.

Exemple de SI : Un étudiant parle une ou plusieurs langues avec un niveau. Chaque langue est donc parlée par 0 ou n étudiants avec un niveau. Pour chaque niveau, il y a 0 ou plusieurs étudiants qui parlent une langue.

MCD :



MLDR :

ETUDIANT (id_Etudiant, Nom_Etudiant)

NIVEAU (id_Niveau, Nom_Niveau)

LANGUE (id_Langue, Nom_Langue)

PARLE (id_Etudiant*, id_Niveau*, id_Langue*)

Chaque enregistrement de la table PARLE a comme clés primaires la concaténation de id_Etudiant, id_Niveau et id_Langue. Chaque enregistrement est donc forcément unique. Ainsi, pour chaque Langue, il est possible de retrouver tous les Etudiants ayant ou non un Niveau précis, de manière unique et distincte.

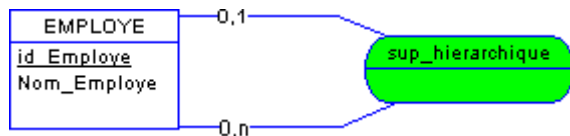
Règle 5 : Association Réflexive

* **Premier cas** : cardinalité (X,1) - (X,n), avec X=0 ou X=1

La Clé Primaire de l'entité se dédouble et devient une Clé Etrangère dans la relation qui devient elle-même une nouvelle table. Exactement comme si l'entité se dédoublait et était reliée par une relation binaire (X,1) - (X,n) (Cf règle 2).

Exemple de SI : Prenons l'exemple d'une société organisée de manière pyramidale : chaque employé a 0 ou 1 supérieur hiérarchique direct. Simultanément, chaque employé est le supérieur hiérarchique direct de 0 ou plusieurs employés.

MCD:



MLDR :

EMPLOYE (id_Employe, Nom_Employe, id_Sup_Hierarchique*)

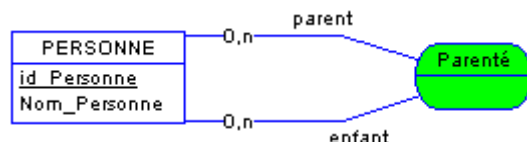
id_Sup_Hierarchique* est l'identifiant (id_Employe) du supérieur hiérarchique direct de l'employé considéré.

* **Deuxième cas** : cardinalité (X,n) - (X,n), avec X=0 ou X=1

De même, tout se passe exactement comme si l'entité se dédoublait et était reliée par une relation binaire (X,n) - (X,n) (Cf règle 3). Il y a donc création d'une nouvelle table.

Exemple de SI : Prenons cette fois l'exemple d'une organisation de type familiale : chaque personne a 0 ou n ascendants directs (parents), et a aussi 0 ou n descendants directs (enfants).

MCD :



MLDR :

PERSONNE (id_Personne, Nom_Personne)

PARENTE (id_Parent*, id_Enfant*)

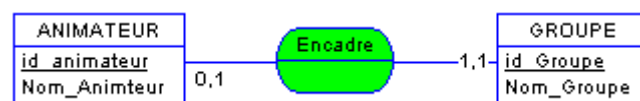
id_Parent* est l'identifiant (id_Personne) d'un ascendant direct de la personne. id_Enfant* est l'identifiant (id_Personne) d'un descendant direct de la personne. La table PARENTE sera en fait l'ensemble des couples (parents-enfants) présent dans cette famille.

Règle 6 : Relation binaire aux cardinalités (0,1) - (1,1)

La **Clé Primaire** de la table à la cardinalité (0,1) devient une **Clé Etrangère** de la table à cardinalité (1,1) :

Exemple de SI : Dans ce centre de vacances, chaque animateur encadre en solo 0 ou 1 groupe, chaque groupe étant encadré par un et un seul animateur.

MCD :



MLDR :

ANIMATEUR (id_Animateur, Nom_Animateur)

GROUPE (id_Groupe, Nom_Groupe, id_animateur*)

