

Le modèle Entité-Association

C'est un modèle important pour la conception des bases de données relationnelles. Il est très répandu, très documenté.

Il aide à concevoir une base de données sans redondance, sans incohérence (si on le construit avec rigueur et réflexion).

Représentation graphique des informations en termes d'*entités* (de groupes d'informations) reliées par des *associations*, il aide à comprendre les relations qui existent entre les informations. C'est un bon support pour expliquer ce que contient une base de données.

Le modèle Entité-Association est utilisé dans la méthode Merise pour représenter le **MCD** (Modèle Conceptuel des Données). Il apparaît dans l'étude préalable (MCD de l'existant et ébauche du MCD de la nouvelle solution) et dans l'étude détaillée (MCD complet de la nouvelle solution).

Pour faire ce schéma, il faut avoir explicité les règles de gestion, avoir construit un dictionnaire des données.

Modèle Entité-Association les objets ou entités

- Une **entité** est la représentation d'un objet matériel ou immatériel pourvu d'une existence propre et conforme aux choix de gestion de l'entreprise.
- Une **occurrence d'une entité** est une instance (un représentant) de l'entité dans le *monde réel*.
- Une **propriété (ou attribut)** est une donnée élémentaire qu'on perçoit sur l'entité.
- L'**identifiant d'une entité** est une propriété (ou un ensemble de plusieurs propriétés) de l'entité telle que, à chaque valeur de l'identifiant, correspond *une et une seule* occurrence de l'entité. Il peut être créé ex nihilo.

un exemple

Les règles de gestion (simplifiées) :

- Les étudiants ont une identité (nom, prénom, adresse...).
- Les enseignants ont une identité.
- Les étudiants suivent des cours mais ne suivent pas tous les cours.
- Les cours sont hebdomadaires et ont une heure de début, une durée, une heure de fin ; les horaires ne doivent pas se superposer.
- Les cours sont assurés par un seul enseignant.

suite de l'exemple

Le dictionnaire des données :

- Nom étudiant, texte, **E**lémentaire, 30 caractères,
- Nom jeune fille, texte, **E**, 30 caractères, vide autorisé,
- Numéro étudiant, texte **?**, **E**, 17 caractères,
- ...
- Intitulé cours, texte, **E**, 70 caractères,
- Jour, numérique, **E**, entier, 1 pour lundi, 2 pour mardi...
- Heure Début, numérique, **E**, réel, $7 \leq hd \leq 20$
- Durée, numérique, **E**, réel, $0 \leq d \leq 5$
- Heure Fin, numérique, **C**alculé, réel, $hf = hd + d$
- ...

suite de l'exemple

ÉTUDIANTS

NoEtudiant

nom

nom jeune fille

prénom

adresse

ENSEIGNANTS

NoEnseignant

nom

prénom

adresse

COURS

NoCours

intitulé

jour

heure début

durée

Règles à vérifier sur les entités (à appliquer avec bon sens)

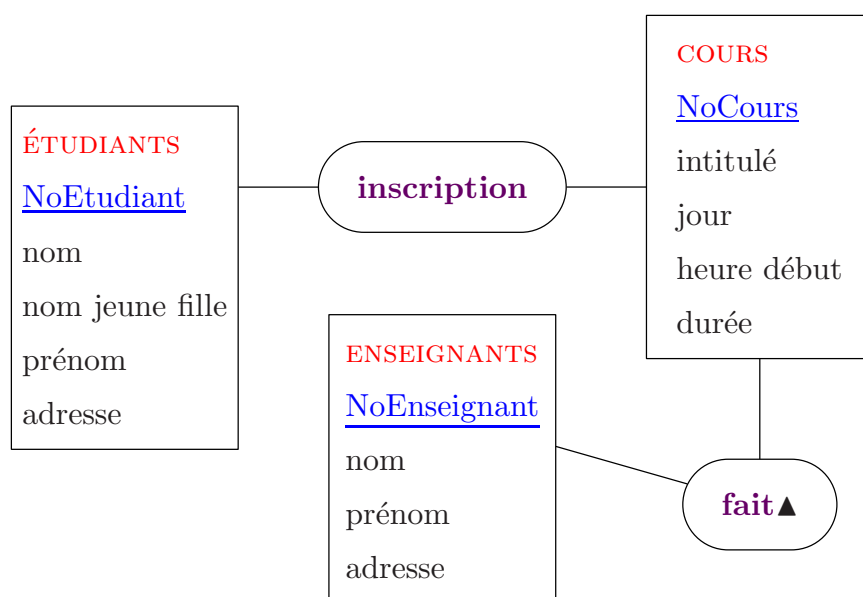
1. Toute propriété est élémentaire ;
2. Une propriété ne doit pas être “instable”, pas “calculable” ;
3. Toute propriété d’une instance aura au plus une valeur ;
4. Une propriété doit permettre d’éviter la redondance des valeurs sur l’ensemble des instances ;
5. Toute entité possède un identifiant ;
6. Toute propriété dépend (directement) de l’identifiant ;
7. Toute propriété dépend de *tout* l’identifiant ;

les associations

- Une **association** est une relation que les règles de gestion établissent entre deux entités (ou plus).
- Une **occurrence d'une association** est une instance de l'association dans le *monde réel*.
- Une association peut posséder des propriétés.

Règles à vérifier sur les associations : règles des entités. L'identifiant d'une association est la concaténation des identifiants des entités qu'elle relie.

suite de l'exemple



les cardinalités

La **cardinalité d'une entité par rapport à une association** s'exprime sous forme d'un couple :

(**cardinalité minimale** : **cardinalité maximale**)

cardinalité minimale : c'est le nombre minimal de fois où une occurrence d'un objet participe aux occurrences de l'association ; elle vaut très souvent **0 ou 1**.

cardinalité maximale : c'est le nombre maximal de fois où une occurrence d'un objet participe aux occurrences de l'association ; elle vaut **1, un entier fixé ou n** ("n" pour indiquer l'impossibilité de fixer une borne maximale).

les cardinalités

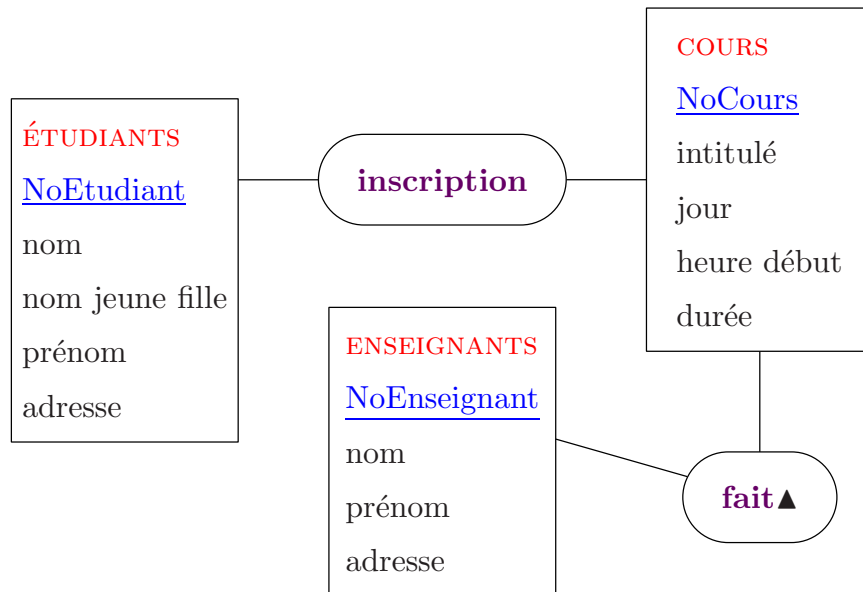
ENTITÉA $\overset{cmA:cMA}{-----} \bigcirc \overset{cmB:cMB}{-----}$ ENTITÉB

cmA : Vous l'obtenez en répondant à la question *"quand je considère un élément de l'entité A, combien d'éléments de l'entité B lui sont obligatoirement associés ?"*.

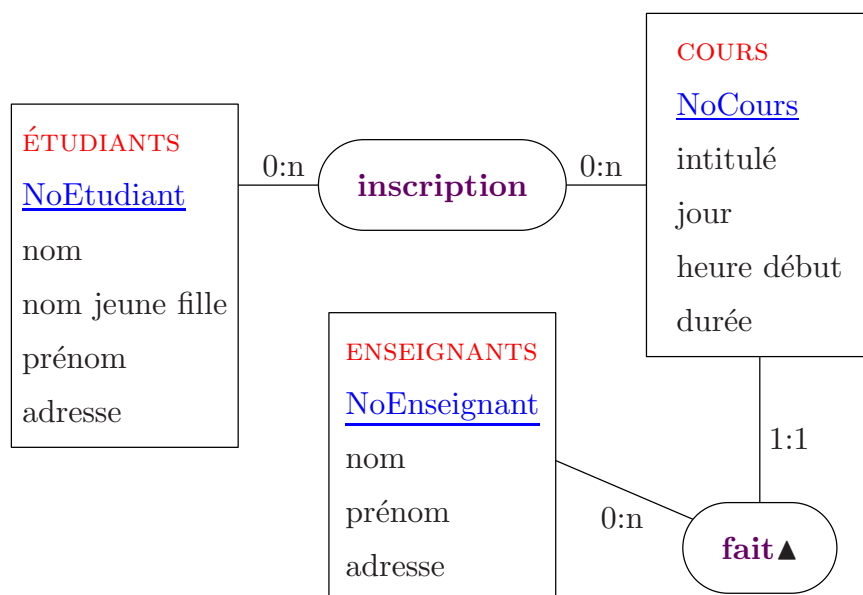
cMA : *"quand je considère un élément de l'entité A, combien d'éléments de l'entité B lui sont associés au maximum ?"*.

cmB et cMB : Symétriquement, *"quand je considère un élément de l'entité B, combien d'éléments de l'entité A lui sont associés, au minimum (cmB) et au maximum (cMB) ?"*.

suite de l'exemple



suite de l'exemple



Vérification

Vérifier :

- les règles sur les entités ;
- les règles sur les associations ;
- les règles globales :
 1. Une propriété ne figure qu'une fois dans le MCD ;
 2. Les propriétés calculées ne figurent pas dans le MCD (mais il faut s'assurer qu'avec le MCD on puisse les calculer) ;
 3. On ne fait pas figurer les associations qui se déduisent par transitivité.

exercice (gestion de commandes)

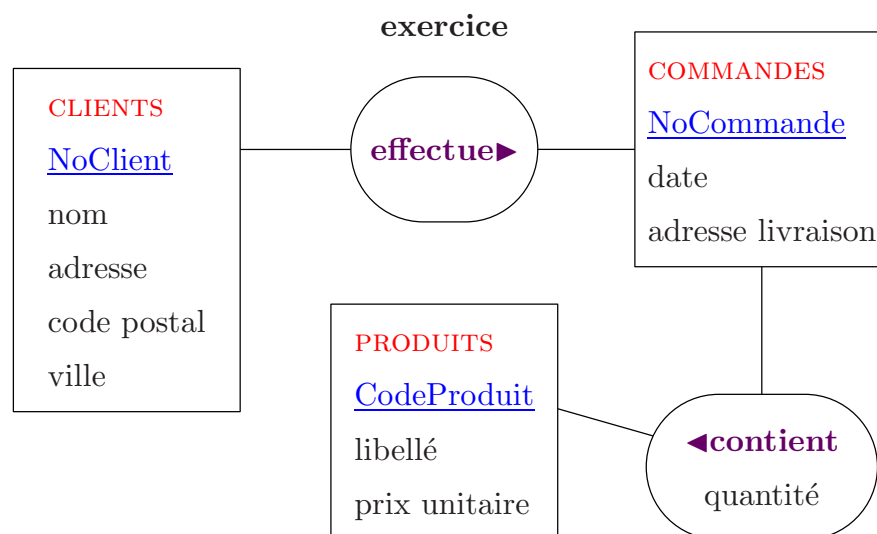
Les règles de gestion :

- Le magasin vend des produits à des clients.
- Les produits possèdent une référence (un code), un libellé et un prix unitaire.
- Les clients ont une identité (nom, prénom, adresse...).
- Les clients passent des commandes de produits. On mémorise la date de la commande.
- Pour chaque commande, le client précise une adresse de livraison.
- La commande concerne un certain nombre de produits, en une quantité spécifiée pour chaque produit.

suite de l'exercice

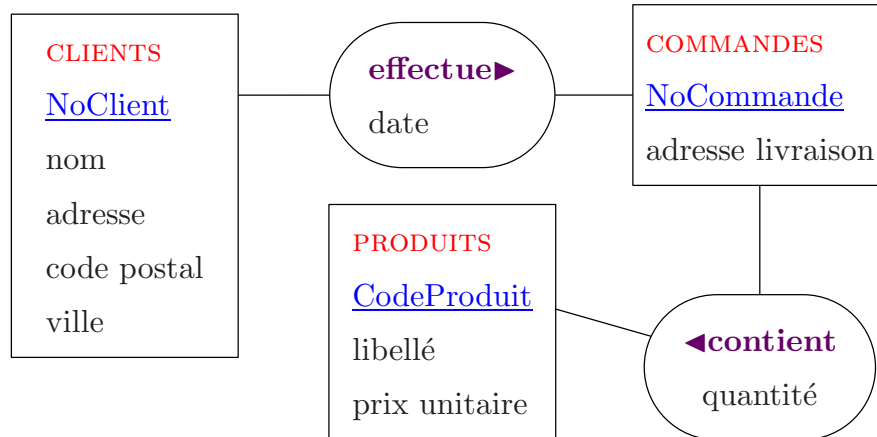
Le dictionnaire des données :

- Code Produit, texte, E, 5 caractères,
- Libellé, texte, E, 40 caractères,
- Prix, numérique, E, réel, $0 \leq \text{prix}$,
- Nom client, texte, E, 30 caractères,
- Adresse, texte, E, 40 caractères,
- ...
- Date Commande, numérique, E, date,
- Adresse livraison, texte, E, 50 caractères,
- Quantité commandée, numérique, E, entier, $0 \leq \text{quantité}$
- Nombre produits commandés, numérique, C, entier,
 $\text{nombre} = \text{compte}(\text{Produits commandés})$



exercice

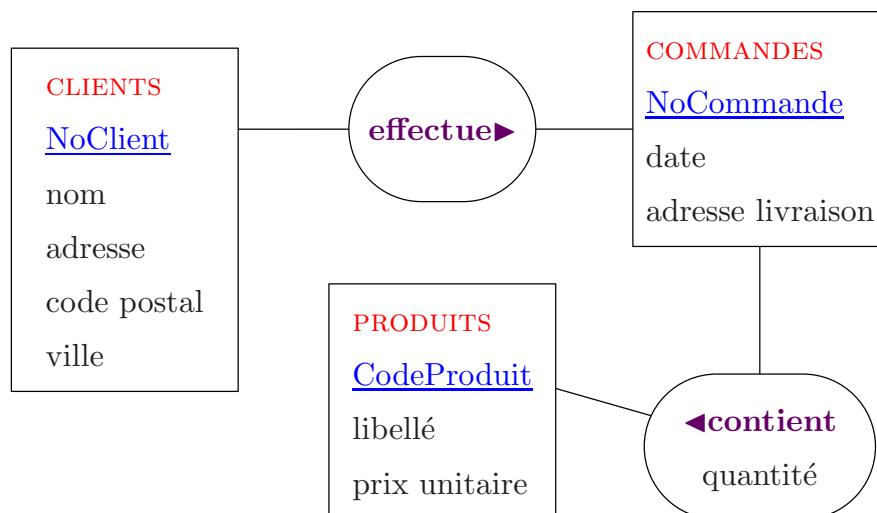
On aurait pu faire



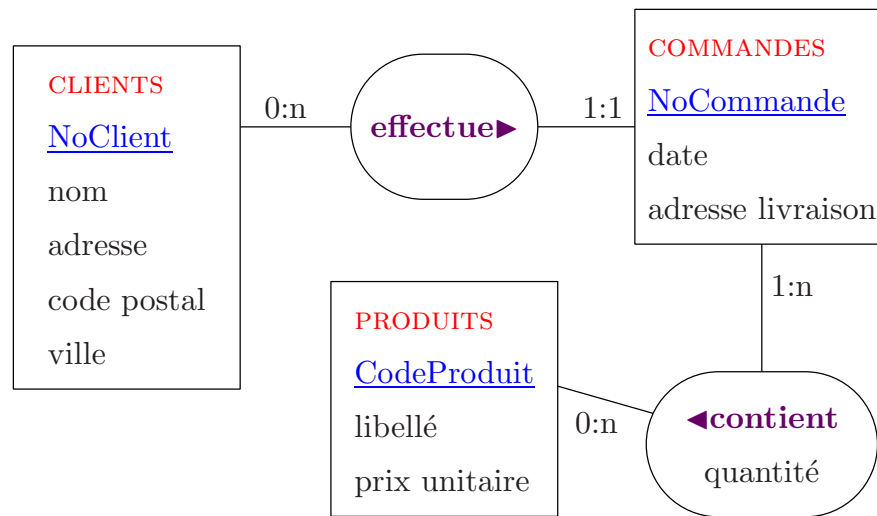
mais la règle *une propriété dépend de tout l'identifiant* n'est pas respectée.

Le NoCommande détermine seul la date.

exercice - cardinalités ?



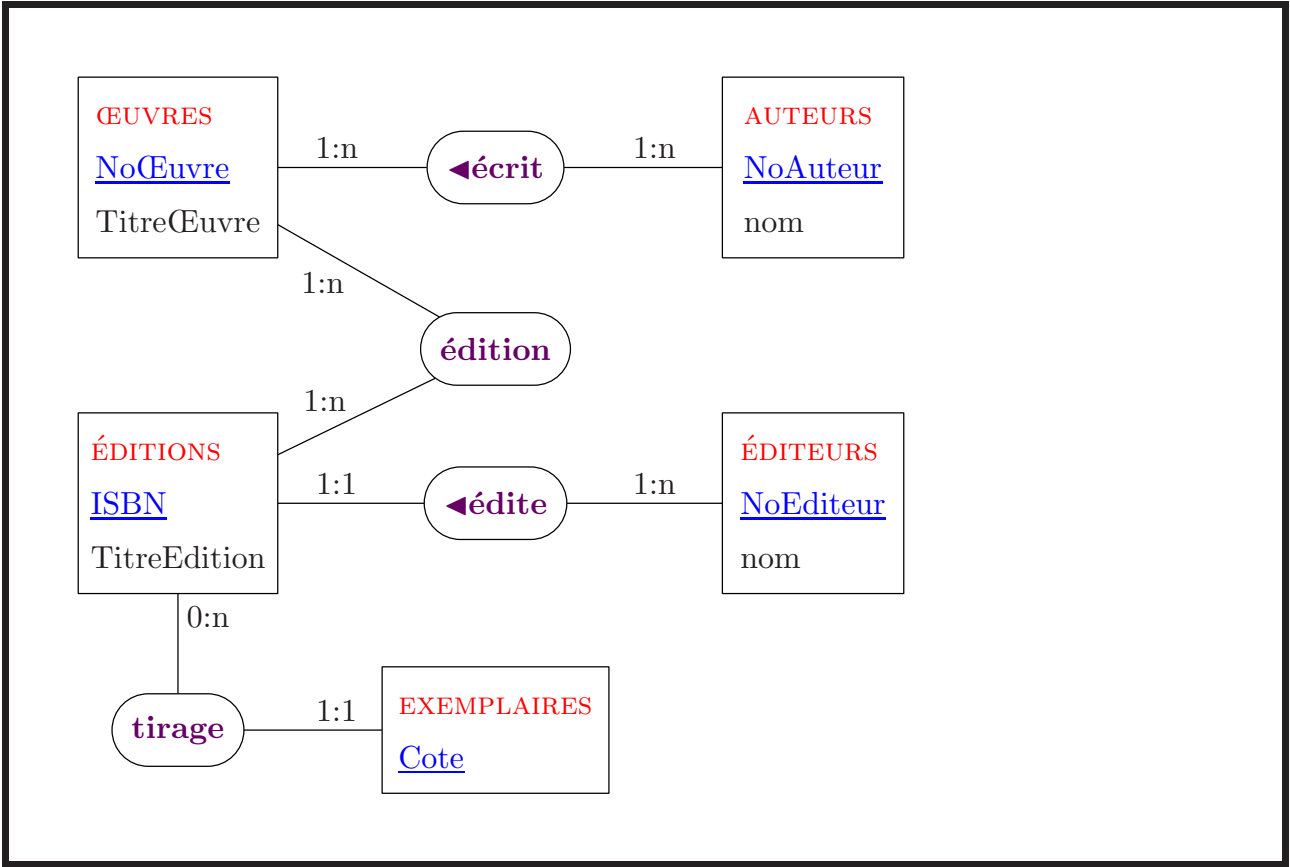
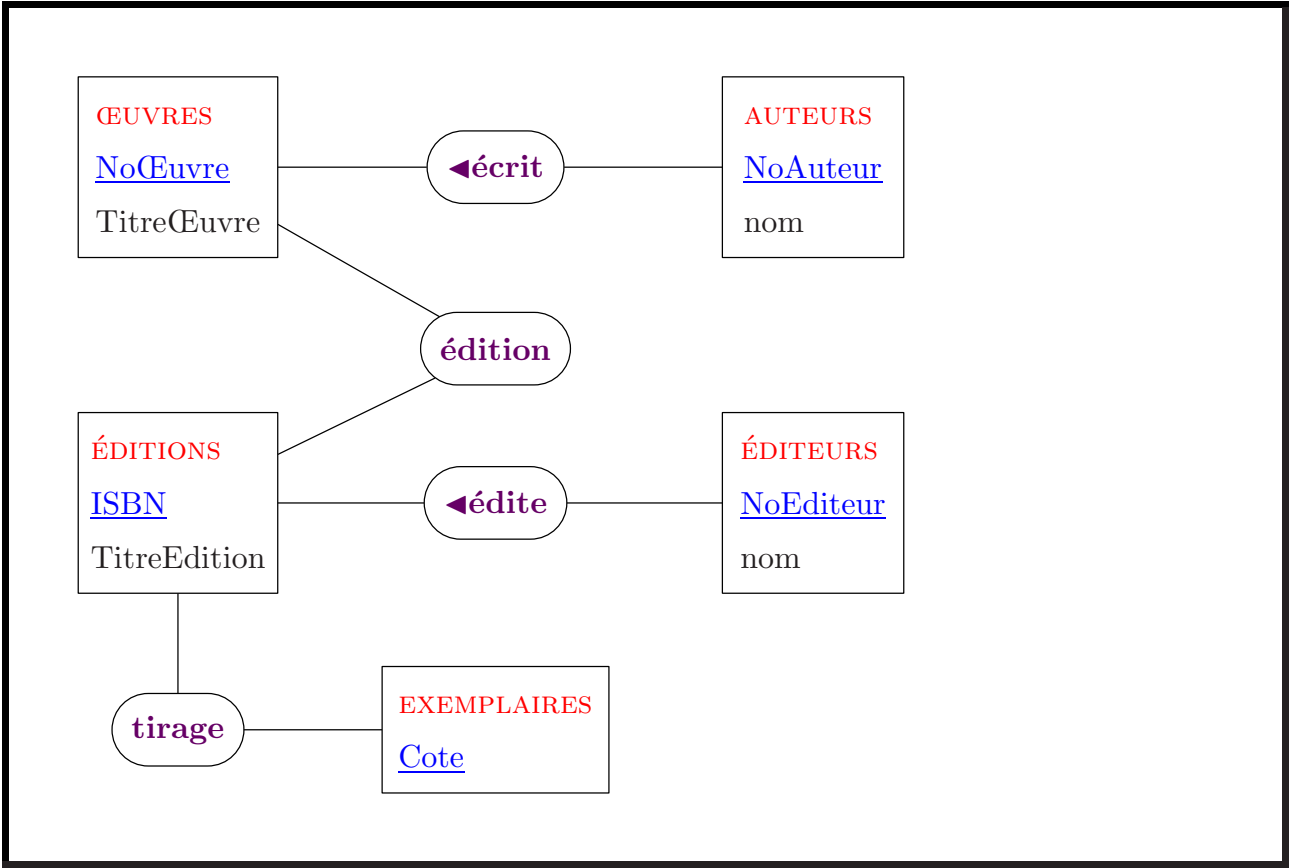
solution



exemple

Règles de gestion pour une bibliothèque

- On appellera œuvre une création littéraire.
- Une œuvre a un auteur au minimum. Elle peut faire l'objet de plusieurs éditions (première édition, réédition, livre de poche...).
- Une édition possède un ISBN unique. Elle peut contenir plusieurs œuvres (recueil, anthologie...). Elle est faite par un seul éditeur.
- Un livre peut exister en plusieurs exemplaires dans la bibliothèque. On ne mémorise pas d'informations sur des œuvres qu'on ne possède pas en bibliothèque ; mais, si un livre disparaît, on ne supprime pas ce qu'on avait déjà enregistré.



exercice

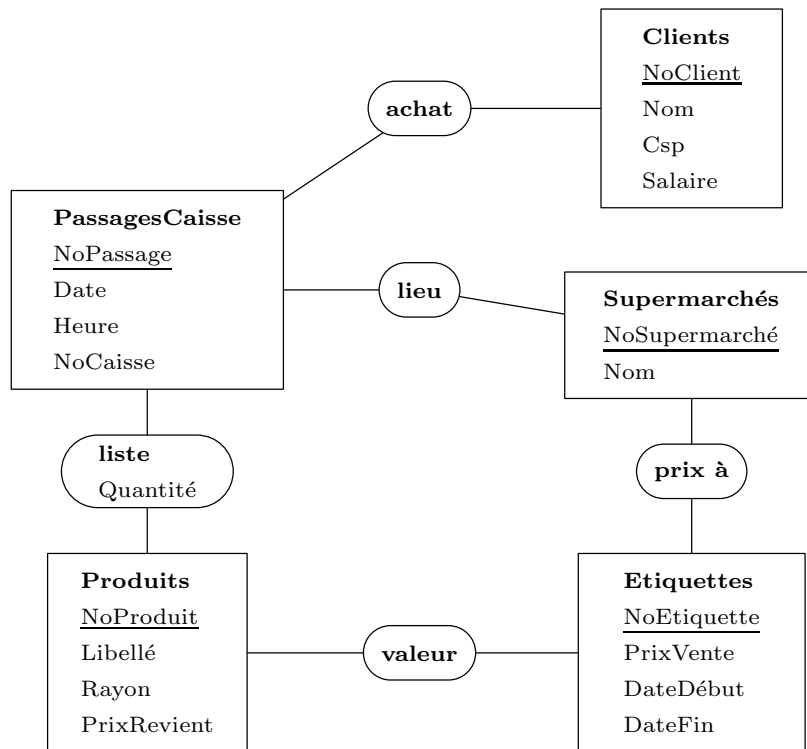
Gestion de réservations pour des spectacles de théâtre.

- Dans la région, il existe de nombreuses salles de spectacle qui ont un nom et se situe dans une ville.
- Pour gérer les réservations, on limitera les informations concernant un spectacle à son titre et au nom de compagnie.
- Un spectacle peut effectuer une tournée dans plusieurs salles. Mais à une date donnée, il ne peut avoir lieu que dans une seule salle. Pour nommer les entités, on appellera *soirée* un spectacle donné à une date donnée.
- Les spectateurs possèdent une identité et une adresse. On voudra savoir si les spectateurs vont principalement dans des salles de leur ville.
- Ils peuvent réserver plusieurs places pour une soirée donnée.

exercice

Règles de gestion pour une chaîne de supermarchés (exam 03). On propose des cartes de fidélité et on mémorise toutes les ventes.

- Dans l'entité **Produits**, le prix de revient est identique pour tous les magasins de la chaîne ; le prix de vente dépend du magasin.
- Dans l'entité **Etiquettes**, on mémorise le prix de vente d'un produit dans un magasin, pour une période donnée ; le prix actuel se repère par une date de fin particulière : 31/12/2100.
- L'entité **PassagesCaisse** représente le passage d'un client à une caisse d'un supermarché.
- Certains achats ne seront pas associés à un client physique. On a prévu un client fictif (Monsieur **Nobody**) dont la fiche existe dans la table des clients et qui remplacera les clients dont on ne connaît pas l'identité. Toutes les ventes des magasins sont donc mémorisées.

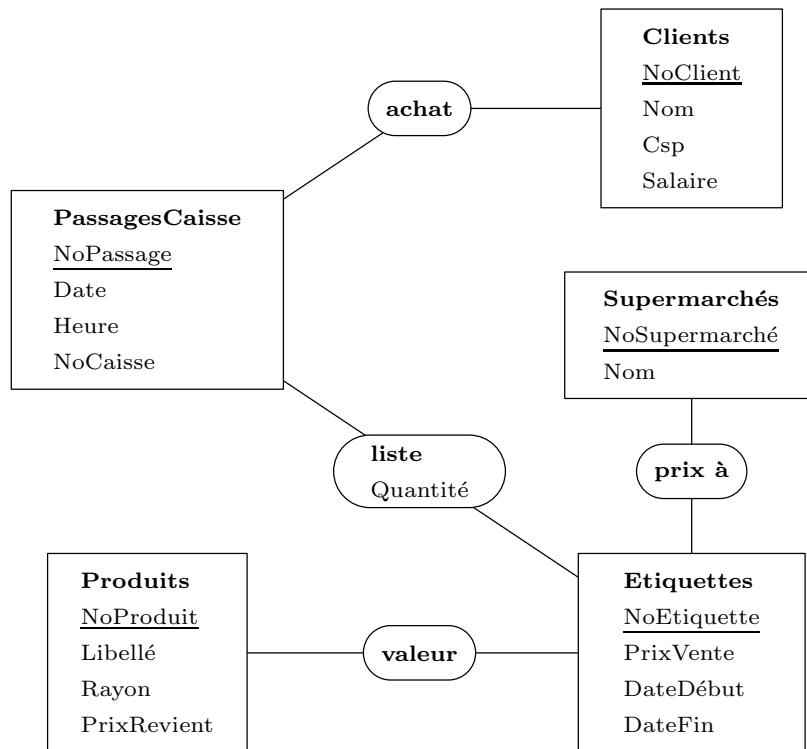


exercice

Moins naturelle, la solution suivante serait peut-être meilleure.

Au lieu d'associer, un ticket de caisse (**PassagesCaisse**) à une liste de produits, le ticket est directement associé aux étiquettes des produits achetés.

Une étiquette étant associée à un produit et à un magasin, on retrouve aisément la liste des produits achetés et le lieu.



exercice

Règles de gestion supplémentaires.

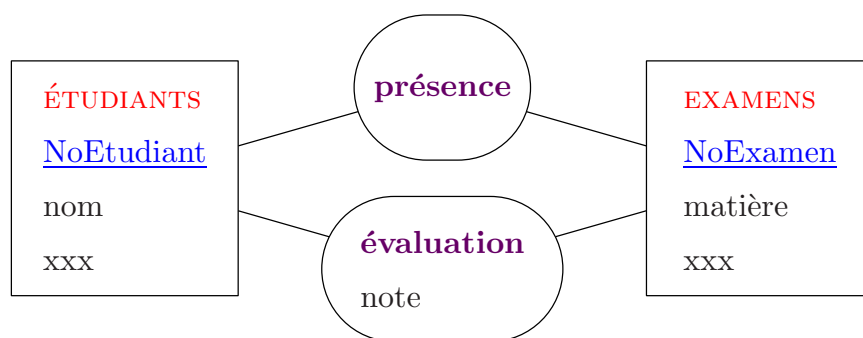
- Les clients proviennent de toutes les villes avoisinantes ; le code postal pourra être pris comme identifiant. Le supermarché est situé dans une localité et on supposera qu'il est impossible qu'aucun client ne vienne de cette localité.
- Des hotesses d'accueil sont employées aux caisses. Il y a forcément une hotesse à la caisse lors du passage du client. Une hotesse peut changer de caisse selon les besoins, selon l'affluence... Lorsqu'elle prend la responsabilité d'une caisse, elle entre son numéro d'identification et tous les passages à cette caisse lui sont attribués jusqu'à ce qu'elle quitte la caisse ; on ne crée pas d'entité pour les caisses. Une hotesse travaille toujours dans le même supermarché.

quelques exemples particuliers

- Une association peut être ternaire. Exemple : VENDEURS, LIEUX, ACHETEURS reliées par *vente*. Posez-vous alors la question “*et si je remplaçais l’association par une entité, ne serait-ce pas plus clair ?*”... une entité ACTESDEVENTE.
- Une association peut être réflexive. Exemple : l’entité INDIVIDUS reliée avec elle-même par l’association *filiation*.
- Deux entités peuvent être reliées par plusieurs associations. Exemple : TRAJETS SNCF reliée à GARES par les associations *GareDépart*, *GareArrivée*, *Etapas*
- Les cardinalités, bien qu’étant très souvent 0:1, 1:1, 0:n ou 1:n, peuvent avoir d’autres valeurs. Exemple du tiercé :

PARIEURS $\overset{1:n}{\text{---}} \bigcirc \overset{1:1}{\text{---}}$ TICKETS $\overset{3:3}{\text{---}} \bigcirc \overset{0:n}{\text{---}}$ CHEVAUXPARTANTS

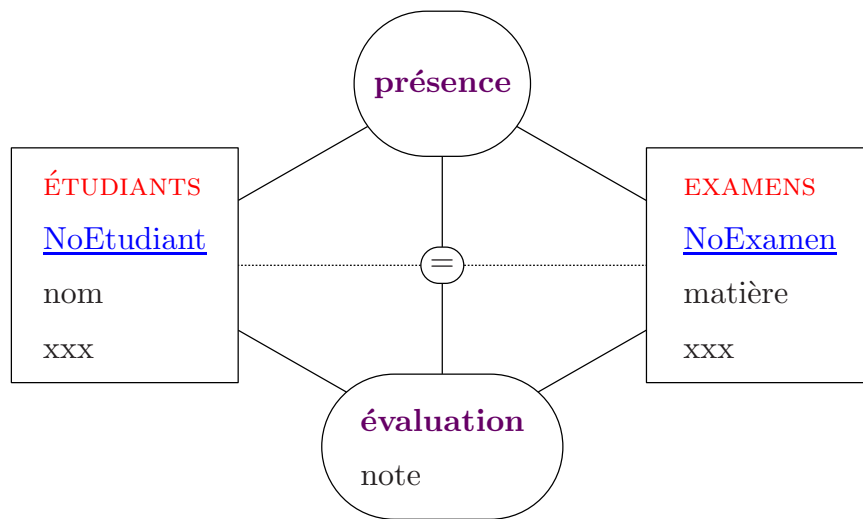
Extensions du modèle



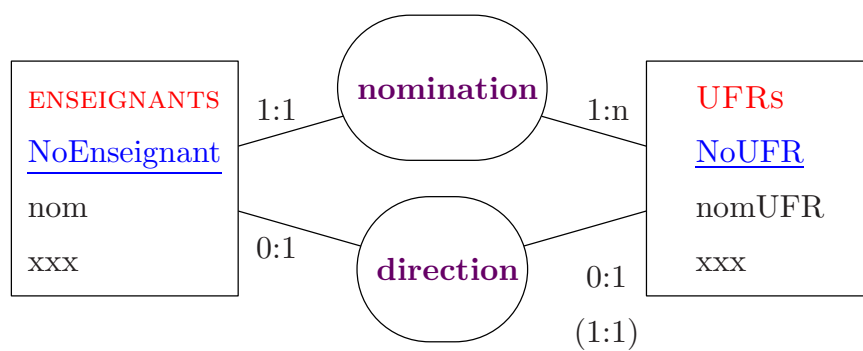
L’association *présence* représente la liste des présents au moment de l’examen et l’association *évaluation* représente le relevé des notes fourni par l’enseignant, après correction.

Pourrait-on indiquer qu’un étudiant présent doit avoir une note ?

Extensions du modèle

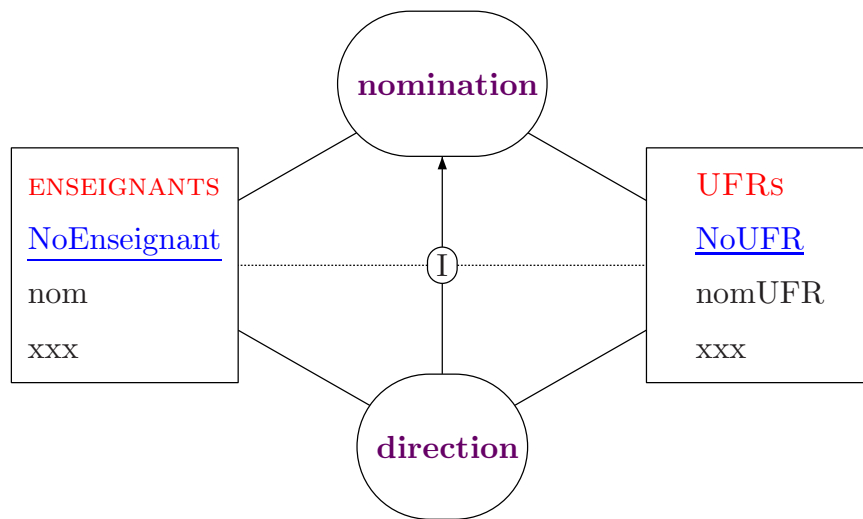


Extensions du modèle

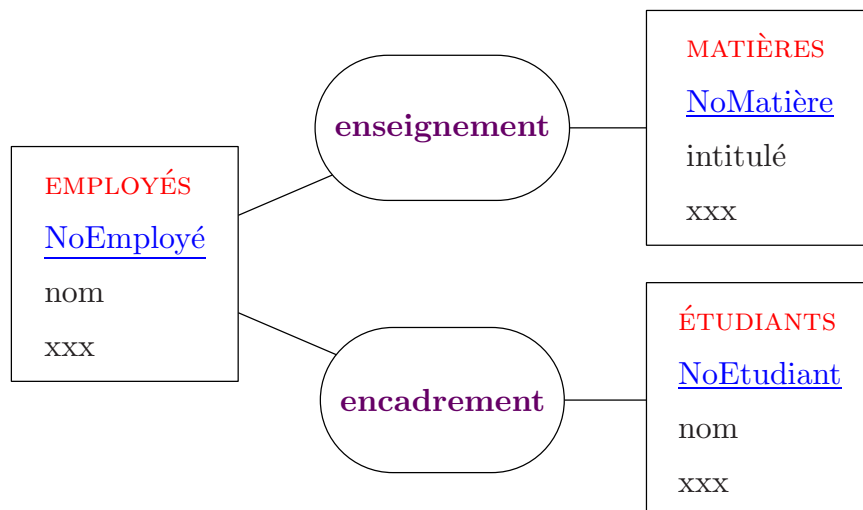


Pourrait-on indiquer que le directeur doit être un enseignant nommé dans l'UFR ?

Extensions du modèle

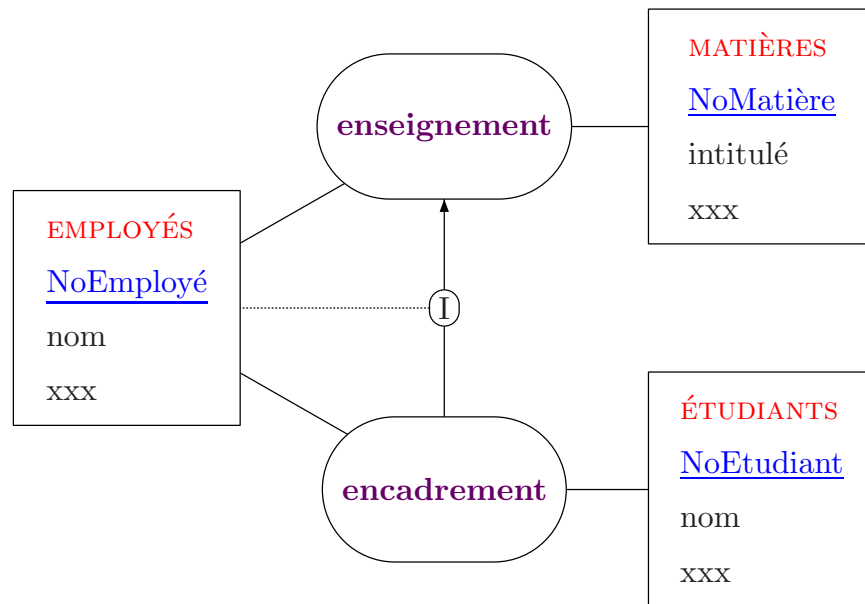


Extensions du modèle

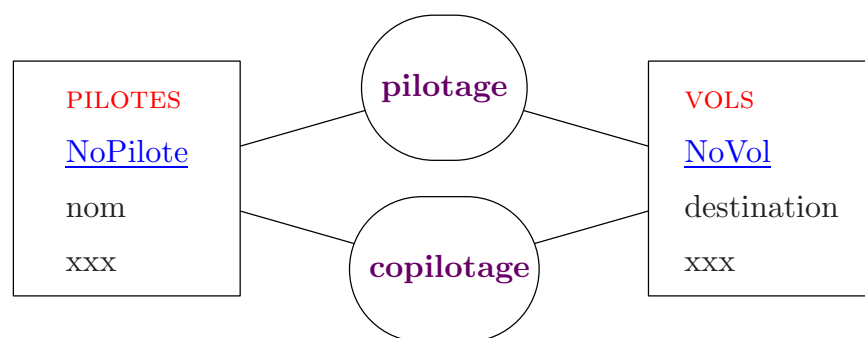


Pourrait-on indiquer qu'un encadrant doit être un enseignant ?

Extensions du modèle

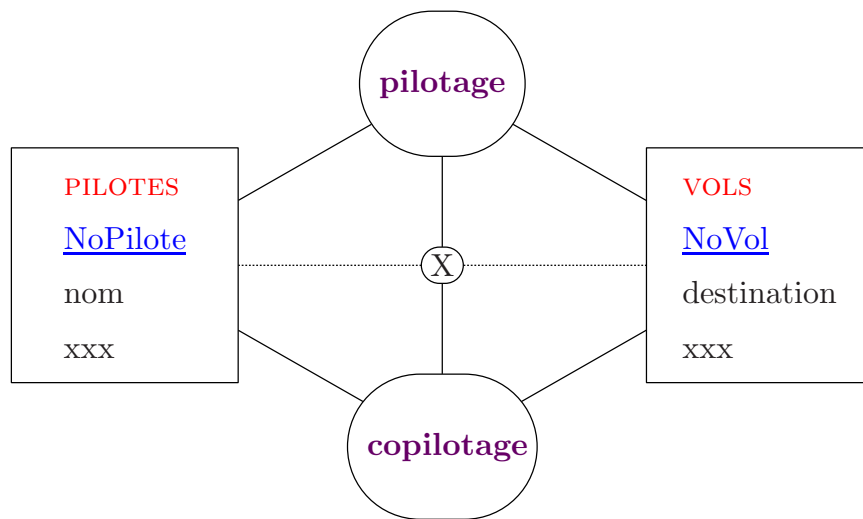


Extensions du modèle

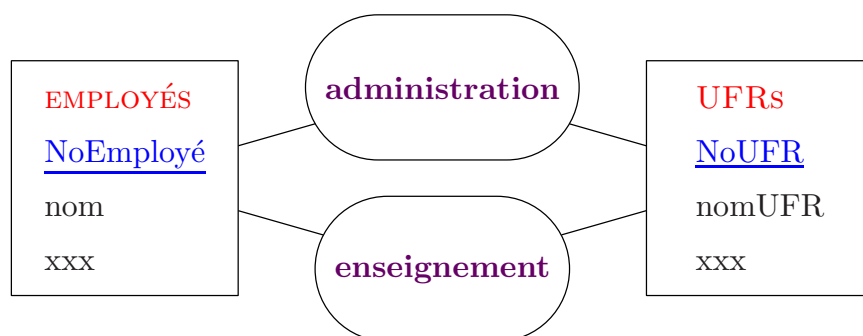


Pourrait-on indiquer que le pilote d'un vol est obligatoirement différent du copilote ?

Extensions du modèle

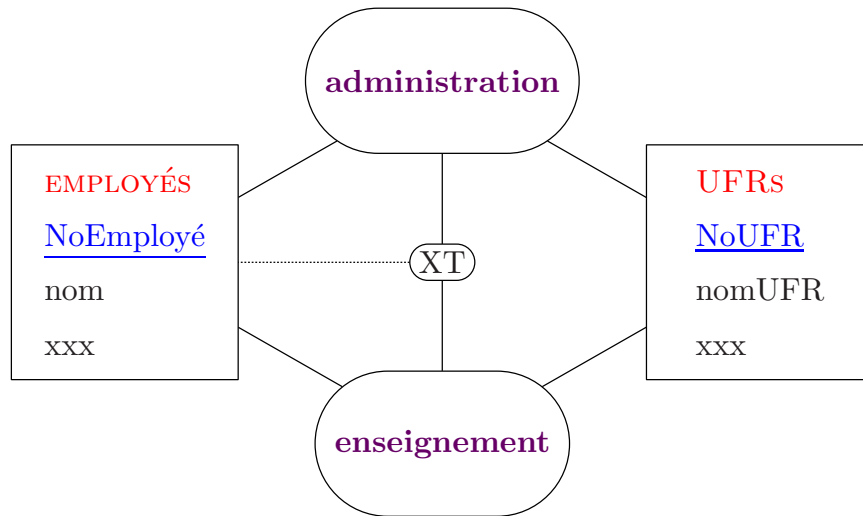


Extensions du modèle



Pourrait-on indiquer qu'un employé de l'université est soit un enseignant, soit un administratif (et forcément l'un ou l'autre) ?

Extensions du modèle



Extensions du modèle

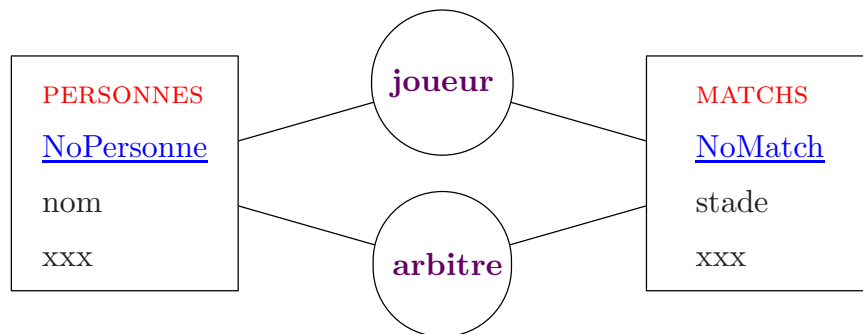
On peut exprimer

- l'égalité (=), $E_1 = E_2$
- l'inclusion (I), *orienté* $E_1 \subseteq E_2$
- l'exclusion (X), $E_1 \cap E_2 = \emptyset$
- la totalité (T), $E_1 \cup E_2 = E$
- l'exclusion et totalité. $E_1 \cap E_2 = \emptyset$ **et** $E_1 \cup E_2 = E$

sur les associations ou sur les arcs de l'association

Il est aussi possible de composer des associations pour indiquer une contrainte sur la composition.

Exercice



Comment indiquer que l'arbitre d'un match n'est pas un joueur de ce match, et inversement ?

Comment indiquer qu'aucun arbitre n'est un joueur, et inversement ?

Comment indiquer que tout arbitre a joué au moins un match ?

Comment indiquer que toute personne est soit arbitre, soit joueur ?

Contraintes d'intégrité fonctionnelle

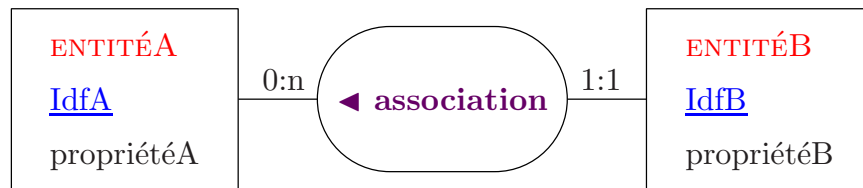
Il y a contrainte d'intégrité fonctionnelle quand la connaissance d'une occurrence dans une (ou plusieurs) entité(s) détermine la connaissance de l'occurrence associée dans l'autre entité.

Cela ressemble à des cardinalités 1:1 sur une association qui serait intitulée *détermine...*

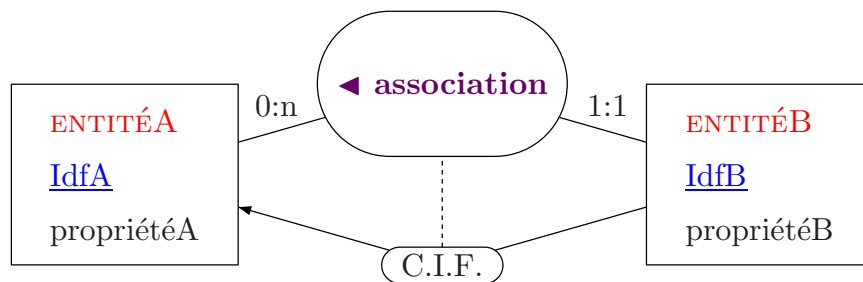
Inversement, si une association possède des cardinalités 1:1, il y a C.I.F.

Indiquer une C.I.F. apporte peu d'information dans le cas des associations binaires.

Contraintes d'intégrité fonctionnelle



pourrait être notée



Contraintes d'intégrité fonctionnelle

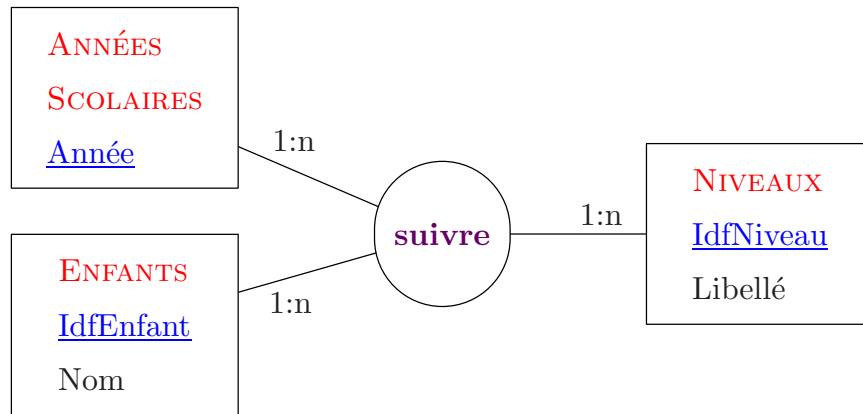
Dans le cas des associations ternaires, mettre en évidence une C.I.F. permet de faire apparaître des propriétés qui n'étaient pas indiquées.

Elle permet d'indiquer que la connaissance d'une occurrence dans deux entités détermine la connaissance d'une occurrence dans la troisième entité.

Vous remarquerez, dans l'exemple suivant, que la C.I.F. ne correspond pas forcément à des cardinalités 1:1.

Contraintes d'intégrité fonctionnelle

Exemple : un élève suit un niveau scolaire chaque année.



Contraintes d'intégrité fonctionnelle

La C.I.F. permet d'indiquer qu'une année et un élève déterminent un niveau.

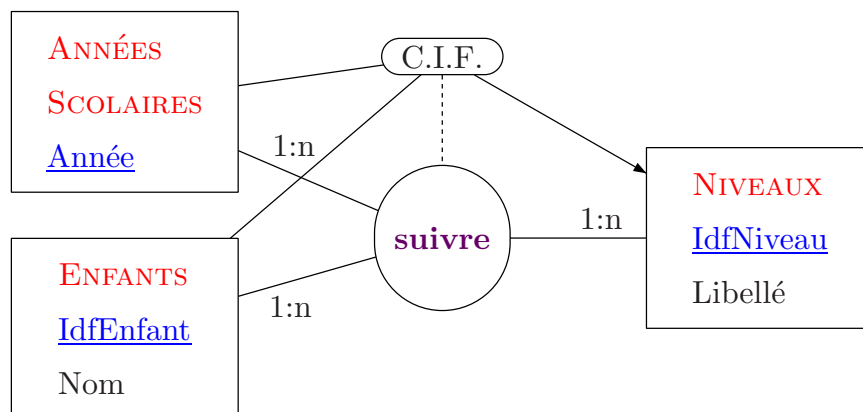


Schéma conceptuel - Schéma logique

Le schéma conceptuel répondait à la question : quelles informations ?

Le schéma logique répond à la question : dans quel domaine, dans quel environnement ?

Nous n'étudierons que le modèle relationnel (le plus répandu dans les bases de données). Mais il existe des environnements arborescents, réseaux...

Schéma logique relationnel

Le schéma conceptuel doit se transformer en un *schéma logique relationnel*.

Le schéma logique relationnel est une description de la structure des tables qui constituent la base de données.

Exemple : ENSEIGNANTS (IdfEnseignant, Nom, Prénom, Adresse, #RefUFR)

Nous allons étudier le passage du schéma conceptuel au schéma logique relationnel en utilisant la notation graphique utilisée dans Merise (le **MLD**).

Schéma logique relationnel

- Le **domaine** est l'ensemble des valeurs que peut prendre une donnée.
- Une **table** porte un nom et est composée d'**attributs** prenant leurs valeurs dans les domaines correspondants.
Une table est un sous-ensemble du produit des domaines ; une table est donc un ensemble d'enregistrements (ou tuples, ou lignes).
- Une **clé** est constituée de 1 ou plusieurs attributs telle que une valeur de la clé détermine exactement l'enregistrement.
- Toute table possède une **clé primaire** et, éventuellement, des **clés candidates**.

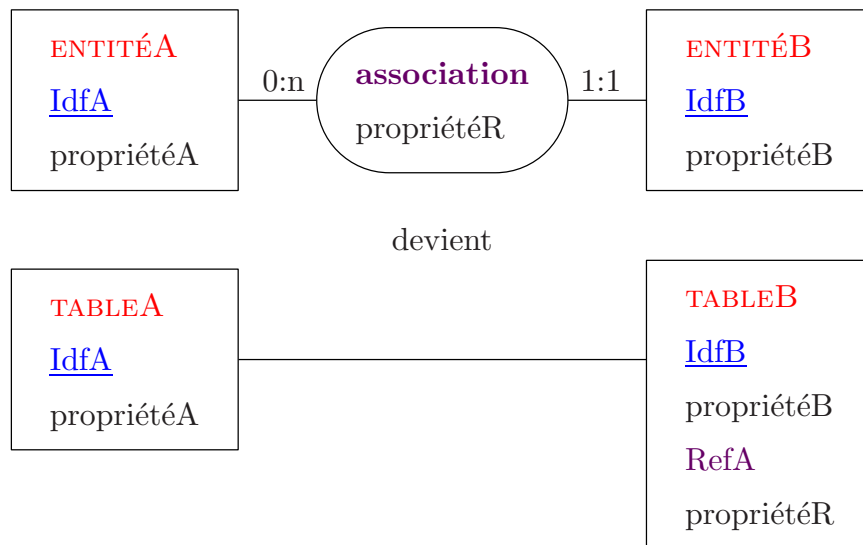
le passage du conceptuel au logique relationnel

Pour les entités. Toute entité devient une table, les propriétés de l'entité sont les attributs de la table, l'identifiant de l'entité est la clé primaire de la table.

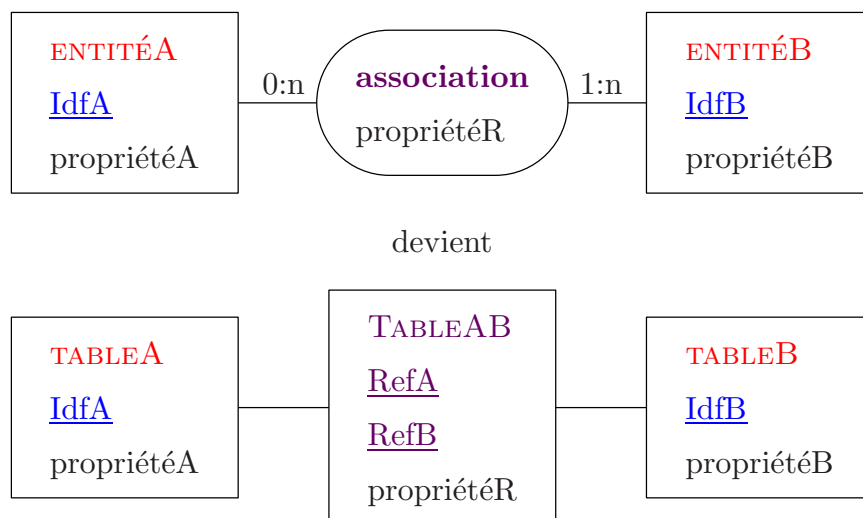
Pour les associations. Cela dépend des cardinalités. Deux cas sont possibles :

- association $\square - \overset{1:1}{-} - \bigcirc - \overset{1:n}{-} - \square$: la relation est matérialisée par l'ajout d'une clé étrangère.
- association $\square - \overset{1:n}{-} - \bigcirc - \overset{1:n}{-} - \square$: la relation donne lieu à la création d'une table.
- les cardinalités 0:1 (resp. 0:n) se traitent comme les cardinalités 1:1 (resp. 1:n).

le passage du conceptuel au logique relationnel

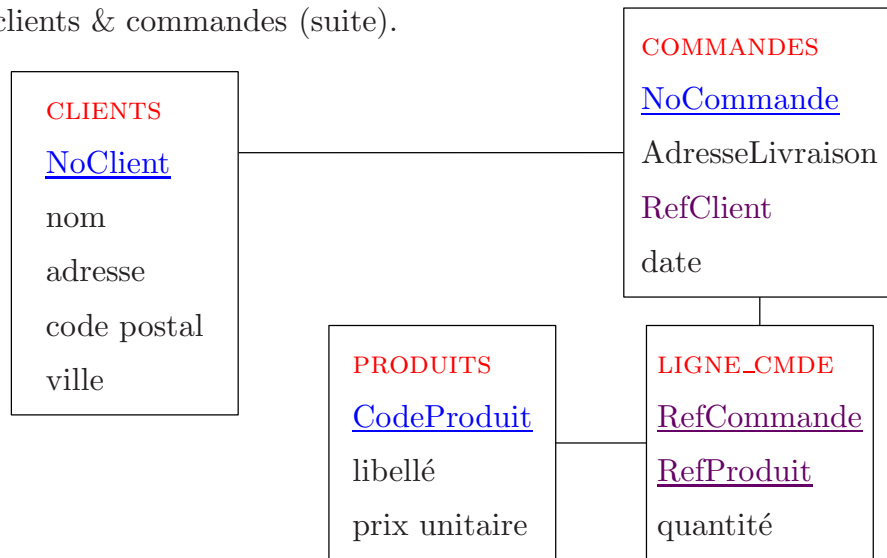


le passage du conceptuel au logique relationnel

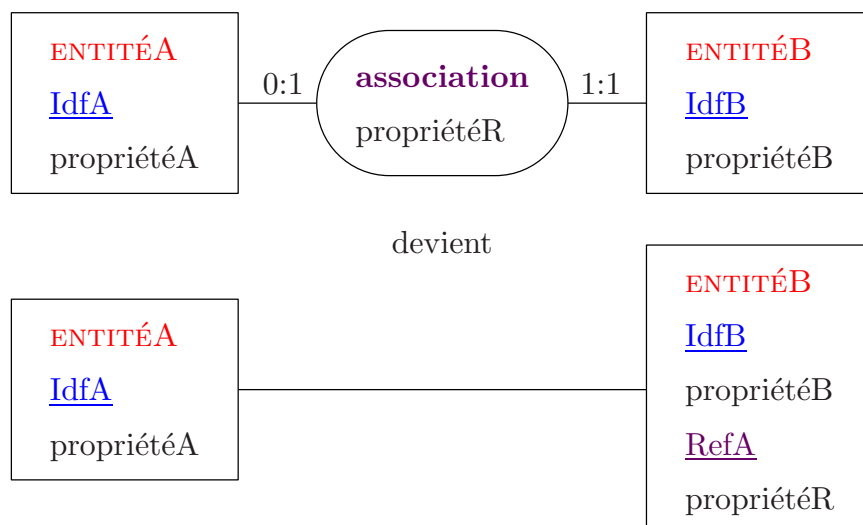


Exemple

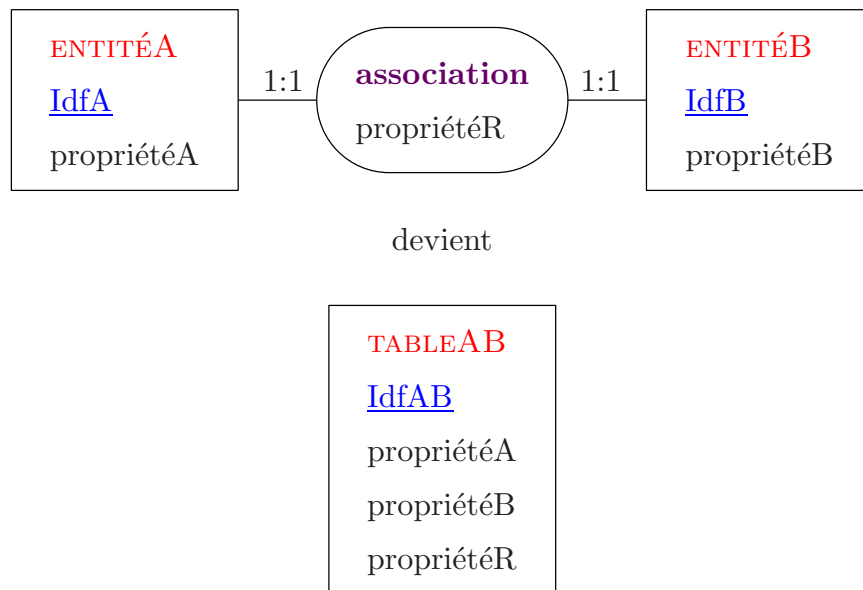
Exemple : clients & commandes (suite).



Cas particuliers

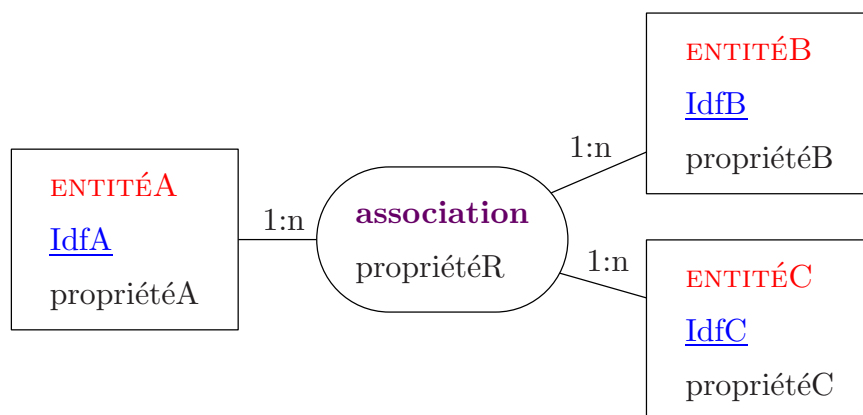


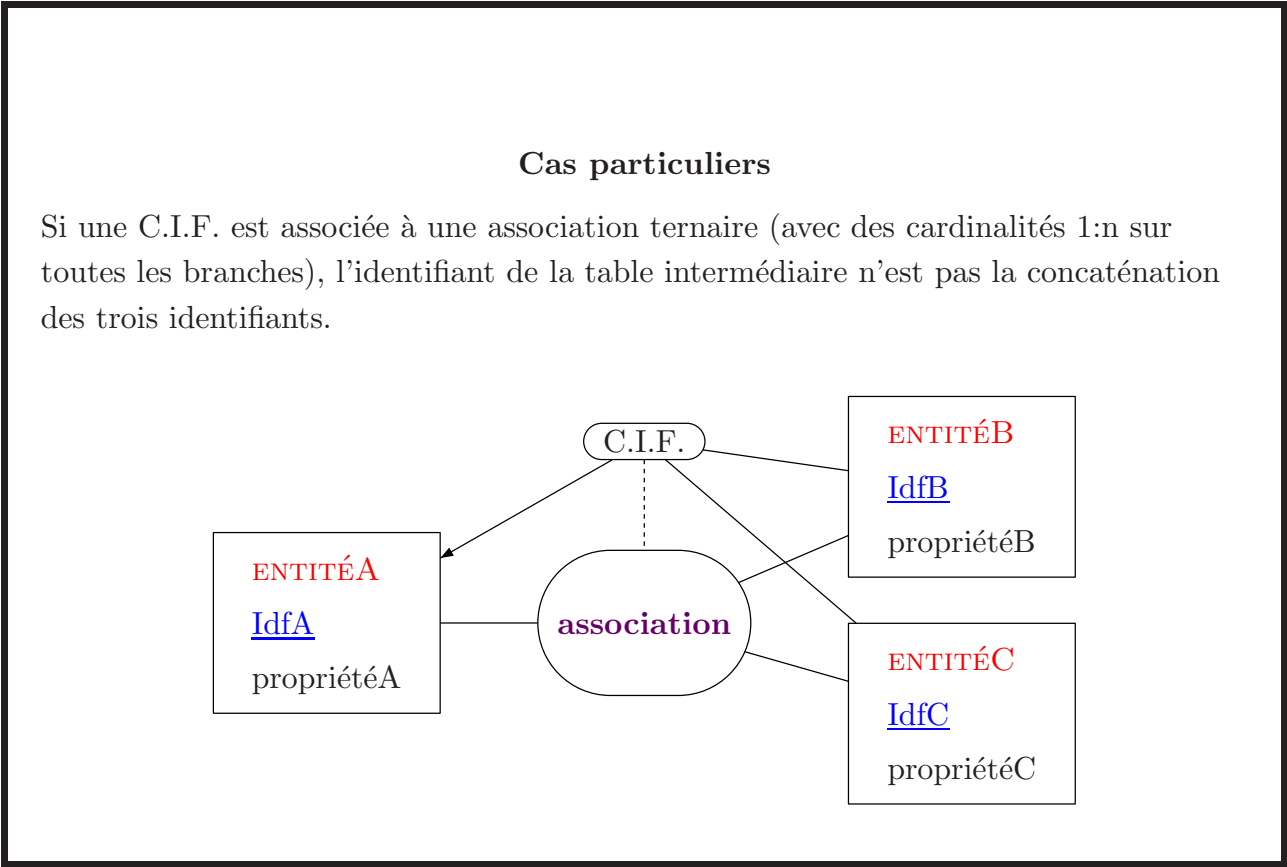
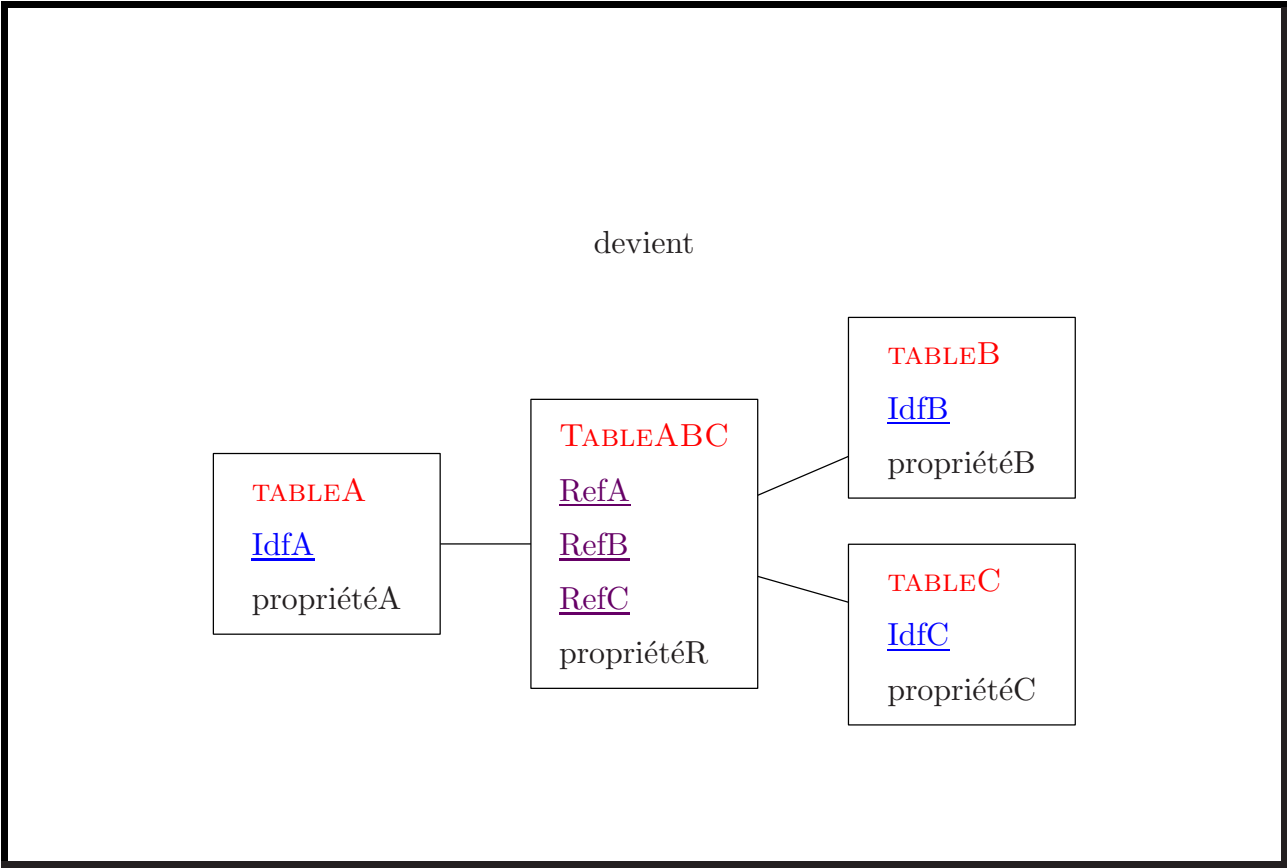
Cas particuliers

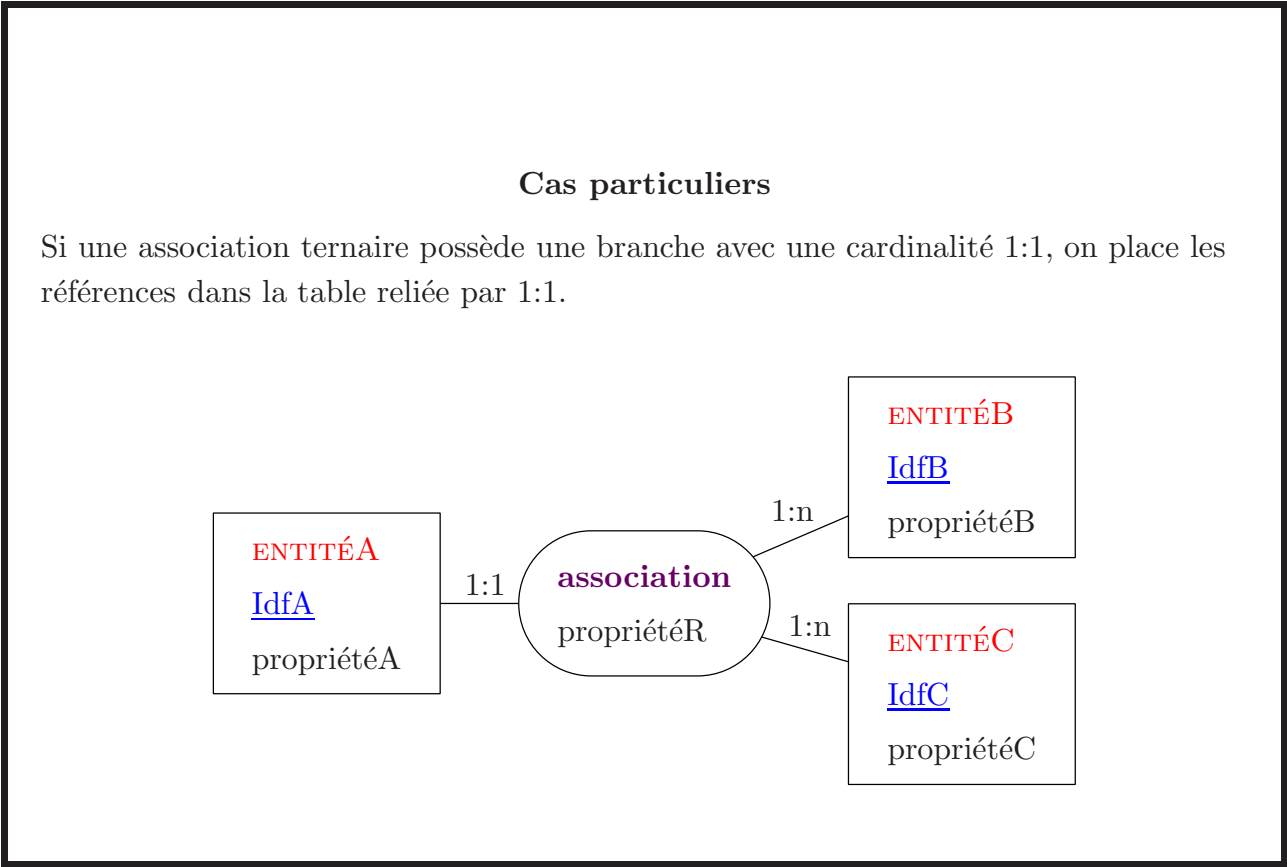
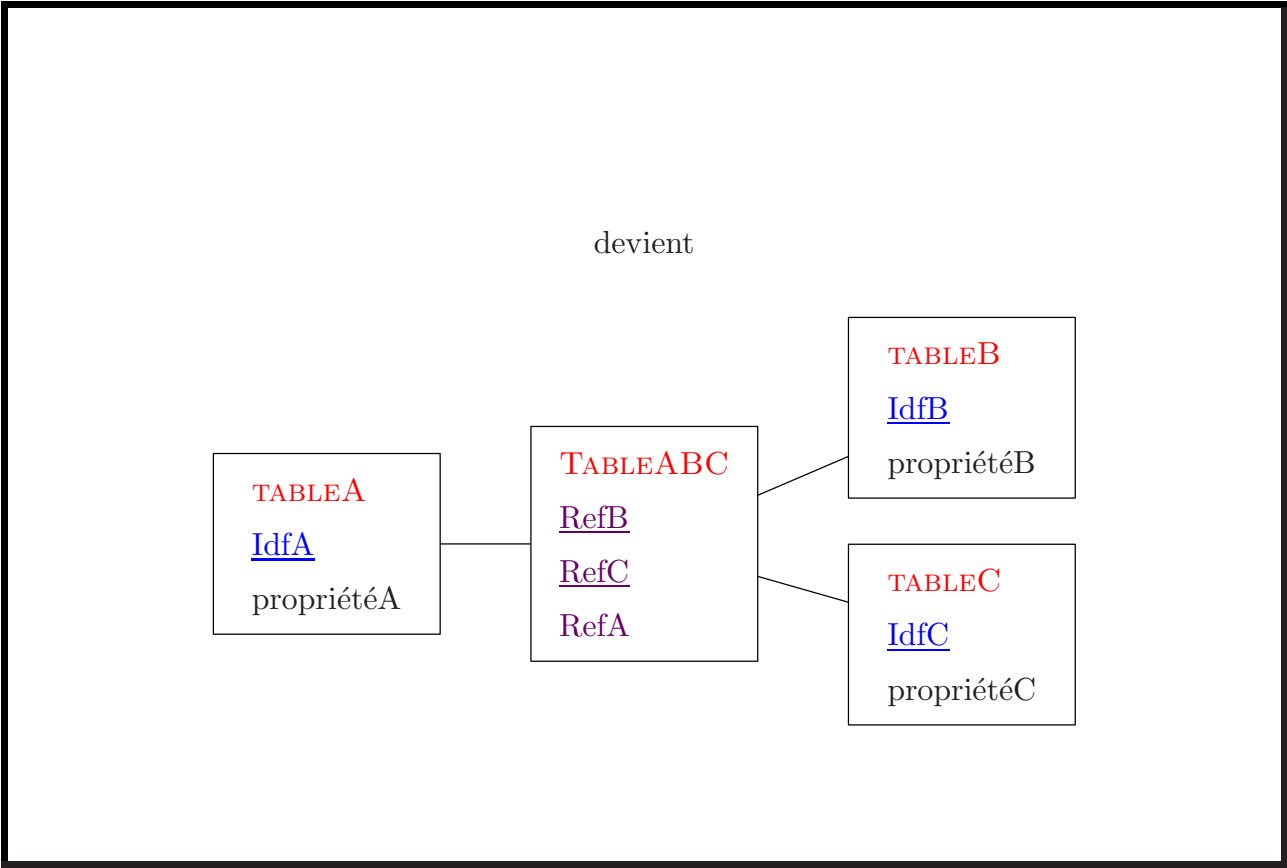


Cas particuliers

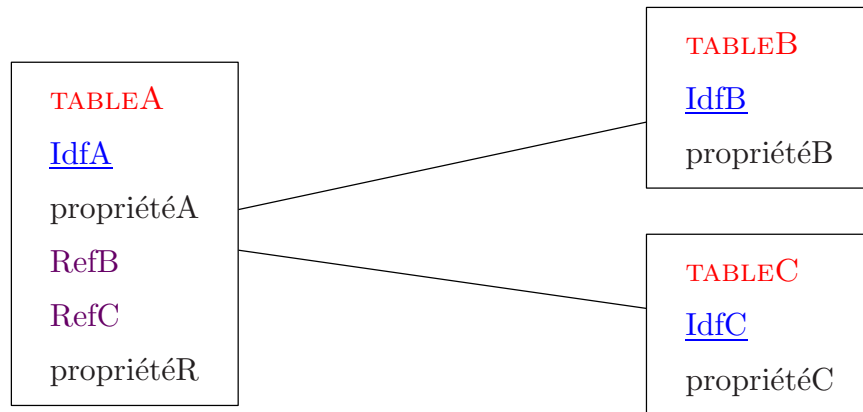
Une association ternaire devient une table si les cardinalités sont 1:n sur toutes les branches.







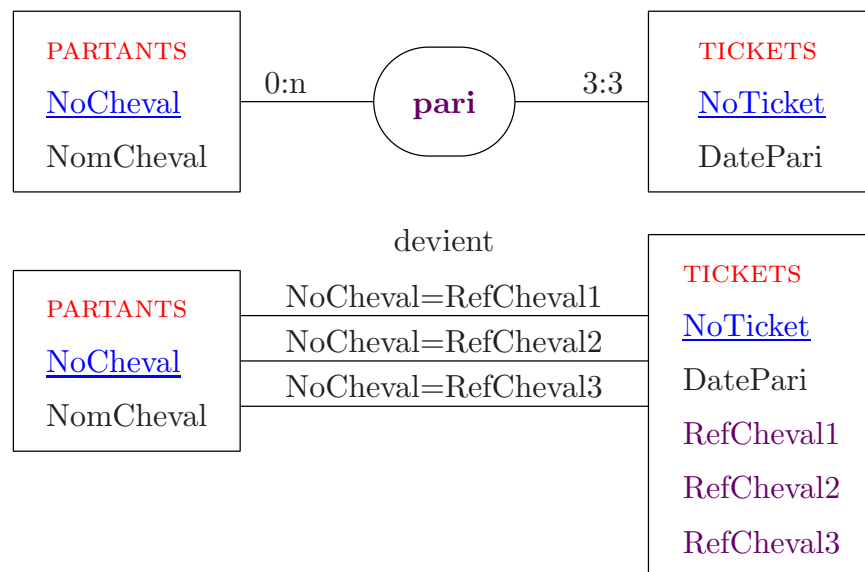
devient



Cas particuliers

Si plusieurs associations relient deux entités, on traite chacune des associations indépendamment des autres, ce qui peut donner lieu à l'ajout de plusieurs références. Exemple : **TRAJETS SNCF** (NoTrajet, HeureDépart, HeureArrivée, RefGareDepart, RefGareArrivée).

Les cardinalités k:k donnent aussi lieu à l'ajout de plusieurs références. Exemple du tiercé : **TICKETS** (NoTicket, DatePari, RefCheval1, RefCheval2, RefCheval3).



Optimisation

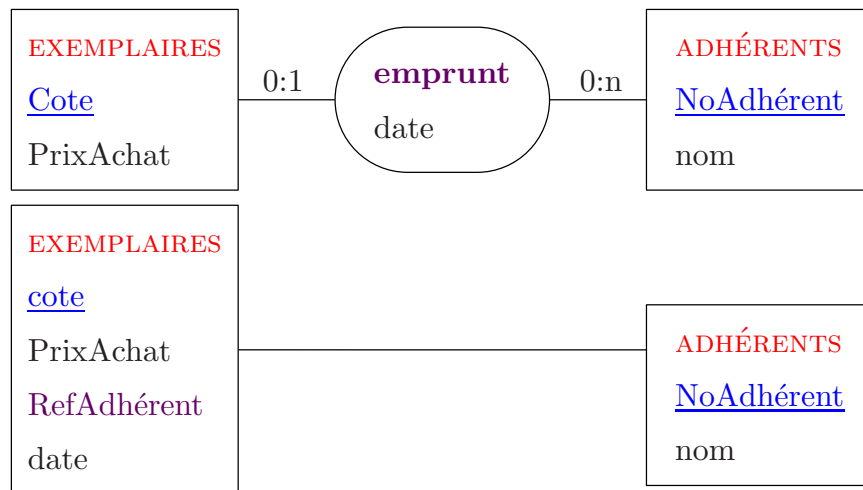
Le système de gestion des données doit répondre au souci d'une gestion sûre des données. Ceci est assuré par la construction des schémas conceptuel et logique. Mais cette gestion se doit aussi d'être efficace. Par conséquent, il faut **optimiser** le schéma logique, c'est-à-dire, par exemple :

- les jointures étant coûteuses, supprimer les tables inutiles ;
- parfois, il peut être utile d'ajouter des données calculées pour accélérer certains traitements ;
- il peut être intéressant de gagner de l'espace mémoire, en ne respectant pas les règles de transformation.

Ces choix doivent être justifiés.

Optimisation

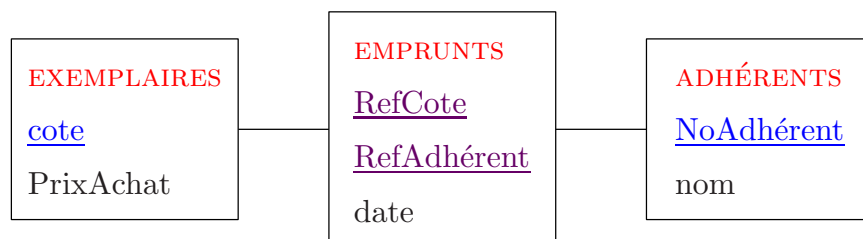
Les cardinalités 0:1 se traitent comme les cardinalités 1:1.



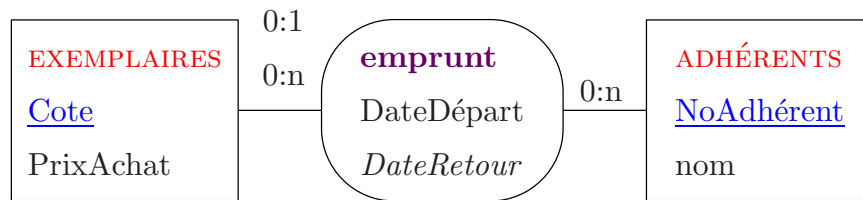
La valeur *Null* est autorisée.

Optimisation

Mais si peu de livres sont empruntés, de nombreuses fiches auront une référence *Null*.
Ne serait-il pas mieux de faire ?



Optimisation

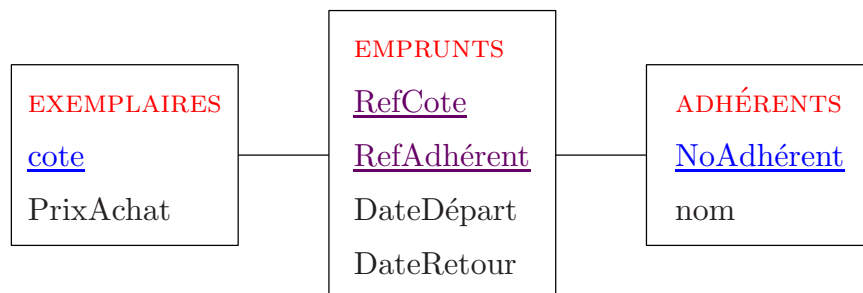


Les cardinalités 0:1 signifient qu'on gère seulement l'emprunt actuel. On ne mémoriserait pas la date de retour puisqu'au retour d'un exemplaire, on supprime la fiche d'emprunt.

Les cardinalités 0:n signifient qu'on gère l'historique des emprunts.

Optimisation

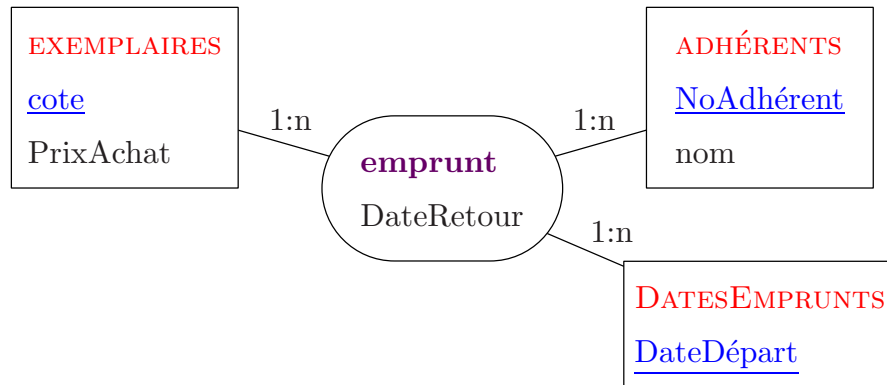
Dans le cas de la gestion de l'historique (cardinalités 0:n), on doit créer une table intermédiaire.



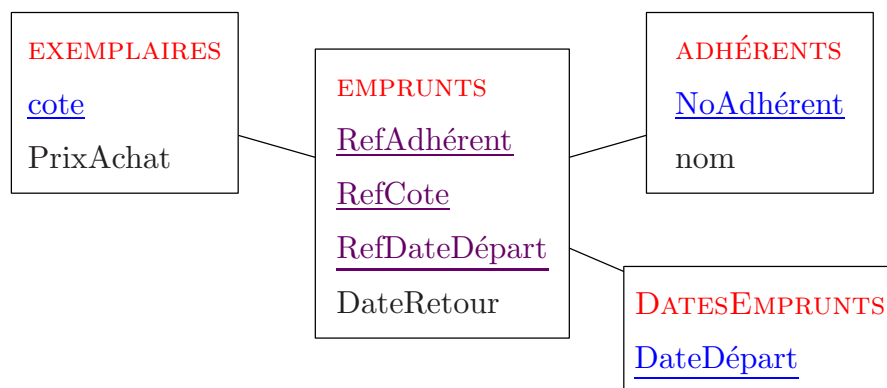
L'identifiant d'une association est la concaténation des identifiants des deux entités. Donc la clé primaire de la table EMPRUNTS serait le couple (RefCote,RefAdhérent). Est-ce le cas ?

Optimisation

Il est assez fréquent qu'une association ayant une date (une heure, une année...) comme propriété, cache une relation ternaire optimisée.



qui devient



La table DATESÉMPRUNTS est supprimée, par optimisation, car elle ne contient qu'une propriété dont les seules valeurs intéressantes sont dans la table EMPRUNTS.

Retour au schéma logique relationnel

Le schéma logique relationnel n'est qu'une écriture condensée des graphiques que nous avons utilisés. Il respecte la syntaxe :

NomTable (liste des attributs)

Des conventions d'écriture sont souvent adoptées :

- nom de l'identifiant principal en premier et souligné
- nom des clés étrangères (les références) précédées du symbole #

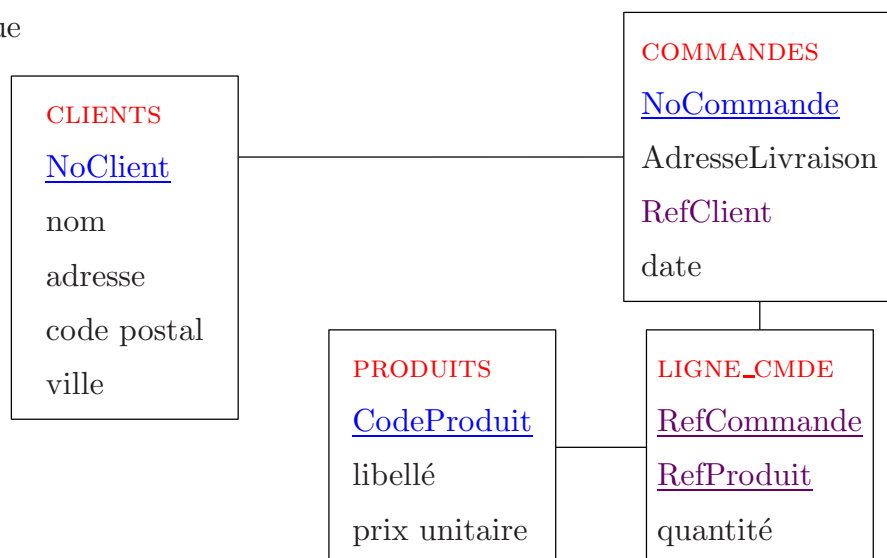
Exemple : ENSEIGNANTS (IdfEnseignant, Nom, Prénom, Adresse, #RefUFR)

IdfEnseignant est l'identifiant principal.

RefUFR est une clé étrangère qui fait référence à un numéro d'UFR dans une autre table.

Exemple

Le graphique



serait écrit

CLIENTS (NoClient, nom, adresse, code postal, ville)

COMMANDES (NoCommande, AdresseLivraison, #RefClient, date)

LIGNE_CMDE (#RefCommande, #RefProduit, quantité)

PRODUITS (CodeProduit, libellé, prix unitaire)

Contraintes d'intégrité

de structure : valeur de la clé primaire unique et toujours définie,

de domaine : les valeurs prises par un attribut doivent vérifier des contraintes,

de référence : les valeurs d'une clé étrangère doivent correspondre à des valeurs existantes dans la table d'origine.

La vérification des contraintes assurent que la base reste **intègre**. Cette vérification s'effectue, soit directement par le système de gestion de bases de données utilisé (quand c'est possible), soit par l'écriture de programmes.