



Systèmes de Base de Données

Université de Lille2 – IUT C – STID2
2013/2014

Dr Khalid GABER



Introduction

- L'analyse des données constitue le point de passage obligé de toute conception d'application mettant en œuvre un SGBDR (système de gestion de base de données relationnelle). La méthode MERISE, basée sur le modèle entité-association, est un outil simple et efficace, très répandue chez les développeurs européens.



Systeme d'Informations (SI)

- le système d'information d'une organisation (société, entreprise, club, université, etc.) regroupe l'ensemble des moyens (humain, matériel, logiciel, ...) permettant d'acquérir, de mémoriser, de traiter et de communiquer des informations relatives à l'organisation concernée.

- La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. La **modélisation** consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse. Ce type de méthode est appelé **analyse**.
- Il existe plusieurs méthodes d'analyse, la méthode la plus utilisée en France étant la méthode MERISE, UML



SI - suite

Dans un SI, il existe 2 types d'éléments :

- Les données (adresse d'un client, prix d'un article) décrivent le Système dans son aspect statique.
- Les traitements décrivent le fonctionnement du système (consultation des factures impayées, moyenne d'un étudiant).

Chacun de ces éléments est indissociable de l'autre.



Dictionnaire de données

- Le dictionnaire de données est réalisé à partir d'un ensemble d'informations sur les données du S.I.

Le dictionnaire qualifie chaque information selon :

1) **Sa nature :**

- **Élémentaire** : les données de base (nom, prénom, date de naissance, ville ...)
- **Calculée** : obtenue par traitement à partir des autres données (moyenne, montant TTC, ...)



Dictionnaire de données

2) Ses **règles de calcul** (ou règles de gestion)

Ne concerne que les informations calculées : c'est la description des traitements à effectuer

Exemple: $\text{prix TTC} = \text{prix HT} * (1 + \text{taux TVA})$

3) Son **mode de représentation des données**

Alphabétique, Alphanumérique, Numérique, Temporel, logique,...

4) Ses **contraintes d'intégrité**

Énoncer les conditions de validité des données.

Exemple: numéro employé unique; note entre 0 et 20



Représentation du dictionnaire des données: sous forme d'un tableau.

Code rubrique	Nom rubrique	Type	Nature	Règle de calcul	Règle d'intégrité
N_Comd	Numéro de commande	Numérique	Elémentaire		unique
Date_Comd	Date de la commande	Date	E		jj:mm:aaaa
NumClient	Numéro du client	Alpha-Numérique	E		unique
NomClient	Nom de client	Texte	E		Non null
Rue_Client	Nom de la rue du client	Texte	E		
Ville_Client	Nom de la ville du client	Texte	E		
Code_Prod	Numéro de produit	Alpha-Numérique			unique
Qte	Quantité du produit commandée	Numérique	E		Entier > 0
PU	Prix unitaire du produit	Numérique	E		
Total	Montant total facture	Numérique	Calculé	Σ montant	



Dépendance fonctionnelle

■ Définitions

- Une propriété (ou attribut) est une rubrique, c'est à dire une donnée élémentaire. Elle peut être simple (numéro, nom, ...) ou composée (adresse, date, ...)
- On dit qu'une propriété B est en dépendance fonctionnelle (ou dépend fonctionnellement) d'une propriété A si la connaissance d'une valeur de A détermine au plus une valeur de B.



Dépendance fonctionnelle

On notera

$$A \longrightarrow B$$

- Pour signifier que B est une dépendance fonctionnelle de A.
- On appelle A la source de la dépendance fonctionnelle et B son but.

Exemple:

Numéro_INSEE \longrightarrow Nom

CodeClient \longrightarrow NomClient



Dépendance fonctionnelle

Dépendance fonctionnelle simple

- C'est une D.F. à deux rubriques

Code_produit \longrightarrow Nom_produit

Dépendance fonctionnelle à partie gauche composée

- C'est une D.F. comportant dans sa partie gauche (la source) plusieurs rubriques indépendantes.

(NuméroFacture, Code_produit) \longrightarrow quantité_facturée



Dépendance fonctionnelle élémentaire

- Une D.F. $A \longrightarrow B$ est dite élémentaire si aucune partie de la source A ne permet de déterminer le but B .
- Toute D.F. à deux rubriques est élémentaire.

Code_produit \longrightarrow Nom_produit
est une D.F. élémentaire

(NuméroFacture, CodeProduit) \longrightarrow quantité_facturée
n'est pas une D.F. élémentaire



Dépendance fonctionnelle directe

- On dit que $A \longrightarrow B$ est une D.F. directe s'il n'existe pas de propriété C telle que :
 $A \longrightarrow C$ et $C \longrightarrow B$.

Exemple :

- $N^{\circ} \text{ prof} \longrightarrow \text{CodeMatière}$: *directe*
- $\text{CodeMatière} \longrightarrow \text{nom_matière}$: *directe*
- $N^{\circ} \text{ prof} \longrightarrow \text{nom_matière}$: *non directe*

Les D.F. qui nous intéressent sont les D.F. élémentaires directes.



Dépendance fonctionnelle : Propriétés

- Réflexivité : $A \rightarrow A$
 $numéroProduit \rightarrow numéroProduit$
- Transitivité : $A \rightarrow B$ et $B \rightarrow C \Rightarrow A \rightarrow C$
 $CodeProduit \rightarrow codeTVA$ et $codeTVA \rightarrow tauxTVA$
 $\Rightarrow CodeProduit \rightarrow tauxTVA$
- Additivité : $A \rightarrow B$ et $A \rightarrow C \Rightarrow A \rightarrow B+C$
 $CodeProduit \rightarrow désignation$ et $CodeProduit \rightarrow prix$
 $unitaire$
 $\Rightarrow CodeProduit \rightarrow désignation + prix unitaire$



Recherche des dépendances fonctionnelles

- On recherche les dépendances fonctionnelles à partir du dictionnaire de données

Démarche en trois étapes :

- 1) Recherche des D.F. simples (à 2 rubriques)
 - *code* \rightarrow *désignation*
- 2) Si le but est une rubrique composée, on découpe la D.F.
 - *NuméroClient* \rightarrow *AdresseClient*
 - *NuméroClient* \rightarrow *rue*, *NuméroClient* \rightarrow *ville*, ...



Recherche des dépendances fonctionnelles

3) Éliminer les D.F. **symétriques** pour ne garder que la plus importante

Exemple :

- $NuméroProduit \rightarrow DésignationProduit$
- $DésignationProduit \rightarrow NuméroProduit$

On pourrait conserver que :

- $NuméroProduit \rightarrow DésignationProduit$

Représentation du dictionnaire des données: sous forme d'un tableau.

Code rubrique	Nom rubrique	Type	Nature	Règle de calcul	Règle d'intégrité
N_Comd	Numéro de commande	Numérique	Elémentaire		unique
Date_Comd	Date de la commande	Date	E		jj:mm:aaaa
NumClient	Numéro du client	Alpha-Numérique	E		unique
NomClient	Nom de client	Texte	E		Non null
Rue_Client	Nom de la rue du client	Texte	E		
Ville_Client	Nom de la ville du client	Texte	E		
Code_Prod	Numéro de produit	Alpha-Numérique			unique
Qte	Quantité du produit commandée	Numérique	E		Entier > 0
PU	Prix unitaire du produit	Numérique	E		
Total	Montant total facture	Numérique	Calculé	Σ montant	



Recherche des dépendances fonctionnelles

- Les D.F. simples
 - $N_Comd \rightarrow Date_Comd$
 - $N_Comd \rightarrow NumClient$
 - $Code_Prod \rightarrow PU$
 - $N_Comd, Code_Prod \rightarrow Qte$
 - $NumClient \rightarrow Rue_Client$
 - $NumClient \rightarrow Ville_Client$
 - $NumClient \rightarrow NomClient$



Modèle Conceptuel des Données (MCD)



Modèle Conceptuel de Données : MCD

- Le MCD est une représentation graphique des données élémentaires qui sont utilisées par le SI.
- On établit le MCD après avoir recensé et donné un nom à l'ensemble des données du domaine Étudié. Ensuite on étudie les relations existantes entre ces données (les dépendances Fonctionnelles), pour aboutir au MCD.

Il est défini par le modèle ENTITE/ASSOCIATION.



MCD - Entité

- Une entité représente un ensemble d'objets ayant les mêmes caractéristiques.
- Dans une entité, on met les informations nécessaires et suffisantes pour caractériser cette entité. Ces informations sont appelées **propriétés**. Les propriétés sont collectées lors de l'établissement du dictionnaire des données. Les propriétés prennent des valeurs pour chaque occurrence d'une entité.
- Une propriété particulière, appelée **identifiant**, permet de distinguer sans ambiguïté toutes les occurrences de l'entité. L'identifiant est toujours souligné. L'identifiant est une propriété qui ne peut pas changer au cours du temps pour une occurrence.



Entité : Représentation graphique

Nom de l'entité
Nom de propriété1
Nom de propriété2
.....

Étudiant
<u>numéro</u>
nom
Prénom

Définit le format de
représentation de tous
les étudiants

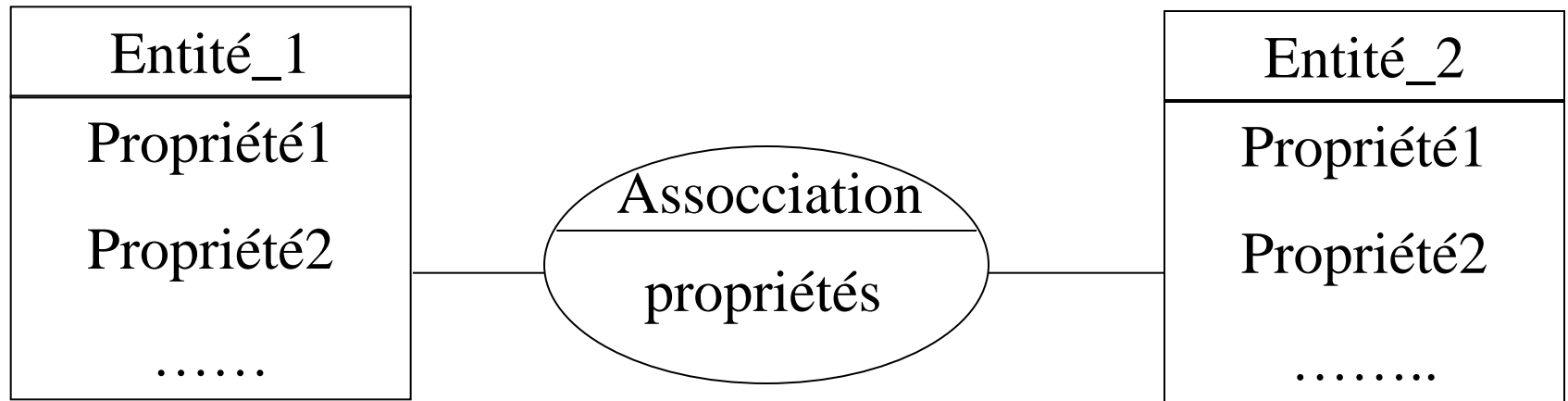


MCD - Association

- Une *association* (appelée aussi parfois *relation*) est un lien sémantique entre deux ou plusieurs entités.
- On doit lui donner un nom, souvent un verbe, qui caractérise le type de relation entre les entités. Une association possède parfois des propriétés.

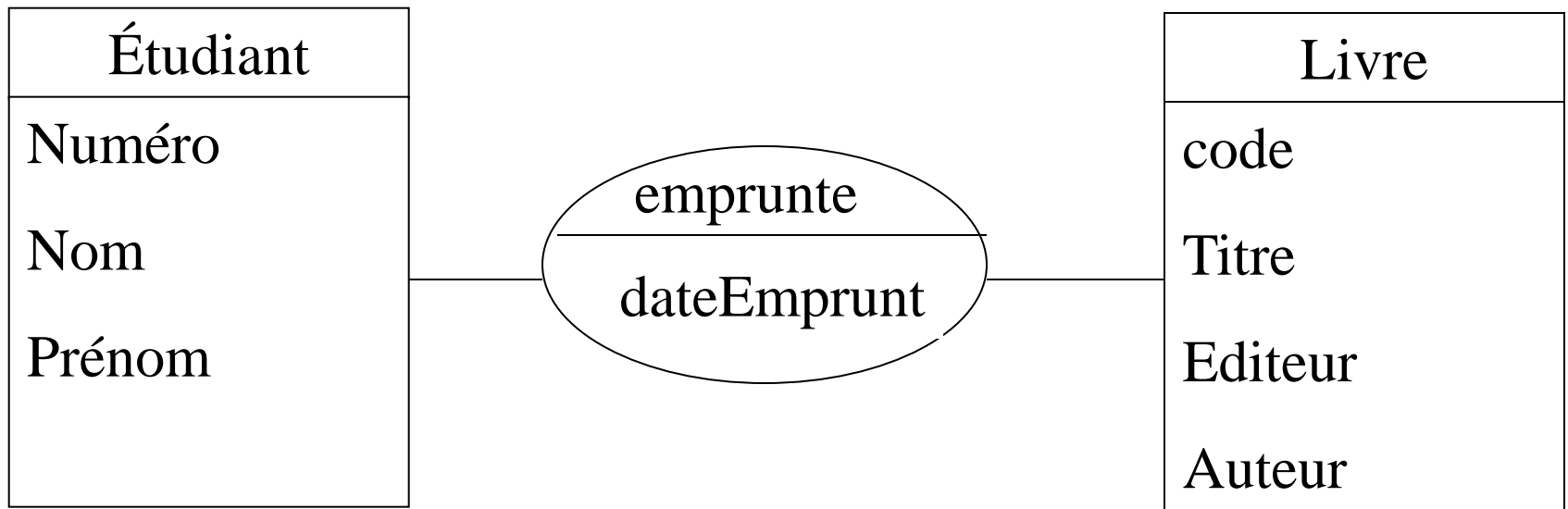


Association : Représentation graphique



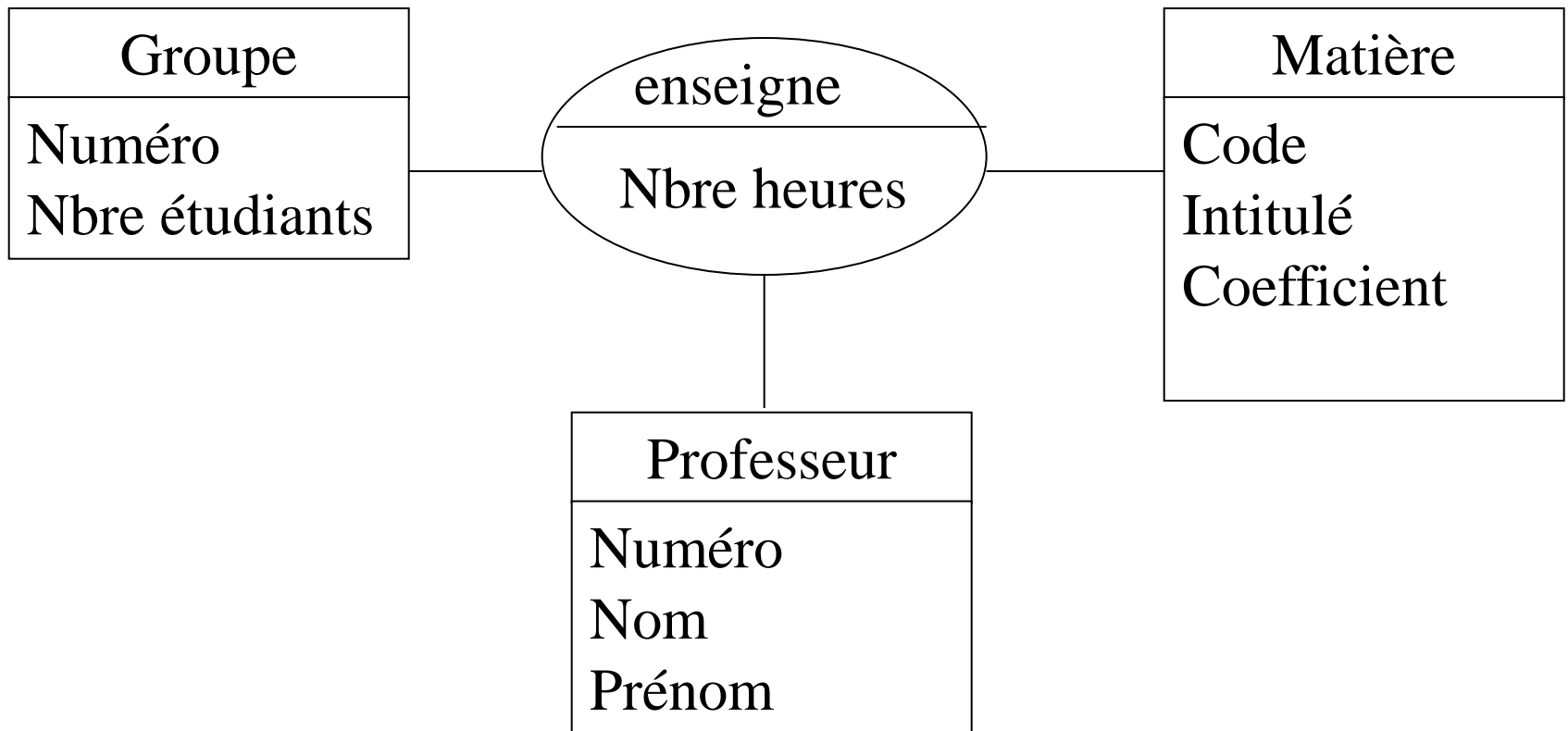


Association Binaire





Association Ternaire





Occurrence d'une propriété

Une occurrence d'une propriété est une valeur possible pour cette propriété.

Pour l'entité étudiant, la propriété prénom peut prendre les valeurs « jean », « Ali », « Alice », etc.

L'ensemble de ces valeurs forme l'ensemble des occurrences de la propriété nom (pour l'entité étudiant).



Occurrence d'une entité

Une occurrence d'une entité est un ensemble d'occurrences de ses propriétés.

Pour l'entité étudiant, nous pouvons distinguer les 4 occurrences suivantes :

Étudiant
ET01
Baba
Ali

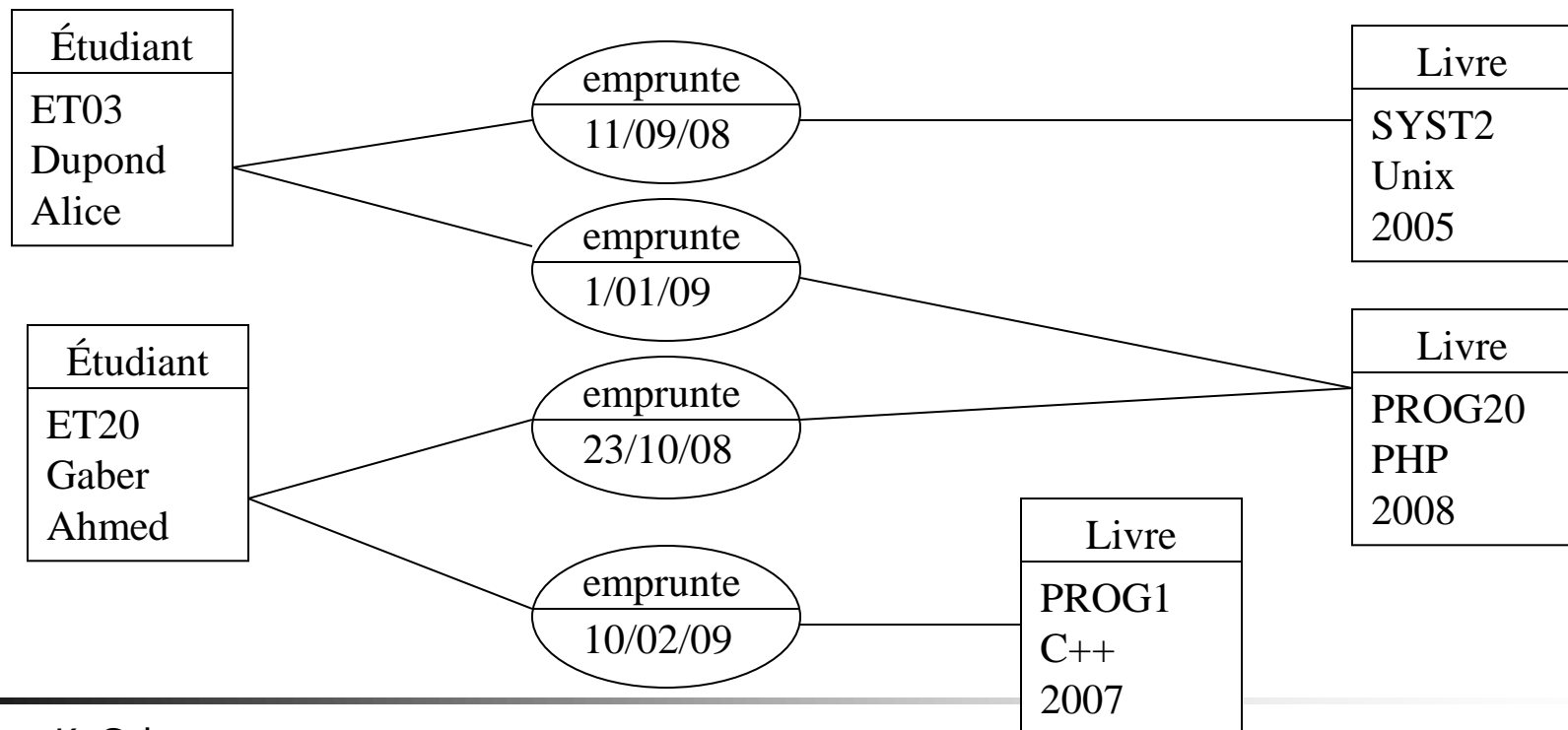
Étudiant
ET02
Quentin
Jean

Étudiant
ET03
Dupond
Alice

Étudiant
ET20
Gaber
Ahmed

Occurrence d'une association

L'occurrence d'une association correspond au nombre de fois où des occurrences d'entités sont liées :

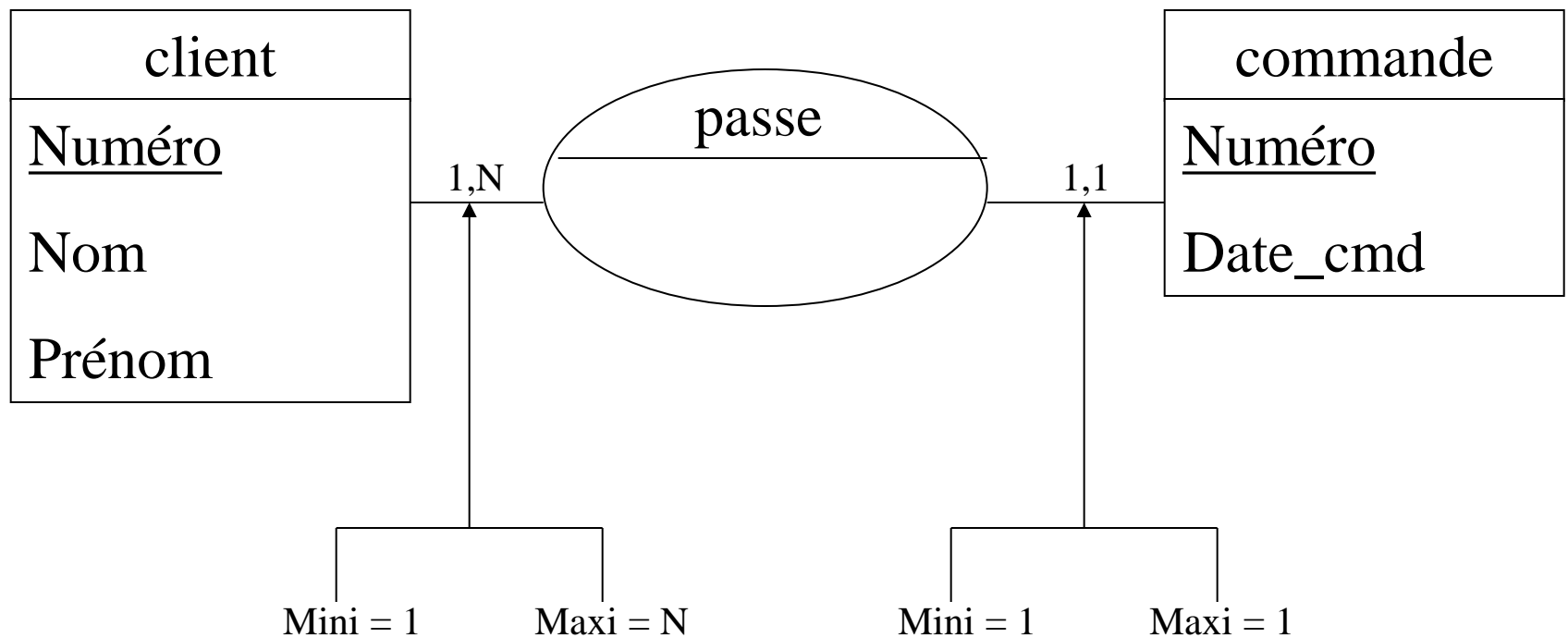




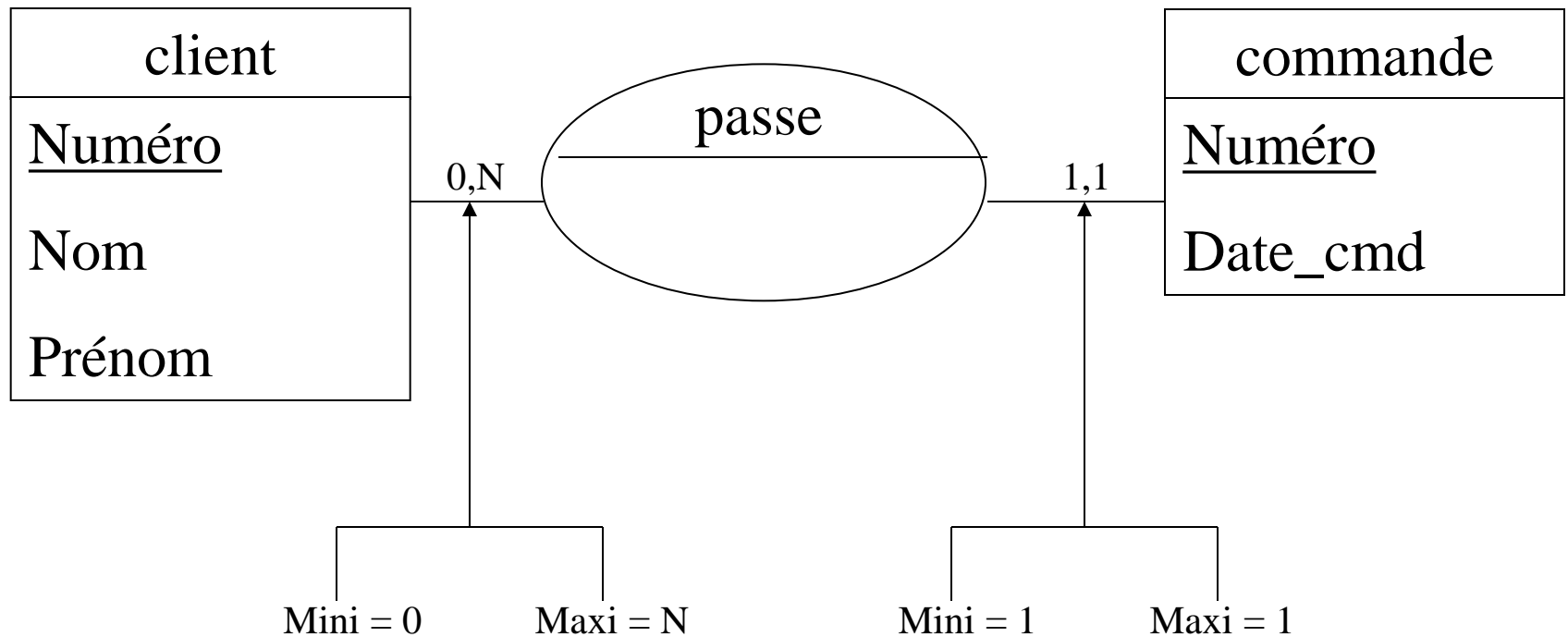
MCD - Cardinalité

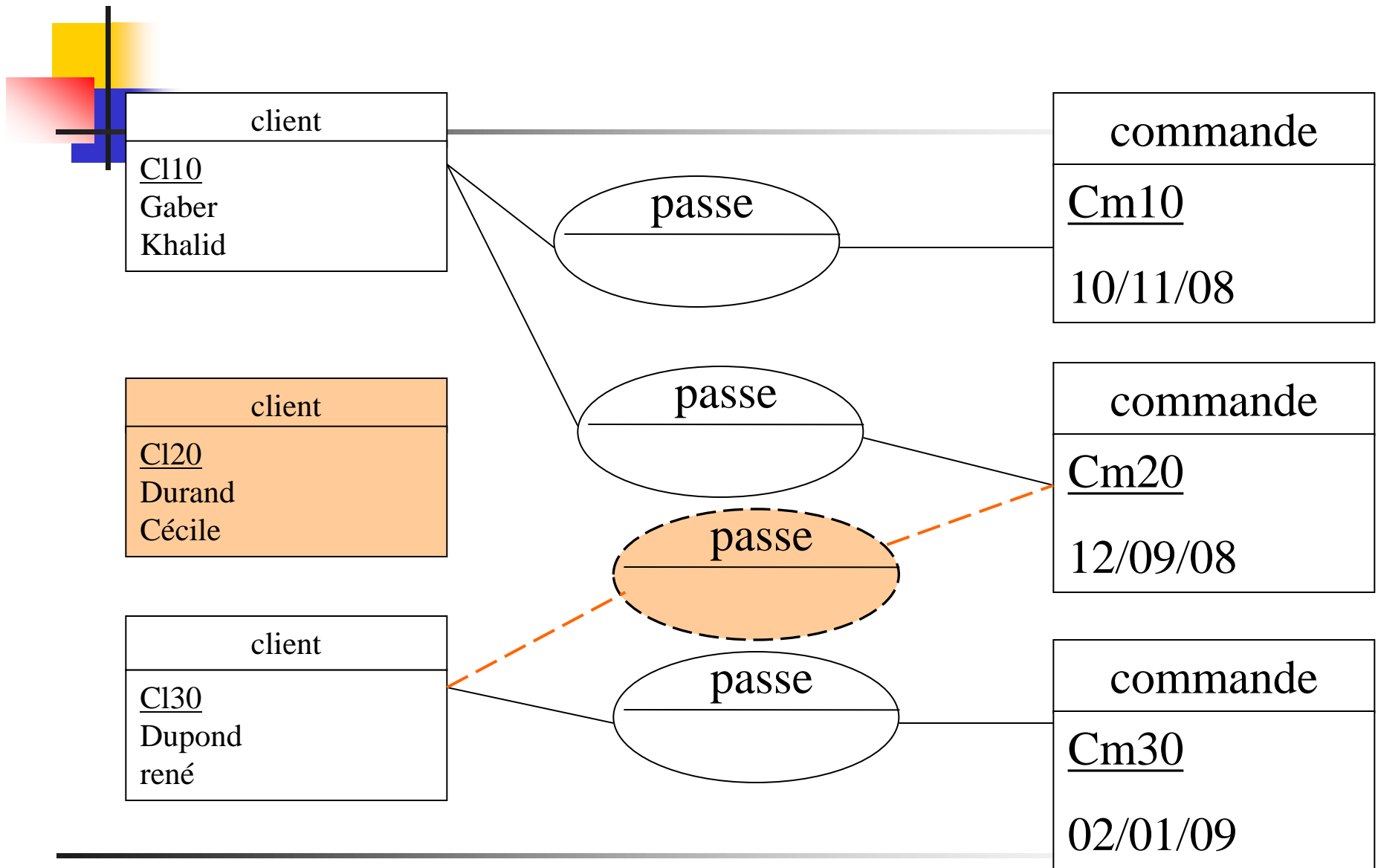
- Ce sont des expressions qui permettent d'indiquer combien de fois au minimum et au maximum le lien entre 2 entités peut se produire.
- Il y a trois valeurs typiques : 0, 1 et N (plusieurs).

Cardinalité



Cardinalité

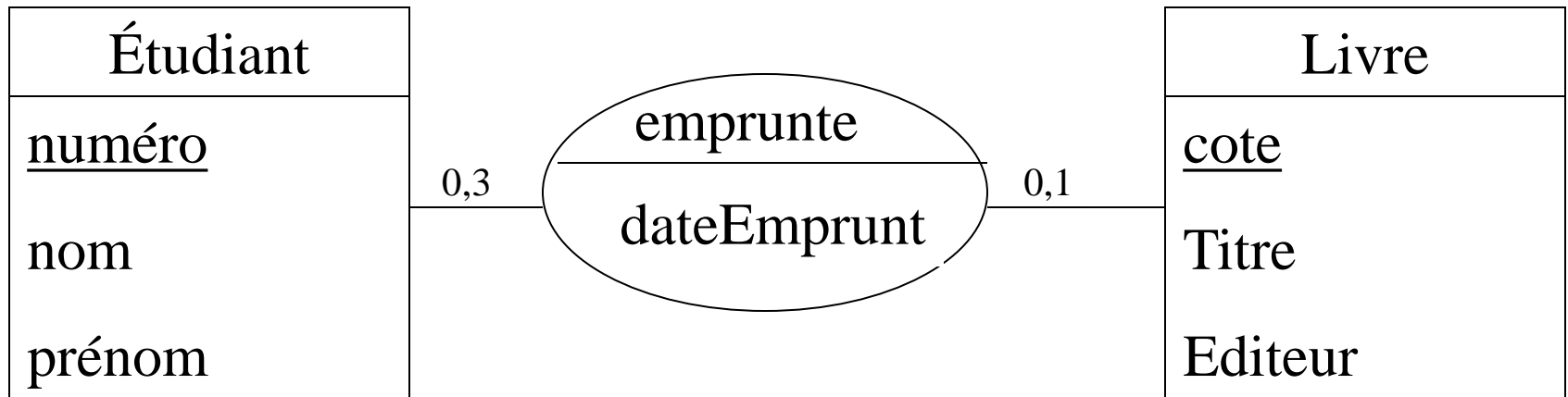




Cardinalité

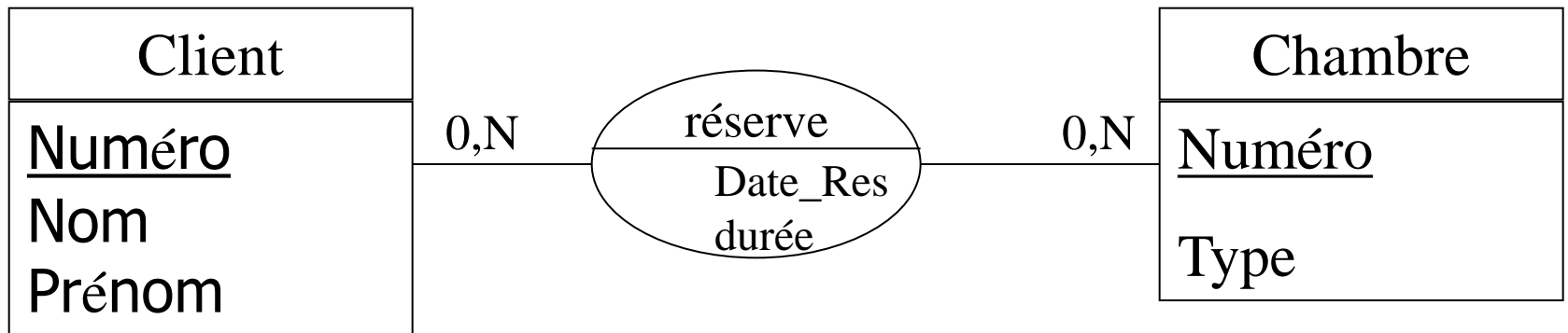
Il peut arriver que la cardinalité maximale soit connue et supérieure à 1.

Par exemple, un étudiant peut emprunter au maximum trois livres :



MCD

- ✓ A chaque occurrence d'une association correspond une et une seule occurrence de chaque entité liée
- ✓ Pour une occurrence d'une association, il doit exister au plus une valeur pour chaque propriété de l'association;



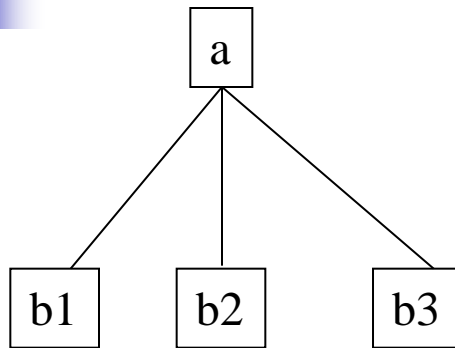
Impossible de réserver deux fois la même chambre



Types de liens

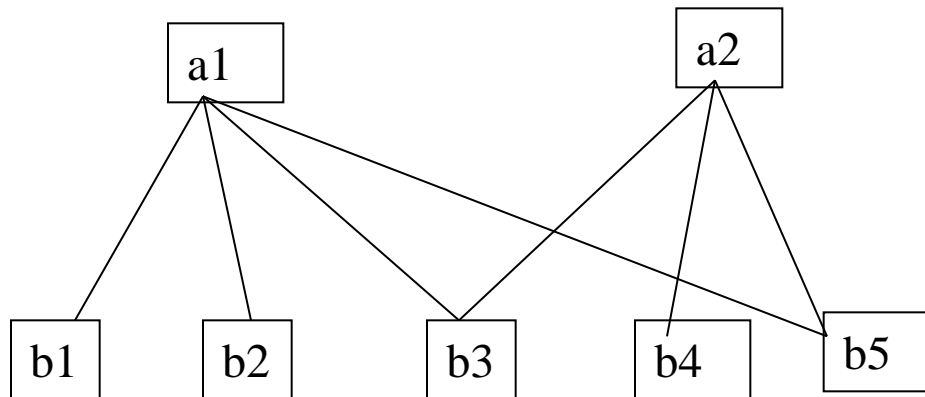
- $1,N \longrightarrow 1,1$: Lien hiérarchique
- $1,1 \longrightarrow 1,N$: Lien fonctionnel
- $1,N \longrightarrow 1,N$: Lien maillé

Types de liens



Racine \longrightarrow feuilles = hiérarchique

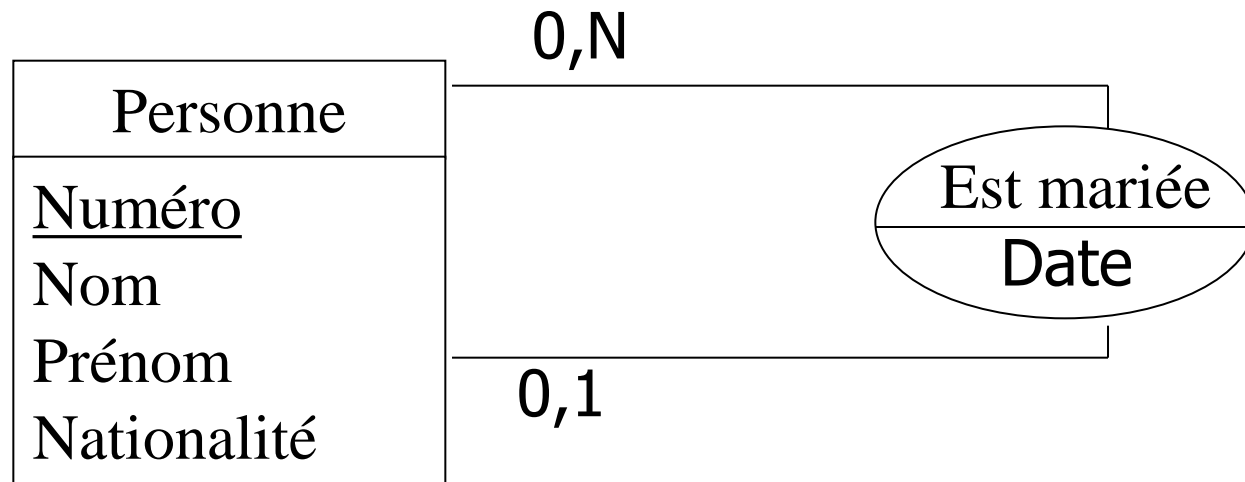
Feuilles \longrightarrow racine = fonctionnel



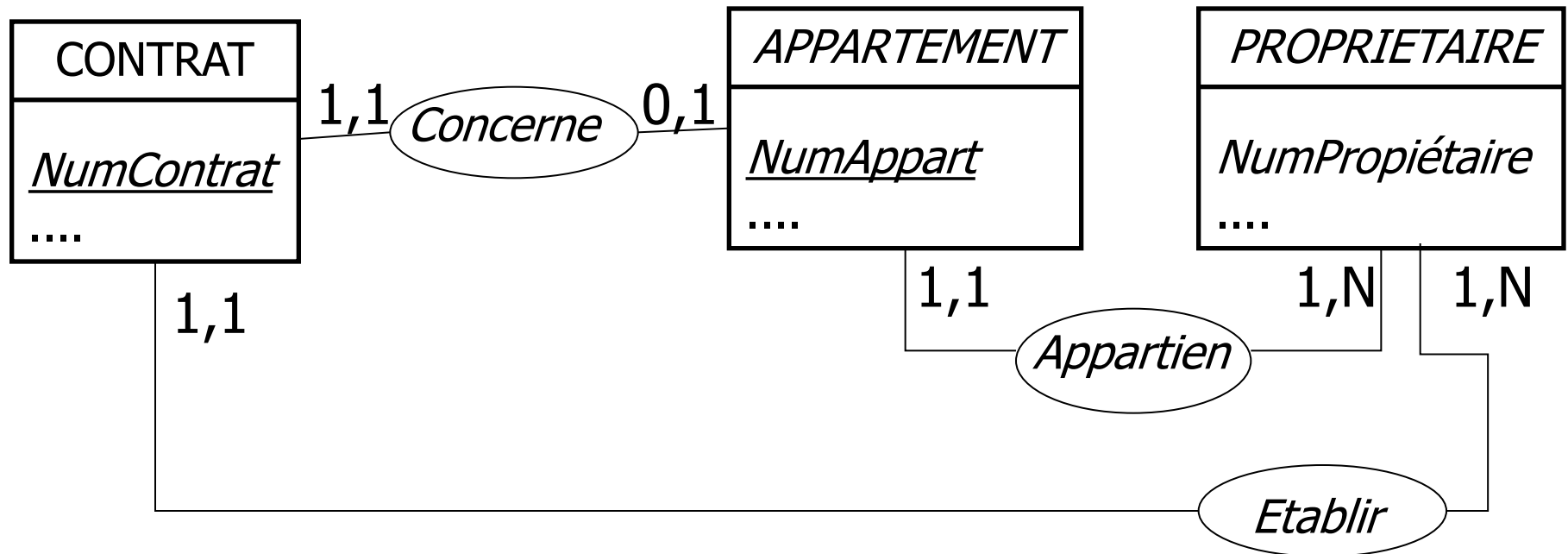
Lien maillé

Associations réflexives

On parle d'association réflexive lorsqu'une association pointe deux fois sur la même entité.



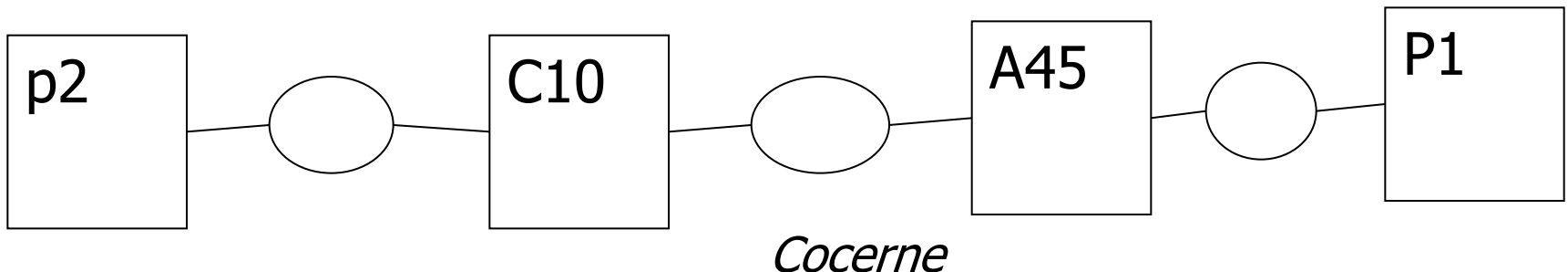
Associations transitives



L'association binaire qui relie l'entité « CONTRAT » et l'entité « PROPRIETAIRE » doit être ôtée du modèle car on peut retrouver le propriétaire à partir des associations « Concerne » et « Appartenir ». Il s'agit d'une association transitive.

Associations transitives

- On veillera à supprimer les dépendances transitives pour ne pas surcharger le MCD car on risque de faire des incohérences comme :





Dépendance fonctionnelle directe

- *Considérons l'entité suivante et quelques une de ses occurrences :*

<i>ARTICLE</i>
<i>Référence</i>
<i>Désignation</i>
<i>PrixUnitaireHT</i>
<i>NoCatégorie</i>
<i>LibelléCatégorie</i>

<i>34GQB</i>
<i>Rateau</i>
<i>3€</i>
<i>A</i>
<i>Jardinage</i>

<i>30CIA</i>
<i>Bêche</i>
<i>10€</i>
<i>A</i>
<i>Jardinage</i>

<i>54GIG</i>
<i>Scie</i>
<i>4€</i>
<i>B</i>
<i>Bricolage</i>



Dépendance fonctionnelle directe

- *Cette entité est juste mais elle implique une redondance d'information relative à la catégorie. L'association entre le numéro de la catégorie et son libellé est en effet répétée dans chaque occurrence de l'entité ARTICLE.*
- *la dépendance fonctionnelle Référence \rightarrow LibelléCatégorie n'est pas directe car il existe la propriété NoCatégorie telle que :*
Référence \rightarrow NoCatégorie et NoCatégorie \rightarrow LibelléCatégorie



RÈGLES DE TRANSFORMATION DU MCD AU MLD

- **Transformation des entités :**

Toute entité est transformée en table (sauf des cas particuliers comme les "dates"). Les propriétés de l'entité deviennent les attributs de la table. *L'identifiant* de l'entité devient *la clé primaire* de la table.



RÈGLES DE TRANSFORMATION DU MCD AU MLD : $x,N - x,1$

- **Transformation des associations binaires du type $x,N - x,1$:**
 - l'association disparaît.
 - 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



RÈGLES DE TRANSFORMATION DU MCD AU MLD : $x,1 - x,1$

■ Association binaire $0,1 - 1,1$:

- On duplique la clé primaire de la table basée sur l'entité à cardinalité $0,1$ dans la table basée sur l'entité à cardinalité $1,1$.

■ Association 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.



RÈGLES DE TRANSFORMATION DU MCD AU MLD : $x, N - x, N$

■ Transformation des associations du type $x, N - x, N$:

- On crée une table supplémentaire ayant comme clé primaire une clé composée des clés primaires de toutes les tables reliées.
- Lorsque la relation contient elle-même des propriétés, celles-ci deviennent attributs de la table supplémentaire.

→ Ces règles s'appliquent aussi bien pour les associations "*réflexives*".



Les formes normales

- Il peut arriver qu'il existe une redondance d'informations. Ceci augmente la consommation en mémoire de la base de données ainsi que la saisie des données. De plus, la mise à jour des données peut être longue voir incohérente dans certains cas. Il apparait alors nécessaire de normaliser les tables.



La première forme normale

- Une table est en première forme normale (1FN) si :
 - Elle possède une clef primaire.
 - Ses attributs sont atomiques c'est à dire qu'ils ne peuvent pas être décomposés et ne représentent pas une liste. (exemple : adresse n'est pas atomique car elle peut être décomposée en rue, code postal, ville).



Exemple

- CITOYEN(NumSecuSociale, Nom, Adresse)
n'est pas en 1FN.

Après correction, nous obtenons le schéma relationnel suivant :

- CITOYEN(NumSecuSociale, Nom, Rue,
CodePostal, Ville)
est en 1FN.



La deuxième forme normale

- Une relation est en seconde forme normale (2FN) si :
 - Elle est en 1FN.
 - Tous les attributs n'appartenant pas à la clef primaire dépendent de la totalité de cette dernière.



Exemple

COMMANDE(NumCom, DateCom, #NumCli);
PRODUIT(RefProd, descriptionProd);
LigneCommande(#NumCom, #RefProd,
PrixUnitaire, Quantité, libelleProd);

- LigneCommande est bien en 1FN mais l'attribut **libelleProd** ne dépend que d'une partie de la clef primaire **#RefProd**.

Cette relation n'est donc pas en 2FN.



Après correction, nous obtenons les relations suivantes :

- COMMANDE(NumCom, DateCom, #NumCli);
- PRODUIT(RefProd, libelleProd, descriptionProd);
- LigneCommande(#NumCom, #RefProd, PrixUnitaire, Quantité)



La troisième forme normale

- Une relation est en troisième forme normale (3FN) si :
 - Elle est en 2FN.
 - Tous les attributs n'appartenant pas à la clef primaire ne dépendent que de cette dernière et non d'autres attributs non clefs.



Exemple :

CLIENT(NumCli, NomCli, MailCli,
NumRegionCli, NomRegionClient);

- Cette relation est bien en 1FN et en 2FN.
Par contre, elle n'est pas en 3FN car
l'attribut NomRegionClient dépend
fonctionnellement de NumRegionCli.



Après correction, nous obtenons les relations suivantes :

- CLIENT(NumCli, NomCli, MailCli,
#NumRegionCli);
- REGION(NumRegion, NomRegion);