

Conception des systèmes d'information Merise et UML

Caroline DEVRED

Université d'Angers

Plan

Introduction

Merise

- Niveau conceptuel – Données
- Niveau organisationnel – Données
- Niveau logique – Données
- Niveau physique ou opérationnel
- Niveau conceptuel – Traitements
- Niveau organisationnel – traitements
- Niveau logique – Traitements

UML

- Diagramme de cas d'utilisation (Use cases)
- Diagramme de classes
- Diagramme de séquences
- Diagramme d'états-transitions

Génie logiciel

- ▶ Recherche à fournir des méthodes pour concevoir des programmes qui tiennent la route (un bug peut coûter très très cher).
- ▶ Le génie logiciel est donc ensemble de méthodes et outils pour la conception de logiciels de qualité
- ▶ Plusieurs méthodes proposées, dont :
 - ▶ Merise (vieille méthode ayant fait ses preuves)
 - ▶ UML (plus adaptée aux applications objets).
- ▶ Malheureusement rien ne peut garantir une application parfaite, restez vigilant.

A vertical toolbar containing 18 icons for document editing and presentation navigation. From top to bottom, the icons are: a red circular arrow (refresh), a red magnifying glass (search), a red curved arrow (undo), a red curved arrow (redo), a red bar chart (analytics), a red upward-pointing triangle (previous slide), a red bar chart (analytics), a red downward-pointing triangle (next slide), a red upward-pointing triangle (previous slide), a red bar chart (analytics), a red downward-pointing triangle (next slide), a red upward-pointing triangle (previous slide), a red square with a smaller square inside (full screen), a red downward-pointing triangle (next slide), a red upward-pointing triangle (previous slide), a red square (print), and a red downward-pointing triangle (next slide).

Système

Système

Ensemble d'éléments (matériels ou immatériels : hommes, machines, méthodes, règles etc.) en interaction, dont les éléments sont organisés et coordonnés pour atteindre un objectif.

Étudier le système

- ▶ Identifier le système et son environnement.
- ▶ Décomposer en sous systèmes.
- ▶ Mettre en évidence les flux.
- ▶ On ne s'intéressera qu'aux systèmes constitués par des organisations (entreprises, associations, services etc.).

Système (suite)

- ▶ En général :
- ▶ Un système opérant (SO).
- ▶ Un système de pilotage (SP).
- ▶ Un système d'information (SI).

Système d'information

- ▶ Interface entre le SO et le SP.
- ▶ Stocke et gère les informations du SO ; et les met à la disposition du système de pilotage.
- ▶ Interaction SO – SI :
 - ▶ mise-à-jour du SI par le SO ;
 - ▶ SO fait une action uniquement si le SI lui donne telle ou telle information.
- ▶ SI mémoire de l'organisation : données, structures de données, règles, contraintes, méthodes de l'organisation etc.

Système d'information (suite)

- ▶ Un aspect statique : données du monde extérieur, règles et contraintes de l'extérieur.
- ▶ Un aspect dynamique : possibilité de mise à jour (données + règles).

On informatise quoi ?

Le SO fait des actions programmées ou obéit à des choix. Le choix reste le propre de l'homme (même si certains choix peuvent être automatisés :-(). Le SI ne peut que mettre à disposition du SO des informations aidant à la décision.

Vocation de Merise

- ▶ Merise va chercher à modéliser le système.
 - ▶ Un modèle pour comprendre le système.
 - ▶ Un modèle pour communiquer.
- ▶ Proposer une méthode de conception des SI.
- ▶ Démarche méthodologique de développement des SI.

Proposer une méthode de conception des SI

- ▶ Approche globale du SI menée parallèlement sur les données et les traitements.
- ▶ Description du SI par niveau (du plus abstrait au moins abstrait). Appelé aussi cycle d'abstraction.
 - ▶ **conceptuel** (déterminé par les choix de gestion) ;
 - ▶ organisationnel (déterminé par les choix d'organisation) ;
 - ▶ logique (déterminé par les contraintes techniques) ;
 - ▶ physique (ou opérationnel – déterminé par les contraintes techniques).
- ▶ Description du SI avec le modèle «entité – relation ».
- ▶ 2 premiers niveaux indépendants de l'organisation physique et informatique.
- ▶ Représentation graphique.

Démarche méthodologique de développement des SI

- ▶ Découpage du développement en 4 étapes et deux axes.
 - ▶ Étude préalable.
 - ▶ Étude détaillée.
 - ▶ Réalisation/Mise en œuvre.
 - ▶ Les données.
 - ▶ Les traitements.
- ▶ On parle aussi de cycle de vie (naissance, maturité, maintenance). Un bouleversement profond de l'organisation et de son environnement conduit à un nouveau cycle de vie.
- ▶ Le cycle de vie et le cycle d'abstraction se mélangent.

Niveau conceptuel – Données

Niveau conceptuel

- ▶ Posons nous la question de savoir quelles sont les contraintes pesant sur le système, quels sont les résultats attendus.
- ▶ Correspond à la finalité de l'organisation.
- ▶ S'agit de décrire le QUOI.
- ▶ 2 modèles (2 invariants) :
 - ▶ Modèle Conceptuel des Données (**MCD**). Niveau statique.
 - ▶ Modèle Conceptuel des Traitements (**MCT**) Niveau Dynamique.
- ▶ INDÉPENDANT de la manière dont cela sera implémenté.

MCD

- ▶ Description des données et de leurs interactions.
 - ▶ Entité.
 - ▶ Relation.
 - ▶ Propriété.
- ▶ 2 approches :
 - ▶ Pragmatique ou intuitive.
 - ▶ Technique.

Concept d'entité

Exercice

La société «les trois belges » fonctionne de la façon suivante.
2 fois par an, elle dépose dans les boîtes aux lettres un catalogue des produits qu'elle vend. Chaque personne intéressée va passer auprès de notre société une commande. Cette commande peut bien sûr porter sur plusieurs produits. Après une quinzaine de jours, la commande est envoyée au client (le vert du pull ne correspond absolument pas à celui de la photo) ainsi qu'une facture.

Chercher à identifier les acteurs (matériel ou immatériel) intervenant (interagissant) lors du fonctionnement de l'entreprise les «trois belges». N'oubliez pas, on cherche à modéliser le SI et non le SO ou le SP.

Concept d'entité (suite)

Solution

Il y a quatre acteurs : les clients, les commandes, les produits et les factures.

Formellement

Une **entité** est un objet (! dans le sens de chose !) pourvue d'une existence propre et conforme aux choix de gestion de l'organisation.

Nom de l'entité

- ▶ On la représente de la façon suivante :
- ▶ Note pour plus tard : une entité est **complètement** définie si elle possède des propriétés (des informations).
- ▶ Entité représente ce qui est **réellement** perçu.

Concept de relation

Exercice

Les entités sont liées. Donnez les liens entre les entités des «trois belges».

Concept de relation (suite)

Solution

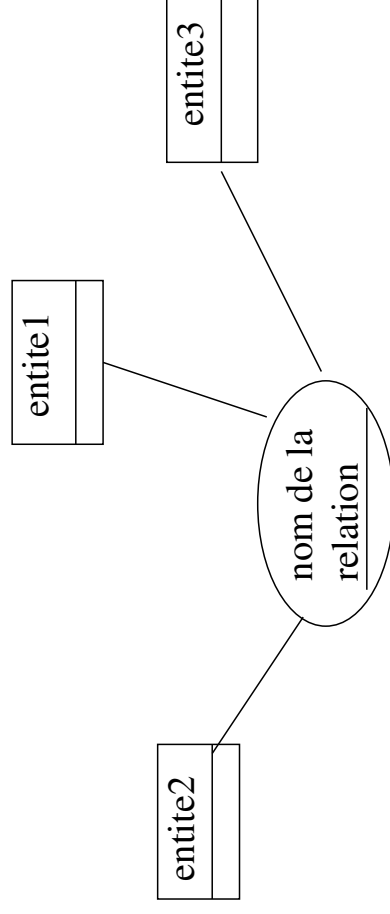
- ▶ Un client **pass**e une commande.
- ▶ Une commande **porte** sur des produits.
- ▶ Un produit **est factur**é .
- ▶ À une commande **est associ**é une ou plusieurs factures.

Concept de relation (presque fin)

Formellement

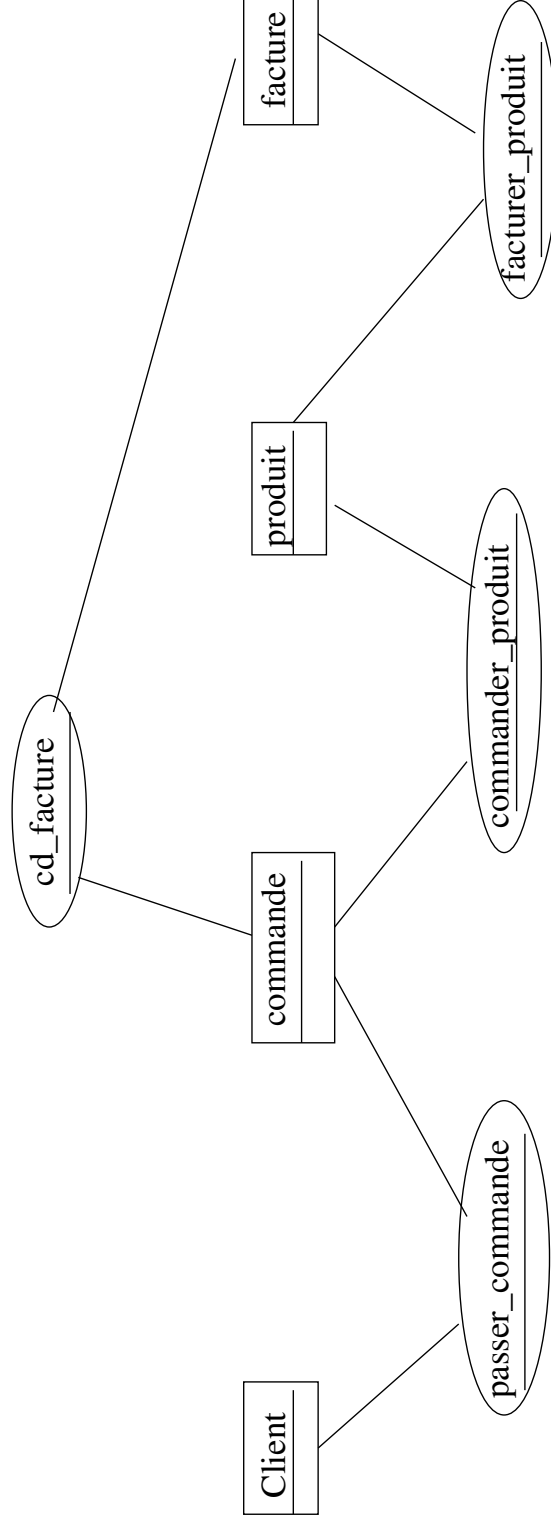
Une **relation** entre entités est une **association** perçue dans le réel par une ou plusieurs entités.

- ▶ Elle se représente ainsi :



Concept de relation (fin)

Solution



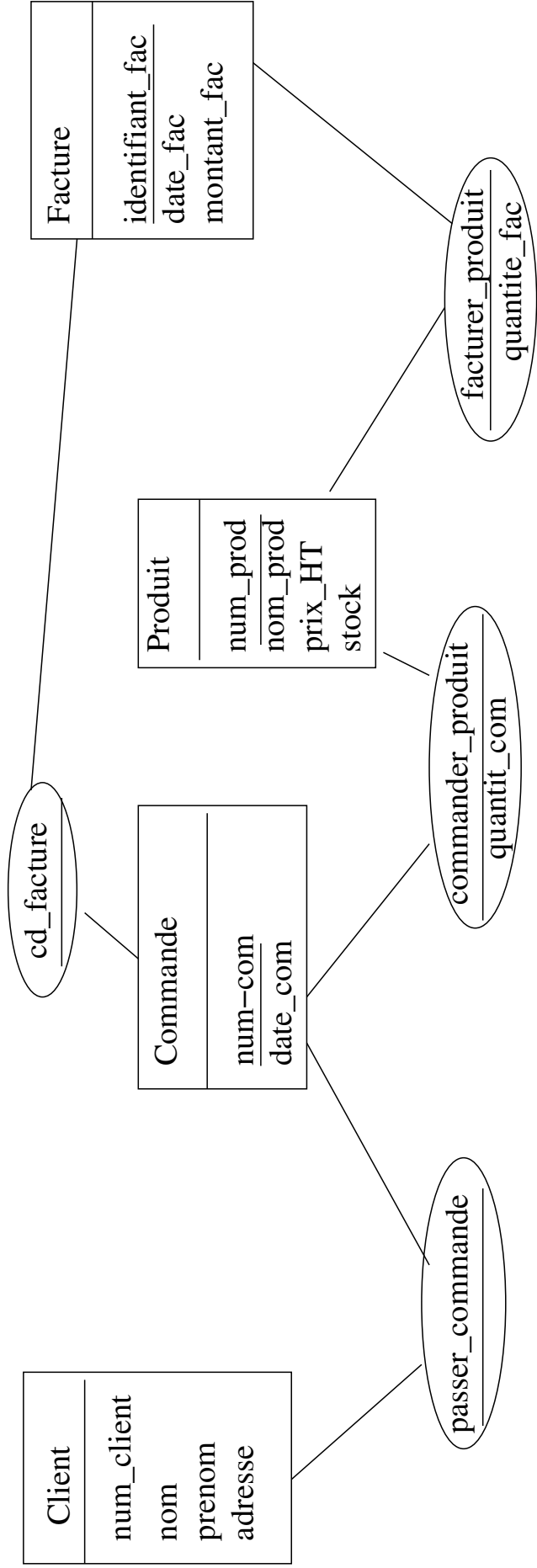
Concept de propriété

- ▶ On complète les entités et associations par les informations qui les concernent.
- ▶ Quelques idées sur notre exo fil rouge ? (Ce n'est pas une question, juste une figure de style).

Formellement

Une **propriété** (ou attribut) est une donnée élémentaire que l'on perçoit sur une entité ou sur une relation entre entités.

Propriétés de l'exemple fil rouge



Mais...

- ▶ Dans la vie réelle jamais les systèmes ne sont aussi simples, et construire les MCD à la volée ne garantit pas d'avoir un MCD valide :(
- ▶ Il faut faire les choses avec méthode :
 - ▶ dictionnaire des données,
 - ▶ dictionnaire des données élémentaires,
 - ▶ dépendances fonctionnelles,
 - ▶ graphe épuré des dépendances fonctionnelles,
 - ▶ passage au MCD,
 - ▶ cardinalités.

Dictionnaire des données

- ▶ À la recherche des propriétés perdues.
- ▶ Le but : identifier chaque propriété (chaque [information](#)), la nommer, lui donner sa signification, son type (voir plus loin), sa longueur (i.e. la place qu'elle va prendre) sa nature et une contrainte d'intégrité ou une règle de calcul.
- ▶ Toujours garder une place à table pour les identifiants.

Un peu plus sur les propriétés

- ▶ Plusieurs natures (ou encore classes) :
 - ▶ Élémentaire (**E**) ou concaténée (**CO**).
 - ▶ Calculée (**C**).
 - ▶ Paramètre (**P**).
- ▶ Plusieurs types :
 - ▶ Numérique (**N**).
 - ▶ Alphabétique (**A**).
 - ▶ Alphanumérique (**α**).
 - ▶ Date (**D**).
- ▶ Contrainte d'intégrité (CI) : ce que la propriété doit vérifier pour être reconnue comme valable.
- ▶ Règle de calcul (RC) : calcul à effectuer pour obtenir la valeur de la propriété.

Dictionnaire des données des «trois belges»

Nom	Signification	Type	Longueur	Nature	Cl ou RC
num_client	l'id du client	N	4	E	pas 2 pareils
nom	le nom du client	A	20	E	
prénom	le prénom du client	A	20	E	
adresse	l'adresse du client	α	150	CO	pas 2 pareils \leq date du jour
num_com	le numéro de la commande	N	4	E	
date	la date de la commande	D	20	E	
client_com	client passant commande	N	4	E	pas 2 pareils
num_prod	identifiant du produit	N	4	E	
nom_prod	nom du produit	α	50	E	
prix_HT	prix HT du produit	N	4	E	≥ 0 $= 0.186$ $\text{prix_TTC} = \text{prix_HT} \times (1 + \text{TVA})$
TVA	TVA	N	4	E	
prix_TTC	prix avec TVA	N	4	P	
stock	indique le stock du produit	N	4	C	≥ 0
prod_com	produit commandé	N	4	E	
prod_fact	produit facturé	N	4	E	
quantité_com	quantité de produit commander	N	4	E	≥ 0 $\leq \text{quantité_com}$
quantité_fac	quantité de produit facturée	N	4	E	
identifiant_fac	numéro de facture	N	6	E	
date_fac	date de la facture	D	20	E	pas 2 pareils \leq date du jour
montant_fac	montant de la facture	N	6	CA	
					somme des prix_TTC facturés

Dictionnaire des données élémentaires

- ▶ On épure le dictionnaire des données :
 - ▶ Adieu propriété paramètre ou calculée.
 - ▶ Adieu synonyme (2 propriétés représentant la même chose).
 - ▶ Adieu polysème (2 propriété de même nom mais représentant 2 informations différentes).
 - ▶ Morcellement des données concaténées.

Dictionnaire des données élémentaires des «trois belges»

Nom	Signification	Type	Longueur	Nature	CI ou RC
num_client	l'id du client	N	4	E	pas 2 pareils
nom	le nom du client	A	20	E	
prénom	le prénom du client	A	20	E	
num_ad	num de l'adresse du client	N	3	E	
rue_ad	rue de l'adresse du client	A	20	E	
code_postal	code postal du client	N	5	E	5 chiffres
ville	ville du client	A	20	E	
num_com	le numéro de la commande	N	4	E	pas 2 pareils
date_com	la date de la commande	D	20	E	≤ date du jour
num_prod	identifiant du produit	N	4	E	pas 2 pareils
nom_prod	nom du produit	α	50	E	
prix_HT	prix HT du produit	N	4	E	≥ 0
stock	indique le stock du produit	N	4	E	≥ 0
quantité_com	quantité de produit commander	N	4	E	≥ 0
quantité_fac	quantité de produit facturée	N	4	E	≤ quantité_com
identifiant_fac	numéro de facture	N	6	E	pas 2 pareils
date_fac	date de la facture	D	20	E	≤ date du jour

Dépendances fonctionnelles

- ▶ On cherche les liens entre les éléments du dictionnaire des données élémentaires.

Formellement

- ▶ On dit que 2 propriétés sont reliées par une **dépendance fonctionnelle** si la valeur de l'une détermine au plus une valeur de l'autre.
- ▶ On dit que **b dépend fonctionnellement** de **a** si une valeur de **a** n'entraîne qu'une valeur de **b** .
 - ▶ On note **$a \xrightarrow{df} b$** ;
 - ▶ **a** est la **source** de la dépendance fonctionnelle ;
 - ▶ **b** est le **but** de la dépendance fonctionnelle.

Classification des dépendances fonctionnelles

- ▶ Une **dépendance fonctionnelle simple** n'a qu'un but et qu'une source ($\text{num_client} \xrightarrow{df} \text{nom}$).
- ▶ Une **dépendance fonctionnelle à partie gauche composée** est une dépendance fonctionnelle dont la source est composée de plusieurs propriétés ($(\text{num_com}, \text{num_pro}) \xrightarrow{df}$ quantité_com).
- ▶ Aucune sous-partie de la source d'une **dépendance fonctionnelle élémentaire** ne permet d'en déduire le but ($(\text{num_com}, \text{num_pro}) \xrightarrow{\text{red}} \text{quantité_com et } (\text{num_client}, \text{nom}) \xrightarrow{df} \text{rue_adresse}$).
- ▶ Une **dépendance fonctionnelle directe** $a \xrightarrow{df} b$ est une dépendance fonctionnelle telle qu'il n'existe pas de propriété c telle que $a \xrightarrow{df} c$ et $c \xrightarrow{df} b$.

Les dépendances fonctionnelles en vue d'un MCD

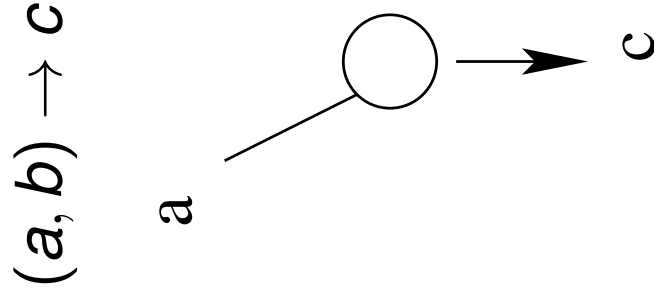
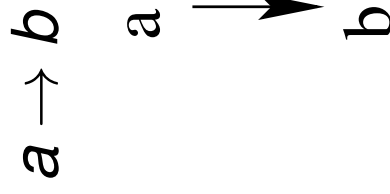
- ▶ Les dépendances fonctionnelles qui nous intéressent sont [les dépendances fonctionnelles élémentaires directes](#) .
- ▶ Cherchez les dépendances fonctionnelles des «trois belges» et morcelez-les pour obtenir uniquement des dépendances fonctionnelles élémentaires directes.

Les dépendances fonctionnelles des «trois belges»

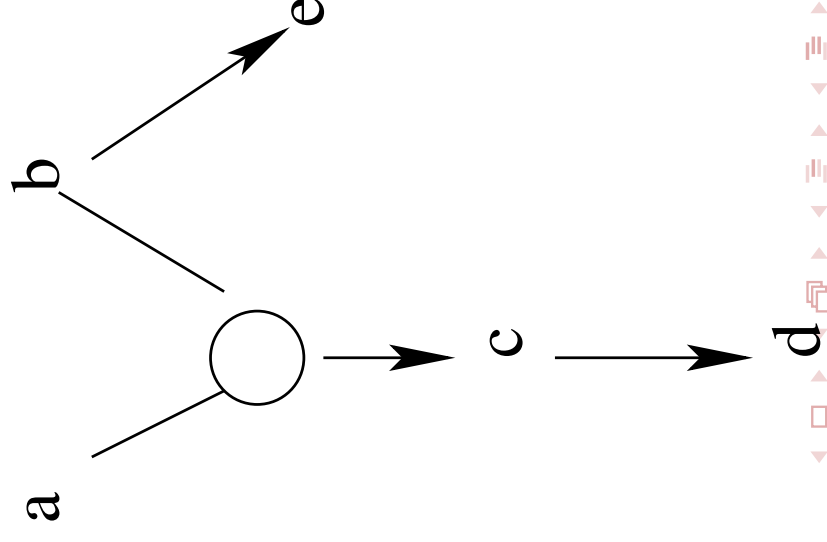
- ▶ num_client → nom
- ▶ num_client → prénom
- ▶ num_client → num_ad
- ▶ num_client → rue_ad
- ▶ num_client →
code_postal
- ▶ num_client → ville
- ▶ num_com → date_com
- ▶ num_produit →
nom_prod

- ▶ num_produit → prix_HT
- ▶ num_produit → stock
- ▶ identifiant_fac → date_fac
- ▶ identifiant_fac → num_com
- ▶ num_com → num_client
- ▶ (identifiant_fac , num_produit)
→ quantité_fac
- ▶ (num_com , num_produit) →
quantité_com

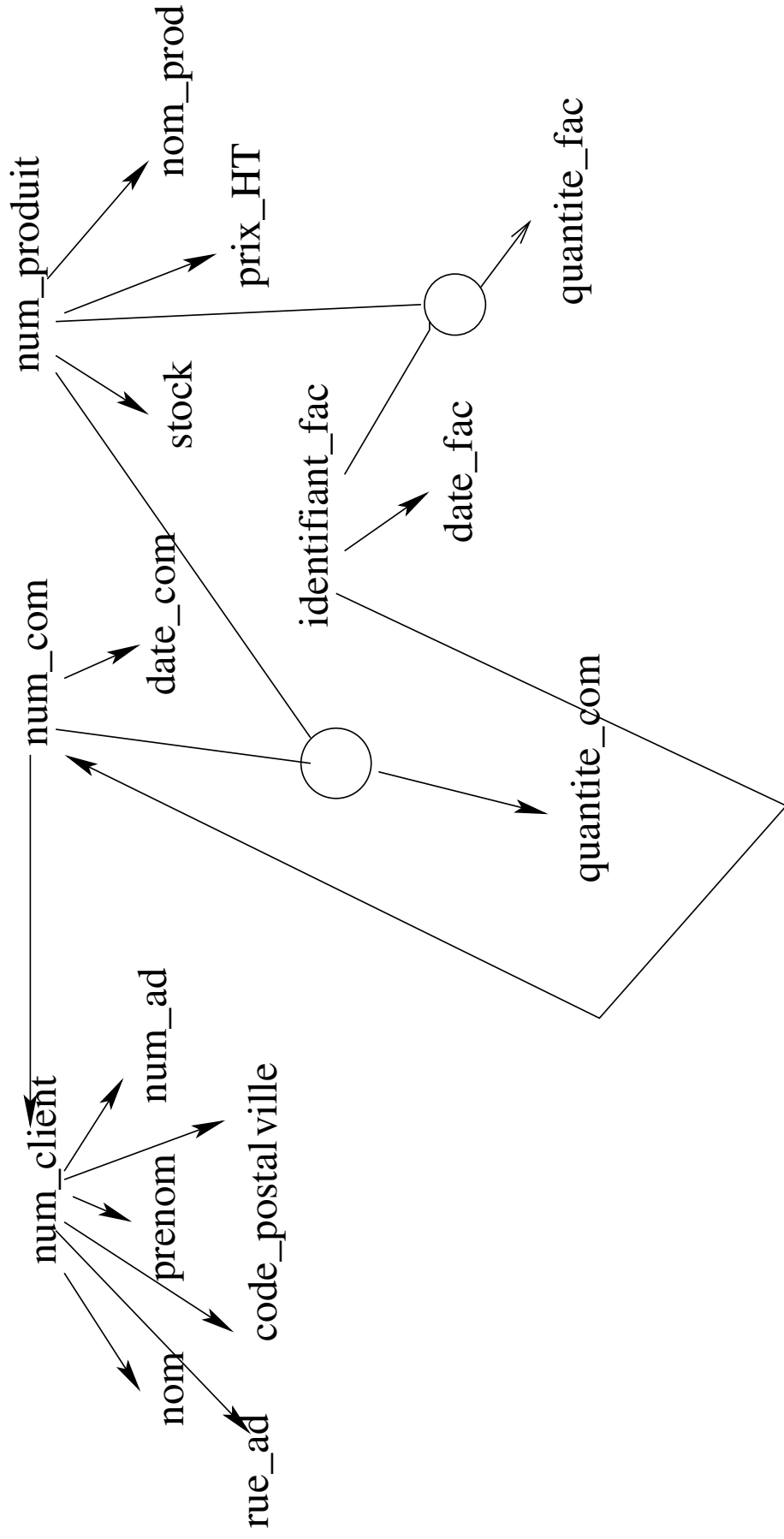
Représentation graphique des dépendances fonctionnelles et graphe des dépendances fonctionnelles



$$(a, b) \rightarrow c; c \rightarrow d; b \rightarrow e$$



Le graphe des dépendances fonctionnelles des «trois belges»

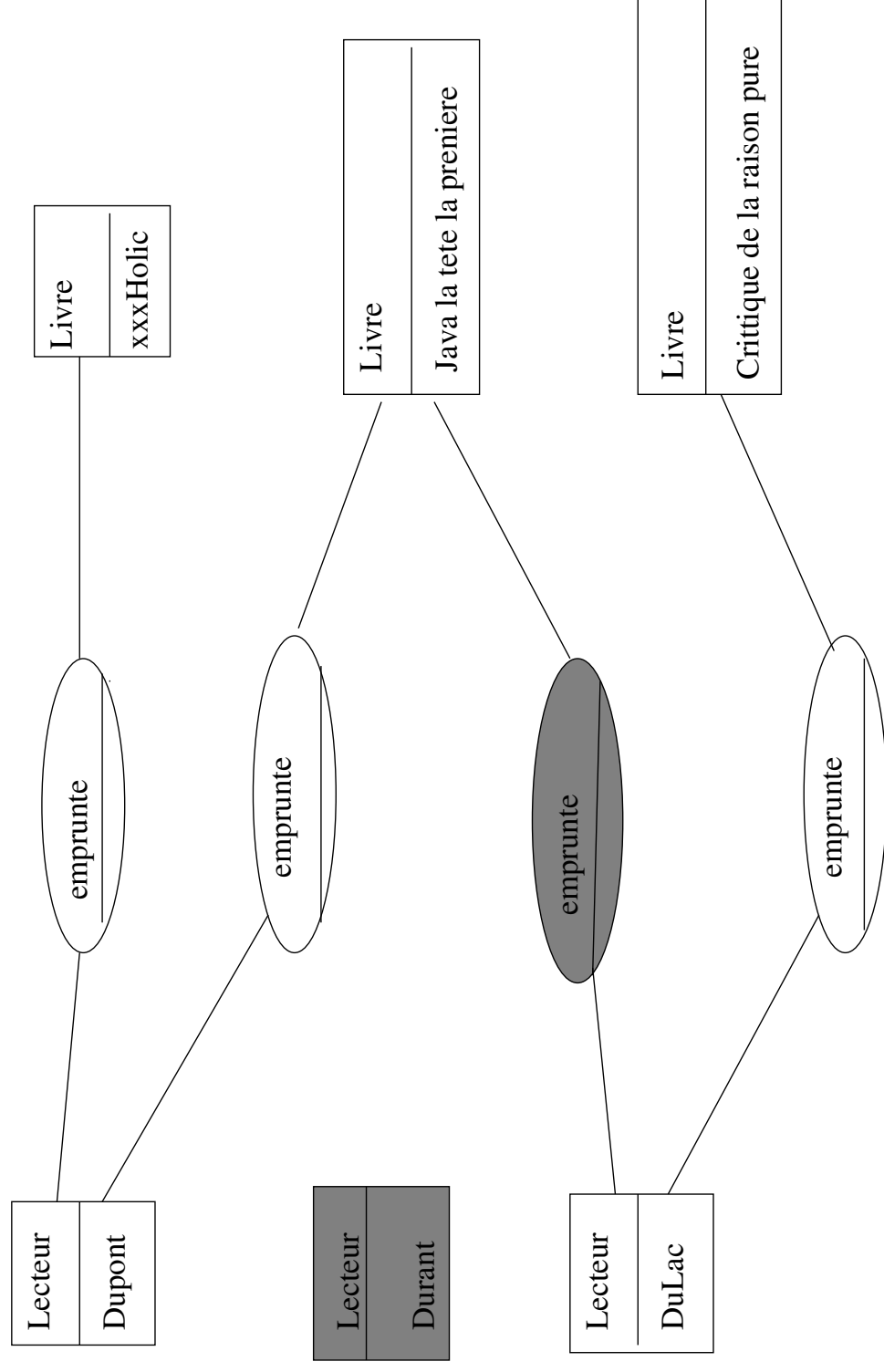


Le passage du graphe des dépendances fonctionnelles au MCD

- ▶ Il ne reste plus qu'à passer du graphe des dépendances fonctionnelles au MCD, i.e. quel groupe de propriétés forme une entité et quelles entités sont en relation.
- ▶ On rappelle qu'une entité est une «chose» réelle du SI et qu'une relation représente les interactions entre les entités.

La cardinalité objet-relation

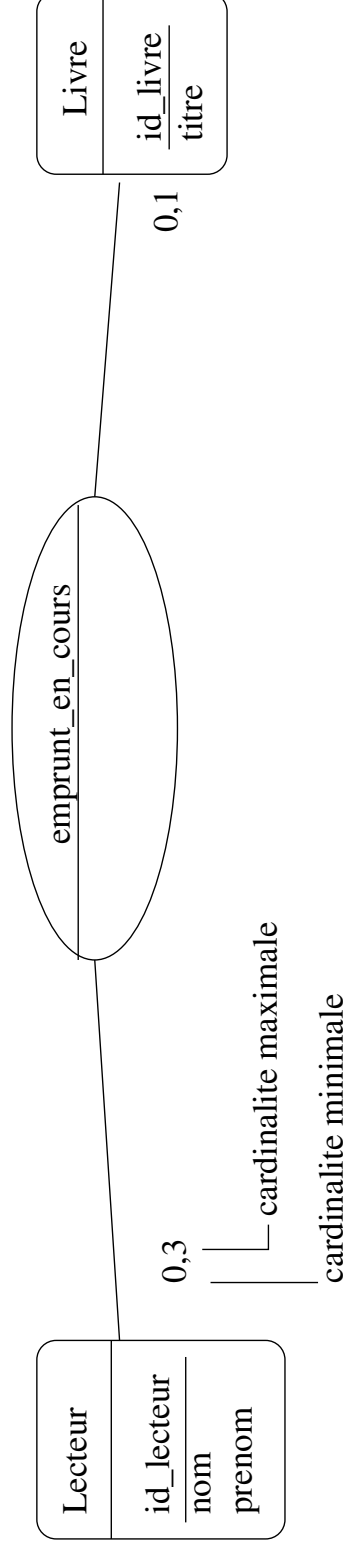
- Le diagramme d'occurrence.



La cardinalité objet-relation

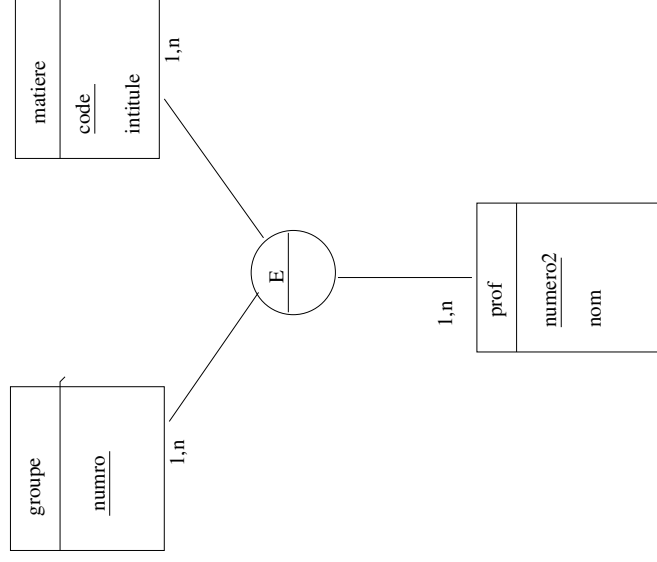
- ▶ La **cardinalité d'une relation** est le nombre minimal ou maximal d'occurrences d'une **entité pouvant prendre part à une relation** .
 - ▶ La **cardinalité minimale** est le nombre minimum de fois où une occurrence d'une entité peut appartenir aux occurrences de la relation.
 - ▶ La **cardinalité maximale** est le nombre maximum de fois où une occurrence d'une entité peut appartenir aux occurrences de la relation.

Exemple de cardinalité objet-relation binaire



Un Lecteur peut emprunter de 0 à 3 livre. Pour un Lecteur combien de livre au minimum, combien de livre au maximum ?

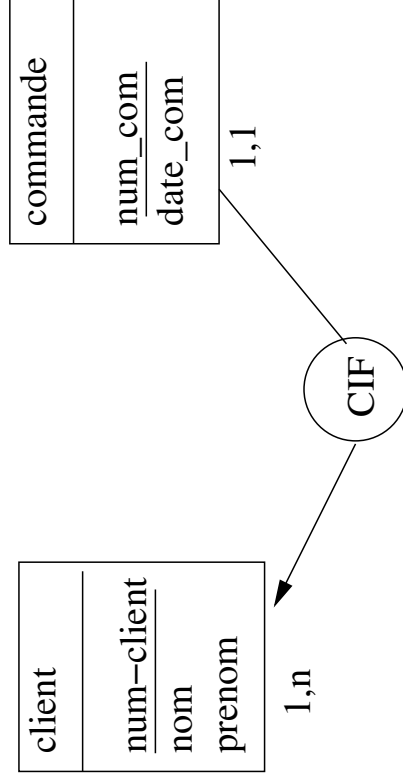
Exemple de cardinalité objet-relation n-à-n



Pour un professeur combien de couples (Groupe, Matière) au minimum, combien de couples (Groupe, Matière) au maximum ?

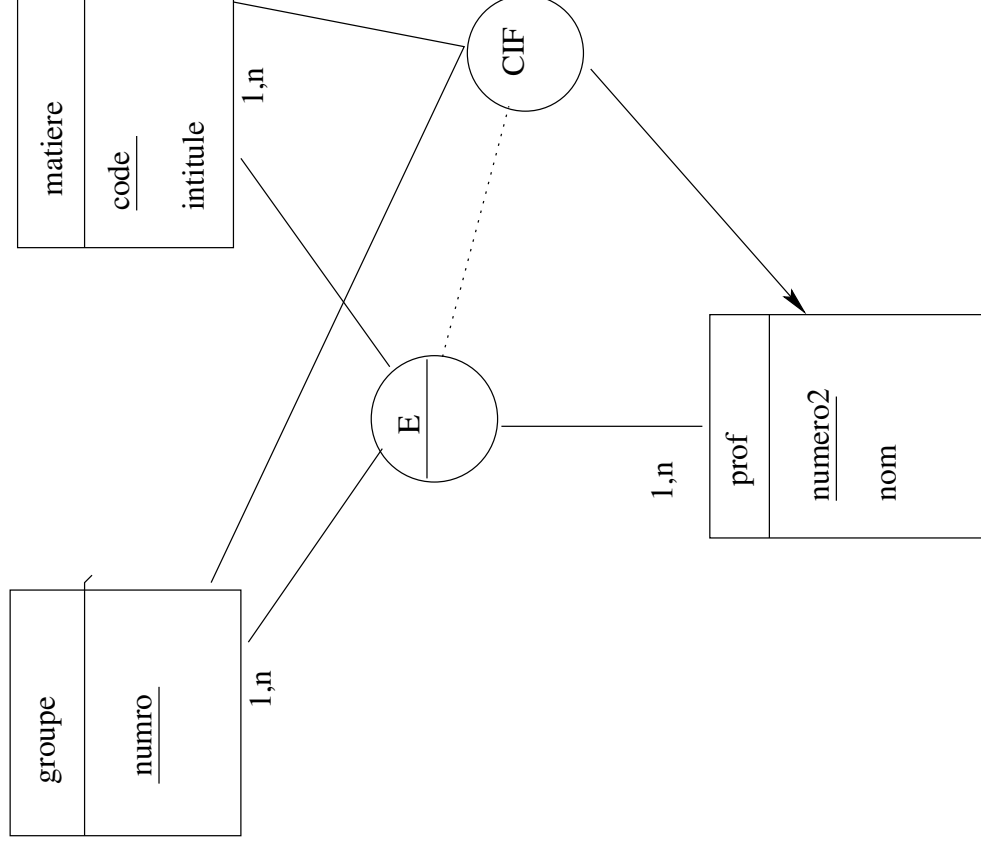
Contrainte d'intégrité fonctionnelle

Certaines cardinalités vont générer des associations particulières : Lorsque la **cardinalité** maximale d'une entité dans une association binaire est égale à 1, c'est à dire lorsqu'une entité est totalement connue lorsqu'on en connaît une autre.



Contrainte d'intégrité fonctionnelle

Il peut y avoir des **CIF** sur une association non binaire.



Les identifiants

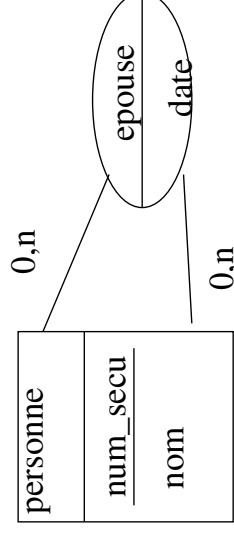
- ▶ D'une entité : c'est une propriété particulière de l'entité telle qu'à chaque valeur de la propriété corresponde une unique occurrence de l'entité. On la représente soulignée.
- ▶ D'une relation : c'est la concaténation des identifiants des entités appartenant à la relation. Elle a le même but.

Dimension d'une relation

- ▶ C'est le nombre d'entités participant à la relation.
- ▶ Une **relation binaire** concerne 2 entités.
- ▶ Une **relation ternaire** concerne 3 entités.
- ▶ Une **relation n-aire** concerne n-entités.

Relation réflexive

- ▶ C'est une relation d'une entité avec elle même.



- ▶ Un petit mot sur la polygamie. En vrai cela s'appelle **règle de gestion**.

Du graphe des dépendances fonctionnelles au MCD

- ▶ Déterminer les rubriques sources.
- ▶ Déterminer les entités.
- ▶ Déterminer les CIF binaire.
- ▶ Déterminer les CIF multiples.
- ▶ Déterminer les autres relations.
- ▶ Ajouter les relations non induites par le graphe des DF.
- ▶ Répartir les propriétés entre les entités et les associations.
- ▶ Réaménager le MCD si besoin est (normalisation).

Toutes les CIF sont représentés dans le graphe des DF, c'est la seule garantie en ce qui concerne les relations.

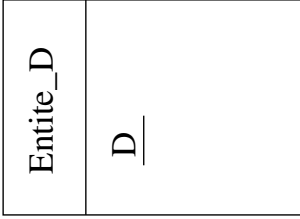
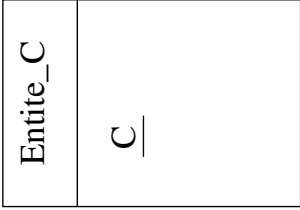
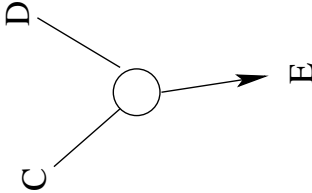
Déterminer les rubriques sources

Une rubrique source est :

- ▶ soit source de dépendance fonctionnelle simple ou d'une dépendance fonctionnelle à partie gauche composée ;
- ▶ soit le but de plusieurs dépendances fonctionnelles sans être la source d'une dépendance fonctionnelle ;
- ▶ soit une rubrique isolée.

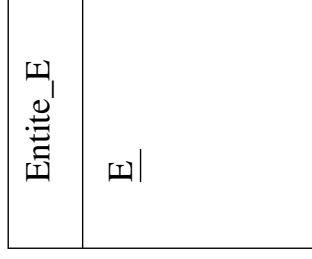
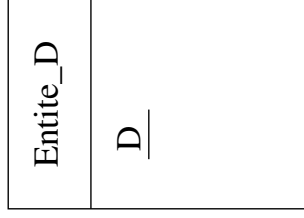
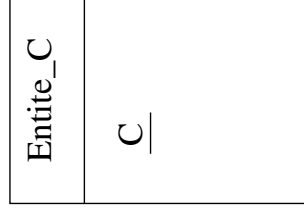
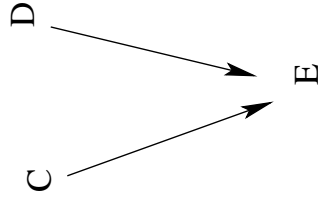
Déterminer les entités

A → B



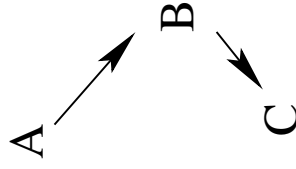
Déterminer les entités (suite)

A



Déterminer les CIF binaires

Une dépendance fonctionnelle simple, dont la source et le but sont des rubriques sources, est une CIF binaire.

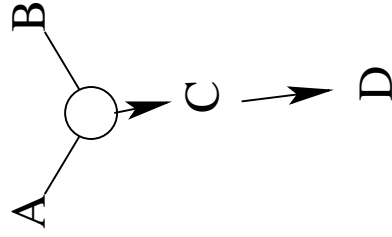


Entite_A
A

Entite_B
B

Déterminer les CIF multiples

Une dépendance fonctionnelle à partie gauche composée, dont le but est une rubrique source, est une CIF multiple.



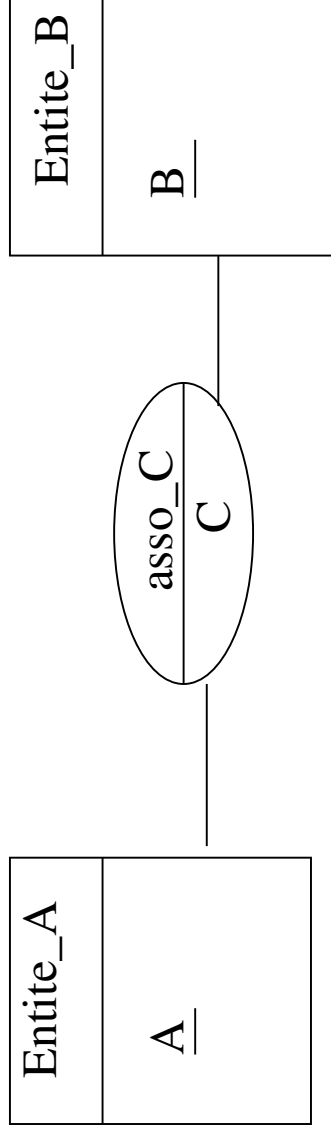
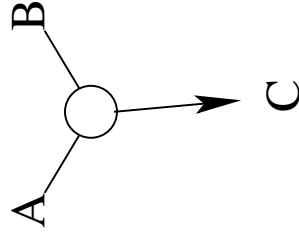
Entite_A
<u>A</u>

Entite_B
<u>B</u>

Entite_C
<u>C</u> D

Déterminer les associations

- ▶ Une dépendance fonctionnelle à partie gauche composée dont le but n'est pas une rubrique source donne une association multiple.

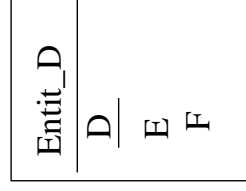
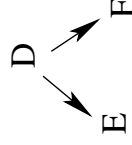
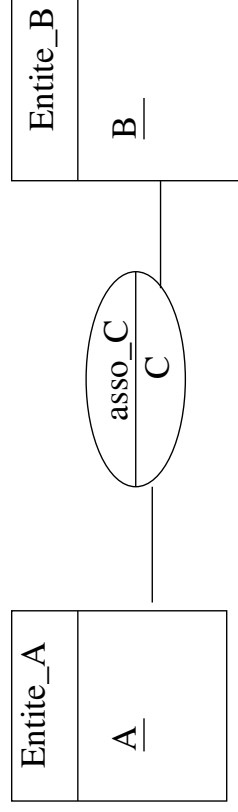
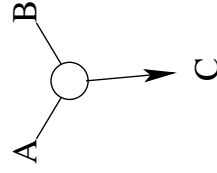


- ▶ **Attention** parfois le contexte peut faire surgir des associations entre deux rubriques sources non reliées par une dépendance fonctionnelle.

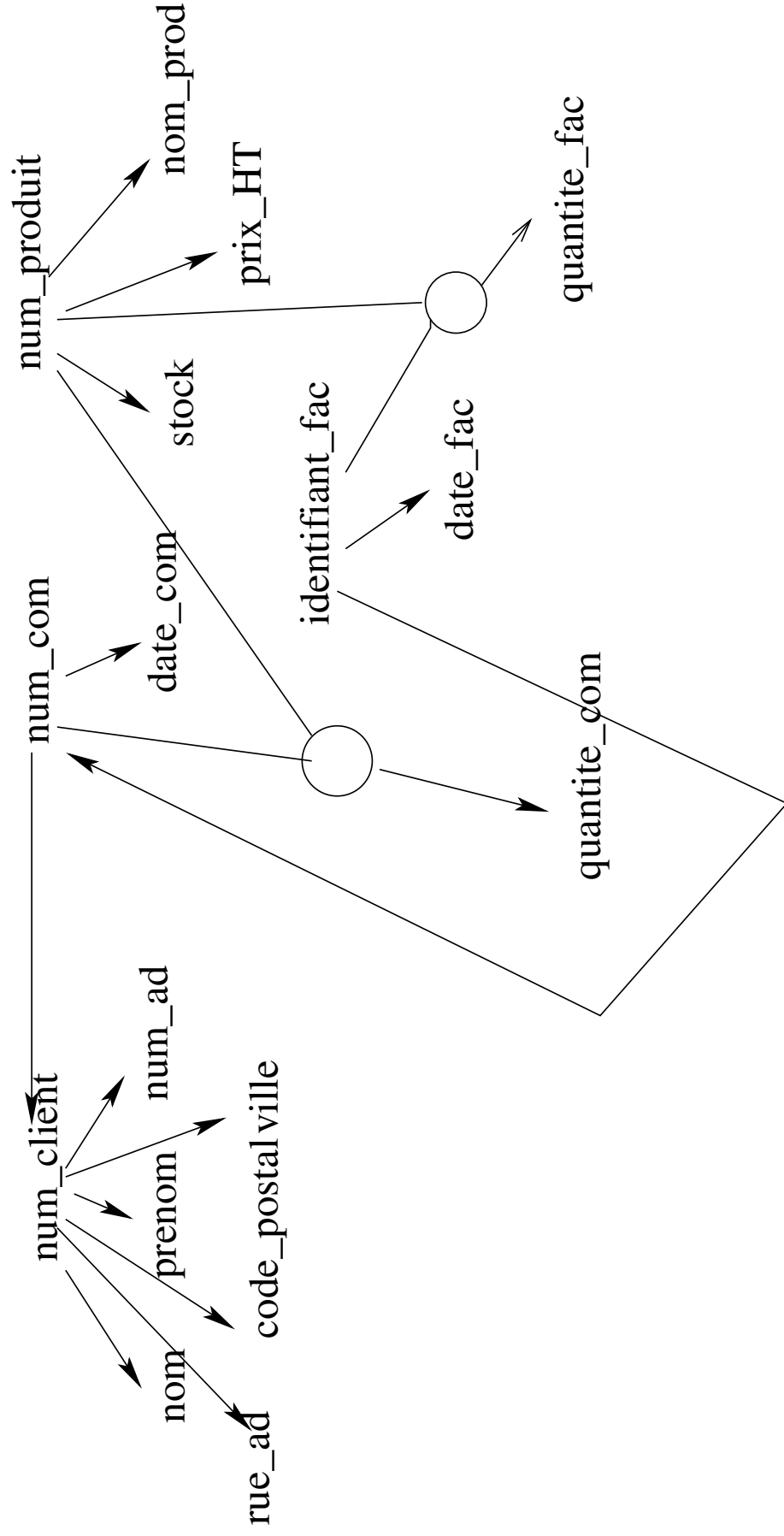
Répartir les propriétés entre les entités et les associations

Les rubriques non-sources sont associées à :

- ▶ une **entité** si elles sont la partie droite d'une dépendance fonctionnelle simple ;
- ▶ une **association** si elles sont la partie droite d'une dépendance fonctionnelle à partie gauche composée.



Hop on applique au graphe des df des «trois belges» »



En résumé : règles à se rentrer dans le crâne

- ▶ Chaque entité possède un identifiant.
- ▶ Toutes les propriétés autres que l'identifiant doivent dépendre fonctionnellement (dépendance élémentaire) de celui-ci (\implies) elles ne peuvent donc prendre qu'une valeur pour une entité – ex. des enfants – et ne **jamais** être nulle).
- ▶ Toutes les propriétés d'une relation doivent dépendre complètement de l'identifiant de la relation (exemple de date et de la chambre d'hôtel) et prend donc une et une seule valeur.
- ▶ Pas de doublon de propriétés, pas de propriété calculée.

En résumé : formellement

- ▶ Toutes les propriétés doivent être élémentaires, c'est-à-dire non décomposable.
- ▶ Chaque objet possède un identifiant et un seul.
- ▶ Les propriétés d'une entité autre que l'identifiant doivent être en dépendance fonctionnelle de cet identifiant.
- ▶ Une propriété ne peut qualifier qu'une seule entité ou qu'une seule relation.
- ▶ Toute dépendance fonctionnelle transitive doit être écartée.
- ▶ Pour chaque occurrence d'une relation, il doit exister une et une seule occurrence par entité participant à la relation.
- ▶ Les propriétés doivent dépendre de la totalité de l'identifiant de la relation. Si ce n'est pas le cas, il faut éclater la relation en autant de relations que nécessaire.

Normalisation

Première forme normale (1FN)

Dans une entité, toutes les propriétés sont élémentaires et il existe au moins une clé caractérisant chaque occurrence de l'objet représenté. Si cette clé est unique elle sera choisie comme identifiant, s'il y en a plusieurs on fait un choix.

- ▶ Plus d'entité client avec juste nom et prénom.
- ▶ Plus de propriété adresse (on fera une exception pour les dates).
- ▶ Dictionnaire élémentaire.
- ▶ Source des dépendances fonctionnelles et association à l'entité.
- ▶ Gagné.

Normalisation (suite)

Deuxième forme normale (2FN)

Toute propriété d'une entité doit dépendre fonctionnellement de l'identifiant par une dépendance fonctionnelle élémentaire.

- ▶ Plus d'entité facture avec le nom du client.
- ▶ Source des dépendances fonctionnelles élémentaires et association à l'entité.
- ▶ Gagné.

Normalisation (suite)

Troisième forme normale (3FN)

Toute propriété d'une entité doit dépendre fonctionnellement de l'identifiant par une dépendance fonctionnelle élémentaire directe.

- ▶ Plus d'entité livre(num_ISBN, id_Style, descriptif_Style).
- ▶ Source des dépendances fonctionnelles élémentaires directe et association à l'entité.
- ▶ Gagné.

Normalisation (suite)

Forme normale de Boyce-Codd (BCNF)

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

- ▶ Dans un restaurant, un menu est composé d'une entrée, d'un plat principal et d'un dessert. Un client ne peut choisir qu'un seul plat par catégorie. On serait tenté de faire une entité `element_repas(id_client, id_categorie, id_plat)` avec `id_client`, `id_categorie` comme identifiant. Mais on a `id_plat` \xrightarrow{df} `id_categorie`, ce n'est donc pas BCNF. La bonne décomposition est `plat(id_plat, id_categorie, descriptif_plat, prix)` en relation avec l'entité client.
- ▶ Là on ne gagne pas, il faut vérifier :'(

Normalisation (le retour du come back)

- ▶ On peut encore faire mieux, on peut imposer qu'une propriété dépende fonctionnellement de l'ensemble des identifiants participant à la relation, mais d'aucun sous-ensemble de cet ensemble.
- ▶ Ce qui est déjà fait lorsqu'on prend des dépendances fonctionnelles élémentaires.

Normalisation (fin)

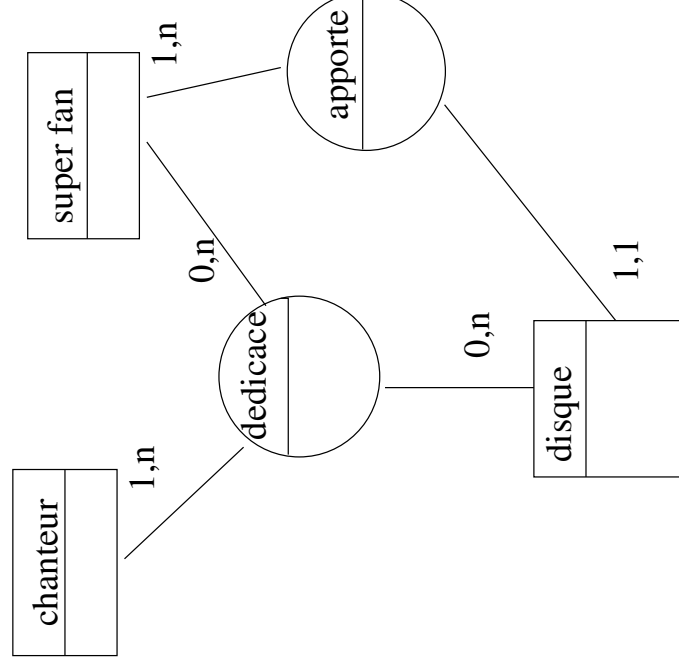
- ▶ Un bon MCD est au moins 3FN.
- ▶ J'attends de vous des BCNF.
- ▶ Il existe des 4FN et 5FN (un tour sur le net vous renseignera).

Décomposition

- ▶ On peut décomposer une relation n-aire en plusieurs relations de dimension plus petite en utilisant les df que l'on peut trouver.
- ▶ Uniquement,
 - ▶ si la cardinalité minimum d'une des entités participant à la relation a pour cardinalité minimale 1 ;
 - ▶ si la df trouvée dépend d'une relation existant dans notre MCD autre que la relation à décomposer, elle doit concerner les mêmes occurrences d'entités que celle de la relation à décomposer.
- ▶ Ce n'est pas une obligation.

Décomposition (exemple)

Une dédicace est organisée pour des chanteurs ou groupes connus. Un fan peut venir faire dédicacer ses disques. Nous imaginerons que le chanteur aura au moins une dédicace à faire (sinon on ne l'aurait pas invité). En revanche à 20 heures tout ferme, un fan peut avoir fait la queue pour rien. On serait tenté de faire le MCD suivant :



Décomposition (exemple – suite)

