

2. Le modèle Entité-association étendu

Version 2 - Septembre 2018

Support du chapitre 16, *Le modèle Entité-association étendu*
de l'ouvrage *Bases de données*, J-L Hainaut, Dunod 2018.

2. Le modèle Entité-association étendu

Contenu

- 2.1 Introduction
- 2.2 Attributs
- 2.3 Types d'associations
- 2.4 Types d'entités
- 2.5 Identifiants
- 2.6 Contraintes d'intégrité
- 2.7 Les classes d'objets génériques
- 2.8 Exemple de schéma EA
- 2.9 Métadonnées
- 2.10 Sémantique interne du modèle EA

2.1 INTRODUCTION

2.1 Introduction

Le modèle *Entité-association de base* offre un jeu de concepts réduit mais suffisant pour modéliser des domaines d'applications simples et d'ampleur limitée :

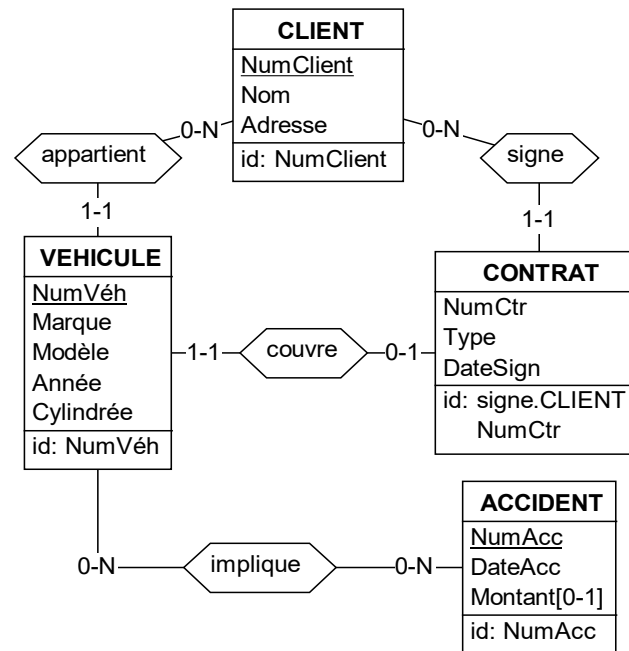
- type d'**entités**
- **attribut** monovalué et atomique (**une** valeur simple par entité)
- **type d'associations** binaire (2 rôles)
- **identifiant** d'un type d'entités.

[relire le chapitre 11](#)

Le modèle Entité-association étendu permet de modéliser de manière plus précise, plus expressive, des domaines d'application complexes et de grande taille.

2.1 Introduction - Rappel

Que pouvons-nous considérer comme acquis ?



2.1 Introduction - Sémantiques

On parle beaucoup de sémantique. Qu'entend-on par là ?

1. Sémantique externe d'un schéma EA

2. Sémantique interne du modèle EA

2.1 Introduction - Sémantiques

Sémantique externe d'un schéma EA

= **correspondance** entre chaque **construction du schéma** et les objets/**concepts du domaine d'application** qu'il représente; permet d'interpréter le schéma en termes accessibles aux utilisateurs; pas nécessairement formelle mais doit être précise et complète.

Exemples :

- Si **cli** est un CLIENT, alors **cli** représente *toute personne physique ou morale qui a passé au moins une commande honorée depuis moins de 5 ans...*
- Si **com** est une COMMANDE, alors **com.DATE** est *la date à laquelle **com** a été validée et enregistrée...*
- Si **com** est une COMMANDE et **cli** un CLIENT, et si (**com**, **cli**) est une association passe, alors **cli** est *réputé avoir passé la commande **com** et est responsable du paiement de l'expédition éventuelle...*

La sémantique externe intéresse le concepteur, le développeur et l'utilisateur.

2.1 Introduction - Sémantiques

Sémantique interne du modèle EA

= interprétation formelle (mathématique) des concepts du modèle; permet d'attribuer une interprétation unique aux concepts du modèle et aux processus de manipulation d'un schéma et de ses instances : langages, transformations, normalisation, etc.

Exemples :

- un type d'entités E est représenté par un ensemble d'entités de nom E ;
- un type d'associations R défini sur les types d'entités E_1, E_2, \dots, E_n est représenté par une relation de nom R définie sur les ensembles E_1, E_2, \dots, E_n .

La sémantique interne intéresse le méthodologue, le chercheur mais aussi le concepteur et le développeur désireux de comprendre leur pratique.

2.1 Introduction - Modèle, schéma, population

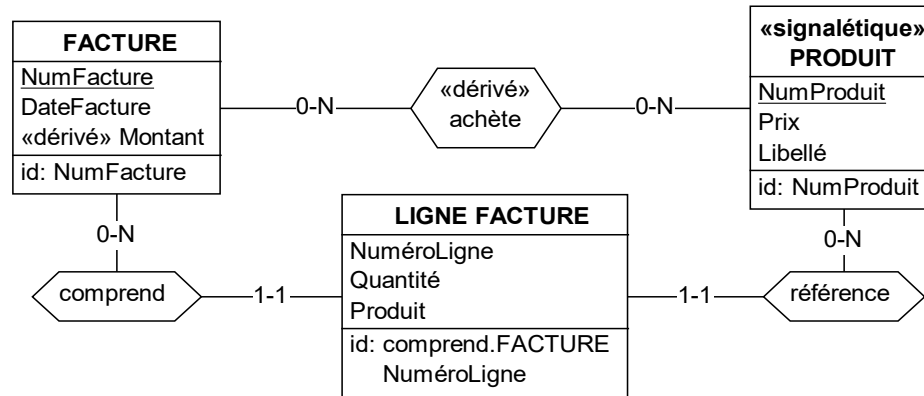
Modèle : système formel général destiné à représenter certains aspects de tous les domaines d'application réels et possibles d'un certain type. *Exemples* : le modèle relationnel, le modèle EA

Schéma : représente les concepts statiques d'un domaine d'application particulier. *Exemples* : la compagnie d'assurance ASSURTOU ou la bibliothèque municipale. S'exprime selon un modèle.

Population : ensemble des instances d'un objet d'un schéma à un instant déterminé. *Exemples* : les entités CLIENT, les lignes de la table COMMANDE.

2.1 Introduction - Stéréotypes

Stéréotype : attribut multivalué facultatif affecté aux objets d'un modèle. Prend des valeurs déterminées pour certains objets d'un schéma. Permet de compenser certains faiblesses d'expression d'un modèle. Emprunté à UML.



2.1 Introduction - Schéma de base de données

Schéma

- Description des structures statiques d'une partie d'un domaine d'application
- Il peut exister plusieurs schémas relatifs à ce même domaine, représentant chacun un aspect ou une partie spécifique à un moment déterminé.
- Un schéma est positionné à un certain niveau d'abstraction : conceptuel, logique, physique.
- Un schéma est conforme à un modèle : EA, UML, SQL2, SQL3, XML, etc.

2.2 ATTRIBUTS

Contenu

- a) **Attributs composés**
- b) **Attributs multivalués**
- c) **Attributs complexes**
- d) **Domaine et type d'un attribut**
- e) **Attributs objet**

2.2 Attributs - Composés, multivalués, complexes

Au delà des types des attributs simples du modèle de base :

- valeurs composées de (sous-) valeurs
- valeurs ensemblistes - plusieurs valeurs par attribut
- valeurs complexes - composées et multivaluées
- domaines de valeurs utilisateur, attributs objet

2.2 Attributs - Composés, multivalués, complexes

Attribut composé : composé d'attributs plus simples; composé = pas atomique

Attribut multivalué : dont chaque valeur est constituée d'un ensemble d'éléments; multivalué = pas monovalué

Attribut complexe : multivalué et composé

EMPRUNTEUR
<u>NumPers</u>
Adresse
Rue
Ville
Téléphone[1-5]
id: NumPers

DOCUMENT
<u>ID-Doc</u>
MotClé[0-10]
id: ID-Doc

CLIENT
<u>NCli</u>
Contact[1-5]
Statut
Adresse
Rue
Ville
Téléphone[0-5]
CodePays[0-1]
CodeZone
NuméroLocal
id: NCli

Remarque : la contrainte de cardinalité indique si l'attribut est mono/multivalué, obligatoire/facultatif.

2.2 Attributs - Domaines et types

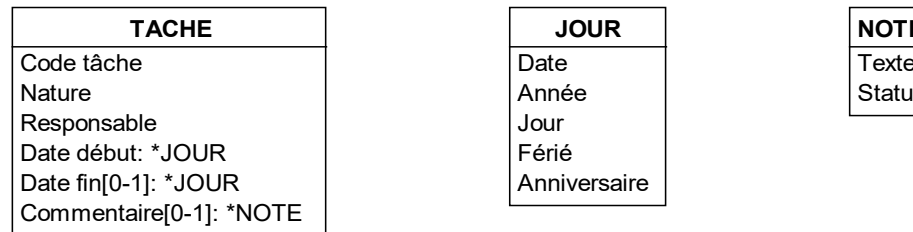
Un attribut est défini sur un *domaine de valeurs* : type ou domaine défini par l'utilisateur (cfr SQL2 et SQL3)

CLIENT
Code client: matricule
Nom: char(32)
Adresse expédition: adresse
Adresse facturation: adresse
Contact[0-3]: téléphone
Compte
Identité: numéro_de_compte
Niveau: montant
Fiabilité: boolean

Types utilisateurs
matricule
code_région: char(2)
numéro_série: num(6)
adresse
numéro: char(8)
rue: char(26)
ville: char(32)
téléphone_principal: téléphone
téléphone
code_pays: num(3)[0-1]
code_zone: num(4)
code_local: num(8)
numéro_de_compte
code_banque: num(4)
code_compte: num(10)
check_sum: num(2)
montant: num(10,2)

2.2 Attributs - Attributs objet

Un attribut est défini sur un domaine qui est un **type d'entités**. On rencontre ce concept dans les *modèles orientés objet*.



Ne pas abuser ! Ne remplacent pas les types d'associations (attribut objet = la moitié d'un type d'associations). A utiliser là où les TA seraient trop encombrants ...

2.3 TYPES D'ASSOCIATIONS

Contenu

- a) TA n-aires
- b) Rôles multitypes
- c) TA de TA
- d) Attributs de TA
- e) Contraintes de cardinalité généralisées

2.3 Types d'associations

Au delà des types d'associations *binaires simples* du modèle de base

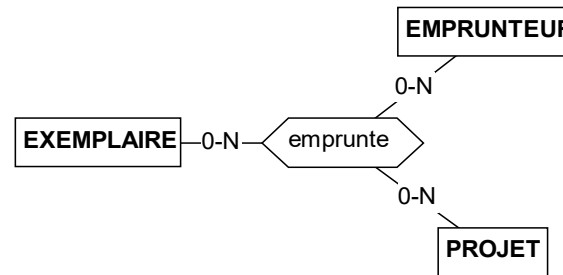
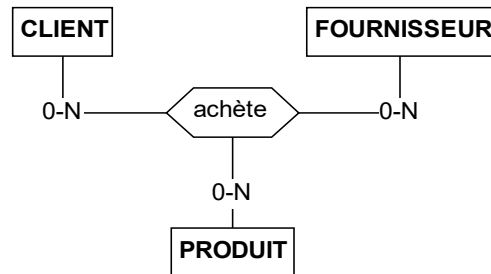
- plus de 2 rôles : n-aires
- rôle joué par plus d'un type d'entités : rôle multitype
- rôle joué par un TA
- TA avec attributs
- contraintes de cardinalité de rôle plus étendues

TA binaire simple :

- binaire
- 1:1 ou 1:N
- sans attribut

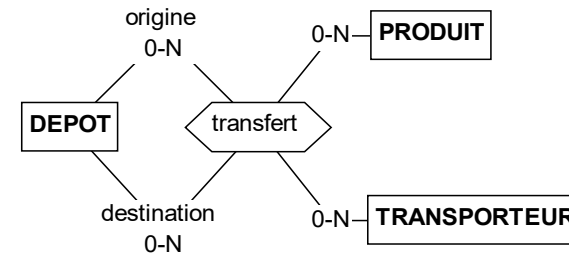
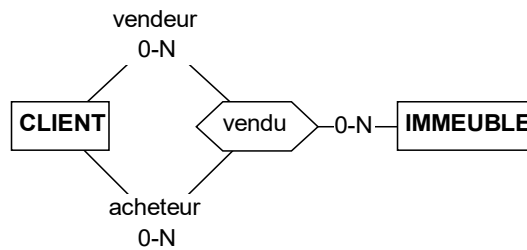
2.3 Types d'associations n-aire

On admet qu'un type d'associations possède plus de 2 rôles



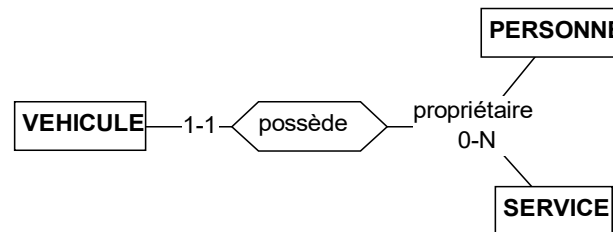
2.3 Types d'associations n-aires cycliques

On admet que plusieurs rôles soient définis sur le même type d'entités



2.3 Types d'associations - Rôle multitype

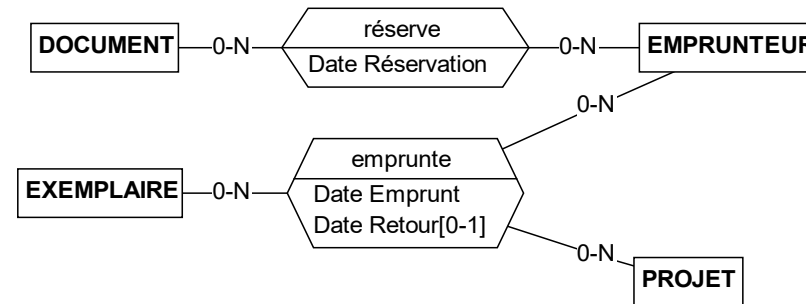
On admet qu'un rôle soit défini sur l'union de plusieurs types d'entités



attention : possède est binaire !

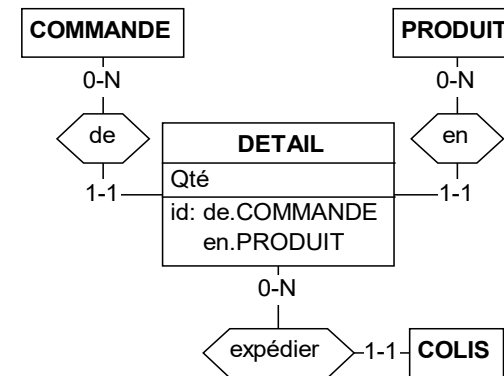
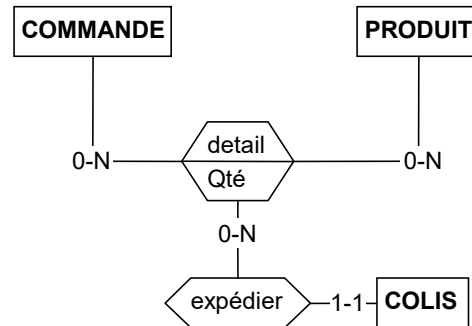
2.3 Types d'associations - Attributs

On admet qu'un TA possède des attributs



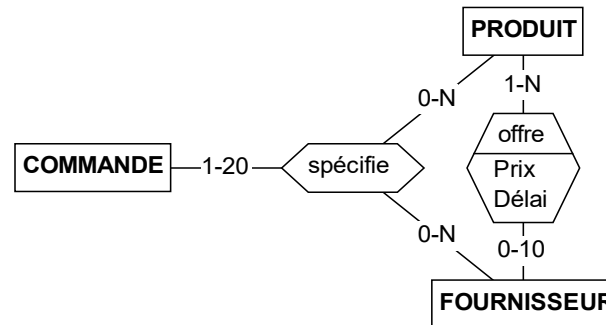
2.3 Types d'associations - De TA

On (n')admet (pas) qu'un rôle soit défini sur un type d'associations



2.3 Types d'associations - Cardinalité

On admet des contraintes de cardinalité de rôle plus étendues
En particulier, la contrainte **[1-N]** est enfin admise !



2.4 TYPES D'ENTITES

2.4 Types d'entités

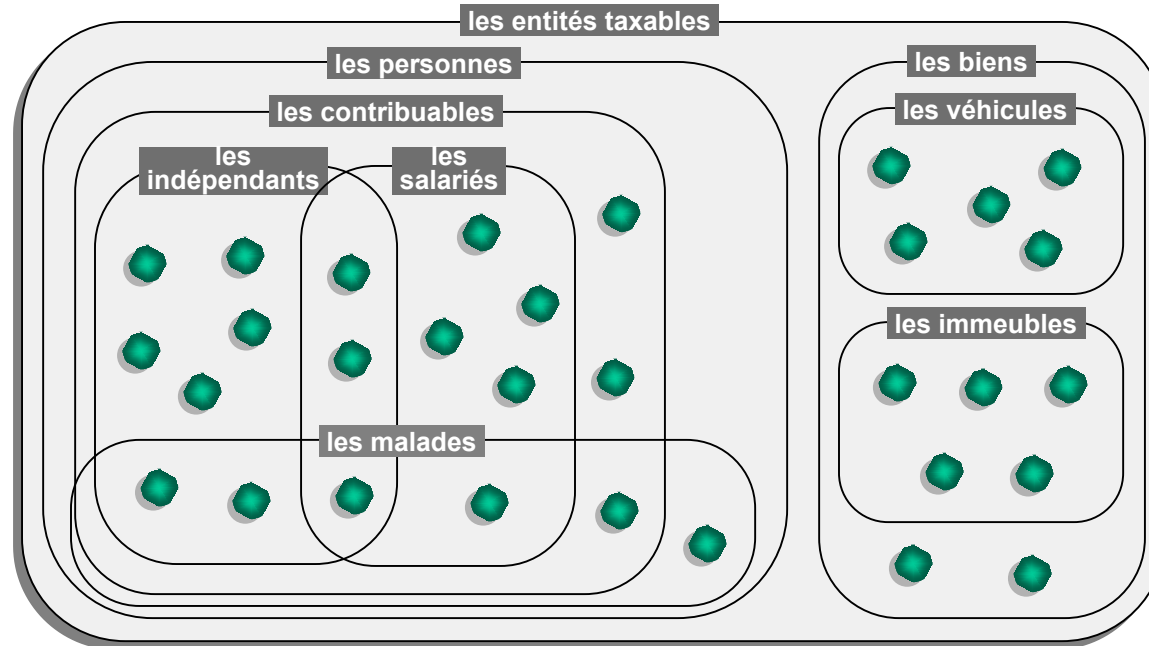
Changement important d'hypothèse

Modèle EA de base : une entité appartient à un et un seul type

Modèle EA étendu : une entité appartient à au moins un type

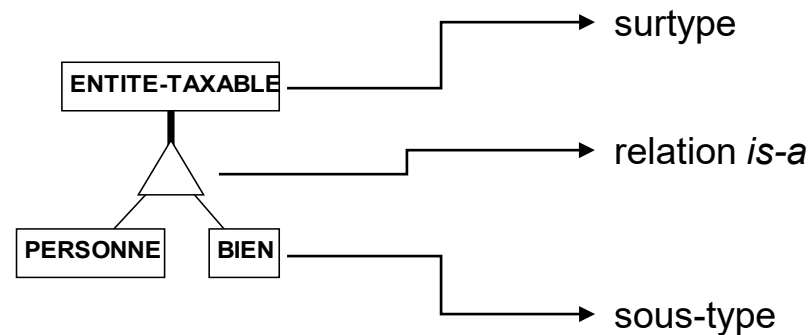
2.4 Types d'entités

Une entité peut appartenir à plus d'un type



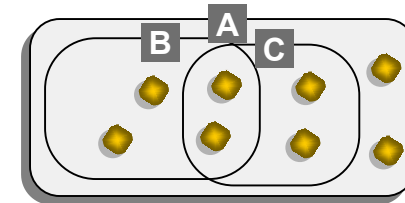
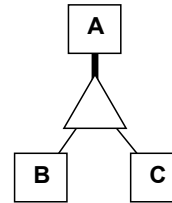
2.4 Types d'entités - Relation *is-a*

la population de PERSONNE est une partie de celle de ENTITE-TAXABLE
OU
toute entité PERSONNE est une (is a) entité ENTITE-TAXABLE

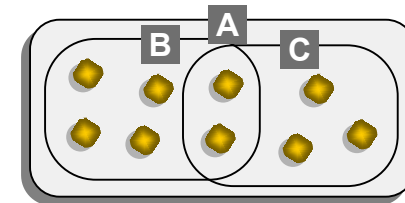
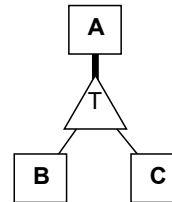


2.4 Types d'entités - Types de distribution

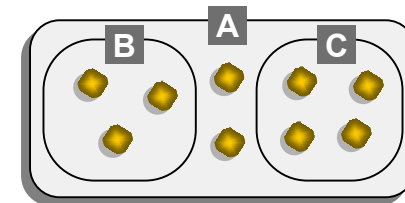
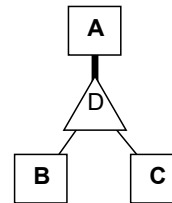
**recouvrement
 couverture partielle**



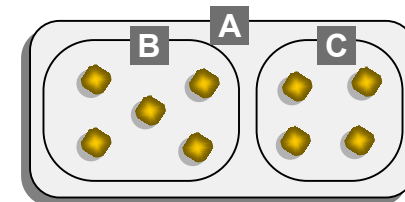
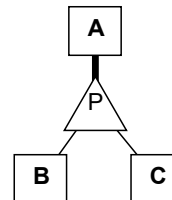
**recouvrement
 couverture totale**



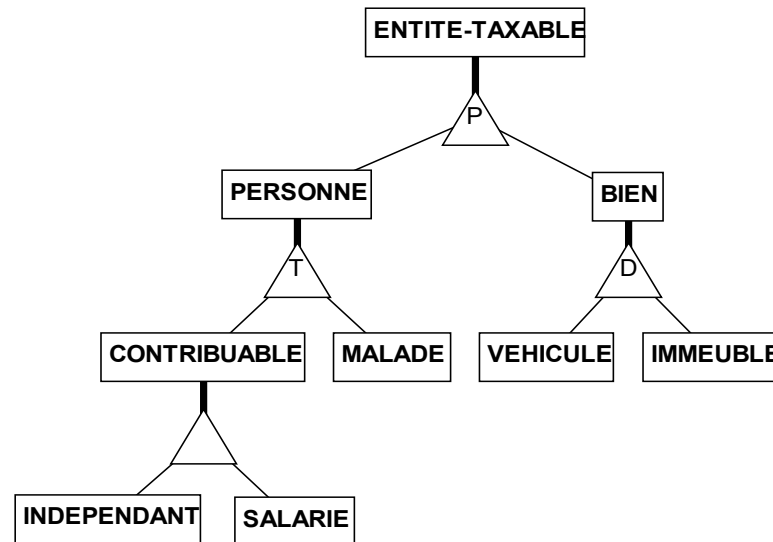
**disjonction
 couverture partielle**



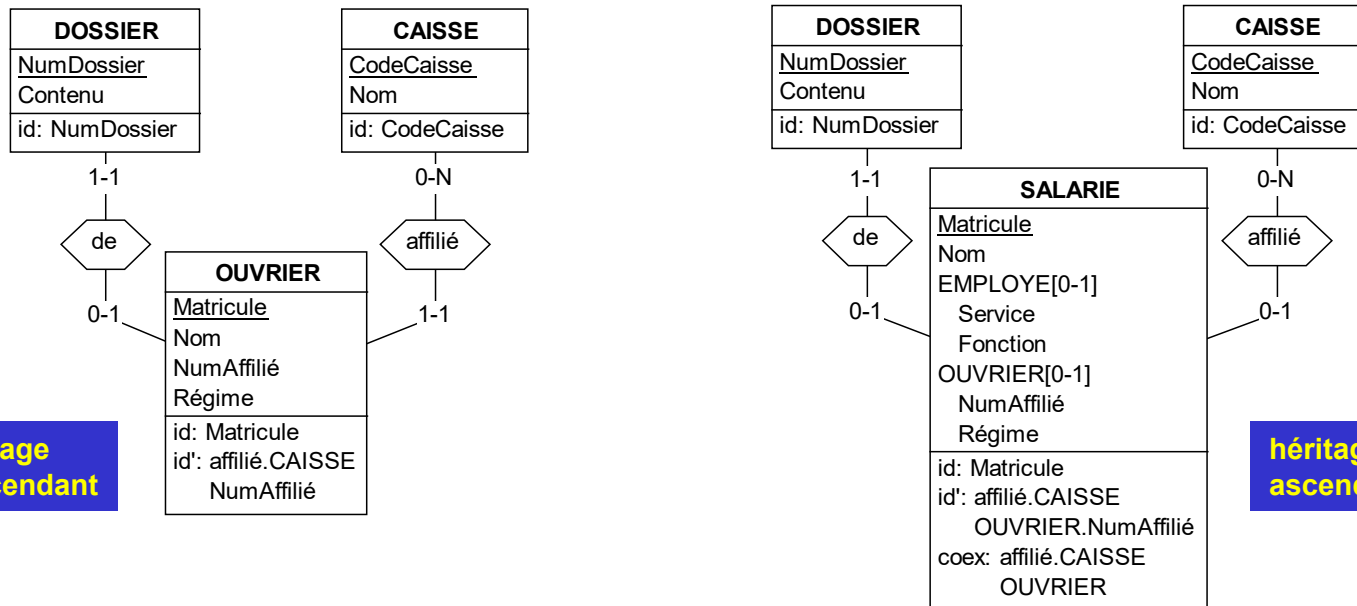
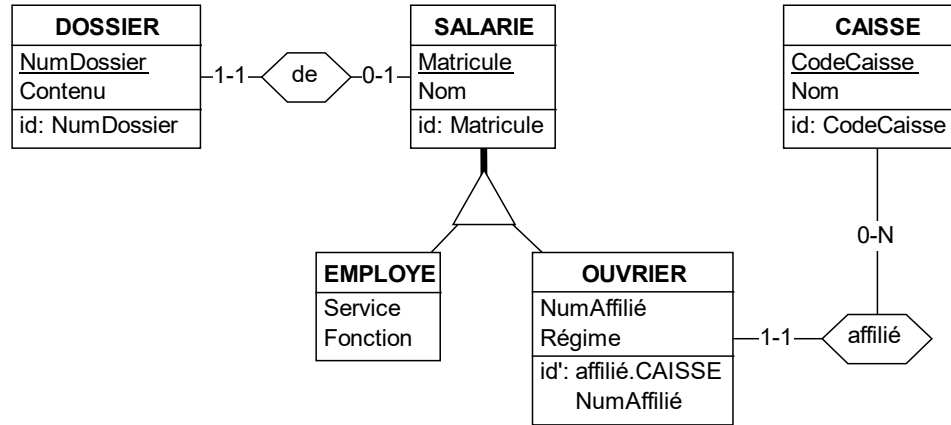
**disjonction
 couverture totale
 =
 partition**



2.4 Types d'entités - Types de distribution



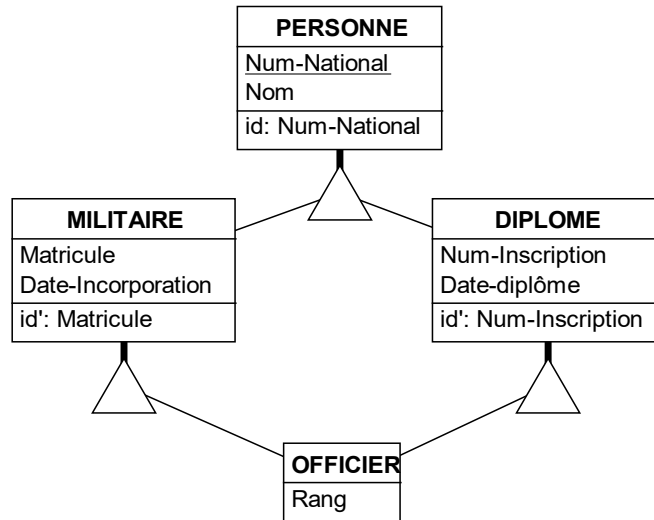
2.4 Types d'entités - Héritage



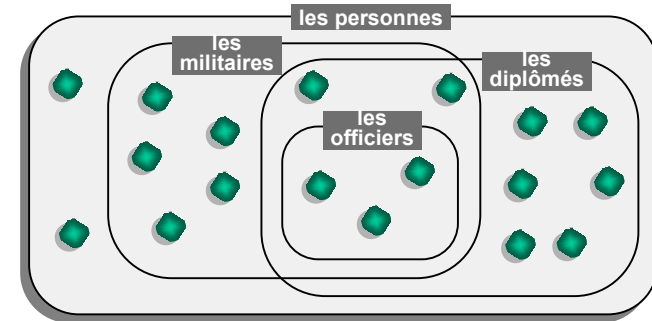
2.4 Types d'entités - Héritage

**L'héritage est une simple conséquence de la propriété
d'inclusion des populations
(contrairement aux langages OO)**

2.4 Types d'entités - Surtypes et héritage multiples



un officier est un militaire et un diplômé

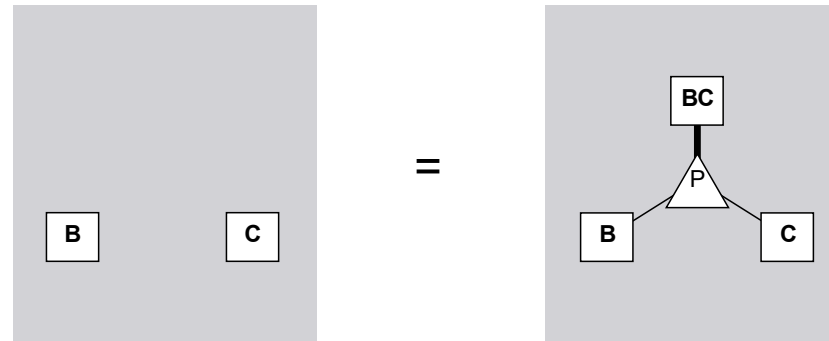


**il existe des militaires diplômés
qui ne sont pas des officiers**

2.4 Types d'entités - Hypothèse

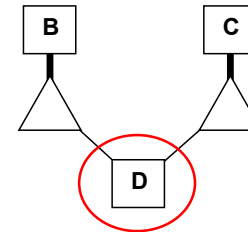
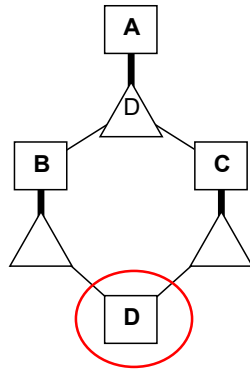
Deux types d'entités sans surtype commun ont-ils obligatoirement des populations disjointes ?

On choisit **arbitrairement** de répondre **OUI**



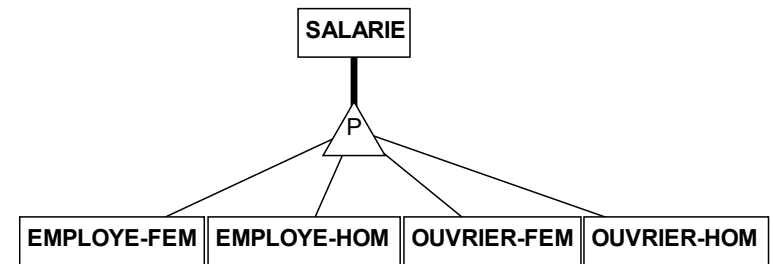
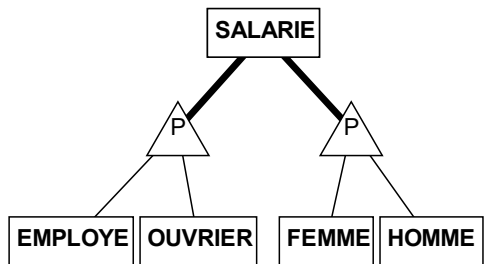
2.4 Types d'entités - Problème de cohérence

Il est possible de dessiner des **schémas impossibles** :
le type d'entités **D** est vide par construction !
On dit que **D** est **non satisfiable**



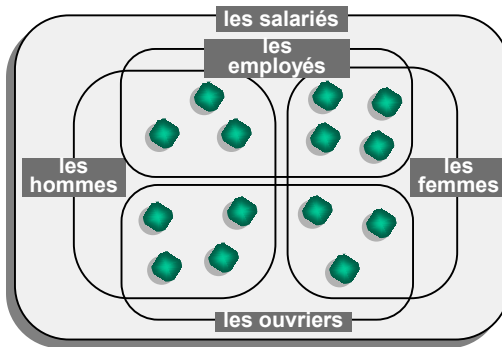
2.4 Types d'entités - Répartitions multiples

Il peut être utile de répartir la population du surtype selon plus d'un critère



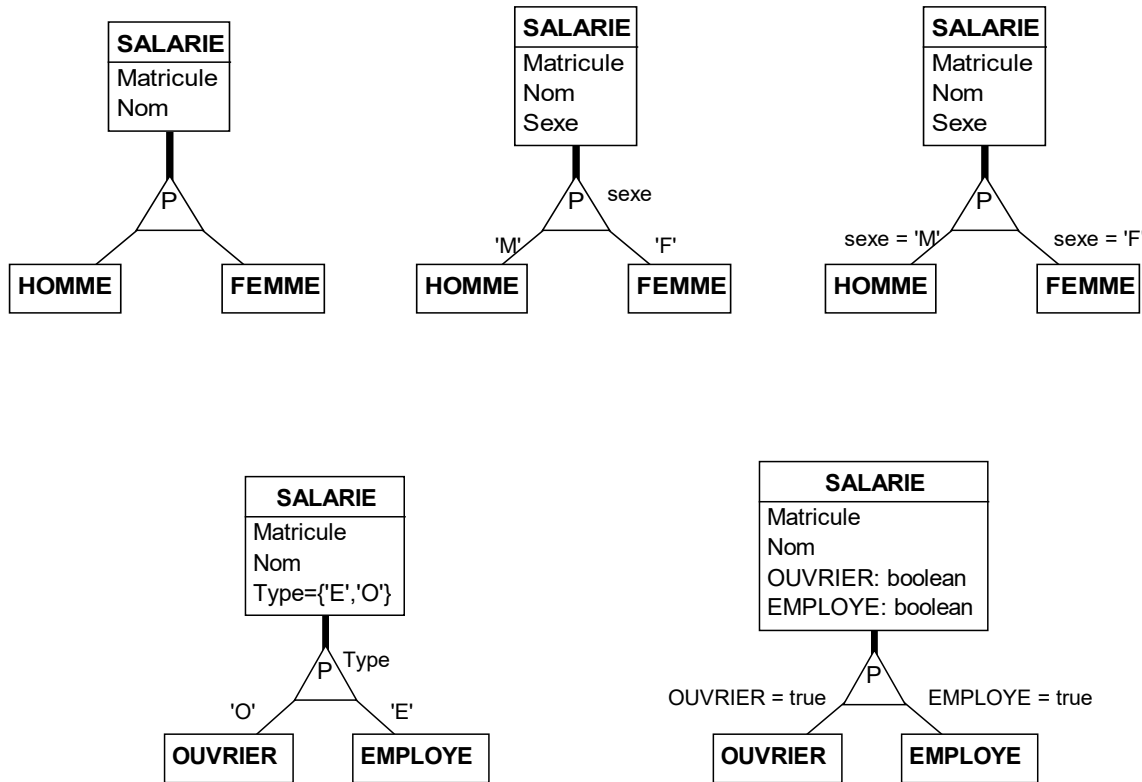
lourd et peu lisible

*autres techniques de transformation
à l'annexe E*



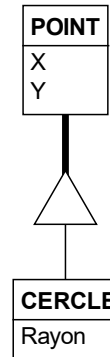
2.4 Types d'entités - Critères de répartition

Répartition en sous-types : intrinsèque ou conditionnelle

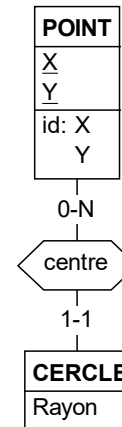


avec indicateur de sous-types

2.4 Types d'entités - Mauvais usage des relations *is-a*



NON



OUI

2.5 IDENTIFIANTS

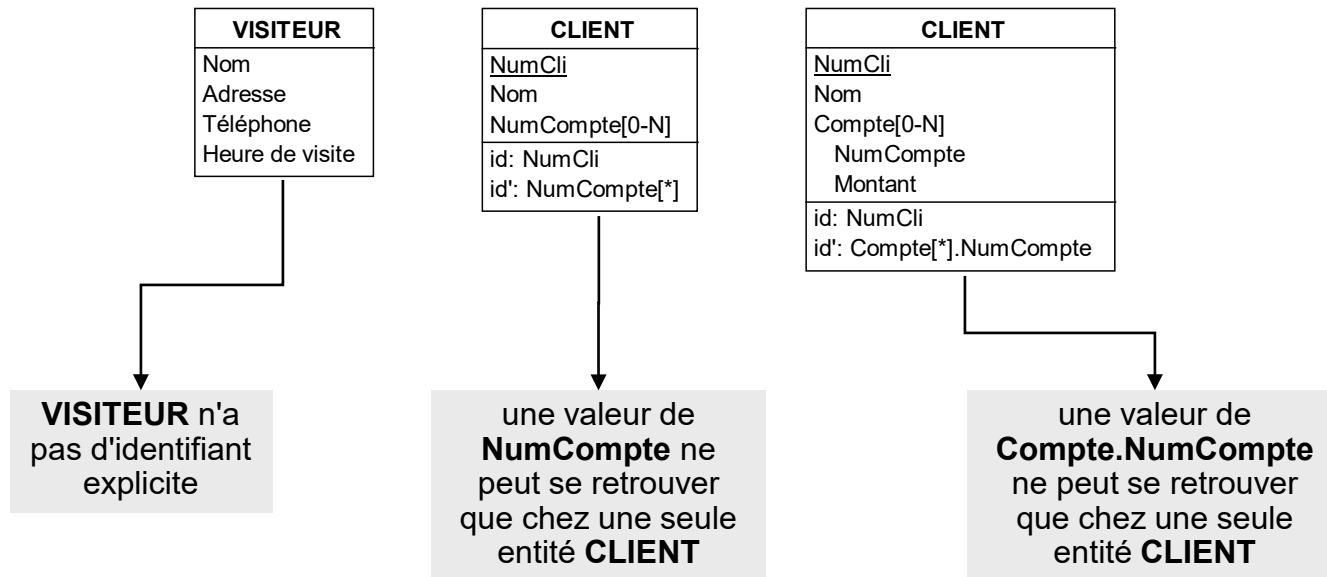
Contenu

- a) Identifiants d'un type d'entités
- b) Identifiants d'un type d'associations
- c) Identifiants d'un attribut

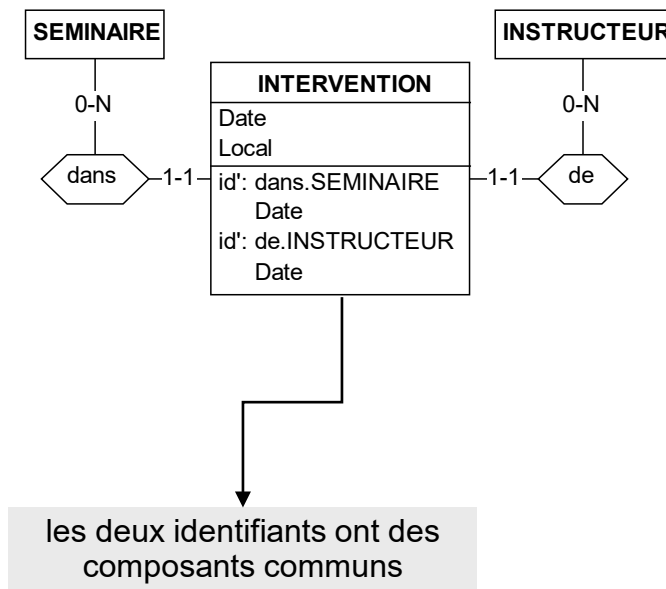
2.5 Identifiants

Les structures de base (TE, TA, attributs) étant plus riches, les possibilités de définition d'identifiants sont plus variées.

2.5 Identifiants - Type d'entités



2.5 Identifiants - Type d'entités, identifiants non disjoints



2.5 Identifiants - Type d'entités, identifiants cycliques

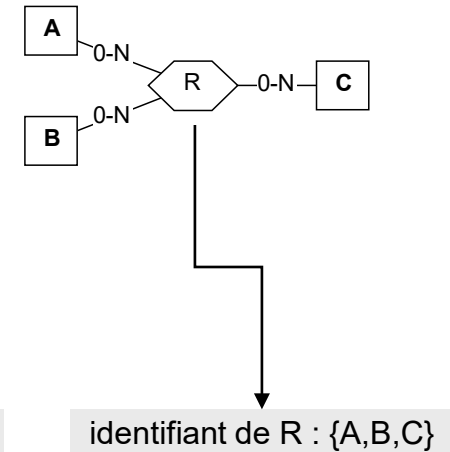
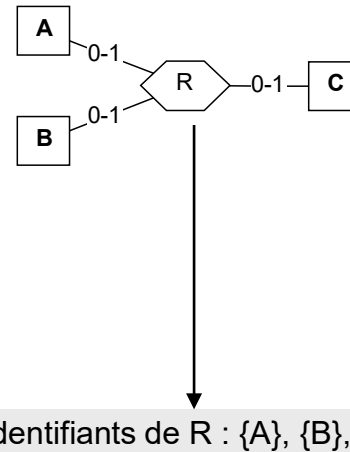
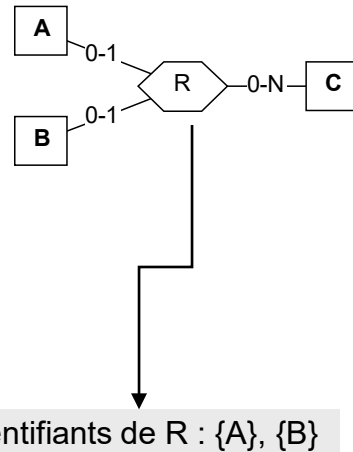
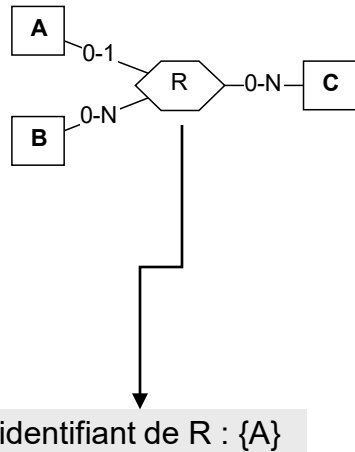


Bonne idée
 mais mauvaise expression

« diviser pour régner »

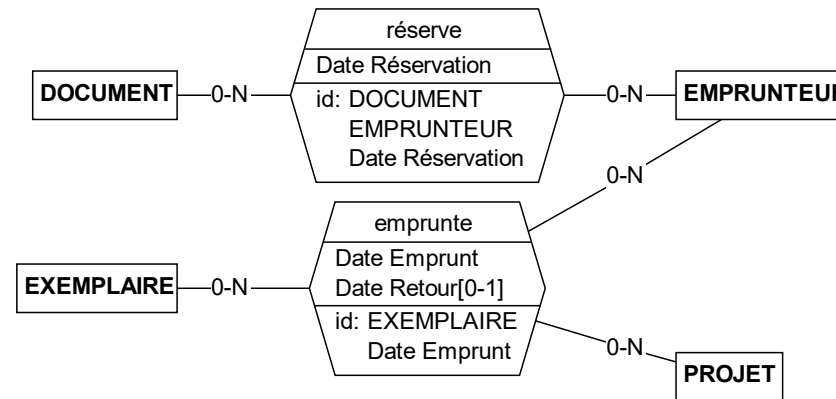
2.5 Identifiants - Type d'associations, identifiants implicites

Implicite = non déclaré explicitement mais déductible des propriétés du TA



- Tout rôle [0-1] ou [1-1] est un identifiant de R
- Si pas de rôle [0-1] ou [1-1] et pas d'identifiant explicite, l'ensemble des rôles est un identifiant minimum de R

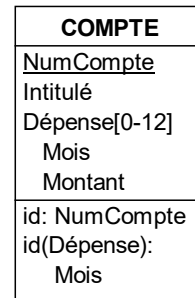
2.5 Identifiants - Type d'associations, identifiants explicites



2.5 Identifiants - Attribut complexe

Un attribut multivalué a des valeurs distinctes pour chaque parent.
Une partie de ces valeurs constitue-t-elle un identifiant ?

parent : TE, TA ou attribut composé
auquel l'attribut est directement attaché



Pour chaque parent (entité **COMPTE**), les valeurs de **Dépense** ont des (sous-)valeurs de **Mois** distinctes.

2.6 CONTRAINTE D'INTEGRITE

Contenu

- a) **Contraintes statiques et dynamiques**
- b) **Contraintes standard**
- c) **Contraintes additionnelles**

2.6 Contraintes d'intégrité - Généralités

Contrainte d'intégrité : propriété associée à un schéma et que les instances (populations) de ses objets et leur évolution doivent respecter, à défaut d'être déclarées incorrectes.

Valable pour tout type de schéma : conceptuel, logique, physique.

Contraintes intrinsèques : intégrées au modèle

Contraintes additionnelles : variété illimitée, réclament un langage d'expression générique

Contraintes statiques : définissent les instances valides

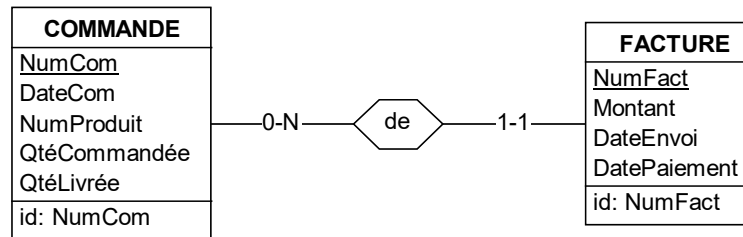
Contraintes dynamiques : définissent les modifications valides des instances

2.6 Contraintes d'intégrité - Standard : domaine, valeurs

Sexe = {'M', 'm', 'F', 'f'}

DateNaissance = [1/1/1920 - 31/12/2019]

contrainte de domaine



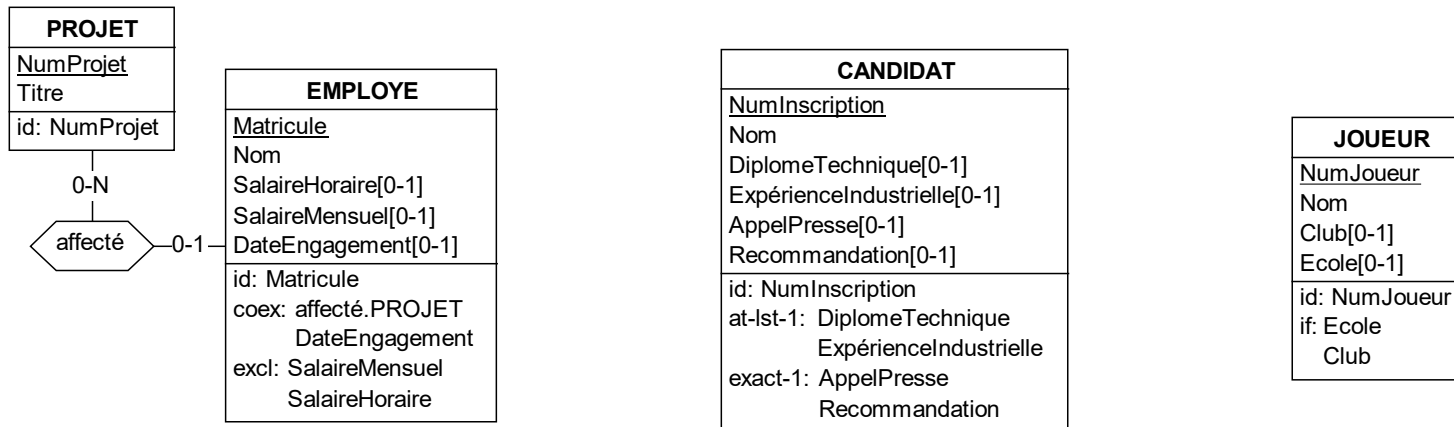
contrainte inter-valeurs

$QtéLivrée \leq QtéCommandée$

$DateEnvoi \leq DatePaiement$

2.6 Contraintes d'intégrité - Standard : existence

Une contrainte d'existence régit la **présence/absence** de valeurs pour certains attributs/rôles



Contrainte de *coexistence* : $(\text{existe}(\text{affecté.PROJET}) \text{ et } \text{existe}(\text{DateEngagement}))$
 ou $(\text{non existe}(\text{affecté.PROJET}) \text{ et } \text{non existe}(\text{DateEngagement}))$

Contrainte d'*exclusion* : $\text{non existe}(\text{SalaireMensuel}) \text{ ou } \text{non existe}(\text{SalaireHoraire})$

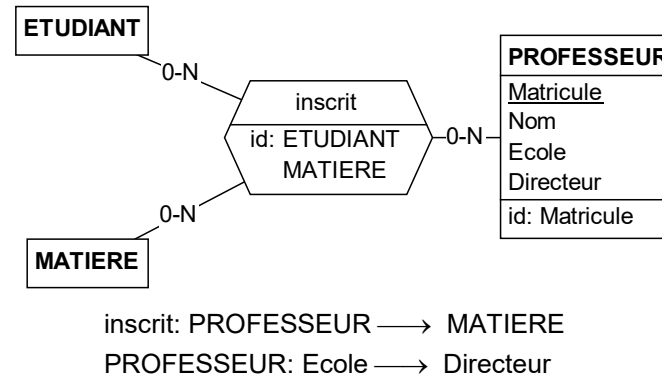
Contrainte *au moins un* : $\text{existe}(\text{DiplomeTechnique}) \text{ ou } \text{existe}(\text{ExpérienceIndustrielle})$

Contrainte *exactement un* : $\text{existe}(\text{AppelPresse}) \text{ ou-exclusif } \text{existe}(\text{Recommandation})$

Contrainte d'*implication* : $\text{si existe}(\text{Ecole}) \text{ alors existe}(\text{Club})$

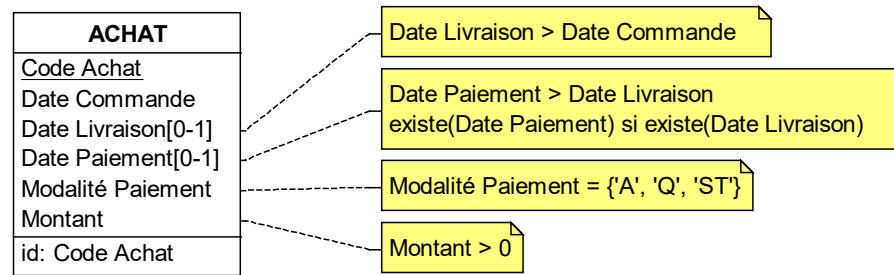
2.6 Contraintes d'intégrité - Standard : dépendance fonctionnelle

Les composants d'un TE ou d'un TA peuvent être soumis à des dépendances fonctionnelles. A corriger ultérieurement si nécessaire !



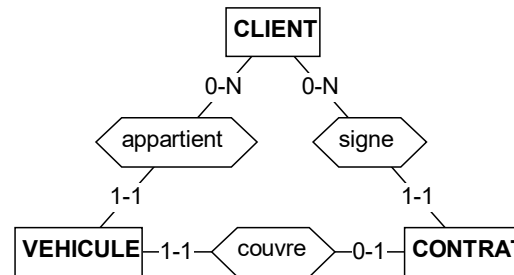
2.6 Contraintes d'intégrité - Additionnelles

Contraintes pour lesquelles il n'existe pas de syntaxe d'expression spécifique.
Nécessitent un langage général, formel ou informel



2.6 Contraintes d'intégrité - Additionnelles : cycliques

Tout cycle dans un schéma est susceptible de faire l'objet d'une contrainte.

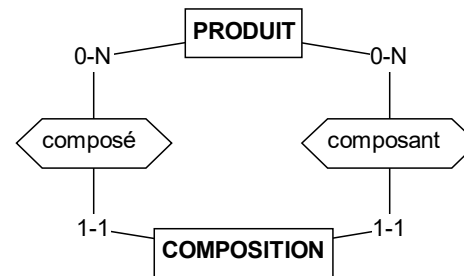


Le client auquel un véhicule appartient est-il le signataire du contrat qui couvre ce véhicule ?
si oui :

$\forall v \in \text{VEHICULE}, v.\text{appartient.CLIENT} = v.\text{couvre.CONTRAT.signe.CLIENT}$
 $\text{VEHICULE.appartient.CLIENT} = \text{VEHICULE.couvre.CONTRAT.signe.CLIENT}$

2.6 Contraintes d'intégrité - Additionnelles : cycliques

Tout cycle dans un schéma est susceptible de faire l'objet d'une contrainte.

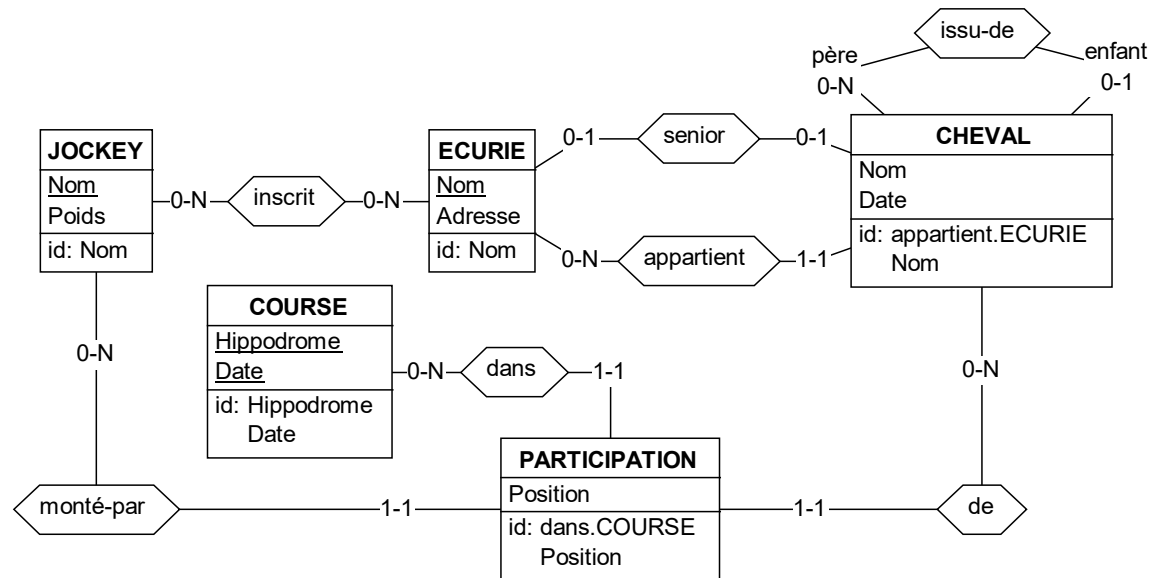


Un produit peut-il être un composant de lui-même ?
si non :

$\forall p \in \text{PRODUIT}, p \notin p.\text{composé}.\text{COMPOSITION}.\text{composant}.\text{PRODUIT}$

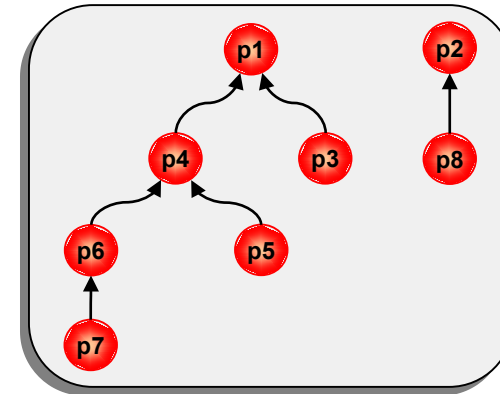
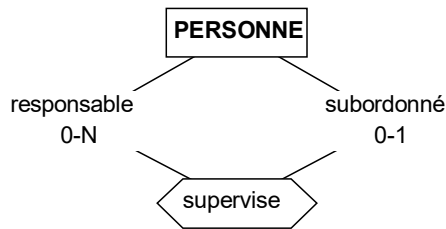
2.6 Contraintes d'intégrité - Additionnelles : cycliques

Tout cycle dans un schéma est susceptible de faire l'objet d'une contrainte.



2.6 Contraintes d'intégrité - Propriété des graphes d'instances

Un cycle dans un schéma ne signifie pas que les instances peuvent présenter des cycles

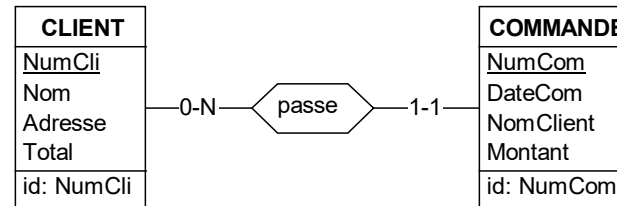


Une personne peut-elle être responsable d'elle-même, directement ou indirectement ?
si non :

contrainte plus difficile à exprimer !

2.6 Contraintes d'intégrité - Redondance

En principe, il n'y a pas de structures redondantes dans un schéma conceptuel, mais ...



$$\forall cli \in \text{CLIENT}, cli.Total = \sum_{com \in cli.passe.COMMANDE} com.Montant$$

$$\forall com \in \text{COMMANDE}, com.NomClient = com.passe.CLIENT.Nom$$

2.7 LES CLASSES D'OBJETS GENERIQUES

Contenu

- a) Types d'entités
- b) Types d'associations

2.7 Les classes d'objets génériques

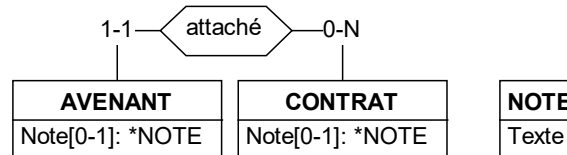
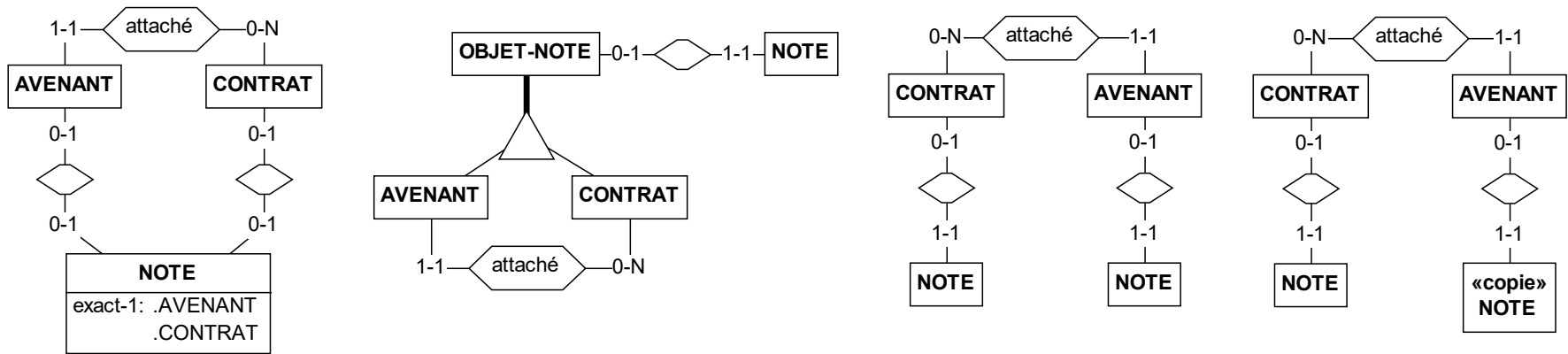
Certains patterns sont à ce point fréquents dans les schémas conceptuels qu'ils ont été intégrés aux modèles (*built-in*):

- relations *is-a*
- composition/agrégation (UML)

Y en a-t-il d'autres ?

2.7 Les classes d'objets génériques - Types d'entités

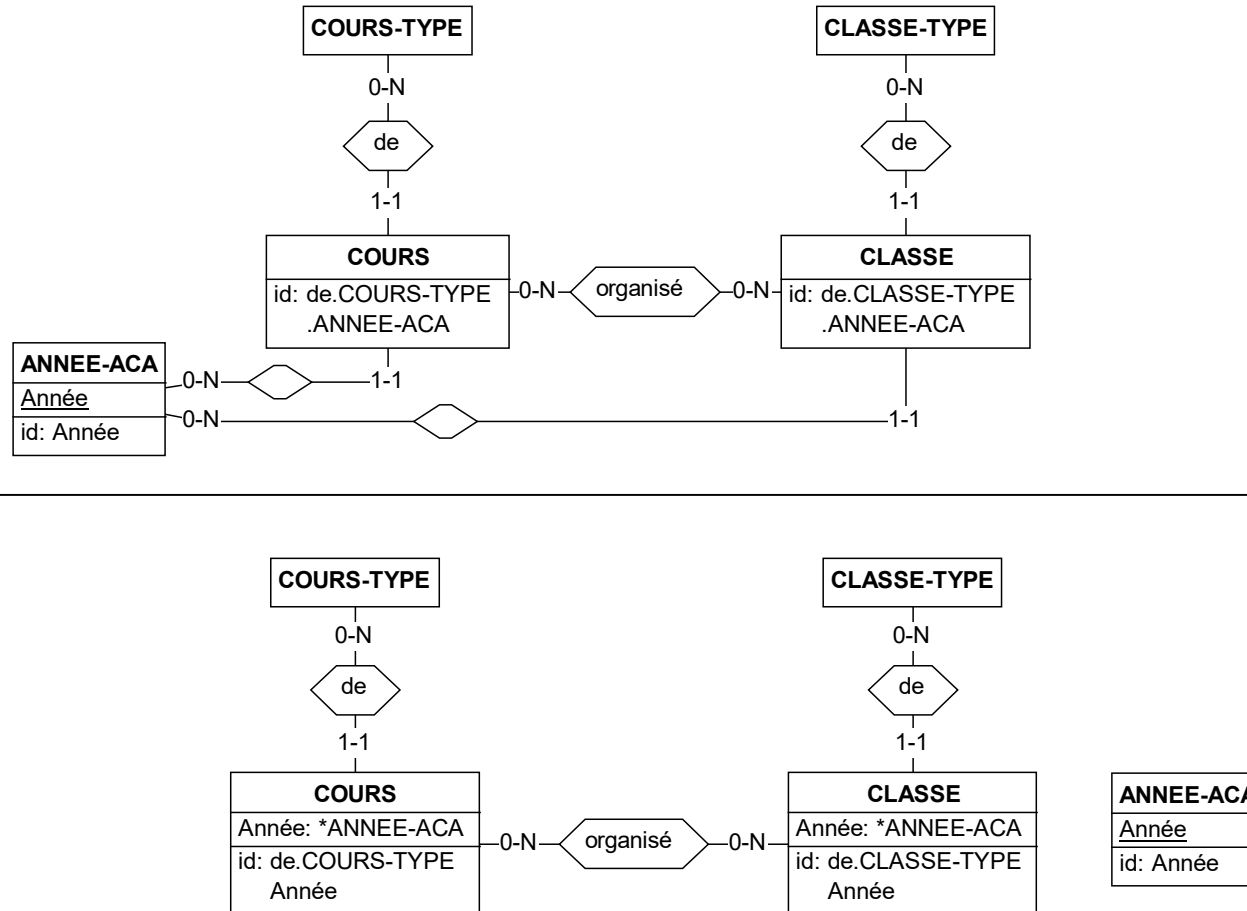
Les types d'entités *utilitaires*



le plus sobre !

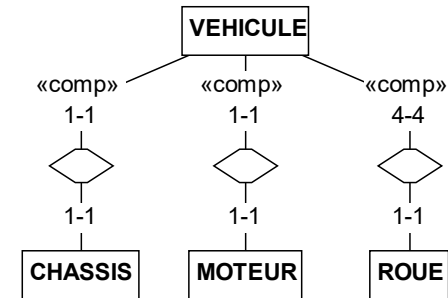
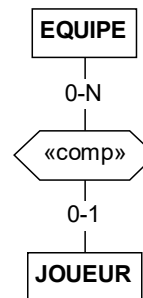
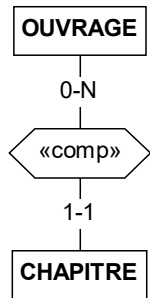
2.7 Les classes d'objets génériques - Types d'entités

Les types d'entités *dimension*



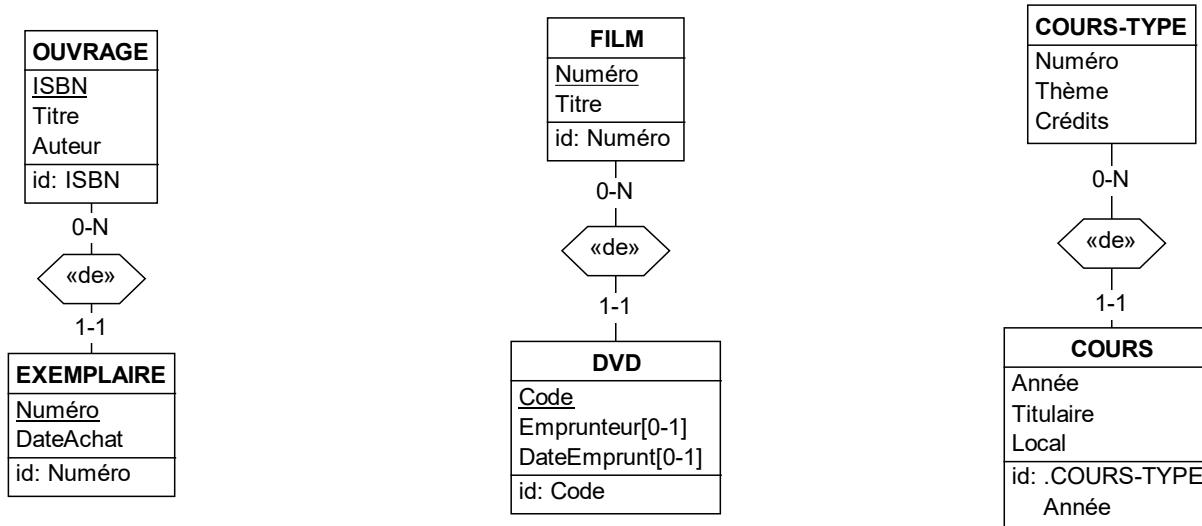
2.7 Les classes d'objets génériques - Types d'associations

Les types d'associations *de composition (whole/part)*



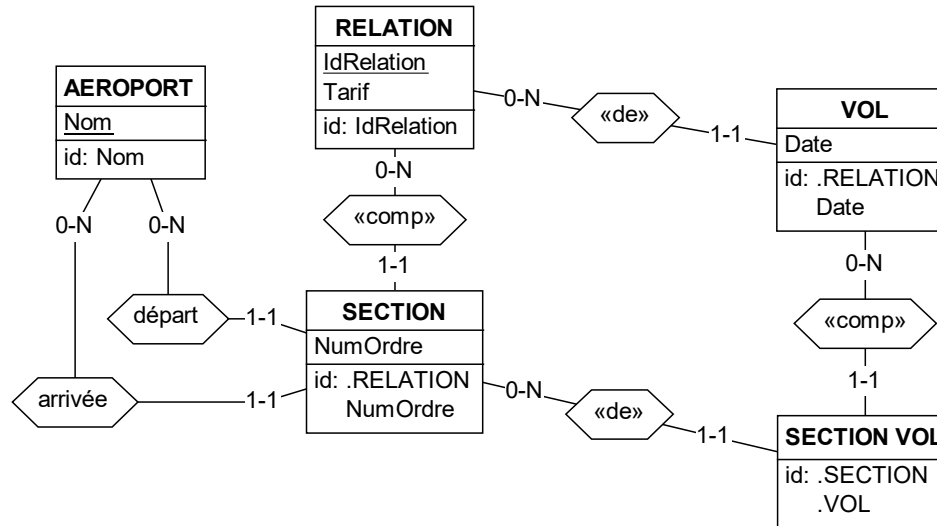
2.7 Les classes d'objets génériques - Types d'associations

Les types d'associations *de matérialisation*



2.7 Les classes d'objets génériques - Types d'associations

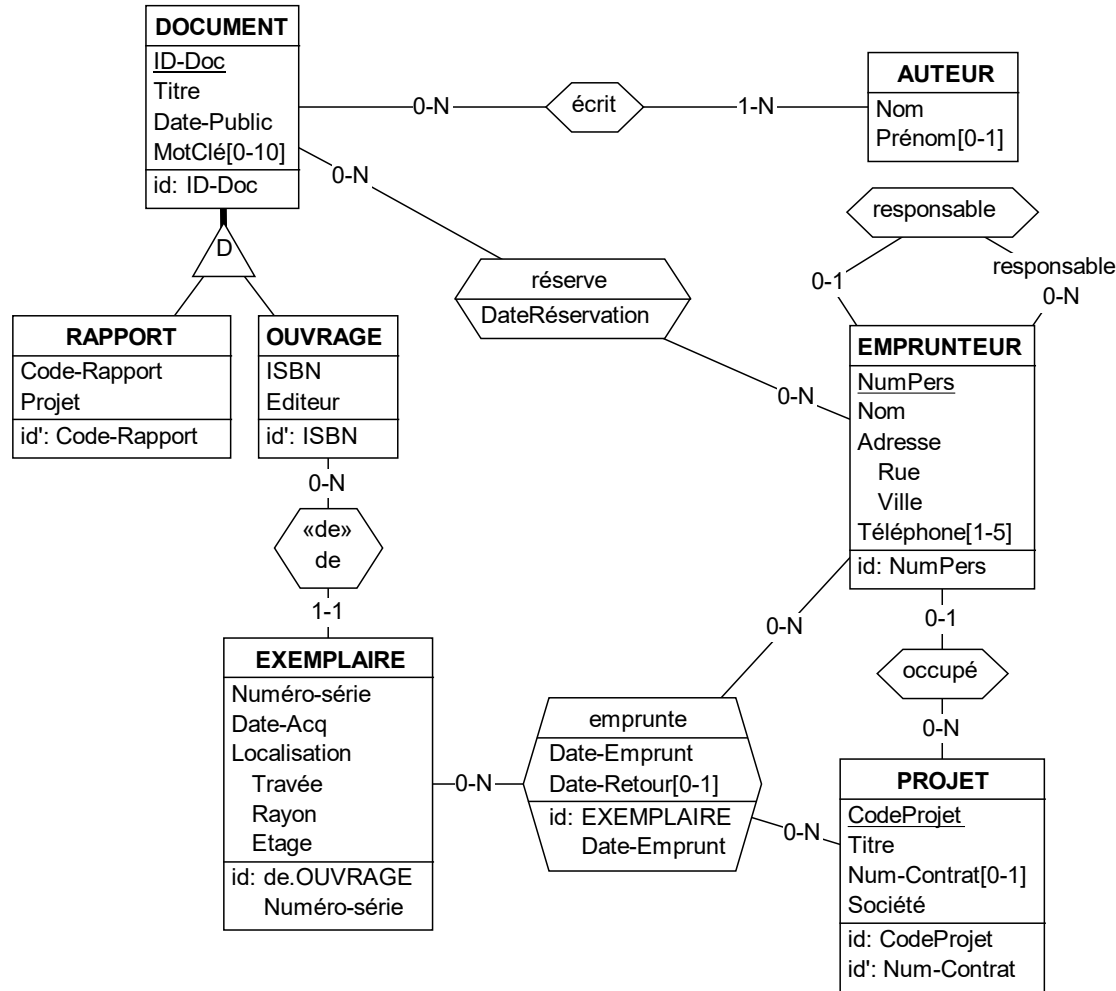
Un exemple plus complet



**le losange est un design pattern typique :
 voyages en voiture, en train, transports en commun, etc.**

2.8 EXEMPLE DE SCHEMA ENTITE-ASSOCIATION

2.8 Exemple de schéma EA



$(de * emprunte)[DOCUMENT, EMPRUNTEUR] \cap réserve = \emptyset$

2.9 METADONNEES

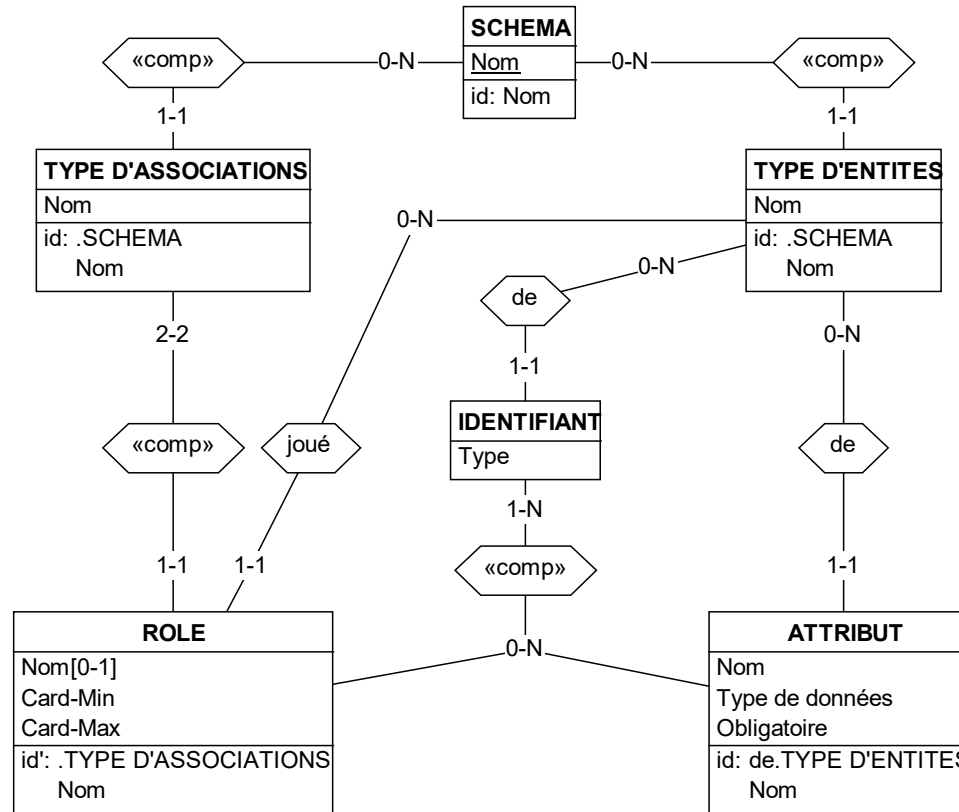
2.9 Métadonnées

Les informations (données) relatives à des schémas constituent des *métadonnées*.

Tout schéma de ces données est un *métaschéma*.

Un exemple bien connu : le catalogue d'une BD relationnelle.

2.9 Métadonnées - Métaschéma du modèle EA de base



ROLE..TYPE D'ASSOCIATIONS..SCHEMA = ROLE..IDENTIFIANT.de.TYPE D'ENTITES..SCHEMA

ATTRIBUT..IDENTIFIANT.de.TYPE D'ENTITES = ATTRIBUT.de.TYPE D'ENTITES

encore incomplet !

2.10 SEMANTIQUE DU MODELE ENTITE-ASSOCIATION

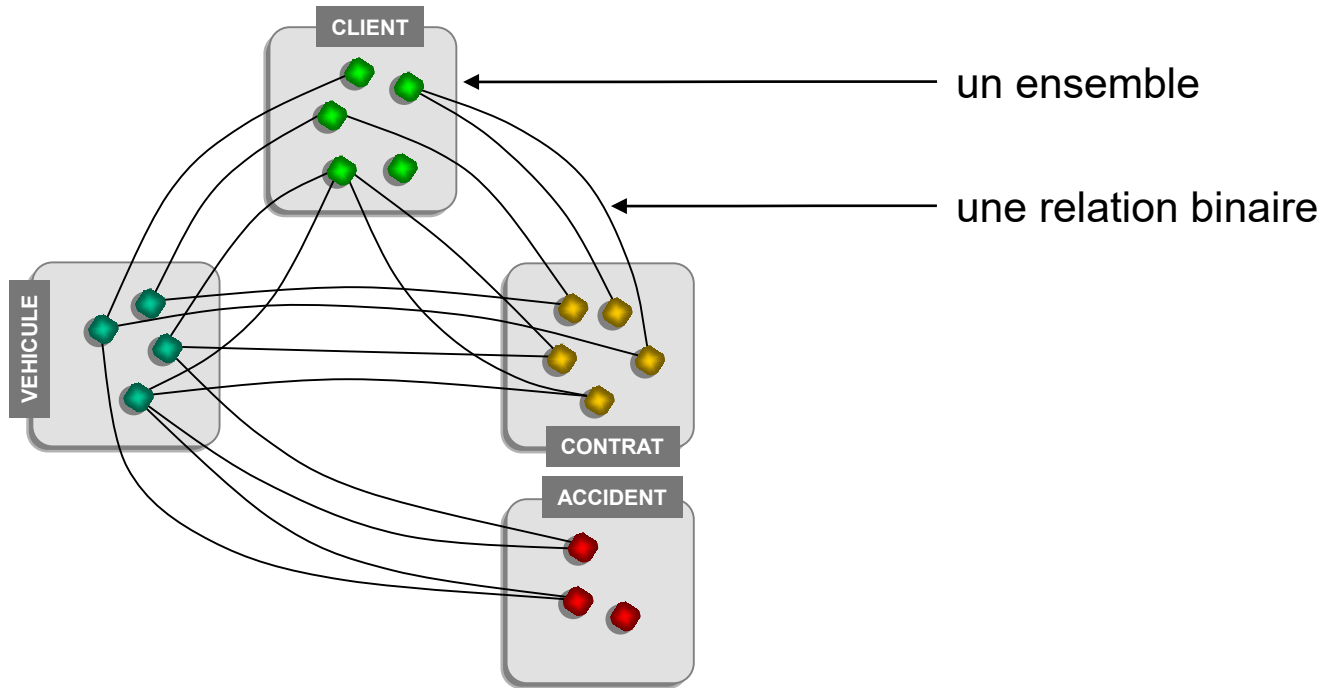
2.10 Sémantique du modèle EA

Sémantique interne du modèle EA

Mathématiquement parlant, qu'est-ce que :

- un type d'entités ?
- un type d'associations ?
- un attribut ?
- une contrainte ?

2.10 Sémantique du modèle EA



Inutile d'inventer une nouvelle théorie, on utilise ce qui est bien connu :

- la théorie des ensembles
- la théorie relationnelle
- la logique des prédicats

2.10 Sémantique du modèle EA

Interprétation des types d'entités

il existe un ensemble initial nommé **entités**, dont chaque élément représente (dénote) une entité du domaine d'application

SALARIE, CAISSE : entités

SALARIE et CAISSE sont des sous-ensembles de entités

EMPLOYE, OUVRIER : SALARIE

EMPLOYE et OUVRIER sont des sous-ensembles (sous-types) de SALARIE

2.10 Sémantique du modèle EA

Interprétation des types d'associations

spécifie(COMMANDE, PRODUIT, FOURNISSEUR)
offre(FOURNISSEUR, PRODUIT)

vendu(vendeur: CLIENT, acheteur: CLIENT, IMMEUBLE)

2.10 Sémantique du modèle EA

Interprétation des rôles multitype d'entités

possède(VEHICULE, propriétaire: PERSONNE \cup SERVICE)

2.10 Sémantique du modèle EA

Interprétation des attributs

descr-SALARIE(SALARIE, Matricule, Nom)

SALARIE → Matricule, Nom
Matricule → SALARIE

descr-SALARIE(SALARIE, Matricule, Nom)

réserve(DOCUMENT, EMPRUNTEUR, Date Réservation)
emprunte(EXEMPLAIRE, EMPRUNTEUR, PROJET, Date Emprunt, Date Retour)

2.10 Sémantique du modèle EA

Interprétation des attributs composés et multivalués

Adresse-type(Rue, Ville)

descr-EMPRUNTEUR(EMPRUNTEUR, NumPers, Adresse: Adresse-type, ...)

descr-EMPRUNTEUR(EMPRUNTEUR, NumPers, Adresse(Rue, Ville), ...)

descr-EMPRUNTEUR(EMPRUNTEUR, NumPers, Adresse, Téléphone[1-5])

descr-CLIENT(CLIENT, NCli, Contact[1-5] (Statut, Adresse(Rue, Ville), ...))

attention : on utilise ici une théorie relationnelle non standard !

2.10 Sémantique du modèle EA

Interprétation des attributs et rôles obligatoires

descr-SALARIE[SALARIE] = SALARIE

ACCIDENT : entités

descr-ACCIDENT(ACCIDENT, NumAcc, Montant[0-1])

descr-ACCIDENT[ACCIDENT] = ACCIDENT

appartient(VEHICULE, CLIENT)

appartient[VEHICULE] = VEHICULE

2.10 Sémantique du modèle EA

Interprétation des identifiants

descr-CLIENT(CLIENT, NumClient, Nom, Adresse)

desc-VOL(VOL, Ligne, Date, NumAppareil)

desc-PROJET(PROJET, CodeProjet, Titre, Budget)

appartient(VEHICULE, CLIENT)

implique(VEHICULE, ACCIDENT)

offre(PRODUIT, FOURNISSEUR, Prix, Délai)

reserve(DOCUMENT, EMPRUNTEUR, Date Réservation)

emprunte(EMPRUNTEUR, PROJET, EXEMPLAIRE, Date Emprunt, Date Retour)

2.10 Sémantique du modèle EA

Interprétation des identifiants hybrides

CONTRAT, CLIENT: entités
descr-CONTRAT(CONTRAT, NumCtr, Type, DateSign)
signe(CONTRAT, CLIENT)
descr-CONTRAT[CONTRAT] = CONTRAT
signe[CONTRAT] = CONTRAT

CONTRAT identifié par CLIENT et NumCtr

on en déduit

descr-CONTRAT[CONTRAT] = signe[CONTRAT]

on retient

signe(CONTRAT, CLIENT)
descr-CONTRAT(CONTRAT, NumCtr, Type, DateSign)
descr-CONTRAT[CONTRAT] = signe[CONTRAT]

2.10 Sémantique du modèle EA

Interprétation des identifiants hybrides

$R(A,B,C,D)$
 $A \longrightarrow B$

\Leftrightarrow

$R1(\underline{A},B)$
 $R2(A,C,D)$
 $R1[A] = R2[A]$

signe(CONTRAT, CLIENT)
descr-CONTRAT(CONTRAT, NumCtr, Type, DateSign)
descr-CONTRAT[CONTRAT] = signe[CONTRAT] (= CONTRAT)

TA fonctionnel : forme standard

\Updownarrow

descr-CONTRAT(CONTRAT, CLIENT, NumCtr, Type, DateSign)
(descr-CONTRAT[CONTRAT] = CONTRAT)

TA fonctionnel : forme compacte

2.10 Sémantique du modèle EA

Expression des autres contraintes

offre(FOURNISSEUR, PRODUIT, Prix, Délai)

$\forall f \in \text{FOURNISSEUR}, 0 \leq |\text{offre}(\text{FOURNISSEUR}=f)| \leq 10$

Cardinalité d'un rôle

PERSONNE = CONTRIBUABLE \cup MALADE

VEHICULE \cap IMMEUBLE = \emptyset

Contraintes de sous-types

descr-EMPLOYE(EMPLOYE, Matricule, Nom,
SalaireHoraire[0-1], SalaireMensuel[0-1],
DateEngagement[0-1], PROJET[0-1])

descr-EMPLOYE(non-null(PROJET))[EMPLOYE]
= descr-EMPLOYE(non-null(DateEngagement))[EMPLOYE]

descr-EMPLOYE(non-null(SalaireHoraire))[EMPLOYE]
 \cap descr-EMPLOYE(non-null(SalaireMensuel))[EMPLOYE] = \emptyset

Contraintes d'existence

2.10 Sémantique du modèle EA

Expression des autres contraintes

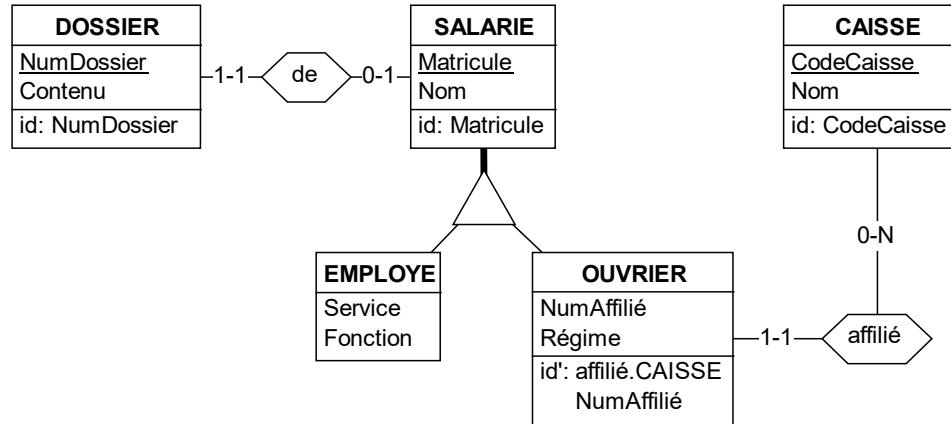
ETUDIANT, MATIERE, PROFESSEUR: entités
descr-PROFESSEUR(PROFESSEUR, Matricule, Nom, Ecole, Directeur)
inscrit(ETUDIANT, MATIERE, PROFESSEUR)
descr-PROFESSEUR: Ecole \longrightarrow Directeur
inscrit: PROFESSEUR \longrightarrow MATIERE

Dépendances fonctionnelles

départ[SECTION, STATION] \cap arrivée[SECTION, STATION] = \emptyset
appartient[VEHICULE, CLIENT] = (couvre*signe)[VEHICULE, CLIENT]

Contraintes cycliques

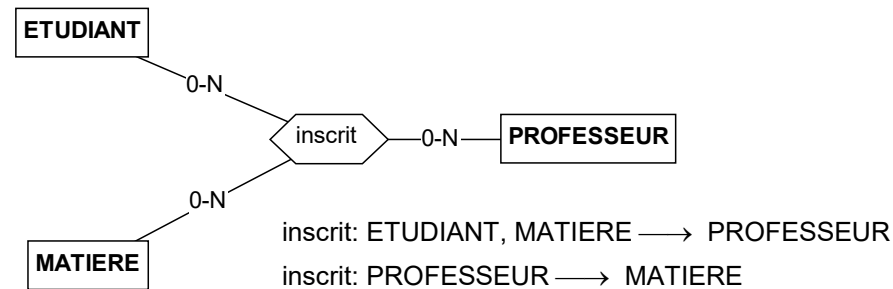
2.10 Sémantique du modèle EA - Exemple complet



DOSSIER, SALARIE, CAISSE : entités; EMPLOYE, OUVRIER : SALARIE
descr-DOSSIER(DOSSIER, NumDossier, Contenu)
descr-SALARIE(SALARIE, Matricule, Nom)
descr-EMPLOYE(EMPLOYE, Service, Fonction)
descr-OUVRIER(OUVRIER, NumAffilié, affilié: CAISSE, Régime)
descr-CAISSE(CAISSE, CodeCaisse, Nom)
de(DOSSIER, SALARIE)
descr-DOSSIER[DOSSIER] = DOSSIER
descr-SALARIE[SALARIE] = SALARIE
descr-EMPLOYE[EMPLOYE] = EMPLOYE
descr-OUVRIER[OUVRIER] = OUVRIER
descr-CAISSE[CAISSE] = CAISSE
de[DOSSIER] = DOSSIER

2.10 Sémantique du modèle EA - Applications

Application : calcul des identifiants d'un TA

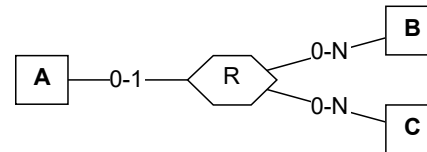


inscrit(ETUDIANT, MATIERE, PROFESSEUR)
inscrit: ETUDIANT, MATIERE \longrightarrow PROFESSEUR
inscrit: PROFESSEUR \longrightarrow MATIERE

deux identifiants {ETUDIANT, MATIERE} et {ETUDIANT, PROFESSEUR}

2.10 Sémantique du modèle EA - Applications

Application : décomposition d'un TA



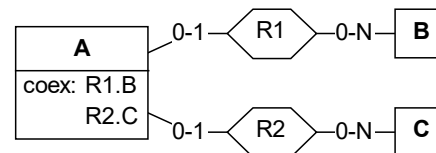
$R(\underline{A}, B, C)$



$R1(\underline{A}, B)$

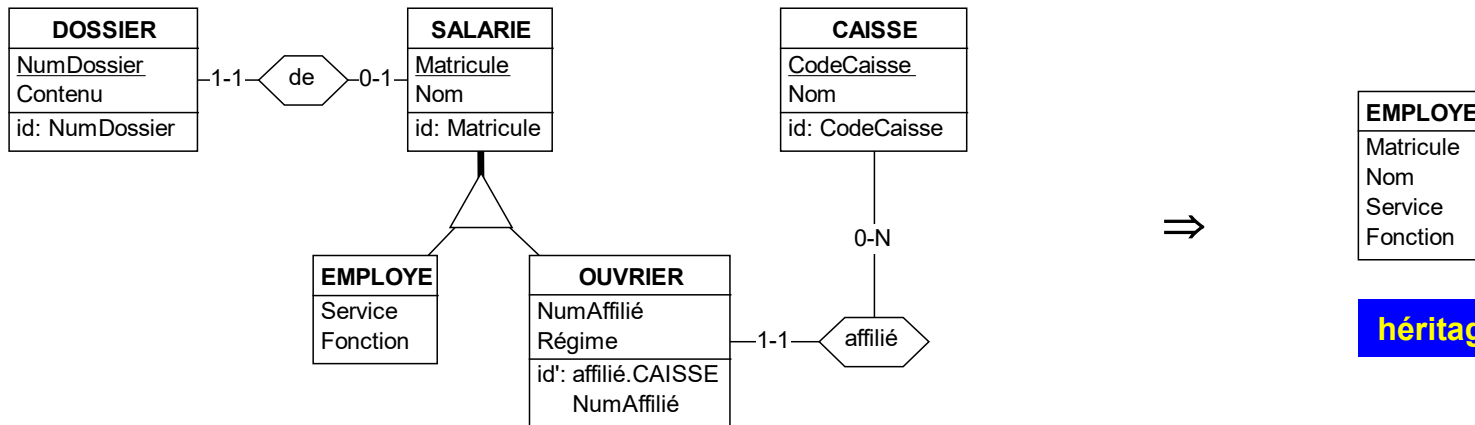
$R2(\underline{A}, C)$

$R1[A] = R2[A]$



2.10 Sémantique du modèle EA

Application : héritage dans les relations *is-a*



héritage descendant

descr-EMPLOYE'(EMPLOYE', Matricule, Nom, Service, Fonction)

=

descr-EMPLOYE(EMPLOYE)*descr-SALARIE(SALARIE) [EMPLOYE, Matricule, Nom, Service, Fonction]

Fin du module 2

Module suivant :

3. Analyse conceptuelle du domaine d'application

1. Méthodologie des BD
- 2. Le modèle Entité-association**
3. Analyse conceptuelle
4. Conception logique relationnelle

5. Conception physique
6. Production du code
7. Rétro-ingénierie