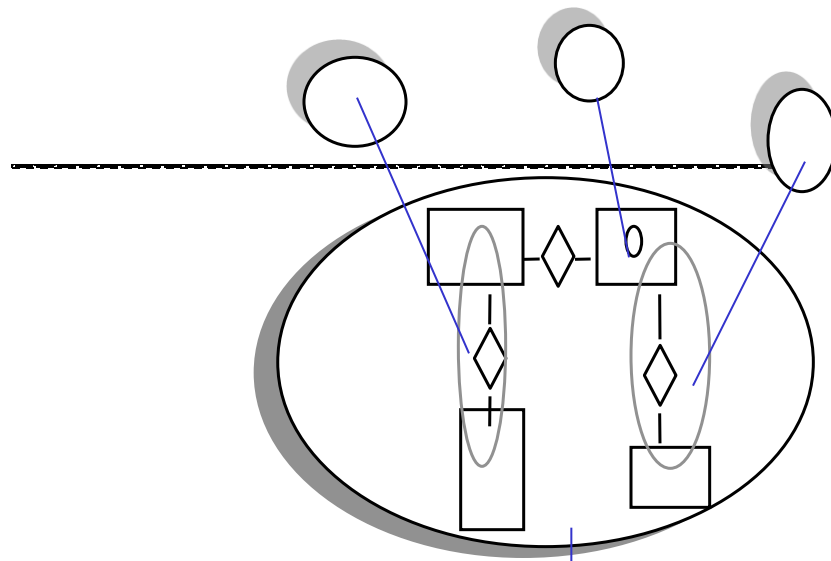


Modélisation de bases de données : Le modèle relationnel

Introduction

- Rappel chapitre 1
- C'est quoi un modèle ?
- Type de modèle :
 - Modèle hiérarchique
 - Modèle réseau
 - Modèle objet
 - Modèle relationnel

Les niveaux des BD



Vues externes

chaque vue représente
la partie de la bd relative
à chaque utilisateur / application

Schéma conceptuel

vue globale de la base :
les données stockées
et les relations parmi les données

R_1 (A number(10), B string, ..., primary key (A), ...)
....
 R_n (X number(15), Y string, ..., primary key (X), ...)

Schéma logique

Représente un modèle de données
conforme au modèle ciblé de SGBD
(par ex. SGBDR)

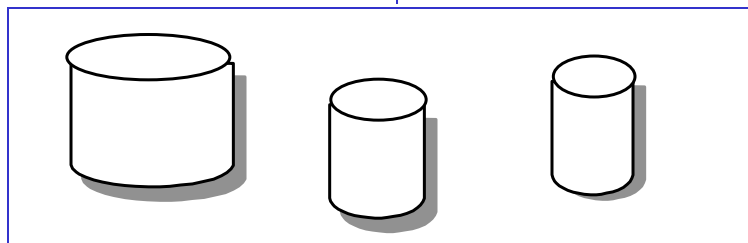


Schéma physique

décrit comment les données
sont stockées dans la bd +
les index

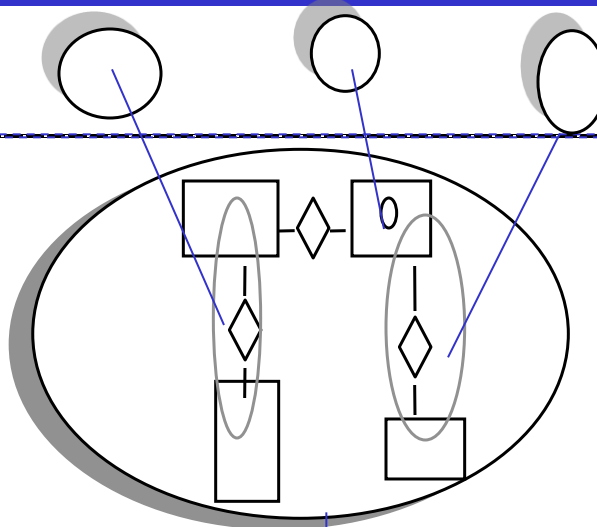
Caractéristiques de l'Architecture

Schéma externe

Schéma conceptuel

Schéma logique

Schéma physique



Indépendance logique :

Un changement du schéma conceptuel (nouvel attribut, nouvelle relation, ...), n'entraîne pas de changement dans les vues existantes

R_1 (A number(10), B string, ..., primary key (A), ...)

....

R_n (X number(15), Y string, ..., primary key (X), ...)

Indépendance physique :

Les changements du schéma interne ne sont pas visibles aux niveaux des schémas conceptuels et logiques; conséquences sur les performances

Cfr. ANSI / SPARC 1975

Quels types d'études dans les BD et les SGBD ?

- Conception de BD
 - À partir de l'analyse du contexte, recueillir les besoins :
 - Quelles informations stocker ? Comment les organiser ?
 - Quelles contraintes prendre en compte ? Comment les représenter ?
 - Quels types d'usages de ces données ?
- Programmation
 - Développement et optimisation de requêtes et programmes
 - Interfaces avec d'autres applications
- Implémentation de BD
 - support de nouveaux types de données, optimiseurs, intégration de données et d'applications, interfaces utilisateurs, langages

Type de modèle

Un schéma conceptuel est donc le résultat d'un processus de modélisation fait en respectant les possibilités d'un modèle de données.

Le modèle de données est une caractéristique de tout SGBD. Il existe quatre grandes classes de modèles de données qui se distinguent par la nature des associations qu'ils permettent de modéliser.

- **Les modèles hiérarchiques**
- **Les modèles réseaux**
- **Les modèles objets**
- **Les modèles relationnels**

Efforts de standardisation et de développement des SGBD.

- **CODASYL (CO nference on DA ta SYstems Language)**
- **ANSI (American National Standard Institute)**
- **GUIDE/SHARE (groupe d'usagers IBM)**

Le groupe CODASYL

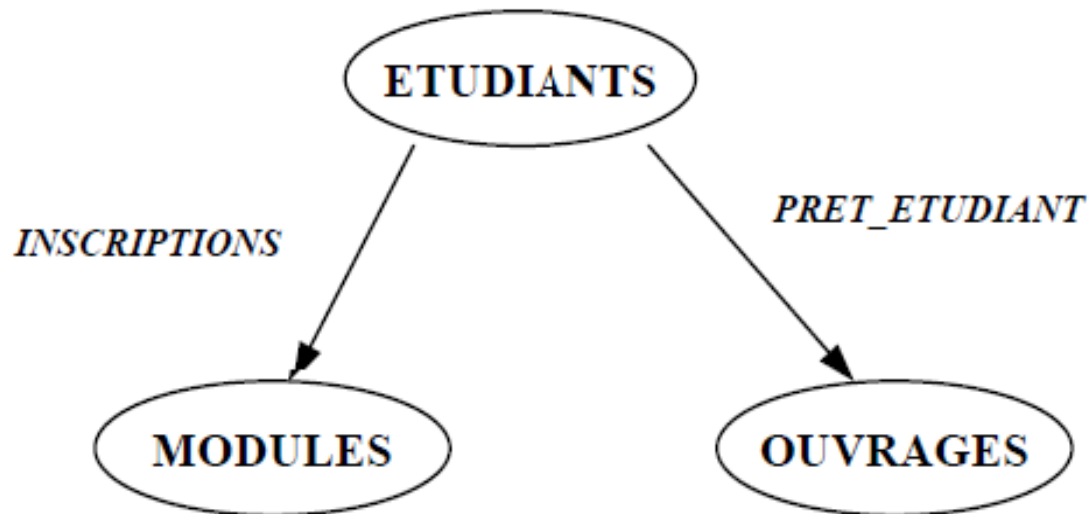
- ***Enregistrement*** : collection d'agrégats et d'atomes rangés consécutivement constituant l'unité d'échange entre la base de données et les applications.
- • ***SETS*** : Association entre un enregistrement père (*propriétaire ou Owner*) et un enregistrement fils (*Membre ou Member*).

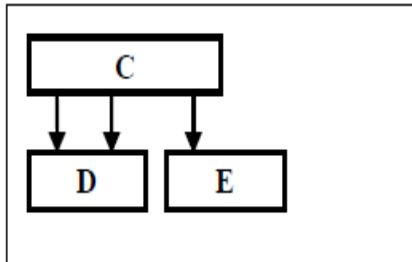
Modèle hiérarchique (1)

- A l'aide du modèle hiérarchique, le schéma conceptuel peut être vu comme un graphe arborescent dont les nœuds correspondent aux classes d'objets (entités) et les arcs entre deux nœuds aux liaisons ou associations entre les entités.
- Un tel graphe possède donc un nœud racine (*sur lequel n'arrive aucun arc !*) et les autres nœuds sont des fils, petit-fils, etc., de cette racine.
- Avec le modèle hiérarchique, le nombre de flèches pouvant arriver sur un nœud est donc égal à un (sauf pour le nœud racine).

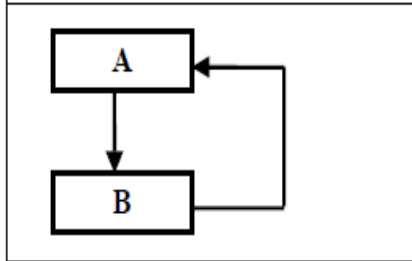
Modèle hiérarchique (2)

exemple de modèle hiérarchique

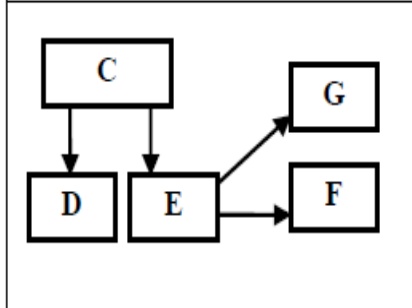




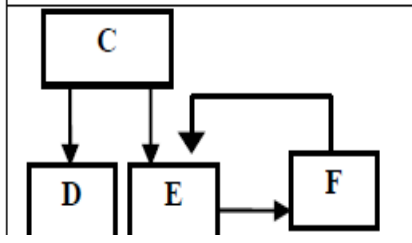
Ce modèle n'est pas conforme au modèle hiérarchique à cause des deux arcs qui arrivent sur le segment **D**



Ce modèle n'est pas conforme au modèle hiérarchique car il ne possède pas de **segment Racine** sur lequel n'arrive aucun arc.



Ce modèle est conforme au modèle hiérarchique car il possède un segment Racine qui est **C** et sur chaque segment n'arrive qu'un seul arc.



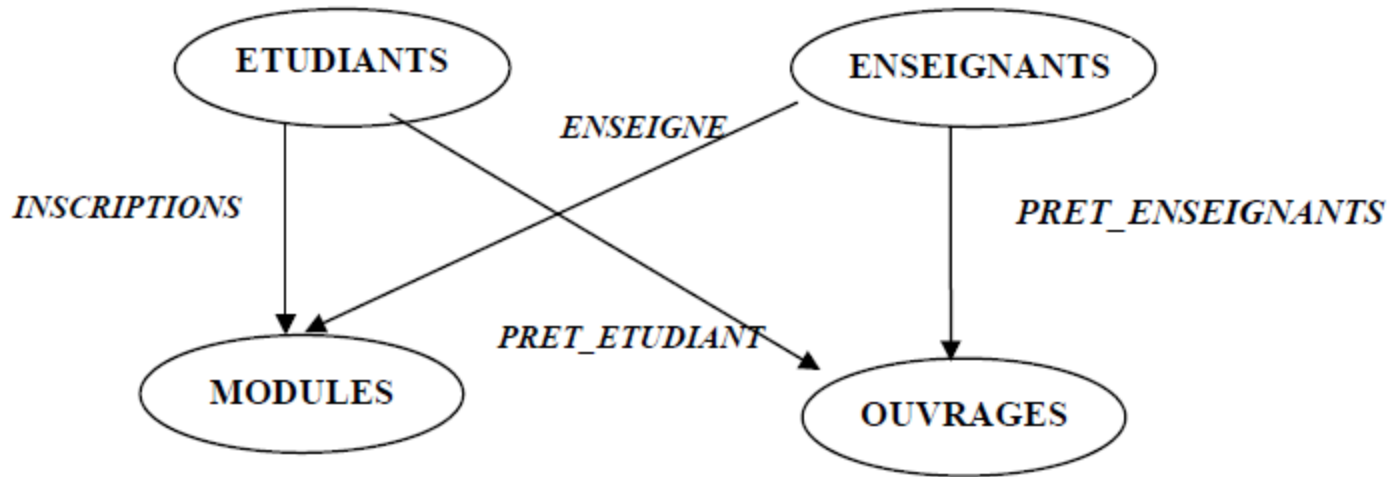
Ce modèle n'est pas conforme au modèle hiérarchique à cause des deux arcs qui arrivent sur le segment **E**

Modèle réseaux (1)

- A l'aide de ce modèle, le schéma conceptuel peut être vu comme un graphe général où les nœuds correspondent aux classes d'objets et les arcs entre deux nœuds aux associations.
- A la différence du modèle hiérarchique on peut avoir ici plusieurs arcs qui arrivent sur le même nœud. De même que la notion de nœud racine n'existe pas avec le modèle réseau.

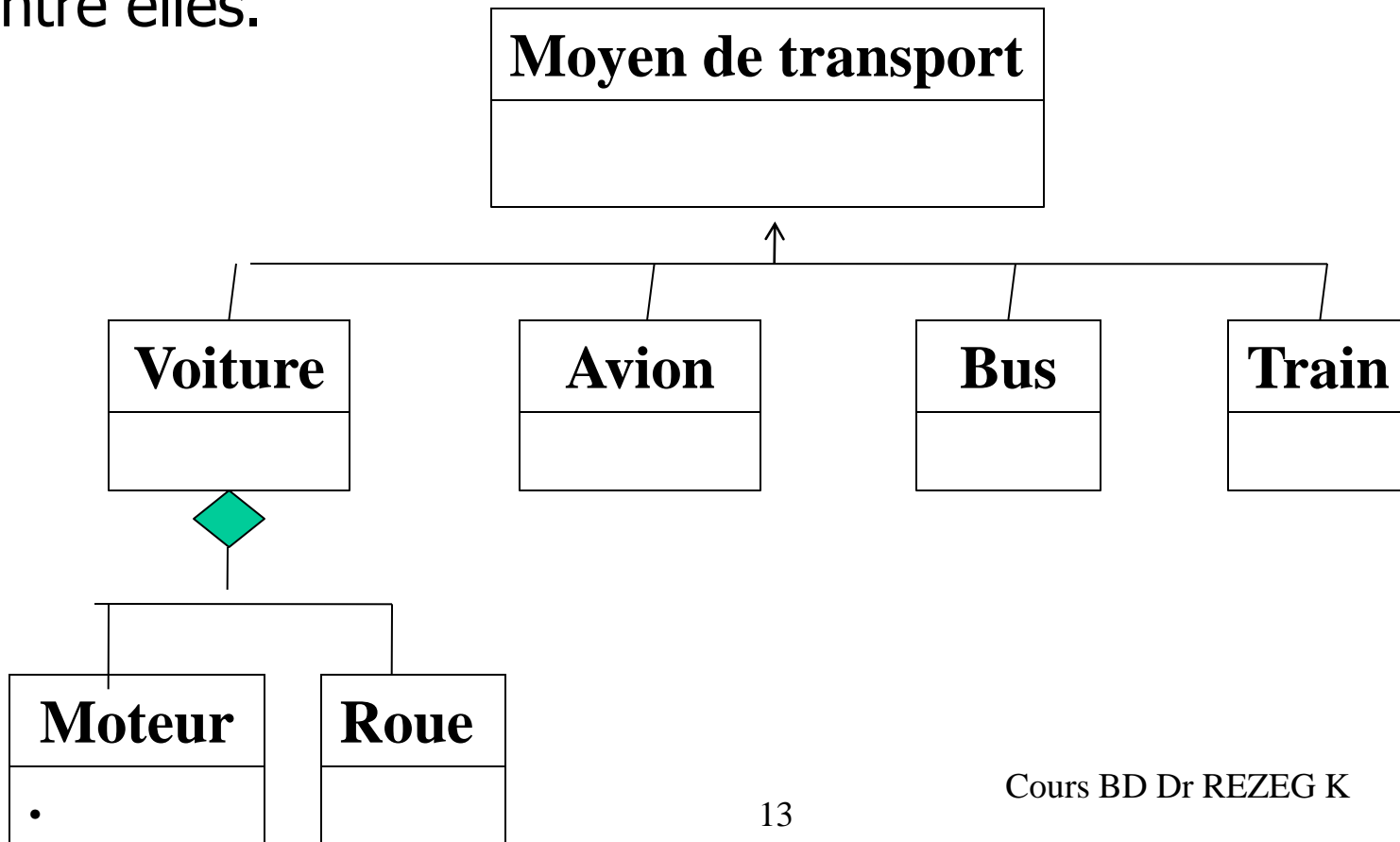
Modèle réseaux (2)

Exemple de modèle Réseau



Modèle Objet (1)

- Permet de voir une base de données comme un ensemble de classes d'objets, ayant des liens d'héritage, d'agrégation, de composition ou de simple association entre elles.



Modèle relationnel (1)

Le modèle relationnel a été introduit au début des années 70, c'est le premier modèle de bases de données indépendant des critères de stockage, il consiste à percevoir la base de données comme un ensemble de relations qui peuvent être représentées sous forme de table à deux dimensions : Les colonnes correspondent aux attributs d'une relation et les lignes correspondent aux tuples. Contrairement aux autres modèles de conception qui distinguent entre les concepts d'entité et de lien ou relation, ce modèle utilise un seul concept qui est la relation, ce qui lui rend plus facile à utiliser même pour les utilisateurs ayant plus ou moins de connaissance en informatique.

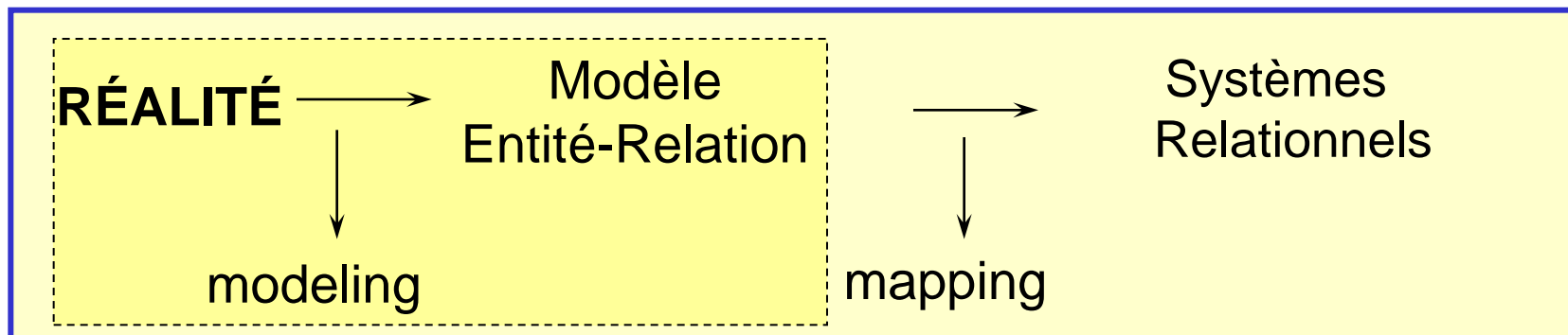
Ce modèle est basé essentiellement sur une théorie mathématique « la théorie des ensembles »

2 - Modèle Entités - Relations

- Contenu
 - Concepts de base
 - Entités, attributs
 - Relations
 - Clés
 - Multiplicités, cardinalités
 - Concepts avancés
 - Héritage
 - Agrégation, Composition
 - Entités faibles

Objectif du modèle de données

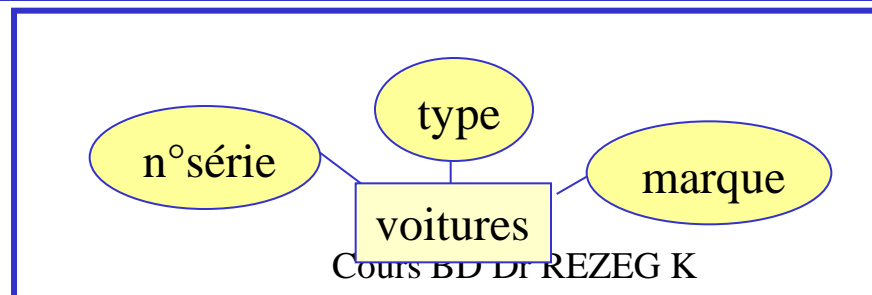
- Apporter une interprétation du contexte applicatif
 - en soulignant les aspects fondamentaux
 - en négligeant les détails
- Identifier les données gérées par la base de données et leur relations
- Produire un diagramme, suivant un formalisme, en vue de convertir le modèle de données en schéma de base relationnelle



Entité -Attributs

- **Entité**
 - "une chose" qui existe et qui peut être distinguée de façon unique.
Ex.: un étudiant, une voiture, une banque
 - abstraite ou concrète
- **Attribut**
 - propriété d'une entité
 - prend des valeurs simples, par exemple entiers ou chaînes de caractères (domaine d'attribut)

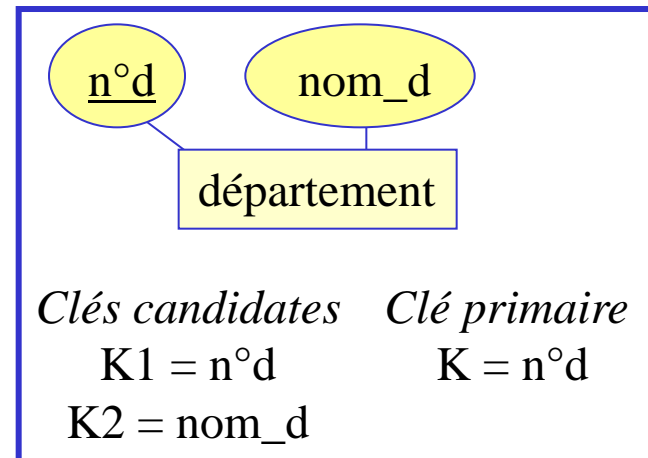
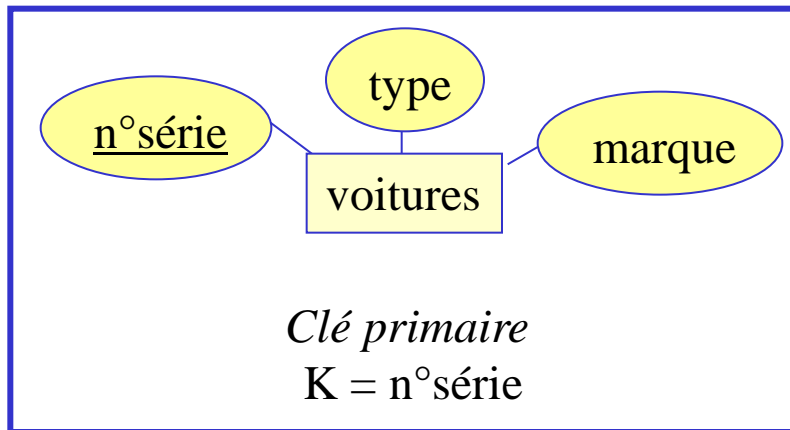
Ensemble d'Entités	Attributs	Domaines
Voitures	n°série marque type	entier (12) chaîne de car. (8) chaîne de car. (10)



Représentation
sous forme de
diagramme

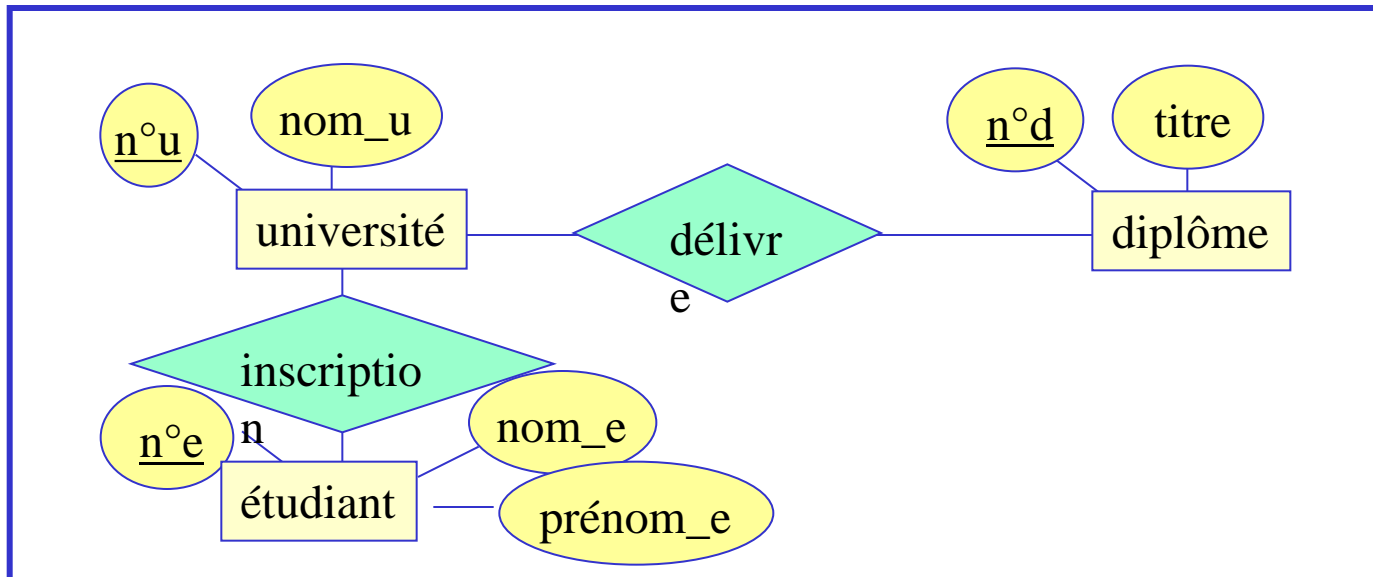
Clé des ensembles d'entités

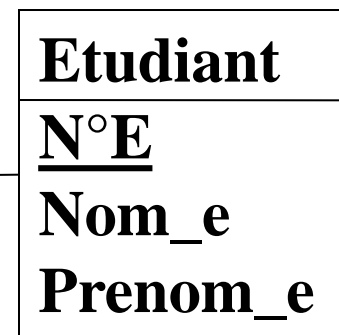
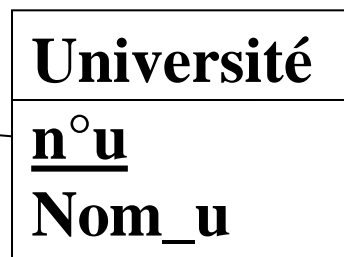
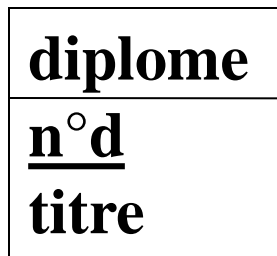
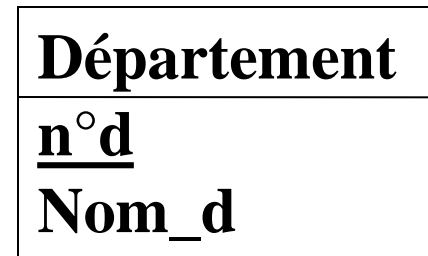
- **clé candidate**: un ensemble minimal d'attributs qui identifie de façon unique une occurrence d'entité
- **clé primaire**: une clé candidate choisie pour identifier de façon unique chaque occurrence d'entité
- **clé composée**: une clé candidate composée de deux ou plusieurs attributs



Relations

- Une relation relie deux ou plusieurs ensembles d'entités
- Ex.:
 - Des universités délivrent des diplômes
 - Des étudiants sont inscrits dans des universités





Ensemble de relations

- La "**valeur**" d'une relation est l'ensemble des listes des entités **réellement** associées par la relation. Chaque liste est obtenue en correspondance des ensembles d'entités en relation.

Exemple: valeur de la relation "délivre"

n°u	n°d
u ₁	d ₂
u ₁	d ₃
u ₁	d ₅
....	...
u ₁₀₀	d ₁
u ₁₀₀	d ₂

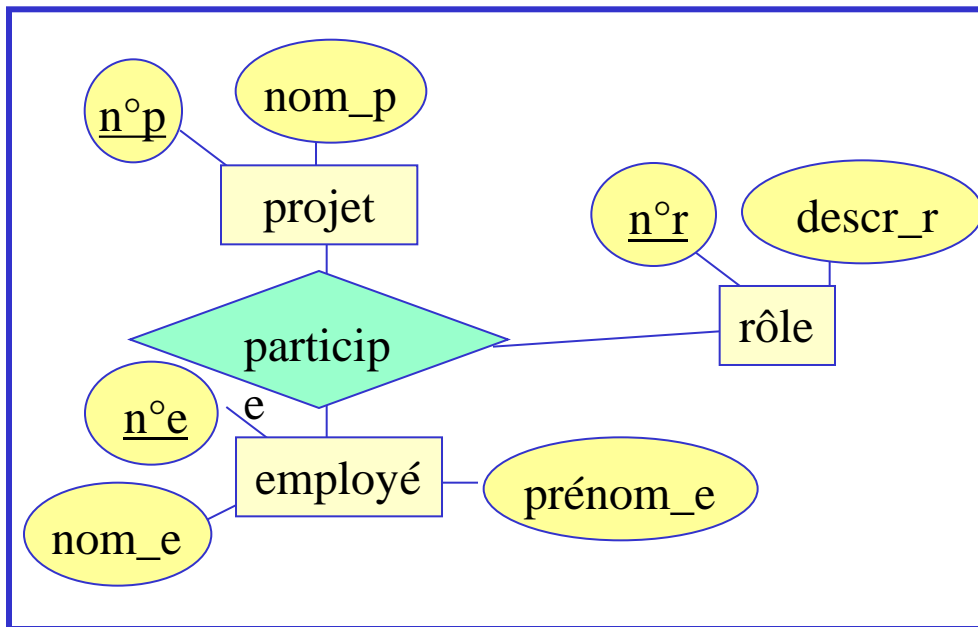
- l'université identifiée par u₁
délivre les diplômes identifiés
par d₂, d₃ et d₅,
...
l'université identifiée par u₁₀₀
délivre les diplômes identifiés
par d₁ et d₂

Ensemble de relations (2)

- Soit r une relation entre k ensembles d'entités E_1, E_2, \dots, E_k . Un ensemble d'entités peut apparaître plus d'une fois dans la liste.
- Soit un k -uplet $(e_1, \dots, e_k) \in r$
 - $e_1 \in E_1, \dots, e_k \in E_k$ sont associés par la relation r
 - e_1, \dots, e_k est dit tuple de r
 - k est le degré de la relation
 - $K = 1$: relation unaire (ou récursive, sur un même ensemble d'entités)
 - $K = 2$: relation binaire
 - $K = 3$: relation ternaire
 - ...
 - $K = n$: relation n -aire

Relations n-aires

- Une relation connectant plus de 2 ensembles d'entités permet de restreindre les combinatoires entre les valeurs des entités en relation
 Ex.: des employés participent à des projets avec des rôles

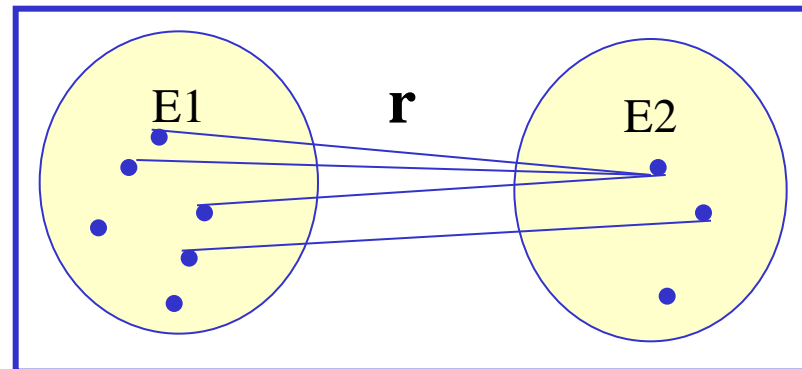


n°p	n°e	n°r
p1	e2	r1
p1	e3	r2
p1	e5	r3
...
p2	e2	r2
p2	e3	r1

Type des relations binaires

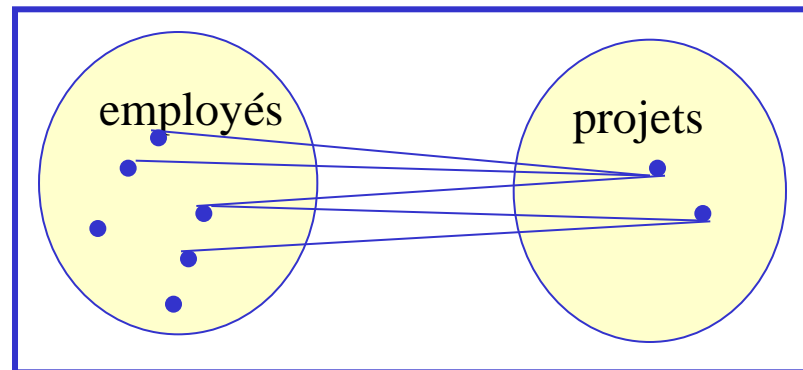
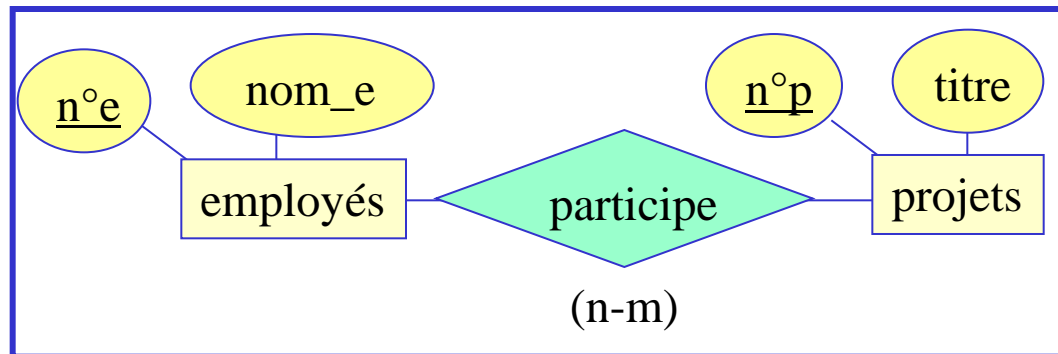
Soit r une relation binaire

Le **type** de r est lié au nombre d'occurrences d'une entité qui peuvent être associées avec une occurrence de l'autre entité



Relations de type m-n

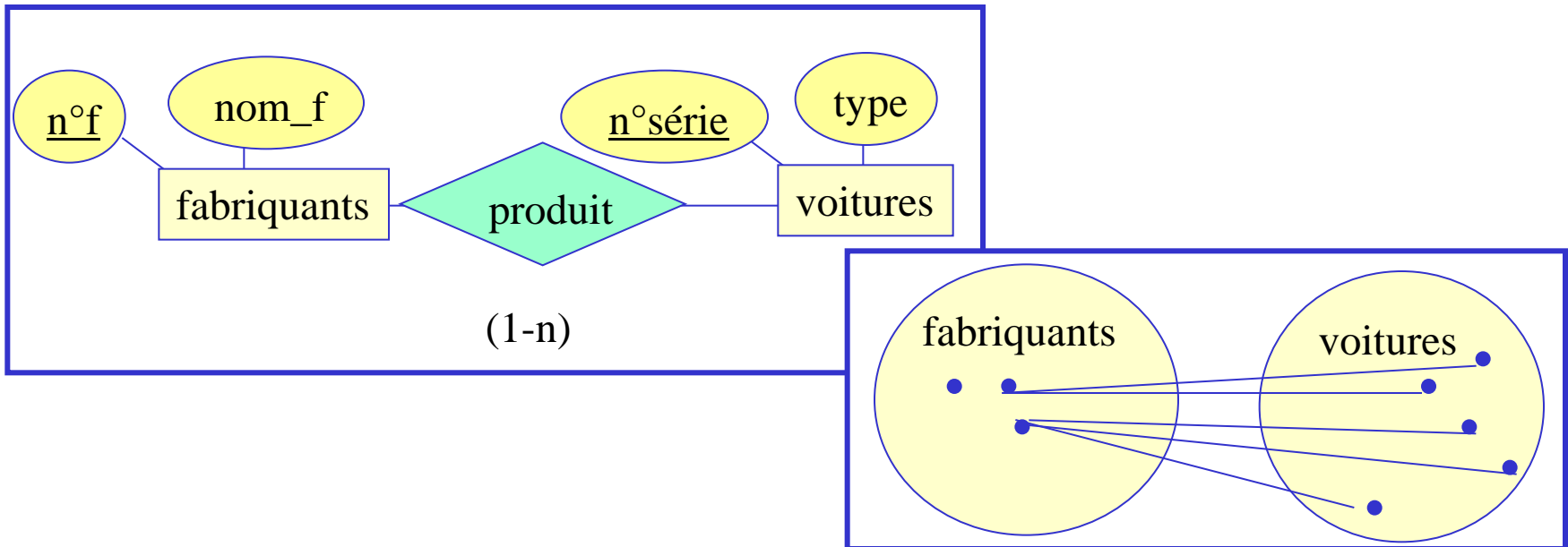
Dans une relation "plusieurs à plusieurs" ("many-many", m-n), une entité de chaque ensemble peut être connectée à plusieurs entités de l'autre ensemble



Relations de type 1-n

Dans une relation "un à plusieurs" ("one-many", 1-n),

- une entité d'un ensemble peut être connectée à au plus une entité du second ensemble
- mais une entité du second ensemble peut ne pas être connectée à aucune entité du premier ensemble, ou être connectée à une ou plusieurs entité de cet ensemble

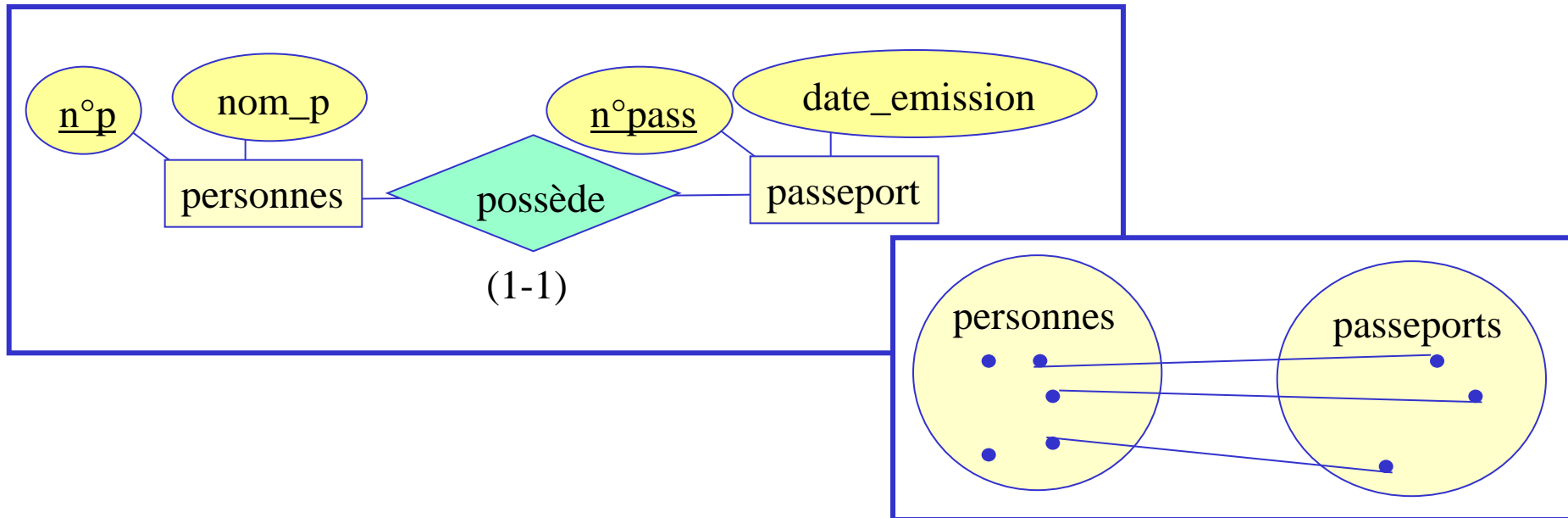


Relations de type 1-1

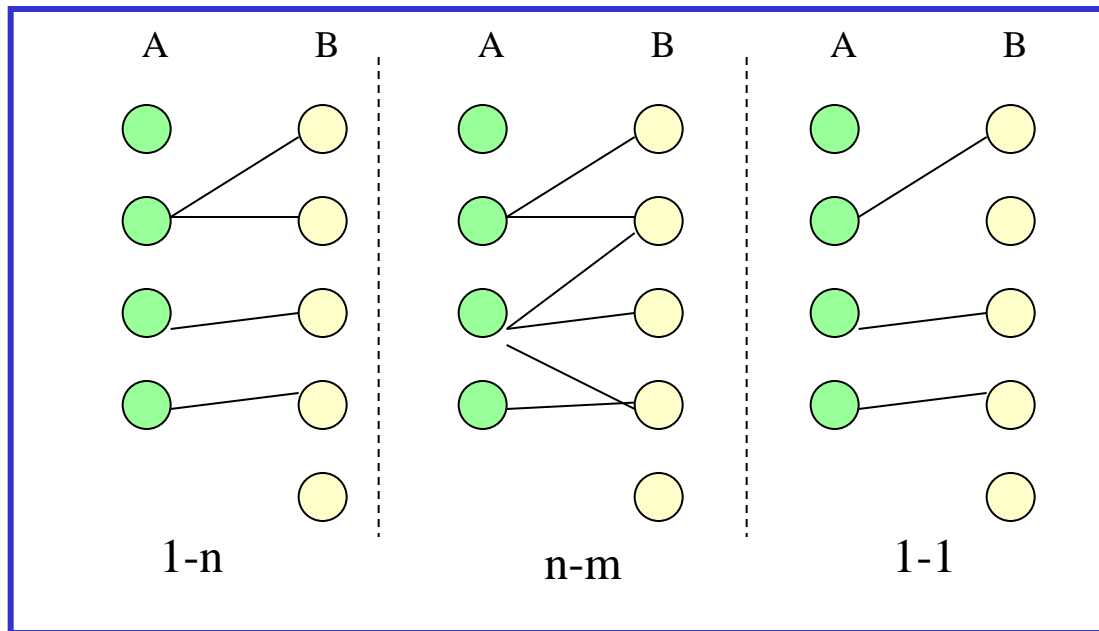
Dans une relation "un à un" ("one-one", 1-1),
une entité de chaque ensemble peut être connectée à au plus
une entité de l'autre ensemble

Exemple: une BD pour l'administration d'un pays

*une personne possède au plus un passeport
et un passeport a un seul titulaire*

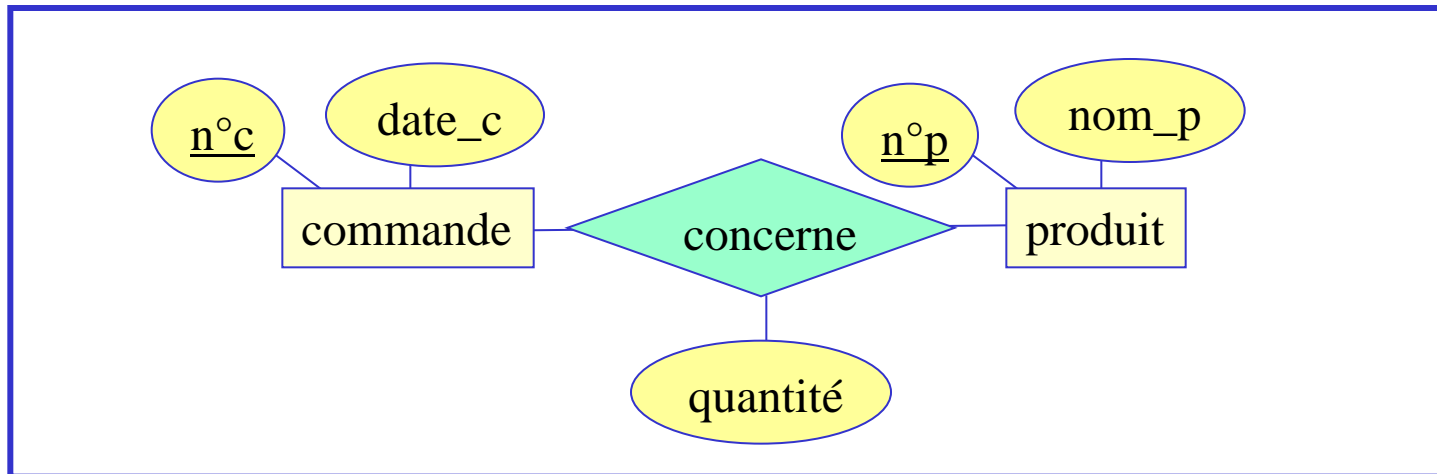


Synthèse sur les types des relations



Attributs de relation

Propriétés dont la valeur dépend des tuples dans l'ensemble de relations.



Cardinalités

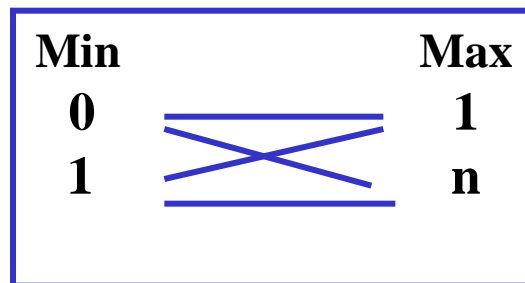
- Les BD sont conçues dans l'hypothèse du "monde fermé"

Les ensembles d'entités sont **FINIS** !!

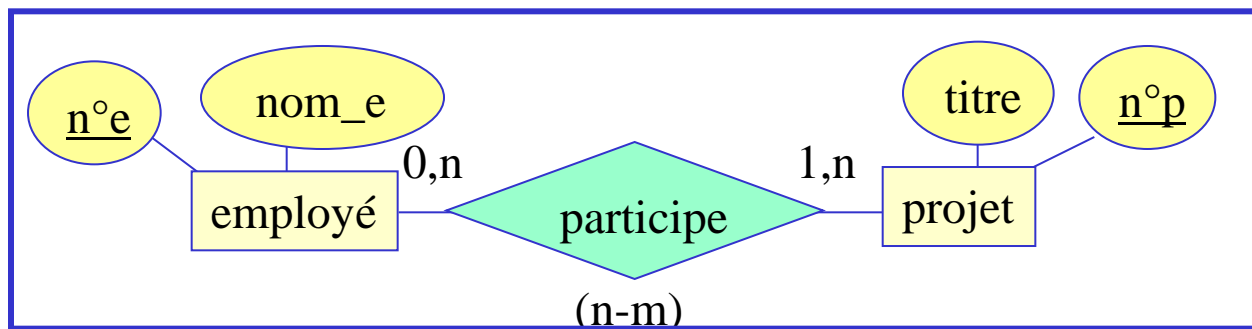
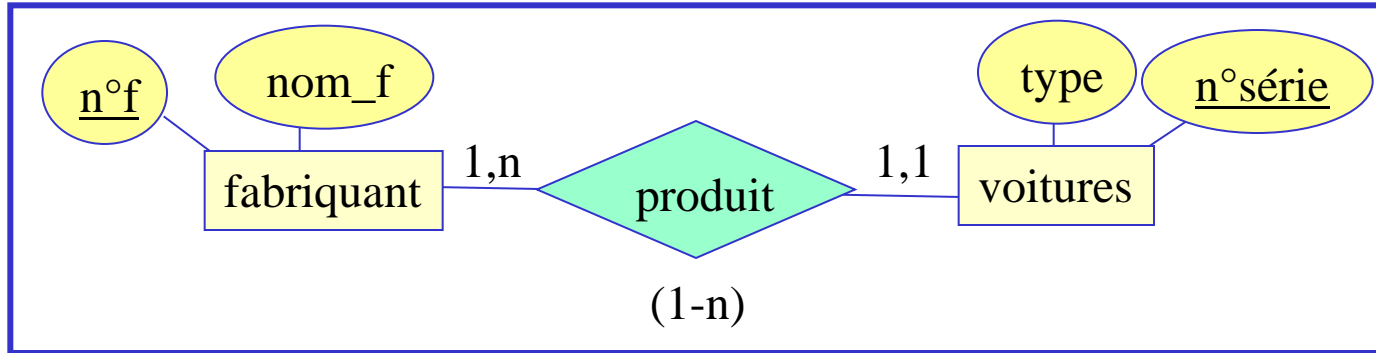
- **Cardinalités**

Couples (min, max) associés à chaque ensemble d'entité relié par une relation r

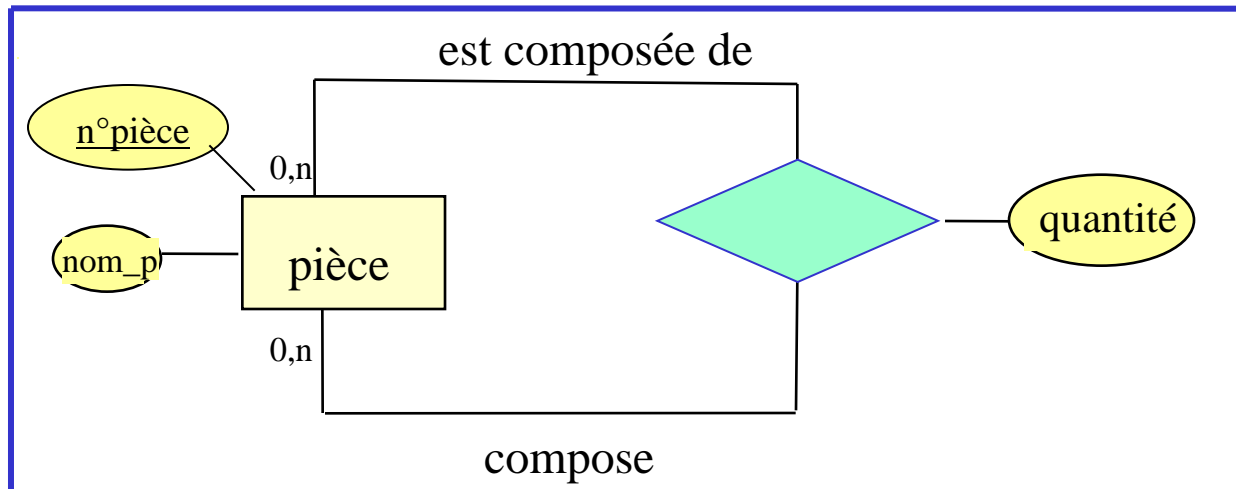
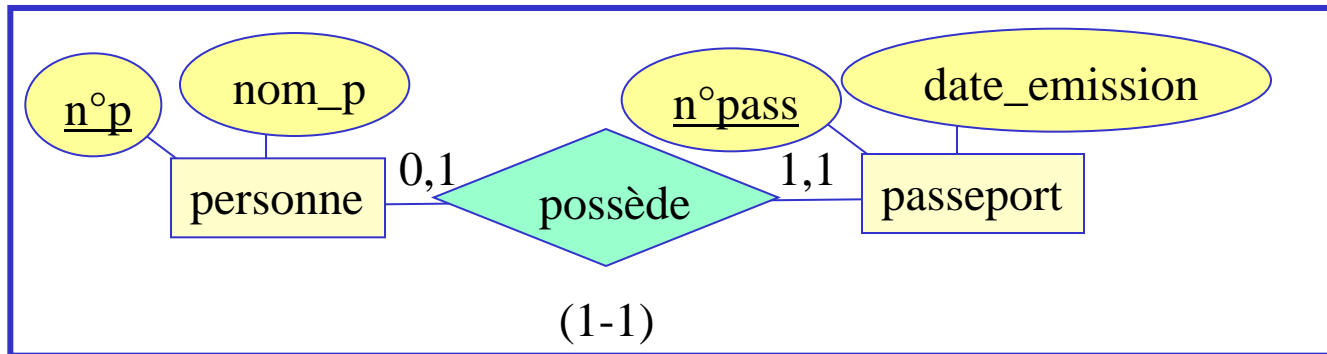
- Si r est binaire, entre E_1 et E_2 :
 - le min (resp. max) associé à E_i ($i=1,2$) représente le nombre minimum (resp. max) d'entités de E_j ($j=2,1$) associées à un élément quelconque de E_i .
- Si r est n -aire:
 - le nombre d'occurrences possibles d'entités associées dans cette relation quand les autres ($n-1$) valeurs sont fixées



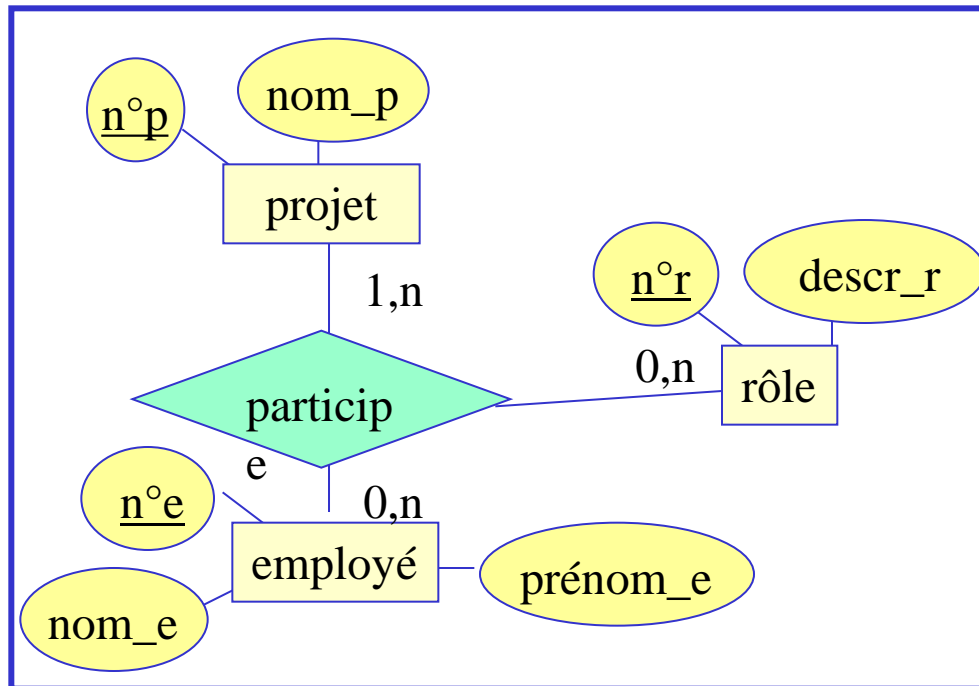
Exemples



Exemples



Exemple



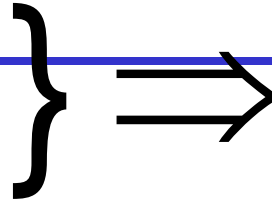
Modèle Entité – Association (synthèse)

Entité :

Une entité est une population d'individus homogènes.

Exemple: les produits ou les articles vendus par une société peuvent être regroupés dans une même entité **Articles**.

- Ordinateurs
- Imprimantes
- Scanner
- ...



Articles

ceci est possible du fait que ces produits ont les mêmes caractéristiques (par exemple : la désignation, le prix unitaire, la quantité, etc).

Clients

Fournisseurs

Association :

Une association est une liaison qui a une interprétation précise entre plusieurs entités.

Exemple : entre l'entité client et article il y a une liaison qui est **Commander** : un client commande un article, et entre fournisseurs et clients il y a la liaison : **Livrer**

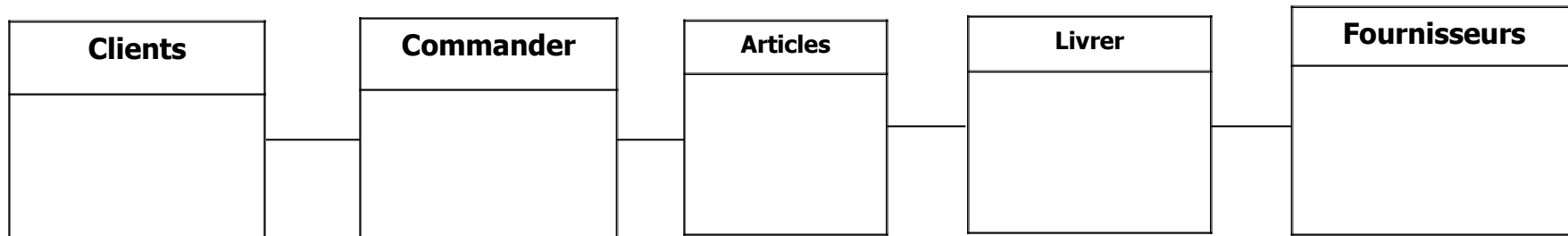


Fig. Associations

Attributs et identifiants :

Un attribut est une propriété (caractéristique) d'une entité ou d'une association.

Exemple: Dans l'exemple de la société, l'entité **Articles** a des attributs que nous avons déjà cités :

- Désignation,
- Quantité,
- prix unitaire,
- ...

L'entité **Client** peut avoir comme attributs:

- adresse client,
- nom et prénom du client,
- ...

Les associations **Commander** et **Livrer** peuvent avoir comme attributs :

- quantité commandée,
- date de livraison,
- ...

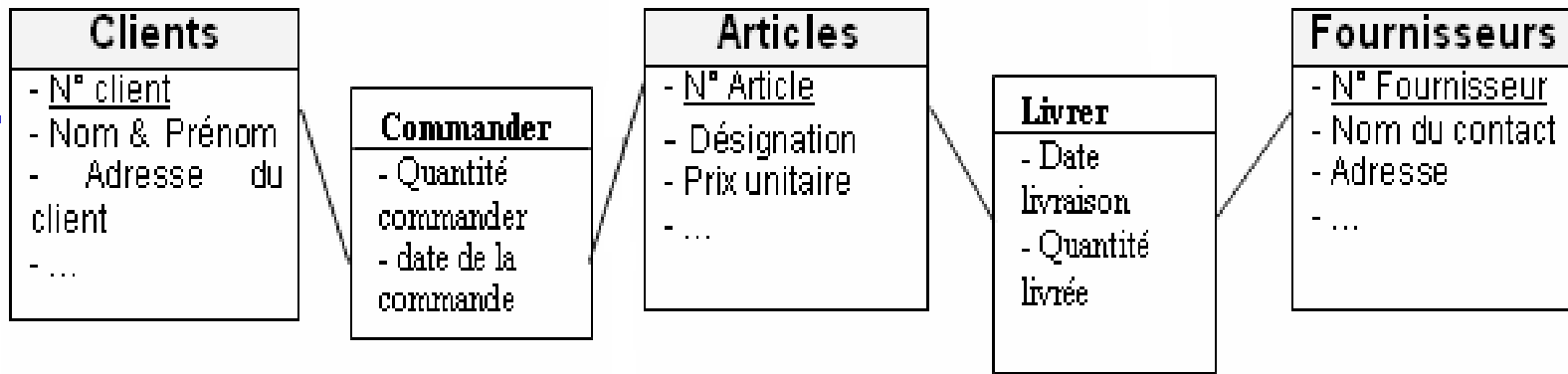


Fig 4. Attributs et identifiants

Chaque individu d'une entité doit être identifiable d'une manière unique et sans ambiguïté,

Exemple: L'individu **Said Hamidi** de l'entité **Clients** ne peut pas être identifier d'une façon unique par son nom :

Plusieurs clients peuvent avoir le même nom



chaque entité doit posséder un attribut sans doublon
(ne prenant pas deux fois la même valeur). Il
s'agit de

l'identifiant

Remarque : Une entité doit posséder au moins un attribut qui est son identifiant, par contre une association peut être dépourvu d'attributs.

Cardinalité

La cardinalité d'un lien entre une entité et une association précise le nombre de fois qu'un individu de l'entité peut être concerné par l'association.

Exemples : un client peut commander de 1 jusqu'à n articles.

Un articles |  tre commander 0 fois jusqu'à m fois



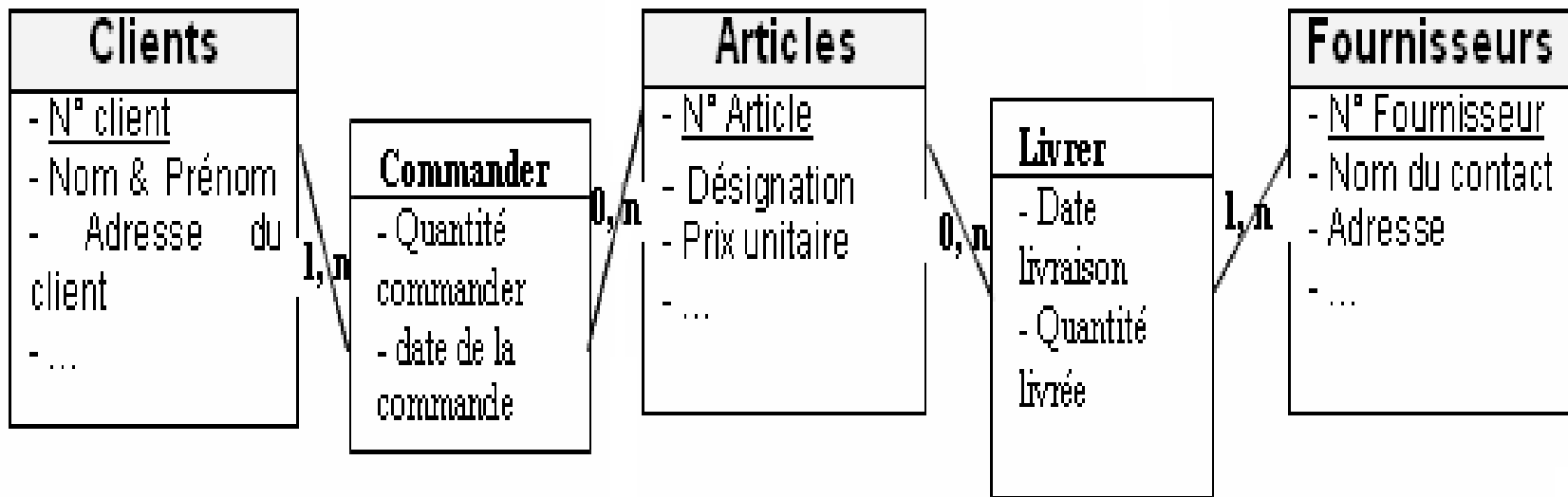


Fig 4. Cardinalité

- Un client ne peut exister que s'il commande au moins un article.
-

Cardinalité minimale pour le client est 1



- un article peut exister dans le stock même s'il n'est pas commandé par aucun client,

Cardinalité minimale pour l'article est 0.



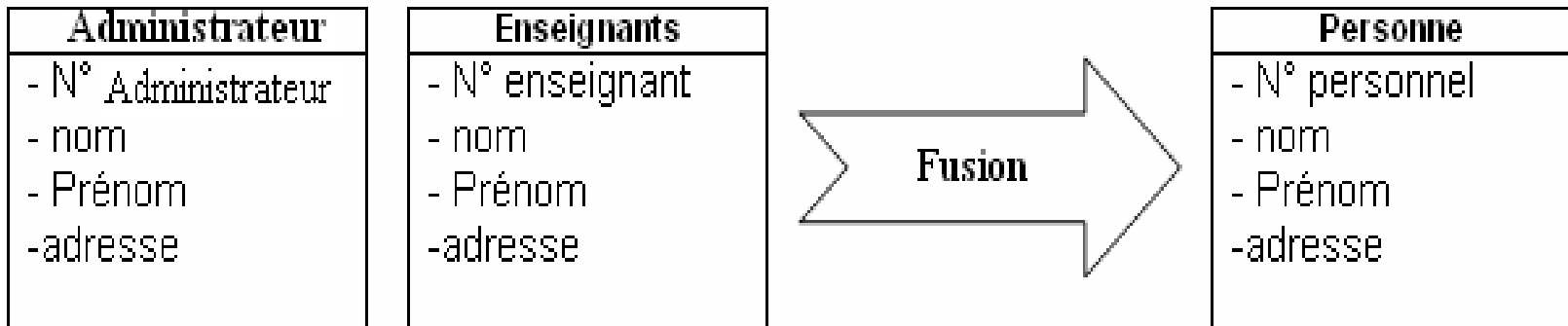
VI.4 Règles de modélisation :

Un bon schéma *Entités-association* doit vérifier certaines règles dites

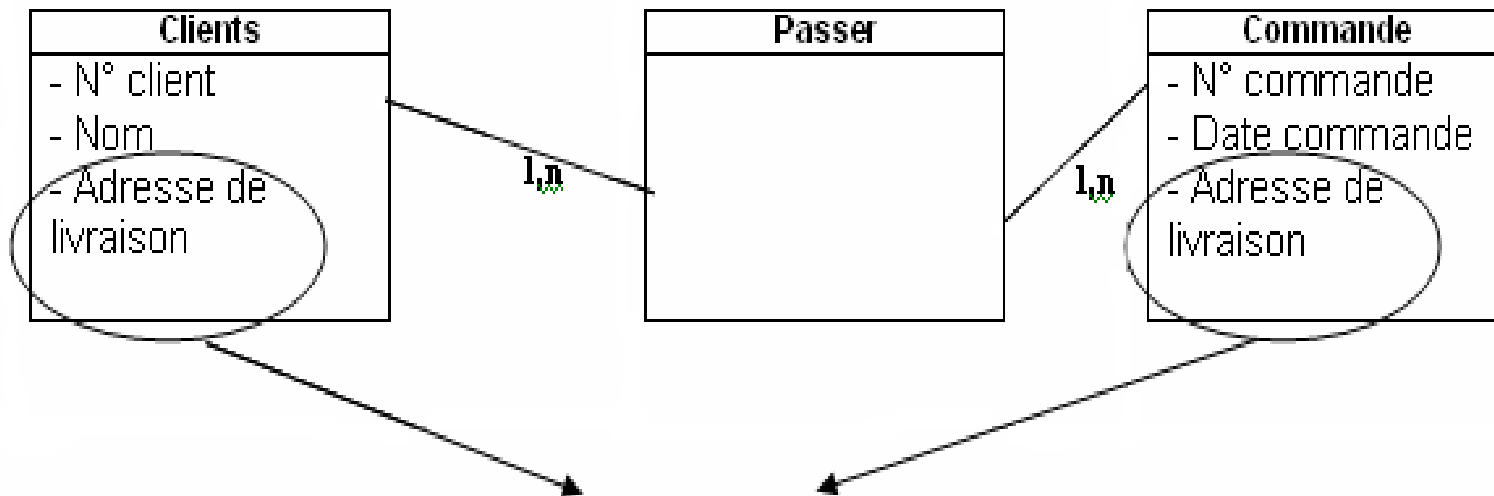
règles de modélisation (normalisation)

Objet : Rassembler les données homogènes et éviter les redondances.

Règ 1: *Normalisation des entités* : Toutes les entités qui sont remplaçables par une association doivent être remplacées.



Deux entités homogènes peuvent être fusionner



Redondance, donc risque d'incohérence

Les adresses peuvent ne pas être les même donc où va-t-on livrer ?

•

Règ 2: Normalisation des noms : le nom d'une entité, d'une association ou d'un attribut doit être unique.

Règ 3: Normalisation d'un identifiant : Chaque entité doit posséder un identifiant.

Règ 4: Normalisation des attributs et des associations: les attributs d'une association doivent dépendre directement des identifiants de toutes les entités en association et il faut éliminer les association superflues.

Règ 5. : Normalisation des cardinalités : une cardinalité minimale est toujours 0 ou 1 (pas 2, 3 ou n) et une cardinalité maximale est toujours 1 ou n (pas 2, 3,...).

Conception d'un MCD à partir d'un problème Réel

1- Identifier les entités en présence.

2- Lister leurs attributs.

3- Ajouter les identifiants.

4- Établir les associations entre les entités.

5- Lister leurs attributs.

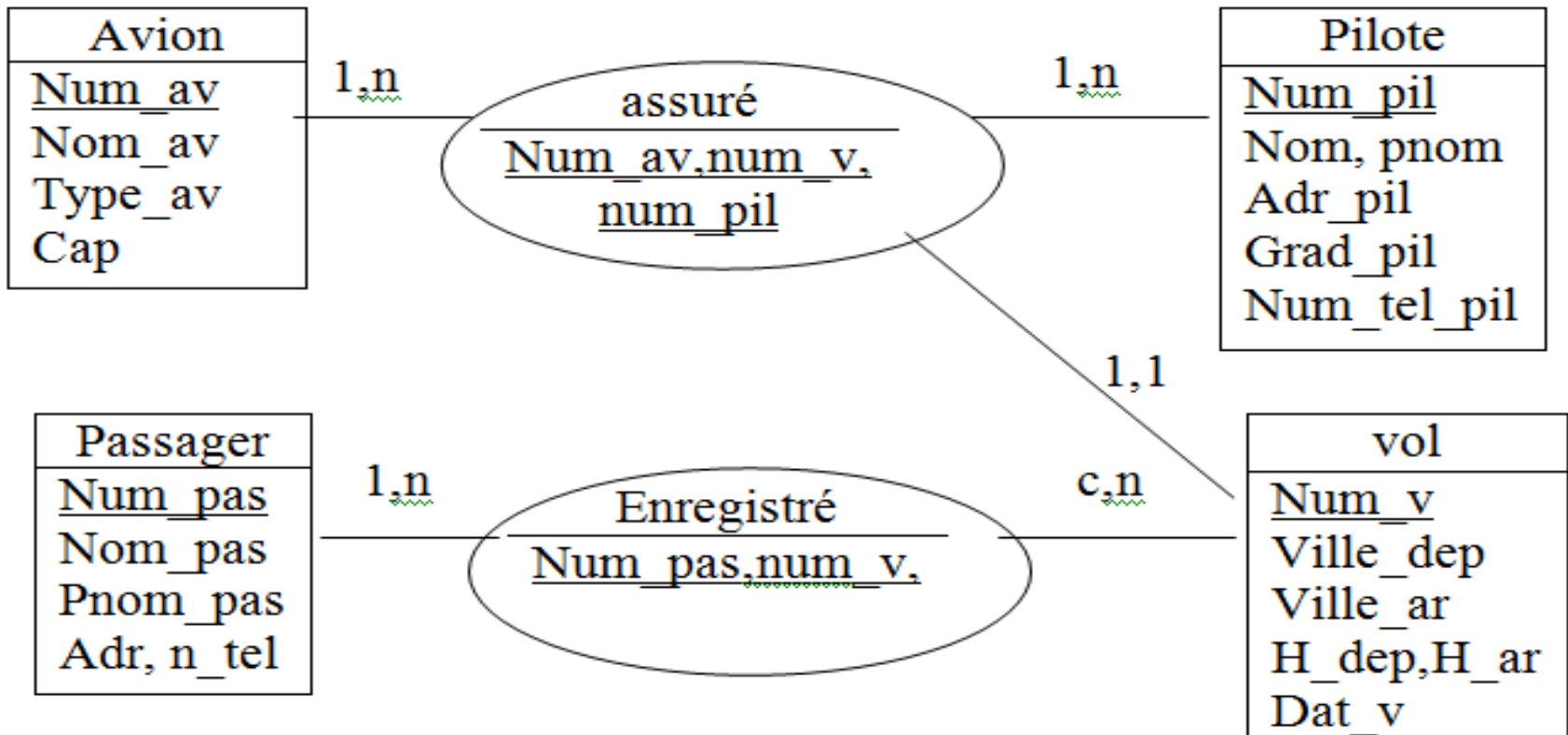
6- Calculer les cardinalités.

7- Vérifier les règles de normalisation.

Exemple 1 (1)

- Exemple: schéma conceptuel : cas d'une compagnie aérienne :
- Soit une compagnie aérienne décrite par les éléments suivants :
- Elle possède un parc d'avion.
- Des pilotes qui assurent un certain nombre de vols.
- Les avions sont identifiés par un numéro et sont caractérisés par un nom, un type et une capacité.
- Les pilotes sont identifiés par un numéro et sont caractérisés par un nom et un prénom, une adresse, un grade et un numéro de téléphone.
- Un vol est assuré par un avion et un pilote.
- Un vol est identifié par un numéro et caractérisé par une ville de départ, une ville d'arrivée, une heure de départ, une heure d'arrivée, et une date.
- Des passagers sont enregistrés sur un vol
- Un passager est identifié par un numéro et caractérisé par un nom et un prénom, une adresse et un numéro de téléphone.

Exemple 1 (2)

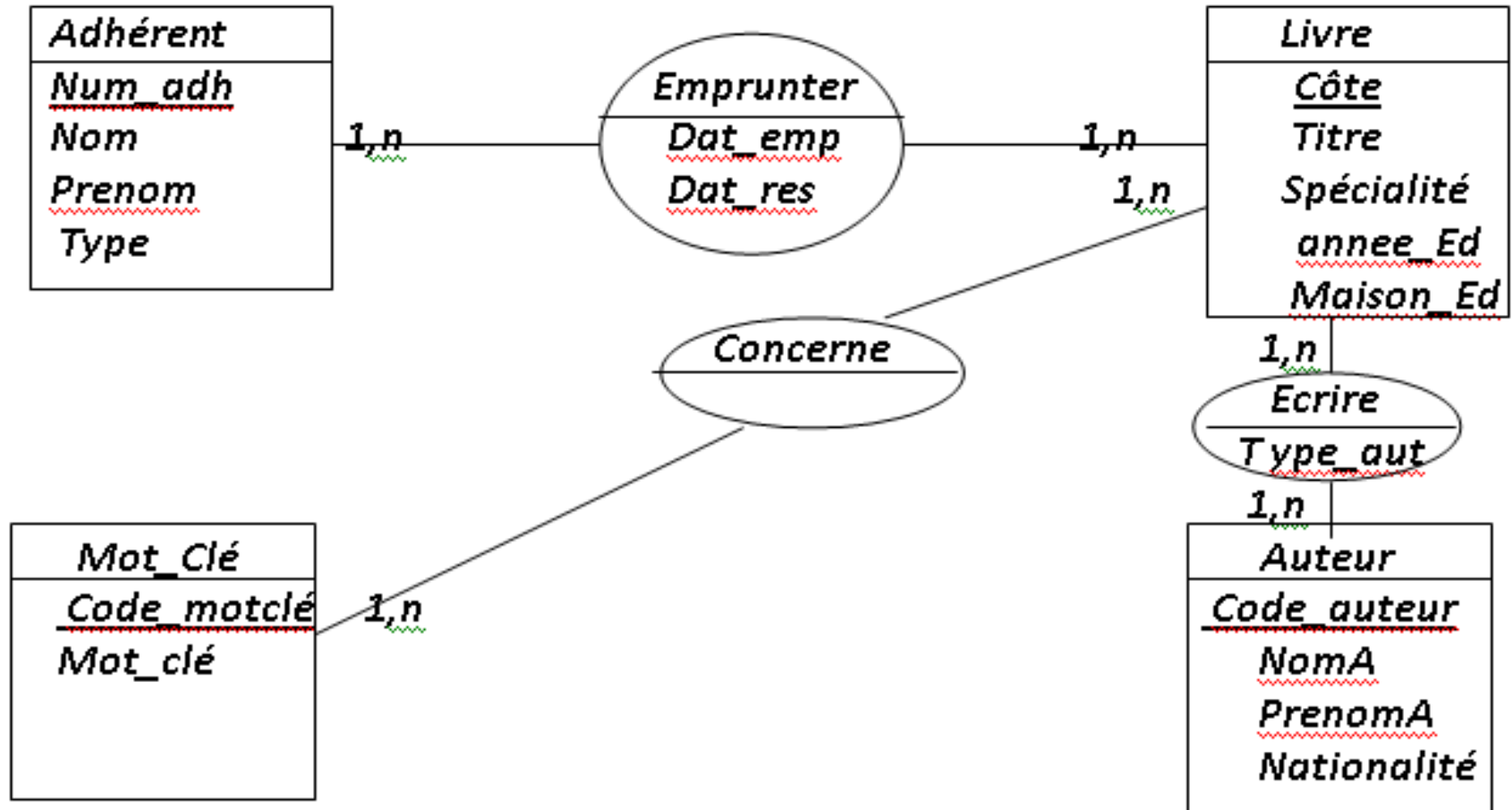


Remarque: pour simplifier ce schéma on peut décomposer la relation *assuré* en utilisant les cardinalités individuelles en deux relations : soit *assure1* et *assure2* leur collection sont (avion, vol) et (pilote, vol) respectivement

Exemple 2 (1)

- Notre établissement universitaire dispose d'une bibliothèque permettant à ses adhérents d'emprunter des livres.
- Un adhérent peut être un étudiant ou un enseignant, il est identifié par un numéro, possède un nom, un prénom et appartient à une catégorie.
- Un livre est identifié par un numéro de référence, possède un titre, une spécialité, une année d'édition et une maison d'édition.
- Un livre est écrit par un ou plusieurs auteurs dont un est principal.
- Un auteur est identifié par un numéro, un nom et un prénom et possède une nationalité.
- Un livre possède un ensemble de mots clés, un mot clé appartient à plusieurs livres et est identifié par un code.
- Un livre peut être emprunté par plusieurs adhérents dans le temps, à chaque opération d'emprunt on doit enregistrer la date et à chaque restitution on doit enregistrer la date.

Exemple 2 (2)



Le modèle relationnel

- **Introduction :**
- introduit au début des années 70, c'est le premier modèle de bases de données indépendant des critères de stockage,
- percevoir la base de données comme un ensemble de relations qui peuvent être représentées sous forme de table à deux dimensions : Les colonnes correspondent aux attributs d'une relation et les lignes correspondent aux tuples.
- utilise un seul concept qui est la relation, ce qui lui rend plus facile à utiliser même pour les utilisateurs ayant plus ou moins de connaissance en informatique.
- Ce modèle est basé essentiellement sur une théorie mathématique « la théorie des ensembles »

Définitions (1)

- **Domaine :**
 - Un domaine est un ensemble de valeurs caractérisé par un nom.
 - Exemple:
 - Domaine des booléens $D1 = \{0, 1\}$
 - Domaine des couleurs $D2 = \{\text{Blanche, Rouge, noire}\}$
 - Domaines des marques de voitures $D3 = \{\text{Peugeot, Fiat, Renault}\}$
 - Domaines des modules $D4 = \{\text{BD, RO, SI, SE, ANG}\}$

Définitions (2)

- ***Le produit cartésien :***
- Le produit cartésien d'un ensemble de domaines D_1, D_2, \dots, D_n noté $D_1 \times D_2 \times \dots \times D_n$ est l'ensemble des n-uplets «tuples» (U_1, U_2, \dots, U_n) tel que $U_i \in D_i$ pour tout $i=1, 2, \dots, n$
- Exemple: En prenant les domaines D2 et D3, le produit cartésien donne 9 couples :

D2	Blanche	Blanche	Blanche	Rouge	Rouge	Rouge	Noire	Noire	Noire
D3	Peugeot	Fiat	Renault	Peugeot	Fiat	Renault	Peugeot	Fiat	Renault

Définitions (3)

- **La relation :**
- Une relation est un sous-ensemble de produit cartésien d'un ensemble de domaine, une relation est désignée par un nom
- Exemple: Soient les domaines suivants:
 - $D1 = \{1, 2, 3\}$
 - $D2 = \{\text{janvier, février}\}$
 - $D3 = \{2002, 2003, 2004\}$
- On peut avoir une relation *date* définie sur les domaines D1, D2, et D3 avec les tuples suivants :

D1	D2	D3
1	Janvier	2004
2	Janvier	2004
2	Février	2003
3	Février	2002

Définitions (4)

- ***L'attribut :***
- Un attribut est une colonne de la relation identifié par un nom, c'est une variable qui prend valeur dans un des domaines sur lesquels la relation est définie.
- Exemple: Dans l'exemple précédent on peut avoir les attributs : jour défini sur le domaine D1, mois défini sur le domaine D2 et année défini sur le domaine D3, d'où la relation *date* devient :

Jour	Mois	Année
1	Janvier	2004
2	Janvier	2004
2	Février	2003
3	Février	2002

Définitions (5)

- ***Le tuple:***
- Un tuple d'une relation désigne une ligne dans la table représentant cette relation.
- Exemple: Dans la relation *date* (1, janvier, 2005) est un tuple
- ***6. Arité d'une relation :***
- L'arité d'une relation est le nombre de ses attributs, c'est aussi le nombre de colonnes de la table si on représente la relation sous forme de table.
- Exemple: L'arité de la relation *date* est égale à 3.

Définitions (6)

- ***Cardinalité d'une relation :***
- La cardinalité d'une relation c'est le nombre de tuples de cette relation, c'est aussi le nombre de lignes de la table qui représente la relation. La cardinalité d'une relation R est un nombre entier noté par $|R|$
- Exemple: la cardinalité de la relation date : $|date| = 4$;
- ***Schéma d'une relation :***
- Un schéma d'une relation est composé du nom de la relation suivi de la liste des attributs avec leurs domaines,
- $R(A1 : D1, A2 : D2, \dots, An : Dn) ;$
- Exemple: le schéma de la relation Etudiant :
- Etudiant(numinsc: integer; nom: char(20); prenom: char(20); adr: char(40) ; depart: dep); où : dep : est le domaine des départements {inf, mat, bio, elec} ;

Définitions (7)

- ***Extension de la relation :***
- Une extension de la relation est un ensemble de tuples de la relation, elle est représentée par un tableau à 2 dimensions où une ligne correspond à un tuple et une colonne à un attribut de la relation
- Exemple: Soit la relation produit, une extension de cette relation peut être :

Cod_prod	Désignation	Prix_unit
A002	Armoire	5400
C004	Chaise	1600
E001	Etager	2500

Définitions (8)

- ***Clé de la relation :***
- La clé d'une relation R est un attribut ou un ensemble d'attributs qui permet d'identifier d'une manière unique chaque tuple de la relation.
- Remarque: D'après la définition même d'une relation qu'elle est un ensemble de tuples n'ayant pas d'éléments en double (ne peut pas contenir deux tuples identiques), toute relation possède au moins une clé à savoir : l'ensemble de ses attributs.
- **Clé candidate:** un attribut ou un ensemble d'attributs est appelé clé candidate s'il peut jouer le rôle de clé dans une relation, une clé doit être soulignée dans un schéma de relation.
- **Clé primaire:** lorsqu'on dispose pour une relation donnée de plusieurs clés candidates, il est nécessaire de ne retenir qu'une seule parmi l'ensemble des clés candidates et c'est celle qu'on utilisera effectivement pour repérer de manière unique les tuples de la relation, la clé retenue s'appellera alors clé primaire.

Définitions (9)

- Le choix de la clé primaire est généralement effectué en fonction des deux critères suivants :
 - On choisit la clé candidate ayant le plus petit nombre d'attributs : il est préférable de minimiser le nombre d'attributs manipulés.
 - On choisit la clé candidate en fonction de son usage pour la localisation des tuples : il s'agit de privilégier la clé candidate dont l'usage serait le plus fréquent pour localiser les tuples de la relation
- Remarque: toute clé candidate qui n'a pas été retenue comme clé primaire constitue une clé secondaire.

Définitions (10)

- ***Base de données relationnelles:***
- C'est une base de données dont le schéma conceptuel est un ensemble de schémas de relations et dont les occurrences sont des tuples de ces relations.
- Exemple:
- Etudiant(num_insc, nom, prénom, adr, département)
- Module(codmod, intitule, coef)
- Inscrit(num_insc, codmod)
- Examen(num_insc, codmod, note)

Le passage d'un schéma conceptuel E-A vers une base de données relationnelle

- Avec le modèle relationnel, le schéma conceptuel de la base de données est représenté sous forme de table. Une telle représentation ne permet pas de distinguer facilement une entité d'une relation, ni de servir de support de communication si on a à discuter le schéma conceptuel avec d'autres personnes. Généralement on préfère utiliser un autre formalisme tel que celui du modèle entité-association, ce dernier nous offre un aide au niveau de la phase de modélisation même si cela demande un travail supplémentaire de transformation de ce modèle vers le modèle relationnel.
- Pour effectuer ce passage on applique des règles pour les individus et les relations.

Les règles de passage

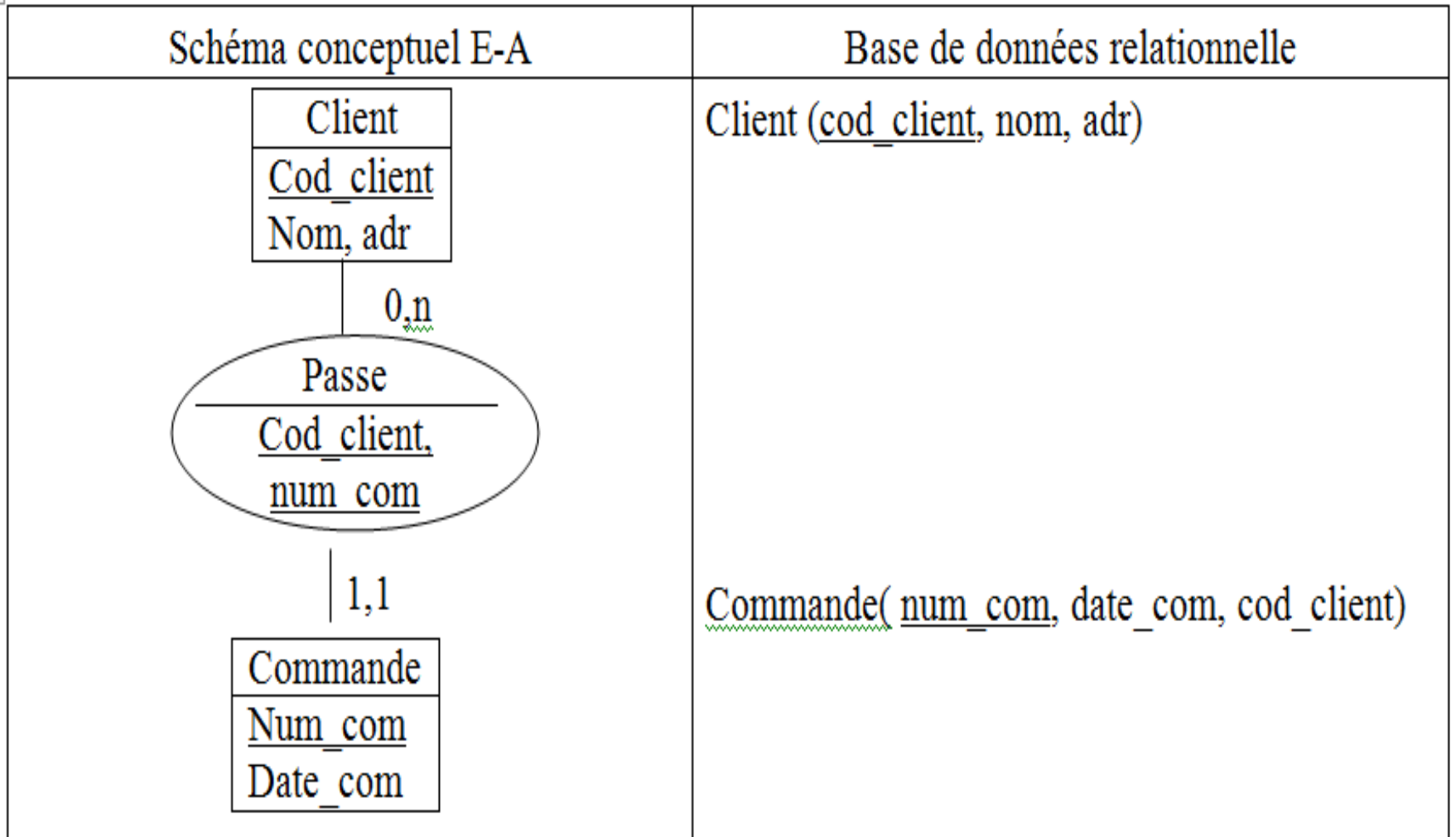
- ***Pour les individus :***
 - Chaque individu se transforme en une table.
 - L'identifiant de l'individu devient la clé primaire de la table.
 - Les propriétés de l'individu deviennent des attributs dans la table.
- ***Pour les relations :***
- Cas des relations père-fils $(x, n - y, 1)$:
- L'individu père devient la table père.
- L'individu fils devient la table fils.
- L'identifiant de l'individu père devient un attribut dans la table fils.
- Les propriétés de la relation deviennent des attributs dans la table fils.

Les règles de passage

- Cas des autres relations :
 - Chaque individu devient une table.
 - L'identifiant de l'individu devient la clé primaire de la table.
 - La relation devient une table.
 - L'identifiant de la relation devient la clé primaire dans la table.
 - Les propriétés de la relation deviennent des attributs dans la table.

Exemples (1)

1^{er} cas :



Exemples (2)

2ème cas :

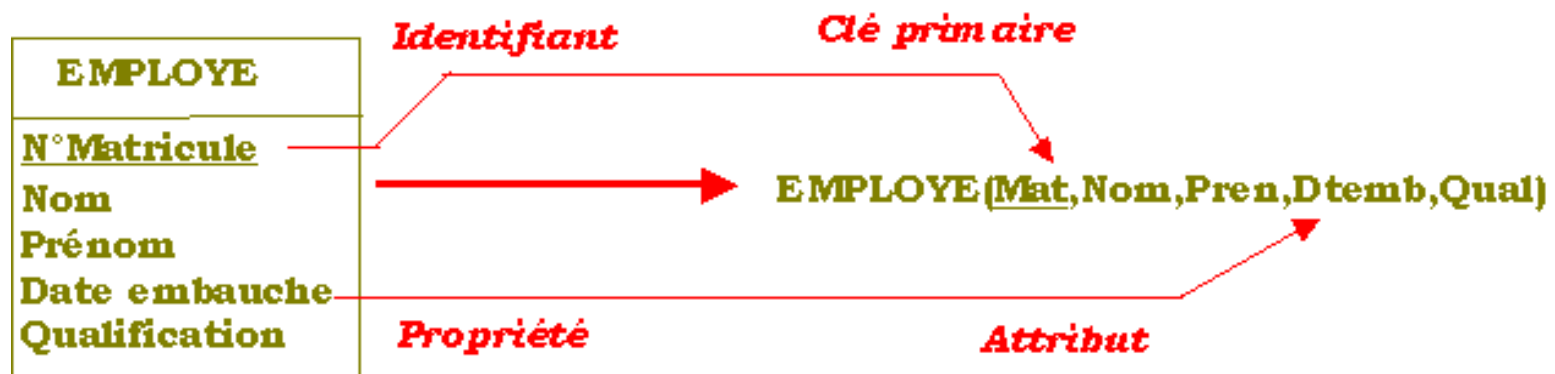
Schéma conceptuel E-A	Base de données relationnelle
<pre>graph TD Fournisseur[Fournisseur] --- 1,n fournir((fournir)) fournir --- 1,n Produit[Produit]</pre> <p>The diagram shows an Entity-Relationship model. At the top is the entity 'Fournisseur' with a primary key 'Cod_f' and attributes 'Nom, adr'. Below it is the relationship 'fournir' (represented by an oval) with primary keys 'Cod_f' and 'cod_p'. At the bottom is the entity 'Produit' with a primary key 'Cod_p' and attributes 'Désign, Pu'. Lines connect 'Fournisseur' to 'fournir' and 'fournir' to 'Produit', both labeled with '1,n' cardinalities.</p>	<p>Fournisseur (<u>cod_f</u>, nom, adr)</p> <p>Fournir (<u>cod_f</u>, cod_p)</p> <p>Produit (<u>cod_p</u>, désign, Pu)</p>

Questions

-

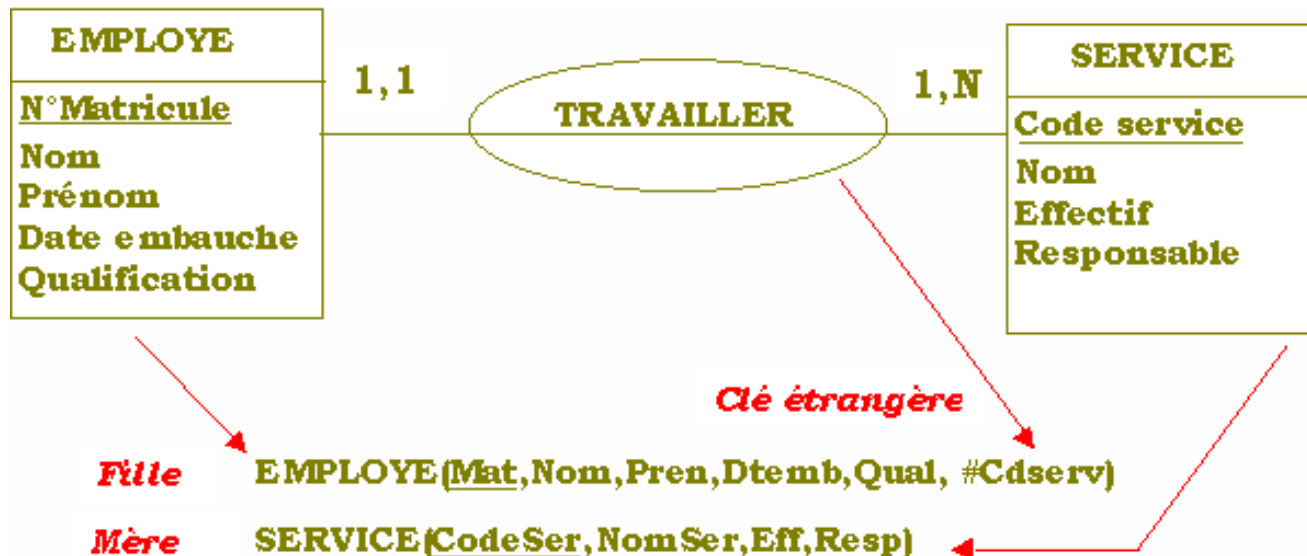
Règles de passage

- REGLE N°1 : TOUTE ENTITE DEVIENT UNE TABLE dans laquelle :
 - les attributs traduisent les propriétés de l'entité
 - la clé primaire traduit l'identifiant de l'entité



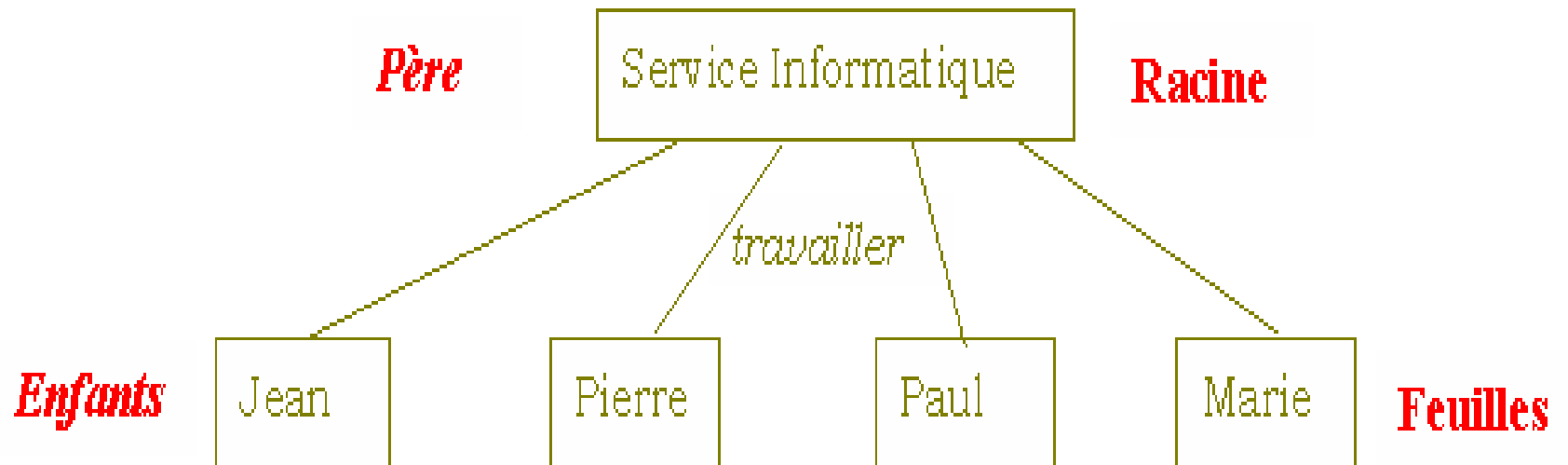
Règles de passage

- REGLE N°2 : UNE ASSOCIATION DE DIMENSION 2 AVEC CARDINALITE 1,1 SE REECRIT EN :
portant dans la relation fille la clé primaire de la relation mère.
L'attribut ainsi ajouté s'appelle clé étrangère. Symbole : #.



Règles de passage

- Relation fille

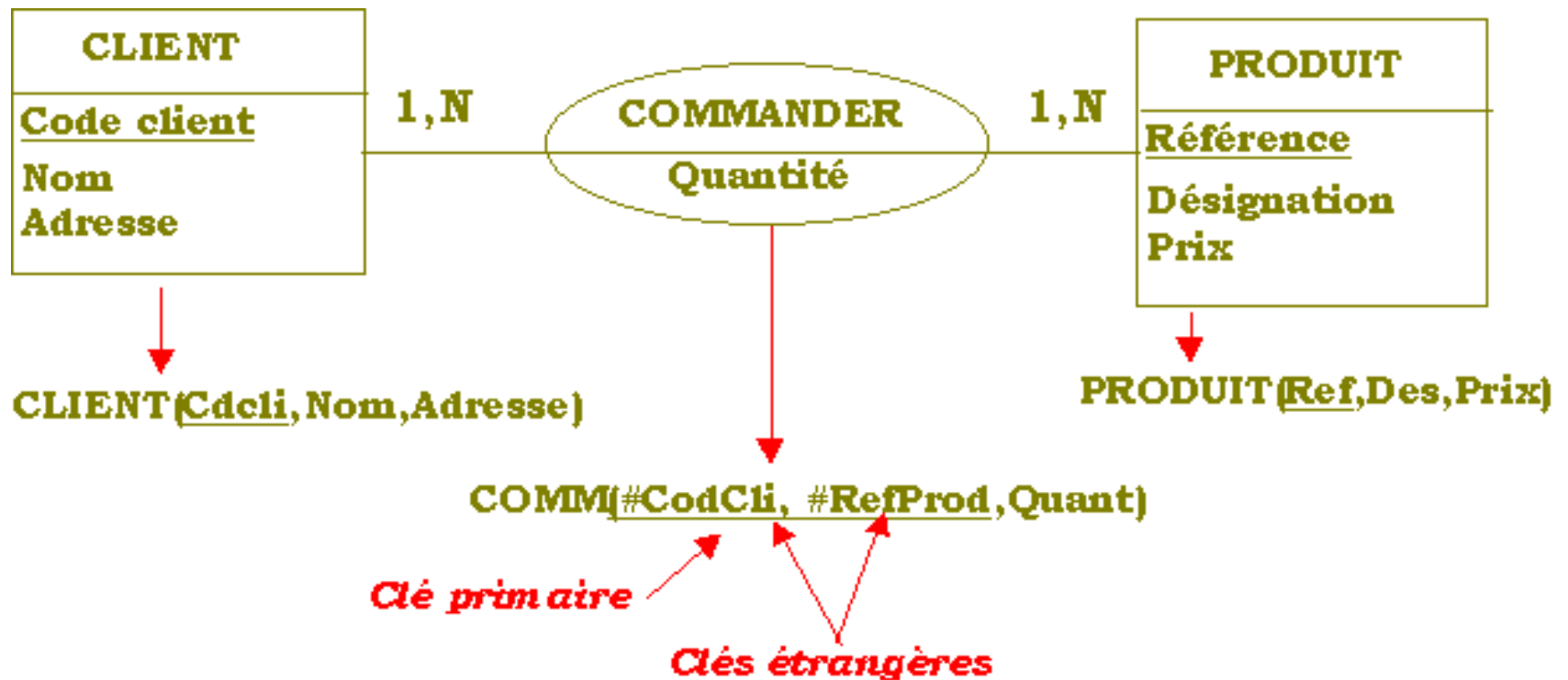


-

Règles de passage

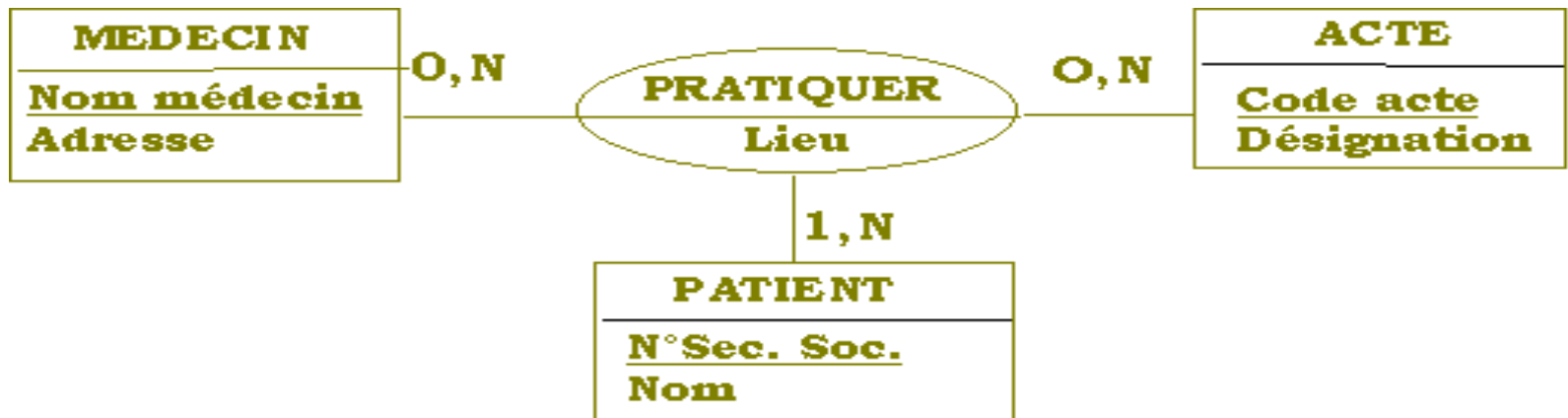
- REGLE N°3 : UNE ASSOCIATION DE DIMENSION 2 AVEC CARDINALITE PLUSIEURS A PLUSIEURS SE REECRIT EN :
 - créant une relation particulière qui contient comme attributs les identifiants des 2 entités associées
 - ces attributs constituent à eux 2 la clé primaire de la relation
 - ils sont individuellement clés étrangères
 - ajoutant la ou les éventuelles propriétés de l'association à cette relation.

Règles de passage



Règles de passage

- REGLE N°4 : UNE ASSOCIATION DE DIMENSION SUPERIEURE A 2 SE REECRIT SELON LA REGLE 3



↓

MEDECIN(NomMed, Adr)
ACTE(CdActe, Des)
PATIENT(SS, NomPat)
PRATIQUER(#Med, #Acte, #Secu, Lieu)

Règles de passage

- Révision