



**Université de Sciences et de la technologie Houari  
Boumediene- Faculté d'Electronique et d'Informatique  
Département d'Informatique**

**Licence Informatique Générale**

**Cours Introduction aux Systèmes d'Information**

**Enseignant responsable : M.AZZOUZ**

**Cours 3 : Conception des Systèmes d'Information**

# Problèmes de conception d'un SI sans méthode

- Vision parcellaire des besoins :
  - ▶ Redondance des besoins
  - ▶ Synonymes (plusieurs termes qui ont le même sens)
  - ▶ Polysèmes (un terme qui a plusieurs sens différents)
- Difficulté d'une maintenance efficace :
  - ▶ Moyens techniques limités
  - ▶ Manqué de dossiers d'analyse
- Utilisateurs non intégré dans la conception du SI
- Difficulté de planification et de suivi du travail



**Apparition des méthodes de conduite des projets**

# Méthode(I)

- Selon le petit robert, une méthode est un ensemble de démarches raisonnées suivies pour parvenir à un but.
- L'objectif des méthodes de conception et de développement de SI est de représenter une démarche et un ensemble de modèles permettant de mettre en place un nouveau système.
- Exemple : Merise, Axial, E/A Chen, OMT,....

# Méthode(2)

- Quelque soit la méthode choisie celle-ci s'articule autour des étapes suivantes :



# Conception

- Création d'un objet, d'un système : action qui donne naissance à quelque chose qui n'existait pas.
- L'analyse et la conception des systèmes d'information automatisés (CSI, ACSI, ACSIA, CSIA) consistent en un ensemble de techniques et de méthodes destinés à améliorer l'efficacité du SI en fonction des objectifs des organisations correspondantes.

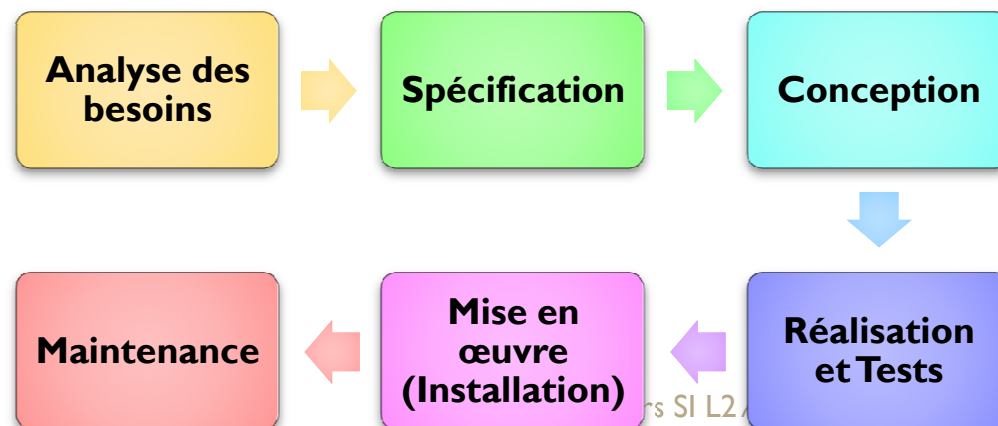
# Bases de données et méthode MERISE(I)

## **Bases de données**

- Une Base de Données (BD) est un ensemble structuré de données ayant un sens et accessibles par l'ordinateur pour satisfaire simultanément plusieurs utilisateurs.
- Une BD est conçue, construite et remplie avec des données dans un but précis

# La méthode Merise – approche générale

- Elle permet de faire un lien de communication entre les différents acteurs d'un projet.
- Merise propose une double démarche : par niveau d'abstraction (conceptuel, organisationnel, physique ou opérationnel) et par étapes (études préalables, détaillées, fonctionnelles, techniques et mise en production)
- ***Cycle de vie d'un projet :***



# Exemple (I)

Mise en place d'un système d'information pour gérer toutes les données nécessaires au bon fonctionnement d'une entreprise de livraison.

## ***Problèmes dans le système actuel :***

- Livraison à mauvaise adresse.
- Courrier en copies multiples.
- Obligation de rappeler à chaque communication téléphonique le nom, le prénom, l'adresse, etc.



## Exemple (2)

### **Remarque :**

- Les mêmes données se présentent avec des libellés différents (synonymes).
- Les mêmes données sont reproduites plusieurs fois dans des services différents (redondance).
- Une même donnée prend parfois des valeurs différentes. Exemple : adresse facturation dans le service commercial et dans le service comptable (polysème).
- Un service n'a pas toujours l'ensemble des données.
- Besoin de mémoriser et de traiter des données de quantité importante.

# Exemple (3)

## ***Suggestion***

- Créer une fiche unique par client qui rassemble toutes les données accessibles par tous les services.
- Gérer toutes les fiches client sur un micro-ordinateur (en utilisant Access, Oracle...).

## ***Questions***

- Quelles sont les données à mémoriser?
- Comment minimiser les données redondantes?
- Comment structurer les données et conserver des liens entre données?
- Comment décrire les traitements sur les données?

# Exemple (4)

## ***Éléments de réponse***

- Méthode de conception d'une Base de Données, exemple : Méthode MERISE.
- Utilisation d'un Système de Gestion de Bases de Données Relationnelles (SGBD-R) exemple : Access, Oracle.

# Conception d'un système d'information

- La conception d'un SI consiste en
  - une modélisation de l'entreprise (pour une vision globale : modèle systémique),
  - une modélisation de ses aspects statiques (données : entités et association inter-entités);
  - une modélisation de ses aspects dynamiques (traitements : opérations déclenchées par des événements).
- Cette conception nécessite une approche progressive (niveaux d'abstraction).

# Niveaux d'abstraction(I)

Un SI doit être durable et adaptable aux changements.

- Par exemple, un changement de machine ne doit pas entraîner une modification majeure du système.
- Il est donc nécessaire de dégager des niveaux correspondant aux préoccupations différentes.

# Niveaux d'abstraction(2)

## Niveau conceptuel

- A ce niveau, sont représentées
  - les informations et leurs relations,
  - les utilisations qui en sont faites et les contraintes associées.
- Ce niveau définit les finalités de l'entreprise, c.-à-d. la réponse à la question « que fait le système? »
- C'est le niveau **le plus stable**, il est invariant.
  - ➔ Recueillir, organiser et structurer l'information
  - ➔ Modèle Conceptuel de Données et de Traitements « MCD et MCT »

# Niveaux d'abstraction(3)

## Niveau organisationnel (ou logique)

- Ce niveau est **moins stable**.
- Il définit l'organisation à mettre en place dans l'entreprise : il répond aux questions «comment?», « qui fera quoi, où et quand? »
  - ➔ Modèle Logique de Donnée « MLD »
  - ➔ Modèle Organisationnel de Traitement « MOT »

# Niveaux d'abstraction(4)

## Niveau opérationnel (technique ou physique)


- Ce niveau est **le moins stable**.
- Il est souvent mis en cause par le changement du matériel et/ou des logiciels.
- Il définit l'ensemble des moyens techniques, composé de machines, de programmes et de fichiers, pour répondre aux objectifs posés.

➔ Modèle Physique de Donnée « MPD »

➔ Modèle Physique de Traitement « MPT »



# Niveaux d'abstraction(5)



Niveau	Données	Traitements
Conceptuel	MCD	MCT
Logique et Organisationnel	MLD	MOT
Physique (opérationnel ou technique)	MPD	MPT

Analyse **descendante** : information globale → information élémentaire

## Remarque :

Différencier données et traitements

- Donnée : entité = objet, individu du système; association = lien, relation entre ces entités; etc ...
- Traitement : opération déclenchée par un évènement
- Evènement : fait survenant



# **Le modèle conceptuel des données MCD**

## **La problématique des données**

- Il ne suffit pas de s'intéresser au nom et aux propriétés élémentaires (type, longueur, valeurs) des données.
- Il faut s'intéresser à la donnée elle-même, ses sens et ses usages.
- Les acteurs peuvent utiliser les mêmes mots avec des sens ou des contenus différents (synonymes, polysèmes).

## **Objectif du MCD**

- Décrire les données du SI, indépendamment de tout choix d'implantation physique.

# Dictionnaire des données(I)

- Inventaire des données du domaine étudié.
- Le dictionnaire des données est un tableau qui regroupe toutes les données du SI, pour chaque donnée il faut préciser : Code de la donnée, désignation, type, taille (longueur), une observation si cela est nécessaire.
- Le dictionnaire des données est un outil nécessaire pour la construction de MCD.

## Dictionnaire des données(2)

- **Exemple:** Prenant l'exemple du SI lié au service de vente. Nous allons extraire toutes les données contenues dans les deux documents commande et facture.

### Commande

N° commande : .....

Date commande : .....

N° client : .....

Nom client : .....

Adresse client : .....

Ref	Désignation	PU	Quantité

### Facture

N° facture : .....

Code commande : .....

Date facture : .....

Matricule client : .....

Ref	Désignation	PU	Quantité	Montant

Montant total :

# Dictionnaire des données(3)

## Dictionnaire des données brut

Code de la donnée	Désignation	Type	Taille	Observation
Num-C	Numéro de la commande	N	4	JJ/MM/AAAA
Date-C	Date de la commande	Date	10	
Num-CL	Numéro client	N	4	
Nom-CL	Nom client	A	15	
Adr-CL	Adresse client	A N( c)	40	
Ref	Référence produit	C	10	
Désignation	Désignation produit	C	20	
PU	Prix unitaire produit	N	8	
Quantité	Quantité produit	C	6	
Num-F	Numéro facture	N	4	
Date-F	Date facture	D	10	
Code-C	Code commande	N	4	
Mat-C	Matricule client	N	4	
Ref	Référence produit	C	10	
Desgn	Désignation produit	C	20	
PU-P	Prix unitaire produit	C	6	
Quantité	Quantité produit	N		
Montant	Montant produit	N		
Mont-T	Montant total	N		

# Dictionnaire des données(4)

## Epuration de dictionnaire des données

- Certain données doivent être éliminées de ce dictionnaire.
- **Synonymes** : des propriétés qui désigne la même donnée :
  - Les donnée Num-C et Code-C désigne la même donnée « Numéro de la commande ».
  - Num-CL et Mat-CL désigne la même donnée : « Numéro client ».
  - Il faut éliminer tous les synonymes et ne garder qu'une seule donnée, dans ce cas on garde que Num-C et Mat-CL.



# Dictionnaire des données(5)

## Epuration de dictionnaire des données

- Polysémie : La donnée qte désigne une quantité mais il y a deux quantité différentes :
- Quantité-C et Quantité-F. Qte est une donnée qui a deux sens, il faut le supprimer et créer deux données Qte-fac et Qte-com.
- Valeur calculée : Les données calculées doivent être supprimées du dictionnaire des données.  
(Mont-Total)
- Les données concaténées : La donnée Adr-CL est une donnée (composée) qu'il faut décomposer comme suit : Ville-CL et Rue-CL.

# Dictionnaire des données(6)

## Dictionnaire des données épuré

Code de la donnée	Désignation	Type	Taille	Observation
Num-C Date-C Nom-CL Rue-CL Ville-Cl Réf-Prod Des-Prod PU-p Num-F Date-F Num-C Qte-C Qte-f				



# **Le modèle conceptuel des données : le modèle entité/association**

- Le modèle E/A a été élaboré par CHEN en 1976 pour la modélisation des données et des liens existants entre elles avec des concepts simples et efficaces, c'est une représentation naturelle du monde réel du SI à étudier.
  - Concepts de base du modèle E/A.
  - Vérification et normalisation du modèle E/A.

## Concepts de base du modèle E/A (I)

- **Entité** : tout objet concret ou abstrait ayant une existence propre et conforme aux besoins de gestion de l'organisation.

Exemple : le client «Ahmed», le produit de référence «a456»...

- **Classe d'entités (ou entité-type)** : ensemble des entités décrites par les mêmes caractéristiques.

Exemple : la classe CLIENT dont «Ahmed» est une **occurrence** (ou instance).

## Concepts de base du modèle E/A (2)

- **Association:** n-uplet d'entités «sémantiquement liées».

Exemple: («Ahmed», «I4523 I I8 I6») indiquant que la personne Ahmed est propriétaire de la voiture immatriculée I4523 I I8 I6.

- **Classe d'associations (ou association-type) :** regroupe toutes les associations constituées des mêmes types d'entités jouant le même rôle dans l'association.

Exemple: PROPRIETAIRE (PERSONNE, VOITURE)

- Les occurrences de cette classe d'association sont un sous ensemble du produit cartésien.

## Concepts de base du modèle E/A (3)

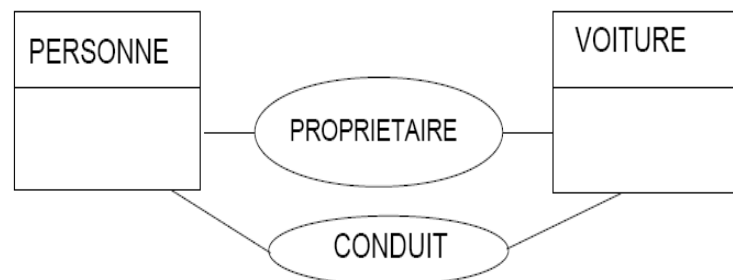
- **PERSONNE x VOITURE** (c.à.d. une partie de l'ensemble des couples possibles de personnes et de voitures).



- **Remarque**

On peut avoir plusieurs classes d'associations sur les mêmes classes d'entités.

Exemple : PROPRIETAIRE (PERSONNE, VOITURE) et CONDUIRE (PERSONNE, VOITURE)

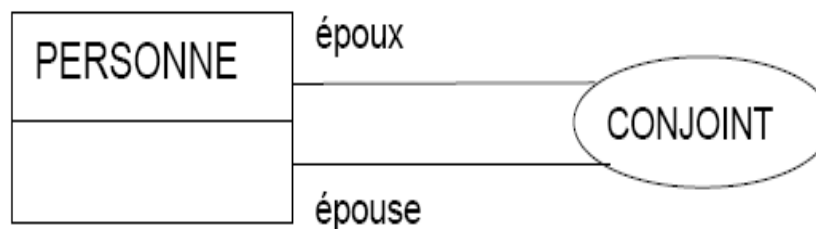


# Concepts de base du modèle E/A (4)

## Remarque (suite)

- On peut avoir une classe d'association sur une seule classe d'entités (on parle d'association «Réflexive »). On ajoute souvent dans ce cas des noms de rôles pour distinguer les deux occurrences.

Exemple : CONJOINT (PERSONNE, PERSONNE)

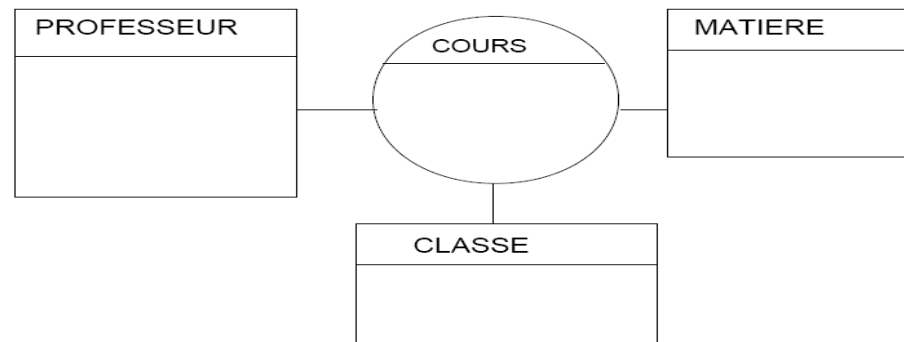


# Concepts de base du modèle E/A (5)

## Remarque (suite)

- On peut avoir une classe d'association définie sur n classes d'entités (association n-aire ou d'arité n ou de dimension n ou à « n pattes »).

Exemple: COURS (MATIERE, CLASSE, PROFESSEUR)



- **Attention** : les arités élevées sont rares. Elles dénotent souvent des faiblesses dans l'analyse.

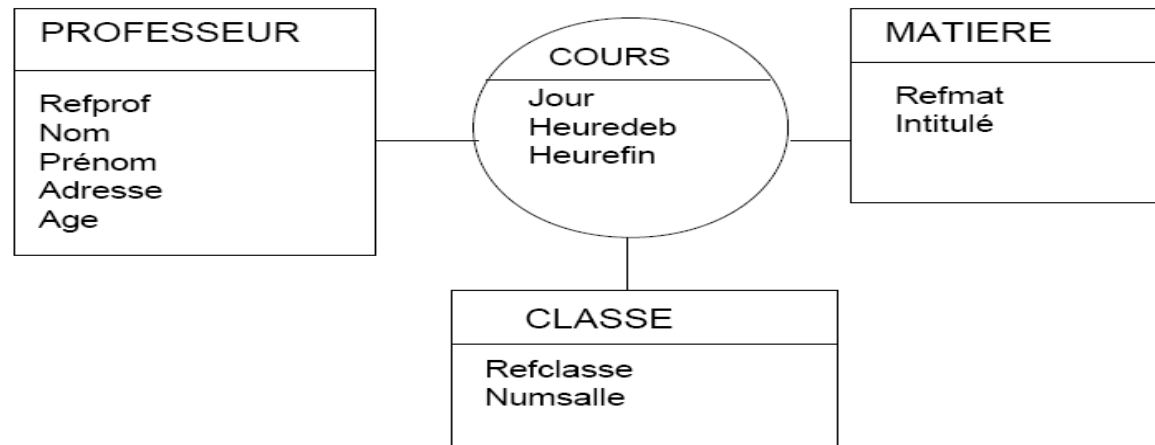
arité 2 : 80%; arité 3 : <20%; arité > 3 : très peu.

## Concepts de base du modèle E/A (6)

- **Propriété** : donnée élémentaire permettant de caractériser les entités et associations.

Exemple:

- Nom, Prénom propriétés de PROFESSEUR
- Jour, Heuredéb propriétés de COURS





## Concepts de base du modèle E/A (7)

- **Identifiant** : propriété ou groupe de propriétés permettant d'identifier de manière unique chaque occurrence de la classe d'entités.

Exemple : N° immatriculation pour VOITURE.  
Nom ne suffit pas pour PERSONNE. N° Client pour CLIENT (propriété ajoutée)

- Les identifiants sont en général soulignés.



## Concepts de base du modèle E/A (8)

- **Cardinalités** : indiquent pour chaque classe d'entités de la classe d'association, les nombres mini et maxi d'occurrences de l'association pouvant exister pour une occurrence de l'entité.

La cardinalité **minimum** est 0 ou 1.

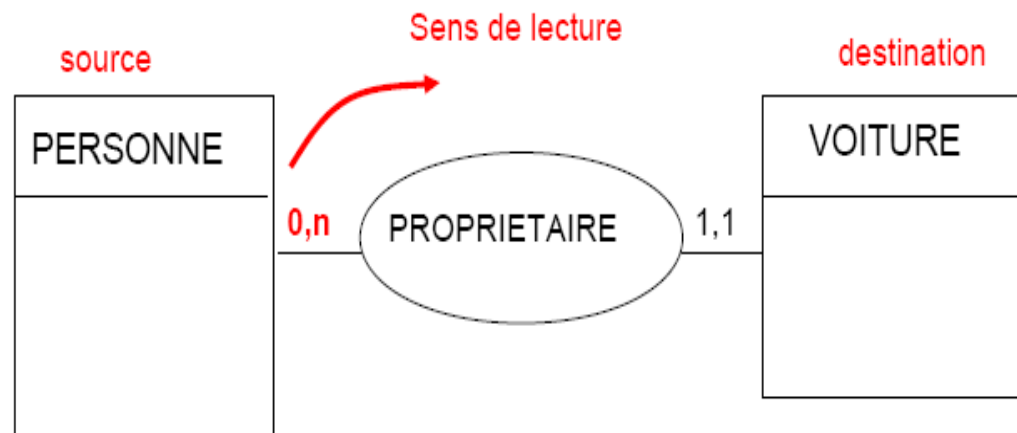
La cardinalité **maximum** est 1 ou n.

Une cardinalité minimum à 0 signifie qu'il est possible d'observer (un jour) une occurrence d'entité sans occurrence d'association.

- Donc 4 combinaisons possibles :
  - ✓ 0,1 : au plus 1
  - ✓ 1,1 : 1 et 1 seul
  - ✓ 1,n : au moins 1
  - ✓ 0,n : un nombre quelconque

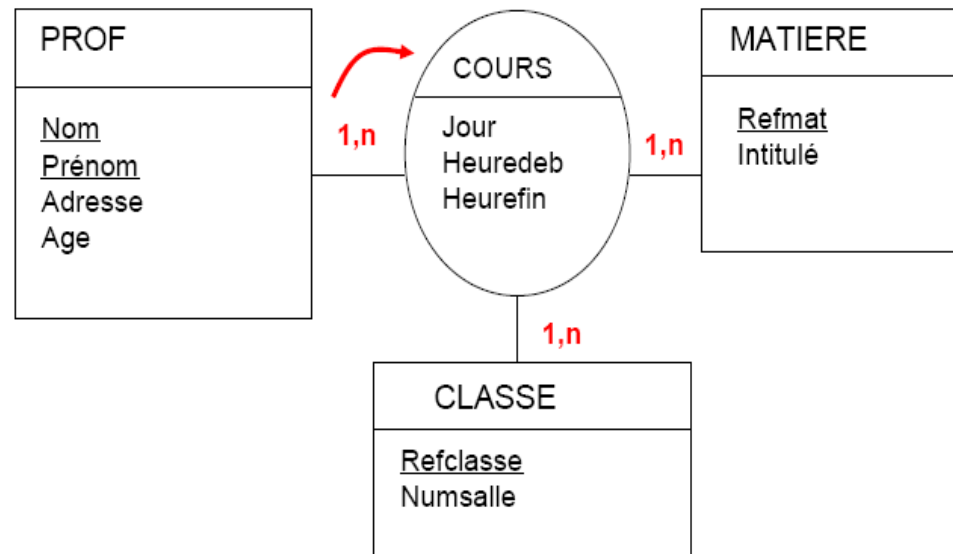
# Concepts de base du modèle E/A (9)

- Exemple: PROPRIETAIRE (PERSONNE [0,n], VOITURE [1,1])
- Une personne a 0 à n voitures; une voiture a 1 et 1 seul propriétaire.
- **Représentation graphique :**



# Concepts de base du modèle E/A (10)

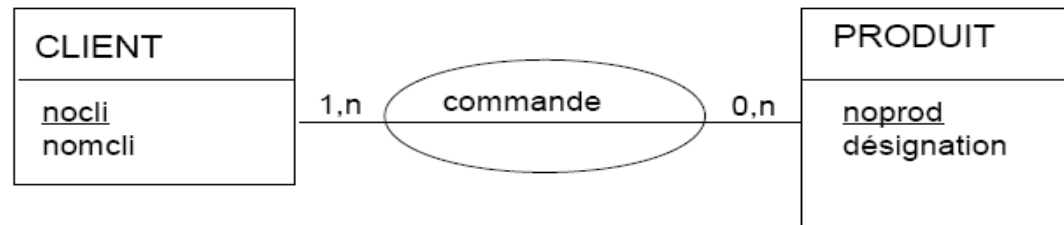
- Exemple: ENSEIGNER (MATIERE [1,n], CLASSE [1,n], PROF[1,n])
- Un prof. enseigne 1 à n cours, une matière est enseignée dans 1 à n cours, une classe a 1 à n cours .
- **Représentation graphique :**



# Concepts de base du modèle E/A (II)

## Difficultés: choix entre entité et association ?

- *Solution avec association*

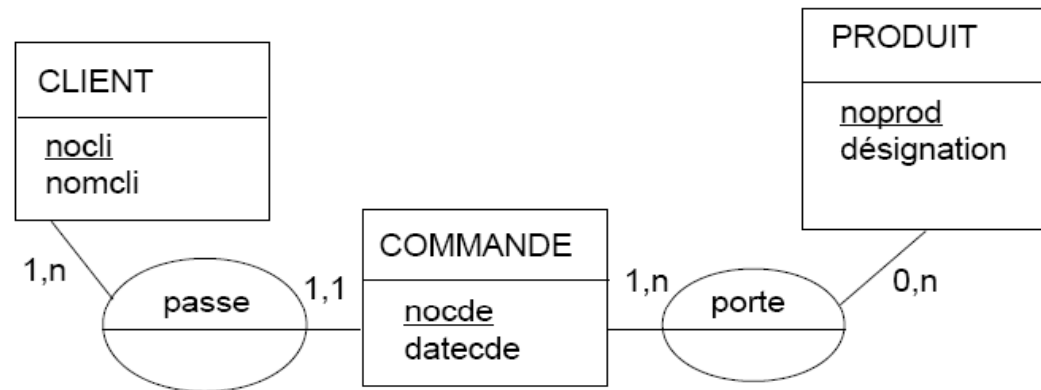


- Dans cette première solution la commande n'est pas une entité gérée pour elle même. Elle existe tant que le client et le produit existent.
- Ce peut être le SI du domaine 'fabrication' : on a juste besoin de savoir que les produits sont destinés à des clients.

# Concepts de base du modèle E/A (I2)

## Difficultés: choix entre entité et association ?

- *Solution avec entité*



- Dans cette seconde solution, les commandes sont identifiées (identifiant nocde) et décrites : on les gère en tant que telles.
- Elles peuvent être conservées même si le produit ou le client n'existent plus.
- Ce peut être le SI du domaine financier.

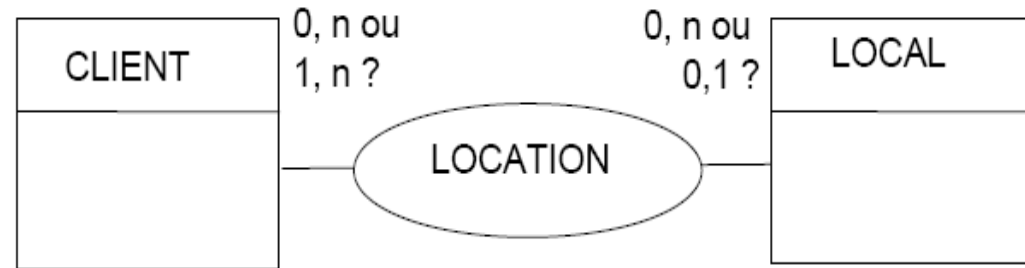
# Concepts de base du modèle E/A (I3)

## Difficultés: choix entre entité et association ?

- **Quelques « critères » de choix**
  - Une entité a une existence propre et un identifiant.
  - Une association n'existe que si ses extrémités existent et n'a pas d'identifiant explicite.
  - Une entité peut être associée à d'autres entités, une association non.

# Concepts de base du modèle E/A (14)

## Difficultés : choix des cardinalités ?



- Un client peut il avoir 0 location ? Est-ce encore un client ?
- Un local peut il être loué plusieurs fois ? Non si la base représente une situation instantanée et si le local n'est pas partageable. Oui si on gère un historique ou si le local est partageable.

# Concepts de base du modèle E/A (15)

## Difficultés : choix des cardinalités ?

- **Les cardinalités** sont élément essentiel pour définir la sémantique (signification) des données, pas une « décoration » accessoire. Derrière cette notion on trouvera des contrôles (par le SGBD ou les programmes).
- **Remarque :**
- Pour une situation donnée, il n'existe pas une «solution» unique.
- Le « bon modèle » est celui qui est accepté par les personnes concernées par le projet.

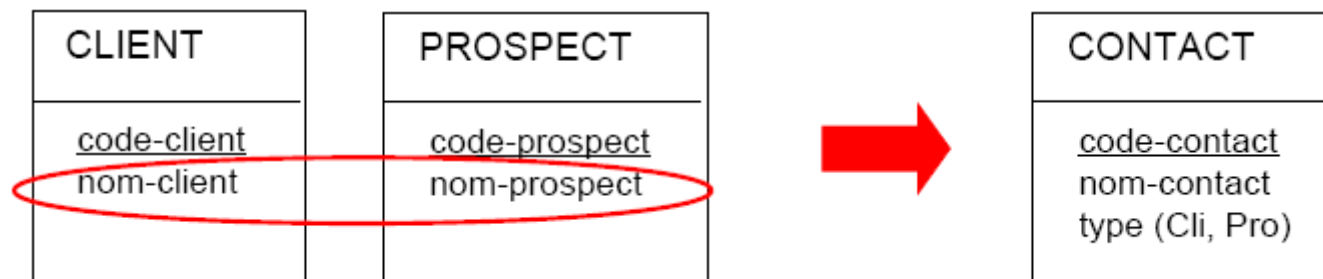
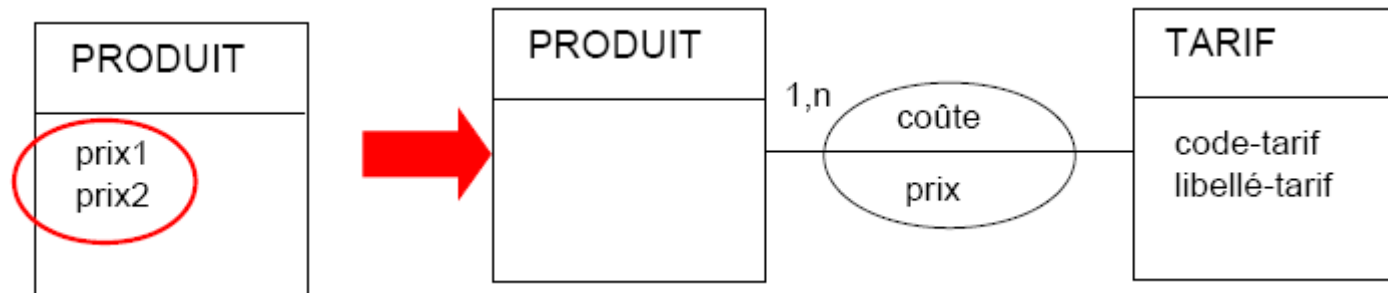


# Vérification et Normalisation(I)

- Contrôler la qualité du modèle, notamment pour supprimer (ou au moins diminuer) la redondance des données (grâce à des règles de normalisation).
- => Permet de détecter certaines incohérences dans la construction des modèles.
- **Règles Générales**
  - Toute propriété doit apparaître une seule fois dans un modèle.
  - Il faut éliminer la redondance des propriétés dans la même entité (avec des noms différents) ou dans des entités distinctes :

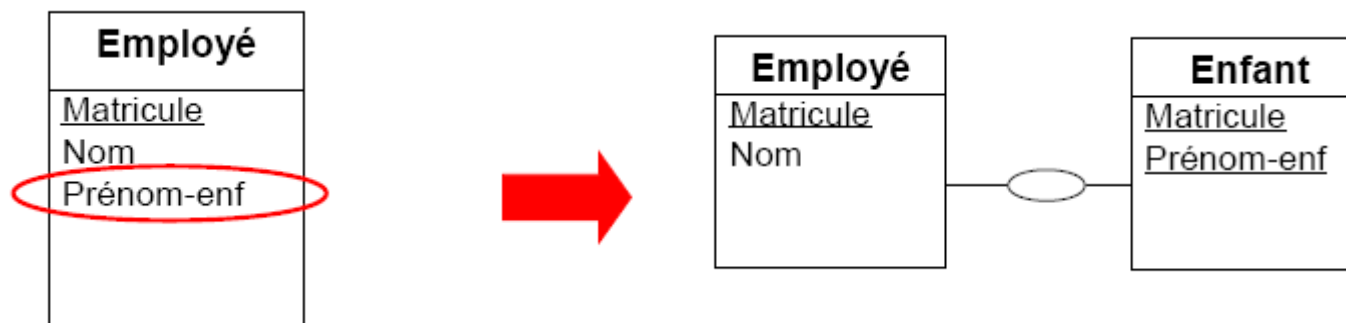
# Vérification et Normalisation(2)

- Exemples:



# Vérification et Normalisation(3)

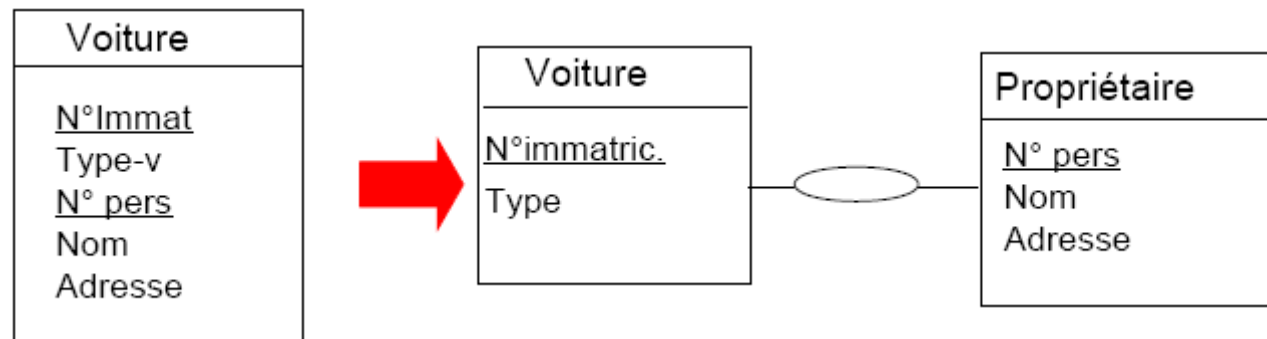
- Règles sur les entités
  - Règle de l'identifiant : Toutes les entités ont un identifiant.
  - Règle de vérification des entités: Pour une occurrence d'une entité, chaque propriété ne prend qu'une seule valeur => Attribut **MONO-VALUEE**



On décompose l'entité Employé en deux entités : Employé, et Enfant

# Vérification et Normalisation(4)

- Règles de normalisation des entités
- Les dépendances fonctionnelles (DF) entre les propriétés d'une entité doivent vérifier la règle suivante : toutes les propriétés de l'entité doivent **dépendre fonctionnellement de l'identifiant et uniquement de l'identifiant**.
- **Définition:** une DF  $X \rightarrow Y$  si à une valeur de  $X$  correspond une et une seule valeur de  $Y$  (la réciproque n'est pas vraie).



DF:  $N^\circ \text{ pers} \rightarrow \text{Nom, Adresse}$  contredit la règle ci-dessus.

# Vérification et Normalisation(5)

En d'autres termes, toutes les DF doivent être élémentaires.

- **Définition** : DF élémentaire : La DF  $X \rightarrow Y$  est élémentaire, s'il n'existe pas  $Z$  inclus dans  $X$  tel que :  $Z \rightarrow Y$ .

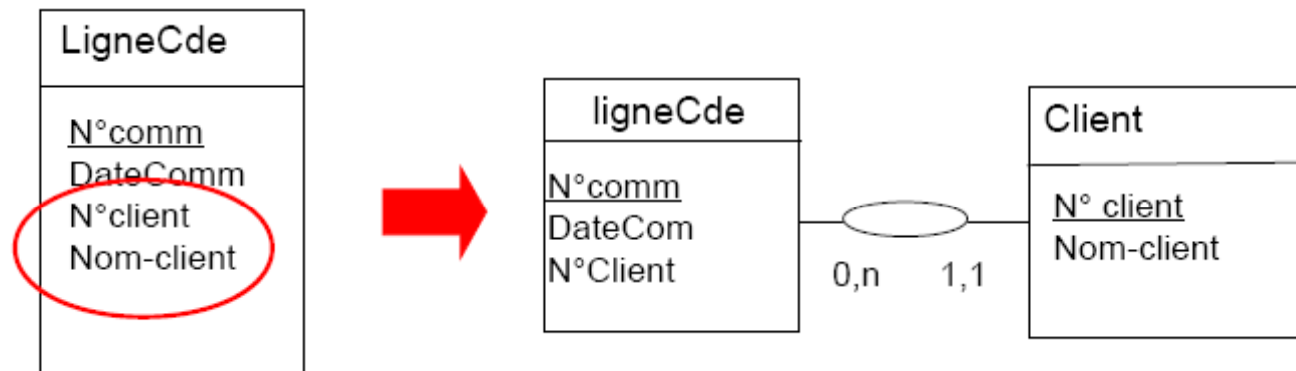
- Dans l'exemple précédent :

1ère entité : on a l'identifiant (N°immat, N°Pers) est en DF avec Nom et adr par exemple, alors que N°pers suffit ! Donc, on en fait deux entités.

## Vérification et Normalisation(6)

- Une partie de l'identifiant ne peut pas déterminer certaines propriétés.

Une ligne de commande concerne plusieurs clients.



La DF  $N^{\circ}\text{client} \rightarrow \text{Nom-client}$  contredit la règle. On décompose l'entité LigneCde en deux entités.

**DF directe** : Si on a les DF  $X \rightarrow Y$  et  $Y \rightarrow Z$ , alors on a la DF  $X \rightarrow Z$  (transitivité). Une relation est en 3 FN si pas de DF directe (pas de transitivité)

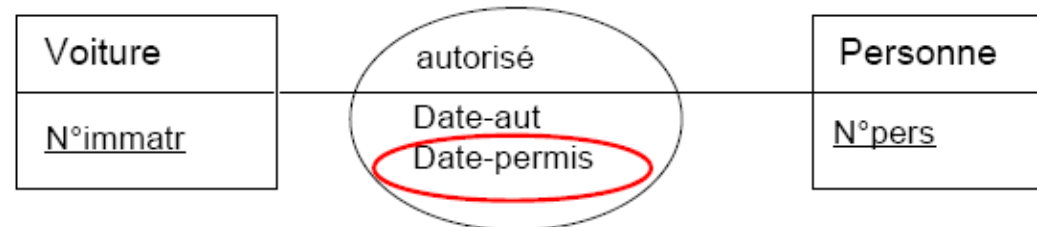
# Vérification et Normalisation(7)

- Règles sur les associations
  - Règle de vérification des associations

Pour une occurrence d'association, chaque propriété ne prend qu'une seule valeur.

- Règle de normalisation sur les propriétés des associations

Toutes les propriétés de l'association doivent dépendre fonctionnellement de tous les identifiants des entités portant l'association, et uniquement d'eux.



La DF N°-pers → Date-permis pose problème (Date-permis doit dépendre des 2 identifiants) => déplacer date-permis vers Personne.

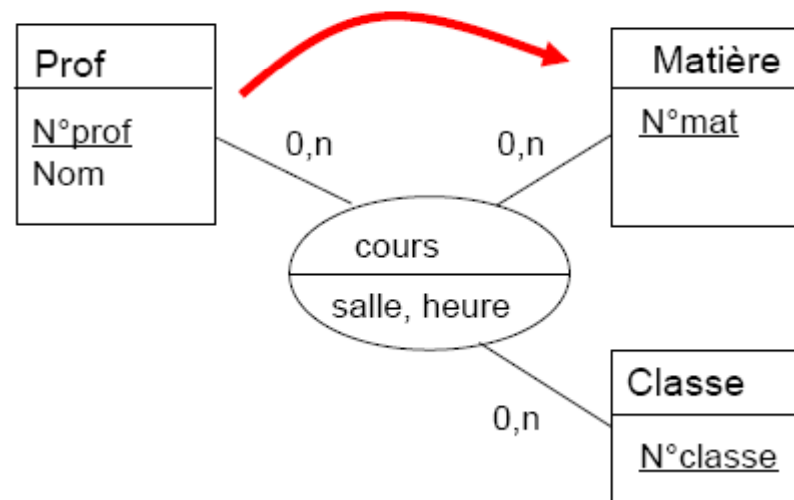


## Vérification et Normalisation(8)

- **La décomposition des associations n-aires**

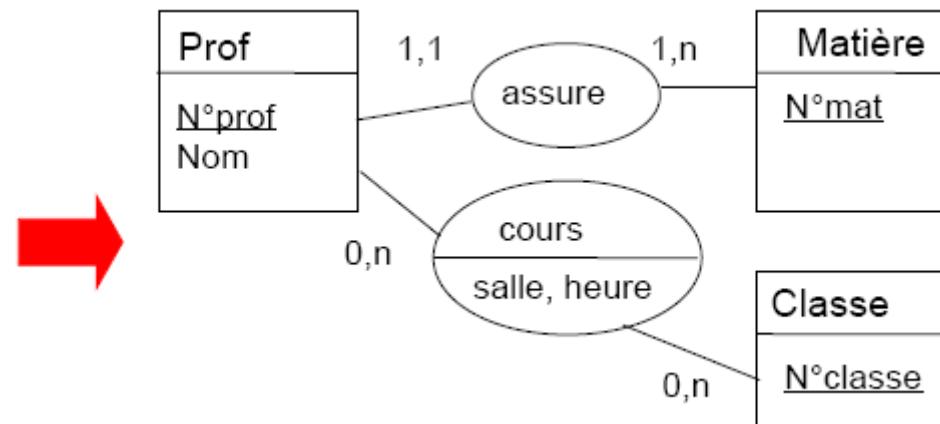
Il faut garder un minimum d'associations d'arité  $> 2$  (n-aire,  $n > 2$ ).

Si on a une DF entre deux identifiants, on peut décomposer l'association n-aire.

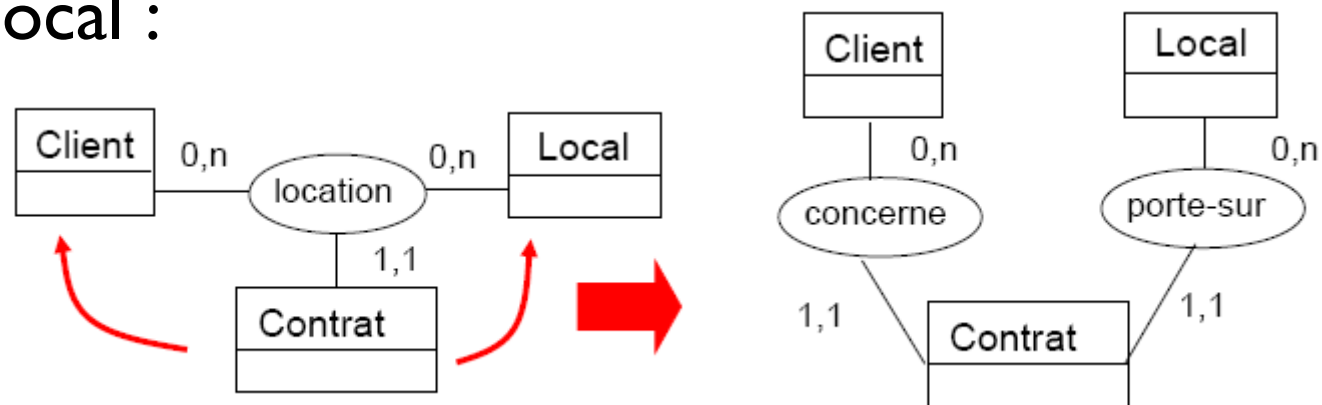


Une éventuelle DF  $N°prof \rightarrow N°mat$  (c.à.d. si un prof enseigne une seule matière) conduit à la décomposition :

# Vérification et Normalisation(9)



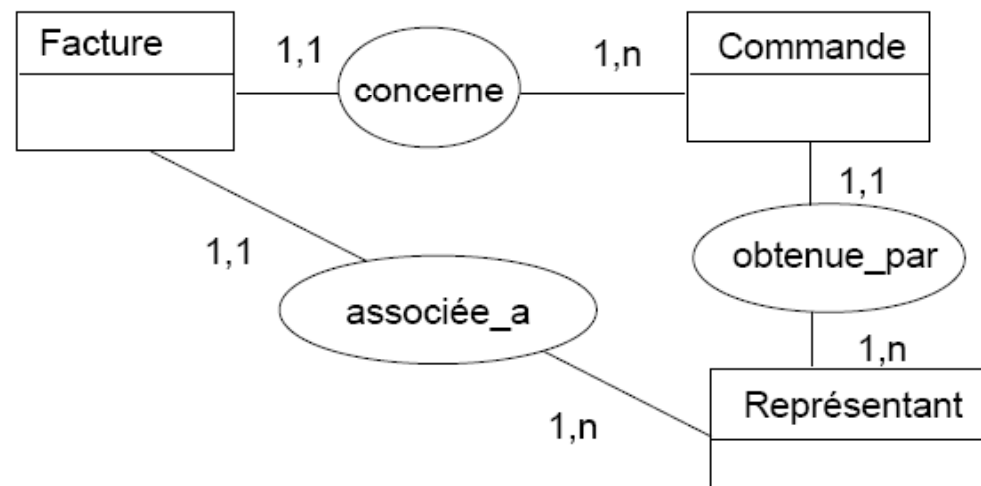
- C'est le cas, quand une patte a une cardinalité 1,1. Par exemple à un contrat est associé un client et un seul local :



# Vérification et Normalisation(10)

- **La suppression des associations transitives**

Toute association pouvant être obtenue par transitivité de n autres associations peut être supprimée. La transitivité s'évalue en regardant la signification des associations.



On supprime l'association **associée\_a**, car elle peut être obtenue par transitivité sur les associations **concerne** et **obtenue\_par**

# Vérification et Normalisation(II)

- **Quelques contraintes d'intégrité importantes:**

Les CI définissent des propriétés qui doivent être vérifiées par les données de la base.

- **Contraintes intégrées au modèle E/A**

- a. Contrainte d'identifiant***

Les valeurs prises par l'identifiant sont uniques (dans le temps) et toujours définies.

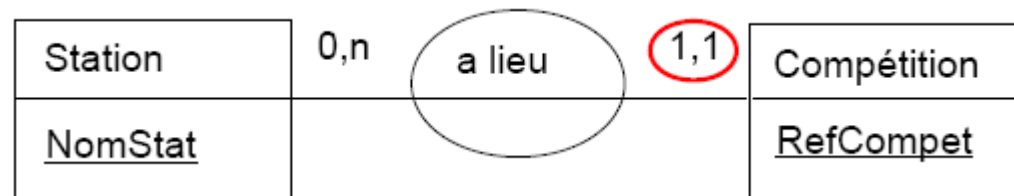
Ex : identifiant de l'entité PERSONNE

- ✓ nom + prénom ; n'est pas suffisant
- ✓ n° téléphone : n'est pas stable dans le temps

# Vérification et Normalisation(I 2)

## ***b. Contraintes de cardinalité***

Les cardinalités portées par les entités membres d'association imposent des nombres minis et maxis d'occurrence dans l'association.



**Une cardinalité mini de 1 rend l'existence d'une occurrence d'entité dépendante de l'existence d'une occurrence d'une autre entité.**

- Une compétition ne peut exister que si la station où elle se déroule existe.
- Une station peut exister de manière indépendante de toute compétition.

# Le modèle logique des données relationnel MLD

- **Définition :**

- Le Modèle Logique des Données (MLD) est une étape intermédiaire pour passer du modèle E/A, qui est un modèle sémantique, vers une représentation physique des données : fichiers, SGBD hiérarchique, SGBD réseau, SGBD relationnel.
- Nous nous limitons au seul MLD relationnel, qui prépare le passage aux SGBD relationnels.

# Le modèle logique des données relationnel MLD

- **Modèle logique relationnel**

Une table relationnelle correspond à un type d'entité ou d'association du SI (ex: commande, ligne de commande, client ...). Elle est composée d'attributs (colonnes) qui décrivent ce type (ex: numéro de cde, date de cde ...). Elle possède une clé primaire : ensemble minimum d'attributs qui permet de repérer de manière univoque chaque n-uplet (ligne) de la table, c'est-à-dire chaque instance du type.



# Le modèle logique des données relationnel MLD

- Exemple : Commande (nucom, datcom, adr\_livr).

Les liens entre tables sont exprimés par les clés étrangères. Une clé étrangère est un ensemble d'attributs d'une table T2 qui est clé primaire dans une table T1.

- Exemple :

Client (nocli, nomcli, adrcli)

Commande(nucom, datcom, adrliv, **nocli\***)

La clé étrangère doit correspondre à une clé primaire existante (contrainte d'intégrité référentielle).

## Passage du Modèle E/A au MLR

**Entité** : Toute Entité devient une table dont la clé primaire est l'identifiant de l'Entité.

Client
<u>Codcli</u>
Nomcli
Adrccli

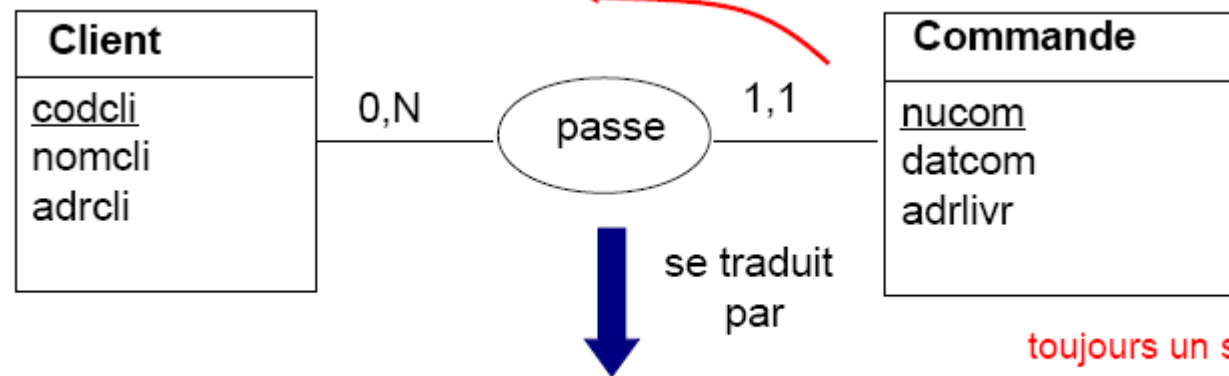


se traduit par

Client (codcli, nomcli, adrccli)

## Passage du Modèle E/A au MLR

- **Association binaire (1/1) - (0/N) ou (1/1) - (1/N)** : se traduit en ajoutant une clé étrangère (identifiant de l'entité de cardinalité (0,N) ou (1,N) ) à la table provenant de l'entité dont la cardinalité est (1,1).



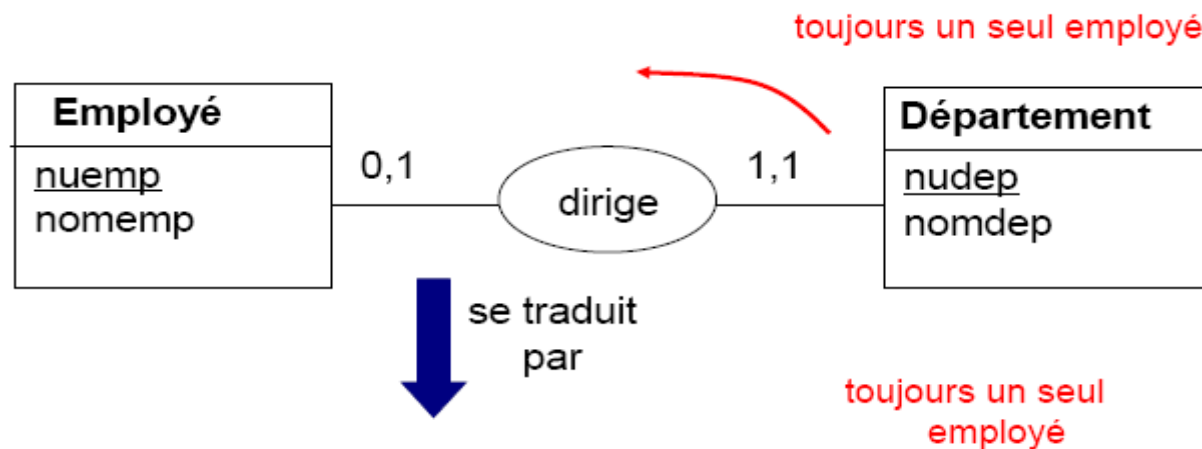
Client (codcli, nomcli, adrcli)

Commande (nucom, datcom, adrlivr, **codcli\***)

toujours un seul  
client (attribut  
monovalué)

## Passage du Modèle E/A au MLR

- **Association binaire (1/1) - (0/1)** : se traduit en ajoutant une clé étrangère (identifiant de l'entité de cardinalité (0,1) ) à la table provenant de l'entité dont la cardinalité est (1,1).

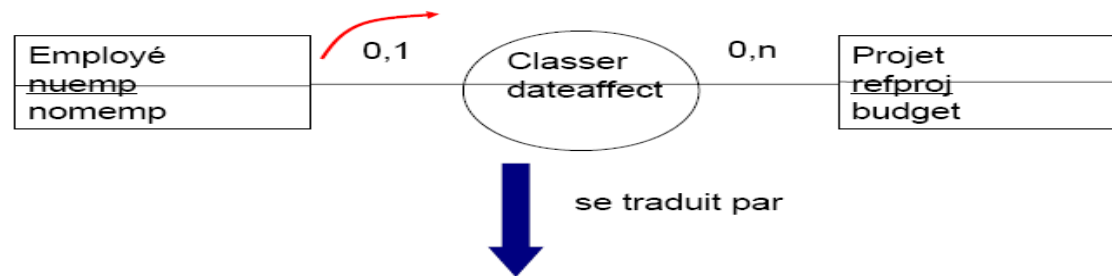


Employé (nuemp, nomemp)

Département (nudep, nomdep, **nuemp\***)

# Passage du Modèle E/A au MLR

- **Association binaire (0/1) - (0/n) :**

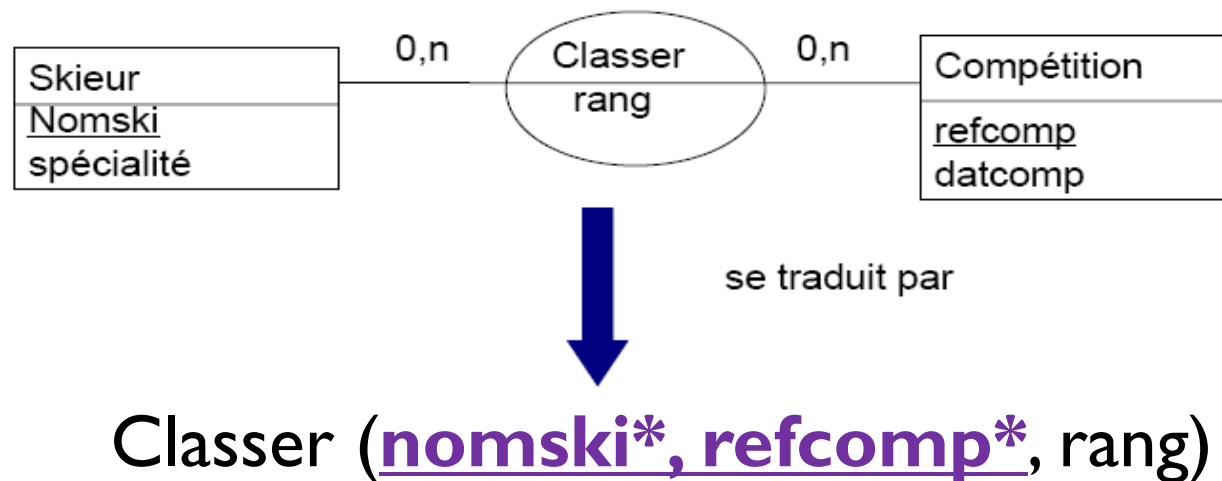


Employé (nuemp, nomenp, **refproj\***, dateaffect) (solution 1)

Classer(nuemp, **refproj\***, dateaffect) (solution 2)

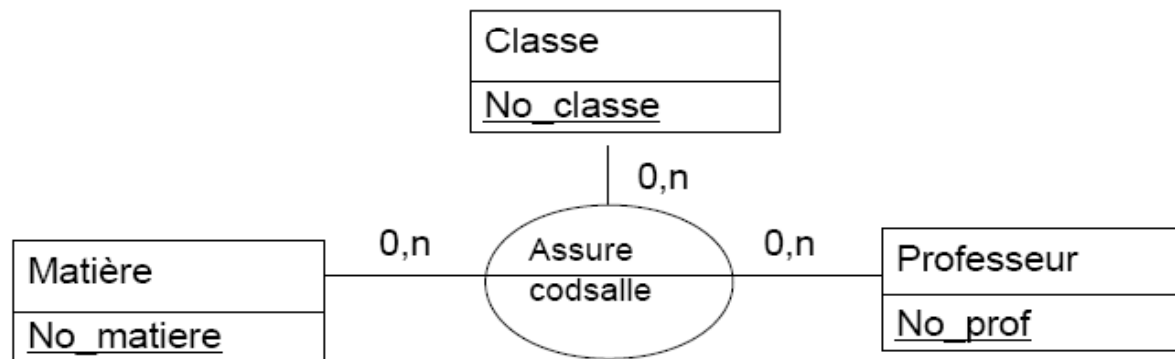
## Passage du Modèle E/A au MLR

**Association binaire (0/N) ou (1/N) - (0/N) ou (1/N) :** se traduit par une nouvelle table dont la clé primaire est composée des identifiants des deux entités. Les éventuelles propriétés de l'association deviennent les attributs de cette table.



## Passage du Modèle E/A au MLR

**Association n-aire ( $n > 2$ )** : on crée une table ayant pour clé primaire les identifiants des différentes entités de l'association. Les éventuelles propriétés de l'association deviennent les attributs de la table.



se traduit par

Assure (no classe\*,no matiere\*,no prof\*,codsalle)