

CHAPITRE 2

Modèle Entités-Relations (E-R)

Contenu du chapitre 2

Après la collecte et l'analyse des besoins de usagers, il faut créer le schéma conceptuel de haut niveau. Nous utiliserons le modèle E-R.

- *Entités, attributs et classe d'entités*
- *Notions de relations entre les entités*
- *Contraintes structurelles d'une classe de relations*
- *Extension, restriction et agrégation*
- *Notation graphique du modèle E-R*
- *Exemple de conception haut-niveau d'une BD*

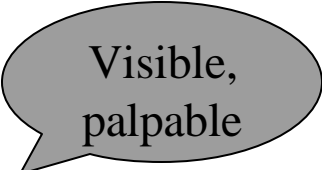
Entités

Entité

- *L'objet de base du modèle **E-R**.*
- *Un objet du monde réel qui est discernable des autres objets.*

On peut distinguer deux types d'entités :

- **Entités concrètes** : *pièce mécanique, personne*
- **Entités abstraites** : *période de temps, action*



Visible,
palpable



Pas
visible

Ex: Individu : Gaston Bernier, Code permanent BERG 28086401

Cours : Base de données à l'ÉTS

Représentent des **objets uniques** dans l'univers :

- ♦ *une personne*
- ♦ *un certain cours à l'ÉTS.*

Attributs

Attributs

➤ *Propriétés qui compose l'entité*

Ex: Employé : NAS, Prénom, Nom , Adresse

Compte d'épargne : Numéro, Solde

➤ *Comporte des domaines de valeurs autorisées*

Ex: Nom : chaîne de 12 caractères

Numéro : ensemble d'entiers positifs

Types d'attributs:

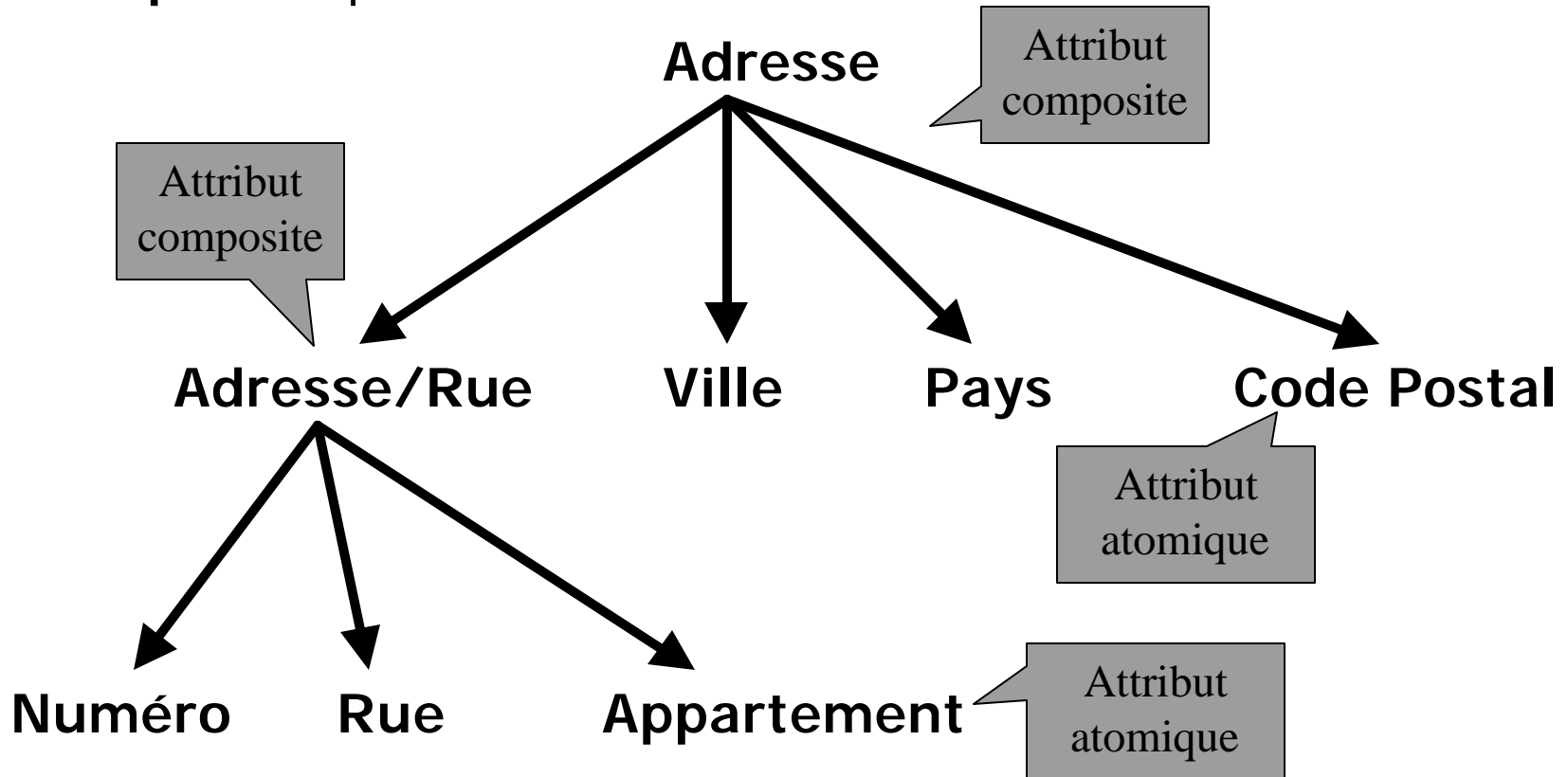
- Valeur simple ou valeurs multiples
- Dérivé
- Simple (atomique) ou composé (composite)

Attributs

Attributs

Composite: composé de plusieurs autres attributs de base.

Atomique: pas divisible.



Attributs

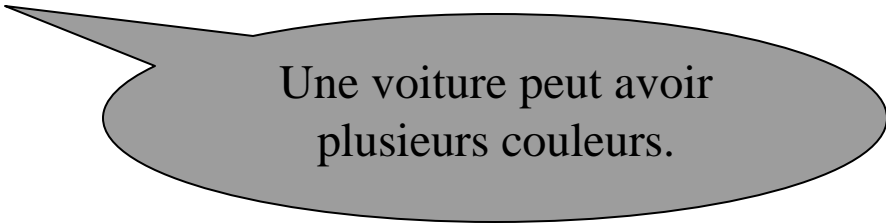
Attributs

Valeur simple: possède qu'une seule valeur (valeur unique).

Ex: *Âge possède une seule valeur.*

Valeurs multiples: possède un ensemble de valeurs pour la même entité

Ex: *Couleur pour une voiture (rouge, verte, bleue, jaune, etc...)*



Une voiture peut avoir plusieurs couleurs.

Dérivé: attribut qui est en étroite relation avec un autre attribut.

Ex: *Âge et date de naissance.*

Attributs

Attributs clés

- Ensemble d'un ou plusieurs attributs qui permet d'identifier **de façon unique** une entité dans l'ensemble des entités.

*Ex: Le NAS de l'entité **EMPLOYÉ** suffit de distinguer un employé d'un autre.*

*Ex: Le code permanent de l'entité **ÉTUDIANT** suffit de distinguer un étudiant d'un autre.*

Caractéristiques d'une clé

- Superclé
- Clé secondaire ou clé candidate
- Clé primaire

Attributs

Attributs clés

SuperClé

- Tout ensemble d'un ou plusieurs attributs qui permet d'identifier de façon unique une entité dans l'ensemble des entités.

Remarque: Chaque entité possède au moins une superclé qui est l'ensemble de tous ses attributs.

Clé

- C'est une superclé à laquelle on ne peut plus enlever d'attributs.

Ex : L'ensemble d'attributs {NAS, Nom, Age} est une superclé de l'entité ÉTUDIANT, mais pas une clé.

Clé Candidate

- *Clé possédant les caractéristiques pour être une clé primaire.*

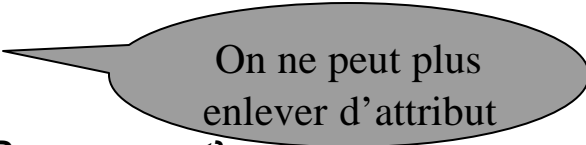
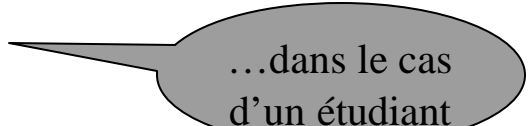
Attributs

Attributs clés

Clé Primaire

- *La clé désignée parmi les clés candidates par le concepteur de la base de données pour identifier de façon unique une entité.*
- *Soulignée dans le schéma de E-R.*

Exemple: ÉTUDIANT{NAS, Code Permanent, Nom, Age}

- *SuperClé: {NAS, Code Permanent Nom, Age} ou {NAS, Code Permanent, Nom}*
- *Clé: {NAS} ou {Code Permanent}* 
- *Clé candidate: {NAS} ou {Code Permanent}*
- *Clé primaire: {Code Permanent}* 

Classes d'entités

Type d'entités

➤ *Ensemble d'entités du même type.*

Ex: **Employé** : ensemble des personnes qui travaille pour une compagnie.

SHÉMA :

EMPLOYÉ
Nom, Age, Salaire

COMPAGNIE
Nom, Siège Social, Président

INSTANCES :

e1.
(John Smith, 55, \$80000)

e2.
(Fred Brown, 40, \$30000)

e3.
(Judy Clark, 25, \$20000)

·
·
·

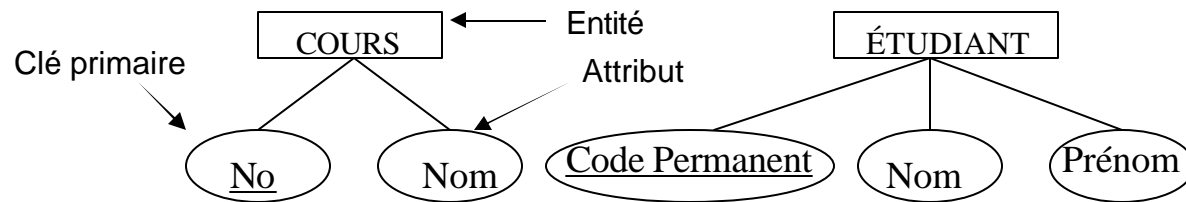
c1.
(Sunco Oil, Houston, John Smith)

c2.
(Fast Computer, Dallas, Bob King)

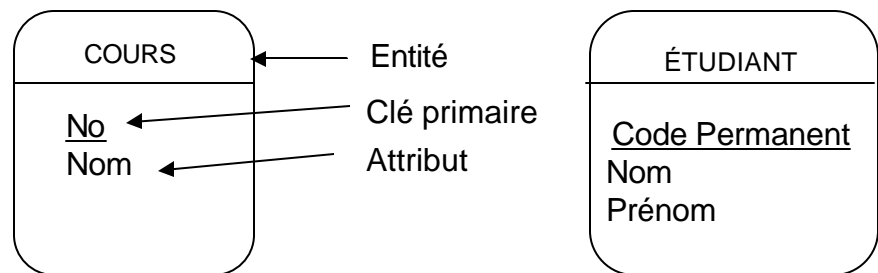
·
·
·

Représentation graphique de l'entité et l'attribut

1ère méthode:



2ième méthode:

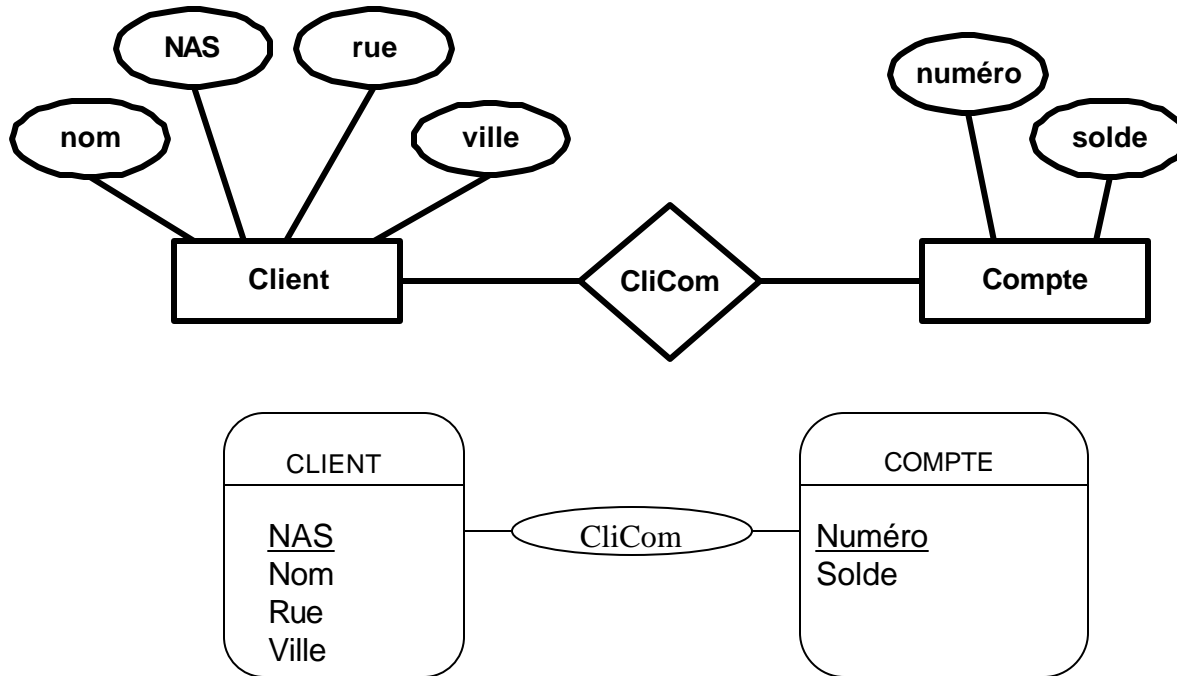


Notions de relations entre les entités

RELATION

Une relation associe **plusieurs** entités.

Ex : étudiant Claveau " Suit le cours " de Base de données
client Olivier " Possède " le compte #259

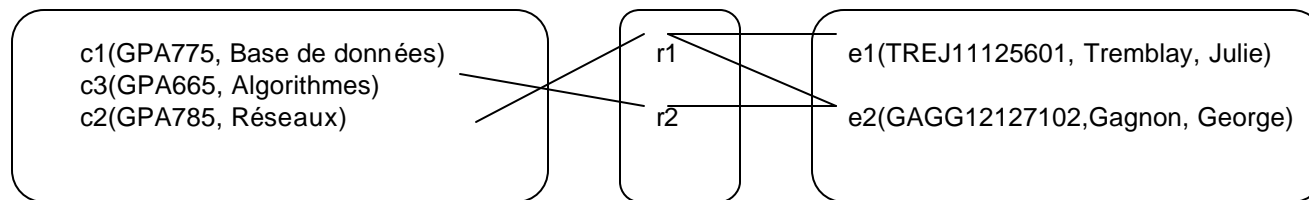


Notions de relations entre les entités

TYPE DE RELATIONS

Un **type de relations** est un ensemble de relations qui agissent sur les mêmes types d'entités.

Ici, l'ensemble de relations (r1,r2) représente un type de relation.



Notions de relations entre les entités

Degré du type relation

Le **degré** du type de **relation** est le nombre de types (classes) d'entités participantes.

*Ex : Le **degré** du type de relations Clicom est égale à deux.*

- *Une relation de **degré deux** est appelé **relation binaire**.*
- *Une relation de **degré trois** est appelé **relation ternaire**.*
- *Une relation peut être de n'importe quel degré. En général, elles sont binaires.*

Notions de relations entre les entités

Relation ternaire

Exemple
tableau

- Représente plus d'information que trois types de relations binaires.

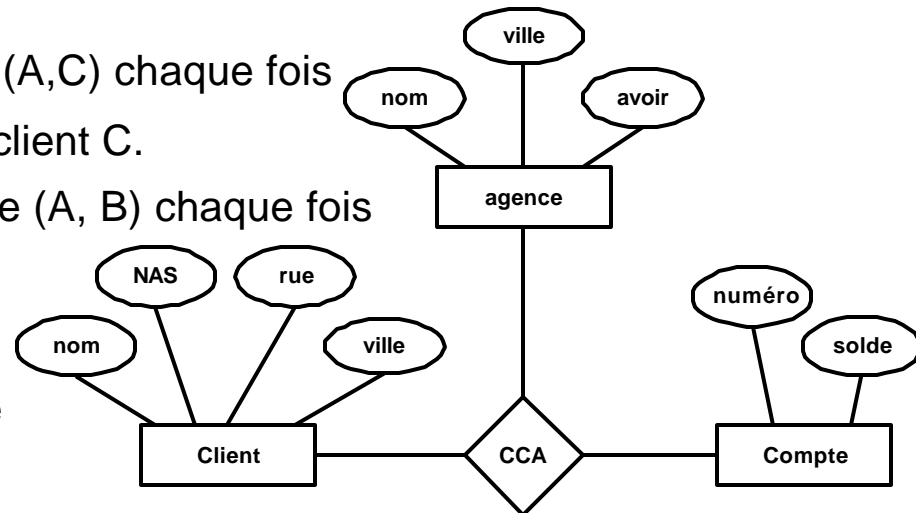
Exemple : Cette relation associe trois entités (client, agence et compte), chaque fois que le client C accède au compte B de l'agence A.

Supposons maintenant que :

AGENCE -> CLIENT inclut l'instance (A,C) chaque fois que l'agence A communique avec le client C.

AGENCE -> COMPTE inclut l'instance (A, B) chaque fois qu'une agence A vérifie un compte B.

CLIENT -> COMPTE inclut l'instance (C, B) chaque fois que le client vérifie le solde de son compte B.



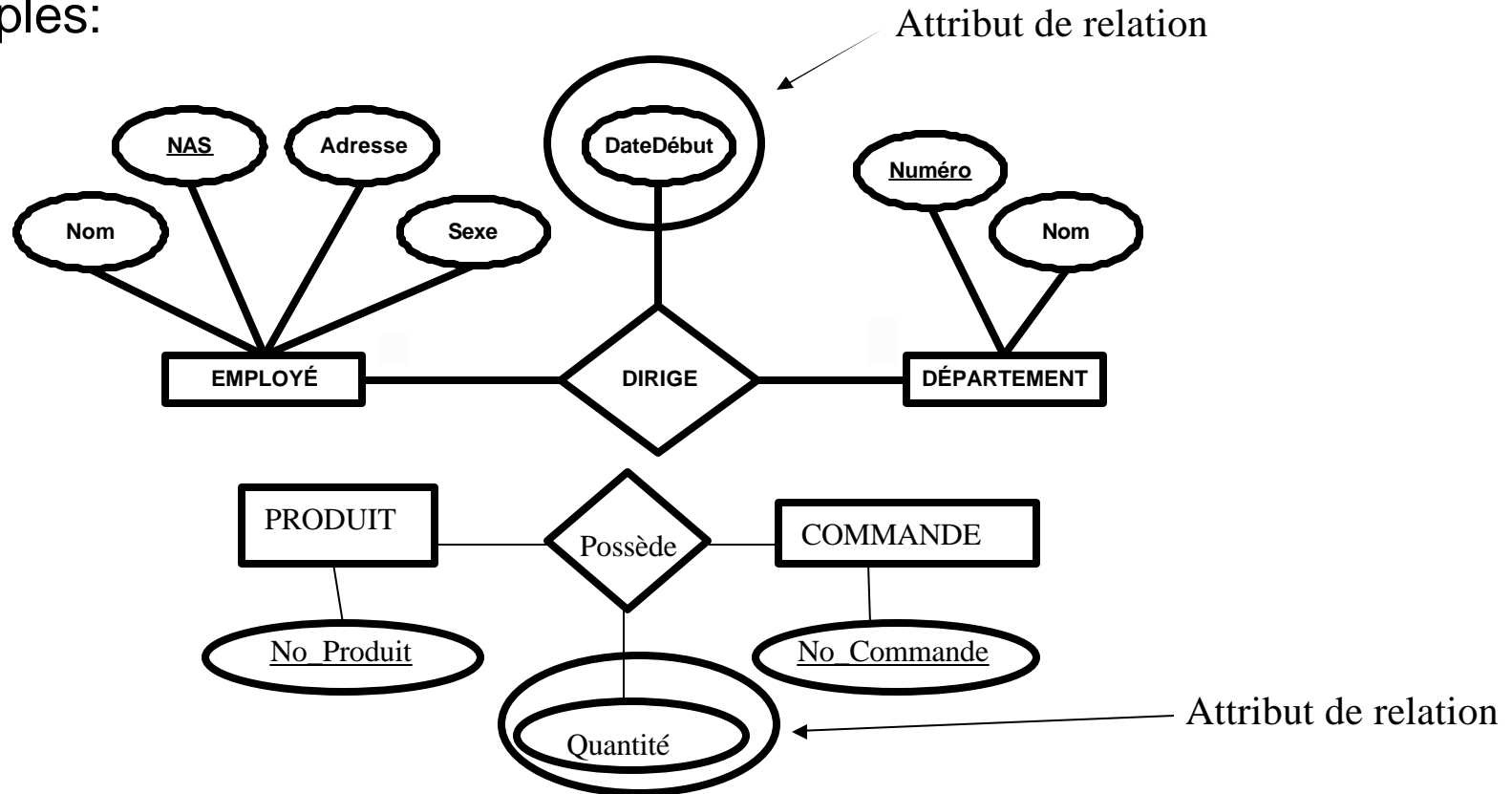
Les 3 instances (A,C), (A,B) et (C,B) n'implique pas nécessairement que l'instance (A,B,C) existe dans la relation ternaire ABC.

Notions de relations entre les entités

ATTRIBUTS D'UNE RELATION

Une relation peut posséder des attributs.

Exemples:



Notions de relations entre les entités

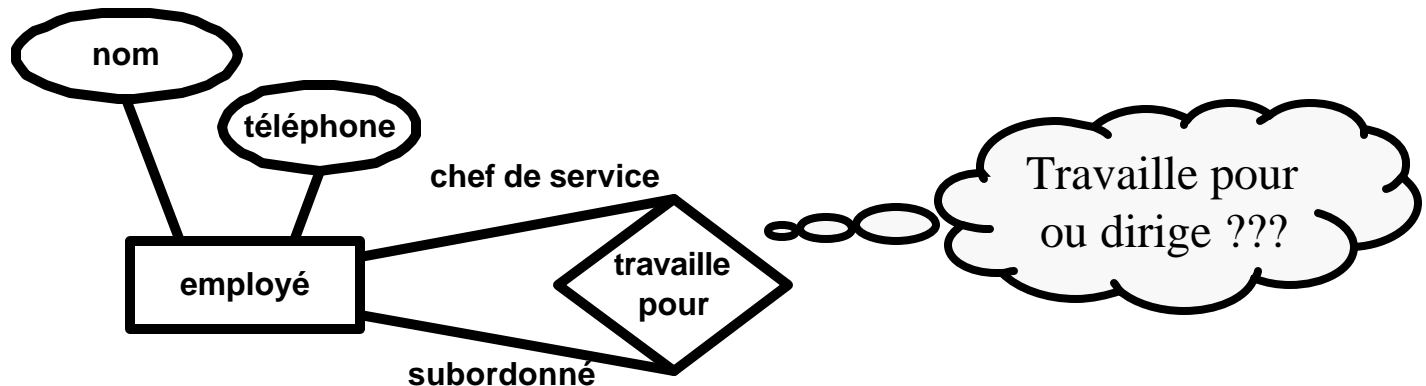
Rôle d'une entité dans une relation

Le nom du rôle d'une entité est habituellement implicite.

Ex: (client 1, compte 2) définition implicite, le client 1 **possède** le compte 2.

Pour d'autres situations, la relation peut être ambiguë:

Ex:



Ici, la même classe d'entité participe plus d'une fois dans un type de relation. C'est ce qu'on appelle un type de **relations récursives**.

Contraintes structurelles d'une classe de relations

Contraintes structurelles

- *limitent les combinaisons possibles des entités participantes dans les instances de relations.*
- *Ces **contraintes** proviennent du monde réel.*

Il existe deux types de contraintes:

- **Contraintes de correspondance (ratio de cardinalité)**
- **Contraintes de participation**

Contraintes structurelles d'une classe de relations

Contrainte de correspondance

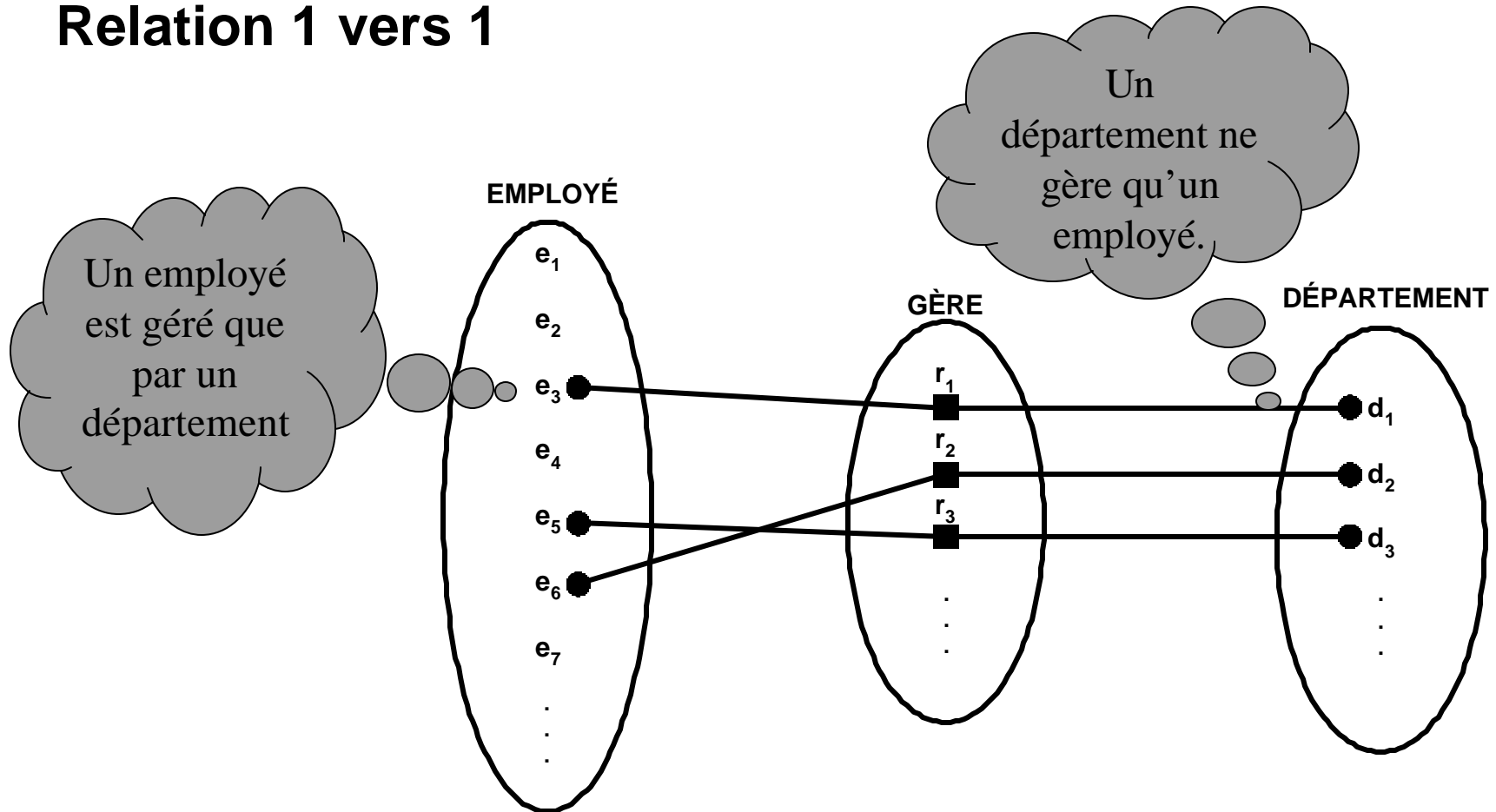
- *définit le nombre (ratio de cardinalité) d'entités auxquelles est associée une entité donnée par la relation considérée.*

Pour les **relations binaires**, les **ratios de cardinalité** rencontrés sont :

- **1 : 1** dit 1 vers 1
- **1 : N** dit 1 vers n **ou** **N : 1** dit n vers 1
- **N : N** dit n vers n

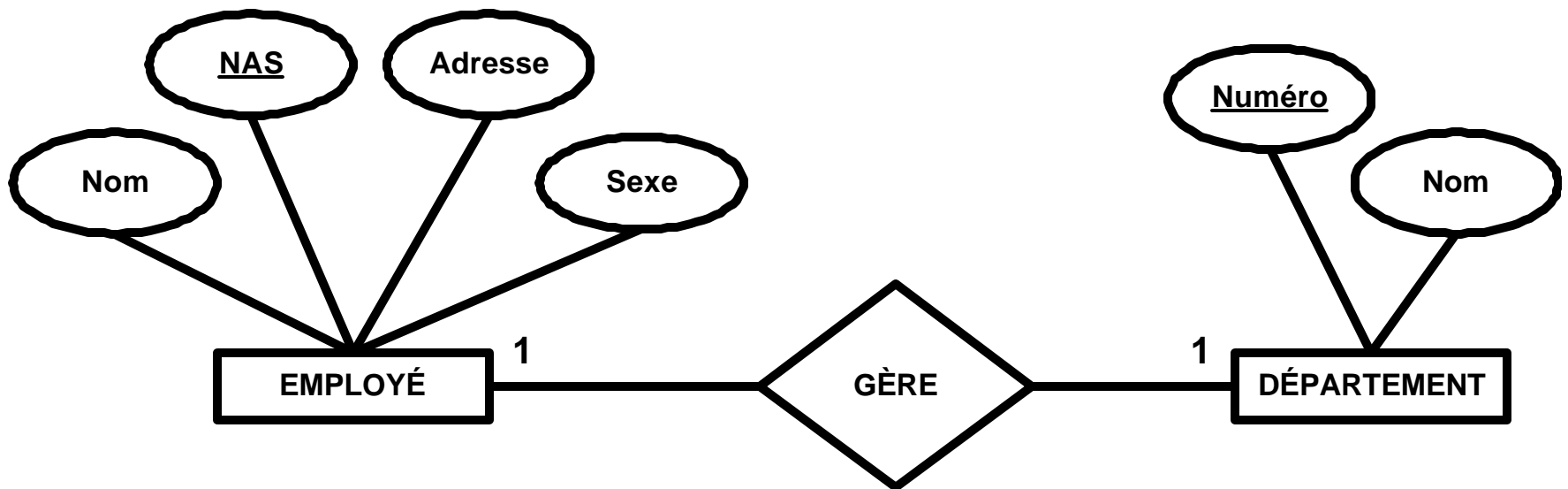
Contraintes structurelles d'une classe de relations

Relation 1 vers 1



Contraintes structurelles d'une classe de relations

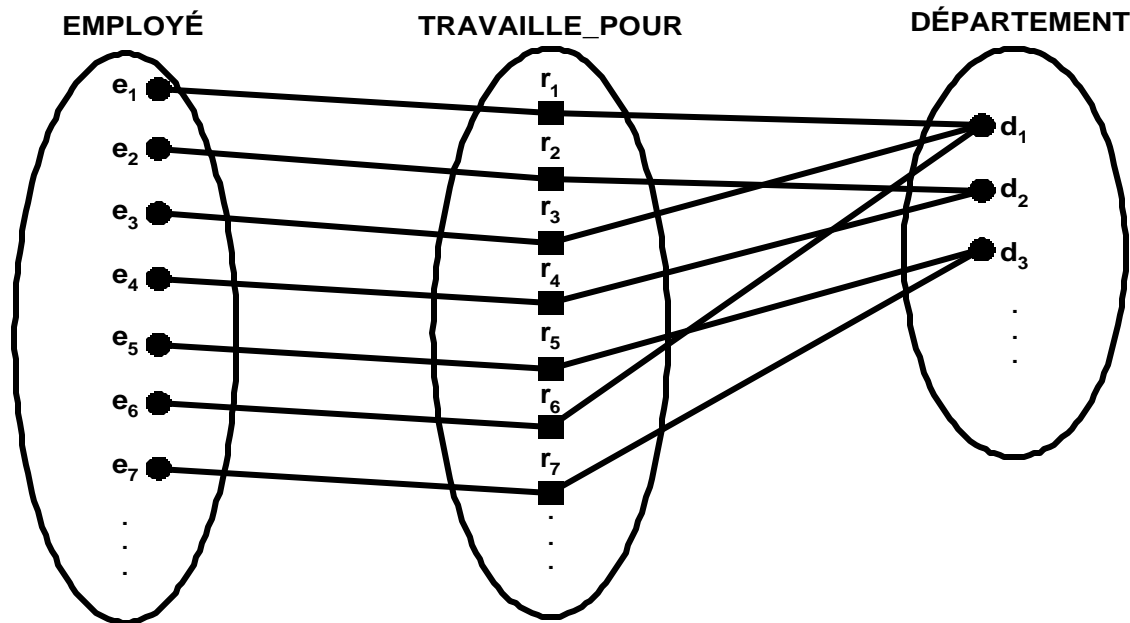
Relation 1 vers 1



Contraintes structurelles d'une classe de relations

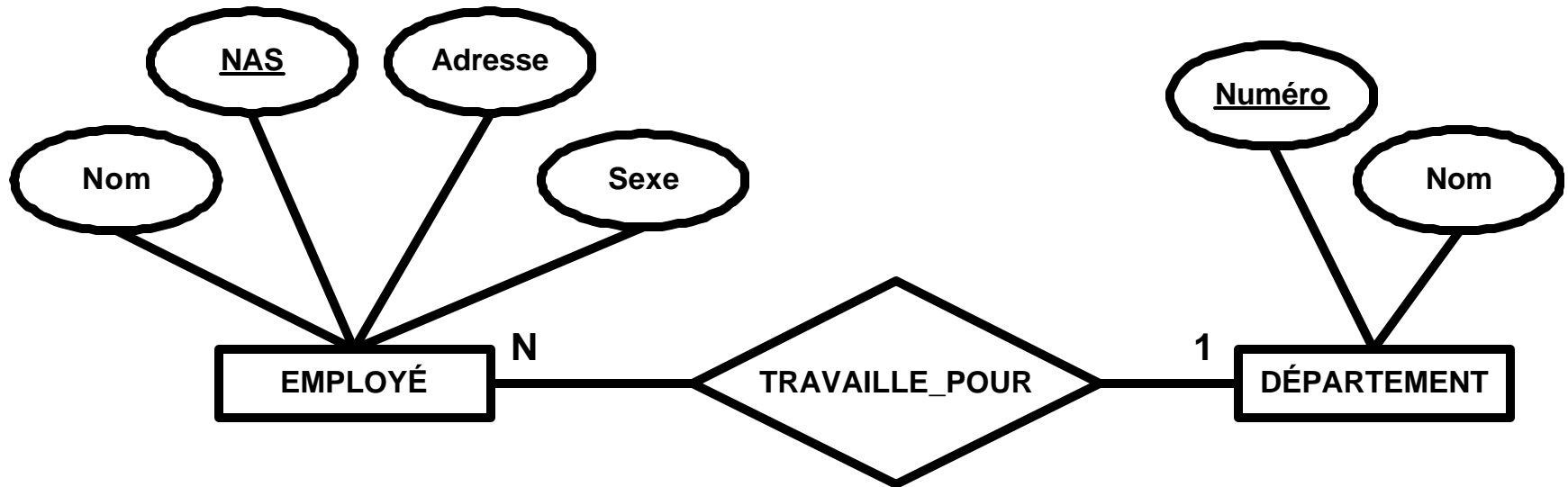
Relation 1 vers n ou n vers 1

- l'employé doit travailler pour un seul département
- le département peut avoir plusieurs employés.



Contraintes structurelles d'une classe de relations

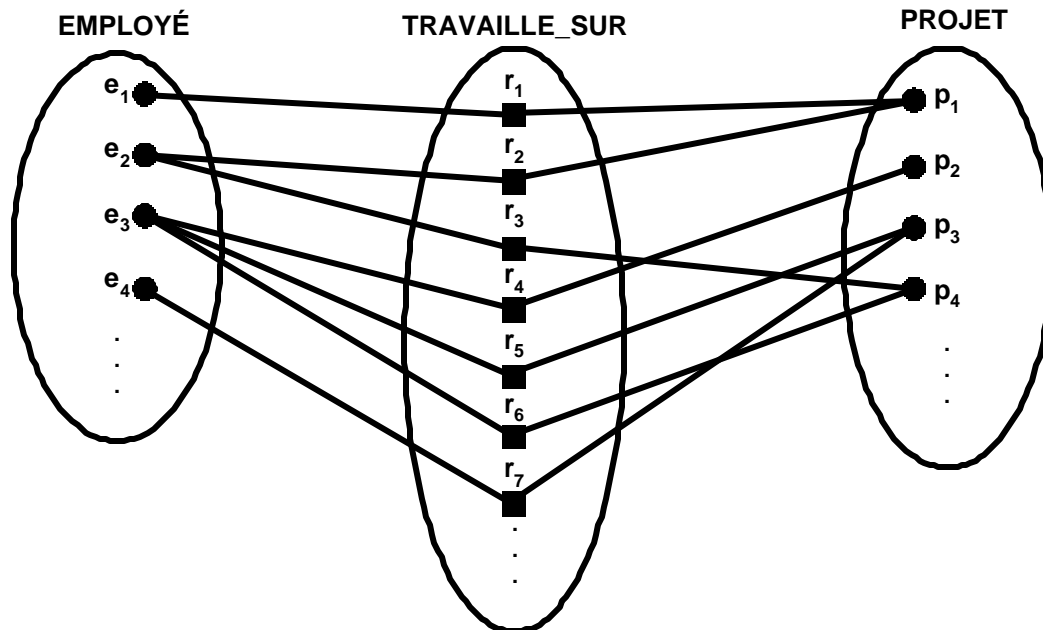
Relation 1 vers n ou n vers 1



Contraintes structurelles d'une classe de relations

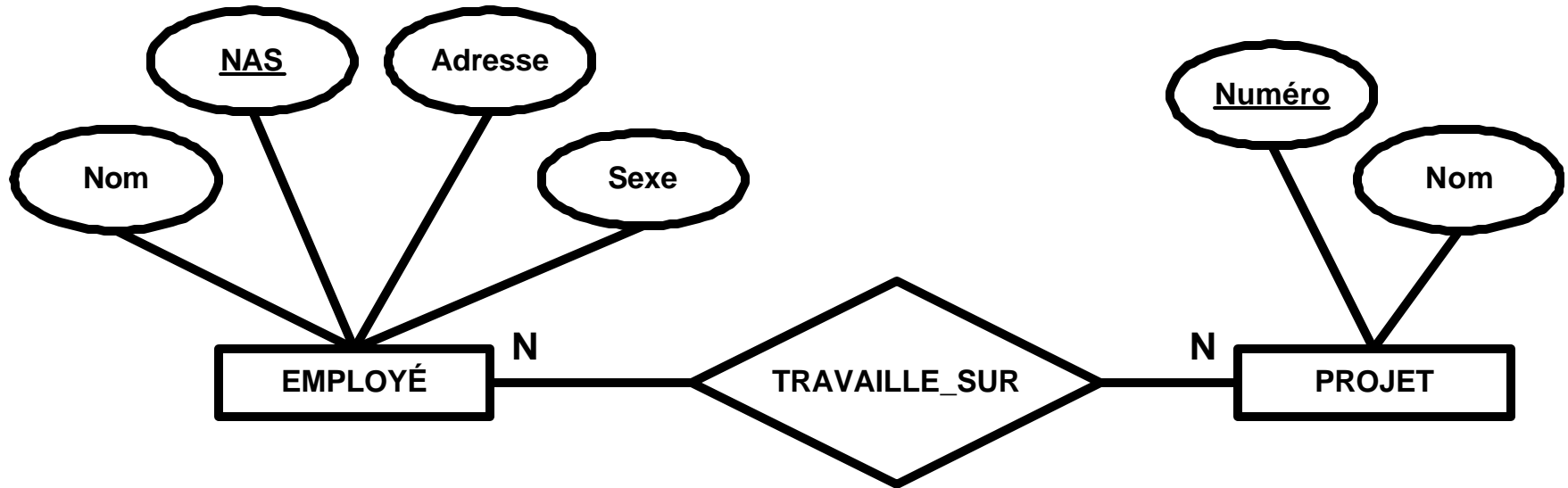
Relation n vers n

- l'employé peut travailler pour un ou plusieurs projets
- le projet peut avoir un ou plusieurs employés.



Contraintes structurelles d'une classe de relations

Relation n vers n



Contraintes structurelles d'une classe de relations

Attributs de la relation et ratio cardinalité

Le **ratio de cardinalité** peut servir à déterminer ce qu'il adviendra des **attributs de relation**.

- **Relation 1:1**, l'attribut de relation peut être inclus dans l'une ou l'autre des entités participantes.

Ex: Relation GÈRE, contrainte 1:1  (tableau)

- L'attribut de relation DATE_DÉBUT peut être dans les entités EMPLOYÉ ou DÉPARTEMENT sans préférence.

- **Relation 1:N**, l'attribut de relation peut être seulement du côté N de la classe de relations.

Ex: Relation TRAVAILLE_POUR, contrainte 1:N  (tableau)

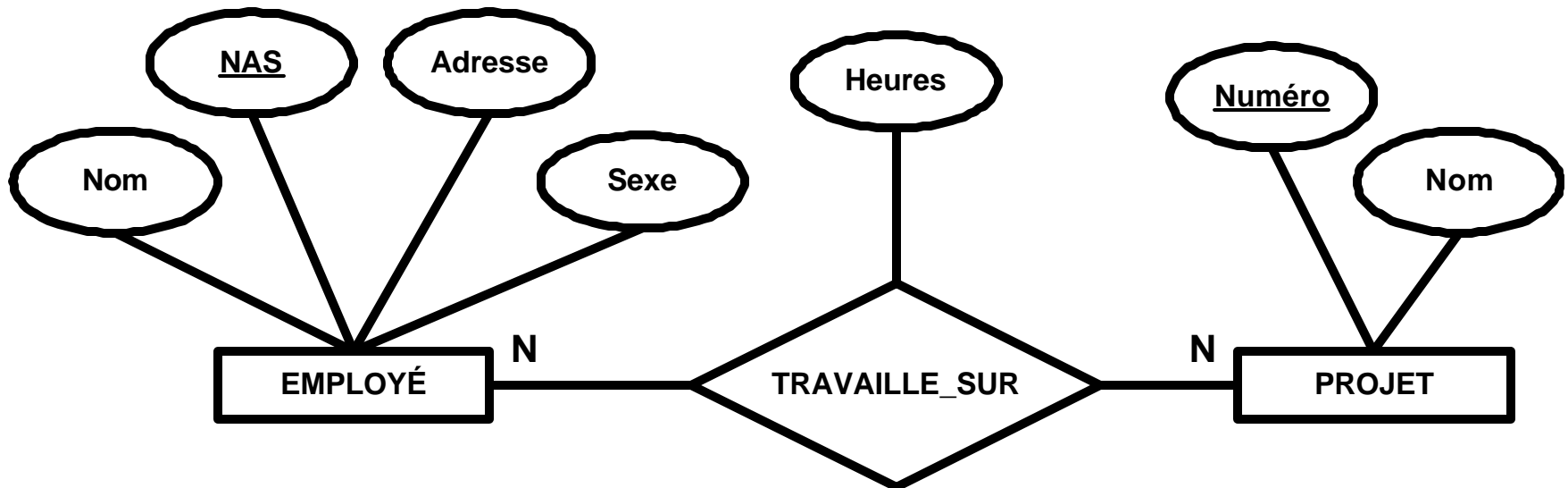
- L'attribut DATE_DÉBUT peut être inclus dans l'entité EMPLOYÉ.

Contraintes structurelles d'une classe de relations

Attributs de la relation et ratio cardinalité

➤ **Relation N:N**, l'attribut doit rester comme attribut de relation.

*Ex: L'attribut HEURES dans TRAVAILLE_SUR reste comme attribut de la classe de relations **N:N**.*



Contraintes structurelles d'une classe de relations

Contraintes de participation

Spécifie si l'existence d'une entité est dépendante d'une autre entité via le **type de relation**.

*Il y a 2 types de **contraintes de participation**:*

➤ *Participation **Totale***

➤ *Participation **Partielle***

Ex:

La participation de EMPLOYÉ dans TRAVAILLE POUR est **totale**.



➤ *Tout employé dans l'ensemble total des employés doit être relié à une entité département.*

La participation de EMPLOYÉ dans GÈRE est **partielle**.



➤ *Une partie seulement des employés sont reliés à une entité département via GÈRE.*

Contraintes structurelles d'une classe de relations

Contraintes de participation

On associe une **paire d'entiers** (min, max) pour chaque participation d'une classe d'entité dans une classe de relation.

Où : $0 \leq \min \leq \max$ et $\max \geq 1$

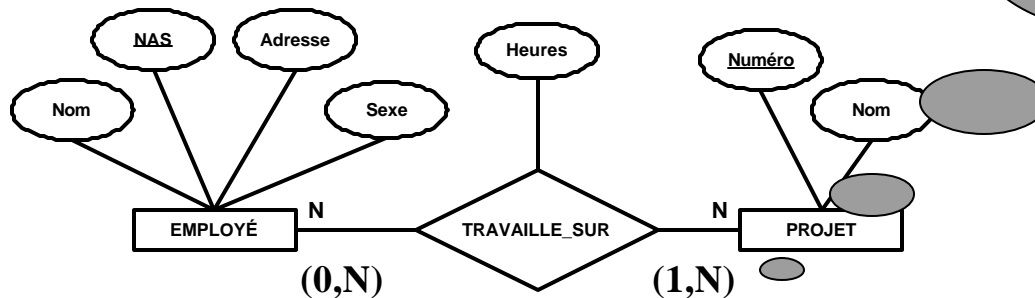
Signifie: Pour tout $e \in E$, e participe dans des instances de relations R avec au moins min et au plus max, à n'importe quel moment.

Remarque:

Min = 0 \rightarrow participation **partielle**

Min > 0 \rightarrow participation **totale**

Ex:



L'employé peut ne pas être associé à un projet (0), mais un projet doit avoir au moins un employé!

Contraintes structurelles d'une classe de relations

Force d'une entité

Classe d'entités faibles

- *ne possèdent pas d'attributs clé.*
- *elle est dépendante d'une autre entité*
- *elle a une contrainte de participation totale*

Classe d'entités fortes

- ***classe d'entités*** à laquelle est reliée la ***classe d'entités faibles***.

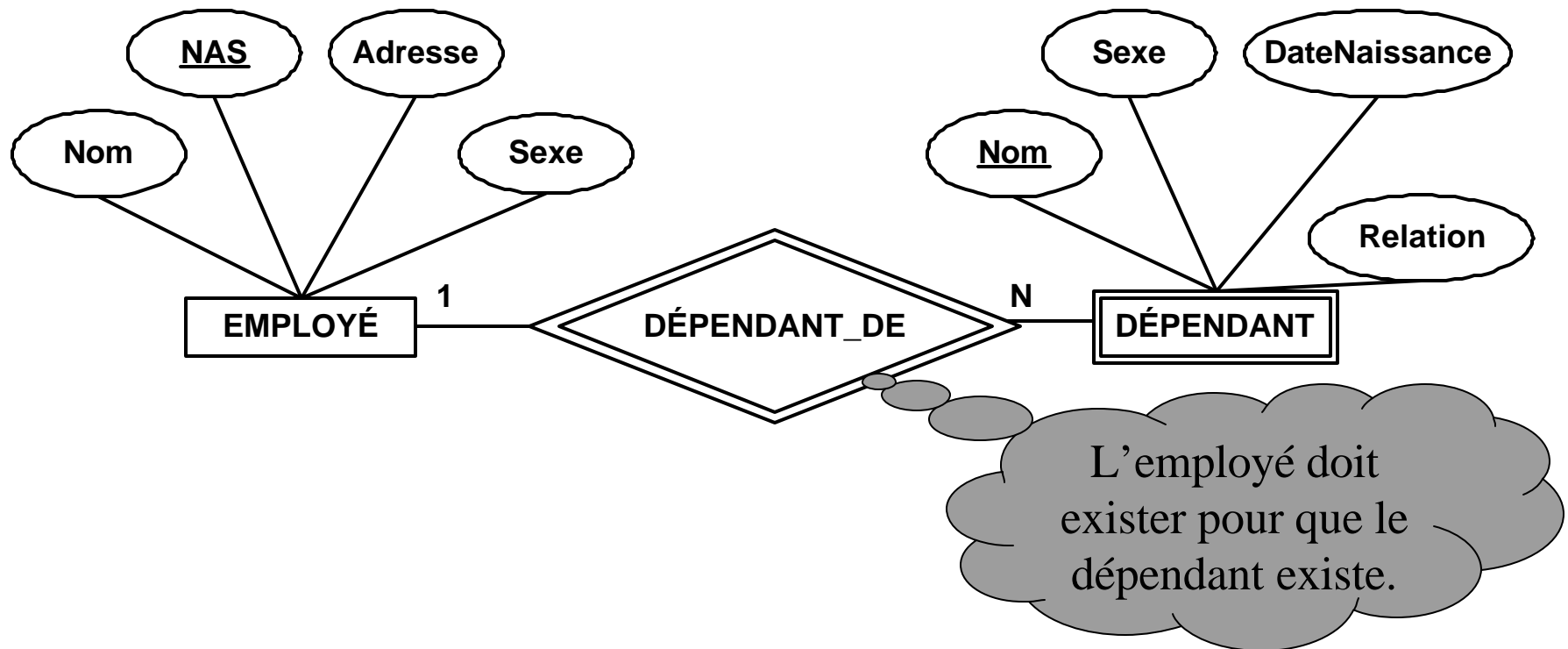
Relation d'identification

- *Classe de relations qui relie la classe d'entité forte et faible.*

Contraintes structurelles d'une classe de relations

Force d'une entité

Lorsqu'un employé est supprimé de la base de données, tous ses dépendants disparaissent aussi.



Contraintes structurelles d'une classe de relations

Force d'une entité

Avantages des entités faibles

- Évite la duplication des données; ex: le NAS de l'employé
- Reflète la dépendance d'une entité faible par rapport à une entité forte;
- Effacement automatiquement de l'entité faible à l'effacement de l'entité forte;
- L'entité faible peut être stockée physiquement avec l'entité forte; (Facilite la recherche)

Contraintes structurelles d'une classe de relations

Force d'une entité

Identification d'une entité faible

- *créer un " **jeu d'attributs** " qui permet de distinguer de façon unique une entité faible.*

Jeu d'attributs = **discriminant** (entité faible) + **clé primaire** (entité forte)

Le **discriminant** permet de différencier les entités dépendants d'une même entité forte.

Ex: Le **discriminant** de DÉPENDANTTest NomDépendant.

La **clé primaire** de EMPLOYÉ est NAS.

Pour retrouver tous les dépendants qui sont associés à l'employé:

Jeu d'attributs = NomDépendant + NAS

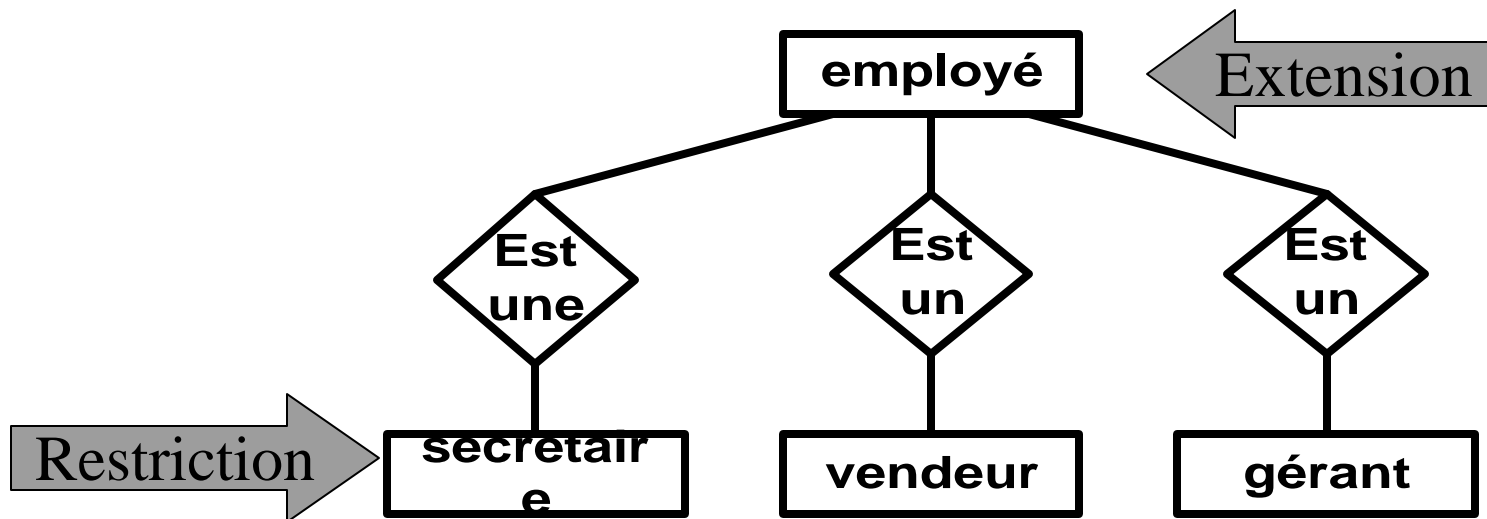
Extension et restriction

Extension :

- **Réunir** deux ou plusieurs entités d'un niveau pour créer une entité de niveau supérieur.

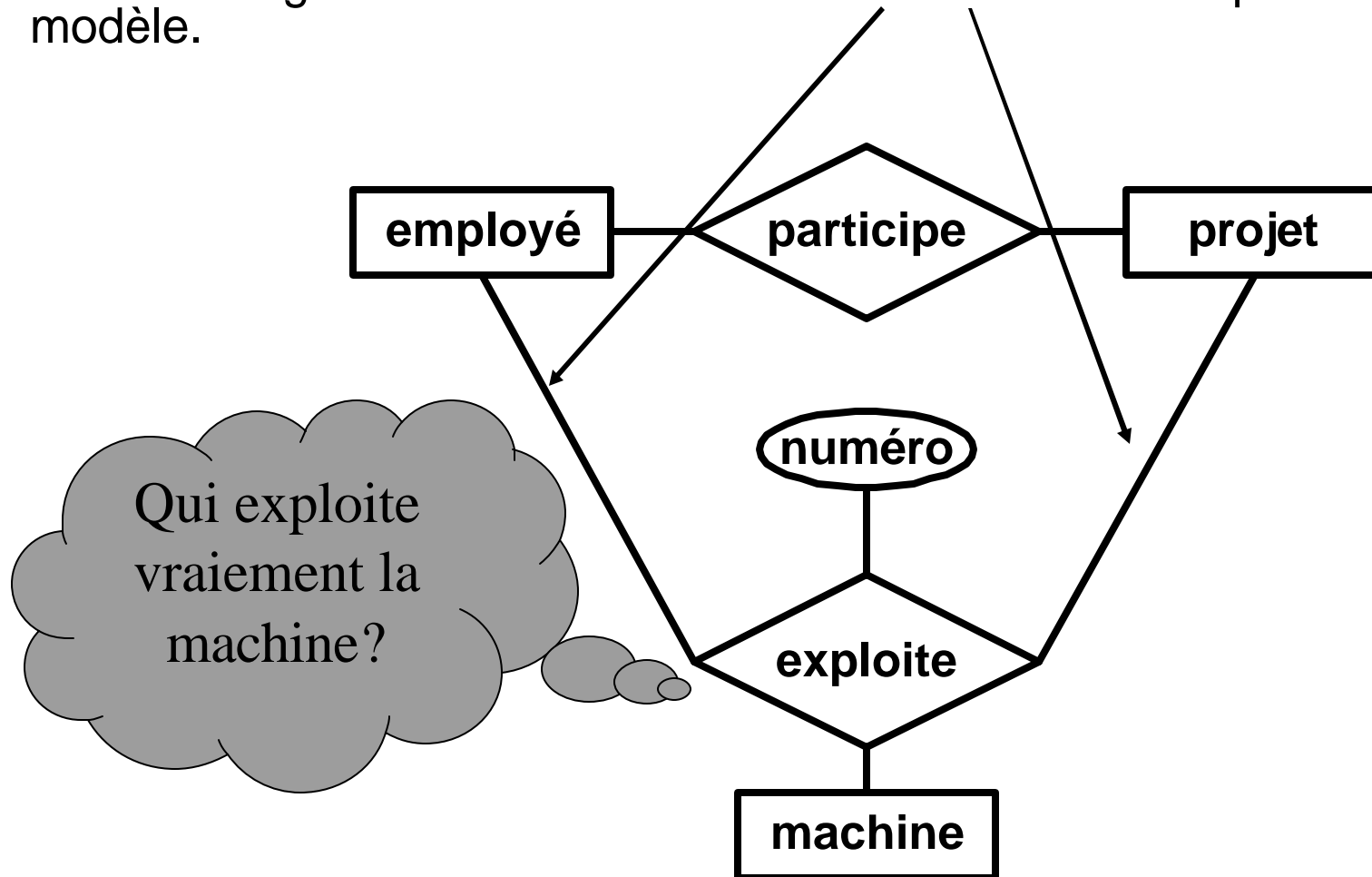
Restriction :

- **Prélever** un sous-ensemble d'une entité d'un niveau pour créer une entité de niveau inférieur.



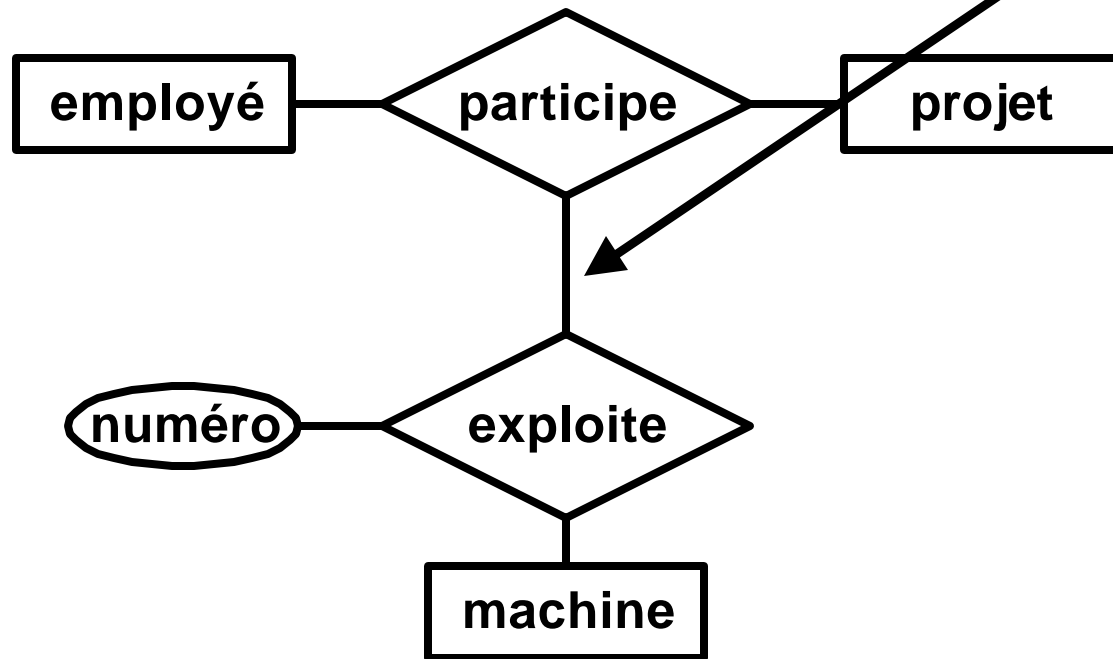
Agrégation

Voici un diagramme E-R avec relation redondantes non permise par le modèle.



Agrégation

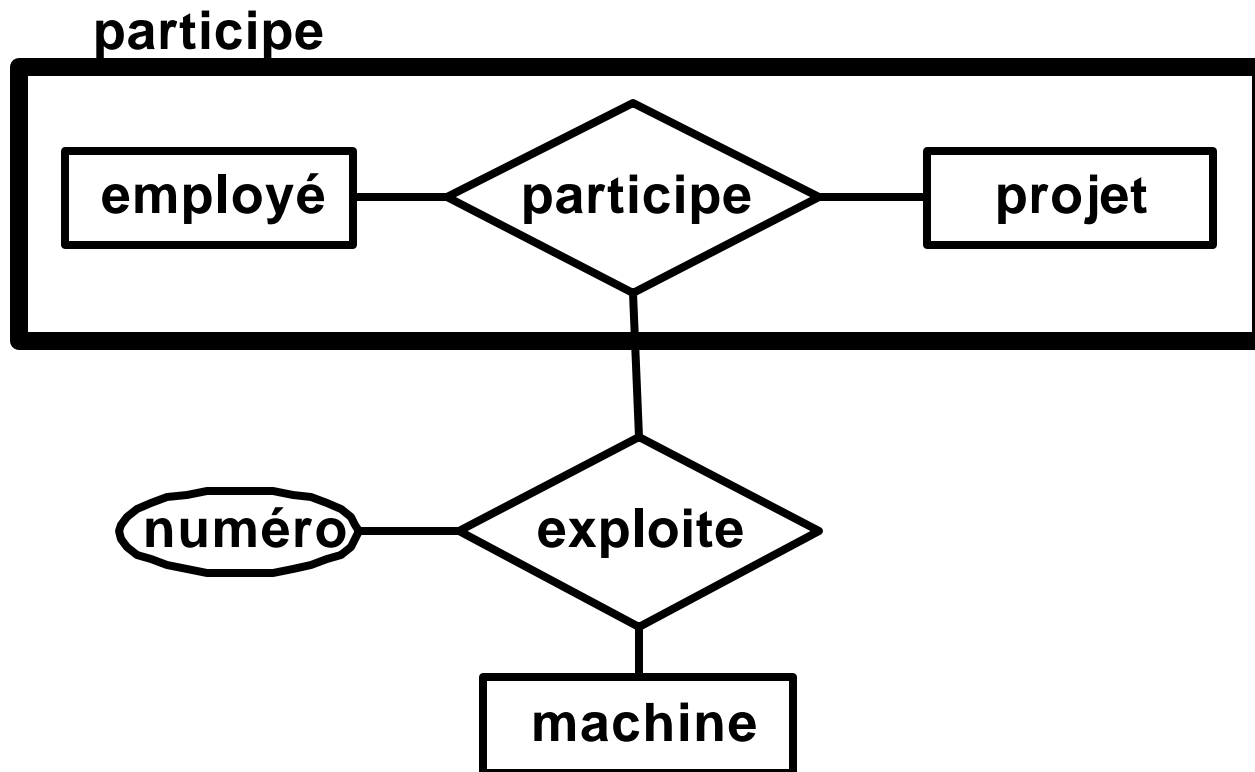
On pourrait être tenté de résoudre ce problème en créant un lien entre les deux relations.





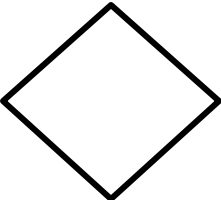
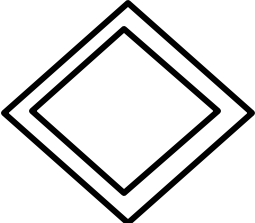
MAIS, le modèle E-R ne peut représenter des relations entre relations.

Agrégation

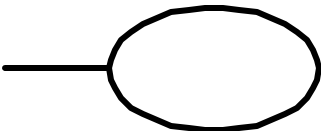
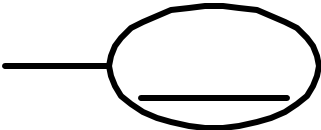
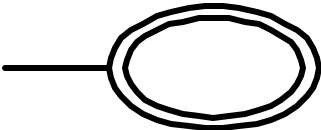
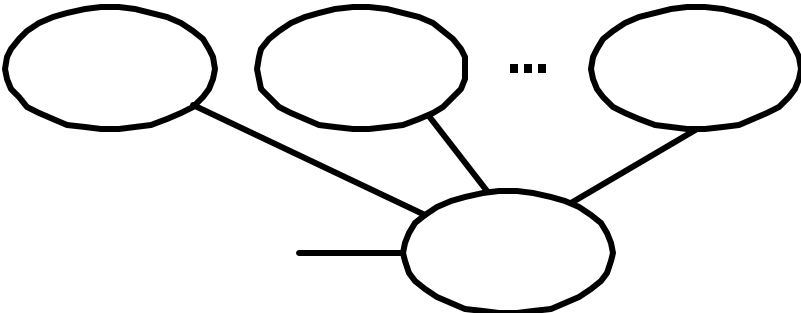
L'**agrégation** permet cette représentation en créant une entité spéciale (participe).




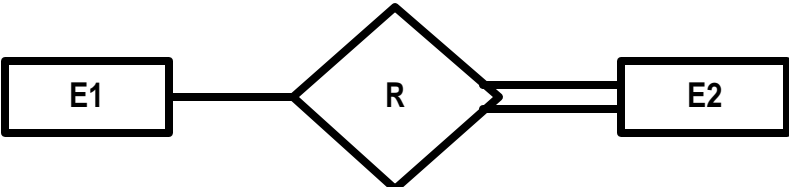
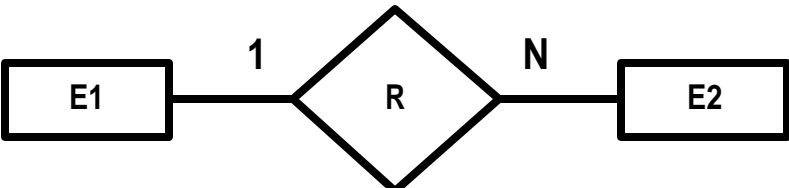
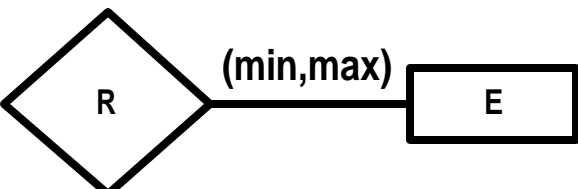
Notation graphique du modèle E-R

<u>Symbole</u>	<u>Signification</u>
	ENTITÉ
	ENTITÉ FAIBLE
	RELATION
	RELATION D'IDENTIFICATION (IDENTIFYING RELATIONSHIP)

Notation graphique du modèle E-R

<u>Symbole</u>	<u>Signification</u>
	ATTRIBUT
	ATTRIBUT CLÉ
	ATTRIBUT MULTIVALUÉ
	ATTRIBUT COMPOSITE

Notation graphique du modèle E-R

<u>Symbole</u>	<u>Signification</u>
	ATTRIBUT DÉRIVÉ
	PARTICIPATION TOTALE DE E2 DANS R
	RATIO DE CARDINALITÉ 1:N POUR E1:E2 DANS R
	CONTRAINTE DE PARTICIPATION (min, max) DE E DANS R

Processus de modélisation du monde réel en modèle conceptuel

Processus au tableau (Figure 3.3)

Énoncé de la base de données : COMPANY

Supposons que la première étape “ **Collection des besoins et analyse**” de la méthode de conception a été complétée et que la description fonctionnelle suivante en résulte.

Description fonctionnelle abrégée de la BD COMPAGNY

La BD COMPANY garde la trace des employés de la compagnie, des départements et des projets.

- *La compagnie est organisée en départements. Chaque **département** possède un nom, un numéro, et un employé qui gère le département. On garde la trace de la date du début de gestion du département par cet employé. Un **département** peut avoir plusieurs emplacements.*
- *Un département contrôle un nombre de **projets**. Chaque **projet** possède un nom, un numéro et un seul emplacement.*

Énoncé de la base de données : COMPANY

- Pour chaque **employé**, on doit stocker son nom, son NAS, adresse, salaire, sexe et date de naissance. Un **employé** est affecté à un seul **département**, mais peut travailler sur plusieurs **projets** qui ne sont pas nécessairement contrôlés par le même **département**. On garde trace du nombre d'heures par semaine qu'un **employé** met sur le **projet**. On garde trace aussi du superviseur direct de chaque **employé**.
- On veut garder trace des **dépendants** de chaque **employés** pour des fin d'assurance. On garde pour chaque **dépendant**: son nom, sexe, date de naissance et relation avec l'employé.

Design conceptuelle initiale de la BD COMPANY

À cette étape, on peut définir les types d'entités de la **BD COMPANY** et leurs attributs. Nous raffinerons par la suite notre design par l'introduction du concept Relation.

Selon les besoins décrits, on peut identifier 4 classes d'entités.

1. Une classe d'entité **DEPARTEMENT** avec les attributs: **Nom**, **Numéro**, **Emplacement**, **Directeur** et **Date de début du directeur**.
 - ***Emplacements** est le seul attribut à valeur multiple.*
 - *On peut spécifier que **Nom** et **Numéro** du département sont des **clés candidates** (deux départements ne peuvent avoir un même nom ou un même numéro).*

Design conceptuelle initiale de la BD COMPANY

2. Une classe d'entité **PROJET** avec les attributs: **Nom, Numéro, Emplacement et Département de contrôle.** **Nom** et **Numéro** sont des **clés candidates**.
3. Une classe d'entité **EMPLOYÉ** avec les attributs: **Nom, NAS, Sexe, Adresse, Salaire, Date de Naissance, Département et Superviseur.**
 - ***Nom** et **Adresse** peuvent être des attributs composites. Par contre, ceci n'a pas été spécifié dans les besoins. On doit retourner aux utilisateurs pour voir si l'un d'eux doit référer aux composantes individuelles pour*
 - ♦ **Nom** - Prénom, Initial, Nom de famille
 - ♦ et **Adresse** – Numéro, Rue, Ville, Code Postal, Pays.
4. Une classe d'entité **DÉPENDANT** avec les attributs **Employé, Nom du dépendant, Sexe, Date de Naissance et Relation.**

Design conceptuelle initiale de la BD COMPANY

Nous n'avons pas encore présenté le fait que :

- ◆ un **employé** peut travailler sur plusieurs projets
- ◆ le **nombre d'heures** par semaine qu'un employé travaille sur un projet

Deux choix s'offre à nous:

- *Nous pouvons le représenter comme **attribut composite à valeurs multiples** de **EMPLOYÉ** appelé **Travaille** et on appelle ses composantes (**Projet, Heures**).*

OU

- *On peut le représenter comme **attribut composite à valeurs multiples** de **PROJET** appelé **Travailleur** avec ses composantes (**Employé, Heures**).*

Nous retenons la 1ère alternative.

Design conceptuelle initiale de la BD COMPANY

Le tableau suivant présente le **Design Conceptuel préliminaire** des classes d'entités de la BD COMPANY

Entités	Attributs
DEPARTEMENT	Nom, Numéro, {Emplacements}, Directeur, DateDébutDirecteur
PROJET	Nom, Numéro, Emplacement, DépartementDeContrôle
EMPLOYÉ	Nom(Prénom, Initial, NomFamille), NAS, Sexe, Adresse, Salaire, DateNaissance, Département, Superviseur, {Travail(Projet, Heures)}
DÉPENDANT	Employé, NomDépendant, Sexe, DateNaissance, Relation

Dans le tableau ci-dessus, les **attributs à valeurs multiples** sont entre **accolades**. Les **composantes** des attributs composites sont entre **parenthèses**.

Design conceptuelle initiale de la BD COMPANY

Dans le design conceptuel préliminaire de la BD COMPANY, il y a plusieurs relations implicites entre les classes d'entités.

Chaque fois qu'un attribut d'une classe d'entité réfère à une autre classe d'entité, une **relation** existe.

Ex:

- ***Directeur** du **DÉPARTEMENT** réfère à **EMPLOYÉ** qui est à la tête du département.*
- ***DépartementDeContrôle** du **PROJET** réfère au département contrôlant le projet.*

Dans le modèle E-R, ces références ne doivent pas être représentées comme des attributs, mais plutôt comme des relations.

RAFFINEMENT DU DESIGN E-R POUR LA BD COMPANY

On peut maintenant raffiner le design de la BD COMPAGNY en changeant les **attributs** qui représentent des relations en **classe de relations**.

Les **contraintes de ratio de cardinalité** et de **participation** de chaque **classe de relations** sont déterminées à partir de la liste des besoins.

- *Si on n'est pas capable de déterminer certaines contraintes à partir de la liste des besoins, alors on retourne voir les utilisateurs.*

RAFFINEMENT DU DESIGN E-R POUR LA BD COMPANY

Dans notre exemple, on spécifie les classes de relations suivantes :

1. **GÈRE**, un type de relations **1:1** entre **EMPLOYÉ** et **DÉPARTEMENT**.
 - ◆ *La participation de EMPLOYÉ est **partielle**.*
 - ◆ *La participation de DÉPARTEMENT n'est pas claire à partir de l'étude des besoins. On retourne voir les utilisateurs, qui disent qu'un département doit avoir tout le temps un directeur. Ce qui implique une participation **totale**.*
 - ◆ *L'attribut **DateDébut** est affecté à cette classe de relations.*
2. **TRAVAILLE_POUR**, un type de relations **1:N** entre **DÉPARTEMENT** et **EMPLOYÉ**.
 - ◆ *Les deux participations sont **totales**.*

RAFFINEMENT DU DESIGN E-R POUR LA BD COMPANY

3. **CONTRÔLE**, un type de relations **1:N** entre **DÉPARTEMENT** et **PROJET**.
- *La participation de PROJET est **totale**.*
 - *La participation de DÉPARTAMENT est **partielle**. (Celle-ci sera déterminée ainsi après consultation des utilisateurs.)*
4. **SUPERVISION**, un type de relations **1:N** entre **EMPLOYÉ** (dans le rôle de superviseur) et **EMPLOYÉ** (dans le rôle de supervisé).
- ♦ *Les deux participations sont déterminées comme **partielles** après que les utilisateurs indiquent que pas tous les employés sont superviseurs et pas tous les employés possèdent un superviseur.*

RAFFINEMENT DU DESIGN E-R POUR LA BD COMPANY

5. **TRAVAILLE_SUR**, un type de relations **M:N** avec l'attribut **Heures**, après que les utilisateurs indiquent qu'un projet peut avoir plusieurs employés travaillant dessus.

◆ *Les deux participations sont **totales**.*

6. **DÉPEND**, un type de relations **1:N** entre **EMPLOYÉ** et **DÉPENDANT**, qui est la relation identifiante de la classe d'entités faibles **DÉPENDANT**.

◆ *La participation de **EMPLOYÉ** est **partielle**.*

◆ *La participation de **DÉPENDANT** est **totale**.*

Diagramme E-R de la base de données COMPANY.

