

## Chapitre 4

### Le modèle **relational**

Le modèle **relational** a été proposé par Codd au début des années 70, avec comme objectif essentiel l'accroissement de l'indépendance des programmes vis-à-vis de la représentation des données.

#### I. Concepts de base

##### *I.1. Domaine*

Un domaine est un ensemble de valeurs.

Exemple:

Domaines  $D1 = \{\text{entiers}\}$ .

$D2 = \{\text{chaînes de caractères}\}$ .

##### *I.2 Attribut*

Un attribut est une variable prenant ses valeurs dans un domaine.

Exemple :

attribut  $A1 = \text{NCI}$  à valeurs dans  $D1$ ,

attribut  $A2 = \text{NomCl}$  à valeurs dans  $D2$ ,

attribut  $A3 = \text{AdrCl}$  à valeurs dans  $D2$ .

##### *I.3 Relation*

Une relation  $n$ -aire sur les attributs  $A1, A2, \dots, An$ , de domaines respectifs  $D1, D2, \dots, Dn$ , est un sous-ensemble du produit cartésien des domaines  $D1, D2, \dots, Dn$ .

Un élément appartenant à une telle relation sera appelé  $n$ -uplet ou Tuple. Il sera noté  $(d1, d2, \dots, dn)$  où  $di \in Di, \forall 1 \leq i \leq n$ .

L'ensemble des  $n$ -uplets d'une relation sera appelé extension de la relation.

Exemple:

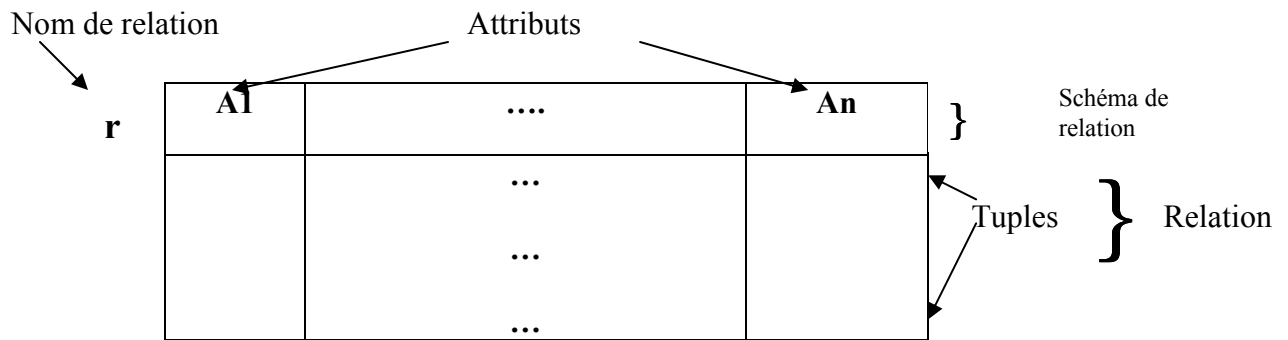
Soit  $r$  une extension constituée de trois tuples :

$r = \{(2140, \text{ALI}, \text{TUNIS}), (1123, \text{SALAH}, \text{SFAX}), (3425, \text{MHAMED}, \text{GABES})\}$ .

##### *I.4 Représentation d'une relation*

Chaque tuple ( $n$ -uplet) de la relation (appelée aussi Table) est écrit dans une ligne d'un tableau, dont les noms de colonnes sont les attributs de la relation.

Chaque tuple est unique. Les duplications ne sont pas autorisées. L'ordre des tuples est indifférent.



Exemple:  
Table Client :

NCI	NomCI	AdrCI
2140	ALI	TUNIS
1123	SALAH	SFAX
3425	MHAMED	GABES

### 1.5 Schéma de relation /Contraintes d'intégrité

Le schéma **R** d'une relation **r** est la liste des attributs de **r** avec, pour chacun, son domaine, parfois sous-entendu.

Exemple:

Le schéma de **r** (voir exemple paragraphe précédent) est **R** = (NCI : D1, NomCI : D2, AdrCI: D2), écrit en abrégé **R** = (NCI, NomCI, AdrCI).

On dit que **r** est une relation de schéma **R** (attention : *ne pas confondre r et R*).

Chaque relation est vue comme un tableau dont les colonnes désignent les attributs et les lignes les n-uplets.

Lorsqu'on repère chaque attribut d'une relation par un nom, l'ordre des colonnes n'est pas important.

On pourra parfois attacher à une relation un ensemble **P** de propriétés que doit vérifier chacun de ses tuples. Ces propriétés sont appelées **Contraintes d'intégrité**. A un schéma de relation sont donc rattachées des contraintes d'intégrité.

Exemples

CLIENT(NCI, NomCI, AdrCI, DateNaissance)

Pour ce schéma de relation, la date de naissance du client doit être inférieure à la date du jour.

COMMANDE (NCmd, DateCmd, DateLivr)

Pour ce schéma de relation, la date de la livraison (DateLivr) doit être supérieure à la date de la commande (DateCmd).

### 1.6 Clé d'une relation

Une des contraintes d'intégrité d'un schéma est l'unicité d'identification des n-uplets d'une relation. Cette identification unique est assurée par la notion de **clé** de relation.

Une clé peut être composée d'un seul attribut ou d'une liste d'attributs qui caractérise un tuple (n-uplet) de la relation de manière unique.

Une relation peut avoir plusieurs clés. Une clé comportant un minimum d'attributs sera choisie comme étant **clé primaire**, les autres clés possibles sont appelées **clés candidates**. Par convention, la clé primaire d'une relation est soulignée dans un schéma de relation.

Exemple: Client (NCl, NomCl, PrenomCl, AdrCl)

NCl	NomCl	PrenomCl	AdrCl
2140	ALI	Mohamed	TUNIS
1123	SALAH	Ali	SFAX
453	SALAH	Zied	TUNIS
3425	MHAMED	Fatma	GABES

- (NCl), (NomCl, PrenomCl) sont des clés.
- (NCl) est clé primaire.
- (NomCl, PrenomCl) est une clé candidate. Par contre (NomCl) n'est pas une clé à elle seule.

Remarque : si on choisit pour clé (NomCl, AdrCl), la modélisation ne permet pas des homonymes habitant la même ville.

## I.7 Clé étrangère

### Définition

Une clé étrangère est un ensemble d'une ou de plusieurs colonnes d'une table qui fait référence à une clé primaire d'une autre table. Toutes les valeurs des clés étrangères apparaissent dans une autre relation comme valeurs d'une clé. Par convention, la clé étrangère d'une relation précédée par le symbole # dans un schéma de relation.

### Exemple

Soient les schémas de relations suivants

Client(NCl, NomCl, AdrCl)

- Désigne l'ensemble des clients.

Commande (NCmd, DateCmd, #NCl)

- Désigne l'ensemble des commandes.

L'attribut NCl dans la table Commande est une clé étrangère. Il prend ses valeurs dans le domaine de valeurs de l'attribut NCl qui se trouve, dans le schéma de relation Client. Une commande est toujours passée par un Client existant dans la base de données.

## I.8 Schéma de base de données relational le

Une base de données **relational** le est une collection de relations. L'ensemble des schémas des relations de la collection est appelé schéma **relational** de la base. Formellement, un schéma de base de données **relational** le B est un ensemble de schémas de relations R1, R2,..., Rp. Une base de données b de schéma B est un ensemble de relations r1, r2, ..., rp de schémas respectifs R1, R2, ..., Rp.

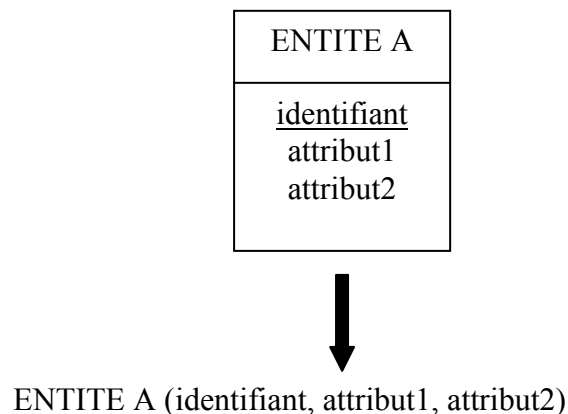
## II. Traduction d'un modèle Entité - Association en modèle relational

L'absence de SGBD supportant directement le modèle Entité-Association amène à transformer le schéma conceptuel en un schéma conforme au modèle de données du SGBD cible. Dans ce paragraphe, nous présentons les règles qui permettent de passer d'un schéma Entité-Association à un schéma **relational**.

### II.1 Traduction des entités

Toute entité est traduite selon les trois règles suivantes :

- L'entité se transforme en une relation.
- L'identifiant de l'entité devient la clé primaire de la relation.
- Les propriétés de l'entité deviennent des attributs de la relation.



### II.2 Traduction des associations

Nous distinguons deux catégories d'associations : les associations binaires et les associations n-aires. La traduction d'une association s'effectue selon les cardinalités relatives aux entités participant à l'association. Plusieurs cas peuvent se présenter.

#### a. Traduction des associations binaires

Soient deux entités A et B reliées par une association AssAB

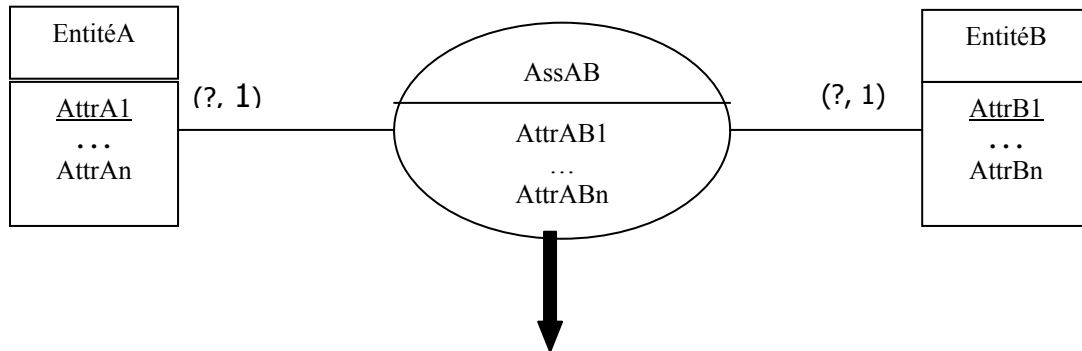
##### Cas1 : Association Un-à-Un

**Cardinalité entité A 0, 1 ou 1, 1 et Cardinalité entité B 0, 1 ou 1, 1**

Pour ce type d'association deux traductions sont possibles :

- **Solution1** : Les deux entités et l'association seront transformées en une seule relation contenant les attributs des deux entités ainsi que les attributs éventuels de l'association, l'identifiant de l'entité A ou de l'entité B sera choisie comme clé primaire de la nouvelle relation. Cette solution est surtout utilisée dans le cas où les deux entités ont des cardinalités 1,1 qui ne sont pas sujettes à des modifications dans le temps (Voir exemple).
- **Solution2** : Les deux entités seront transformées en deux relations. Une de ces deux relations sera choisie et étendue par la liste des attributs éventuels de

l'association ainsi que de l'identifiant de l'autre entité en tant que clé étrangère. Ce choix se base sur la séquence temporelle de création des entités. L'entité qui sera créée en second lieu aura comme clé étrangère l'identifiant de l'entité créée en premier lieu. Cette solution est la plus adaptée dans le cas où une ou les deux cardinalités minimales sont nulles.



Solution1 :

$R(\underbrace{AttrA1, \dots, AttrAn}_{\text{Attributs de l'Entité A}}, \underbrace{AttrB1, \dots, AttrBn}_{\text{Attributs de l'Entité B}}, \underbrace{AttrAB1, \dots, AttrABn}_{\text{Attributs de l'association}})$

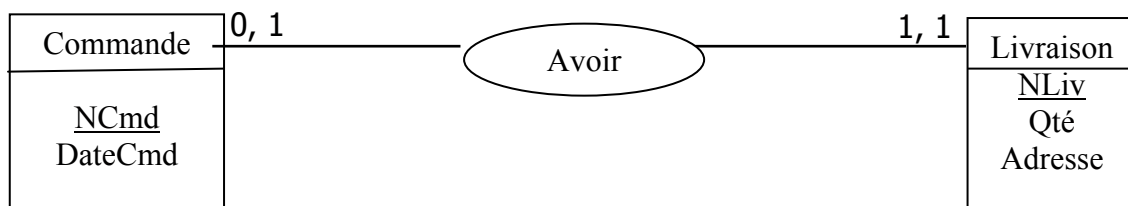
Solution2 :

EntitéA(AttrA1, ..., AttrAn)  
 EntitéB(AttrB1, ..., AttrBn,  $\#AttrA1$ ,  $AttrAB1, \dots, AttrABn$ )

$\downarrow$                        $\downarrow$   
 Clé étrangère              Attributs de l'association

Exemples :

Exemple1



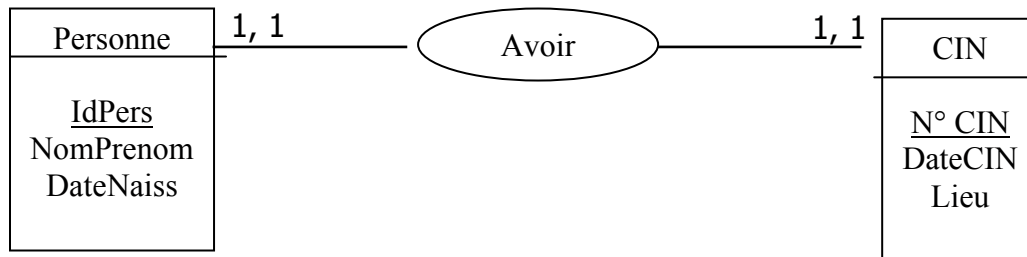
Le modèle **relational** correspondant est le suivant :

**Commande** (NCmd, DateCmd)  
**Livraison** (NLiv, Qté, Adresse,  $\#NCmd$ )

$\underbrace{\hspace{10em}}$   
 Clé étrangère

La relation 'Livraison' a comme clé étrangère l'identifiant de 'Commande' car la création d'une livraison survient après la création d'une commande.

### Exemple2



Le modèle **relational** correspondant est le suivant :

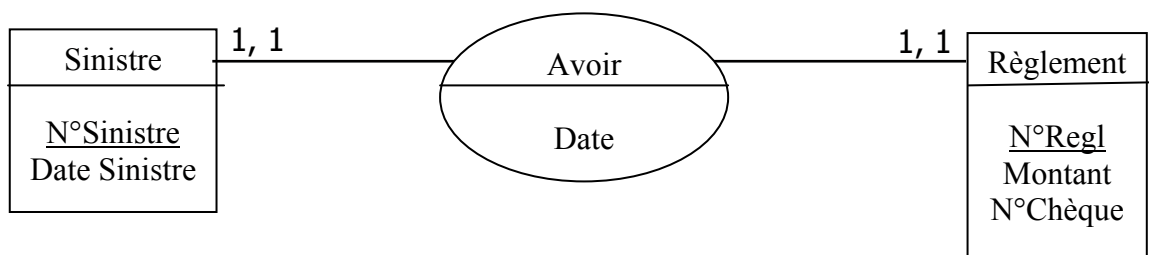
**Personne** (IdPers, NomPrenom, DateNaiss)

**CIN** (N° CIN, DateCIN, Lieu, # IdPers)

Clé étrangère

La relation 'CIN' a comme clé étrangère l'identifiant de 'Personne' en supposant que la création d'une CIN survient après la création d'une personne. Il est possible également d'utiliser la deuxième solution et de fusionner les deux tables 'Personne' et 'CIN' car les cardinalités 1,1 de chaque côté ne risquent pas de changer dans le temps. En effet, une personne a une et une seule CIN et une CIN correspond à une et une seule personne; et cette règle ne risque pas de changer dans l'avenir.

### Exemple3



Le modèle **relational** correspondant est le suivant :

**Sinistre** (N°Sinistre, Date Sinistre)

**Règlement** (N°Regl, Montant, N° Chèque, Date, # N°Sinistre)

Clé étrangère

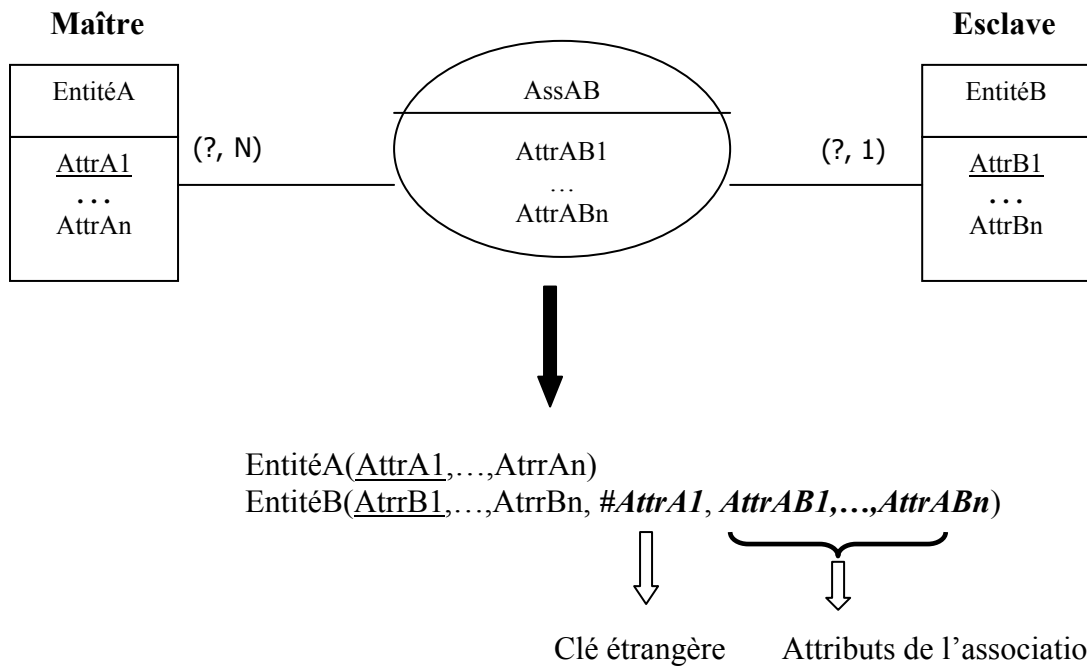
La relation 'Règlement' a comme clé étrangère l'identifiant de 'Sinistre' car un règlement fait obligatoirement référence au sinistre qui lui a donné naissance.

## Cas2 : Association Un-à-plusieurs (Maître-Esclave) :

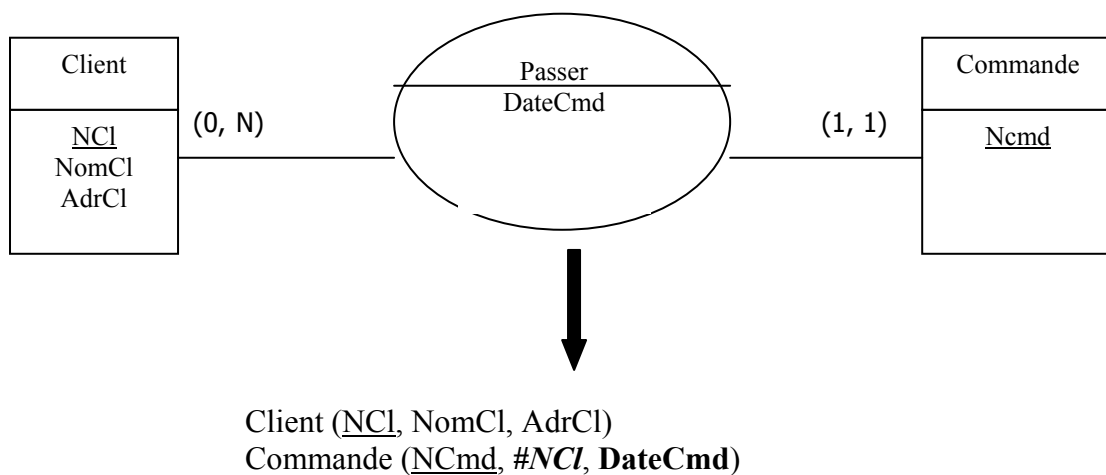
Cardinalité entité A (Maître) 0, N ou 1, N et Cardinalité entité B (Esclave) 0, 1 ou 1, 1

Les règles de traduction de ce type d'association sont les suivantes :

- L'entité Maître (Entité A) devient la relation Maître.
- L'entité Esclave (Entité B) devient la relation Esclave.
- L'identifiant de l'entité Maître devient attribut de la relation Esclave. Cet attribut est désigné comme clé étrangère.
- Les attributs éventuels de l'association (AssAB) migrent vers la relation esclave et deviennent ses attributs.



## Exemple

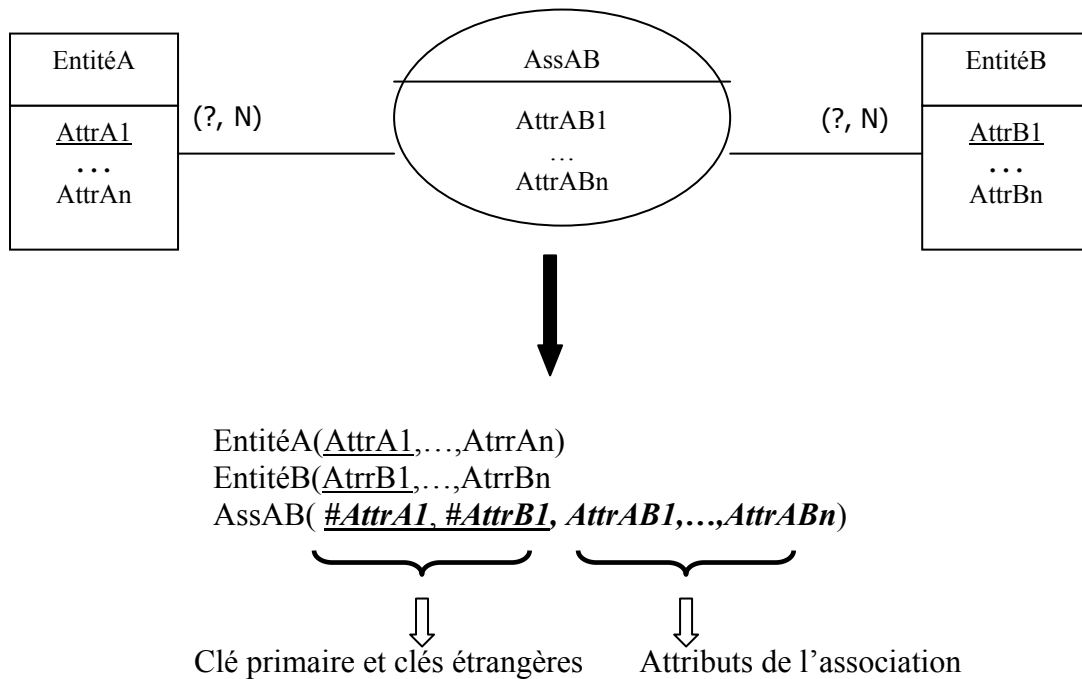


### Cas 3 : Association Plusieurs-à-Plusieurs:

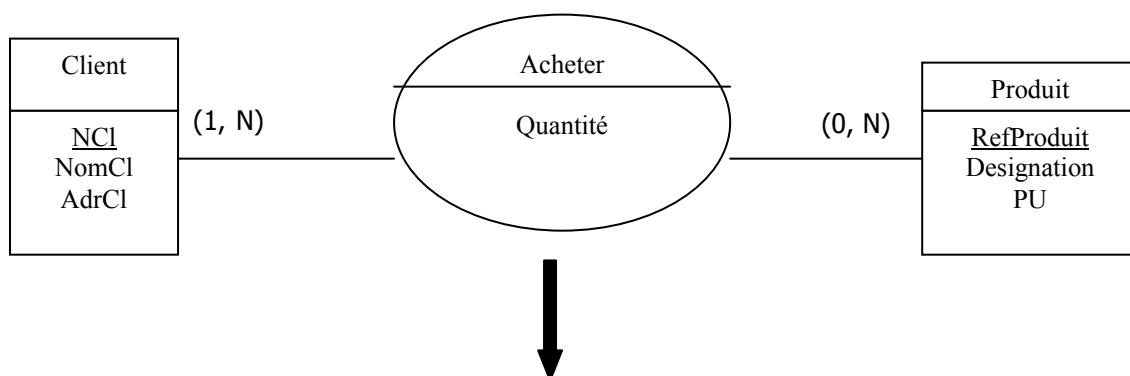
Cardinalité entité A 0, N ou 1, N et Cardinalité entité B 0, N ou 1, N

Les règles de traduction de ce type d'association sont les suivantes :

- Chaque entité (Entité A et Entité B) devient une relation.
- L'association sera transformée aussi en une relation ayant comme clé la concaténation des deux clés issues des entités A et B. Les attributs éventuels de l'association seront stockés dans cette relation en tant qu'attributs.



### Exemple :



Le modèle **relational** correspondant est le suivant :

**Client** (NCI, NomCl, AdrCl)

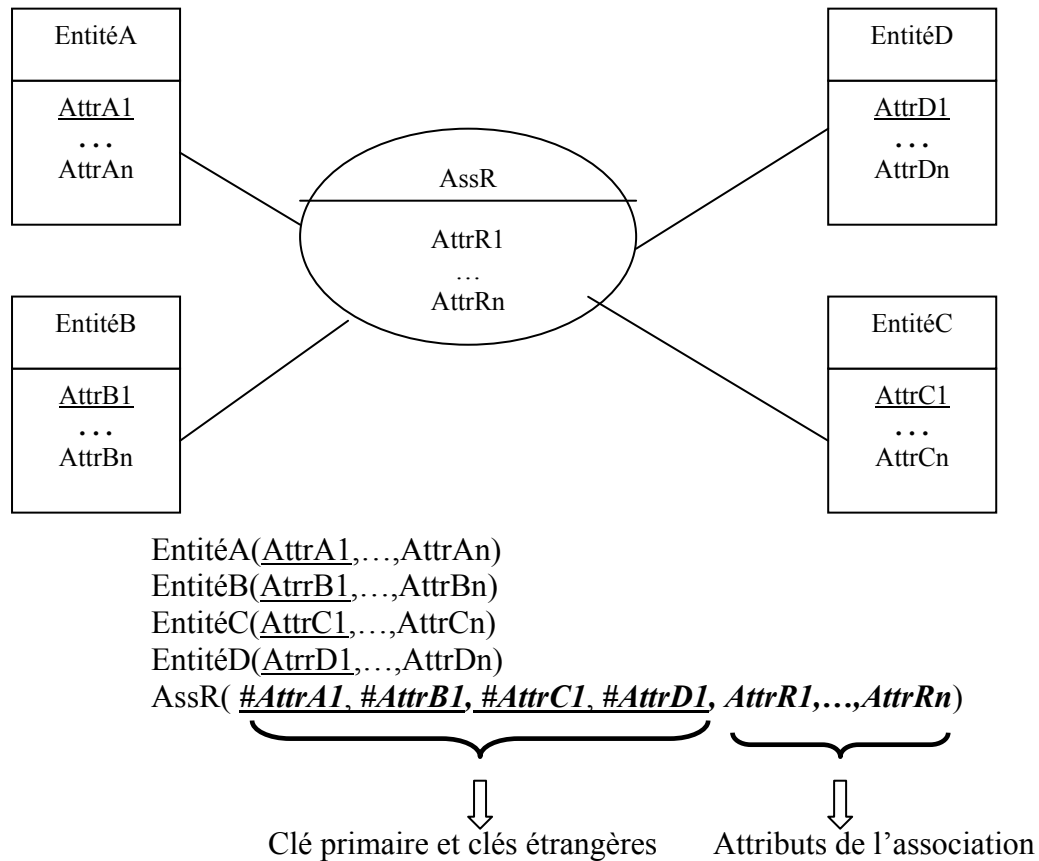
**Produit** (RefProduit, Designation, PU)

**Acheter** (#NCI, #RefProduit, Quantité)



### b. Traduction des associations n-aires

Ce type d'association sera transformé en une relation ayant comme liste d'attributs la liste des clés des relations correspondantes aux entités qui participent à cette association en plus de ses attributs éventuels. Une clé minimale sera choisie parmi la liste des attributs ainsi constituée.



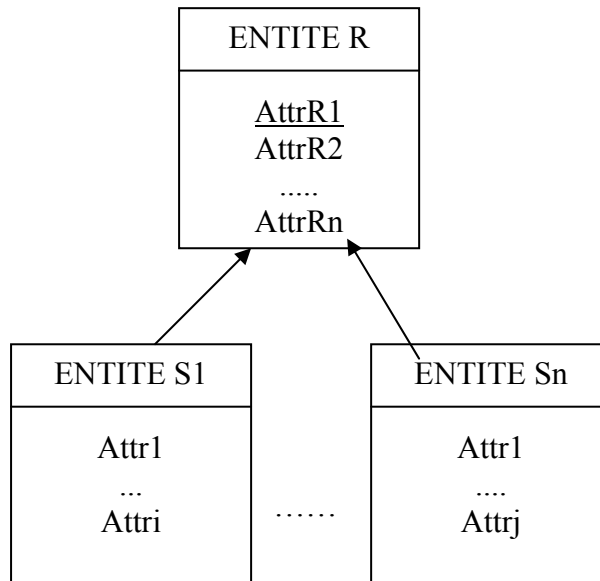
### II.3 Traduction du lien is-a

La traduction du lien is-a peut se faire selon plusieurs règles. Dans ce qui suit, nous considérerons une entité mère R avec n entités filles S1, S2, ..., Sn.

La traduction d'un lien is-a se fait selon l'une des trois règles suivantes :

#### R1 : Représentation de l'entité mère et de ses entités filles

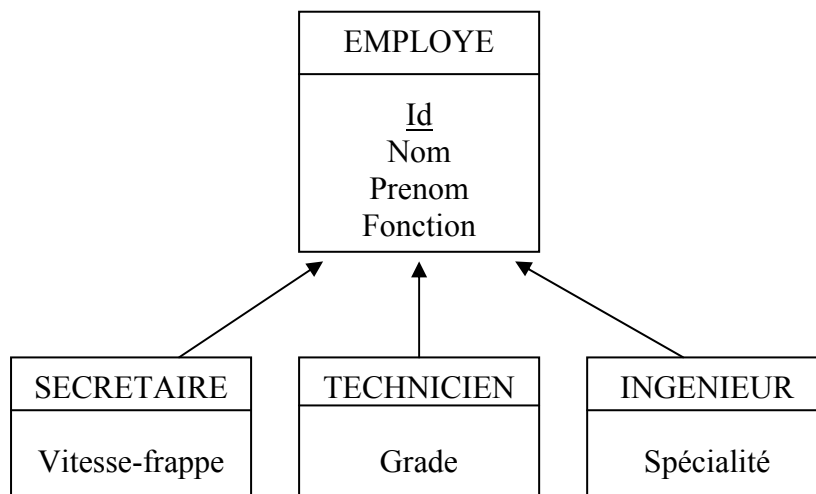
- L'entité mère sera transformée en une nouvelle relation avec ses attributs.
- Chaque entité fille Si sera transformée en une relation comportant comme clé primaire l'identifiant de l'entité mère et comme attributs les attributs de Si.



EntitéR(AttrR1,...,AttrRn)  
 EntitéS1(#AttrR1,Attr1,...,Attri)  
 ...  
 EntitéSn(#AttrR1,Attr1,...,Attrj)

Cette règle est adaptée pour tout type de spécialisation ce qui permettra de représenter l'entité mère et les entités filles explicitement.

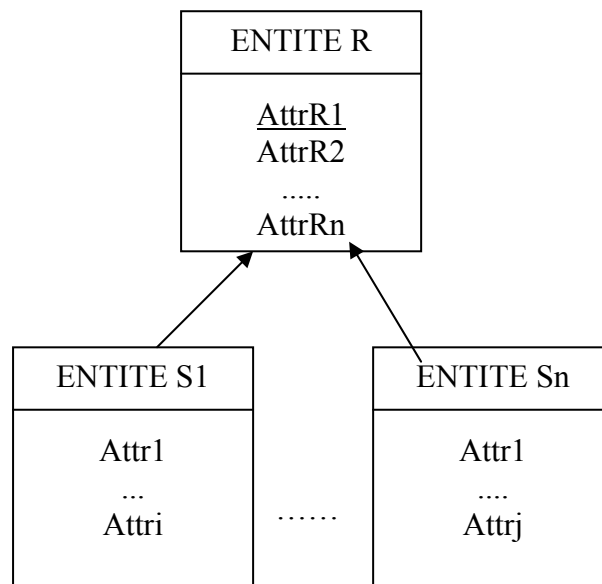
### Exemple



EMPLOYE(Id, Nom, Prénom, Fonction)  
 SECRETAIRE(#Id, Vitesse-frappe)  
 TECHNICIEN(#Id, Grade)  
 INGENIEUR(#Id, Spécialité)

## R2 : Pas de représentation de l'entité mère

Chaque entité fille Si sera transformée en une relation comportant comme clé primaire l'identifiant de l'entité mère et comme attributs les attributs de Si en plus des attributs de l'entité mère.

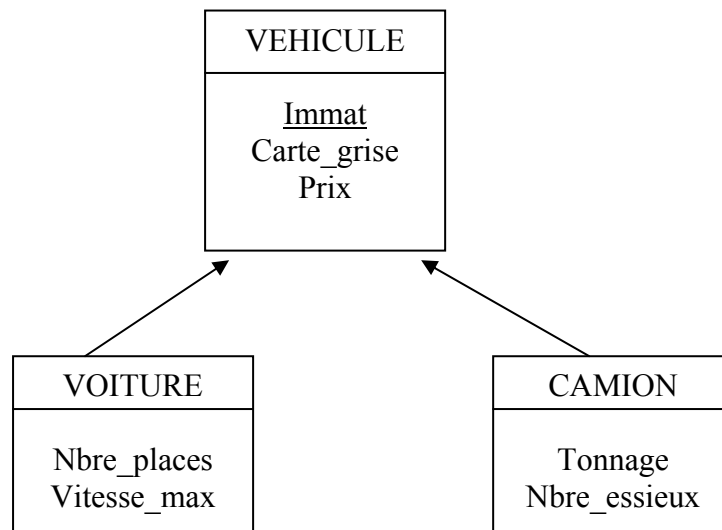


EntitéS1(AttrR1,Attr1,...,Attri, AttrR2,...,AttrRn)

...

EntitéSn(AttrR1,Attr1,...,Attrj, AttrR2,...,AttrRn)

## Exemple



VOITURE(Immat, Carte\_grise, Prix, Nbre\_place, Vitesse\_max)

CAMION(Immat, Carte\_grise, Prix, Tonnage, Nbre\_essieux)

Cette règle pose un problème lorsque les sous-entités ne sont pas disjointes. Dans ce cas, il peut y avoir duplication de certaines données. Certains problèmes d'incohérence peuvent alors avoir lieu.

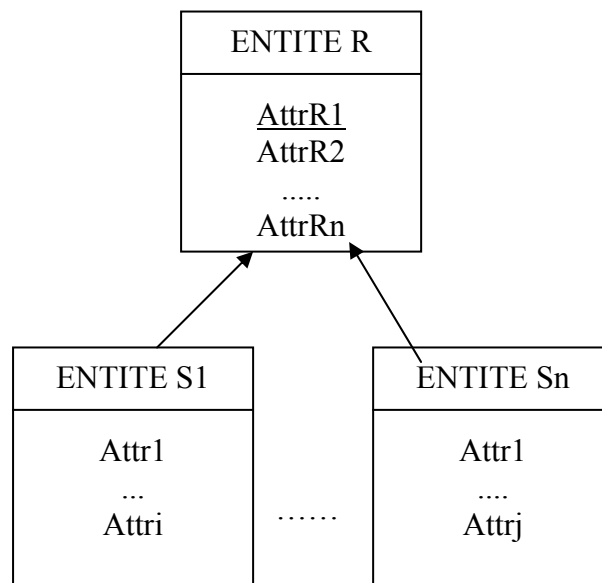
Cette règle est applicable donc, dans le cas de sous-entités sont totalement disjointes, tels que Homme, Femme → Personne

ou aussi, Alimentaire, Habillement, Electroménager → Article.

Pour le cas, Etudiant, Employé → Personne cette règle conduirait à dupliquer les données héritées pour des employés étudiants.

### R3 : Fusion des entités filles et de l'entité mère

L'entité mère et ses entités filles seront transformées toutes en une seule relation ayant comme clé primaire l'identifiant de l'entité mère et comme attributs les attributs de toutes les entités (mère et filles).



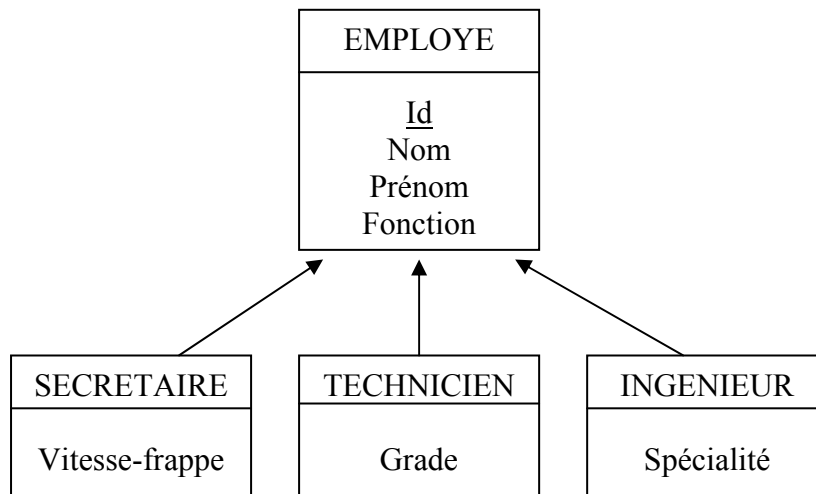
$R(\underline{AttrR1}, AttrR2, \dots, AttrRn, Attr1, \dots, Attri, Attr1, \dots, Attrj)$

Le problème posé par cette règle est que certains attributs risquent d'avoir une valeur nulle.

Par exemple, pour la hiérarchie Homme, Femme → Personne, suite à l'utilisation de cette règle les attributs spécifiques aux hommes seront nuls pour les femmes et vice versa.

En utilisant cette règle par exemple pour la hiérarchie Etudiant, Employé → Personne, tout étudiant non employé aura les attributs spécifiques aux étudiants nuls, et tout employé non étudiant aura les attributs d'étudiants nuls.

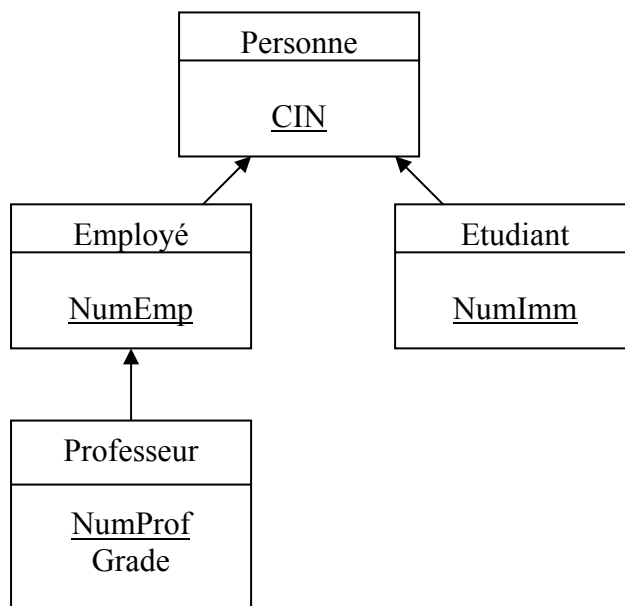
### Exemple 1



EMPLOYE(Id, Nom, Prénom, Fonction, Vitesse-frappe, Grade, Spécialité)

### Exemple 2

Soit l'exemple suivant :



Pour traduire cette hiérarchie nous utilisons deux règles :

Pour le deuxième niveau de la hiérarchie Professeur Employé nous pouvons utiliser la troisième règle et nous obtiendrons la relation suivante :

Employé (NumEmp, NumProf, Grade)

Pour le premier niveau de la hiérarchie nous utilisons la première règle, nous obtiendrons alors comme modèle **relational** final :

Personne (CIN)

Employé (#CIN, NumEmp, NumProf, Grade)

Etudiant (#CIN, NumImm)

### ***III. Conclusion***

Ce chapitre a été consacré à la présentation des principaux concepts du modèle **relational** et aux règles de passage du modèle entité association au modèle **relational** . Le chapitre suivant sera dédié à l'algèbre **relational** et ses langages prédicatifs.

## Exercices

### Exercice 1 :

Au niveau d'un faculté, on dispose d'un réseau Intranet comportant un ensemble d'informations utiles aux différents utilisateurs. Afin d'assister les étudiants et les aider à comprendre leurs cours, chaque enseignant met à leur disposition les cours qu'il enseigne ainsi qu'un ensemble d'exercices qui ont été proposés dans des examens antérieurs (précédents) relativement à ces cours.

Un cours est relatif à une matière identifiée par un code et décrite par une désignation. Une matière peut avoir plusieurs cours. Ce dernier est identifié par un code et décrit par un titre et une adresse sur Intranet.

Un cours peut être élaboré par plusieurs enseignants. Un enseignant est identifié par son numéro de carte d'identité et il est décrit par son nom, son grade et sa spécialité.

A un cours, sont associés plusieurs examens dont chacun est identifié par un numéro, un type et une date de déroulement. Chaque examen comprend plusieurs exercices. Chacun est identifié par un numéro et possède un barème relativement à un examen. Il est à noter qu'un exercice peut être repris dans plusieurs examens.

❶ Construire le modèle E/A relatif à cette base en donnant les différentes entités, associations et attributs.

❷ Traduire ce modèle selon les règles du modèle **relational**.

### Exercice 2

Soit la base de données suivante :

**Bus** (Codb, Marque, Modèle, Nmat, Nbpas, Nbpdeb)

**Service** (Cods, Desgs)

**Employé** (Codemp, Nom, Adr, Fct, Datrec)

**Affectation\_bus** (#Codb, #Cods, DataffB)

**Affectation\_employé** (#Codemp, #Cods, DataffE)

**Ligne** (Codl, Desgl, #Cods, Nbkm)

#### Description de la base de données

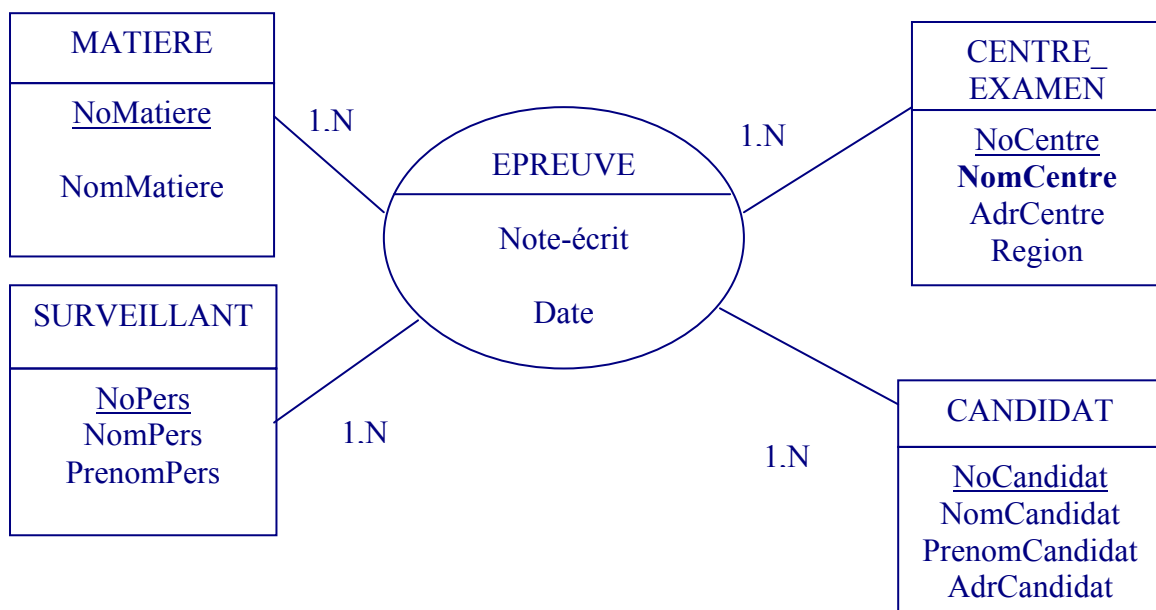
Code	Description
<b>CODB</b>	Code du bus
Marque	Marque du bus
Modèle	Modèle du bus
Nmat	Numéro d'immatriculation du bus
Nbpas	Nombre de places assises d'un bus
Nbpdeb	Nombre de places debout d'un bus
Cods	Code du service
Desgs	Désignation du service

Codemp	Code de l'employé
Nom	Nom de l'employé
Adr	Adresse de l'employé
Fct	Fonction de l'employé (Chauffeur ou contrôleur)
Datrec	Date de recrutement de l'employé
DataffB	Date d'affectation du bus à un service
DataffE	Date d'affectation de l'employé à un service
Codl	Code de la ligne
Desgl	Désignation de la ligne
Nbkm	Nombre de kilomètres de parcours d'une ligne

Déduire à partir du modèle **relational** ci-dessus, le modèle E/R correspondant.

### Exercice 3

Transformer ce modèle entité/Association en un modèle **relational**.



### Exercice 4

On souhaite informatiser une partie de la gestion d'une école.

Le personnel de cette école est composé de professeurs et de secrétaires. Chaque membre du personnel est identifié par un numéro matricule, par son nom, son prénom et son adresse.

L'école est composée de locaux (identifiés par un numéro) qui sont soit des bureaux (dans ce cas ils sont pourvus d'un unique téléphone), soit des salles de cours (qui comprennent un certain nombre de places). Un bureau est occupé par un professeur et/ou plusieurs secrétaires.

Un étudiant est doté d'un numéro matricule étudiant. On souhaite également disposer dans la base de données, des nom, prénom et adresse des étudiants inscrits dans l'école.

Un étudiant s'inscrit dans une unique année d'étude, identifiée par un code et un nom.



Le programme d'une année d'étude consiste en un ensemble de cours (décrits par un code , un intitulé, et un nombre d'heures). Un cours peut regrouper plusieurs années d'étude.  
On souhaite voir figurer dans la base de données le titulaire du cours.

- ❶ Construire le modèle E/A relatif à cette base en donnant les différentes entités, associations et attributs.
- ❷ Traduire ce modèle selon les règles du modèle **relational** .