

5 Concevoir et raffiner des schémas de bases

5.1 Les étapes de la conception d'un schéma relationnel

1. Analyse des besoins (informelle)
2. Utilisation d'un outil "graphique" pour une première modélisation : méthode *Entité Association* (EA ou ER).
3. Passage de la représentation graphique au schéma relationnel (avec ses *dépendances fonctionnelles* (DF, une classe de contraintes sur les données)) : presque mécanique.
4. Analyse du schéma \mathcal{S} obtenu. Satisfaisant ? Si oui, terminé. SINON, *décomposition* de \mathcal{S} selon certains critères (formels), qui utilisent les DF.

5.2 Dépendances fonctionnelles

Notion préliminaire fondamentale pour la problématique de la conception de schéma.

Les DF jouent déjà un rôle dans la phase EA :

par exemple, fixer la clé “primaire” d’une table R = fixer certaines DF qui devront être satisfaites par le contenu de la table.

5.2.1 Notions de base sur les Dépendances Fonctionnelles

- **Dépendances fonctionnelles** : un type de contraintes d'intégrité. Participent à la spécification du format des données.
- X et Y ensembles d'attributs inclus dans le schéma d'une table R . Ecriture d'une dép.fonct. :

$$X \rightarrow Y$$

- Lecture : X détermine (ou donne) Y .

Aussi : Y dépend fonctionnellement de X . Signification : La table r dont le nom est R satisfait $X \rightarrow Y$ ssi :

qqques soient les n -uplets t, t' de r , si $t(X) = t'(X)$ alors $t(Y) = t'(Y)$.

(Notation : $r \models X \rightarrow Y$).

- Convention : si $X = \{A_1, \dots, A_n\}$ et $Y = \{B_1, \dots, B_m\}$ on peut écrire, si pas de confusion possible :

$A_1 \cdots A_n \rightarrow B_1 \cdots B_m$ (pas de $\{, \}$, pas de virgule).

Exemple. Une table possible pour Films :

| Titre | Date | Long | Couleur | NomStudio | IdProd |
|---------------------|------|------|---------|-----------|--------|
| King Kong | 1933 | 100 | False | RKO-Pathé | 601 |
| King Kong | 2005 | 187 | True | Universal | 798 |
| Million Dollar Baby | 2004 | 142 | True | Warner | 900 |
| Les Choristes | 2004 | 152 | True | Pathé | 886 |

Titre → *Couleur* n'est pas satisfaite.

Titre Date → *Couleur* est satisfaite.

Titre Date → *Couleur NomStudio* est satisfaite.

Titre Date → *Titre Date Long NomStudio Couleur IdProd* est satisfaite.

- Associer un ensemble de DF à un schéma de base S sert aussi à fixer en avance les *clés* des tables.
- Clé d'une table r ?? Intuition : identifiant de chaque n -uplet de r .
- Formellement : si S est un schéma d'une relation, r une relation sur S et $X \subseteq S$ alors :
 X est une clé pour la table r ssi
 1. $r \models X \rightarrow S$
 2. X est minimal par rapport à cette propriété : $\nexists Z$ t.q. $Z \subset X$ et $r \models Z \rightarrow S$
- **Exemple précédent** : *Titre Date* est une clé, *Titre* seul non, *Date* seul non plus.
Titre Date Couleur n'est pas une clé mais c'est une *super clé*.

Problème : Comment fixer par avance, les clés d'une table R à partir du schéma de R et des DF ?

- **Exemple**. Schéma(R)= $\{A, B, C, D\}$, $F = \{AB \rightarrow C, C \rightarrow D\}$.

La dép. fonct. $AB \rightarrow A B C D$ est **impliquée** par F .

Donc, AB sera une clé de toute table associée à R satisfaisant toutes les dépendances de F .

(Question : Une clé ou bien une super clé ?)

- **“Impliquée”** = ??

Déf. : Soit F un ensemble de dépendances fonctionnelles et f une dép.

fonct. On dit que **F implique f** (f est impliquée par F) ssi toute relation r satisfaisant toutes les dép. fonct. éléments de F satisfait aussi f .

- Exemples :

- $F_1 = \{A \rightarrow B, B \rightarrow CD\}$ implique $A \rightarrow CD$ et implique aussi $A \rightarrow ABCD$.

- $F_2 = \{A \rightarrow B, A \rightarrow C\}$ implique $A \rightarrow BC$.

- $F_3 = \{A \rightarrow BC\}$ implique $A \rightarrow B$ et $A \rightarrow C$.

- $F_4 = \{AB \rightarrow C, AD \rightarrow E\}$ **n'implique pas** $AB \rightarrow CE$.

Voir au tableau comment construire un **contre-exemple** à cette implication.

5.2.2 Plus sur les dépendances fonctionnelles

- Dans l'exemple on a vérifié des implications en utilisant la définition d'implication déjà donnée.
- Existe-t-il une autre façon de faire ?
- Le système axiomatique dit d'Armstrong (voir TD) est correct et complet par rapport à la notion d'implication définie :

Thm. F implique f ssi il existe une déduction de f à partir de F dans le système d'Armstrong.

- Mais, **Problème** : ce système ne fournit aucun algorithme pour tester si F implique f !

Comment calculer de façon mécanique si f est impliquée par un ensemble de dép. fonct. F ?

- **Déf.** : Soit Z l'ensemble des attributs dans F , $X \subseteq Z$.

La *fermeture* de X par rapport à F (X^+_F) est :

$$\max\{Y \subseteq Z \mid F \text{ implique } X \rightarrow Y\}$$

- **Théorème** : F implique $X \rightarrow Y$ ssi $Y \subseteq X^+_F$.
- **Donc**, pour savoir si F implique $X \rightarrow Y$ il suffit de savoir calculer X^+_F !.

ALGO DE CALCUL DE X^+_F :

Entrée : F, X, Att , où Att est l'ensemble de tous les attributs dans F

Sortie : X^+_F comme valeur de la variable res .

$res := X;$

répéter :

pour chaque $Y \rightarrow Z \in F$, **si** $Y \subseteq res$ **alors** $res := res \cup Z$

jusqu'à quand res n'a pas changé ou $res = Att$;

renvoyer res

Question 1 : Complexité dans le pire des cas ?

Exemple de calcul de AB^+_F pour $F = \{A \rightarrow C, BC \rightarrow D, ED \rightarrow E\} \rightsquigarrow$
tableau.

Question 2 : S =schéma d'une relation, $X \subseteq S$. Comment utiliser cet algo pour décider si X est une clé de toute relation sur S (fixée par F) ?

Exemples

1. On fait tourner l'algorithme pour

$$U_1 = \{A, B, C, D\}, \quad F_1 = \{A \rightarrow B, BC \rightarrow D\}$$

pour tester si F_1 implique $A \rightarrow D$ et si F_1 implique $AC \rightarrow D$.

2. On fait tourner l'algorithme pour

$$U_2 = \{A, B, C, D, E\}, \quad F_2 = \{AB \rightarrow C, B \rightarrow D, D \rightarrow B\}$$

pour calculer les clés d'une table $R(A, B, C, D, E)$ fixées par F .

5.3 Problèmes possibles pour un schéma d'une base

Relation FOURNISSEUR

| <i>Nom</i> | <i>Adr</i> | <i>Produit</i> | <i>Prix</i> |
|------------|------------|----------------|-------------|
| n1 | a1 | i1 | p1 |
| n1 | a1 | i2 | p2 |
| n2 | a2 | i2 | p3 |
| n2 | a2 | i3 | p4 |

$$F = \text{Nom} \rightarrow \text{Adr}, \text{Nom Produit} \rightarrow \text{Prix}$$

- **Redondances**, car l'adresse est répétée
- **Anomalies de mise à jour** : si on met à jour une adresse... attention, partout !
- **Anomalies d'insertion** : si pas de valeur nuls, impossibilité d'enregistrer une adresse d'un fournisseur qui n'a pas (encore) de produit.
- **Anomalies de suppression** : si effacement de tous les articles de n1, perte de l'adresse de n1.

Anomalies de ce schéma : liés au comportement du schéma par rapport aux DF (ici : dépendance “partielle”, “2nde forme normale” violée).

(Voir la suite pour les défs. formelles de la notion de “formes normales”).

C’est pour éviter ce type d’anomalies que l’on essaie de concevoir, dès le départ, des schémas de base qui soient satisfaisants.

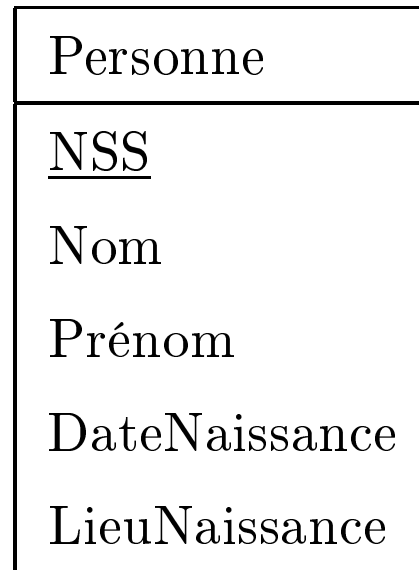
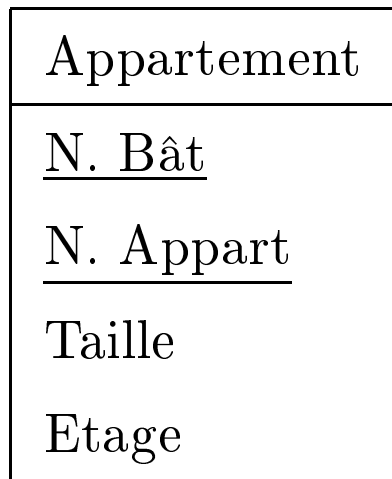


5.4 Méthode Entité-Association (dite aussi “Modèle EA”)

(Merci à M. M. Rahoual pour ses sources pédagogiques sur le sujet)

- Entité = un nom qui désigne un ensemble d’objets (instances).
- Par ex. : Personne, Film, Aéroport, Appartement, sont des entités. “Orly” et “Roissy - Charles de Gaulle” sont 2 instances de Aéroport.
- Attributs d’une entité : propriétés des objets de l’entité. Par ex. attributs de Appartement : N.Bât, N.Appart, Taille, Etage.
- Ensemble d’attributs permettant d’identifier les instances d’une entité : *clé*. Par ex., clé de Appartement : N.Bât, N.Appart.

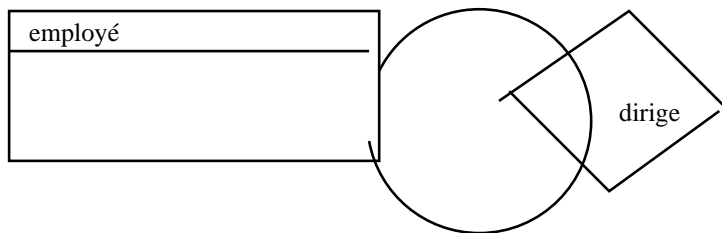
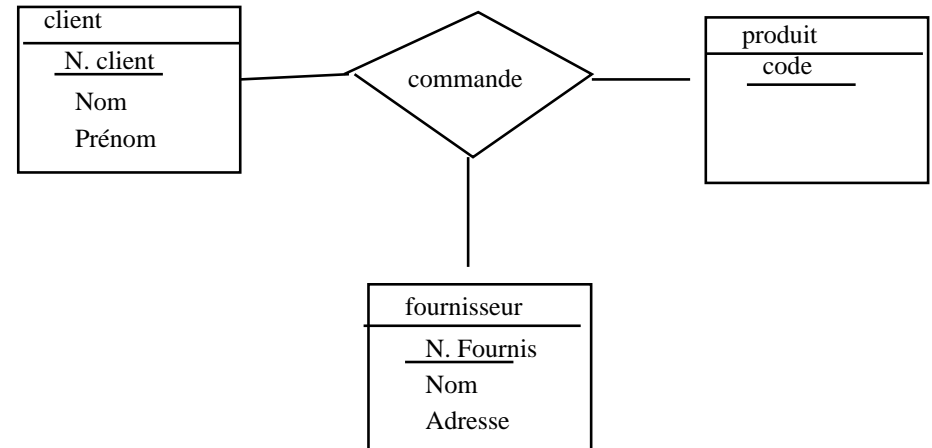
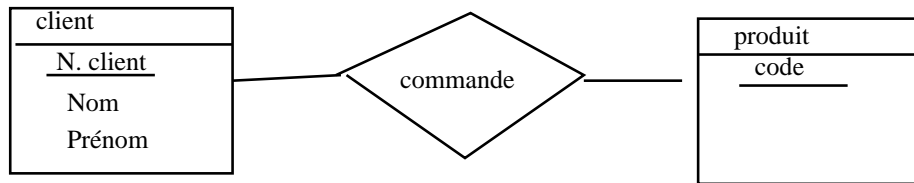
Représentation graphique d'une entité : un rectangle, clé soulignée.



- Association : Un lien A qui relie au moins deux entités (éventuellement E avec E).
- Idée : Si $E1$ et $E2$ sont reliées par A , A indique un ensemble de couples $\langle e_1, e_2 \rangle$ tels que e_1 est une instance de $E1$ et e_2 est une instance de $E2$.
- Par ex., l'association (binaire) Commande entre Client et Produit regroupe des couples $\langle c, p \rangle$ tq c est un client et p est un produit.
- Représentation graphique : les entités sont les sommets d'un graphe dont les arcs sont les associations. Un arc est étiqueté par un losange avec le nom de l'association.

Exemples de représentations graphiques des associations : 1 :

commande, binaire, entre client et produit, **2** : commande, ternaire, entre client, produit et fournisseur, **3** : dirige, binaire mais réflexive, entre employé et employé.



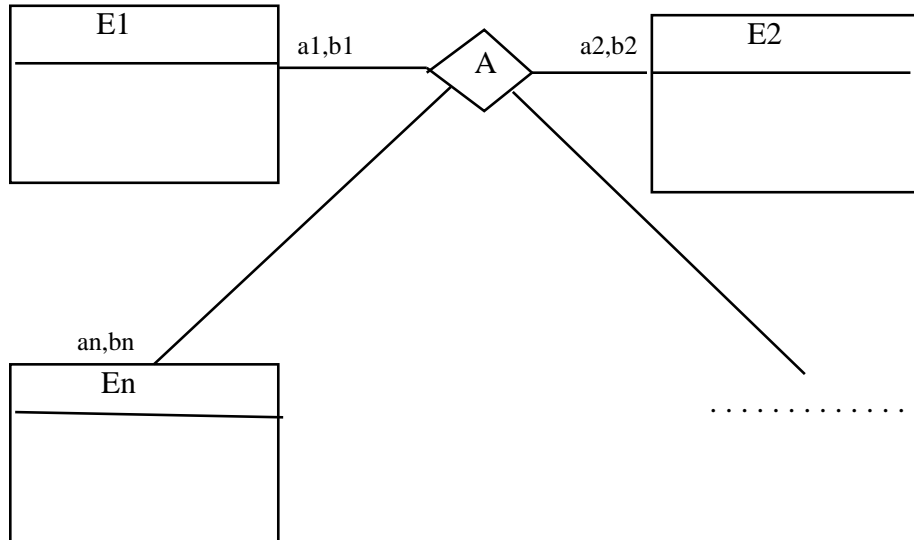
- Une **association A** a comme **attributs les clés des entités reliées**. Ces attributs forment la clé de A.
- A peut avoir aussi des **attributs spécifiques**, mais, attention : ils dépendent de toutes les entités reliées.

Par ex. : association Catalogue entre Fournisseur, dont la clé est NFournisseur, et Produit, dont la clé est code.

Attributs de Catalogue : NFournisseur, code + Prix.

Prix est spécifique. *Il dépend à la fois du fournisseur et du produit !*

Cardinalités



Pour $1 \leq i \leq n$:

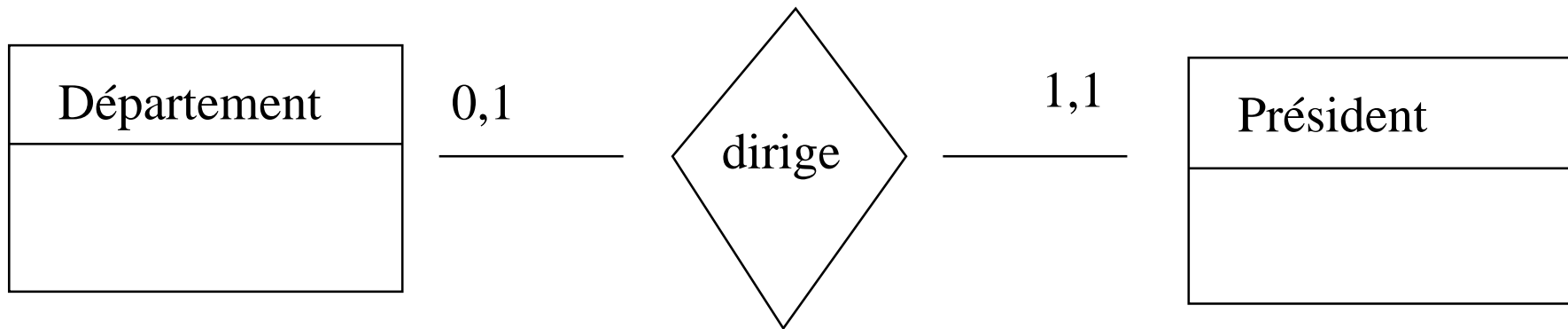
a_i = nombre minimal de $n - 1$ -uplets de $E_1 \times E_2 \times \cdots \times E_{i-1} \times E_{i+1} \times E_n$ auxquels une instance qqe de E_i peut être reliée par A.

Valeurs possibles : 1 et 0.

b_i = nombre maximal de $n - 1$ -uplets de $E_1 \times E_2 \times \cdots \times E_{i-1} \times E_{i+1} \times E_n$ auxquels une instance qqe de E_i peut être reliée par A.

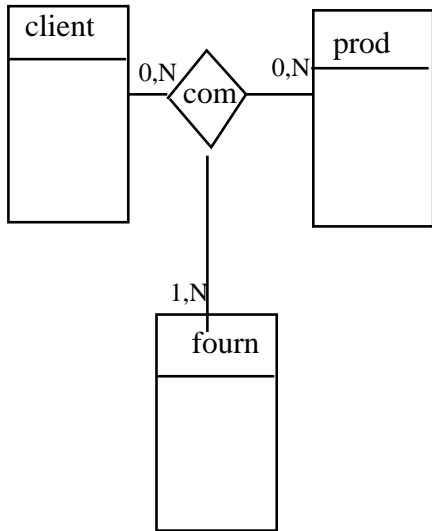
Valeurs possibles : 1 et N (= tout $m > 1$).

Exemple 1



Un département peut ne pas avoir de président et il en a au max. 1. Un président dirige exactement un département (pas +, pas -).

Exemple 2



Un client peut ne passer aucune commande de produit à un fournisseur, mais peut aussi en passer autant qu'il veut.

Un produit peut être associé à un nombre quelconque de couples (client, fournisseur), éventuellement à aucun.

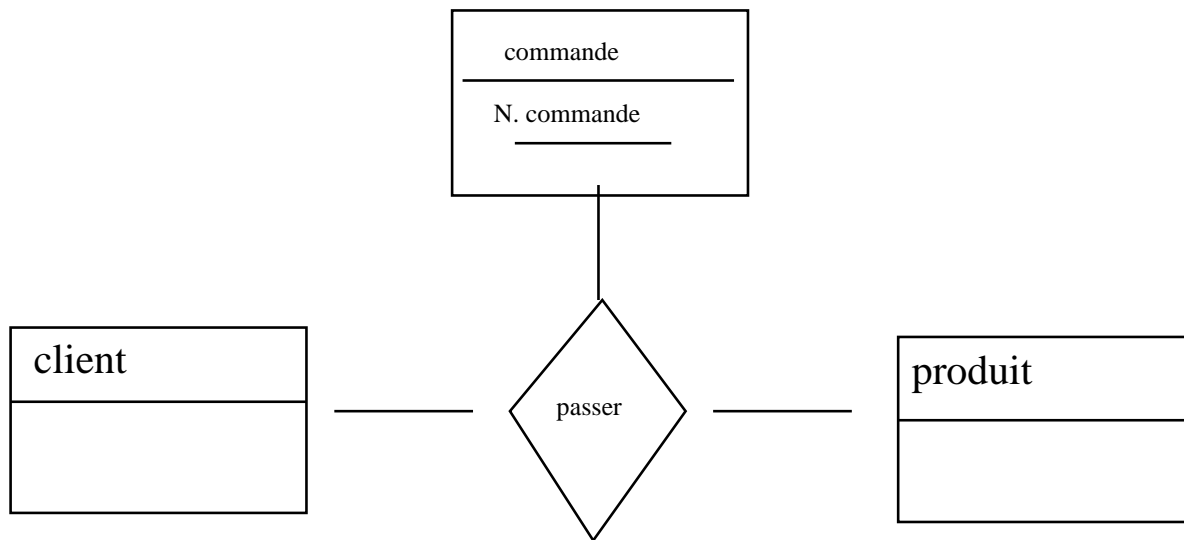
Un fournisseur doit fournir au moins un produit à au moins un client, éventuellement il peut recevoir plusieurs commandes.

Règles à suivre pour avoir un diagramme EA OK

Règle 1 : Existence d'un identifiant pour chaque entité. (PB. : Modéliser une "situation" par une entité ou par une association ?)

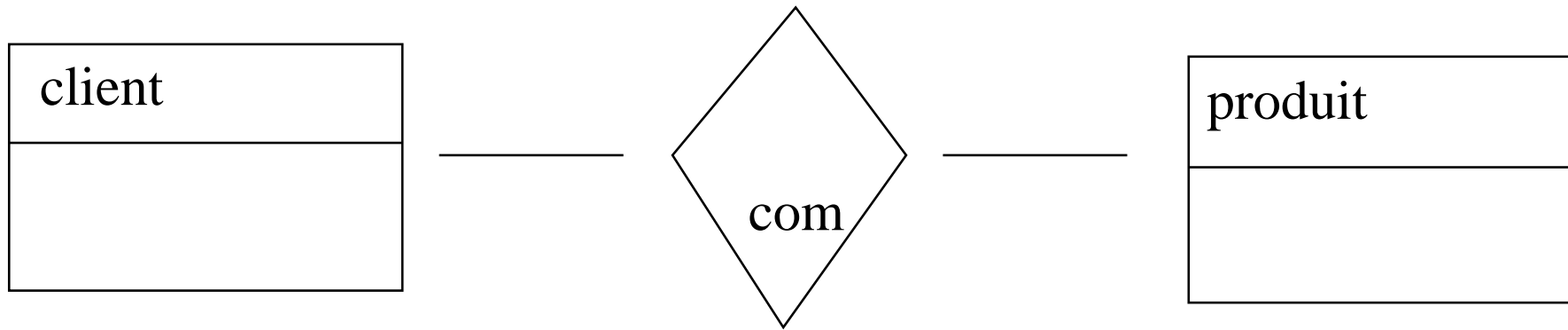
Exemple. Situation : des clients commandent des produits.

On suppose que le concept de commande inclue un num. de commande unique pour chaque commande. Modéliser ce concept par une entité :



On ne fait **pas** cette hypothèse : page suivante.

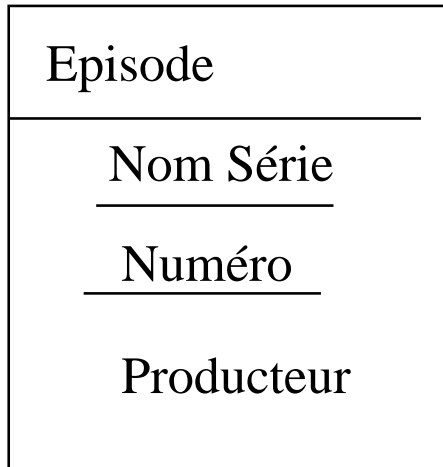
On peut modéliser par une association :



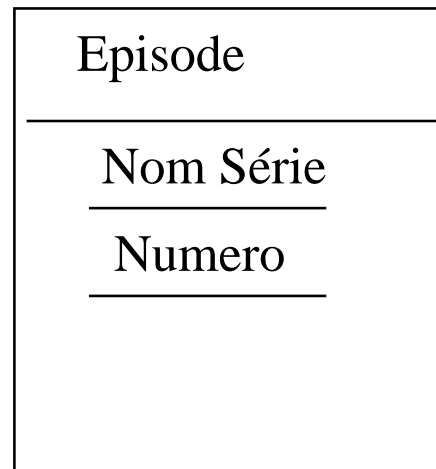
Clé de l'association : clé de Client, clé de Produit.

Règle 2A : Pour chaque entité E , tout attribut de E qui n'est pas dans la clé C de E doit être déterminé par C , mais un sous-ensemble strict C' de C ne doit pas suffire à le déterminer.

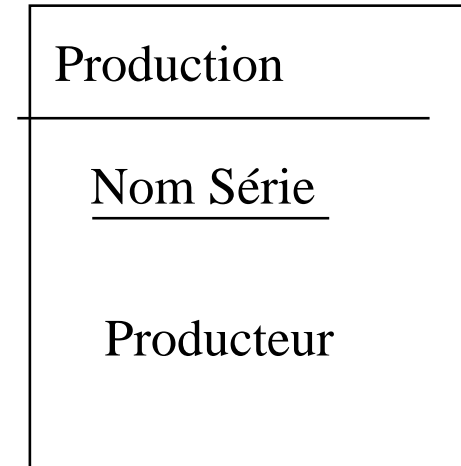
Exemple. Hyp : toute série (TV) a un seul producteur. Violation de la règle 2, puis solution alternative \rightsquigarrow page suivante.



PAS OK



OK



Puisque on avait :

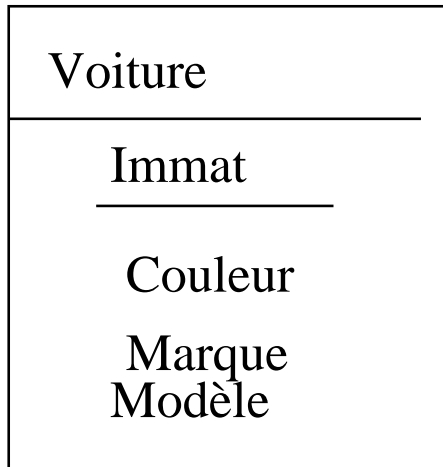
NomSérie *détermine* Producteur (violation de la règle 2) on a enlevé Producteur de Episode et on a crée la nouvelle entité Production.

Motivation intuitive de la règle 2A ? On veut éviter, dans le schéma relationnel résultat, des “dépendances partielles”.

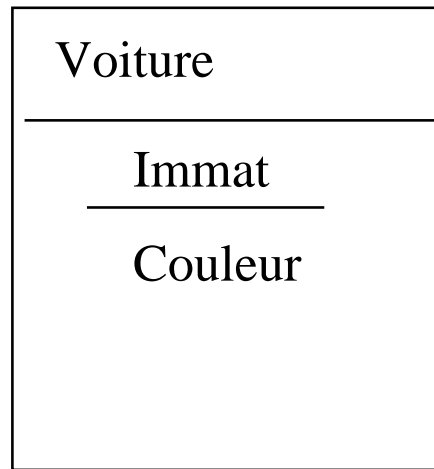
On veut forcer le respect de la 2^{nde} Forme Normale.

Règle 2B : Pour chaque entité ayant clé C , si un attribut A est déterminé par un ensemble d'attributs E , on doit avoir $E \subseteq C$. (Dépendance "directe" de A à partir de C).

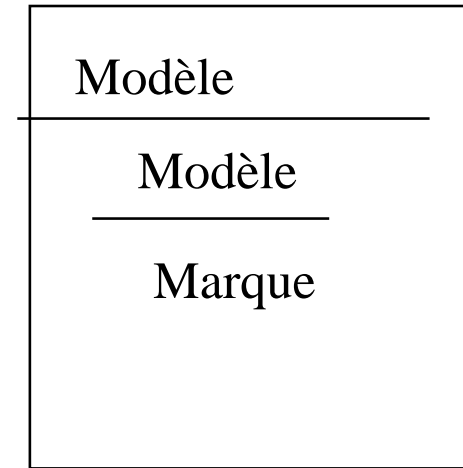
Exemple. Hyp : tout modèle de voiture détermine la marque de la voiture, mais l'entité Voiture a comme clé Immatriculation. Violation de la règle 2B, puis solution alternative \rightsquigarrow page suivante.



PAS OK



OK



Puisque on avait :

a) Immatriculation, qui est la clé de voiture, *détermine* Modèle;

b) Modèle *détermine* Marque (mais Modèle ne fait pas partie de la clé)

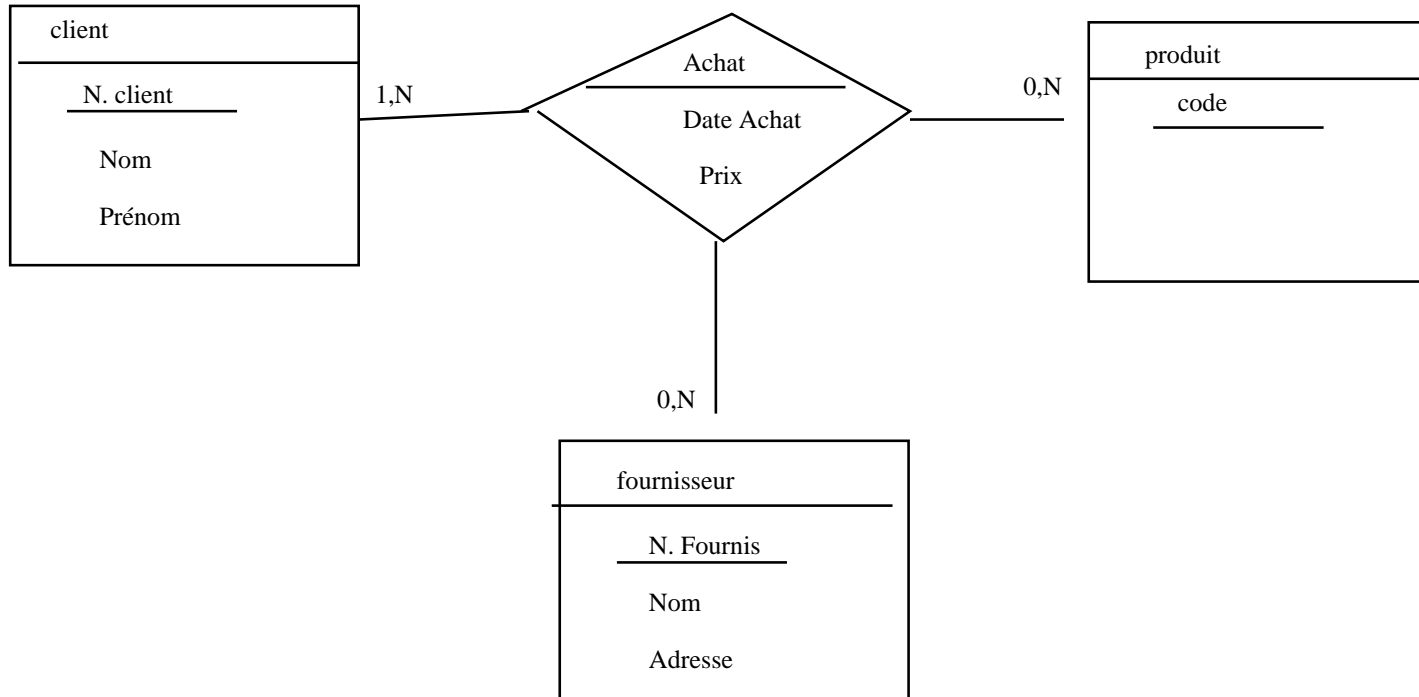
on a enlevé Marque et Modèle de Voiture et on a crée la nouvelle entité Modèle, dont la clé est Modèle et l'autre attribut est Marque.

Motivation intuitive de la règle 2B ? On veut éviter, dans le schéma relationnel résultat, des “dépendances transitives”.

On veut forcer le respect de la 3ème Forme Normale.

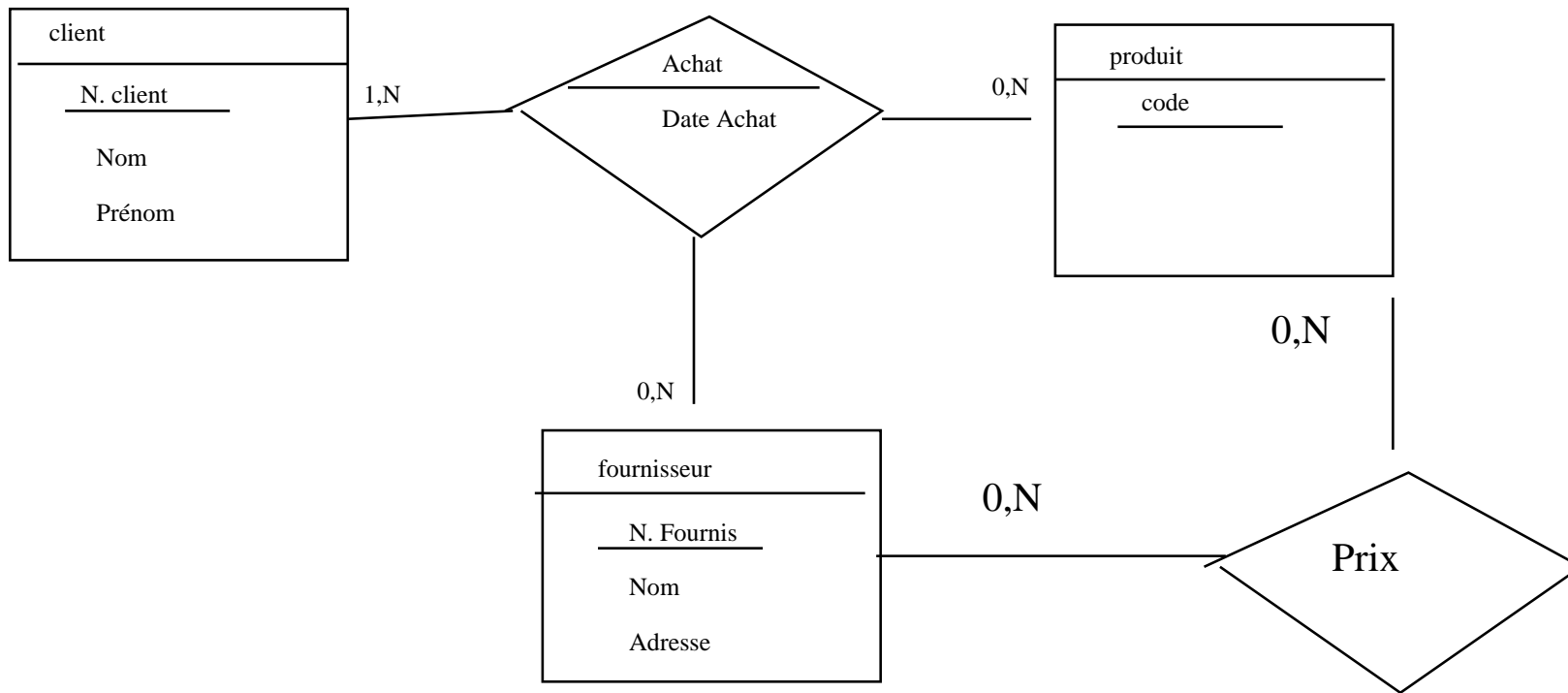
Règle 3 : Tous les attributs d'une association A doivent dépendre complètement de la clé de cette association (= ensemble des clés des entités reliées).

Cas de violation de la Règle 3 :



Le prix dépend seulement du fournisseur et du produit, pas du client. **Solution ?**

↪ page suivante.



On a effacé l'attribut Prix de l'association Achat. Il est devenu une association dont les seuls attributs sont sa clé : N. Fournis (clé de Fournisseur), code (clé de Produit).

Motivation intuitive de la règle 3 ?

A nouveau, on veut éviter, dans le schéma relationnel résultat, des “dépendances partielles”.

On veut forcer le respect de la 2^{de} Forme Normale.

Rôle des règles de conception des données : contraindre le schéma relationnel qui résultera du diagramme EA à avoir une *forme normale* qui permettra de ne pas avoir certains problèmes et “anomalies” \rightsquigarrow voir la suite.

Transformer un diagramme EA en un schéma relationnel

Les entités et les associations vont devenir des (schémas de) tables.

Principes concernant les entités

1. Entité $E \rightsquigarrow$ Schéma S_E d'une table E .
2. Attribut de l'entité $E \rightsquigarrow$ Attribut de S_E .
3. Clé de l'entité $E \rightsquigarrow$ Clé de la table E . Si $At = \{A_1, \dots, A_n\}$ est l'ensemble des attributs de l'entité E et $C \subseteq At$ est la clé de cette entité, alors on impose la dép. fonct. $C \rightarrow A_1 \dots A_n$.

| |
|------------------|
| Client |
| <u>N. Client</u> |
| Nom |
| Prénom |

devient le schéma de table :

| | | |
|-----------|---------------|--------|
| | Client | |
| N. Client | Nom | Prénom |

avec la dépendance fonct. : N. Client \rightarrow Prénom Nom

Principes concernant les associations

A1) Association binaire A ayant N en cardinalité max des 2 cotés \rightsquigarrow Schéma S_A d'une table A . Attributs de S_A : ceux de A , clé C pour la table A : clé C de l'association. On dit que chaque élément de C est une **clé étrangère de la table créée.**

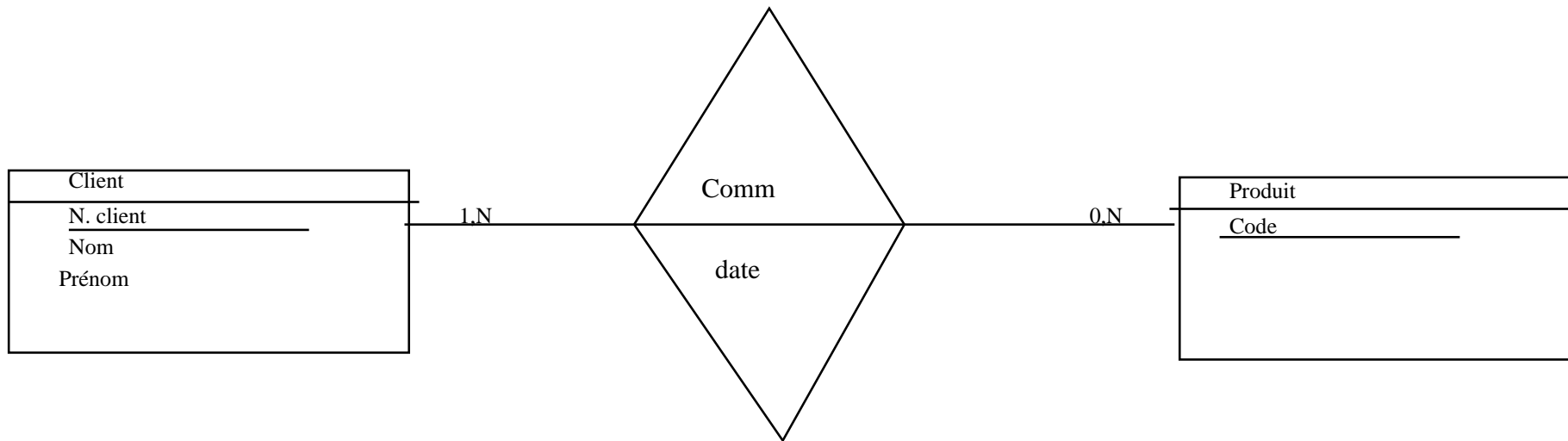


Schéma relationnel correspondant :

Client(N.client, Nom, Prénom), Produit(Code), Comm(N.client, Code, date).

(Deux clés étrangères de Comm : 1) N.client et 2) Code)

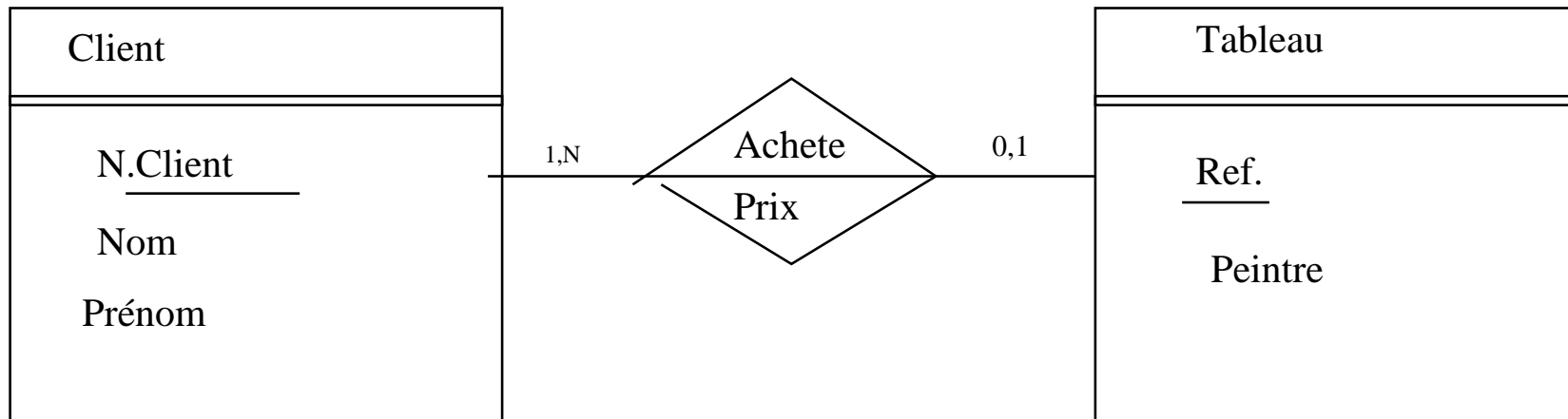
DF : N.client \rightarrow Nom Prénom et N.client Code \rightarrow date

A2) Association binaire A ayant 0,1 en cardinalité sur au moins un des 2 cotés :

Si SGBD tolère NULL \rightsquigarrow pas de table A . Les attributs de l'association \rightsquigarrow des attributs de la table associée à l'entité ayant 1 en cardinalité max.

Sinon \rightsquigarrow Schéma S_A d'une table A . Attributs de S_A : ceux de A ; clé pour la table $A =$ la clé de l'association.

Exemple



Exemple, suite

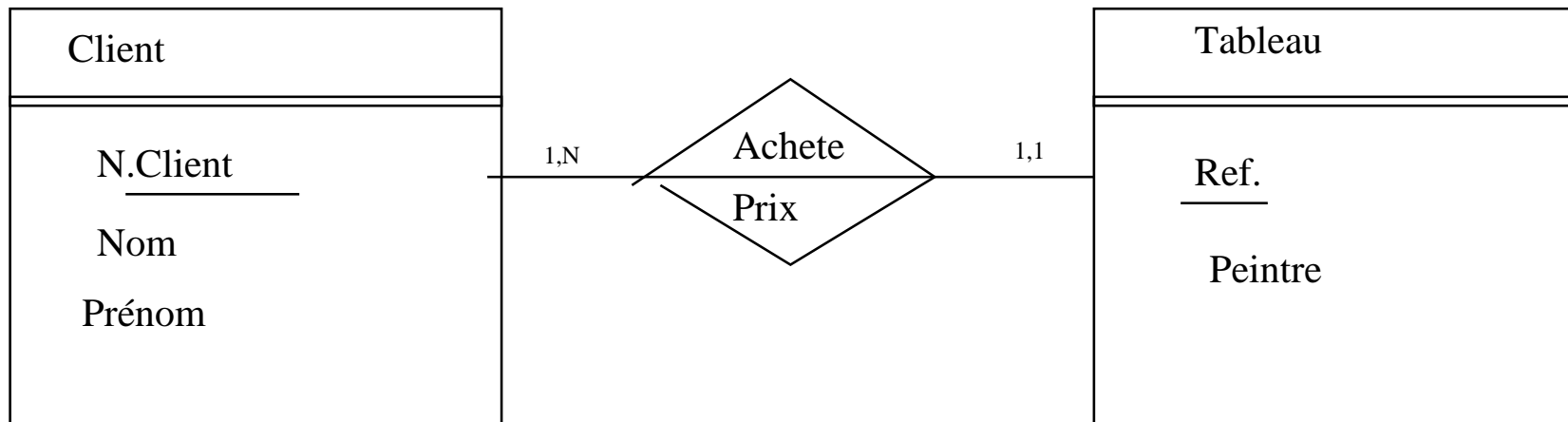
- NULL accepté : Client(N.Client, Nom, Prénom), Tableau(Réf, Peintre, N.Client, Prix). **NB** : N.Client est attribut de la table Tablealeau !.

Que se passe-t-il si le tableau X de Th. Van Gogh n'est jamais vendu, ici ?

- NULL interdit : Client(N.Client, Nom, Prénom), Achète(N.Client, Réf, Prix), Tableau(Réf, Peintre).

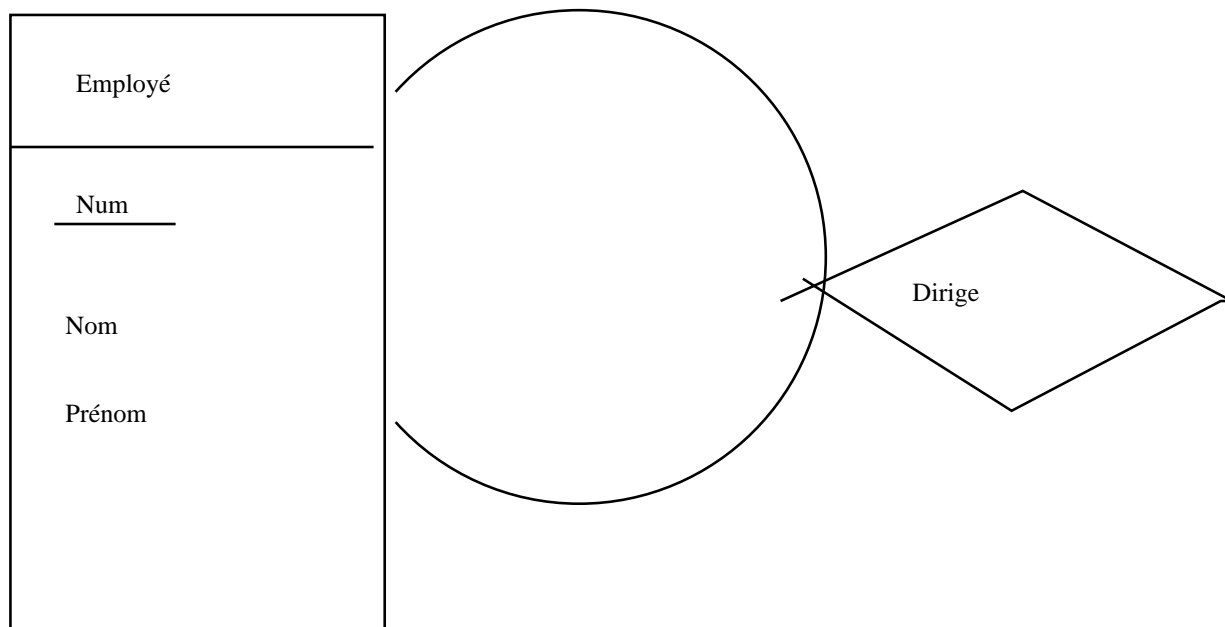
Que se passe-t-il si le tableau X de Th. Van Gogh n'est jamais vendu, ici ?

A3) Association binaire A, entre E1 et E2, ayant 1,1 en cardinalité sur au moins un des 2 cotés : \rightsquigarrow pas de table A. Soit E2 l'entité "du côté 1,1".
 Dans la table T2, on insère les attributs de l'association, dont, en particulier, la clé de E1, qui devient clé étrangère de T2.



Client(N.Client, Nom, Prénom), Tableau(Réf, Peintre, N.Client, Prix).
 Clé étrangère de Tableau : N.Client.

A4) Association binaire A REFLEXIVE : \rightsquigarrow toujours un nouveau schéma de table (peu important les cardinalités), comme suggéré par l'exemple.

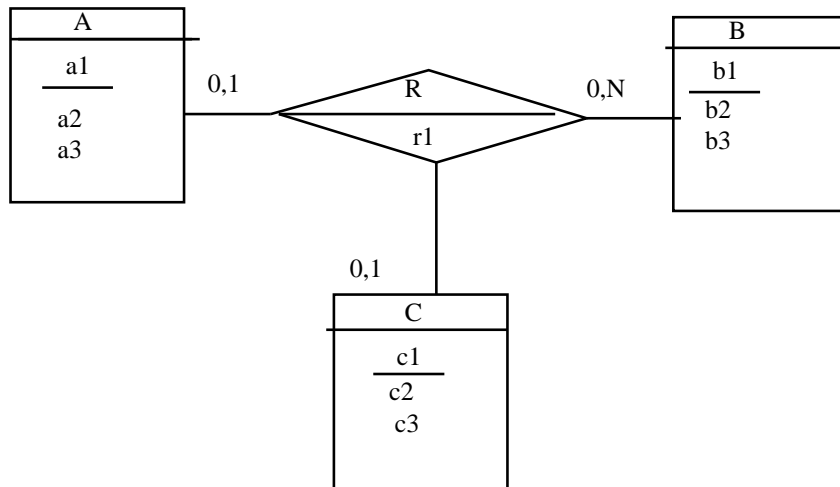


Employé(Num, Nom, Prénom), Dirige(Num_Sup, Num_Subalt)

A5) Association n -aire A avec $n > 2 \rightsquigarrow$ toujours un nouveau schéma de table $S(A)$.

Si toutes les cardinalités max sont N , la clé de la table A est l'union des clés associées aux entités.

Sinon, prendre comme clé de la nouvelle table la clé d'une entité ayant 1 en cardinalité max.



$A(\underline{a1}, a2, a3)$, $B(\underline{b1}, b2, b3)$, $C(\underline{c1}, c2, c3)$ et soit $R(\underline{a1}, b1, c1, r1)$ soit $R(a1, b1, \underline{c1}, r1)$.

NB : les deux possibilités sont OK, car on aura $a_1 \rightarrow b_1 c_1 r_1$ et $c_1 \rightarrow a_1 b_1 r_1$, tandis que choisir b_1 comme clé de R n'aurait pas marché.