

# 3. MODELE RELATIONNEL ET NORMALISATION

Version 3 - septembre 2018

Support du chapitre 3, *Modèle relationnel et normalisation*  
de l'ouvrage *Bases de données*, J-L Hainaut, Dunod 2018.

## 3. MODELE RELATIONNEL ET NORMALISATION

### Contenu

- 3.1 Introduction**
- 3.2 Le modèle relationnel**
- 3.3 Identifiants**
- 3.4 Dépendances fonctionnelles**
- 3.5 Contraintes d'inclusion et clés étrangères**
- 3.6 Calcul des identifiants d'une relation**
- 3.7 Décomposition d'une relation**
- 3.8 Normalisation d'une relation**

## 3.1 Introduction

## 3.1 Introduction

### Le modèle *relationnel* fait son entrée dans le monde en 1970 :

Codd, E., F., A Relational Model of Data for Large Shared Data Banks, in *Comm. ACM*, Vol. 13, No 6, June 1970.

### = modèle théorique sous la forme d'une algèbre

- modèle abstrait : pas de détails techniques (~~index, support de stockage, volumes, temps d'accès~~)
- structure de données simple (voire simpliste) : tabulaire
- compréhensible par l'utilisateur final
- nombre minimal de concepts
- interprétation sans ambiguïté (interprétation mathématique)
- pourrait même faire l'objet d'une implémentation !
- grand succès scientifique mais indifférence des milieux industriels !

## 3.1 Introduction

**Un exploit rarissime en informatique : *la théorie précède l'exploitation industrielle !***

**Mais l'industrie s'est cruellement vengée : SQL2 puis SQL3 !**

**Quelle a été l'utilité de ce modèle théorique ?**

- simplifier les modèles de bases de données
- diffuser et populariser le concept de base de données
- comprendre les problèmes fondamentaux des modèles de données
- élaborer des stratégies de conception de bases de données
- construire des langages universels de bases de données
- conduire à des implémentations de plus en plus efficaces : SYSTEM/R (1974), INGRES (1974), Oracle (1979), SQL/DS (1982), etc.

## 3.2 Le modèle relationnel

## 3.2 Le modèle relationnel - Les concepts

Deux concepts de base :

- le domaine de valeurs = ensemble prédéfini de valeurs simples
- la relation = partie du produit cartésien de domaines

*et c'est tout ?*

*Ben oui, c'est (à peu près) tout !*

domaines : CHAINE, PRODUIT, PRIX

relation : OFFRE(CHAINÉ, PRODUIT, PRIX)  $\subseteq$  CHAINE X PRODUIT X PRIX

domaines : string, number

relation : OFFRE(CHAINÉ: string, PRODUIT: string, PRIX: number)

**schéma**

**attributs**

**contenu**

## 3.2 Le modèle relationnel - Les concepts

**Remarque :** une relation est un *ensemble* de lignes (ou n-uplets)

**Donc :**

- l'ordre des lignes est indifférent
- les lignes sont distinctes

**En outre :**

- les attributs sont obligatoires (pas de valeur **null**)

**Donc, une *table* n'est pas une *relation* mais une plus ou moins bonne approximation technique d'une relation.**



## 3.2 Le modèle relationnel - Les opérateurs algébriques

Comment *interroger* ces données (domaines et relations) ?

En construisant d'autres domaines et relations au moyen d'**opérateurs**.

- **Opérateurs ensemblistes** : *union, intersection, différence*

- **Opérateurs relationnels** : *projection, jointure, sélection*

On peut montrer que ces opérateurs sont suffisants pour exprimer toutes les données calculables à partir d'une base de données relationnelle (sauf quelques *détails* : arithmétique, agrégation, récursivité).

Equivalent à la logique des prédicats du 1er ordre.

## 3.2 Le modèle relationnel - Les opérateurs algébriques

OFFRE		
CHAINE	PRODUIT	PRIX
SUPER-U	Chocolat	6
MAMMOUTH	Sucre	2,2
MAMMOUTH	Chocolat	5
MATCH	Sel	1,4
MAMMOUTH	Sel	1,4

IMPLANTATION	
CHAINE	VILLE
SUPER-U	LILLE
SUPER-U	ANNECY
MAMMOUTH	LYON
MAMMOUTH	LILLE
MATCH	PARIS
MATCH	LILLE
MATCH	TOURNAI
CORA	CHARLEROI

*support des exemples*

## 3.2 Le modèle relationnel - Projection d'une relation

### Expression algébrique de la projection

**NIVEAU-des-PRIX** = OFFRE [PRODUIT, PRIX]

OFFRE		
CHAINE	PRODUIT	PRIX
SUPER-U	Chocolat	6
MAMMOUTH	Sucre	2,2
MAMMOUTH	Chocolat	5
MATCH	Sel	1,4
MAMMOUTH	Sel	1,4



NIVEAU-des-PRIX	
PRODUIT	PRIX
Chocolat	6
Sucre	2,2
Chocolat	5
Sel	1,4

**Attention : le résultat est un ensemble !**

### Expression SQL

```
select distinct PRODUIT, PRIX  
from OFFRE
```

## 3.2 Le modèle relationnel - Jointure de relations

### Expression algébrique de la jointure

**OFFRE-VILLE = OFFRE (CHAINE) \* IMPLANTATION (CHAINE)**

**OFFRE-VILLE = OFFRE \* IMPLANTATION**

OFFRE		
CHAINE	PRODUIT	PRIX
SUPER-U	Chocolat	6
MAMMOUTH	Sucre	2,2
MAMMOUTH	Chocolat	5
MATCH	Sel	1,4
MAMMOUTH	Sel	1,4

\*

IMPLANTATION	
CHAINE	VILLE
SUPER-U	LILLE
SUPER-U	ANNECY
MAMMOUTH	LYON
MAMMOUTH	LILLE
MATCH	PARIS
MATCH	LILLE
MATCH	TOURNAI
CORA	CHARLEROI



OFFRE-VILLE			
CHAINE	PRODUIT	PRIX	VILLE
SUPER-U	Chocolat	6	ANNECY
SUPER-U	Chocolat	6	LILLE
MAMMOUTH	Sucre	2,2	LILLE
MAMMOUTH	Sucre	2,2	LYON
MAMMOUTH	Chocolat	5	LILLE
MAMMOUTH	Chocolat	5	LYON
MATCH	Sel	1,4	TOURNAI
MATCH	Sel	1,4	LILLE
MATCH	Sel	1,4	PARIS
MAMMOUTH	Sel	1,4	LILLE
MAMMOUTH	Sel	1,4	LYON

## 3.2 Le modèle relationnel - Jointure de relations

### Expression algébrique de la jointure

`OFFRE-VILLE = OFFRE (CHAINE) * IMPLANTATION (CHAINE)`

`OFFRE-VILLE = OFFRE * IMPLANTATION`

### Expression SQL

```
select O.CHAINE, PRODUIT, PRIX, VILLE
from   OFFRE O, IMPLANTATION I
where  O.CHAINE = I.CHAINE
```

## 3.2 Le modèle relationnel - Sélection de lignes

### Expression algébrique de la sélection

`OFFRE-BUDGET = OFFRE (PRIX < 5)`

### Expression SQL

```
select *  
from OFFRE  
where PRIX < 5
```

## 3.2 Le modèle relationnel - Combinaison d'opérateurs

### Expression algébrique complexe

`PRODUIT-VILLE = (OFFRE (PRIX < 5) * IMPLANTATION) [PRODUIT, VILLE]`

### Expression SQL

```
select PRODUIT, VILLE
from OFFRE O, IMPLANTATION I
where O.CHAINE = I.CHAINE
and PRIX < 5
```

## 3.2 Le modèle relationnel - Autres opérateurs

**L'algèbre relationnelle comporte traditionnellement quelques opérateurs supplémentaires :**

- **complémentaire (les lignes possibles qui ne sont pas dans une relation)**
- **le produit relationnel (jointure sans condition)**
- **division relationnelle (exprime le quantificateur *pour tout*)**
- **renommage (pour lever certaines ambiguïtés dans les expressions)**
- **etc.**

**Intérêt mineur dans le cadre de ce cours**



## 3.3 Identifiants

Dans le langage courant, on parle plutôt de **clé (ou key)**

**Problème :** *unique key, secondary key, superkey, foreign key, access key, record key, alternate record key, relative key, sequence key, encryption key, user key, privacy key, master key, public key, private key, concatenated key, hierarchical key, prefix key, sort key, etc.*

## 3.3 Identifiants

**Identifiant** : *sous-ensemble des attributs tel qu'il ne peut exister à aucun moment plus d'une ligne possédant les mêmes valeurs de ces attributs*

### Quelques propriétés :

- identifiant *minimal*
- tout ensemble d'attributs dont une partie stricte est un identifiant est aussi un identifiant (mais *non minimal*)
- l'ensemble des attributs d'une relation est un identifiant (car . . .)
- plusieurs identifiants minimaux peuvent coexister dans une relation
- un attribut peut appartenir à plusieurs identifiants
- il est possible de calculer automatiquement les identifiants d'une relation

## 3.3 Identifiants - Exemples

VENTE(ARTICLE, MAGASIN, PRIX, ...)

VENTE'(ARTICLE, MAGASIN, PRIX, ...)

VENTE"(ARTICLE, MAGASIN, PRIX, ...)

EMPLOYE(MATR, NSS, NOM, ADRESSE)

HORAIRE(PROF, HEURE, LOCAL)

id: PROF,HEURE

id: HEURE,LOCAL

## 3.4 Dépendances fonctionnelles

## 3.4 Dépendances fonctionnelles

### Dépendance fonctionnelle :

- contrainte d'intégrité très importante du modèle relationnel
- proche de l'identifiant mais plus précis
- à la base de la théorie de la normalisation

## 3.4 Dépendances fonctionnelles

ACHAT		
CLIENT	PRODUIT	PRIX
André	Sucre	2,2
Marc	Sucre	2,2
Marc	Sel	1,4
Anne	Savon	1,4
Anne	Sel	1,4

Quelle que soit la ligne,  
*à une même valeur de PRODUIT est toujours associée la même valeur de PRIX*

*On note* : ACHAT: PRODUIT  $\longrightarrow$  PRIX  
PRODUIT  $\longrightarrow$  PRIX

PRODUIT est un identifiant de ACHAT[PRODUIT, PRIX]

## 3.4 Dépendances fonctionnelles

### Définition standard :

Dans une relation  $R(A,B,C,D)$ , il existe une *dépendance fonctionnelle*  $A \longrightarrow B$  si, à tout instant, deux lignes de  $R$  qui ont même valeur de  $A$  ont aussi même valeur de  $B$ .

- PRODUIT *détermine* (fonctionnellement) PRIX ;
- PRIX *dépend* (fonctionnellement) de PRODUIT ;
- PRODUIT est le *déterminant* et PRIX est le *déterminé* de la dépendance fonctionnelle.

Déterminant et déterminé peuvent être multicomposants

$B \longrightarrow C,D$

$B,C \longrightarrow D$

## 3.4 Dépendances fonctionnelles - Exemple (à ne pas imiter !)

COM(NCLI, NOM, ADRESSE, NCOM, DATE, NPRO, QTE, PRIX-U)

NCOM  $\longrightarrow$  NCLI

*toute commande est émise par un client*

NCLI  $\longrightarrow$  NOM

*tout client a un nom*

NCLI  $\longrightarrow$  ADRESSE

*tout client a une adresse*

NCOM  $\longrightarrow$  DATE

*toute commande est passée à une date*

NCOM, NPRO  $\longrightarrow$  QTE

*dans toute commande, il y a une quantité par produit*

NPRO  $\longrightarrow$  PRIX-U

*tout produit a un (et un seul) prix unitaire*



## 3.4 Dépendances fonctionnelles - Graphe ADF (attributs + DF)

COM(NCLI, NOM, ADRESSE, NCOM, DATE, NPRO, QTE, PRIX-U)

NCOM  $\longrightarrow$  NCLI

*toute commande est émise par un client*

NCLI  $\longrightarrow$  NOM

*tout client a un nom*

NCLI  $\longrightarrow$  ADRESSE

*tout client a une adresse*

NCOM  $\longrightarrow$  DATE

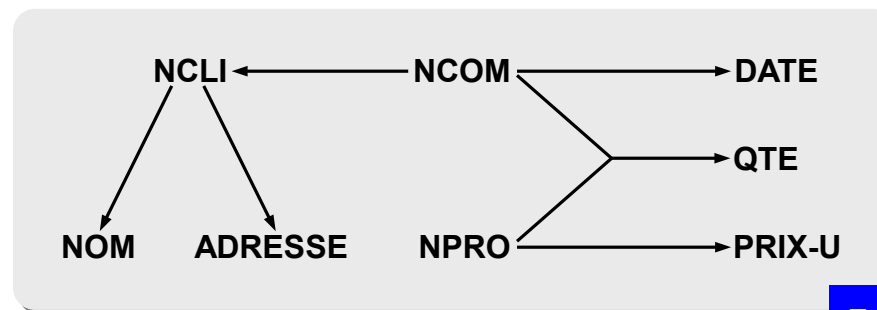
*toute commande est passée à une date*

NCOM, NPRO  $\longrightarrow$  QTE

*dans toute commande, il y a une quantité par produit*

NPRO  $\longrightarrow$  PRIX-U

*tout produit a un (et un seul) prix unitaire*



**= graphe ADF de la relation**

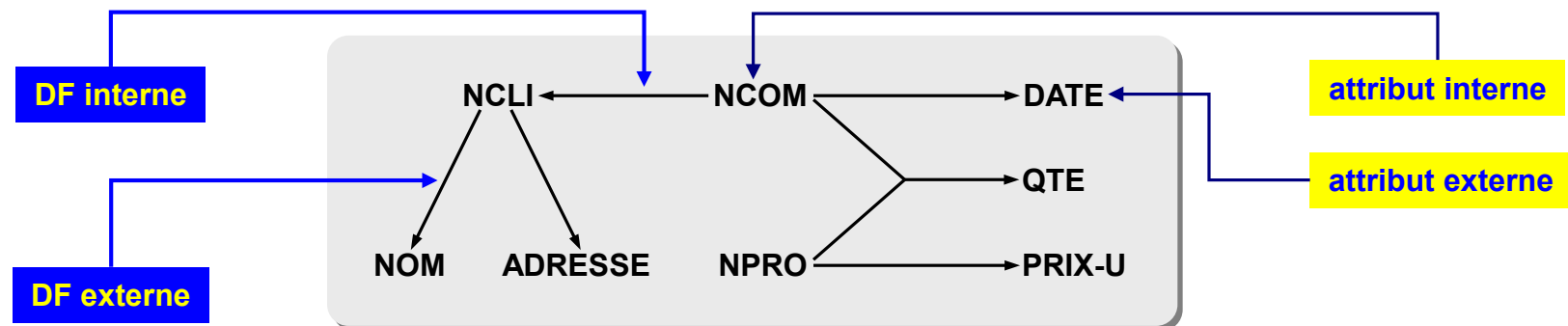
## 3.4 Dépendances fonctionnelles - Graphe ADF

**attribut externe** : non déterminant

**attribut interne** : déterminant

**DF externe** : dont le déterminé est un attribut externe

**DF interne** : dont le déterminé est un attribut interne



## 3.4 Dépendances fonctionnelles - Propriétés

### *Les règles d'inférence d'Armstrong*

1. *Réflexivité* :

Si  $L$  est un sous-ensemble de  $K$ , alors on a  $K \longrightarrow L$ .

Un ensemble d'attributs détermine chacun de ses sous-ensembles.

= *DF triviales*.

2. *Augmentation* :

Si  $K \longrightarrow B$ , alors on a aussi  $KA \longrightarrow B$ .

Si on ajoute un attribut à un déterminant, on obtient encore une DF.

3. *Additivité* :

Si on a  $K \longrightarrow A$  et  $L \longrightarrow B$ , on a aussi  $KL \longrightarrow AB$ .

Si on *additionne* (= union) respectivement les déterminants et les déterminés de deux DF, on obtient encore une DF.

## 3.4 Dépendances fonctionnelles - Propriétés

### 4. *Décomposabilité* :

Si on a  $K \longrightarrow AB$ , on a aussi  $K \longrightarrow A$  et  $K \longrightarrow B$ .

Pour toute DF, il existe une DF entre son déterminant et chaque attribut de son déterminé.

### 5. *Transitivité* :

Si on a  $K \longrightarrow L$  et  $L \longrightarrow M$ , on a aussi  $K \longrightarrow M$ .

La composition de deux DF est encore une DF.

### 6. *Pseudo-transitivité* :

Si on a  $K \longrightarrow L$  et  $LA \longrightarrow M$ , on a aussi  $KA \longrightarrow M$ .

*Autres règles d'inférence : préservation vis à vis des opérateurs, et notamment,*

### 7. *Préservation dans la projection* :

Si  $K \longrightarrow L$  existe dans  $R$ , elle existe aussi dans toute projection de  $R$  qui inclut les attributs de  $K$  et  $L$ .

## 3.4 Dépendances fonctionnelles - Propriétés

En réalité, les propriétés 1, 2 et 5 sont suffisantes. Les autres (3, 4, 6) s'en déduisent.

## 3.4 Dépendances fonctionnelles - Propriétés

ORDRE (FOURN, NPRO, NCOM, ADR, REGION, QTE)

FOURN  $\longrightarrow$  ADR, NPRO

ADR  $\longrightarrow$  REGION

NCOM, NPRO  $\longrightarrow$  QTE

*On a aussi, en appliquant les règles d'inférence :*

FOURN, NCOM  $\longrightarrow$  FOURN, NCOM

ADR, NCOM  $\longrightarrow$  REGION

ADR, NCOM, NPRO  $\longrightarrow$  REGION, QTE

FOURN  $\longrightarrow$  NPRO

FOURN  $\longrightarrow$  REGION

FOURN, NCOM  $\longrightarrow$  QTE

ORDRE[FOURN, NPRO, REGION] : FOURN  $\longrightarrow$  NPRO, REGION

## 3.4 Dépendances fonctionnelles - Propriétés

<b>DF minimale</b>	dont le déterminant est minimal
<b>DF élémentaire</b>	dont le déterminé ne comporte qu'un seul attribut
<b>DF de base</b>	doit être donnée
<b>DF dérivée</b>	est calculable à partir des DF de base à l'aide des règles d'inférence

## 3.4 Dépendances fonctionnelles - DF de base et dérivées

COLIS(NCOL, NCLI, DATE, NOM, ADRESSE)

NCOL  $\longrightarrow$  NCLI

df1

NCOL  $\longrightarrow$  DATE

df2

NCLI  $\longrightarrow$  NOM

df3

NCLI, DATE  $\longrightarrow$  ADRESSE

df4

**DF de base**

*On a aussi, par transitivité :*

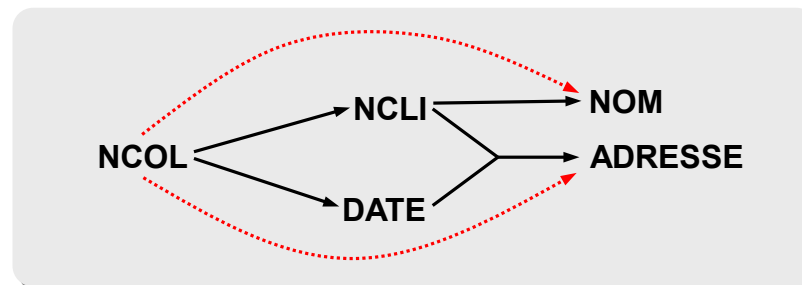
NCOL  $\longrightarrow$  NOM

df5

NCOL  $\longrightarrow$  ADRESSE

df6

**DF dérivées**





## 3.4 Dépendances fonctionnelles - Fermeture et couvertures

### Fermeture d'un ensemble $F$ de DF

ensemble de toutes les DF qu'il est possible de dériver à partir de  $F$  à l'aide des règles d'Armstrong; inclut  $F$ ; il n'existe qu'une seule fermeture;

**toute l'information possible**

### Couverture d'un ensemble $F$ de DF

tout ensemble à partir duquel il est possible de dériver  $F$  à l'aide des règles d'Armstrong;

### Couverture minimale d'un ensemble $F$ de DF

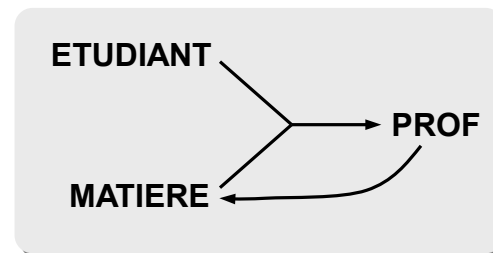
couverture de  $F$  telle qu'aucun de ses sous-ensembles strict n'est une couverture de  $F$ ; il peut exister plusieurs couvertures minimales;

**l'information minimale**

## 3.4 Dépendances fonctionnelles - Circuit de DF

**Le graphe ADF d'une relation peut présenter un ou plusieurs circuits**

INSCRIPTION(ETUDIANT, MATIERE, PROF)  
PROF  $\longrightarrow$  MATIERE  
ETUDIANT, MATIERE  $\longrightarrow$  PROF



**pas d'attributs externes !**

**fréquent  
mais problèmes garantis !**

## 3.5 Contraintes d'inclusion et clés étrangères

## 3.5 Contraintes d'inclusion et clés étrangères

OFFRE		
CHAINE	PRODUIT	PRIX
SUPER-U	Chocolat	6
MAMMOUTH	Sucre	2,2
MAMMOUTH	Chocolat	5
MATCH	Sel	1,4
MAMMOUTH	Sel	1,4

IMPLANTATION	
CHAINE	VILLE
SUPER-U	LILLE
SUPER-U	ANNECY
MAMMOUTH	LYON
MAMMOUTH	LILLE
MATCH	PARIS
MATCH	LILLE
MATCH	TOURNAI
CORA	CHARLEROI

$OFFRE[CHAINE] \subseteq IMPLANTATION[CHAINE]$

## 3.5 Contraintes d'inclusion et clés étrangères

Si les attributs cibles constituent un identifiant de leur relation :  
 = *contrainte référentielle*

CLIENT					
NCLI	NOM	ADRESSE	LOCALITE	(CAT)	COMPTE
B062	GOFFIN	72, r. de la Gare	Namur	B2	-3200
B112	HANSENNE	23, r. Dumont	Poitiers	C1	1250
B332	MONTI	112, r. Neuve	Genève	B2	0
B512	GILLET	14, r. de l'Eté	Toulouse	B1	-8700
C003	AVRON	8, r. de la Cure	Toulouse	B1	-1700
C123	MERCIER	25, r. Lemaître	Namur	C1	-2300
C400	FERARD	65, r. du Tertre	Poitiers	B2	350
D063	MERCIER	201, bvd du Nord	Toulouse		-2250
F010	TOUSSAINT	5, r. Godefroid	Poitiers	C1	0
F011	PONCELET	17, Clos des Erables	Toulouse	B2	0
F400	JACOB	78, ch. du Moulin	Bruxelles	C2	0
K111	VANBIST	180, r. Florimont	Lille	B1	720
K729	NEUMAN	40, r. Bransart	Toulouse		0
L422	FRANCK	60, r. de Wépion	Namur	C1	0
S127	VANDERKA	3, av. des Roses	Namur	C1	-4580
S712	GUILLAUME	14a, ch. des Roses	Paris	B1	0

COMMANDE		
NCOM	NCLI	DATECOM
30178	K111	22/12/2008
30179	C400	22/12/2008
30182	S127	23/12/2008
30184	C400	23/12/2008
30185	F011	2/01/2009
30186	C400	2/01/2009
30188	B512	2/01/2009

COMMANDE[NCLI]  $\subseteq$  CLIENT[NCLI]

## 3.5 Contraintes d'inclusion et clés étrangères

Si les attributs cibles constituent un identifiant de leur relation :  
 = *contrainte référentielle*

CLIENT					
NCLI	NOM	ADRESSE	LOCALITE	(CAT)	COMPTE
B062	GOFFIN	72, r. de la Gare	Namur	B2	-3200
B112	HANSENNE	23, r. Dumont	Poitiers	C1	1250
B332	MONTI	112, r. Neuve	Genève	B2	0
B512	GILLET	14, r. de l'Eté	Toulouse	B1	-8700
C003	AVRON	8, r. de la Cure	Toulouse	B1	-1700
C123	MERCIER	25, r. Lemaître	Namur	C1	-2300
C400	FERARD	65, r. du Tertre	Poitiers	B2	350
D063	MERCIER	201, bvd du Nord	Toulouse		-2250
F010	TOUSSAINT	5, r. Godefroid	Poitiers	C1	0
F011	PONCELET	17, Clos des Erables	Toulouse	B2	0
F400	JACOB	78, ch. du Moulin	Bruxelles	C2	0
K111	VANBIST	180, r. Florimont	Lille	B1	720
K729	NEUMAN	40, r. Bransart	Toulouse		0
L422	FRANCK	60, r. de Wépion	Namur	C1	0
S127	VANDERKA	3, av. des Roses	Namur	C1	-4580
S712	GUILLAUME	14a, ch. des Roses	Paris	B1	0

COMMANDE		
NCOM	NCLI	DATECOM
30178	K111	22/12/2008
30179	C400	22/12/2008
30182	S127	23/12/2008
30184	C400	23/12/2008
30185	F011	2/01/2009
30186	C400	2/01/2009
30188	B512	2/01/2009

DETAIL		
NCOM	NPRO	QCOM
30178	CS464	25
30179	CS262	60
30179	PA60	20
30182	PA60	30
30184	CS464	120
30184	PA45	20
30185	CS464	260
30185	PA60	15
30185	PS222	600
30186	PA45	3
30188	CS464	180
30188	PA45	22
30188	PA60	70
30188	PH222	92

COMMANDE[NCLI]  $\subseteq$  CLIENT[NCLI]  
 DETAIL[NCOM]  $\subseteq$  COMMANDE[NCOM]  
 COMMANDE[NCOM]  $\subseteq$  DETAIL[NCOM]

**COMMANDE[NCOM] = DETAIL[NCOM]**

## 3.5 Contraintes d'inclusion et clés étrangères - Propriétés

1. **Réflexivité :**

$$R[A] \subseteq R[A]$$

2. **Décomposabilité :**

Si on a  $S[A,B] \subseteq R[A,B]$ , on a aussi  $S[A] \subseteq R[A]$  et  $S[B] \subseteq R[B]$

3. **Transitivité :**

Si on a  $T[A] \subseteq S[A]$  et  $S[A] \subseteq R[A]$ , on a aussi  $T[A] \subseteq R[A]$

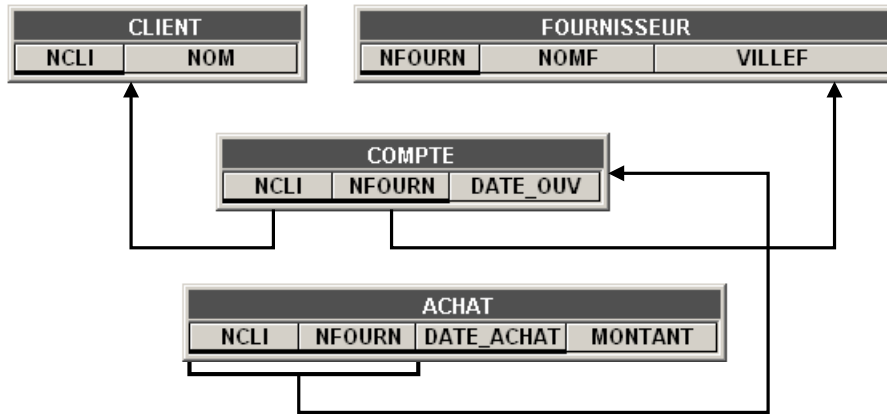
4. **Propagation des DF :**

Si on a  $S[A,B] \subseteq R[A,B]$  et  $R: A \longrightarrow B$ , on a aussi  $S: A \longrightarrow B$ .

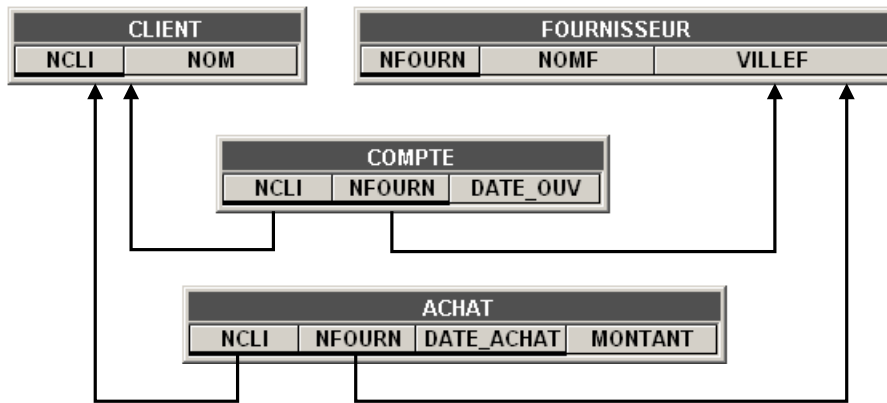
5. **Propagation dans la projection :**

Si on a  $S[A,B] \subseteq R[A,B]$  et  $S'=S[A,B,C]$ , on a aussi  $S'[A,B] \subseteq R[A,B]$

## 3.5 Contraintes d'inclusion et clés étrangères - Propriétés



$COMPTE[NCLI] \subseteq CLIENT[NCLI]$   
 $COMPTE[NFOURN] \subseteq FOURNISSEUR[NFOURN]$   
 $ACHAT[NCLI, NFOURN] \subseteq COMPTE[NCLI, NFOURN]$



$COMPTE[NCLI] \subseteq CLIENT[NCLI]$   
 $COMPTE[NFOURN] \subseteq FOURNISSEUR[NFOURN]$   
 $ACHAT[NCLI] \subseteq CLIENT[NCLI]$   
 $ACHAT[NFOURN] \subseteq FOURNISSEUR[NFOURN]$



## 3.6 Calcul des identifiants d'une relation

## 3.6 Calcul des identifiants d'une relation - Introduction

### Comment déterminer les identifiants d'une relation ?

**Intuitivement, par observation du domaine d'application :**

- NCLI est l'identifiant de la relation CLIENT
- NCOM est l'identifiant de la relation COMMANDE
- NPRO est l'identifiant de la relation PRODUIT

**Facile ! Mais quel est l'identifiant de la relation suivante ?**

ORDRE (FOURN, NPRO, NCOM, ADR, REGION, QTE)

FOURN → ADR, NPRO

ADR → REGION

NCOM, NPRO → QTE

## 3.6 Calcul des identifiants d'une relation - Introduction

**Comment déterminer les identifiants d'une relation ?**

**Les identifiants d'une relation R ne se choisissent pas,  
ils se calculent à partir des dépendances fonctionnelles de R**

## 3.6 Calcul des identifiants d'une relation - Introduction

### Deux observations

- **tout identifiant d'une relation détermine chaque attribut de cette relation**
- **tout groupe d'attributs qui détermine chacun des attributs de la relation, est un identifiant de la relation.**

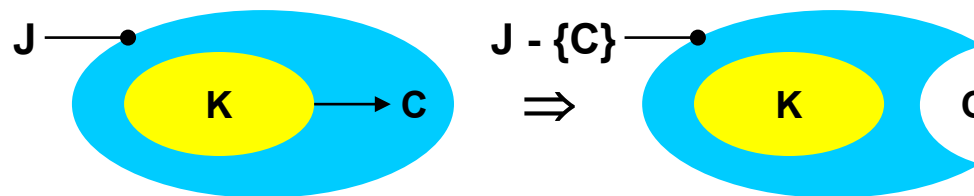
$\text{CLIENT} (\underline{\text{NCLI}}, \text{NOM}, \text{ADRESSE}) \Rightarrow \text{NCLI} \longrightarrow \text{NCLI}, \text{NOM}, \text{ADRESSE}$

$\text{NCLI} \longrightarrow \text{NCLI}, \text{NOM}, \text{ADRESSE} \Rightarrow \text{CLIENT} (\underline{\text{NCLI}}, \text{NOM}, \text{ADRESSE})$

## 3.6 Calcul des identifiants d'une relation - Introduction

### Deux propriétés fondamentales

1. L'ensemble des attributs d'une relation est un identifiant de cette relation.  
 $\{A,B,C\}$  est un identifiant de  $R(A,B,C)$ .
2. Soit  $J$  un identifiant multi-attributs de  $R$ .  
Soit  $K$  un sous-ensemble de  $J$   
Soit  $C$  un attribut de  $J$  extérieur à  $K$ .  
Si  $K \longrightarrow C$ , alors  $J - \{C\}$  est aussi un identifiant de  $R$ .



### Exemple

Si  $R(\underline{A}, \underline{B}, C, D)$  et  $B \longrightarrow C$ , alors  $R(\underline{A}, \underline{B}, C, D)$ .

## 3.6 Calcul des identifiants d'une relation - Identifiant unique

### Procédure élémentaire (Proc1)

1. Un premier identifiant **I** est constitué de l'ensemble des attributs de la relation.
2. On recherche dans **I** un attribut **A externe**; on retire **A** de **I**.
3. On répète l'étape 2 jusqu'à ce qu'il ne soit plus possible de retirer d'attribut à **I**.
4. **I** est un identifiant de la relation.

## 3.6 Calcul des identifiants d'une relation - Identifiant unique (application)

### Application

COM(NCLI, NOM, ADRESSE, NCOM, DATE, NPRO, QTE, PRIX-U)

NCOM  $\longrightarrow$  NCLI

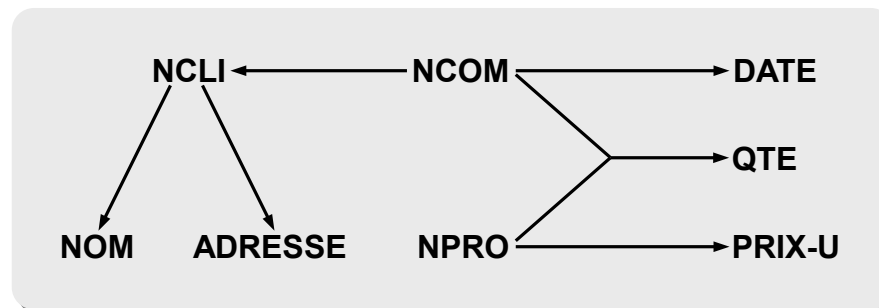
NCLI  $\longrightarrow$  NOM

NCLI  $\longrightarrow$  ADRESSE

NCOM  $\longrightarrow$  DATE

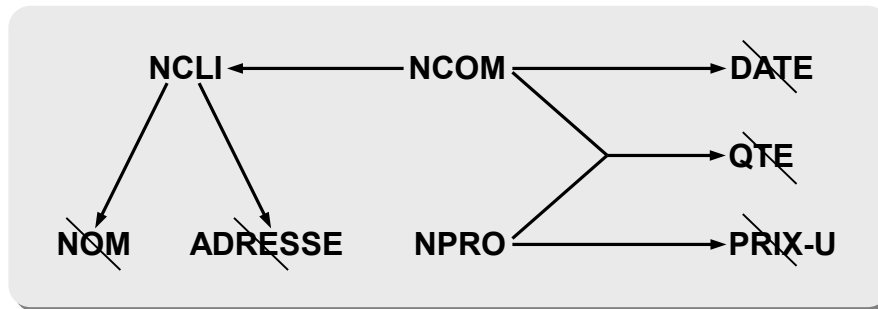
NCOM, NPRO  $\longrightarrow$  QTE

NPRO  $\longrightarrow$  PRIX-U

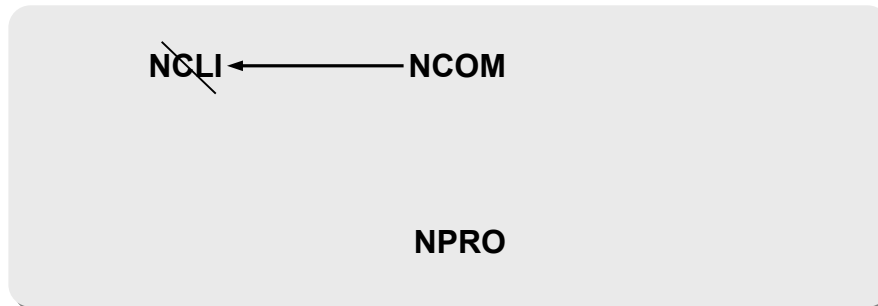


## 3.6 Calcul des identifiants d'une relation - Identifiant unique (application)

### Application



$\text{id}(\text{COM}) = \{\text{NCLI}, \text{NOM}, \text{ADRESSE}, \text{NCOM}, \text{DATE}, \text{NPRO}, \text{QTE}, \text{PRIX-U}\}$



$\text{id}(\text{COM}) = \{\text{NCLI}, \text{NCOM}, \text{NPRO}\}$



## 3.6 Calcul des identifiants d'une relation - Identifiant unique (application)

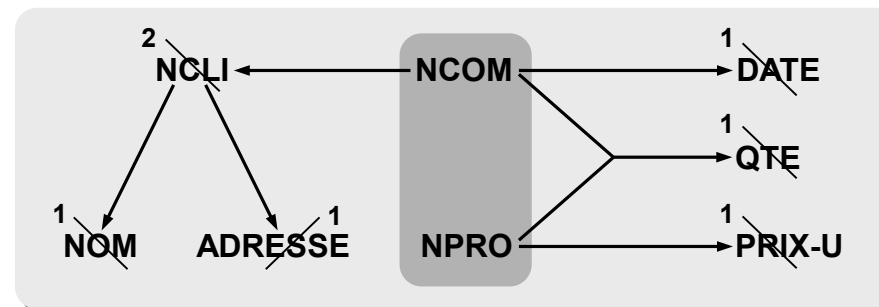
### Application

NCOM

NPRO

$\text{id}(\text{COM}) = \{\text{NCOM}, \text{NPRO}\}$

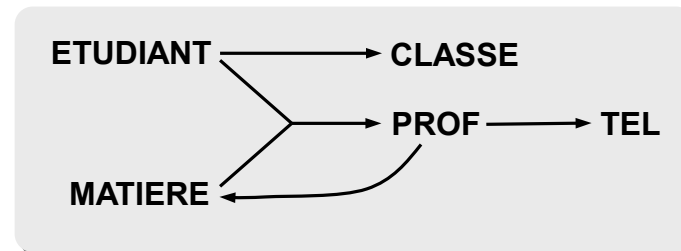
COM(NCLI, NOM, ADRESSE, NCOM, NPRO, DATE, QTE, PRIX-U)



## 3.6 Calcul des identifiants d'une relation - Identifiants multiples

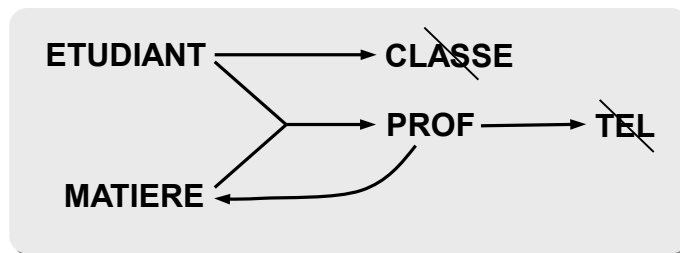
Est-ce que ça marche si la relation possède plus d'un identifiant ?

COURS(MATIERE, ETUDIANT, PROF, CLASSE, TEL)  
ETUDIANT  $\longrightarrow$  CLASSE  
PROF  $\longrightarrow$  TEL  
PROF  $\longrightarrow$  MATIERE  
ETUDIANT, MATIERE  $\longrightarrow$  PROF

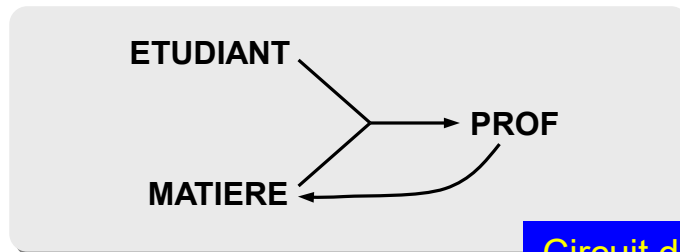


## 3.6 Calcul des identifiants d'une relation - Identifiants multiples

Est-ce que ça marche si la relation possède plus d'identifiant ?



$\text{id}(\text{COURS}) = \{\text{ETUDIANT}, \text{PROF}, \text{MATIERE}, \text{CLASSE}, \text{TEL}\}$



$\text{id}(\text{COURS}) = \{\text{ETUDIANT}, \text{PROF}, \text{MATIERE}\}$

**pas minimal !**

... et ensuite ?

## 3.6 Calcul des identifiants d'une relation - Identifiants multiples

**La règle 2 n'est pas applicable aux attributs d'un circuit !**

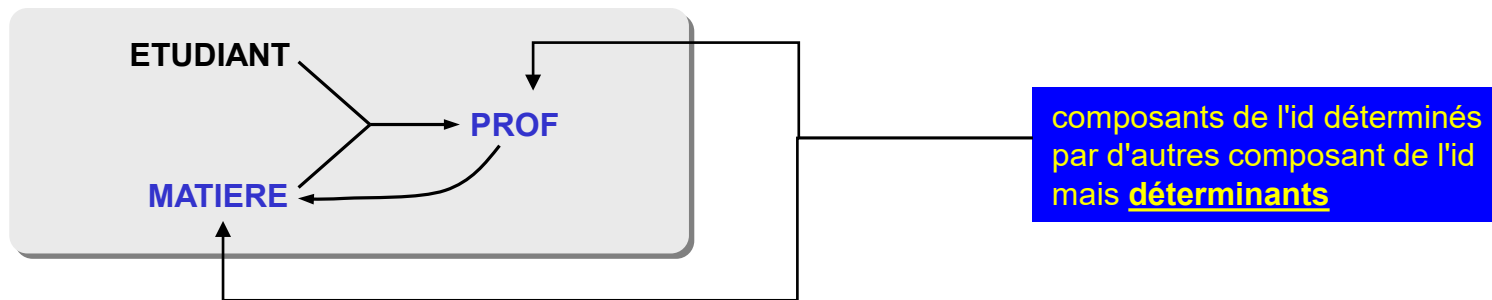
2. On recherche dans  $I$  un **attribut A externe**; on retire  $A$  de  $I$ .

**La procédure Proc1 ne trouve qu'un unique identifiant non minimal**

## 3.6 Calcul des identifiants d'une relation - Identifiants multiples

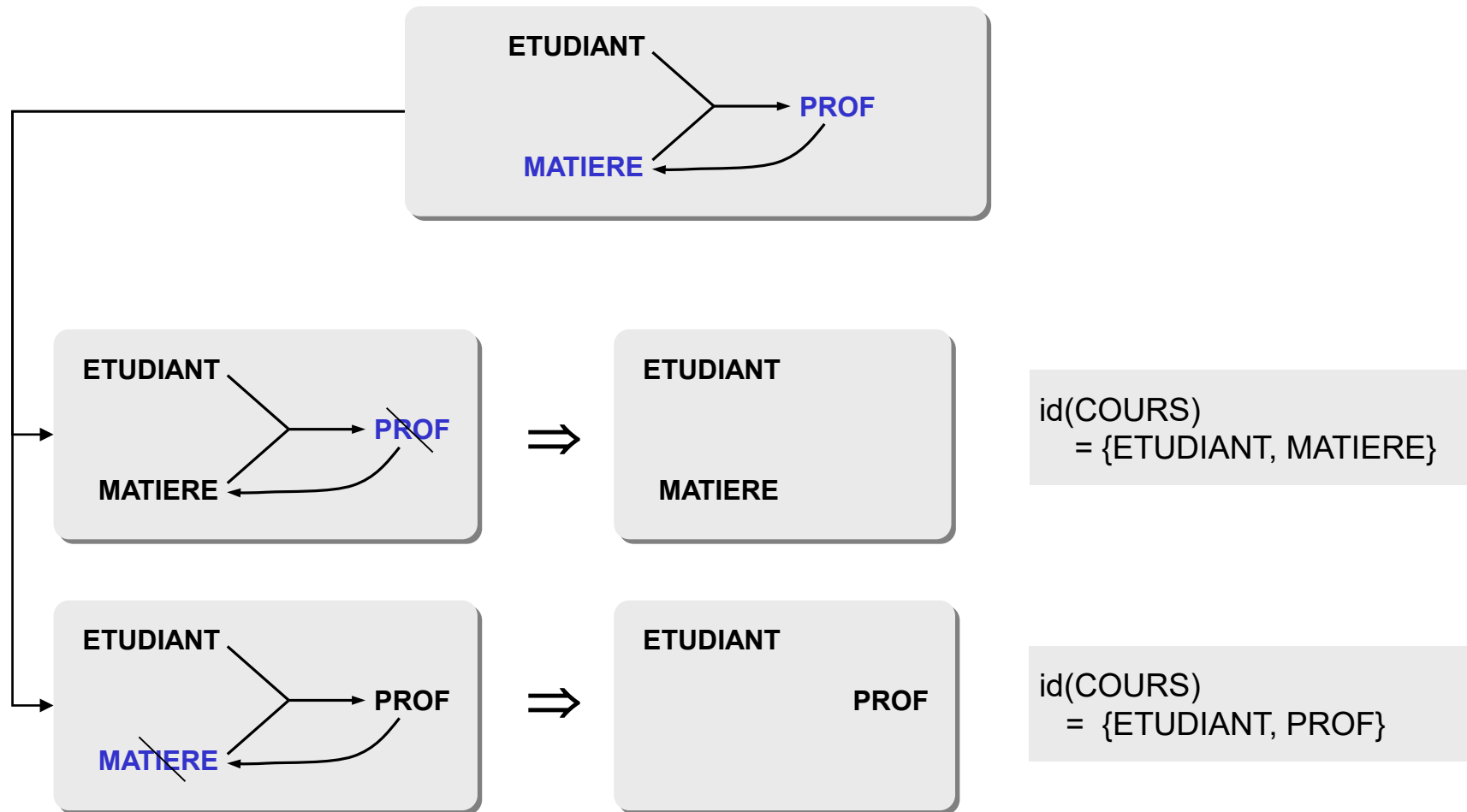
### On reprend la propriété 2, toujours valable

2. Soit  $J$  un identifiant multi-attributs de  $R$ .  
Soit  $K$  un sous-ensemble de  $J$   
Soit  $A$  un attribut de  $J$  extérieur à  $K$ .  
Si  $K \longrightarrow A$ , alors  $J - \{A\}$  est aussi un identifiant de  $R$ .



$\text{id}(\text{COURS}) = \{\text{ETUDIANT}, \text{PROF}, \text{MATIERE}\}$

## 3.6 Calcul des identifiants d'une relation - Identifiants multiples



## 3.6 Calcul des identifiants d'une relation - Identifiants multiples

### Procédure générale (Proc2)

1. Un identifiant est obtenu par application de la procédure **Proc1**
2. Si cet identifiant n'est le siège d'aucune DF, la procédure est terminée
3. Sinon, le graphe ADF résiduel comporte un ou plusieurs circuits. Pour chaque attribut A appartenant à un circuit, on applique l'étape 4
4. On retire A du graphe ADF ainsi que toutes les DF dans lesquelles il apparaît. On applique au graphe résultant la procédure **Proc2**

## 3.7 Décomposition d'une relation



## 3.7 Décomposition d'une relation - Théorème de décomposition

Nous avons *oublié* une propriété très importante des DF !

$$R(A,B,C,D): A \longrightarrow B \quad \Rightarrow \quad R = R[A,B] * R[A,C,D]$$

= *théorème de décomposition*

Les projections d'une relation selon une DF préservent les données

*(puisque'on peut recalculer les données par une jointure)*

## 3.7 Décomposition d'une relation - Théorème de décomposition

On peut donc, sous certaines conditions, remplacer une relation par ses projections

$R(A,B,C,D)$   
 $A \longrightarrow B$



$R1(\underline{A},B)$   
 $R2(A,C,D)$

*pas équivalent*

$R(\text{déterminant}, \text{déterminé}, \text{résidu})$



$R1(\underline{\text{déterminant}}, \text{déterminé})$   
 $R2(\text{déterminant}, \text{résidu})$

$R(A,B,C,D)$   
 $A \longrightarrow B$



$R1(\underline{A},B)$   
 $R2(A,C,D)$   
 $R1[A] = R2[A]$

*équivalence stricte*

## 3.7 Décomposition d'une relation - Théorème de décomposition

En fait, quelle relation entre  $R1[A]$  et  $R2[A]$  ?

$R(A,B,C,D)$   
 $A \longrightarrow B$



$R1(\underline{A},B)$   
 $R2(A,C,D)$   
 $R1[A] \subseteq R2[A]$

$R1(\underline{A},B)$   
 $R2(A,C,D)$   
 $R1[A] = R2[A]$

$R1(\underline{A},B)$   
 $R2(A,C,D)$   
 $R2[A] \subseteq R1[A]$

\*

## 3.7 Décomposition d'une relation - Exemples

ACHAT (CLIENT, PRODUIT, PRIX)  
PRODUIT  $\longrightarrow$  PRIX



TARIF (PRODUIT, PRIX)  
ACHAT (CLIENT, PRODUIT)  
ACHAT[PRODUIT]  $\subseteq$  TARIF[PRODUIT]

ACHAT (CLI, PRO, MAG, PRIX)  
PRO, MAG  $\longrightarrow$  PRIX



TARIF (PRO, MAG, PRIX)  
ACHAT (CLI, PRO, MAG)  
ACHAT[PRO, MAG]  $\subseteq$  TARIF[PRO, MAG]

COM(NCOM, CLI, PRO, QTE)



PASSE (NCOM, CLI)  
COM (NCOM, PRO, QTE)  
PASSE[NCOM] = COM[NCOM]

## 3.7 Décomposition d'une relation - Remarque

### La décomposition ...

ACHAT (CLI, PRO, PRIX)  
PRO  $\longrightarrow$  PRIX

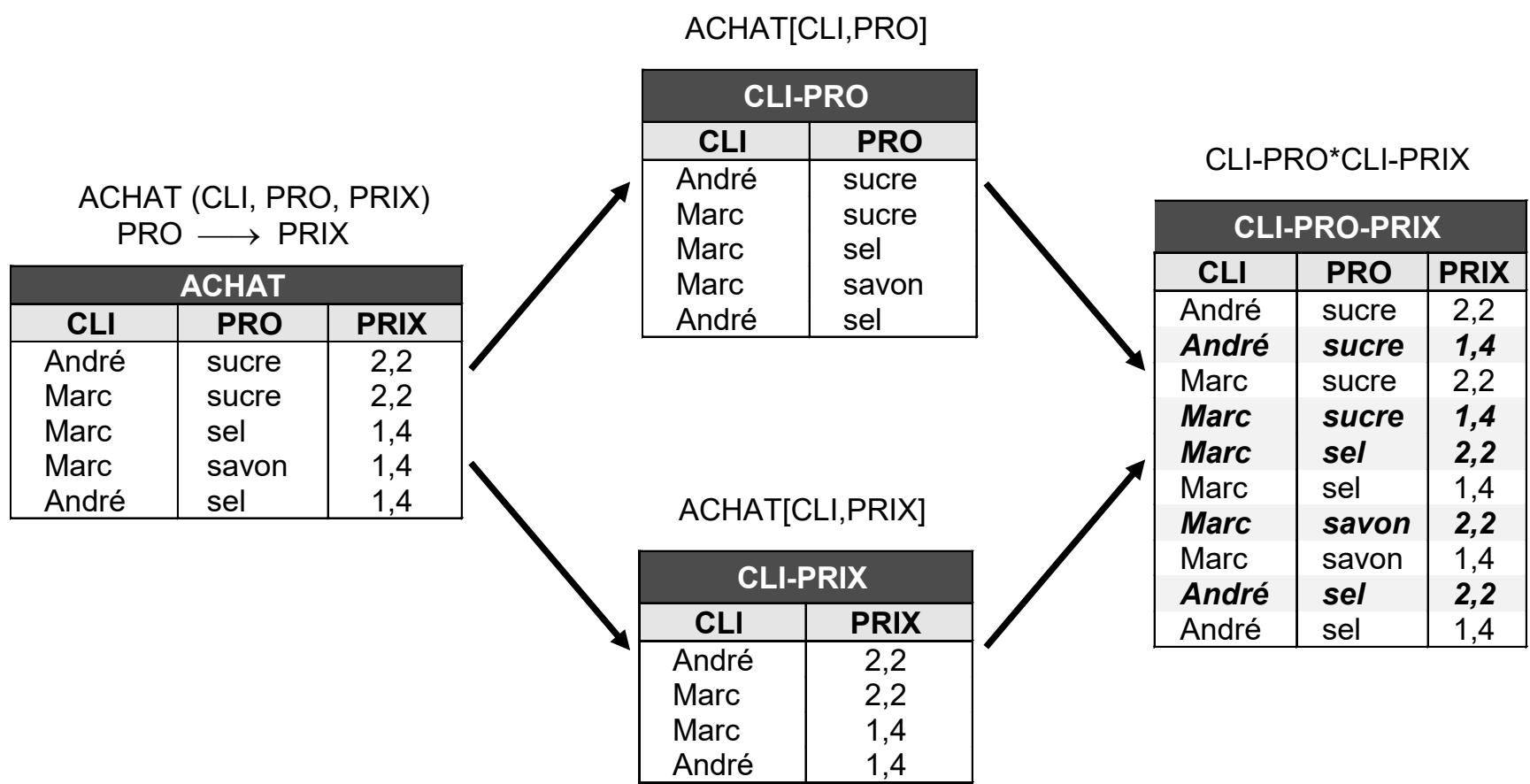


R1 (CLI, PRIX)  
R2 (CLI, PRO)  
R1[CLI] = R2[CLI]

... ne suit pas la règle mais est parfaitement valide. Cependant, elle n'est pas réversible. En toute généralité :

$$\text{ACHAT} \subseteq \text{R1} * \text{R2}$$

### 3.7 Décomposition d'une relation - Exemples



## 3.8 Normalisation d'une relation

## 3.8 Normalisation d'une relation - 1re forme normale

### Avertissement

On ne considère que les relation "plates",  
dont les attributs sont définis sur des **domaines simples**, formés de **valeurs élémentaires**.

ACHAT		
CLIENT	PRODUIT	PRIX
André	Sucre	2,2
Marc	Sucre	2,2
Marc	Sel	1,4
Anne	Savon	1,4
Anne	Sel	1,4

**valeur élémentaire**

Elles sont dites en **1re forme normale** (1FN)

Une relation définie sur des domaines complexes est dite **non en 1re forme normale** (N1FN; en anglais : NFNF ou NF<sup>2</sup>)



## 3.8 Normalisation d'une relation - Observation

ACHAT(CLIENT, PRODUIT, PRIX)

PRODUIT → PRIX

ACHAT		
CLIENT	PRODUIT	PRIX
André	Sucre	2,2
Marc	Sucre	2,2
Marc	Sel	1,4
Anne	Savon	1,4
Anne	Sel	1,4

**Observation** : combien de fois chaque fait élémentaire est-il répété ?

André a acheté du sucre : 1 fois

Marc a acheté du sucre : 1 fois

Marc a acheté du sel : 1 fois

Anne a acheté du savon : 1 fois

Anne a acheté du sel : 1 fois

le sucre coûte 2,2 : **2 fois**

le sel coûte 1,4 : **2 fois**

le savon coûte 1,4 : 1 fois

le pain coûte 2,2 : **0 fois**

**redondance  
et lacunes !**

## 3.8 Normalisation d'une relation - Observation

### Origine du problème

La relation est une structure trop complexe qui représente **deux** types de faits indépendants

On résout le problème en décomposant ACHAT de manière à isoler les deux types de faits

ACHAT'	
CLIENT	PRODUIT
André	Sucre
Marc	Sucre
Marc	Sel
Anne	Savon
Anne	Sel

TARIF	
PRODUIT	PRIX
Sucre	2,2
Sel	1,4
Savon	1,4

ACHAT (CLIENT, PRODUIT, PRIX)  
PRODUIT  $\longrightarrow$  PRIX



TARIF (PRODUIT, PRIX)  
ACHAT' (CLIENT, PRODUIT)  
ACHAT'[PRODUIT]  $\subseteq$  TARIF[PRODUIT]

*la décomposition est légale !*

## 3.8 Normalisation d'une relation - Observation

### Observations

1. La redondance est liée à la DF  $\text{PRODUIT} \longrightarrow \text{PRIX}$
2. La DF  $\text{PRODUIT} \longrightarrow \text{PRIX}$  a ceci de particulier que *son déterminant n'est pas un identifiant de ACHAT*
3. La décomposition selon la DF litigieuse  $\text{PRODUIT} \longrightarrow \text{PRIX}$  règle le problème; on isole la représentation d'un type de faits dans une relation TARIF autonome
4. Une DF qui provoque une redondance est dite **anormale**

## 3.8 Normalisation d'une relation - Relation normalisée

### Définition

Une relation est normalisée si

1. elle est en 1re forme normale
2. **tout déterminant est un identifiant**

Il existe plusieurs formes normales. Celle-ci est la *forme normale de Boyce-Codd*.

## 3.8 Normalisation d'une relation - Relation normalisée

### Les relations suivantes sont-elles normalisées ?

ACHAT(CLIENT, PRODUIT, PRIX)  
PRODUIT  $\longrightarrow$  PRIX


ACHAT (CLIENT, PRODUIT, MAGASIN, PRIX)  
CLIENT, PRODUIT  $\longrightarrow$  PRIX

INSCRIPTION(ETUDIANT, MATIERE, PROF)  
PROF  $\longrightarrow$  MATIERE  
ETUDIANT, MATIERE  $\longrightarrow$  PROF

COM(NCLI, NOM, ADRESSE, NCOM, DATE, NPRO, QTE, PRIX-U)  
NCOM  $\longrightarrow$  NCLI  
NCLI  $\longrightarrow$  NOM  
NCLI  $\longrightarrow$  ADRESSE  
NCOM  $\longrightarrow$  DATE  
NCOM, NPRO  $\longrightarrow$  QTE  
NPRO  $\longrightarrow$  PRIX-U

## 3.8 Normalisation d'une relation - Normalisation d'une relation

### Normalisation : où en est-on ?

	horticulture	modèle relationnel
diagnostic	rosiers atteints par l'oïdium	relation non normalisée
technique	sulfate de cuivre	théorème de décomposition
traitement	pulvérisation bouillie bordelaise	<b>normalisation</b> 
résultat	rosiers sains	relation normalisée

## 3.8 Normalisation d'une relation - Normalisation d'une relation simple

### Procédure de normalisation

**Principe** : on décompose la relation [selon une DF anormale](#)

**Deux questions** :

1. Quelles DF ?
2. Dans quel ordre traite-t-on ces DF ?

#### 1. Quelles DF ?

Les DF de base. On ignore donc les DF dérivées.

#### 2. Dans quel ordre traite-t-on ces DF ?

En commençant par les DF externes.

## 3.8 Normalisation d'une relation - Normalisation d'une relation simple

### Procédure de normalisation

#### Proc3: Normalisation d'une relation R (procédure simplifiée)

1. Dessiner le graphe ADF de la relation réduit aux DF de base.
2. Calculer les identifiants minimaux.
3. Marquer les DF anormales.
4. Tant qu'il existe une **DF anormale externe**  $K \longrightarrow L$ ,
  - générer une relation  $RL(\underline{K}, L)$  et retirer L de R ;
  - préciser la contrainte référentielle.
5. Finaliser



## 3.8 Normalisation d'une relation - Normalisation d'une relation simple

### Application

COM(NCLI, NOM, ADRESSE, NCOM, DATE, NPRO, QTE, PRIX-U)

NCOM  $\longrightarrow$  NCLI

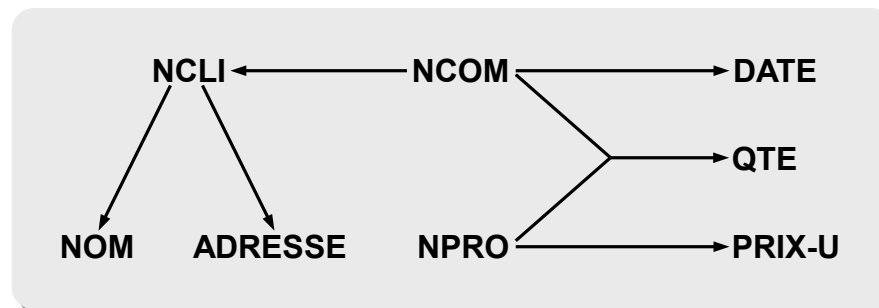
NCLI  $\longrightarrow$  NOM

NCLI  $\longrightarrow$  ADRESSE

NCOM  $\longrightarrow$  DATE

NCOM, NPRO  $\longrightarrow$  QTE

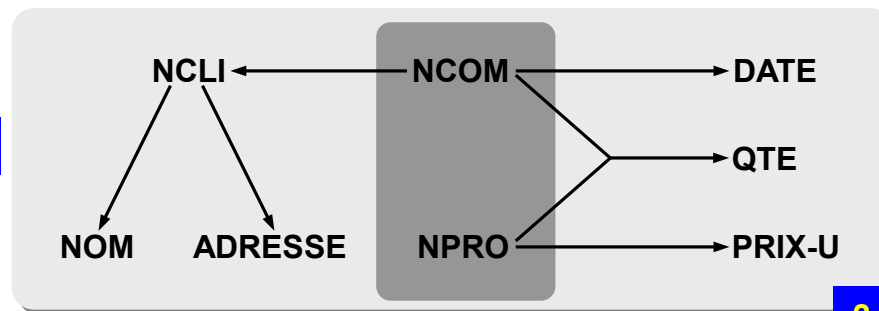
NPRO  $\longrightarrow$  PRIX-U



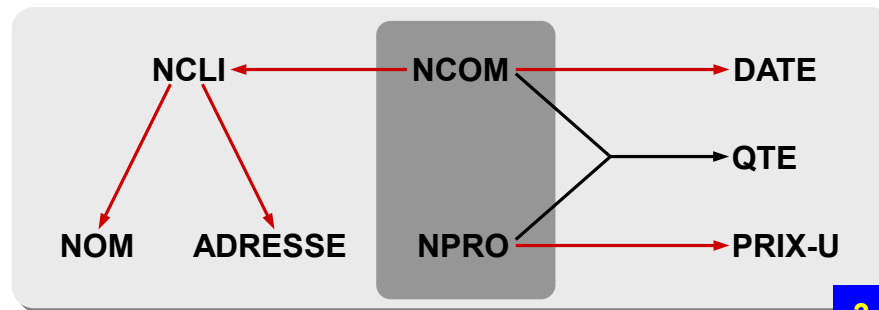
## 3.8 Normalisation d'une relation - Normalisation d'une relation simple

### Application

1. graphe ADF réduit

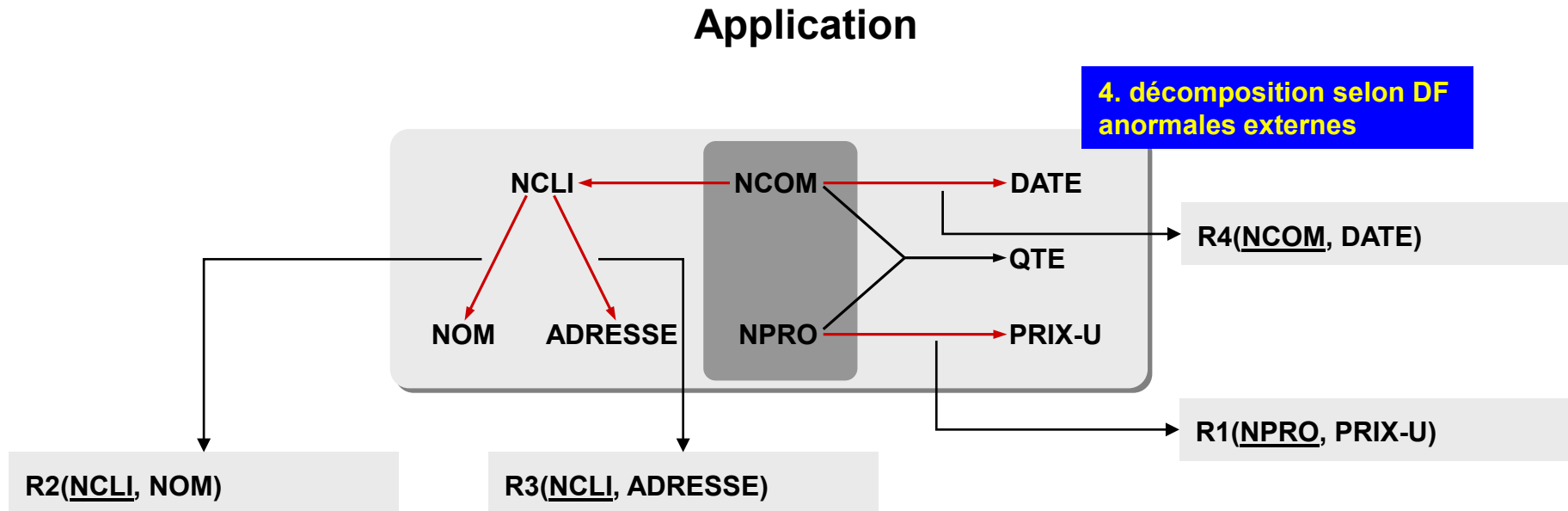


2. identifiant calculé



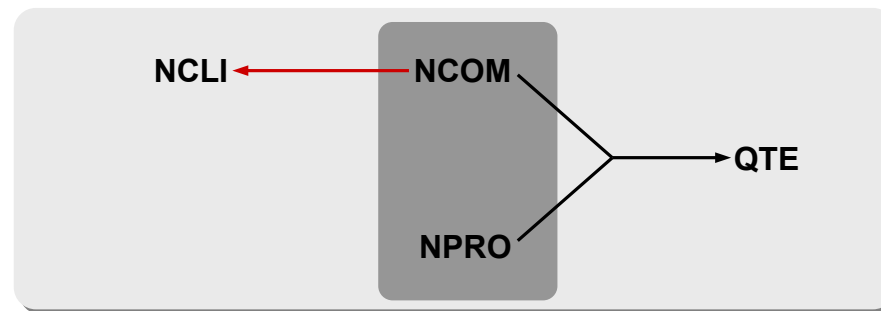
3. DF anormales marquées

## 3.8 Normalisation d'une relation - Normalisation d'une relation simple



## 3.8 Normalisation d'une relation - Normalisation d'une relation simple

### Application



=

COM(NCOM, NPRO, NCLI, QTE)

+

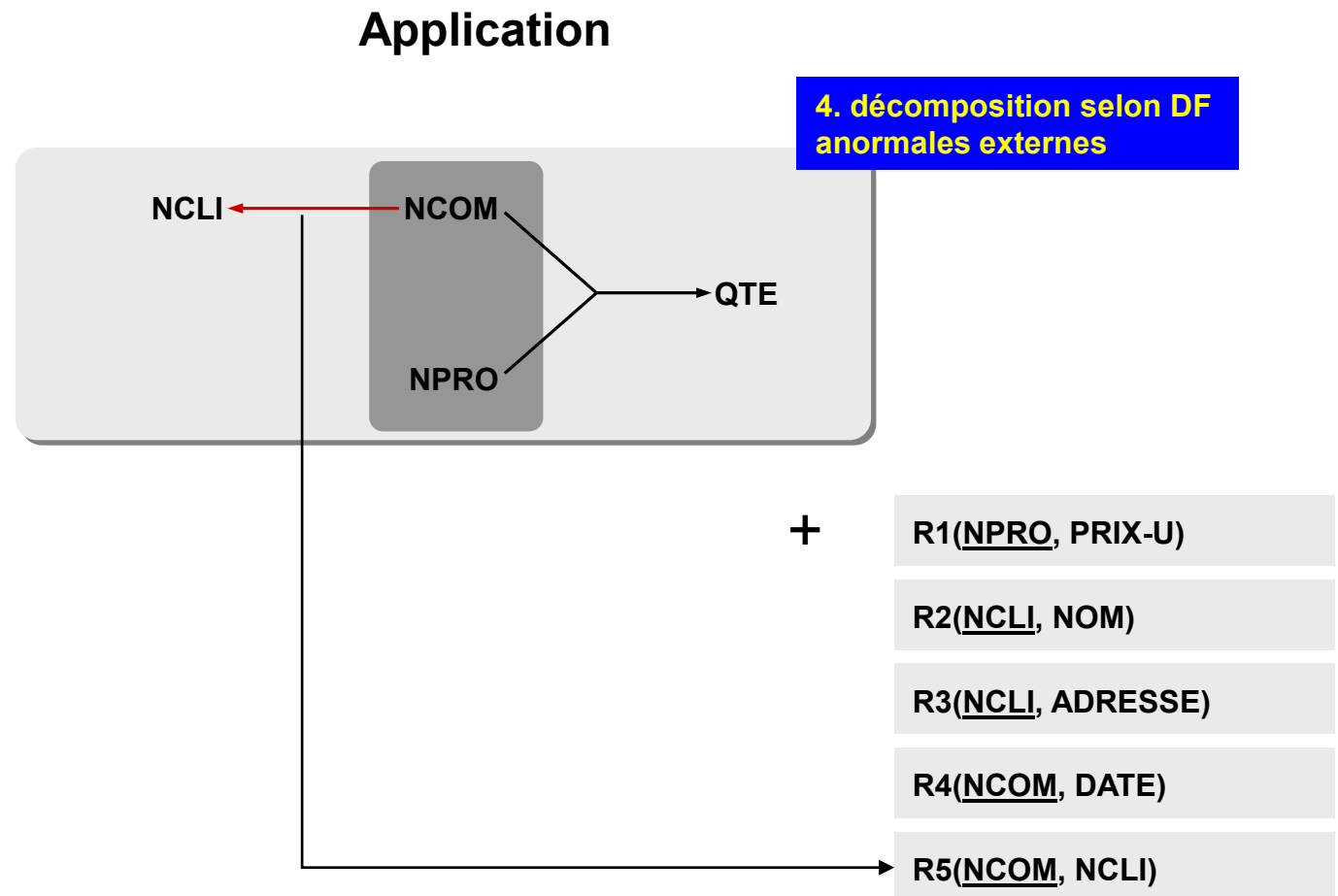
R1(NPRO, PRIX-U)

R2(NCLI, NOM)

R3(NCLI, ADRESSE)

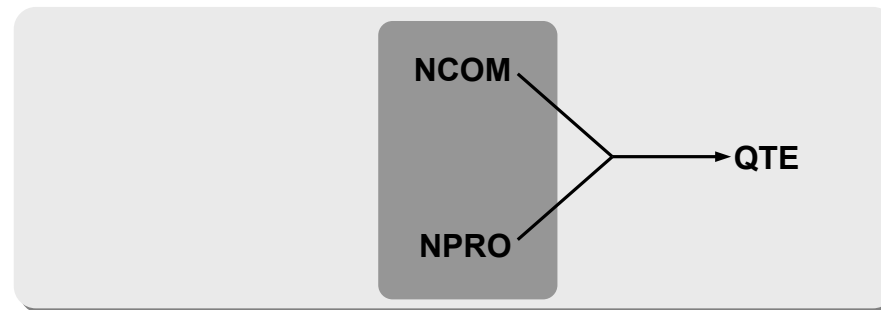
R4(NCOM, DATE)

## 3.8 Normalisation d'une relation - Normalisation d'une relation simple



## 3.8 Normalisation d'une relation - Normalisation d'une relation simple

### Application



=

COM(NCOM, NPRO, QTE)

+

R1(NPRO, PRIX-U)

R2(NCLI, NOM)

R3(NCLI, ADRESSE)

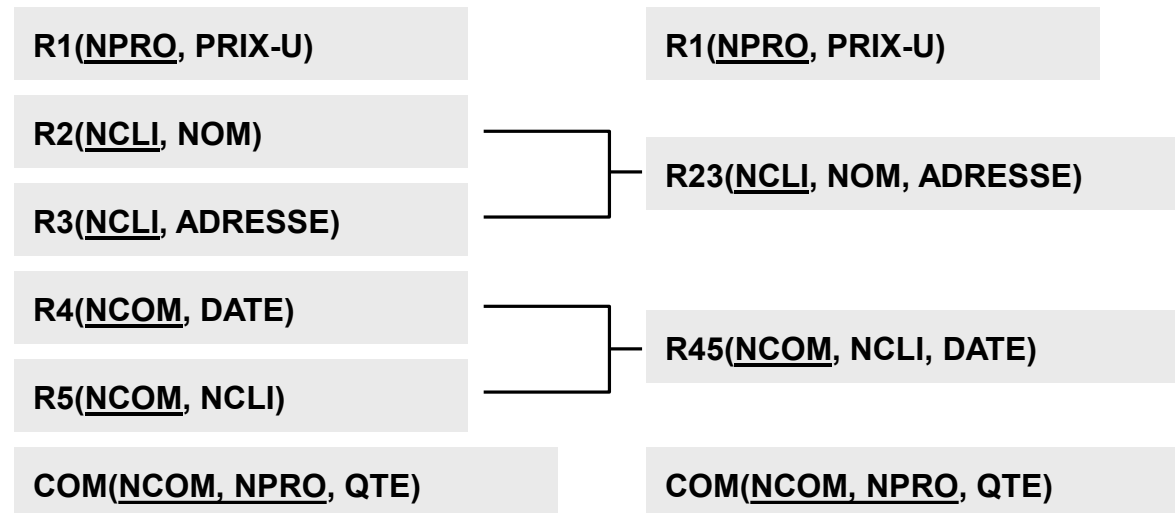
R4(NCOM, DATE)

R5(NCOM, NCLI)

## 3.8 Normalisation d'une relation - Normalisation d'une relation simple

### Application (finalisation)

Regroupement  
selon identifiants



## 3.8 Normalisation d'une relation - Normalisation d'une relation simple

### Application (finalisation)

TARIF(NPRO, PRIX-U)

CLIENT(NCLI, NOM, ADRESSE)

COMMANDE(NCOM, NCLI, DATE)

DETAIL(NCOM, NPRO, QTE)

**Nom significatifs**



## 3.8 Normalisation d'une relation - Normalisation d'une relation simple

### Application (finalisation)

TARIF(NPRO, PRIX-U)

CLIENT(NCLI, NOM, ADRESSE)

COMMANDE(NCOM, NCLI, DATE)

DETAIL(NCOM, NPRO, QTE)

COMMANDE[NCLI]  $\subseteq$  CLIENT[NCLI]  
DETAIL[NCOM] = COMMANDE[NCOM]  
DETAIL[NPRO]  $\subseteq$  TARIF[NPRO]

**Contraintes d'inclusion**

## 3.8 Normalisation d'une relation - Normalisation d'une relation simple

### Procédure de normalisation (complétée)

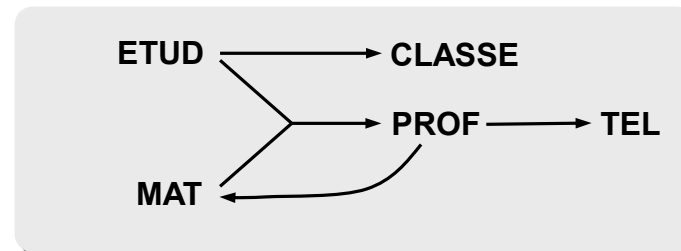
#### Proc3: Normalisation d'une relation R (procédure simplifiée)

1. Dessiner le graphe ADF de la relation réduit aux DF de base.
2. Calculer les identifiants minimaux.
3. Marquer les DF anormales.
4. Tant qu'il existe une **DF anormale externe**  $K \longrightarrow L$ ,
  - générer une relation  $RL(K, L)$  et retirer  $L$  de  $R$  ;
  - préciser la contrainte référentielle.
5. Finaliser :
  - regrouper selon même identifiant ;
  - noms significatifs;
  - ajuster contraintes référentielles (dont les clés étrangères).

## 3.8 Normalisation d'une relation - Normalisation d'un noyau irréductible

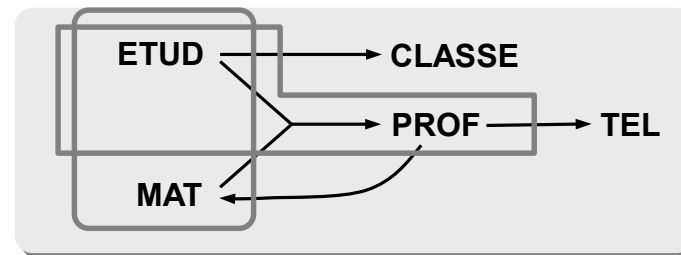
**Rien n'est simple, tout se complique !**

INSCR(MAT, ETUD, PROF, CLASSE, TEL)  
ETUD  $\longrightarrow$  CLASSE  
PROF  $\longrightarrow$  TEL  
PROF  $\longrightarrow$  MAT  
ETUD, MAT  $\longrightarrow$  PROF

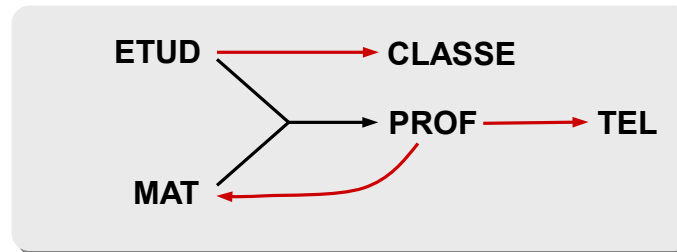


## 3.8 Normalisation d'une relation - Normalisation d'un noyau irréductible

1. le graphe ADF est réduit

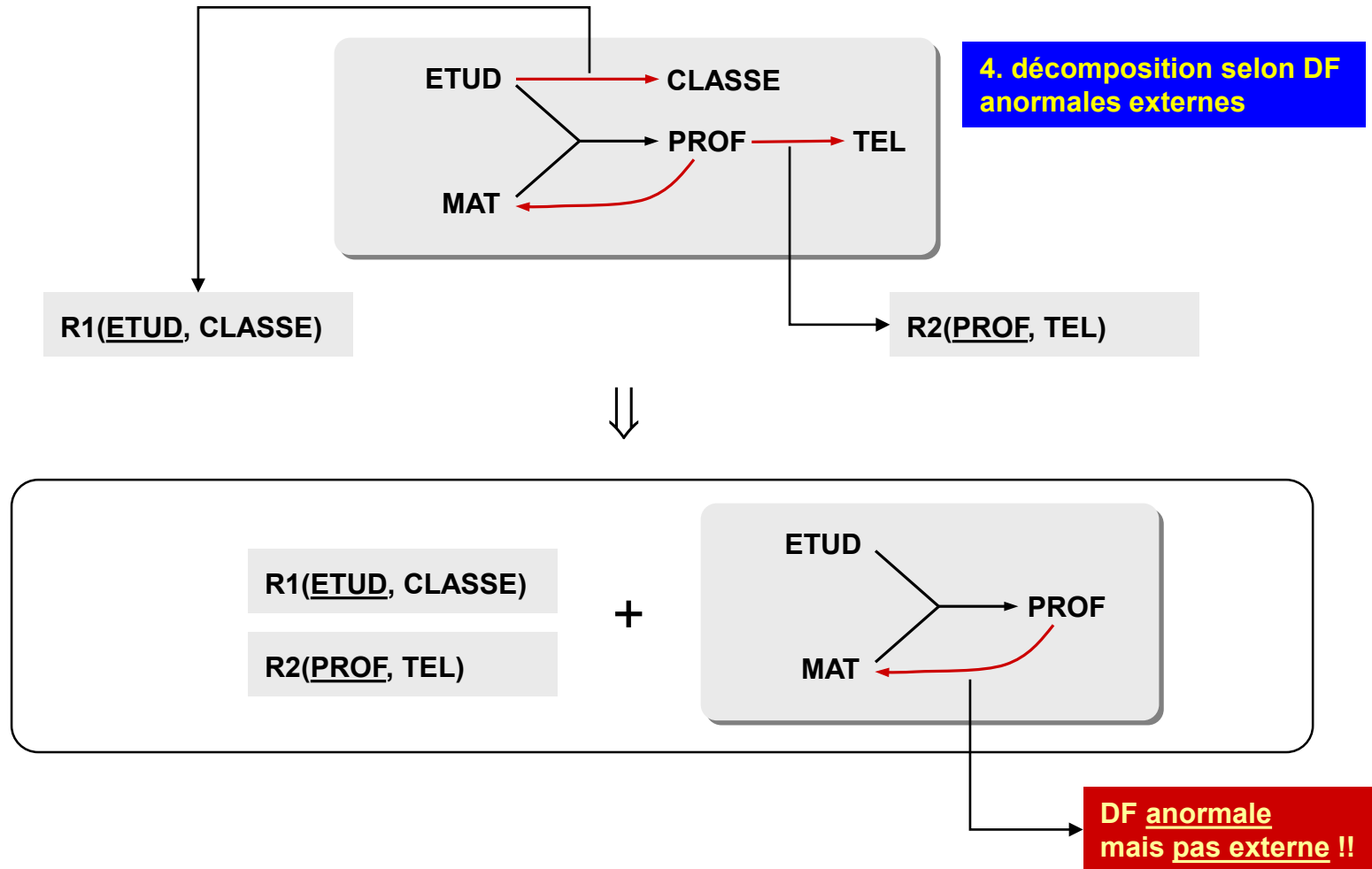


2. les identifiants sont calculés



3. les DF anormales sont marquées

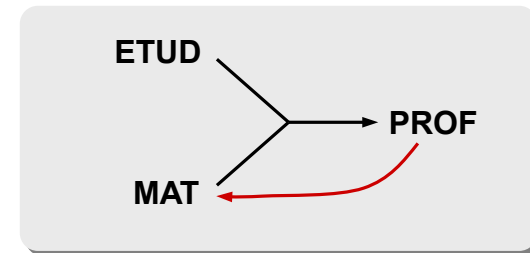
## 3.8 Normalisation d'une relation - Normalisation d'un noyau irréductible



## 3.8 Normalisation d'une relation - Normalisation d'un noyau irréductible

Que faire d'un noyau irréductible mais anormal ?

INSCR(MAT, ETUD, PROF)  
PROF  $\longrightarrow$  MAT  
MAT, ETUD  $\longrightarrow$  PROF



**Mauvaise surprise** : il n'y a pas de solutions totalement satisfaisantes !

**Trois solutions**

1. la **peste**
2. le **choléra**
3. la **peste et le choléra**

## 3.8 Normalisation d'une relation - Normalisation d'un noyau irréductible

### La peste

On conserve le noyau tel quel.

INSCR(MAT, ETUD, PROF)  
PROF  $\longrightarrow$  MAT  
MAT, ETUD  $\longrightarrow$  PROF



INSCR(MAT, ETUD, PROF)  
PROF  $\longrightarrow$  MAT

- Observations :**
1. 😞 on conserve la redondance due à la DF anormale
  2. 😊 toutes les DF sont conservées
  3. 😐 en SQL, la DF anormale peut être gérée par des déclencheurs

## 3.8 Normalisation d'une relation - Normalisation d'un noyau irréductible

### Le choléra

On force la décomposition en affaiblissant la condition "**DF anormale externe**".

```
INSCR(MAT, ETUD, PROF)
PROF → MAT
MAT, ETUD → PROF
```

**DF anormale (interne)**



```
COURS(PROF, MAT)
INSCR(ETUD, PROF)
INSCR[PROF] ⊆ COURS[PROF]
COURS*INSCR: MAT, ETUD → PROF
```

- Observations :**
1. 😊 on élimine la redondance due à la DF anormale
  2. 😞 une DF est perdue
  3. 😞 nouvelle contrainte très complexe (DF sur une vue !)
  4. 😊 en SQL, la DF perdue peut être gérée par des déclencheurs



## 3.8 Normalisation d'une relation - Normalisation d'un noyau irréductible

### La peste et le choléra

On fusionne les deux solutions précédentes.

```
INSCR(MAT, ETUD, PROF)
PROF → MAT
MAT, ETUD → PROF
```



```
COURS(PROF, MAT)
INSCR(MAT, ETUD, PROF)
INSCR[PROF, MAT] ⊆ COURS[PROF, MAT]
```

- Observations :**
1. 😞 on conserve la redondance due à la DF anormale
  2. 😞 on introduit une redondance supplémentaire (COURS)
  3. 😊 toutes les DF sont conservées
  4. 😊 en SQL, la contrainte d'inclusion peut s'exprimer simplement

## 3.8 Normalisation d'une relation - Normalisation d'un noyau irréductible

### La peste et le choléra

La contrainte d'inclusion peut s'exprimer simplement et **complètement** en SQL2 pur (sans triggers)

```
COURS(PROF, MAT)  
INSCR(MAT, ETUD, PROF)  
INSCR[PROF, MAT]  $\subseteq$  COURS[PROF, MAT]
```



```
create table COURS (PROF char(20) not null,  
                  MAT char(20) not null,  
                  primary key (PROF),  
                  unique (PROF, MAT))  
  
create table INSCR (MAT char(20) not null,  
                  ETUD char(20) not null,  
                  PROF char(20) not null,  
                  primary key (MAT, ETUD),  
                  foreign key (PROF, MAT) references COURS (PROF, MAT));
```

**pas glorieux  
mais efficace**

## 3.8 Normalisation d'une relation - Normalisation d'un noyau irréductible

### Synthèse

INSCR(MAT, ETUD, PROF)  
PROF  $\longrightarrow$  MAT  
MAT, ETUD  $\longrightarrow$  PROF



INSCR(MAT, ETUD, PROF)  
PROF  $\longrightarrow$  MAT

peste

COURS(PROF, MAT)  
INSCR(ETUD, PROF)  
INSCR[PROF]  $\subseteq$  COURS[PROF]  
COURS\*INSCR: MAT, ETUD  $\longrightarrow$  PROF

choléra

COURS(PROF, MAT)  
INSCR(MAT, ETUD, PROF)  
INSCR[PROF, MAT]  $\subseteq$  COURS[PROF, MAT]

peste et choléra

## 3.8 Normalisation d'une relation - Normalisation d'un noyau irréductible

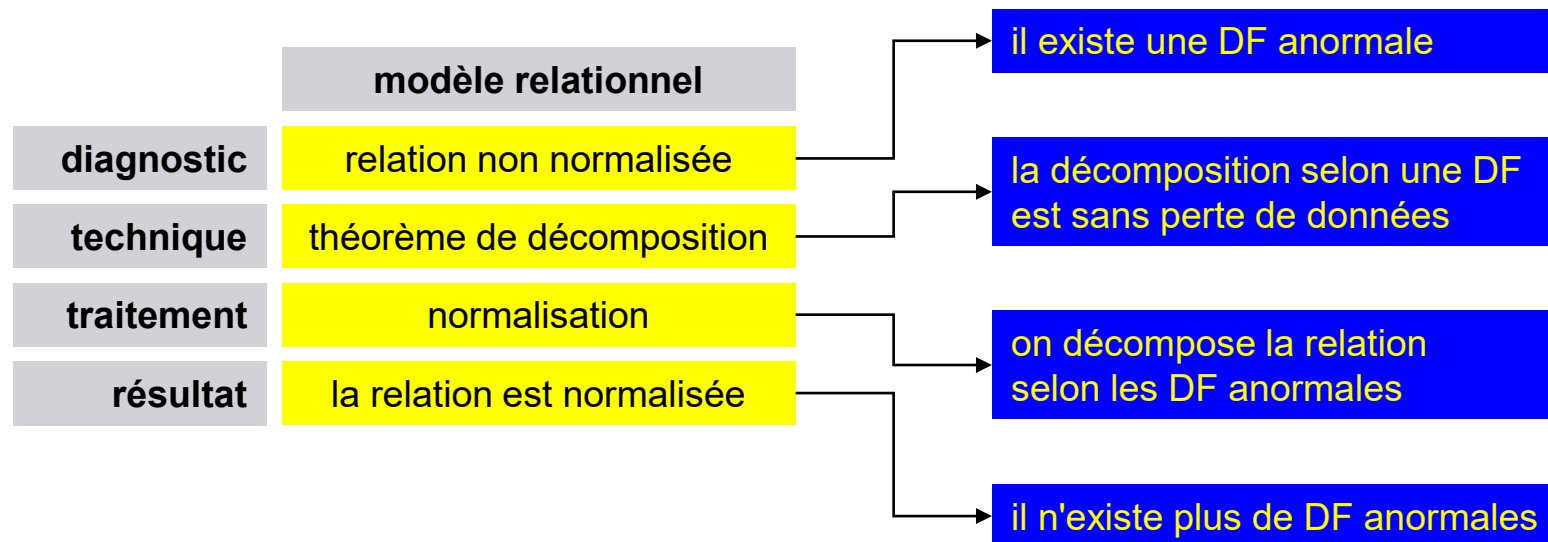
La littérature a donné à ces formes des noms moins agressifs

### Trois solutions

- 1. la **peste** → 3e forme normale (3FN)
- 2. le **choléra** → forme normale de Boyce-Codd (FNBC)
- 3. la **peste et le choléra** → forme normale à clé élémentaire (FNCE)

## 3.8 Normalisation d'une relation - Normalisation d'une relation

### Normalisation : résumé



## 3.8 Normalisation d'une relation - Autres formes normales

La théorie de la normalisation se réduit-elle à cela ?

On relève (au moins) 7 formes normales

1. 1re forme normale (relation plate)
2. 2e forme normale (pré-3e forme normale)
3. 3e forme normale (la *peste*)
4. forme normale à clé élémentaire (la *peste* et le *choléra*)
5. **forme normale de Boyce-Codd** (le *choléra*)
6. 4e forme normale (basée sur les dépendances multivaluées)
7. 5e forme normale (basée sur les dépendances de jointure)

**généralisent des DM**

**généralisent des DF**

## Fin du module 3

**Module(s) suivant(s) :**

***Partie I, mod. 4 : Implémentation des structures de données***

**ou**

***Partie II, mod. 1 : Le langage SQL DDL***

