



IF AND ONLY IF

# **Systèmes de Gestion des Bases de Données Relationnelles**

Lamine DIOP

Associate Professor at EPITA Engineering School

EPITA Research Laboratory, France

[lamine.diop@epita.fr](mailto:lamine.diop@epita.fr)

Web page: <https://sites.google.com/ugb.edu.sn/laminediop>

# Acquis d'apprentissage visé

- Comprendre les différentes étapes de la modélisation d'une base de données relationnelles
- Produire un modèle conceptuel et un modèle logique à partir d'un cahier de charge
- Apprécier l'exactitude d'un modèle relationnel
- Mettre en place une base de données cohérente normalisée

# Plan & Organisation

## Partie 1

- Algèbre relationnelle
- Contraintes d'intégrité
- Dépendances fonctionnelles (DF)
- Couvertures minimales

- CM
- TD: Analyse critique de modèles relationnels
- TP: Elaboration du dictionnaire des données et calcul de DFs

## Partie 2

- Les formes normales
- Dépendances multivariées
- Modélisation conceptuelle

- CM
- TD: Modélisation et normalisation
- TP: Conception d'une base de données relationnelles normalisée *from scratch*.

# Mode d'évaluation

- TP & TD : 2 parties à rendre et notées chacune par groupe de 2 à 3
- Projet
- Examen (28/03/2025)
- Formule de calcul de la note finale :  $(TD\&TP + 2*Projet + 2*Examen)/5$

# Plan & Organisation

## Partie 1

- Algèbre relationnelle
- Contraintes d'intégrité
- Dépendances fonctionnelles (DF)
- Couvertures minimales

- CM
- TD: Analyse critique de modèles relationnels
- TP: Elaboration du dictionnaire des données et calcul de DFs

## Partie 2

- Les formes normales
- Dépendances multivariées
- Modélisation conceptuelle

- CM
- TD: Modélisation et normalisation
- TP: Conception d'une base de données relationnelles normalisée *from scratch*.

# Bases de données (BD)

**Définition:** Ensemble structuré et organisé permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation (ajout, mise à jour, recherche de données)

Le terme database serait apparu au milieu des années 1960 dans le domaine des systèmes d'information militaires.

# Bases de données (BD)

- Exemples de structuration
  - Bases de données hiérarchiques
  - Bases de données réseaux
  - **Bases de données relationnelles**
  - Bases de données orientées objet
  - Bases de données multidimensionnelles

# Système de Gestion de Base de Données (SGBD)

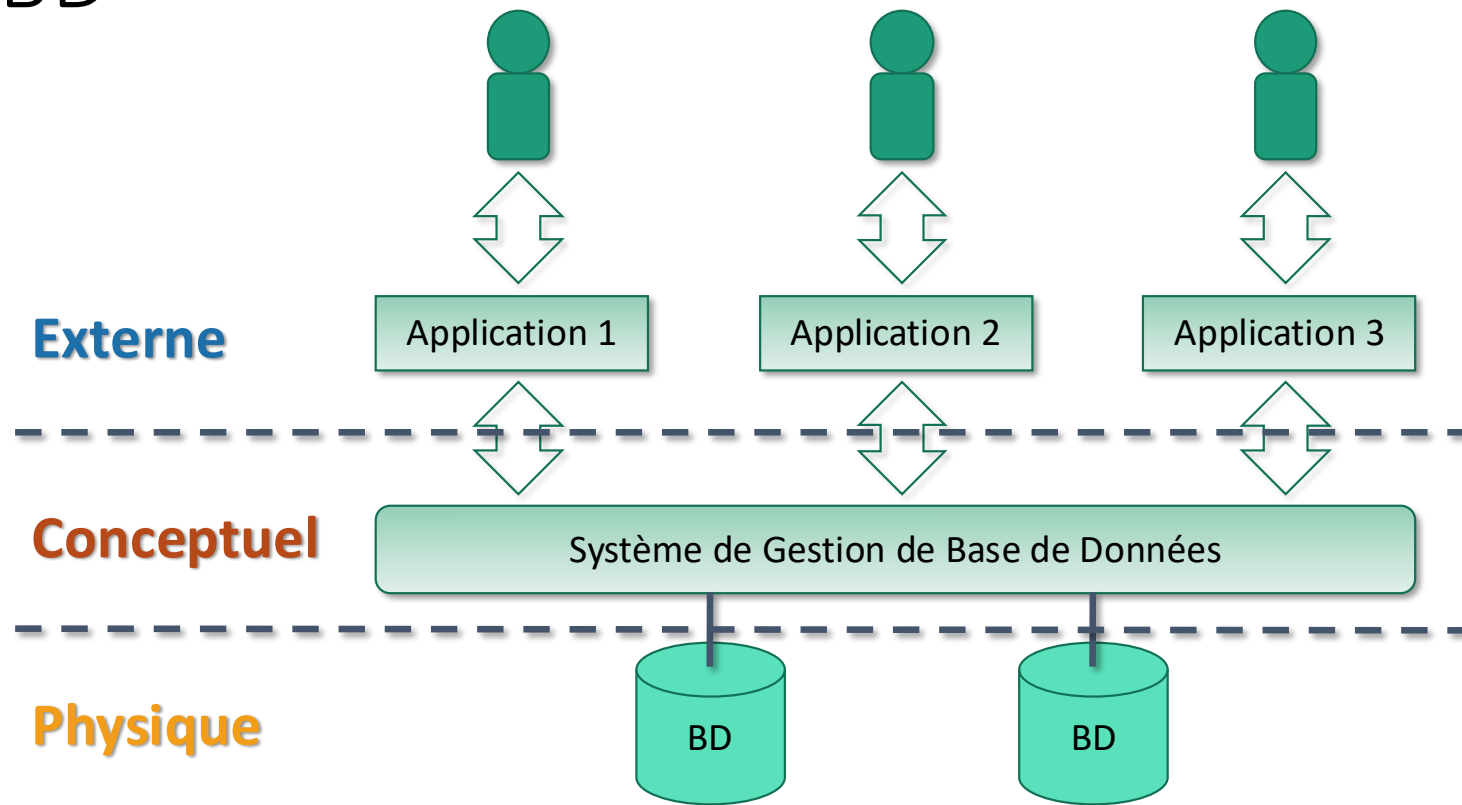
## **Ensemble de programmes assurant la gestion et l'accès à une base de données**

- Système de gestion de fichiers
- Système d'exécution de requêtes
- Interface utilisateurs (ou applications)

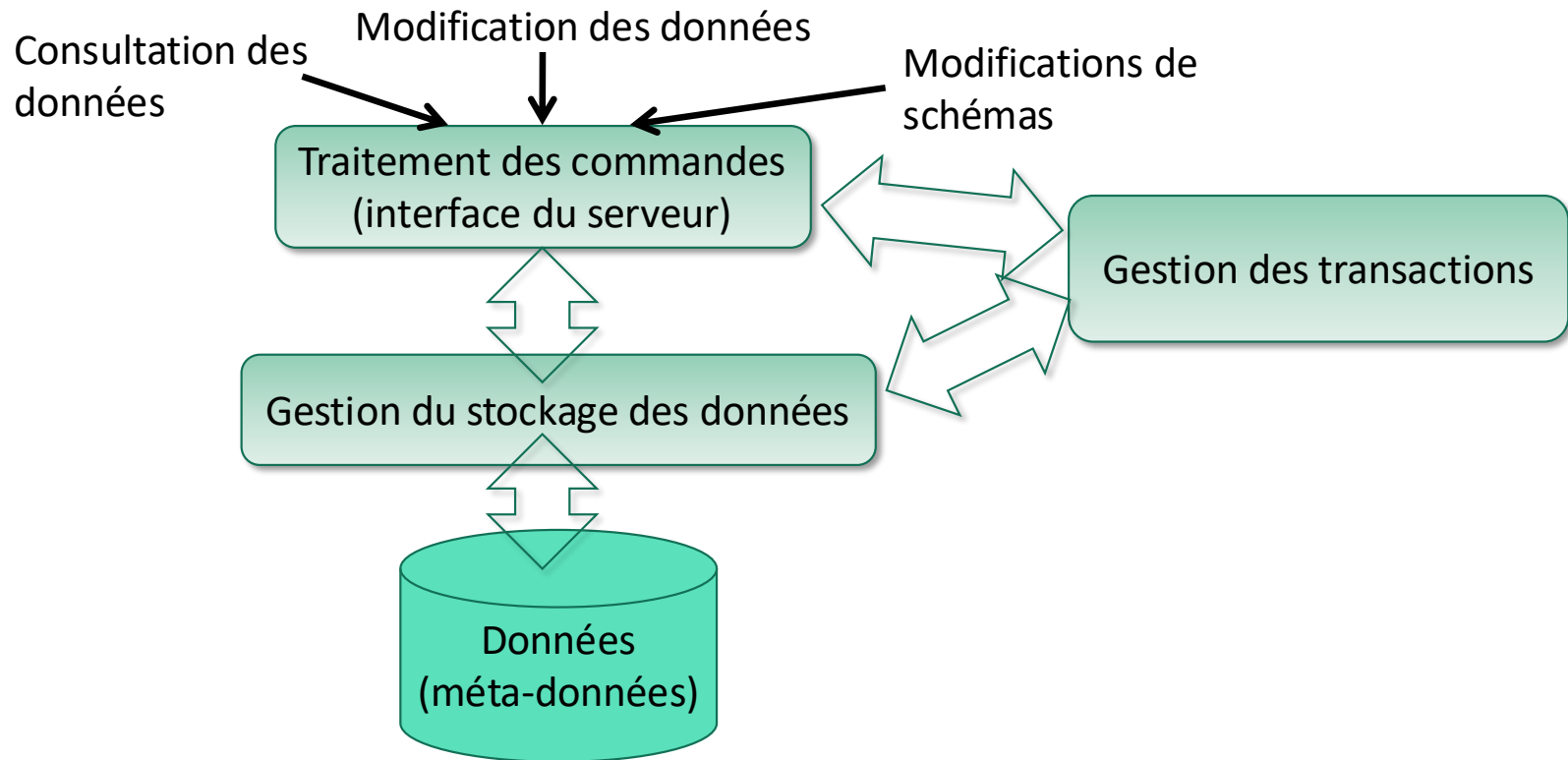
Gestion de plusieurs bases de données destinées à des logiciels ou des thématiques différentes



# SGBD



# SGBD : architecture



# SGBD : fonctionnalités (1)

- **Indépendance logique**

Modification du schéma conceptuel sans remettre en cause les schémas externes ou les programmes d'application

- **Indépendance physique**

Modification du schéma physique sans remettre en cause le schéma conceptuel

- **Cohérence des données**

Modification de la base de données sans remettre en cause les règles de cohérence

## SGBD : fonctionnalités (2)

- **Partage des données**

Rendre transparent le partage des données entre différents utilisateurs

- **Souplesse d'accès aux données**

Fournir un langage adressant les concepts du modèle

- **Performances**

Accès efficace (rapidité) à des volumes de données important (capacité de stockage)

- **Contrôle**

Offrir des outils d'administration pour garantir la confidentialité de certaines données (utilisateur, groupe d'utilisateurs)

# SGBD : architecture

- Traitement des commandes
  - Systèmes de caches
  - Ré-écriture de requêtes
  - Application d'index
- Gestionnaire de stockage
  - Limiter les accès disques (buffers)
  - Optimiser les accès disques

# SGBD : langages

- **Langage de définition de données (LDD)**  
Permet la définition du schéma interne et externe (vues)
- **Langage de contrôle de données (LCD)**  
Permet l'administration des utilisateurs et des groupes
- **Langage d'interrogation de données (LID)**  
Permet d'interroger la base de données
- **Langage de manipulation de données (LMD)**  
Permet la mise à jour et l'ajout de données

# SGBD-R

## **SGBD-R ou SGBD relationnelles**

- Bases de données organisées sous forme de tableaux
- Langage SQL (Structured Query Language)
- Exemples : PostgreSQL, MySQL, Oracle Database, Microsoft Access, IBM DB2, Sybase, dBase, SQLite...



# SGBD-R : Modèle relationnel

- Modèle abstrait créé par Codd en 1970 (IBM)
- **Ensembles** de tables (**relations**) avec des lignes (enregistrements/nuplets/**tuples**) et des colonnes (**attributs**)
- Exemple de table: `Client (NumClient, Nom, ville)`

Carnet de clients

NumClient	Nom	Ville
001	Albert	Dakar
002	Moussa	New York
003	Olivier	Paris



# Modélisation d'un Système d'Information

- Modèle logique associé aux SGBD relationnelles (ex : Oracle, SQL Server, Acces, DBASE, etc.)
- Fondé sur la théorie mathématique des ensembles et la notion de base qui lui est rattachée : la relation
- représentation de l'information selon trois composantes :
  - des structures de données
  - des contraintes qui permettent de spécifier les règles qui doivent respecter les données de la base de données
  - des opérations qui pour manipuler les données en interrogation et en mise en jour
- Modèle relationnel assure l'indépendance logique et physique

# Rappels mathématiques

- ❖ Un ensemble est une collection d'éléments
- ❖ On note par
  - $x \in S$  :  $x$  appartient à l'ensemble  $S$
  - $\emptyset$  : l'ensemble vide
  - $|S|$  cardinalité de l'ensemble  $S$
  - $\mathcal{P}(S)$  ou  $2^S$  : l'ensemble des parties de l'ensemble  $S$
- ❖ Opérations sur les ensembles:  $\cap, \cup, -, \times$
- ❖ Compareurs ensemblistes:  $=, \neq, \subset, \not\subset$

# Rappels mathématiques

- **Partition**

Soit  $S$  un ensemble non vide et  $I$  un intervalle, une partition de  $S$  est une famille d'ensembles  $\{S_i \mid i \in I\}$  telle que

1.  $\bigcup_{i \in I} S_i = S$
2.  $S_i \cap S_j = \emptyset$  pour tout  $i \neq j$
3.  $S_i \neq \emptyset$  pour tout  $i \in I$

Exemple

$$S = \{a, b, c, d\}$$

$S_1 = \{a, c\}$  et  $S_2 = \{b, d\}$  forment une partition de  $S$

$S_1 = \{a, b\}$  et  $S_2 = \{b, c, d\}$  ne forment pas une partition de  $S$

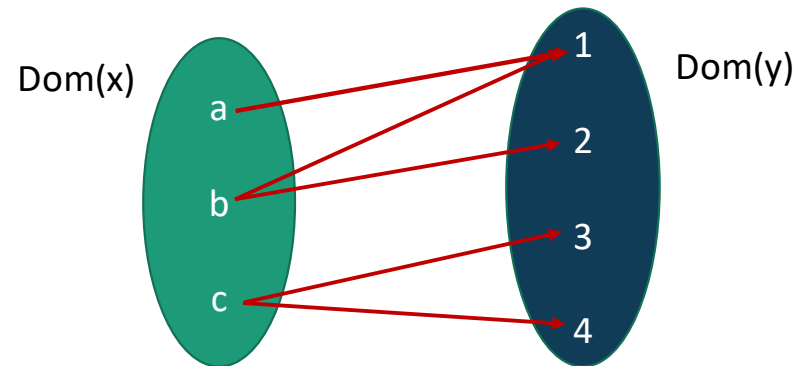
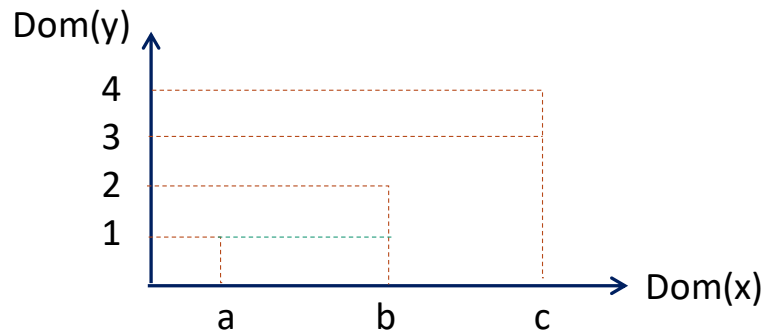
# Relation binaire

- Définition : Une relation binaire  $R$  sur un ensemble  $S$  est un sous-ensemble de  $S \times S$ .

Donc une relation est un ensemble.

On note  $R(x, y)$  pour indiquer que  $(x, y) \in R$ .

On appelle  $(x, y)$  couple ou doublet ou 2-uplet.



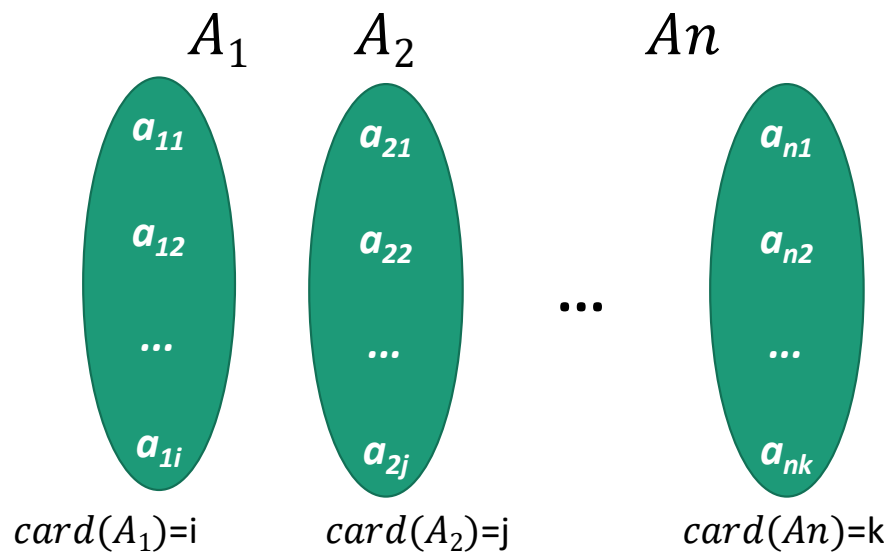
# Relation n-aire

## **Définition (relation n-aire)**

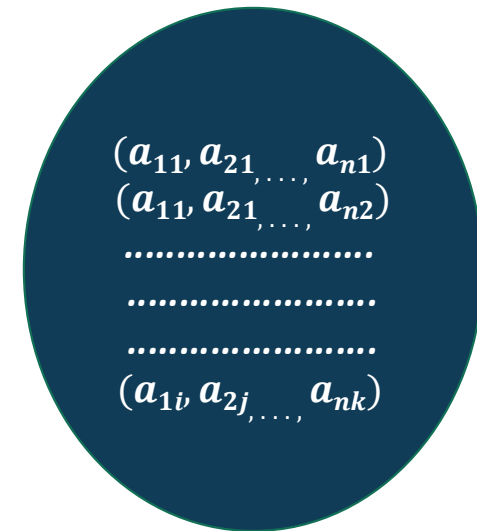
Une relation n-aire  $R$  sur  $S_n$  est définie de manière analogue comme un sous-ensemble de  $S_n$  on note  $R(x_1, \dots, x_n)$  pour indiquer que  $(x_1, \dots, x_n) \in R$  on appelle  $(x_1, \dots, x_n)$  n-uplet ou tuple.

# Relation n-aire

Soit  $R$  la relation définie sur les ensembles  $A_1, A_2, \dots, A_n$ , on a  $R \subset A_1 \times A_2 \times \dots \times A_n$



$$A_1 \times A_2 \times \dots \times A_n$$



$$\text{card}(A_1 \times A_2 \times \dots \times A_n) = i \times j \times k$$

# Représentation de R en extension (TABLE)

Colonne/attributs

Lignes/tuple

$A_1$	$A_2$	...	$A_m$	...	$A_n$
$a_{11}$	$a_{21}$		$a_{m1}$		$a_{n1}$
$a_{12}$	$a_{22}$		$a_{m2}$		$a_{n2}$
...	...		...		...
$a_{1i}$	$a_{2j}$		$a_{mp}$		$a_{nk}$

Table/relation

NumProd	Désignation	PrixUnitaire
0001	Table	120
0002	Chaise	50
0003	Armoire	90

# Tuple

## Définition (tuple)

Soit  $R$  un ensemble d'attributs, un tuple  $t$  sur  $R$  est une application

$$t: R \rightarrow \bigcup \text{Dom}(A_i) \text{ où } t(A_i) \in \text{Dom}(A_i)$$

Une relation  $r$  sur  $R$  est un ensemble de tuple.  **$R$  est dit schéma de la relation  $r$ .**



# Schéma de la relation

## Définition (Schéma de la relation)

- Description de la relation en intention

$$R(A_1, A_2, \dots, A_n)$$

$R$  : Nom de la relation

$A_1, A_2, \dots, A_n$  : Nom des **attributs** de la relation

$(a_{1i}, a_{2i}, \dots, a_{ni})$  : n-uplet

**n** : **Degré** ou **arité** de la relation (Relation n-aires)

## Exemple de schéma relationnel

**CLIENT** (codClient, nomClient, prenomClient, ville)

**PRODUIT** (codPproduit, designation, prixUnitaire, tauxTVA, stock)

**COMMANDE** (numCommande, dateCommande, commandeReglee)

**LIGNES\_COMMANDE** (numCommande, codeProduit, quantite)

**FOURNISSEUR** (codFournisseur, nomFourn, prenomFourn, ville, telephone)

**FOURNISSEUR\_PRODUIT** (codFournisseur, codProduit, quantite)

# Représentation de R en extension (TABLE)

**CLIENT**

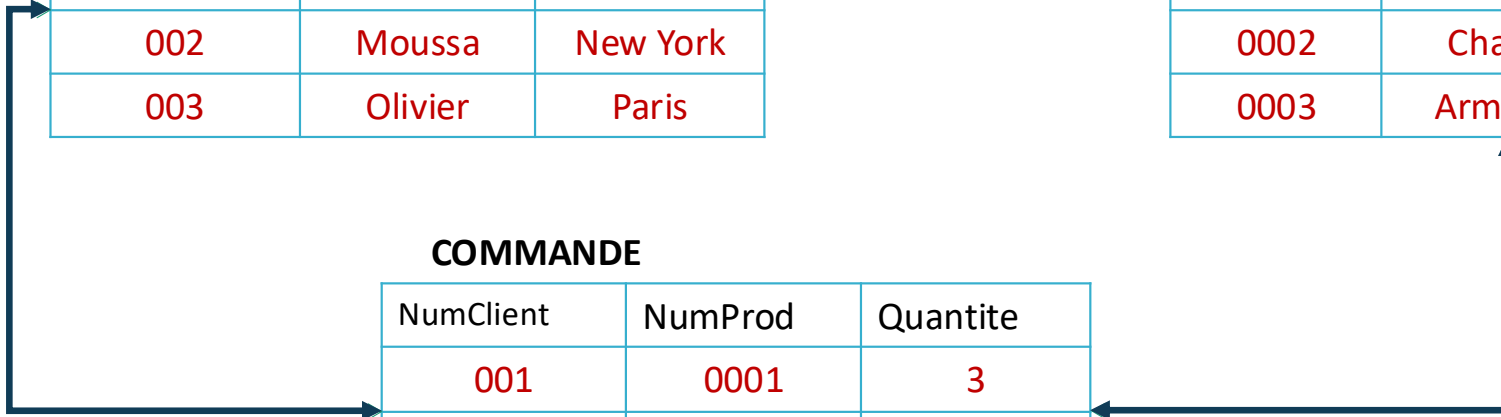
NumClient	Nom	Ville
001	Albert	Dakar
002	Moussa	New York
003	Olivier	Paris

**PRODUIT**

NumProd	Désignation	PrixUnitaire
0001	Table	120
0002	Chaise	50
0003	Armoire	90

**COMMANDE**

NumClient	NumProd	Quantite
001	0001	3
002	0002	4
003	0003	5



# Algèbre relationnelle

- Langage procedural : indique comment construire une nouvelle relation à partir d'une ou plusieurs relations existantes
- Langage abstrait, avec des opérations qui travaillent sur une (ou plusieurs) relation(s) pour définir une nouvelle relation sans changer la (ou les) relation(s) originale(s)
- Le résultat de toute opération est une relation (propriété de fermeture)

# Les operations de l'algèbre relationnelle

- Operateurs relationnels spécifiques
  - selection
  - projection
  - jointure
  - division

Soient  $R(a_1, a_2, \dots, a_N)$  et  $S(a'_1, a'_2, \dots, a'_N)$  deux relations

# Les operations de l'algèbre relationnelle

- **selection** :  $\sigma_{predicate}(R)$

La selection travaille sur R et definit une relation qui ne contient que les tuples de R qui satisfont a la condition (ou prédicat) spécifiée.

Exemple:

$\sigma_{Ville = Dakar}(CLIENT)$

**CLIENT**

NumClient	Nom	Ville
001	Albert	Dakar
002	Moussa	New York
003	Olivier	Paris

# Les operations de l'algèbre relationnelle

- **projection:**  $\pi_{a_1, \dots, a_N}(R)$

projection travaille sur R et definit une relation restreinte à un sous-ensemble des attributs de R, en extrayant les valeurs des attributs spécifiés et en supprimant les doublons.

Exemple:

$\pi_{Nom, Ville}(CLIENT)$

**CLIENT**

NumClient	Nom	Ville
001	Albert	Dakar
002	Moussa	New York
003	Olivier	Paris

# Les operations de l'algèbre relationnelle

- **jointure thêta ( $\theta$  – join):  $R \bowtie_P S$**

La thêta-jointure définit une relation qui contient les tuples qui satisfont le predicat P du produit cartésien de R et S. Le predicat P est de la forme  $R.a_i \theta S.a'_i$  ou  $\theta$  est l'un des operateurs de comparaison ( $<, \leq, >, \geq, =, \neq$ ). Si le predicat P est l'égalité ( $=$ ), on parle d'equijointure

Exemple:

$CLIENT \bowtie_P COMMANDE$

$CLIENT \bowtie_P COMMANDE$				
NumClient	Nom	Ville	NumProd	Quantite
001	Albert	Dakar	0001	3
002	Moussa	New York	0002	4
003	Olivier	Paris	0003	5

$P=(CLIENT.NumClient = COMMANDE.NumClient)$



# Les opérations de l'algèbre relationnelle

- **jointure naturelle :  $R * S$**

La jointure naturelle est une equijointure des relations  $R$  et  $S$  sur tous les attributs communs en retirant les occurrences multiples d'attributs.

# Contraintes d'intégrité : Unicité de la Clé

- Une relation est un ensemble de tuples, il ne peut y avoir 2 tuples identiques dans une relation
- La clé est le plus petit sous ensemble d'attributs dont les valeurs permettent de distinguer les n-uplets les uns des autres ;

## Définition:

Soit  $X \subseteq R$  un ensemble d'attributs, notons  $t[X]$  la restriction du tuple  $t$  à  $X$ .

$X$  est clé si et seulement si :

$$\forall (t_i, t_j) \in r^2, \forall A \in R \ t_i[X] = t_j[X] \Rightarrow t_i[A] = t_j[A]$$

## Contraintes d'intégrité : Unicité de la Clé

- ❖ Toute relation possède au moins une clé. S'il existe plusieurs clés possibles, on en choisit une qui est alors *clé primaire*, les autres sont appelées *clés candidates*

**Exemple :** Dans la relation **Personne**(num\_cni, nom, lieu\_nais, d\_naiss)  
num\_cni constitue la clé primaire

# Contraintes d'intégrité : Unicité de la Clé

## ❖ Contraintes d'entité

- Lorsque la valeur d'un attribut est inconnue, on utilise une valeur conventionnelle appelée valeur nulle
- impose que toute relation possède une clé primaire et que tout attribut participant à cette clé primaire soit non nul.

## ❖ Contraintes de domaine

- Obligation pour tout attribut de prendre des valeurs dans le domaine qu'on lui a défini. Elle se traduit sous forme d'assertion logique.

Exemple:

$\text{PrixUni} > 0 \text{ ET } \text{PrixUni} < 1000$

## Contraintes d'intégrité : Contraintes référentielle

- Caractérisée par la présence de la clé d'une relation dans une autre relation
- Définit la notion de clé étrangère (attribut qui est clé primaire dans une autre relation)

**CLIENTS** (codeClient, nomClient, prenomClient, ville)  
**PRODUITS** (codeProduit, designation, prixUnitaire, tauxTVA, stock)  
**COMMANDES** (numCommande, dateCommande, codeClient)  
**LIGNES\_COMMANDES** (numCommande, codeProduit, quantite)  
**FOURNISSEURS** (codeFournisseur, nomFourn, prenomFourn, ville, telephone)  
**FOURNISSEURS\_PRODUIT** (codFournisseur, codeProduit, quantite)

La relation **Lignes\_commande** fait référence aux relations **Commandes** et **Produits** respectivement par les clés **num\_commande** et **cod\_produit**. Elle est *référençante* tandis que les deux autres sont dites *référéncées*

# 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

# 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

*$\forall$  ACHAT, la valeur de PRIX dépend de PRODUIT*

# Dépendances fonctionnelles

- But : simplifier et éviter les redondances inutiles

- **Définition:**

Soit  $X \subseteq R$  un ensemble d'attributs,  $A \in R$ . Soit  $r$  l'extension de  $R$ .  $A$  est en dépendance fonctionnelle avec  $X$  ou  $X$  détermine  $A$  si :

$$\forall (t_i, t_j) \in r^2, t_i[X] = t_j[X] \Rightarrow t_i[A] = t_j[A]$$

On note  $r \models X \rightarrow A$

- La connaissance de la valeur de  $X$  dans  $R$  entraîne la connaissance d'au plus une valeur de  $A$ .
- $X$  est appelé **source** et  $A$  **cible**



# 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

PRODUIT **détermine** (fonctionnellement) PRIX ;

PRIX **dépend** (fonctionnellement) de PRODUIT ;

PRODUIT est le **déterminant** et PRIX est le **déterminé** de la dépendance fonctionnelle

# Dépendances fonctionnelles

**EXO:** Dites si ces dépendances sont vraies ou fausses?

A	B	C	D	E
a1	b1	c1	d3	e2
a1	b1	c3	d4	e3
a2	b2	c4	d2	e1
a3	b1	c1	d3	e2
a2	b2	c4	d2	e1

A → B  
A → C  
BC → D  
AC → D  
B → D  
D → E  
AC → E

Déterminant et déterminé peuvent être multicomposants

# Dépendances fonctionnelles

**EXO:** Dites si ces dépendances sont vraies ou fausses?

A	B	C	D	E
a1	b1	c1	d3	e2
a1	b1	c3	d4	e3
a2	b2	c4	d2	e1
a3	b1	c1	d3	e2
a2	b2	c4	d2	e1

A → B    oui  
A → C    non  
BC → D    oui  
AC → D    oui  
B → D    non  
D → E    oui  
AC → E    oui

Déterminant et déterminé peuvent être multicomposants

# Dépendances fonctionnelles

On note  $\text{dep}(r)$  l'ensemble des dépendances fonctionnelles de  $r$ .

$$\text{dep}(r) = \{X \rightarrow A \mid X \cup \{A\} \subseteq R, r \models X \rightarrow A\}$$

**Attention :** une dépendance fonctionnelle est valable sur toutes les valeurs possibles de n-uplets et pas seulement sur celles présentes à un moment donné.

# Dépendances fonctionnelles

- **Exemple**

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

NCOM → NCLI                      *toute commande est émise par un client*

NCLI → NOM                      *tout client a un nom*

NCLI → ADRESSE                      *tout client a une adresse*

NCOM → DATE                      *toute commande est passée à une date*

NCOM, NPRO → QTE                      *dans toute commande, il y a une quantité par produit*

NPRO → PRIX-U                      *tout produit a un (et un seul) prix unitaire*

# Dépendances fonctionnelles

- **Définition (Dépendance minimale)**

Une DF  $X \rightarrow A$  est minimale si  $A$  ne dépend d'aucun sous ensemble propre de  $X$ , i.e si  $Y \rightarrow A$  n'est pas valide dans  $R$  pour tout  $Y \subset X$ .

$X \rightarrow A$  est minimale  $\Leftrightarrow \forall Y \subset X, Y \rightarrow A \notin \text{dep}(r)$ .

Si  $X$  est un agrégat d'attributs, par exemple  $X = (X_1, X_2, X_3)$  et si  $X \rightarrow Z$  et  $X_1 \rightarrow Z$ , la dépendance fonctionnelle  $X \rightarrow Z$  n'est pas minimale.

- **Définition (Dépendance triviale)**

La DF  $X \rightarrow A$  est triviale si  $A \in X$ .

# Dépendances fonctionnelles

- **Définition** (Dépendance transitive)

Soit  $X$  un ensemble d'attributs,

$Y$  un ensemble d'attributs avec  $Y \not\subseteq X$ , et  $Y \twoheadrightarrow X$

$A$  un attribut,  $A \notin Y$

La dépendance fonctionnelle

$X \rightarrow A$  est directe

S'il n'existe pas  $Y$  tel que

$$X \rightarrow Y$$

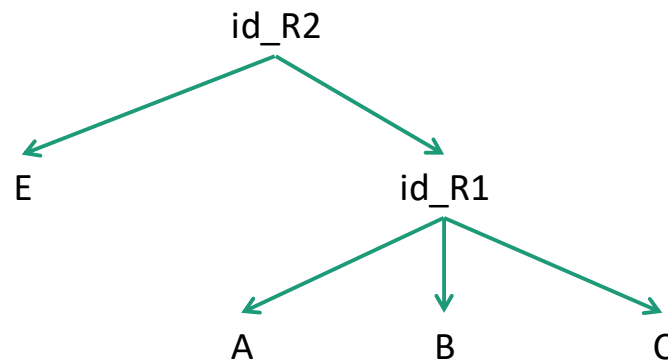
$$Y \rightarrow A$$

Si  $Y$  existe alors la dépendance est transitive

# Le graphe de dépendances fonctionnelles (GDF)

- Moyen de visualiser les DFs
- Les sommets correspondent aux attributs
- Les arcs correspondent aux DFEs (DFs élémentaires ou minimales)

R1(id\_R1, A, B, C)  
R2(id\_R2, E, id\_R1)



**Graphe de dépendances fonctionnelles (GDF) ou Graphe attributs et des DF (ADF )**



# Le graphe de dépendances fonctionnelles (GDF)

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

NCOM  $\longrightarrow$  NCLI *toute commande est émise par un client*

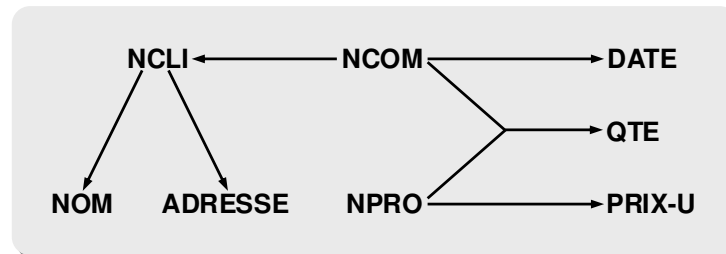
NCLI  $\longrightarrow$  NOM *tout client a un nom*

NCLI  $\longrightarrow$  ADRESSE *tout client a une adresse*

NCOM  $\longrightarrow$  DATE *toute commande est passée à une date*

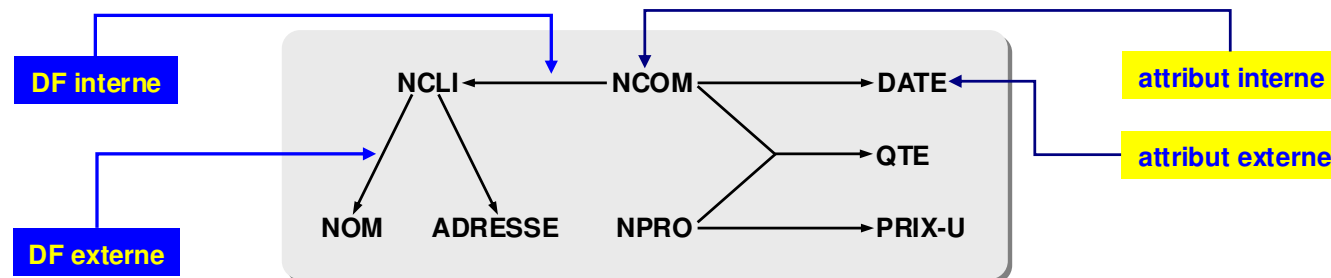
NCOM, NPRO  $\longrightarrow$  QTE *dans toute commande, il y a une quantité par produit*

NPRO  $\longrightarrow$  PRIX-U *tout produit a un (et un seul) prix unitaire*



# Le graphe de dépendances fonctionnelles (GDF)

**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



# Dépendances fonctionnelles

**DF minimale** dont le déterminant est minimal

**DF élémentaire** dont le déterminé ne comporte qu'un seul attribut

**DF de base** pas dérivable, doit être donnée

**DF dérivée** est calculable à partir des DF de base à l'aide des règles d'inférence

# Dépendances fonctionnelles

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

NCOL  $\longrightarrow$  NCLI

NCOL  $\longrightarrow$  DATE

NCLI  $\longrightarrow$  NOM

NCLI, DATE  $\longrightarrow$  ADRESSE

df1

df2

df3

df4

DF de base

*On a aussi, par transitivité :*

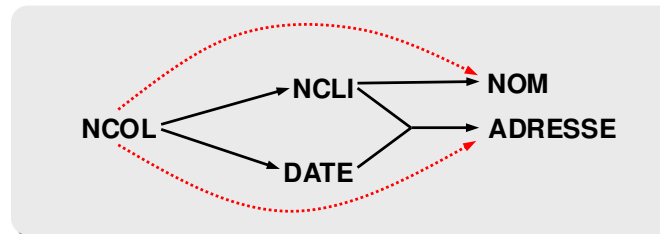
NCOL  $\longrightarrow$  NOM

NCOL  $\longrightarrow$  ADRESSE

df5

df6

DF dérivées



# Propriétés des DF (Axiomes d'Armstrong)

- **Réflexivité** : Si  $Y \subseteq X$  alors  $X \rightarrow Y$
- **Augmentation** : Si  $X \rightarrow Y$  alors  $X, Z \rightarrow Y$
- **Transitivité** : Si  $X \rightarrow Y$  et  $Y \rightarrow Z$  alors  $X \rightarrow Z$
- **Union** : Si  $X \rightarrow Y$  et  $X \rightarrow Z$  alors  $X \rightarrow Y, Z$
- **Pseudo transitivité** : Si  $X \rightarrow Y$  et  $A, Y \rightarrow Z$  alors  $X, A \rightarrow Z$
- **Décomposition** : Si  $X \rightarrow Y$  et  $Z \subseteq Y$  alors  $X \rightarrow Z$

# Fermeture transitive d'un ensemble de DF : $(\text{dep}(r))^+$

$(\text{dep}(r))^+ = \text{dep}(r) \cup \{\text{df élémentaires obtenues en appliquant les axiomes}\}$

$\forall f \in (\text{dep}(r))^+, \text{dep}(r) \models f$  (les dépendance de  $r$  satisfont  $f$ )

**Exemple :**

$$\begin{aligned} \text{dep}(r) &= \{A \rightarrow B, B \rightarrow C, B \rightarrow D, A \rightarrow E\} \\ (\text{dep}(r))^+ &= \text{dep}(r) \cup \{A \rightarrow C, A \rightarrow D\} \end{aligned}$$

Deux ensembles de DF,  $F1$  et  $F2$ , sont équivalents si leurs fermetures sont égales :  $F1 \equiv F2 \Leftrightarrow F1^+ = F2^+$

**Exemple :**

$$\begin{aligned} \{A \rightarrow B, B \rightarrow C, B \rightarrow D\} &\equiv \{A \rightarrow B, A \rightarrow D, B \rightarrow CD\} \\ \{A \rightarrow B, B \rightarrow C\} &\equiv \{A \rightarrow B, B \rightarrow C, A \rightarrow C\} \end{aligned}$$

## Retour sur la notion de clé

Une clé (ou superclé) de  $R(\Delta)$  d'ensemble d'attributs  $\Delta$  et de dépendances  $dep(r)$  est un groupe d'attributs (non nécessairement minimum)  $X$  tel que :  $X \rightarrow \Delta \in dep(r)^+$ .

Une clé **minimale** (ou clé **candidate**)  $X$  est une clé sur  $\langle R(\Delta), dep(r) \rangle$  telle que:

$$X \rightarrow \Delta \in dep(r)^+ \text{ et } \nexists Y \subset X \text{ tel que } Y \rightarrow \Delta \in dep(r)^+$$

- Une relation peut porter plusieurs clés
- Une clé primaire  $X$  de  $\langle R(\Delta), dep(r) \rangle$  est une clé candidate privilégiée pour la relation  $R$ .

# Couverture minimale

- **Objectif** : simplifier, supprimer les redondances dans les DF

On appelle couverture minimale de  $F$ , notée **min(F)**, un ensemble  $G$  obtenu en supprimant les dépendances fonctionnelles redondantes, c'est à dire celles qui peuvent être déduites à partir de  $\text{min}(F)$  tel que  $F^+ = G^+$  et qui vérifie les propriétés suivantes :

- Aucune partie gauche ne contient d'élément redondant :

$$\forall X \rightarrow A \in G \text{ et } Z \subset X, (G - \{X \rightarrow A\} \cup \{Z \rightarrow A\})^+ \neq G^+$$

- Il n'y a pas d'élément superflue :

$$\forall X \rightarrow A \in G, (G - \{X \rightarrow A\})^+ \neq G^+$$

$$(\text{min}(F))^+ = F^+$$



# Algorithme de détermination d'une couverture minimale

## Etape 1 (Décomposition):

$$F' = \{X \rightarrow A_i \mid (A_i \in Y)(X \rightarrow Y \in F)\}$$

## Etape 2 (Irréductibilité):

Pour tout  $f_j \in F'$  de la forme  $A_1, A_2, \dots, A_m \rightarrow B_j$ , si  $F' \models \{A_1, \dots, A_{i-1}, A_{i+1}, \dots, A_m\} \rightarrow B_j$  est vrai alors supprimer  $A_i$  dans  $f_j$ ,  $i$  allant de 1 à  $m$ .

## Etape 3 (Non redondance):

Pour tout  $f_j \in F'$  tester si  $F' - \{f_j\} \models f_j$  est vrai, alors  $F' := F' - \{f_j\}$

## Etape 4 (Regroupement):

regrouper les DFs ayant la même partie gauche

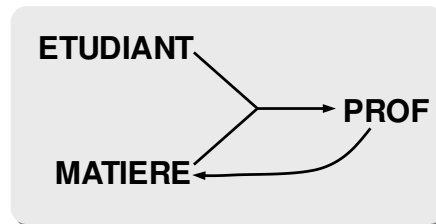
# Propriétés des couvertures

- Un ensemble  $F$  de DFs est dit **non redondant** s'il n'existe pas de couverture  $G$  de  $F$  telle que  $G \subset F$
- Un ensemble  $F$  de DFs est dit **minimal** s'il n'existe pas de couverture  $G$  de  $F$  tel que  $|G| \leq |F|$
- $F$  est dit **optimal** s'il n'existe pas de couverture  $G$  de  $F$  avec moins d'attributs que dans  $F$
- Une couverture minimale est non redondante
- Une La couverture minimale n'est pas forcément unique. Parmi celles-ci, certaines sont optimum ; malheureusement, leur calcul est un problème difficile dans le cas général (NP-Complet)

# 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 !**

# Fermeture et couvertures: résumé

toute l'information  
possible

## **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;

## **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

# 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]

# 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

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

# 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]

# 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 :**

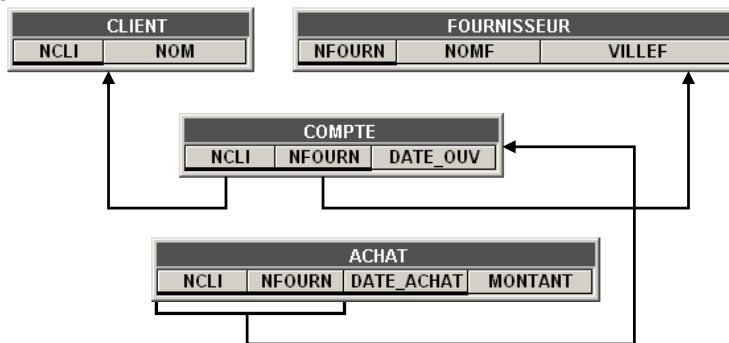
Si on a  $S[A,B] \subseteq R[A,B]$  et  $R: A \longrightarrow B$ , on a aussi  $S: A \longrightarrow B$ .

5. **Propagation dans la projection :**

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

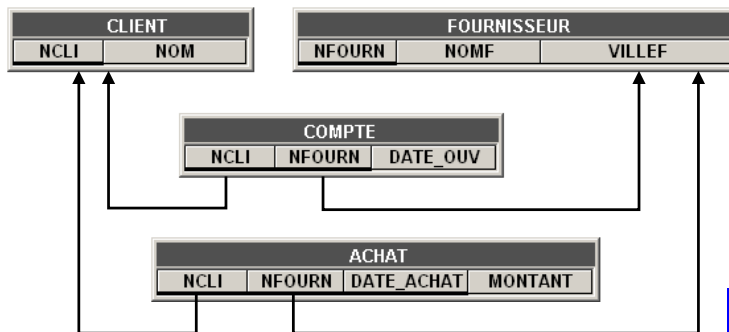


# Contraintes d'inclusion et clés étrangères: Propriétés



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

⇓ ?



$\text{COMPTE}[\text{NCLI}] \subseteq \text{CLIENT}[\text{NCLI}]$   
 $\text{COMPTE}[\text{NFOURN}] \subseteq \text{FOURNISSEUR}[\text{NFOURN}]$   
 $\text{ACHAT}[\text{NCLI}] \subseteq \text{CLIENT}[\text{NCLI}]$   
 $\text{ACHAT}[\text{NFOURN}] \subseteq \text{FOURNISSEUR}[\text{NFOURN}]$

**Correct !!**  
**Mais la dépendance**  
 **$\text{ACHAT}[\text{NCLI}, \text{NFOURN}] \subseteq \text{COMPTE}[\text{NCLI}, \text{NFOURN}]$  est perdue**

# Calcul des identifiants d'une relation

## 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  $\longrightarrow$  ADR, NPRO

ADR  $\longrightarrow$  REGION

NCOM, NPRO  $\longrightarrow$  QTE

# Calcul des identifiants d'une relation

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

# Calcul des identifiants d'une relation

## 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.

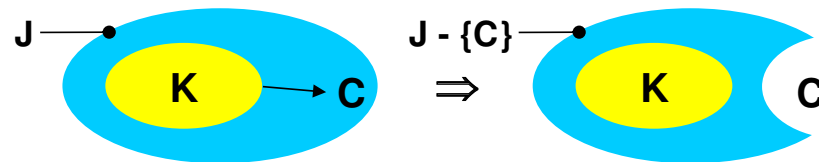
CLIENT (NCLI, NOM, ADRESSE)  $\Rightarrow$  NCLI  $\longrightarrow$  NCLI, NOM, ADRESSE

NCLI  $\longrightarrow$  NCLI, NOM, ADRESSE  $\Rightarrow$  CLIENT (NCLI, NOM, ADRESSE)

# Calcul des identifiants d'une relation

## 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  $R(\underline{A}, \underline{B}, C, D)$ .

# 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.

# Calcul des identifiants d'une relation: Identifiant unique

## Application

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

NCOM  $\longrightarrow$  NCLI

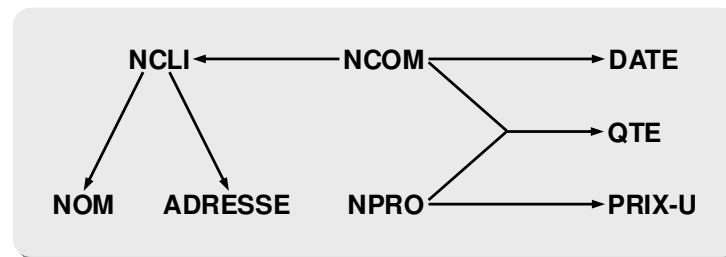
NCLI  $\longrightarrow$  NOM

NCLI  $\longrightarrow$  ADRESSE

NCOM  $\longrightarrow$  DATE

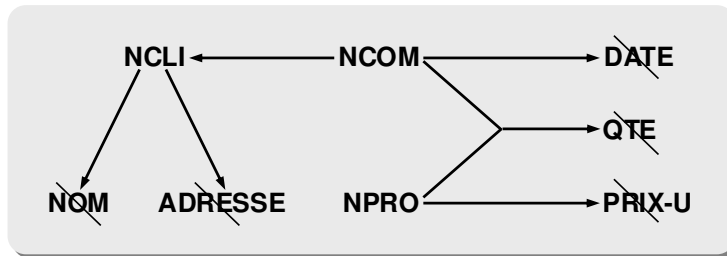
NCOM, NPRO  $\longrightarrow$  QTE

NPRO  $\longrightarrow$  PRIX-U

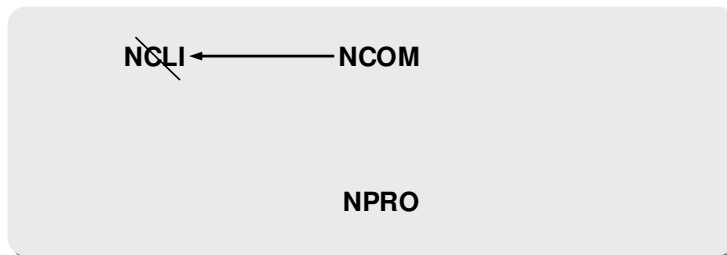


# Calcul des identifiants d'une relation: Identifiant unique

## Application



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



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



# Calcul des identifiants d'une relation: Identifiant unique

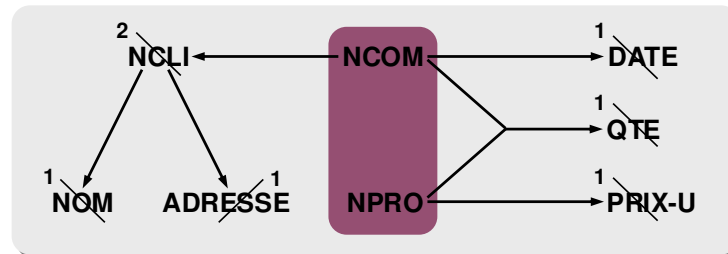
## Application

NCOM

NPRO

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

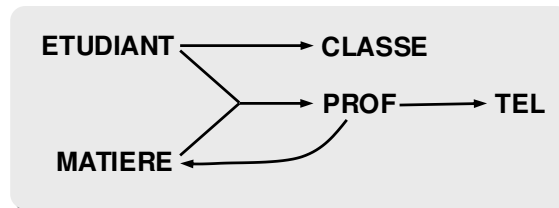
COM(NCLI, NOM, ADRESSE, NCOM, NPRO, DATE, QTE, PRIX-U)



# Calcul des identifiants d'une relation: Identifiants multiples

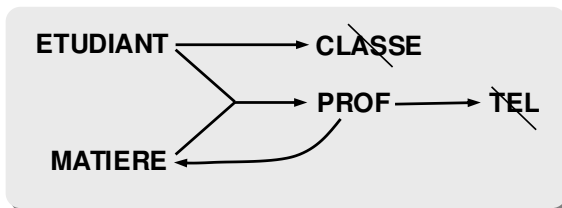
**Est-ce que ça marche si la relation possède plus d'un identifiant ?**

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

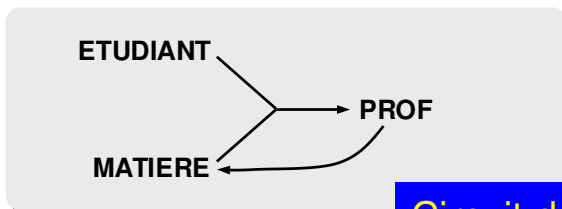


# Calcul des identifiants d'une relation: Identifiants multiples

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



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



Circuit de DF !

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

pas minimal !

... et ensuite ?

# Calcul des identifiants d'une relation: Identifiants multiples

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

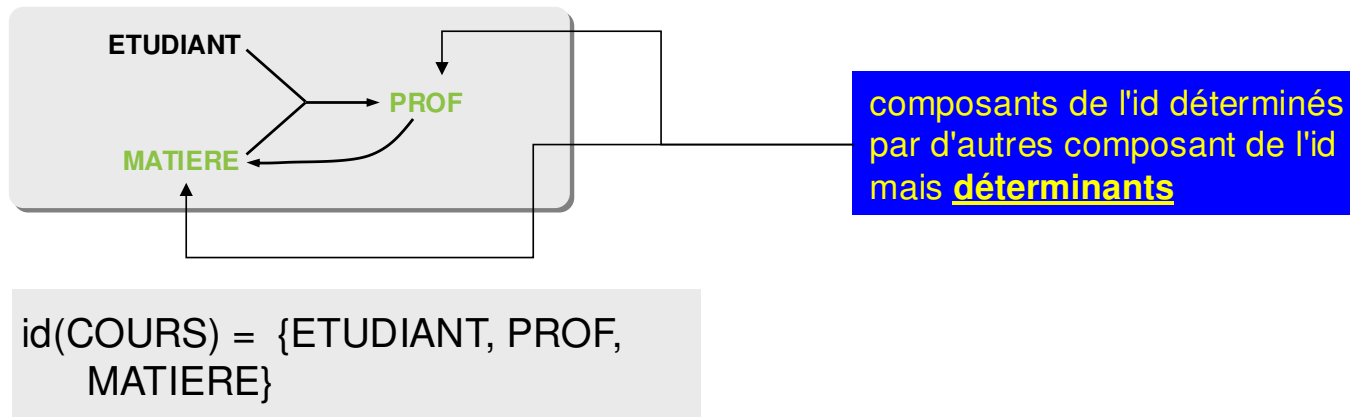
2. On recherche dans **I** un **attribut A externe**; on retire **A** de **I**.

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

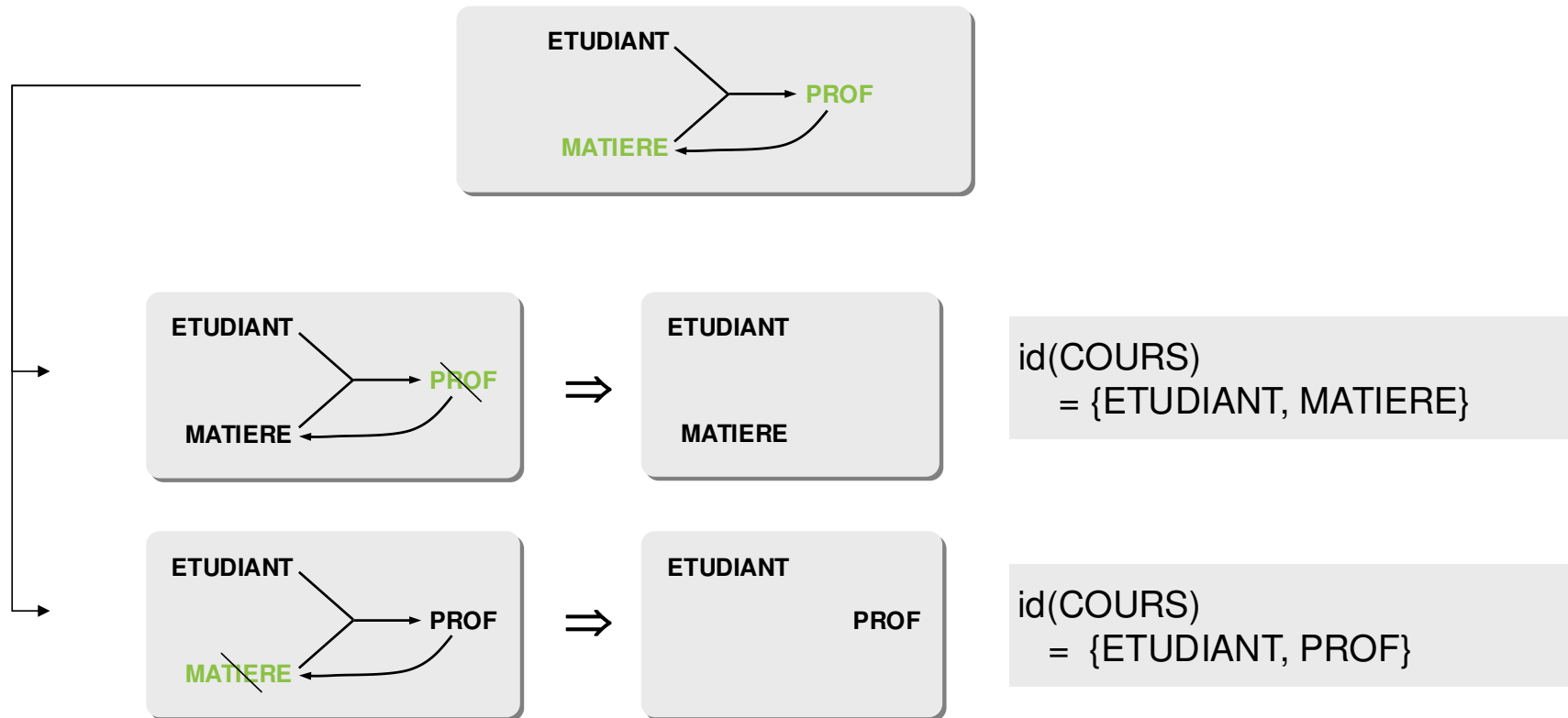
# Calcul des identifiants d'une relation: Identifiants multiples

## 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.



# Calcul des identifiants d'une relation: Identifiants multiples



# 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**

# Le problème de la décomposition

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	le sucre coûte 2,2 : <b>2 fois</b>
Marc a acheté du sucre :	1 fois	le sel coûte 1,4 : <b>2 fois</b>
Marc a acheté du sel :	1 fois	le savon coûte 1,4 : 1 fois
Anne a acheté du savon :	1 fois	le pain coûte 2,2 : <b>0 fois</b>
Anne a acheté du sel :	1 fois	

**redondance  
et lacunes !**



# Le problème de la décomposition

## 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

$\Rightarrow$

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

la décomposition  
est légale !

# Le problème de la décomposition

## Définition:

Soit  $\mathbf{R}(\mathbf{U})$  un schéma relationnel d'ensemble de dépendances fonctionnelles  $\mathbf{F}$ . L'ensemble des schémas  $\mathbf{R}_1(\mathbf{U}_1), \dots, \mathbf{R}_n(\mathbf{U}_n)$ , avec  $\mathbf{U} = \mathbf{U}_1 \cup \dots \cup \mathbf{U}_n$  d'ensemble de dépendances fonctionnelles respectifs  $\mathbf{F}_1, \dots, \mathbf{F}_n$  constitue une décomposition de  $\mathbf{F}$ . le contenu de chaque relation  $\mathbf{R}_i(\mathbf{U}_i)$  peut être caractérisé à partir de  $\mathbf{R}$  en posant  $\mathbf{R}_i(\mathbf{U}_i) = \mathbf{R}[\mathbf{U}_i]$ .

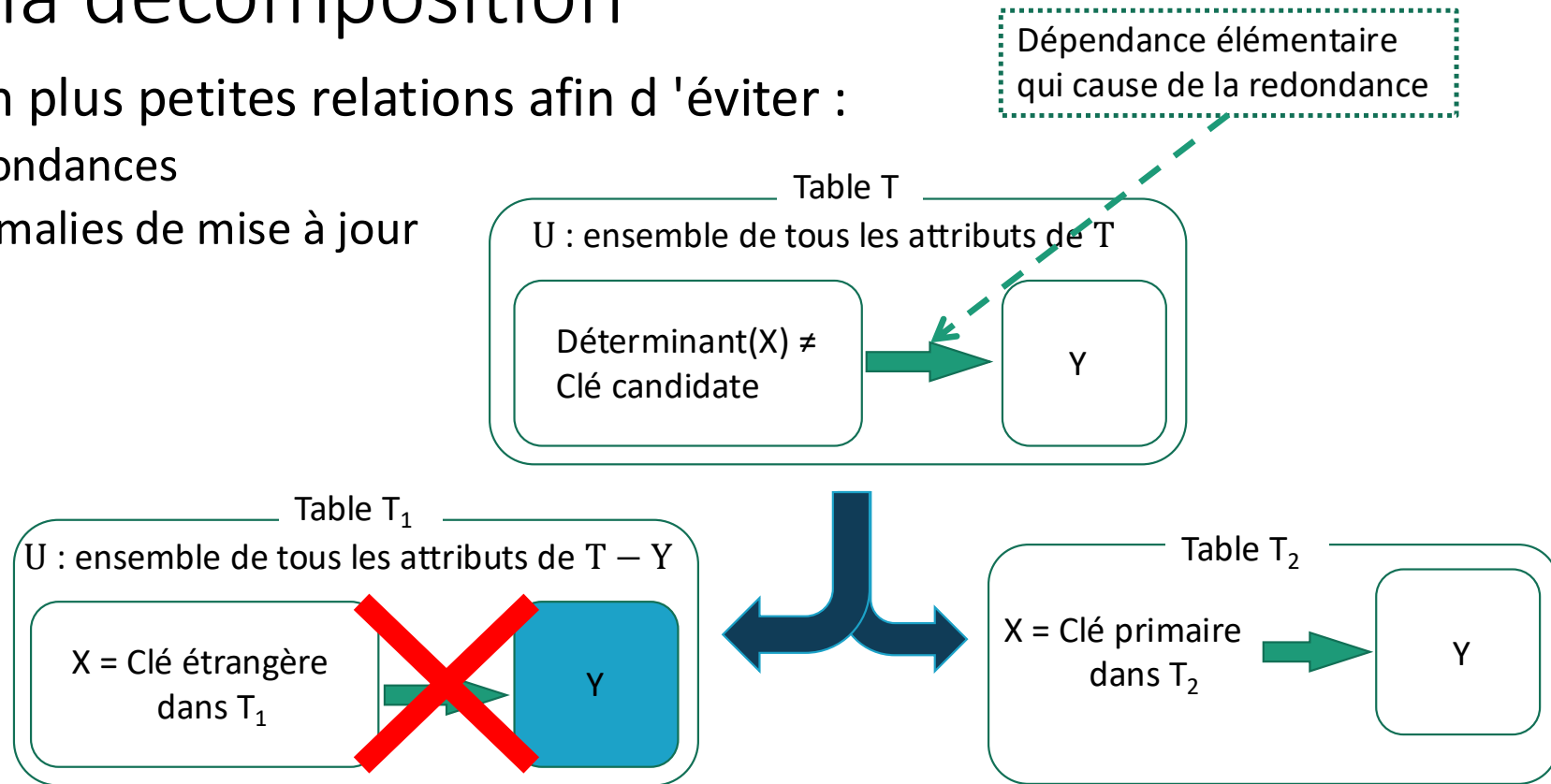
L'ensemble de dépendances fonctionnelles  $\mathbf{F}_i$  est obtenu en projetant l'ensemble  $\mathbf{F}$  sur l'ensemble d'attributs  $\mathbf{U}_i$  comme suit:

$$\mathbf{F}_i[\mathbf{U}_i] = \{X \rightarrow Y \in \mathbf{F}^+ : (X \subseteq \mathbf{U}_i)(Y \subseteq \mathbf{U}_i)\}$$

# But de la décomposition

Casser R en plus petites relations afin d'éviter :

- les redondances
- les anomalies de mise à jour



# Décomposition d'une relation: Théorème de décomposition

[Delobel, 1973]

**Une propriété très importante des DF**

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

*= théorème de décomposition*

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

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

# Décomposition d'une relation: Théorème de décomposition

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

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

$\Rightarrow$

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

*pas équivalent*

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

$\Rightarrow$

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

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

$\Leftrightarrow$

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

*équivalence stricte*

# Décomposition d'une relation: Théorème de décomposition

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

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



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

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

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

# Décomposition d'une relation: exemple

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

$\Rightarrow$

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

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

$\Rightarrow$

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

COM(NCOM, CLI, PRO, QTE)

$\Rightarrow$

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

# Décomposition d'une relation: remarque

**La décomposition ...**

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

$\Rightarrow$

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

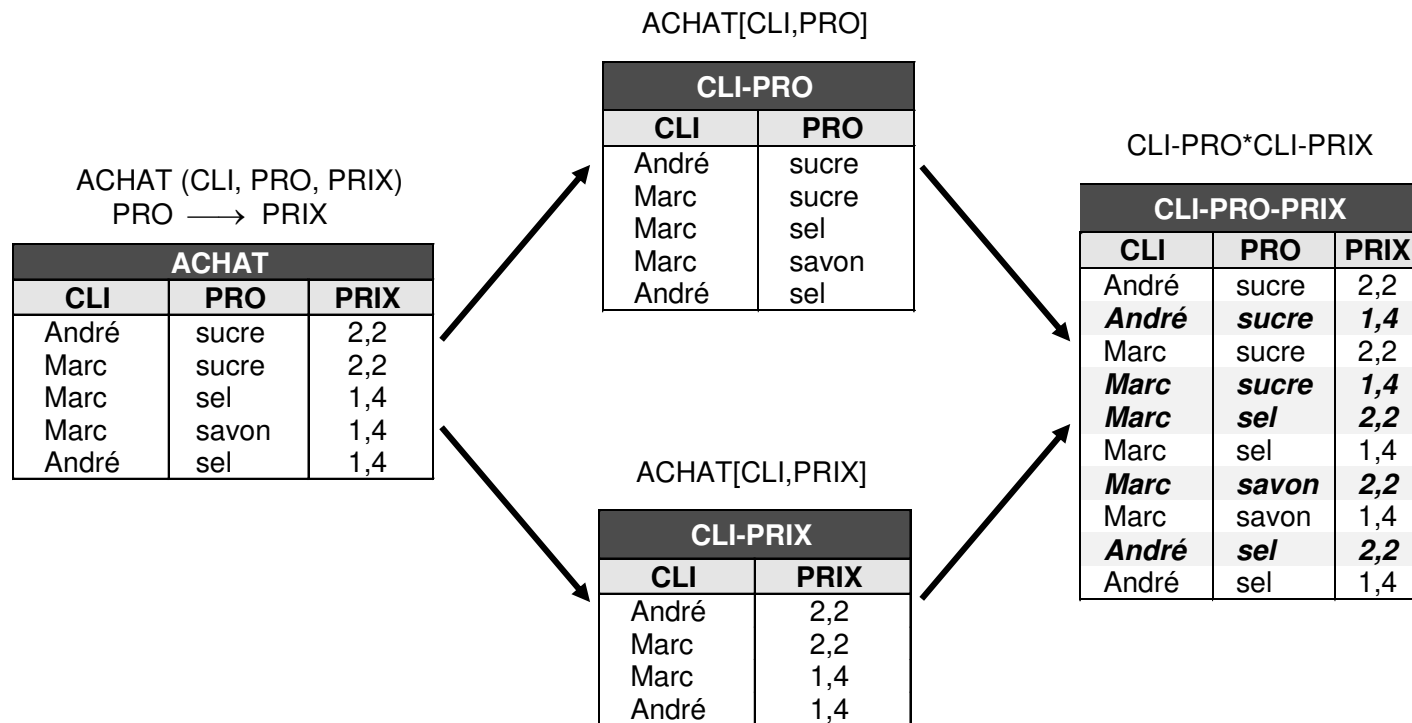
**... ne suit pas la règle mais est parfaitement valide.**

**Cependant, elle n'est pas réversible. En toute généralité :**

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



# Décomposition d'une relation: Exemples



# Propriété d'une bonne décomposition

- **Décomposition Sans perte d'informations**

Soit  $R_1(U_1), \dots, R_n(U_n)$  une décomposition d'un schéma  $R$ .

✓ Soit  $r$  une relation de schéma  $R$ . La décomposition est **sans perte pour**  $r$  ssi

$$r = r_1[U_1] \bowtie \dots \bowtie r_n[U_n]$$

✓ Soit  $F$  un ensemble de DF sur  $R$ . La décomposition est **sans perte par rapport** à  $F$  ssi pour toute relation  $r$  de schéma  $R$  qui satisfait  $F$ , la décomposition est sans perte pour  $r$ , c'est-à-dire

$$r = r_1[U_1] \bowtie \dots \bowtie r_n[U_n]$$

- **Décomposition Sans perte de dépendances**

Une décomposition  $(R_1, \dots, R_m)$  de  $R$  préserve les DFs si la fermeture transitive des DFs de  $R$  est la même que la fermeture transitive de l'union des DFs de  $R_1, \dots, R_m$ .

$$F^+ = (F[U_i])^+$$

# Propriété d'une bonne décomposition

- **Exemple :**

Soit la relation  $R = (A, B, C, D)$  et une décomposition  $\{R_1, R_2\}$  de  $R$  où  $R_1 = (A, B, C)$  et  $R_2 = (A, D)$ .

Dans cette décomposition, la dépendance fonctionnelle  $A, B \rightarrow D$  ne peut être testée pour les relations  $r_1(R_1)$  et  $r_2(R_2)$  sans faire la jointure naturelle  $r_1 \bowtie r_2$ .

# Propriété d'une bonne décomposition

**Théorème :** Soit  $R$  un schéma de relation,  $F$  un ensemble de dépendances fonctionnelles,  $R_1$  et  $R_2$  une décomposition de  $R$ . La décomposition est dite lossless-join si au moins une des dépendances suivantes est dans la fermeture transitive  $F^+$  de  $F$ .

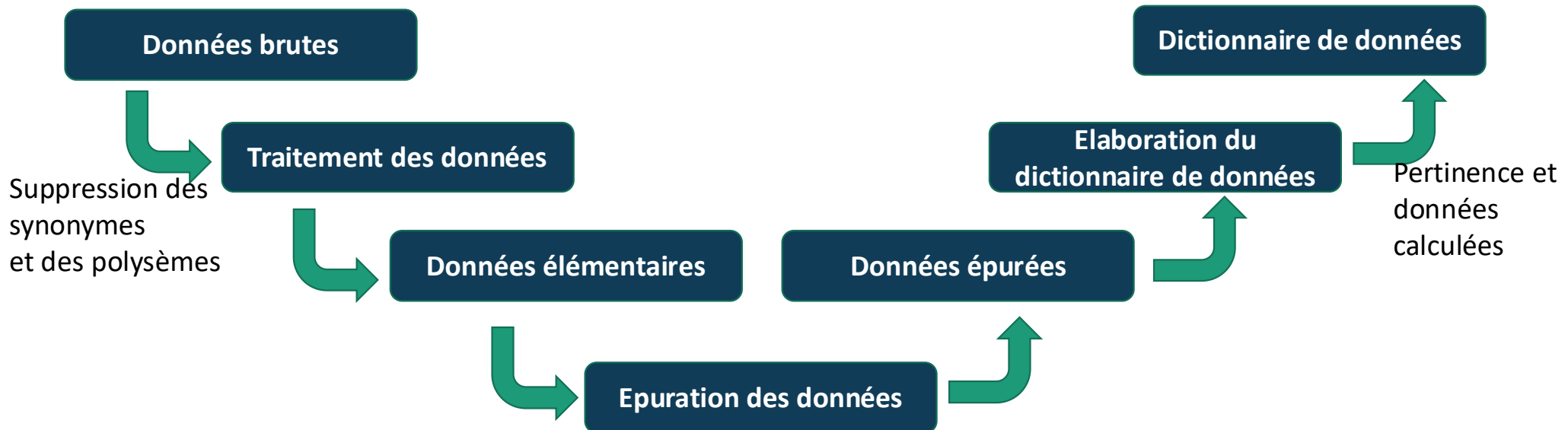
$$R_1 \cap R_2 \rightarrow R_1$$

$$R_1 \cap R_2 \rightarrow R_2$$

En effet, il faut s'assurer que les attributs impliqués dans la jointure naturelle  $R_1 \bowtie R_2$  sont des clés candidates pour au moins une des relations.

# Elaboration du dictionnaire des données

- Liste exhaustive des données nécessaires à la modélisation du SI.



## Exemple : Gestion de commandes

- Un client peut passer une ou plusieurs commandes. Il peut ne pas passer de commande
- Une commande peut concerner un ou plusieurs produits.
- Une commande est lancée par un client.

N°Commande	Date			
N°Client				
Nom				
Adresse				
Code	Désignation	PU	Quantité	Montant
.....				
.....				
.....				
.....				
.....				
.....				
<b>Montant Total:</b>				

# Traitement des données

- lister les données élémentaires (atomiques) exemple : adresse, code postal, . . .

N°	Données	Libellé complé
1.	Numéro	numéro de la commande
2.	Data	data de la commande
3.	Numéro	numéro du client
4.	Nom	nom du client
5.	Prénom	prénom du client
6.	Adresse	adresse du client
7.	Code	code produit
8.	Prix	prix du produit
9.	Designation	libellé du produit
10.	Quantité	quantité commandée du produit
11.	Montant	montant de la commande

# Traitement des données

- Supprimer les synonymes (données de nom différent avec la même signification) et les polysèmes (données de même nom avec une signification différente).

N°	Données	Libellé complé
1.	<b>NumCmd</b>	numéro de la commande
2.	Data	data de la commande
3.	<b>NumClient</b>	numéro du client
4.	Nom	nom du client
5.	Prénom	prénom du client
6.	Adresse	adresse du client
7.	Code	code produit
8.	Prix	prix du produit
9.	Designation	libellé du produit
10.	Quantité	quantité commandée du produit
11.	Montant	montant de la commande



# Elaboration du dictionnaire des données

- Elimination des données non pertinentes
- Mise à part des paramètres (ex : taux euro/franc)
- Elimination des données calculées déterminables à tout moment (ex : age, montant) (exception : calculs lourds)

N°	Données	Libellé complé
1.	<b>NumCmd</b>	numéro de la commande
2.	Data	data de la commande
3.	<b>NumClient</b>	numéro du client
4.	Nom	nom du client
5.	Prénom	prénom du client
6.	Adresse	adresse du client
7.	Code	code produit
8.	Prix	prix du produit
9.	Designation	libellé du produit
10.	Quantité	quantité commandée du produit
11.	Montant	montant de la commande

# Dictionnaire des données

- Numéro
- Signification
- Domaine de définition
- Type : normales, calculées, paramètre
- Précisions sur la nature et la longueur des données

N°	Données	Libellé complé
1.	<b>NumCmd</b>	numéro de la commande
2.	Data	data de la commande
3.	<b>NumClient</b>	numéro du client
4.	Nom	nom du client
5.	Prénom	prénom du client
6.	Adresse	adresse du client
7.	Code	code produit
8.	Prix	prix du produit
9.	Designation	libellé du produit
10.	Quantité	quantité commandée du produit
11.	Montant	montant de la commande

# Matrice de dépendances fonctionnelles

- admet une ligne et une colonne par propriétés du dictionnaire
- "1" placé à l'intersection de la ligne  $i$  et de la colonne  $j$  signifie  $(P_j \rightarrow P_i)$ .
- Remarque : une source identifie de manière certaine un objet.

[illegible]

# Matrice de dépendances fonctionnelles

- les **propriétés identifiantes** : des colonnes qui comportent au moins deux “1”
- les **propriétés** qui ne sont **destinations d'aucune DF** (ex. Quantité)
- Parmi ces propriétés il convient alors de distinguer :
  - les **propriétés paramètres** telles que le taux de TVA
  - les **autres propriétés** pour lesquelles on doit rechercher les dépendances fonctionnelles ayant des sources multi-attributs qui permettent de les atteindre.
  - source de ces dépendances : sous-ensemble des rubriques identifiantes repérées à l'étape précédente
  - codeProduit, NoCommande → Quantité
- pour indiquer une dépendance entre 2 sources sans nécessairement de donnée cible, on ajoute la dépendances  $\text{Source1} + \text{Source2} \rightarrow \emptyset$  ;

# Matrice de dépendances fonctionnelles

[illegible]