# Chapitre III

# La théorie des dépendances et la normalisation des relations

# Le problème

**Exemple** : Considérons le schéma de relation

*FOURNISSEURS(NOM_F, ADRESSE_F, COMB, PRIX).*

Il y a une série de problèmes liés à son utilisation.

1. **Redondance** : L'adresse de chaque fournisseur se trouve répétée.
2. **Possibilité d'inconsistance** suite à des mises à jour *(update anomalies)* : Un fournisseur pourrait avoir deux adresses différentes.
3. **Anomalie d'insertion** : Insérer l'adresse d'un fournisseur qui ne fournit rien.
4. **Anomalie de suppression** : Si un fournisseur ne fournit plus rien, on perd son adresse.

**Solution ?**

Utiliser le schéma :

*FA(NOM_F, ADRESSE_F)*

*FCP(NOM_F, COMB, PRIX)*

## Les Questions à résoudre

- Quand y-a-t-il une redondance gênante ?
- S'il y a redondance :
  - Comment faut-il décomposer la relation ?
  - Y-a-t-il de l'information perdue par la décomposition ?
  - Existe-t-il des algorithmes qui permettent de déterminer la décomposition adéquate ?

# Les dépendances fonctionnelles

D'où vient la redondance ? Du fait que l'information a une structure particulière.

**Exemple** :
*FOURNISSEURS(NOM_F, ADRESSE_F, COMB, PRIX)*

La structure de l'information est telle que si l'on connaît *NOM_F*, alors on connaît *ADRESSE_F* !

Un fournisseur n'a qu'une seule adresse. Autrement dit, on ne peut avoir deux tuples qui ont la même valeur de *NOM_F* et des valeurs différentes de *ADRESSE_F*.

Le problème est de choisir les schémas de relation en accord avec cette structure. Pour ce faire, on a besoin d'une représentation de la structure de l'information.

On va au départ étudier une notion qui permet de caractériser cette structure : la notion de *dépendance fonctionnelle.*

## Les dépendances fonctionnelles : définition

Soit un schéma de relation $R(A_1, \ldots, A_n)$ et soit $X, Y \subseteq \{A_1, \ldots, A_n\}$.

- Les ensembles $X$ et $Y$ définissent une *dépendance fonctionnelle*, notée $X \rightarrow Y$ (lue "$X$ *détermine* $Y$").
- Une relation $r$ de schéma $R$ *satisfait une dépendance fonctionnelle* $X \rightarrow Y$ si pour tous tuples $t_1, t_2 \in r$ :
  si $t_1[X] = t_2[X]$ alors $t_1[Y] = t_2[Y]$.

**Note** : On associe une *dépendance* au schéma d'une relation. Elle indique une *caractéristique des données* que l'on veut modéliser (tout comme le schéma).

On suppose dès lors que toute relation se trouvant dans la base de données satisfait les dépendances. Le système pourrait refuser toute modification qui violerait une dépendance.

# Les dépendances fonctionnelles : notations et exemples

- Ensembles d'attributs : $X, Y$
- Ensembles explicites d'attributs : $A_1A_2A_3$ $(= \{A_1, A_2, A_3\})$
- Union d'ensembles d'attributs : $XY$ $(= X \cup Y)$

**Exemples** :

- *CLIENTS(NOM, ADRESSE, SOLDE_DU)*

  *NOM* $\rightarrow$ *ADRESSE SOLDE_DU* (1)

- *COMMANDES(NO, NOM, COMB, QUANT)*

  *NO* $\rightarrow$ *NOM COMB QUANT* (2)

- *FOURNISSEURS(NOM_F,ADRESSE_F, COMB, PRIX)*

  *NOM_F* $\rightarrow$ *ADRESSE_F* (3)

  *NOM_F COMB* $\rightarrow$ *PRIX* (4)

- **Autres dépendances (triviales)** :

  *NOM* $\rightarrow$ *NOM* (5)

  *NOM COMB* $\rightarrow$ *COMB* (5')

- **Autres dépendances (non-triviales)** :

  *NOM_F COMB* $\rightarrow$ *ADRESSE_F PRIX* (6)

# L'implication de dépendances

**Dans l'exemple** :

- Les dépendances (5), (5') sont *triviales* : elles sont toujours vraies quelle que soit la relation.
- La dépendance (6) est *impliquée* par (3) et (4) : intuitivement, toute relation qui satisfait (3) et (4), satisfait aussi (6).

**Définitions** :
Soit $F$ et $G$ deux ensembles de dépendances.

- Une dépendance $X \rightarrow Y$ est *impliquée logiquement* par $F$ (noté $F \models X \rightarrow Y$) si toute relation qui satisfait $F$ satisfait aussi $X \rightarrow Y$.
- $F$ et $G$ sont *(logiquement) équivalents* (noté $F \equiv G$) si toute dépendance de $G$ est impliquée logiquement par $F$ et vice-versa.
- La *fermeture de* $F$ est l'ensemble $F^+$ des dépendances logiquement impliquées par $F$, c-à-d $F^+ = \{X \rightarrow Y \mid F \models X \rightarrow Y\}$.

Pour exploiter utilement les dépendances fonctionnelles, il faut pouvoir résoudre le problème de l'implication de dépendances

**Exemple** : $R = ABC$ et $F = \{A \to B,\ B \to C\}$

On a par exemple :

- $F \models A \to C$
- $\{A \to B,\ B \to C\} \equiv \{A \to B,\ B \to C,\ A \to C\}$
- $F^+ = \{X \to Y \mid$ 1. $A \in X$, ou
  2. $A \notin X,\ B \in X$ et $A \notin Y$, ou
  3. $C \to C$ ou $C \to \emptyset\}$

  Exemples de dépendances dans $F^+$ :

  $ABC \to AB,\ AB \to ABC,\ A \to C$
  $BC \to C,\ BC \to B,\ B \to \emptyset$

**Note** : Si l'on se base sur les dépendances fonctionnelles pour concevoir des schémas de bases de données, on s'attend à ce que des ensembles de dépendances équivalents donnent les mêmes schémas.

## Propriétés des dépendances fonctionnelles

Les dépendances fonctionnelles satisfont un certain nombre de règles qui permettent de raisonner à leur sujet.

- Si $Y \subseteq X$, alors $X \rightarrow Y$ (réflexivité)
- Si $X \rightarrow Y$ et $Z \subseteq U$, alors $XZ \rightarrow YZ$ (augmentation)
- Si $X \rightarrow Y$ et $Y \rightarrow Z$, alors $X \rightarrow Z$ (transitivité)
- Si $X \rightarrow Y$ et $X \rightarrow Z$, alors $X \rightarrow YZ$ (union)
- Si $X \rightarrow Y$, alors $X \rightarrow Z$ si $Z \subseteq Y$ (décomposition)

La première règle donne des dépendances qui sont toujours vraies ; elles sont dites triviales.

La conséquence des deux dernières règles est qu'une dépendance $X \rightarrow A_1 A_2 \ldots A_n$ est équivalente à l'ensemble de dépendances $X \rightarrow A_1$, $X \rightarrow A_2$, ..., $X \rightarrow A_n$.

# La fermeture d'un ensemble d'attributs

Soit $F$ un ensemble de dépendances et $X \subseteq U$ un ensemble d'attributs pris parmi un ensemble d'attributs $U$.

**Définition**

La fermeture $X^+$ à partir de $X$ par rapport à l'ensemble de dépendances $F$ est l'ensemble des $A \in U$ tels que $F \models X \rightarrow A$.

Si on peut calculer la fermeture d'un ensemble d'attributs, il est simple de déterminer si une dépendance $X \rightarrow Y$ est impliquée logiquement par un ensemble de dépendances $F$.

1. Calculer $X^+$ par rapport à $F$.
2. Si $Y \subseteq X^+$ alors $F \models X \rightarrow Y$, sinon $F \not\models X \rightarrow Y$.

Cela permet aussi de calculer $F^+$ (en principe) :
$F^+ = \{X \rightarrow Y \mid F \models X \rightarrow Y\} = \{X \rightarrow Y \mid X \subseteq U \text{ et } Y \subseteq X^+\}$.

Il faut noter que le nombre de dépendances dans $F^+$ peut être très grand (exponentiel en fonction du nombre d'attributs).

## Un algorithme de calcul de la fermeture

Soit $F$ un ensemble de dépendances et $X \subseteq U$ un ensemble d'attributs.

L'algorithme calcule une suite d'ensembles d'attributs $X^{(0)}, X^{(1)}, \ldots$

**Algorithme**

Données : $X$, $U$, $F$.

1. $X^{(0)} = X$.
2. $X^{(i+1)} = X^{(i)} \cup \{A \mid \exists Z \ : Y \rightarrow Z \in F$ et $A \in Z$ et $Y \subseteq X^{(i)}\}$.
3. Si $X^{(i+1)} = X^{(i)}$, l'algorithme s'arrête.

**Note** : L'algorithme s'arrête toujours puisque $X^{(0)} \subseteq X^{(1)} \subseteq X^{(2)} \ldots \subseteq U$ et $U$ est un ensemble fini.

**Exemple** :

Soit $F$ :

$$\begin{array}{rcl} AB & \rightarrow & C \\ C & \rightarrow & A \\ BC & \rightarrow & D \\ ACD & \rightarrow & B \end{array} \qquad \begin{array}{rcl} D & \rightarrow & EG \\ BE & \rightarrow & C \\ CG & \rightarrow & BD \\ CE & \rightarrow & AG \end{array}$$

$X : BD$ et $U : ABCDEG$

L'algorithme calcule : $X^{(0)} : BD$

$X^{(1)} : BDEG$

$X^{(2)} : BDEGC$

$X^{(3)} : BDEGCA$

$X^{(4)} : BDEGCA$

**Question** : L'algorithme calcule-t-il bien $X^+$ ?

**Théorème** : L'algorithme donné calcule bien $X^+$, c'est-à-dire que, quand l'algorithme s'arrête, $X^{(i_f)} = \{A \mid F \models X \to A\}$.

**Démonstration** :

1. Montrons $X^{(i_f)} \subseteq X^+ = \{A \mid F \models X \to A\}$.
   On démontre par induction que $\forall i$ : $X^{(i)} \subseteq X^+$.

   (a) Initialement $X^{(0)} = X \subseteq X^+$.

   (b) Si $X^{(i)} \subseteq X^+$, alors $X^{(i+1)} \subseteq X^+$.
   Par la définition de $X^{(i+1)}$, il est aisé de voir que
   si on a $F \models X \to A$ pour tout $A \in X^{(i)}$,
   alors on a $F \models X \to A$ pour tout $A \in X^{(i+1)}$.

2. Montrons $X^+ \subseteq X^{(i_f)}$, c-à-d $X^+ - X^{(i_f)} = \emptyset$

   Pour ce faire, nous montrons que, pour toute dépendance $X \rightarrow B$ telle que $B \notin X^{(i_f)}$, nous avons que $F \not\models X \rightarrow B$ et donc que $B \notin X^+$.

   Pour établir que $F \not\models X \rightarrow B$, il suffit de montrer qu'il existe une relation qui satisfait toutes les dépendances de $F$, mais qui ne satisfait pas $X \rightarrow B$.

   Cette relation est la suivante :

$r$ :

| $X^{(i_f)}$ | | $U - X^{(i_f)}$ | |
|---|---|---|---|
| 1 | $1 \cdots 1$ | 1 | $1 \cdots 1$ |
| 1 | $1 \cdots 1$ | 0 | $0 \cdots 0$ |

   - Cette relation ne satisfait pas $X \rightarrow B$ vu que $X \subseteq X^{(i_f)}$ et que $B \notin X^{(i_f)}$.
   - Nous allons montrer qu'elle satisfait $F$.
     En effet, si $r$ ne satisfaisait pas $F$, il y aurait dans $F$ une dépendance $V \rightarrow W$ telle que $V \subseteq X^{(i_f)}$ et $W \not\subseteq X^{(i_f)}$ (par déf. de $R$).
     Mais dans ce cas, l'algorithme de calcul de $X^+$ ne se serait pas arrêté et aurait ajouté les attributs de $W$.

## Couverture d'un ensemble de dépendances

Par définition, deux ensembles de dépendances $F$ et $G$ sont *équivalents* si pour tout $X \to Y \in F$, on a $G \models X \to Y$ et vice-versa, ou, de façon équivalente, si $F^+ = G^+$.

On cherche une description *la plus succincte possible* d'un *ensemble de dépendances à l'équivalence près*.

**Définitions** :

– $F$ est une *couverture* pour un ensemble de dépendances $G$ si $F \equiv G$.

– Une couverture $F$ de $G$ est *minimale* si :

1. $F$ contient uniquement des dépendances du type $X \to A$,
2. $\nexists (X \to A) \in F : F - \{X \to A\} \equiv F$,
3. $\nexists (X \to A) \in F$ et $\nexists Z \subset X$ :
   $(F - \{X \to A\}) \cup \{Z \to A\} \equiv F$.

**Théorème** : Tout ensemble de dépendances a une couverture minimale.

**Démonstration** :

On peut obtenir une couverture minimale en procédant comme suit.

1. Toute dépendance $X \rightarrow Y$ avec $Y = \{A_1, A_2, \ldots A_n\}$ est remplacée par les dépendances $X \rightarrow A_1$, $X \rightarrow A_2$, ..., $X \rightarrow A_n$.
2. On élimine le plus de dépendances (redondantes) possibles.
3. On élimine le plus d'attributs possibles des membres de gauche.

**Remarque** : Les couvertures minimales ne seront en général pas uniques.

**Exemple** :

Soit $F$ :

$$\begin{array}{rcl@{\qquad}rcl} AB & \to & C & D & \to & EG \\ C & \to & A & BE & \to & C \\ BC & \to & D & CG & \to & BD \\ ACD & \to & B & CE & \to & AG \end{array}$$

1. $F$ peut être transformé en :

$$\begin{array}{rcl@{\qquad}rcl} AB & \to & C & D & \to & E \\ & & & D & \to & G \\ C & \to & A & BE & \to & C \\ BC & \to & D & CG & \to & B \\ & & & CG & \to & D \\ ACD & \to & B & CE & \to & A \\ & & & CE & \to & G \end{array}$$

2. – On peut éliminer $CE \to A$
   – On peut éliminer $CG \to B$
3. On peut remplacer $ACD \to B$ par $CD \to B$

## Notion de clé

On a défini une *clé* comme un ensemble d'attributs qui déterminent de manière unique les tuples d'une relation.

La notion de clé peut aussi se définir à partir des dépendances.

**Définitions** : Soit $F$ un ensemble de dépendances.

- Un ensemble d'attributs $X$ est une *clé* d'une relation de schéma $R$ *par rapport à* $F$, si $X$ est un ensemble minimum tel que $F \models X \rightarrow R$.
- Un ensemble d'attributs $X$ est une *super-clé* (*superkey*) d'une relation de schéma $R$ *par rapport à* $F$, si $X$ est un ensemble (non nécessairement minimum) tel que $F \models X \rightarrow R$.

# Formes Normales

On a vu que la présence de dépendances pouvait mener à une duplication de l'information.

*Mais quelles dépendances sont néfastes ?*

Dans le cadre des dépendances fonctionnelles, ce qu'il faut éviter est d'avoir une dépendance non-triviale $Y \rightarrow A$ pour laquelle la partie de tuple $y_1 \dots y_k\, a$ puisse apparaître plusieurs fois.

**Exemple** :

*FOURNISSEURS(NOM_F, ADRESSE_F, COMB, PRIX)*

avec *NOM_F* $\rightarrow$ *ADRESSE_F*
et *NOM_F COMB* $\rightarrow$ *PRIX*

L'adresse d'un fournisseur est répétée autant de fois qu'il y a de combustibles fournis par ce fournisseur.

Il faut définir des contraints sur les schémas de relations qui permettent d'éviter ce type de problème. Ce sont les *formes normales*.

## Forme Normale de Boyce-Codd (BCNF)

**Définition**

Un schéma de relation est en *BCNF* si pour toute dépendance non-triviale $X \rightarrow A$, alors $X$ est une super-clé.

**Exemple** :

Le schéma :

*FA(NOM_F, ADRESSE_F)*
*FCP(NOM_F, COMB, PRIX)*

est en forme normale de Boyce-Codd.

## La normalisation

Le *problème de la normalisation* est celui de remplacer un schéma de base de données par un autre *équivalent* qui est en forme normale de Boyce-Codd.

Pour que le nouveau schéma soit considéré comme équivalent au schéma de départ, il faut qu'il soit toujours possible de reconstituer la base de donnée de départ à partir de celle organisée suivant le nouveau schéma.

## La décomposition en tant que technique de normalisation

On normalise habituellement par décomposition :
– un schéma $R$ est remplacé par un ensemble de schémas $R_i$, où $R_i \subset R$ ;
– la relation correspondante $r$ est remplacée par les projections $\pi_{R_i} r$.
Ce n'est possible que si les relations $\pi_{R_i} r$ contiennent la même information que $r$, ce qui n'est pas toujours le cas !

**Exemple** :

$r$ :

| $A$ | $B$ | $C$ |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

$\pi_{AB}(r)$ :

| $A$ | $B$ |
|---|---|
| 0 | 0 |
| 1 | 0 |

$\pi_{BC}(r)$ :

| $B$ | $C$ |
|---|---|
| 0 | 0 |
| 0 | 1 |

## La décomposition sans perte
## (*lossless join decomposition*)

Soit un schéma $R$, et $\rho = (R_1, \ldots, R_k)$ une décomposition de $R$ telle que $R = R_1 \cup \ldots \cup R_k$.

- Soit $r$ une relation de schéma $R$.
  La décomposition $\rho$ de $R$ est *sans perte pour* $r$ si
  $r = \pi_{R_1}(r) \bowtie \ldots \bowtie \pi_{R_k}(r)$.
- Soit $F$ un ensemble de dépendances sur $R$.
  La décomposition $\rho$ de $R$ est *sans perte par rapport à* $F$ si pour toute relation $r$ de schéma $R$ qui satisfait $F$, la décomposition est sans perte pour $r$, c'est-à-dire : $r = \pi_{R_1}(r) \bowtie \ldots \bowtie \pi_{R_k}(r)$.

**Notation** : $m_\rho(r) = \pi_{R_1}(r) \bowtie \ldots \bowtie \pi_{R_k}(r)$.

## Propriétés d'une décomposition :

**1.** $r \subseteq m_\rho(r)$ (L'inclusion peut être propre).

**Exemple :**

$r$ :

| $A$ | $B$ | $C$ |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

$\pi_{R_1}(r)$ :

| $A$ | $B$ |
|---|---|
| 0 | 0 |
| 1 | 0 |

$\pi_{R_2}(r)$ :

| $B$ | $C$ |
|---|---|
| 0 | 0 |
| 0 | 1 |

$m_\rho(r)$ :

| $A$ | $B$ | $C$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |

et $r \subset m_\rho(r)$

**Démonstration** : Soit $t$ un tuple qcq de $r$. On a : $t[R_1] \in \pi_{R_1}(r)$, ..., $t[R_k] \in \pi_{R_k}(r)$. Donc, par définition de $\bowtie$ : $t \in m_\rho(r)$.

**2.** $\pi_{R_i}(m_\rho(r)) = \pi_{R_i}(r)$.

**Démonstration** :

– On montre $\pi_{R_i}(r) \subseteq \pi_{R_i}(m_\rho(r))$.

Par la propriété 1, on a : $r \subseteq m_\rho(r)$.
Donc $\pi_{R_i}(r) \subseteq \pi_{R_i}(m_\rho(r))$.

– On montre $\pi_{R_i}(m_\rho(r)) \subseteq \pi_{R_i}(r)$.

Considérons un tuple $t_i \in \pi_{R_i}(m_\rho(r))$.
Il existe donc un tuple $t \in m_\rho(r)$ tel que $t_i = t[R_i]$ (par définition de $\pi_{R_i}$).
De plus, puisque $t \in m_\rho(r)$, il existe un tuple $t' \in r$ tel que $t'[R_i] = t[R_i]$ (par définition de $m_\rho(r)$).
Donc $t'[R_i] = t_i$ et par conséquent $t_i \in \pi_{R_i}(r)$. ∎

**Conclusion**

Il n'est pas possible de distinguer la relation $r$ de la relation $m_\rho(r)$ à partir des projections.

Autrement dit, si $r \neq m_\rho(r)$ il n'est pas possible de reconstituer $r$ à partir des $\pi_{R_i}(r)$.

## Critère de décomposition sans pertes

Soit $\rho = (R_1, R_2)$, une décomposition de $R$.

Cette décomposition est *sans perte par rapport à un ensemble de dépendances fonctionnelles* $F$ si

$$(R_1 \cap R_2) \rightarrow (R_1 - R_2) \in F^+$$

ou

$$(R_1 \cap R_2) \rightarrow (R_2 - R_1) \in F^+$$

**Démonstration**

On suppose que $(R_1 \cap R_2) \rightarrow (R_1 - R_2) \in F^+$ ou $(R_1 \cap R_2) \rightarrow (R_2 - R_1) \in F^+$.

On montre que pour toute relation $r$ de schéma $R$ : si $r$ satisfait $F$, alors $r = m_\rho(r)$.

Soit $r$ une relation de schéma $R$ qui satisfait $F$. On sait que $r \subseteq m_\rho(r)$. Montrons que $m_\rho(r) \subseteq r$.

Soit un tuple $t \in m_\rho(r)$. On veut montrer que $t \in r$.

Puisque $t \in m_\rho(r)$, il existe des tuples $t_1, t_2 \in r$ tels que $t_1[R_1] = t[R_1]$ et $t_2[R_2] = t[R_2]$.

Supposons que $(R_1 \cap R_2) \rightarrow (R_1 - R_2) \in F^+$ (l'autre cas se traite symétriquement).

Puisque $t_2[R_2] = t[R_2]$, on a : $t_2[R_1 \cap R_2] = t[R_1 \cap R_2]$.

Et puisque $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$ : $t_2[R_1] = t[R_1]$.

Donc, $t[R] = t_2[R]$ et puisque $t_2 \in r$ : $t \in r$ !

**Exemple :** Soit $R = ABC$ et $F = \{A \to B\}$.

$(AB, AC)$ est une décomposition de $R$ sans perte par rapport à $F$ vu que $AB \cap AC = A$ et $AB - AC = B$ et $A \to B$ est une dépendance de $F^+$.

En revanche, la décomposition de $R$ en $(AB, BC)$ n'est pas sans perte par rapport à $F$. En effet, on peut trouver une relation

$r$ :

| $A$ | $B$ | $C$ |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_2$ | $b_1$ | $c_2$ |

qui satisfait $F$ et telle que $\pi_{AB}(r)$ :

| $A$ | $B$ |
|---|---|
| $a_1$ | $b_1$ |
| $a_2$ | $b_1$ |

et

$\pi_{BC}(r)$ :

| $B$ | $C$ |
|---|---|
| $b_1$ | $c_1$ |
| $b_1$ | $c_2$ |

et $\pi_{AB}(r) \bowtie \pi_{BC}(r)$ :

| $A$ | $B$ | $C$ |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_2$ |
| $a_2$ | $b_1$ | $c_1$ |
| $a_2$ | $b_1$ | $c_2$ |

$\neq \quad r$ !

# Décomposition et dépendances

Comment traiter les dépendances lorsque l'on décompose ? Les dépendances sont elles conservées lors d'une décomposition ?

Soit $R$ un schéma de relation,
$F$ un ensemble de dépendances sur $R$,
$\rho = (R_1, \ldots, R_k)$ une décomposition de $R$.

– La *projection de $F$ sur un ensemble d'attributs $Z \subseteq R$* est
$\pi_Z(F) = \{X \to Y \in F^+ \mid XY \subseteq Z\}$.

– Une décomposition $\rho$ *conserve* $F$ si
$\bigcup_{i=1}^{k} \pi_{R_i}(F)$ couvre $F$.

L'intérêt de la conservation des dépendances lors de la décomposition est de permettre la vérification des dépendances sur la base de données.

Sans cette propriété, un joint est nécessaire pour déterminer si une modification viole une dépendance.

**Exemple :** $R = (CITY, ST, ZIP)$

avec $F = \{\ CITY\ ST \rightarrow ZIP,\ \ ZIP \rightarrow CITY\ \}$.

La décomposition en $(ST\ ZIP,\ \ CITY\ ZIP)$ est sans perte vu que

$ST\ ZIP \cap CITY\ ZIP = \ \ ZIP \rightarrow CITY = CITY\ ZIP - ST\ ZIP$

mais seule la dépendance $ZIP \rightarrow CITY$ est conservée.

**Remarque** : Les 2 propriétés, *conservation des dépendances* et *joint sans perte*, sont indépendantes. On peut avoir l'une sans l'autre et vice-versa.

La méthode directe pour vérifier si une décomposition conserve les dépendances est de :

- calculer $F^+$ et calculer $\bigcup_{i=1}^{k} \pi_{R_i}(F)$
- vérifier si $\bigcup_{i=1}^{k} \pi_{R_i}(F)$ couvre $F$.

## Algorithme de décomposition en BCNF

*Données* : Soit un schéma $R$ et un ensemble de dépendances fonctionnelles $F$.

L'algorithme procède par décompositions successives pour obtenir une décomposition de $R$ sans perte par rapport à $F$, mais sans garantie de conservation des dépendances.

- Si $R$ n'est pas en BCNF, soit une dépendance non triviale $X \rightarrow A$ de $F^+$, où $X$ n'est pas une super-clé.
- On décompose $R$ en $R_1 = R - A$ et $R_2 = XA$ (sans perte vu le critère : $R_1 \cap R_2 = X$ et $R_2 - R_1 = A$).
- On applique l'algorithme à :

  $R_1$, $\pi_{R_1}(F)$ $\qquad$ $R_2$, $\pi_{R_2}(F)$

Puisque les relations deviennent de plus en plus petites, la décomposition doit s'arrêter.

**Exemple** : Relation de schéma $CT\underline{H}R\underline{S}G$

| | | | |
|---|---|---|---|
| $C$ : *course* | $R$ : *room* | $C \to T$ | $\boxed{CS \to G}$ |
| $T$ : *teacher* | $S$ : *student* | $HR \to C$ | $HS \to R$ |
| $H$ : *hour* | $G$ : *grade* | $HT \to R$ | |

Une seule clé : $HS$

$R_1$ : $\underline{CS}G$ avec $CS \to G$ BCNF !

$R_2$ : $CT\underline{H}R\underline{S}$ avec $\boxed{C \to T}$, $HR \to C$, $HT \to R$, $HS \to R$

$R_3$ : $\underline{C}T$ avec $C \to T$ BCNF !

$R_4$ : $C\underline{H}R\underline{S}$ avec $HS \to R$, $\boxed{HC \to R}$, $HR \to C$

$R_5$ : $\underline{CHR}$ avec $HC \to R$, $HR \to C$ BCNF !

$R_6$ : $C\underline{HS}$ avec $HS \to C$ BCNF !

*Résultat* : $CSG$, $CT$, $CHR$, $CHS$.

La dépendance $TH \to R$ n'est pas conservée par la décomposition.

## Faut-il toujours normaliser en BCNF ?

- Reconsidérons le schéma $R = (CITY, ST, ZIP)$ avec $F = \{ CITY\,ST \rightarrow ZIP, ZIP \rightarrow CITY \}$.
- Ce schéma n'est pas en BCNF, mais il peut être décomposé en en $(ST\ ZIP)$, $(CITY\ ZIP)$ qui est en BCNF.
- Toutefois, si l'on fait cette décomposition, la dépendance $CITY\,ST \rightarrow ZIP$ n'est pas conservée et retrouver un code postal à partir d'une ville et d'une rue nécessite un joint.
- Il semble donc préférable de ne pas décomposer, car même si le schéma n'est pas en BCNF, il n'est pas vraiment problématique : il est en *troisième forme normale* (3FN).

## La troisième forme normale (3FN)

Pour définir la troisième forme normale on distingue les attributs
- qui font partie d'une clé (*premiers*) de ceux qui,
- ne font pas partie d'une clé (*non premiers*).

La troisième forme normale se définit comme la BCNF, sauf qu'elle ne tient pas compte des attributs non premiers.

**Définition**

Un schéma de relation est en *3FN* si pour toute dépendance non-triviale $X \rightarrow A$ où $A$ est non-premier, alors $X$ est une super-clé.

L'intérêt de la 3FN est qu'il existe des algorithmes permettant de décomposer en 3FN tout en conservant les dépendances.

## Les première et deuxième formes normales

L'appellation "troisième forme normale" résulte du fait qu'une première et une deuxième forme normale ont été définies.

- La première forme normale (1FN) précise que les attributs d'une relation ont une valeur atomique. Nous avons fait cette hypothèse d'emblée en introduisant le modèle relationnel.
- La deuxième forme normale (2FN) exclut les dépendance non-triviale $X \rightarrow A$ où $A$ est non-premier et où $X$ est un sous-ensemble propre d'une clé. C'est une contrainte moins forte que celle donnée par la 3FN et, a fortiori, par la BCNF.

## Les Dépendances à valeurs multiples

Tant qu'à présent, nous avons utilisé un seul concept pour représenter la structure de l'information menant à la nécessité de normaliser : **la dépendance fonctionnelle**.

Toutefois, la nécessité de normaliser peut apparaître en dehors de la présence de dépendances fonctionnelles ; il y a aussi d'autres types de dépendances et, en particulier, les **dépendances à valeurs multiples (DVM)**.

Intuitivement, dans une relation $R$, on a une dépendance à valeurs multiples entre $X$ et $Y$ ($X, Y \subseteq R$) si les valeurs possibles de $Y$ sont déterminées par $X$, et ce, indépendamment du contenu du reste de la relation. On note cela : $X \twoheadrightarrow Y$ ($X$ *multi-détermine* $Y$).

**Exemple** : $R(COURS, JOUR, ETUDIANT)$

$$COURS \twoheadrightarrow JOUR$$

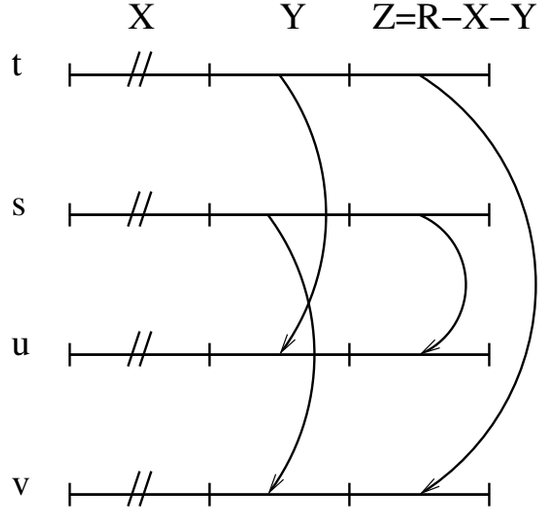
$$COURS \twoheadrightarrow ETUDIANT$$

| $COURS$ | $JOUR$ | $ETUDIANT$ |
|---|---|---|
| #1 | *Mardi* | *A. Dupont* |
| #1 | *Jeudi* | *A. Dupont* |
| #1 | *Mardi* | *B. Durand* |
| #1 | *Jeudi* | *B. Durand* |

# Les Dépendances à valeurs multiples : définition formelle

Une relation $r$ de schéma $R$ satisfait une *dépendance à valeurs multiples (DVM)* $X \twoheadrightarrow Y$ si pour toute paire de tuples $t, s \in r$ tels que $t[X] = s[X]$, il existe deux tuples $u, v \in r$ tels que :

1. $u[X] = v[X] = t[X] = s[X]$
2. $u[Y] = t[Y]$ et $u[R–X–Y] = s[R–X–Y]$
3. $v[Y] = s[Y]$ et $v[R–X–Y] = t[R–X–Y]$



**Exemple** : Relation de schéma $CTHRSG$

| | | | |
|---|---|---|---|
| $C$ : *course* | $R$ : *room* | $C \to T$ | $CS \to G$ |
| $T$ : *teacher* | $S$ : *student* | $HR \to C$ | $HS \to R$ |
| $H$ : *hour* | $G$ : *grade* | $HT \to R$ | |

Les dépendances $C \twoheadrightarrow HR$ et $C \twoheadrightarrow SG$ et $HR \twoheadrightarrow SG$ sont à associer à ce schéma, mais pas les dépendances $C \twoheadrightarrow H$ et $C \twoheadrightarrow R$ comme, pour ce dernier point, le montre l'exemple d'extension ci-dessous.

| *C* | *T* | *H* | *R* | *S* | *G* |
|---|---|---|---|---|---|
| *INFO009* | *ProfX* | *lun 14 :00* | *R3* | *A. Dupont* | *18* |
| *INFO009* | *ProfX* | *mer 14 :00* | *R7* | *A. Dupont* | *18* |
| *INFO009* | *ProfX* | *ven 10 :00* | *I21* | *A. Dupont* | *18* |
| *INFO009* | *ProfX* | *lun 14 :00* | *R3* | *B. Durand* | *14* |
| *INFO009* | *ProfX* | *mer 14 :00* | *R7* | *B. Durand* | *14* |
| *INFO009* | *ProfX* | *ven 10 :00* | *I21* | *B. Durand* | *14* |

## Dépendances à valeurs multiples triviales et non triviales : exemples

1. $R(A, B)$ $\quad A \twoheadrightarrow B$ ? **Oui**, trivialement : toujours satisfaite !

| $A$ | $B$ |
|---|---|
| $a$ | $b$ |
| $a$ | $c$ |
| $d$ | $b$ |
| $d$ | $e$ |

2. $R(A, B, C)$ $\quad A \twoheadrightarrow B$ ?

| $A$ | $B$ | $C$ |
|---|---|---|
| $a$ | $b$ | $d$ |
| $a$ | $c$ | $e$ |

**non**

| $A$ | $B$ | $C$ |
|---|---|---|
| $a$ | $b$ | $d$ |
| $a$ | $c$ | $e$ |
| $a$ | $b$ | $e$ |
| $a$ | $c$ | $d$ |

**oui**

3. $R(A, B, C)$ avec $A \rightarrow B$.

La DVM $A \twoheadrightarrow B$ est toujours satisfaite lorsque $A \rightarrow B$ est satisfaite. En effet, soit $t, s \in r$ tels que $t[A] = s[A]$. Vu que $t[B] = s[B]$, il existe bien des tuples $u, v \in r$ tel que $= u[A] = v[A] = t[A] = s[A]$, $u[B] = t[B]$, $u[C] = s[C]$, $v[B] = s[B]$, et $v[C] = t[C]$ ; ces tuples sont respectivement $s$ et $t$.

## Propriétés des dépendances à valeurs multiples

Les dépendances à valeurs multiples satisfont un certain nombre de règles qui permettent de raisonner à leur sujet.

- Si $Y \subseteq X$, alors $X \twoheadrightarrow Y$ (réflexivité)
- Si $X \twoheadrightarrow Y$, alors $X \twoheadrightarrow (R - XY)$ (complémentation pour DVM)
- Si $X \rightarrow Y$, alors $X \twoheadrightarrow Y$ (reproduction)
- Si $X \twoheadrightarrow Y$ et $X \twoheadrightarrow Z$, alors $X \twoheadrightarrow YZ$ (union pour DVM)
- Si $X \twoheadrightarrow Y$ et $X \twoheadrightarrow Z$, alors $X \twoheadrightarrow Y \cap Z$ (intersection pour DVM)
- Si $X \twoheadrightarrow Y$ et $X \twoheadrightarrow Z$, alors $X \twoheadrightarrow Y \setminus Z$ (différence pour DVM)

**Notes** :

- Les règles de décomposition et de transitivité ne sont pas valides pour les DVM. Par exemple, dans le schéma $CTHRSG$, on a $C \twoheadrightarrow HR$ et $HR \twoheadrightarrow H$ (trivialement), mais $C \not\twoheadrightarrow H$ !
- Comme dans le cas des dépendances fonctionnelles, il est possible de déterminer si une DVM est une conséquence logique d'un ensemble de dépendances, mais cela nécessite des techniques relativement complexes.

## Dépendances à valeurs multiples et décomposition sans perte

Pour toute relation $r$ de schéma $R$, $r$ satisfait la DVM $X \twoheadrightarrow Y$ si et seulement si la décomposition $\rho = (XY, X(R - XY))$ est sans perte pour $r$, c'est-à-dire $r = m_\rho(r)$.

**Démonstration** : Soit $r$ une relation quelconque de schéma $R$.

**si ($\Leftarrow$) :** On suppose la décomposition sans perte pour $r$, càd $r = \pi_{XY}(r) \bowtie \pi_{XZ}(r)$ avec $Z = R - XY$.

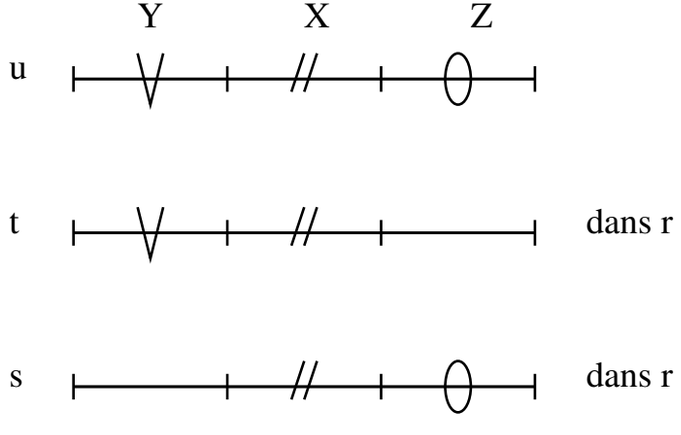
Considérons 2 tuples $t, s \in r$ tels que $t[X] = s[X]$. On a :
$t[XY] \in \pi_{XY}(r)$ et $s[XZ] \in \pi_{XZ}(r)$.

Donc les tuples $u, v$ tel que $u[X] = v[X] = t[X] = s[X]$, $u[Y] = t[Y]$, $u[Z] = s[Z]$, $v[Y] = s[Y]$ et $v[Z] = t[Z]$ appartiennent à $\pi_{XY}(r) \bowtie \pi_{XZ}(r) = r$.

**seulement si ($\Rightarrow$) :** Supposons que $r$ satisfait la dépendance $X \twoheadrightarrow Y$ et montrons que le joint est sans perte, càd que si $u \in \pi_{XY}(r) \bowtie \pi_{XZ}(r)$, alors $u \in r$.

Si $u \in \pi_{XY}(r) \bowtie \pi_{XZ}(r)$, il existe $t, s \in r$ tels que $t[XY] = u[XY]$ et $s[XZ] = u[XZ]$, ou encore $u[X] = t[X] = s[X]$, $u[Y] = t[Y]$, $u[Z] = s[Z]$.

Puisque $r$ satisfait $X \twoheadrightarrow Y$, le tuple $u$ doit aussi être dans $r$ (par définition des DVM).



**Remarque** : Cette propriété peut aussi s'écrire comme suit.

Soit $R$ un schéma de relation, et $\rho = (R_1, R_2)$ une décomposition de $R$.

Pour toute relation $r$ de schéma $R$ :

$$r \text{ satisfait la DVM } (R_1 \cap R_2) \twoheadrightarrow (R_1 - R_2)$$
$$\text{(ou } (R_1 \cap R_2) \twoheadrightarrow (R_2 - R_1), \text{ par complémentation)}$$

**si et seulement si**

la décomposition $(R_1, R_2)$ de $R$ est sans perte pour $r$, c'est-à-dire
$r = m_\rho(r)$.

**Remarques :**

Il y a une parfaite équivalence entre la notion de dépendance à valeurs multiples et celle de décomposition sans perte.

Dans le cas des dépendances fonctionnelles, la présence de la dépendance fonctionnelle permet la décomposition sans perte, mais une relation $r$ peut être décomposable sans perte sans qu'elle ne satisfasse de dépendance fonctionnelle.

## Critère de décomposition sans perte

Soit $\rho = (R_1, R_2)$, une décomposition de $R$.

Cette décomposition est *sans perte par rapport à un ensemble de dépendances (DF et DVM)* $D$ si et seulement si

$$(R_1 \cap R_2) \twoheadrightarrow (R_1 - R_2) \in D^+$$

(ou $(R_1 \cap R_2) \twoheadrightarrow (R_2 - R_1) \in D^+$, par complémentation)

**Démonstration** :

Conséquence directe de la propriété établissant l'équivalence entre satisfaction d'une DVM par une relation et décomposition sans perte pour cette relation.

## Dépendances à valeurs multiples et formes normales

Une dépendance à valeur multiples $X \twoheadrightarrow Y$ est gênante si l'association entre les valeurs de $X$ et celles de $Y$ est répétée, c'est-à-dire apparaît pour plusieurs valeurs des autres attributs $Z = R - XY$) de la relation.

Cela peut se produire, sauf si

- $Z = \emptyset$ ($X \twoheadrightarrow Y$ est alors triviale), ou
- $Y \subset X$ ($X \twoheadrightarrow Y$ est alors triviale), ou
- $X \rightarrow R$ ($X$ est alors une super-clé).

Ceci nous mène à la définition de la **4ème Forme Normale (4FN)**.

## La quatrième forme normale (4FN)

Un schéma de relation est en *4FN* si pour toute dépendance $X \twoheadrightarrow Y$,
– soit $Y \subseteq X$,
– soit $XY = R$,
– soit $X$ est une super-clé de $R$.

**Remarque** : *La 4FN implique la BCNF.*
En effet :
– une dépendance $X \rightarrow Y$ est aussi une dépendance $X \twoheadrightarrow Y$ ;
– si $Y \subset X$, la dépendance $X \rightarrow Y$ est triviale ;
– si $XY = R$, alors $X \rightarrow R$ et $X$ est une super-clé.

Pour obtenir la 4FN, on peut procéder par décomposition comme pour la BCNF.

## Un algorithme de décomposition en 4FN

Soit un schéma $R$ et un ensemble de dépendances (fonctionnelles et à valeurs multiples) $D$.

L'algorithme procède par décompositions successives pour obtenir une décomposition de $R$ sans perte par rapport à $D$, mais sans garantie de conservation des dépendances.

- Si $R$ n'est pas en 4FN, soit une dépendance $X \twoheadrightarrow Y$ de $D^+$ telle que $Y \neq \emptyset$, $Y \not\subseteq X$, $XY \subset R$ et $X$ n'est pas une super-clé.
  On peut supposer $X \cap Y = \emptyset$ puisque $X \twoheadrightarrow (Y - X)$ est impliqué par $X \twoheadrightarrow Y$.
- On décompose $R$ en $R_1 = XY$ et $R_2 = X(R - XY)$ (sans perte vu le critère).
- On applique l'algorithme à : $R_1$, $\pi_{R_1}(D)$, $R_2$, $\pi_{R_2}(D)$

Puisque les relations deviennent de plus en plus petites, la décomposition doit s'arrêter.

Nous n'avons pas vu de méthode systématique pour calculer $D^+$, mais en pratique une utilisation directe des règles de raisonnement données suffit.

**Exemple** : Relation de schéma $CTHRSG$

| | | | |
|---|---|---|---|
| $C$ : *course* | $R$ : *room* | $C \to T$ | $CS \to G$ |
| $T$ : *teacher* | $S$ : *student* | $HR \to C$ | $HS \to R$ |
| $H$ : *hour* | $G$ : *grade* | $HT \to R$ | |

$\boxed{C \twoheadrightarrow HR}$ est une dépendance à associer à ce schéma. Il y a une seule clé : $HS$

Toutes les MVD qui s'appliquent naturellement à ce schéma sont dérivables de l'ensemble de DF ci-dessus et de $C \twoheadrightarrow HR$.

Par exemple :

- De $C \twoheadrightarrow HR$ et $C \to T$, on peut dériver $C \twoheadrightarrow SG$ :
  - de $C \to T$, on obtient $C \twoheadrightarrow T$, et ensuite $C \twoheadrightarrow HRT$ par union ;
  - la règle de complémentation donne alors $C \twoheadrightarrow SG$.
- De $HR \to CT$, on peut dériver $HR \twoheadrightarrow CT$.
- Par la règle de complémentation on obtient $HR \twoheadrightarrow SG$.

Décomposition en 4FN :

$R_1$ : $C\underline{HR}$ avec $C \twoheadrightarrow HR$ 4FN !

$R_2$ : $\underline{CT}SG$ avec $C \twoheadrightarrow T$

$R_3$ : $\underline{C}T$ avec $C \rightarrow T$ 4FN !

$R_4$ : $\underline{CS}G$ avec $CS \rightarrow G$ 4FN !

*Résultat* : $CHR$, $CT$, $CSG$.

Décomposition sans perte où tous les sous-schémas sont en 4FN.

**Remarque** : Si l'on ignore la DVM $C \twoheadrightarrow HR$, alors la décomposition ci-dessus n'est pas nécessairement sans perte.

## Autres types de dépendances

### Dépendances "joint"

Les DVM caractérisent les cas où une décomposition en 2 relations est sans perte.

Il est parfois possible de décomposer une relation sans perte en trois sous-schémas sans qu'il y ait de décomposition en deux sous-schémas qui soit sans perte.

On généralise donc la notion de DVM en la notion de *dépendance-joint*.

Une relation $r$ de schéma $R$ satisfait la *dépendance-joint* $*[R_1, \ldots R_k]$ si

$$r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \ldots \bowtie \pi_{R_k}(r).$$

Ceci mène à la définition d'une **5ème forme normale (5FN)** ou encore **"project-join normal form"**.

## La cinquième forme normale (5FN)

Une relation $r$ de schéma $R$ est en *5$^{ème}$ forme normale (5FN) ou "project-join normal form"* par rapport à un ensemble de dépendances fonctionnelles et joint $F$ si, pour toute dépendance joint $*[R_1, \ldots R_k]$ impliquée par $F$,

– soit elle est triviale (vraie de toute relation de schéma $R$),

– soit chaque $R_i$ est une super-clé de $R$.