SMI6U05 : Bases de données
Conception de base de données :
DFs, décompositions, et formes nomales

Meryem Billami et Luigi Santocanale
LIS, Aix-Marseille Université

14 mars 2019

Plan

Plan

Une table (entité), est elle bien conçue ?

(Contre)Exemple :

Voiture

nv	marque	type	puissance	couleur
872RH75	Peugeot	P206A	7	bleue
975AB80	Peugeot	P206A	7	rouge

- Les données sont redondantes !
- Anomalies de mise à jour :
  - introduction d'incohérences
    Exemple : correction dans une seule tuple
  - introduction de valeurs nulles
    Exemple : ajout d'un nouvel modèle de voiture ;
    `nv` et `couleur` NULL ?

Evolution d'une BD

- Une base évolue : donnée ET schémas
- Souvent (manque de temps, pressions commerciales, arrogance) :
  modification directe du Modèle Physique des Données
- Du coup :
  - incohérence,
  - absence de documentation,
  - mauvaise base.
- Evolution = nouveau MCD, puis MLD !
- Sinon : reverse engineering
- Problématique actuellement très étudiée

Plan

Dépendances fonctionnelles

- Liens fonctionnels entre attributs
- Ces liens sont souvent laissés implicites dans le schéma
- Une bonne conception explicite ces contraintes dans le schéma

Exemple

Voiture	nv	marque	type	puissance	couleur
	872RH75	Peugeot	P206A	7	bleue
	975AB80	Peugeot	P206A	7	rouge

Voiture	nv	marque	type	puissance	couleur

Notation :

$\texttt{type} \rightarrow \texttt{marque}$, $\texttt{type} \rightarrow \texttt{puissance}$,

$\texttt{type} \rightarrow \texttt{marque}, \texttt{puissance}$

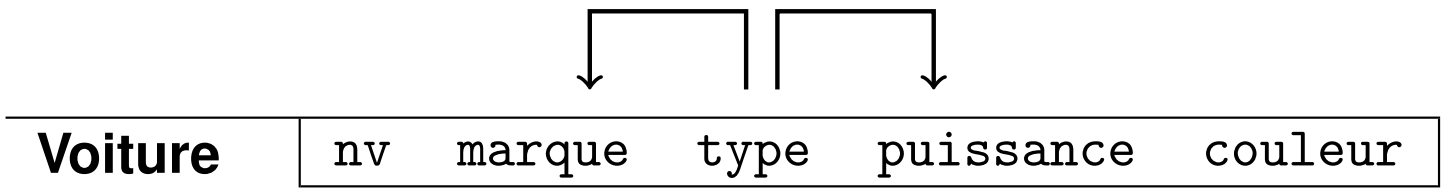
D'autres DFs :

$\texttt{nv} \rightarrow (\texttt{marque}, \texttt{type}, \texttt{puissance}, \texttt{couleur})$

$(\texttt{type}, \texttt{marque}) \rightarrow \texttt{puissance}$

Exemple

Voiture	nv	marque	type	puissance	couleur
	872RH75	Peugeot	P206A	7	bleue
	975AB80	Peugeot	P206A	7	rouge



Notation :

$\texttt{type} \rightarrow \texttt{marque}$, $\texttt{type} \rightarrow \texttt{puissance}$,

$\texttt{type} \rightarrow \texttt{marque},\texttt{puissance}$

D'autres DFs :

$\texttt{nv} \rightarrow (\texttt{marque},\texttt{type},\texttt{puissance},\texttt{couleur})$

$(\texttt{type},\texttt{marque}) \rightarrow \texttt{puissance}$

Exemple

Voiture	nv	marque	type	puissance	couleur
	872RH75	Peugeot	P206A	7	bleue
	975AB80	Peugeot	P206A	7	rouge

Voiture	nv	marque	type	puissance	couleur

Notation :

$\texttt{type} \rightarrow \texttt{marque}, \quad \texttt{type} \rightarrow \texttt{puissance},$

$\texttt{type} \rightarrow \texttt{marque}, \texttt{puissance}$

D'autres DFs :

$\texttt{nv} \rightarrow (\texttt{marque}, \texttt{type}, \texttt{puissance}, \texttt{couleur})$

$(\texttt{type}, \texttt{marque}) \rightarrow \texttt{puissance}$

DFs (II)

- Permettent de repérer :
  - redondances
  - anomalies
- Buts :
  - affiner,
  - préciser,
  - rendre plus concis un MCD
  - data mining
- Aussi : améliorer la performance de certaines requêtes

Amélioration des requêtes

Calculer

$$A := \{\, (x,y,z) \mid \text{il existe } y', z' \text{ tels que } R(x,y,z') \text{ et } R(x,y',z) \,\}$$

Nécessaire calculer une jointure :

$$\Pi_{x,y,z'}\,(\, R \bowtie_{x=x'} R' \,)$$

Si $y = f(x)$ (et donc $y' = f(x) = y$)

$$\begin{aligned} A &= \{\, (x,y,z) \mid \text{il existe } z' \text{ tels que } R(x,y,z') \text{ et } R(x,y,z) \,\} \\ &= \{\, (x,y,z) \mid R(x,y,z) \text{ et } (\text{il existe } z' \text{ tels que } R(x,y,z')) \,\} \\ &= \{\, (x,y,z) \mid R(x,y,z) \,\} = R\,. \end{aligned}$$

Pas de jointure ! ! !

Amélioration des requêtes

Calculer

$$A := \{\, (x,y,z) \mid \text{il existe } y', z' \text{ tels que } R(x,y,z') \text{ et } R(x,y',z) \,\}$$

Nécessaire calculer une jointure :

$$\Pi_{x,y,z'}\,(\, R \bowtie_{x=x'} R' \,)$$

Si $y = f(x)$ (et donc $y' = f(x) = y$)

$$\begin{aligned} A &= \{\, (x,y,z) \mid \text{il existe } z' \text{ tels que } R(x,y,z') \text{ et } R(x,y,z) \,\} \\ &= \{\, (x,y,z) \mid R(x,y,z) \text{ et } (\text{il existe } z' \text{ tels que } R(x,y,z')) \,\} \\ &= \{\, (x,y,z) \mid R(x,y,z) \,\} = R\,. \end{aligned}$$

Pas de jointure ! ! !

Amélioration des requêtes

Calculer

$$A := \{\, (x,y,z) \mid \text{il existe } y', z' \text{ tels que } R(x,y,z') \text{ et } R(x,y',z) \,\}$$

Nécessaire calculer une jointure :

$$\Pi_{x,y,z'}(R \bowtie_{x=x'} R')$$

Si $y = f(x)$ (et donc $y' = f(x) = y$)

$$\begin{aligned} A &= \{\, (x,y,z) \mid \text{il existe } z' \text{ tels que } R(x,y,z') \text{ et } R(x,y,z) \,\} \\ &= \{\, (x,y,z) \mid R(x,y,z) \text{ et } (\text{il existe } z' \text{ tels que } R(x,y,z')) \,\} \\ &= \{\, (x,y,z) \mid R(x,y,z) \,\} = R. \end{aligned}$$

Pas de jointure ! ! !

Amélioration des requêtes

Calculer

$$A := \{\, (x,y,z) \mid \text{il existe } y', z' \text{ tels que } R(x,y,z') \text{ et } R(x,y',z) \,\}$$

Nécessaire calculer une jointure :

$$\Pi_{x,y,z'}(R \bowtie_{x=x'} R')$$

Si $y = f(x)$ (et donc $y' = f(x) = y$)

$$\begin{aligned} A &= \{\, (x,y,z) \mid \text{il existe } z' \text{ tels que } R(x,y,z') \text{ et } R(x,y,z) \,\} \\ &= \{\, (x,y,z) \mid R(x,y,z) \text{ et } (\text{il existe } z' \text{ tels que } R(x,y,z')) \,\} \\ &= \{\, (x,y,z) \mid R(x,y,z) \,\} = R. \end{aligned}$$

Pas de jointure ! ! !

Plan

Décompositions de relations

- Décomposition : projection
- Recomposition : jointure (naturelle).

- Une décomposition d'une relation $R(a_1, a_2, ..., a_n)$ est une collection de relations $R_1, R_2, ...R_m$ telles que
  - $R_i = \Pi_{A_i}(R)$, $i = 1, \ldots, m$
  - $\bigcup_{i=1,...,m} A_i = \{a_1, \ldots, a_n\}$.

- Une décomposition $R_1, R_2, ..., R_m$ de R est sans perte d'information si

$$R = R_1 \bowtie R_2 \bowtie \ldots \bowtie R_m .$$

Perte d'information

- Soit $R_1, R_2 \ldots, R_m$ une decomposition de R.
- $R_1 \bowtie R_2 \ldots \bowtie R_m$ a le même schéma (les mêmes attributs) que R mais pas forcément le même contenu.
- On a, toujours,

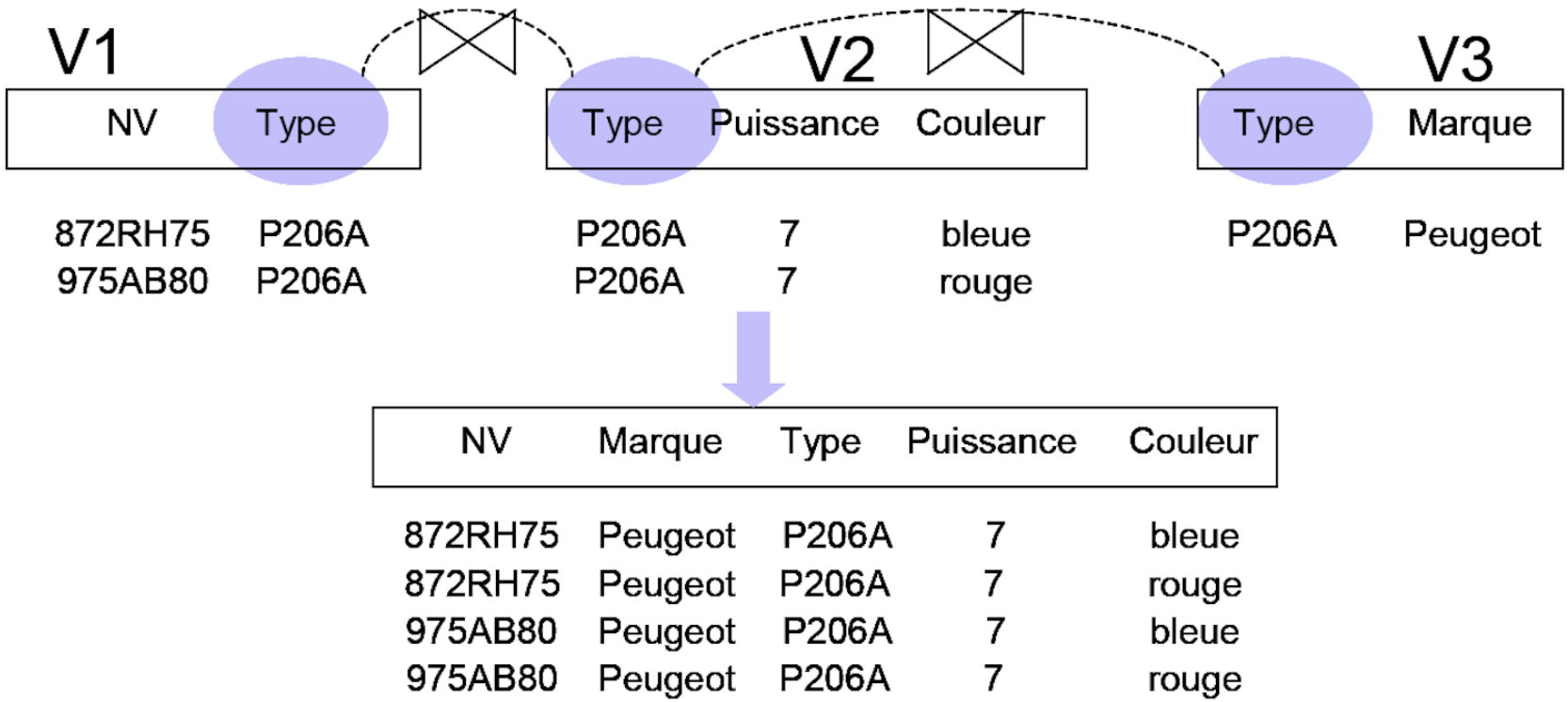
$$R \subseteq R_1 \bowtie R_2 \bowtie \ldots \bowtie R_m .$$

- Si l'inclusion est stricte : perte d'information.
  La jointure est moins détaillée/moins précise/plus générale que R)
- Si égalité : décomposition sans perte d'information

Perte d'information : exemple

Voiture

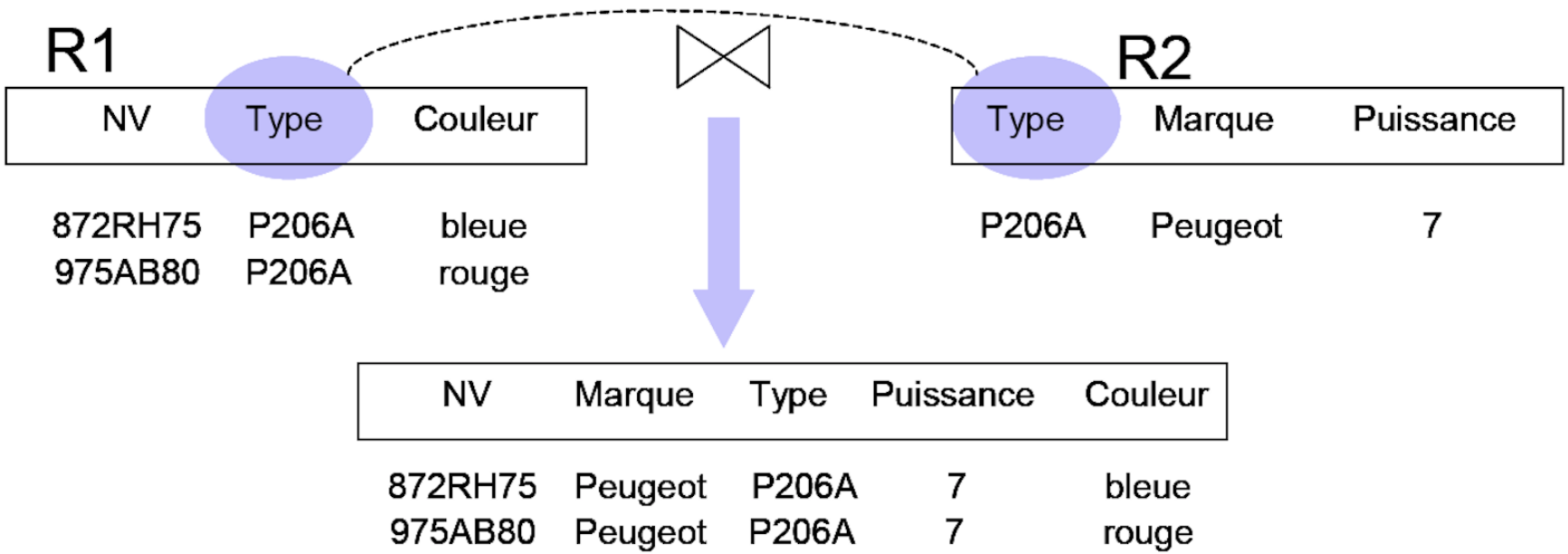
nv	marque	type	puissance	couleur
872RH75	Peugeot	P206A	7	bleue
975AB80	Peugeot	P206A	7	rouge



Sans perte : exemple

Voiture

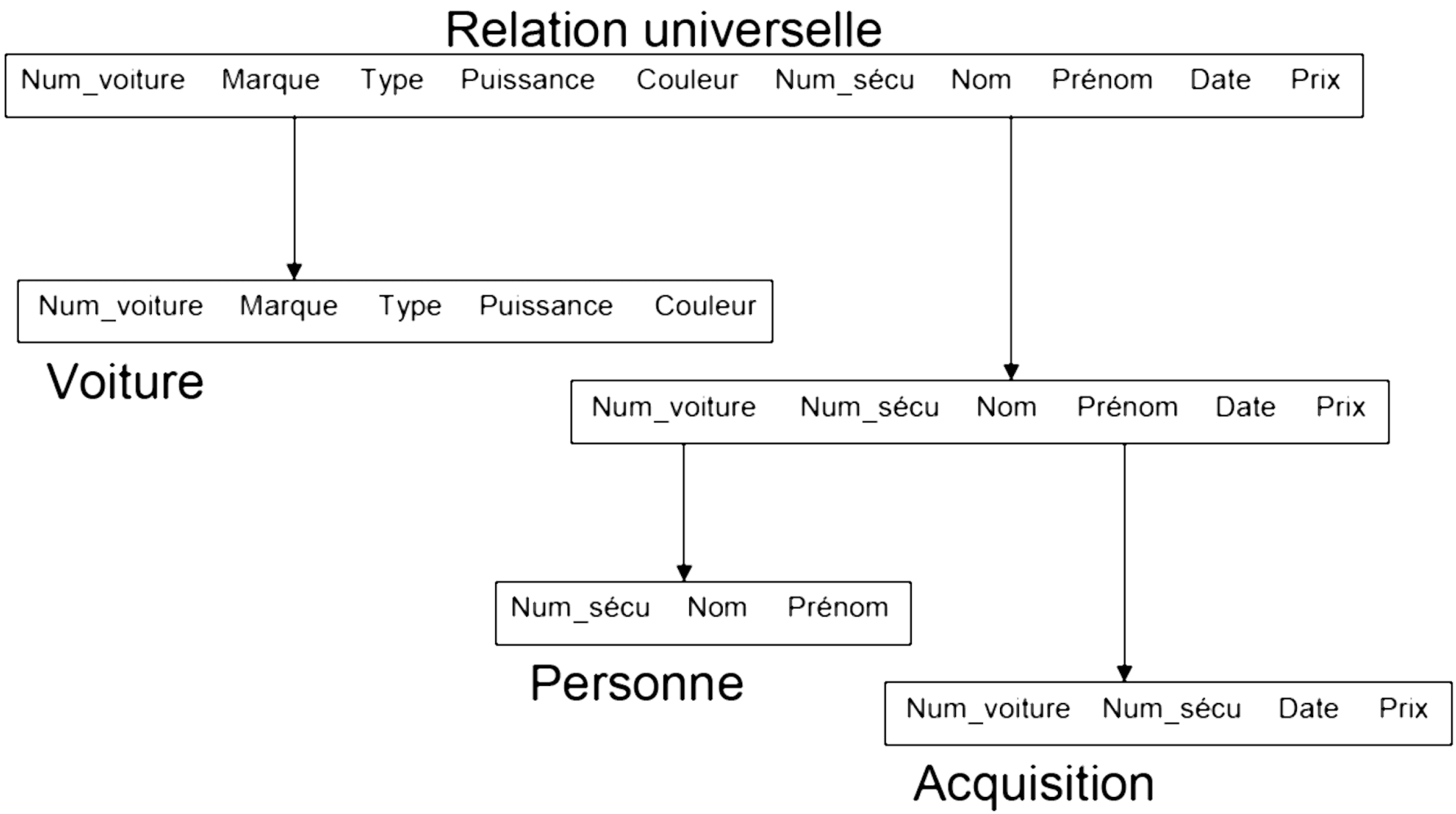
nv	marque	type	puissance	couleur
872RH75	Peugeot	P206A	7	bleue
975AB80	Peugeot	P206A	7	rouge



Modélisation : approche par décomposition

- Relation universelle : relation unique dont le schéma est constitué de tous les attributs (les attributs du même concept ont le même nom, et vice versa).
- A partir de la relation universelle, isoler les entités et associations élémentaires (canoniques, ne pouvant être de re-découpées) par une processus de raffinements successifs.
- Le même approche peut être utilisé avec un schéma/table trop grand.

Approche par décomposition : exemple



Plan

Dépendances fonctionnelles et décomposition

- Les dépendances fonctionnelles permettent de sélectionner les attributs pour une décomposition sans perte.
- Exemple : `marque`, `type` et `puissance` partagent respectivement les mêmes valeurs : ces attributs sont fonctionnellement dépendants et donc il est intéressant que la décomposition les garde ensemble.
- Gain : factorisation.

Dépendance fonctionnelle

- DF : lien sémantique entre 2 attributs.
- Lors de la conception :
  peut exprimer une contrainte à mettre en ouvre dans la BD.
- Quand la BD évolue :
  peut être découvert par analyse des données (lignes) de la base.

Définition formelle

Définition. Soient

- $R(a_1, a_2, ..., a_n)$ une relation,
- $X, Y \subseteq \{a_1, a_2, \ldots, a_n\}$.

On dit que

- Y dépend fonctionnellement de X dans R, ou bien
- X détermine Y dans R, ou encore
- R satisfait $X \rightarrow Y$,

et écrit on cela par

$$R \models X \rightarrow Y,$$

si, pour tout $t, t' \in R$,

$$\pi_X(t) = \pi_X(t') \text{ implique } \pi_Y(t) = \pi_Y(t').$$

Rappel : $\pi_X(t) = t \restriction_X$.

Dépendance fonctionnelle (suite)

- On souhaite manipuler des expression du type

$$X \rightarrow Y$$

  sans faire référence à une relation R à un istant donné.
- Par exemple, lors de la conception, quand la table est vide.
- Y dépend fonctionnellement de X si, étant donnée une valeur de X, il lui correspond une unique valeur de Y.
- C'est une assertion sur toutes les valeurs possibles !
  Non pas par rapport au contenu de la base à un instant donné.
  Assertion sur le monde réel.
- Exemple : les DFs pourraient exister :

$$\textit{Type} \rightarrow \textit{Marque}, \qquad \textit{Type} \rightarrow \textit{Puissance},$$
$$\textit{NV} \rightarrow \textit{couleur},$$
$$(\textit{Type}, \textit{Marque}) \rightarrow \textit{Puissance}.$$

Schéma de relation avec DFs

Définition. Une schéma de relation avec contraintes de DFs est une couple $(R(a_1, \ldots, a_n), \mathbf{F})$ tel que :

- $R(a_1, \ldots, a_n)$ est un schéma de relation (nom de la relation plus noms des colonnes/attributs)
- **F** un ensemble de DFs de la frome $X \rightarrow Y$ avec $X, Y \subseteq \{a_1, \ldots, a_n\}$.

Définition formelle

Définition. On dit que

$$X_1 \to Y_1, \ldots, X_k \to Y_k \quad \text{implique} \quad X_0 \to Y_0$$

si

$$R \models X_1 \to Y_1, \ldots, R \models X_k \to Y_k \quad \text{implique} \quad R \models X_0 \to Y_0 .$$

pour toute relation $R(a_1, a_2, ..., a_n)$ t.q. $\bigcup_{i=0,\ldots,k} X_i \cup Y_i \subseteq \{a_1, a_2, \ldots, a_n\}$.

En particulier (cas $k = 0$) :

Définition. On dit/écrit

$$X \to Y$$

si

$$R \models X \to Y$$

pour toute relation $R(a_1, a_2, ..., a_n)$ telle que $X, Y \subseteq \{a_1, a_2, \ldots, a_n\}$.

Axiomatisation et propriétés des DFs

Notation : $(X, Y) = X \cup Y$.

Axiomatisation de Armstrong 1974 :

- **Réflexivité** : $X \rightarrow Y$, pour $Y \subseteq X$
- **Augmentation** : $X \rightarrow Y$ implique $(X, Z) \rightarrow (Y, Z)$,
- **Transitivité** : $X \rightarrow Y$, $Y \rightarrow Z$, implique $X \rightarrow Z$.

D'autres propriétés se dérivent des précédentes :

- **Union** : $X \rightarrow Y$, $X \rightarrow Z$ implique $X \rightarrow (Y, Z)$,
- **Pseudo-transitivité** : $X \rightarrow Y$, $(W, Y) \rightarrow Z$, implique $(W, X) \rightarrow Z$,
- **Décomposition** : si $Z \subseteq Y$, alors $X \rightarrow Y$ implique $X \rightarrow Z$.

L'axiomatisation de Armstrong est correcte et complète.

Fermeture d'un ensemble de DFs

Soit **F** un ensemble de DFs.

Notons $\mathbf{F}^+$ le plus petit ensemble contenant **F**, fermé par rapport aux règles de réflexivité, augmentation, transitivité. On appelle $\mathbf{F}^+$ la fermeture de **F**.

Posons :

$$R \models \mathbf{F} \text{ ssi } R \models X \to Y, \text{ pour tout } X \to Y \in \mathbf{F},$$
$$\mathbf{F} \models X \to Y \text{ ssi } X \to Y \in \mathbf{F}^+$$

Clairement, si $R \models \mathbf{F}$ et $\mathbf{F} \models X \to Y$, alors $R \models X \to Y$.

Deux ensembles $\mathbf{F}_0$ et $\mathbf{F}_1$ de dependances fonctionnelles sont équivalents (on note cela par $\mathbf{F}_0 \sim \mathbf{F}_1$) si $\mathbf{F}_0^+ = \mathbf{F}_1^+$.

Décomposition sans perte et DF

Plusieurs théorèmes :

- Si $R \models X \rightarrow Y$, alors

$$R = R_1 \bowtie R_2 \quad \text{où}$$
$$R_1 = \Pi_{X,Y}(R), \qquad R_2 = \Pi_{X,Z}(R),$$

  avec $X \cup Y \cup Z$ tous les attributs de R.

- Soient R une relation d'attributs $X \cup Y$ et $\mathbf{F}$ une collection de FDs tels que $R \models \mathbf{F}$.Si
  - $(X \cap Y) \rightarrow (X \setminus Y) \in \mathbf{F}^+$, ou
  - $(X \cap Y) \rightarrow (Y \setminus X) \in \mathbf{F}^+$,

  alors la decomposition $\Pi_X(R), \Pi_Y(R)$ est sans perte :

$$R = \Pi_X(R) \bowtie \Pi_Y(R).$$

DF élémentaires

Soit **F** un ensemble de DFs.

Définition. Une dépendance fonctionnelle élémentaire (de **F**) est une DF de la forme

$$X \rightarrow \{a\}$$

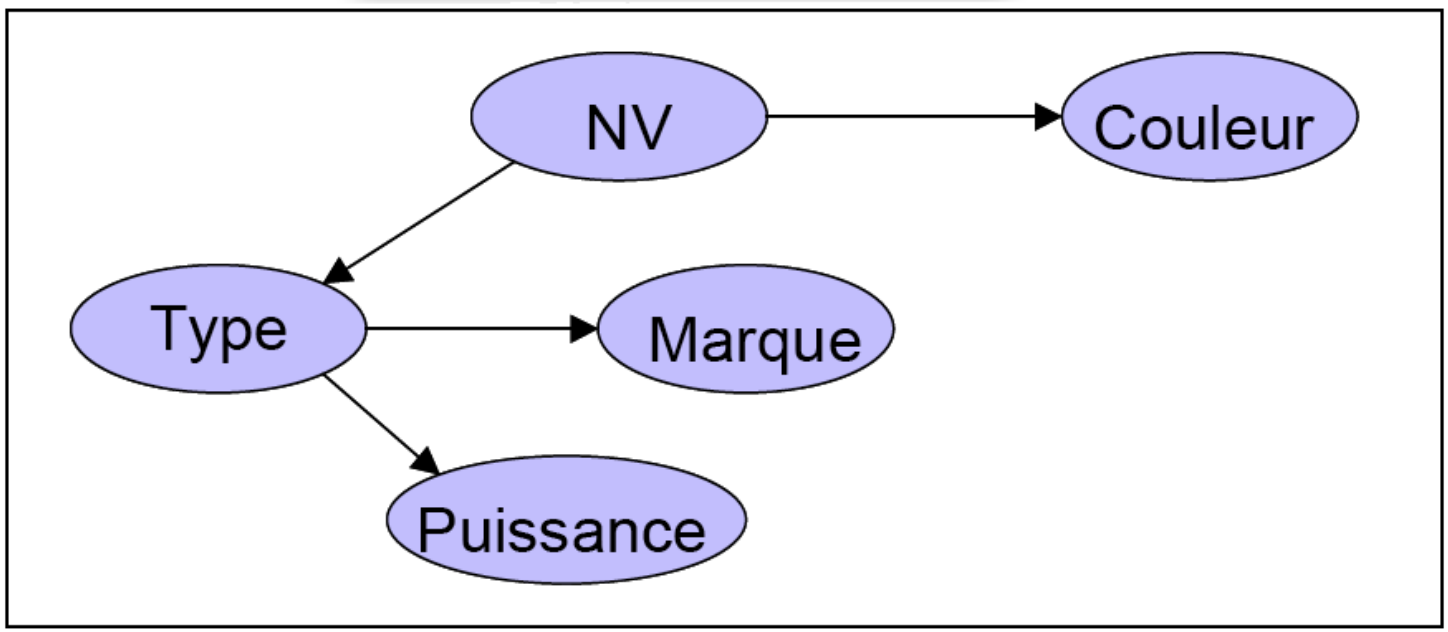
telle que

- $a \notin X$,
- $X' \subseteq X$ et $\mathbf{F} \models X' \rightarrow \{a\}$ implique $X' = X$.

- Exemple : *Type* $\rightarrow$ *Puissance*
- Contre-exemple : (*Type*, *Marque*) $\rightarrow$ *Puissance*
- Dépendances fonctionnelles pour amélioration de schéma : uniquement élémentaires.

Graphe de DF élémentaires

- Grâce à ce graphe la transitivité entre dépendance est facilement lisible.
- Exemple où tout dépendance $X \rightarrow a$ est telle que X singleton) :



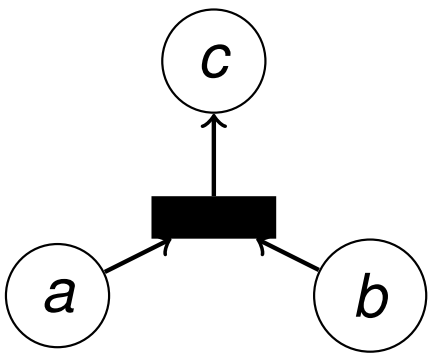
Graphe de DF élémentaires (II)

Si, dans une dépendance élémentaire $X \to a$, X n'est pas un singleton, on utilise un "graphisme" style réseaux de Pétri.

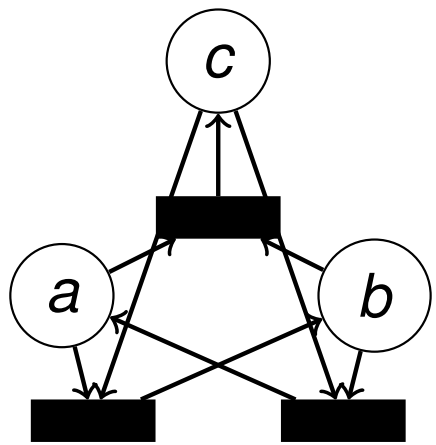
Exemple :

$$ab \to c, \quad ac \to b, \quad bc \to a.$$

On représente $ab \to c$ par :



On représente les trois DFs par :



Couverture minimale

Définition. Soit **F** un ensemble de DFs. Une couverture minimale de **F** est un ensemble **C** de DFs tel que

- $\mathbf{C} \sim \mathbf{F}$,
- tout $X \rightarrow Y \in \mathbf{C}$ est élémentaire,
- pour tout $X \rightarrow Y \in \mathbf{C}$, $\mathbf{C} \setminus \{X \rightarrow Y\} \not\sim \mathbf{F}$.

Algorithme de calcul de la couverture minimale

Entrée : un ensemble **F** de DFs
Sortie : une couverture minimale **C** de **F**

C := **F**

Pour chaque $X \rightarrow \{a_1, ..., a_n\} \in \mathbf{C}$ avec $n \geq 2$:
..... **C** := **C** $\setminus \{X \rightarrow \{a_1, ..., a_n\}\} \cup \{X \rightarrow a_1, X \rightarrow a_2, \ldots, X \rightarrow a_n\}$

Pour chaque $X \rightarrow a \in \mathbf{C}$, pour chaque $b \in X$:
..... soit **D** := **C** $\setminus \{X \rightarrow a\} \cup \{X \setminus \{b\} \rightarrow a\}$
..... si **C** $\sim$ **D** :
..... **C** := **D**

Pour chaque $X \rightarrow a \in \mathbf{C}$:
..... soit **D** := $C \setminus \{X \rightarrow a\}$
..... si **C** $\sim$ **D** :
..... **C** := **D**

Algorithme : exemple

$$\mathbf{F} = ab \to c, \quad c \to a, \quad bc \to d,$$
$$acd \to b \quad d \to eg, \quad be \to c$$
$$cg \to bd, \quad ce \to ag$$

Etape 1 :

$$\mathbf{C} = ab \to c, \quad c \to a, \quad bc \to d,$$
$$acd \to b, \quad \begin{matrix} d \to e, \\ d \to g, \end{matrix} \quad be \to c,$$
$$\begin{matrix} cg \to b, \\ cg \to d, \end{matrix} \quad \begin{matrix} ce \to a, \\ ce \to g. \end{matrix}$$

Algorithme : exemple

Etape 1

$$\mathbf{C} = ab \to c, \quad c \to a, \quad bc \to d,$$
$$acd \to b, \quad \begin{matrix} d \to e, \\ d \to g, \end{matrix} \quad be \to c,$$
$$\begin{matrix} cg \to b, \\ cg \to d, \end{matrix} \quad \begin{matrix} ce \to a, \\ ce \to g. \end{matrix}$$

Au final :

$$\mathbf{C} = ab \to c, \quad c \to a, \quad bc \to d,$$
$$ce \to g, \quad cg \to d, \quad d \to e,$$
$$d \to g, \quad be \to c, \quad cg \to b$$

DFs et clés

- Clé primaire : intuitif jusqu'à maintenant.
- Une clé d'une relation $R(a_1, ..., a_n)$ est un sous-ensemble X des attributs de la relation R tel que les deux conditions ci-dessous sont réunies :
  1. $X \rightarrow \{a_1, ..., a_n\}$,
  2. il n'existe pas de $Y \subseteq X$, $Y \neq X$, tel que $Y \rightarrow \{a_1, ..., a_n\}$
- Moins formellement : une clé est un ensemble minimal d'attributs qui détermine tous les autres.
- Clé candidate (à être primaire) : toute clé.

Plan

Formes normales

- Permettent de faciliter (et guider) la décomposition de relation sans perte d'information.
  La conception est
  - précise,
  - cohérente,
  - sans redondance.
- Plusieurs formes normales :
  - les trois premières :-) pour la décomposition sans perte
  - la forme normal de Boyce-Codd
  - d'autres...

La première forme normale

Définition. Un schéma de relation est en première forme normale si

- tout attribut contient une valeur atomique (non composé).

C'est-à-dire : pas d'attribut multi-valué.

- Autre formulation : il n'y a pas dans la relation d'attribut dont la valeur est, par exemple, une liste/ensemble de valeurs.
- Or, être d'atomique dépend du domaine !
  Exemple : chaînes de caractères.
- Facile à mettre en oeuvre : transformer toute relation $R(K, A)$ où A prend n valeurs en n relations $R_i(K, A)$.

Première forme normale, exemple

Personne(nom, prenoms)

$\rightsquigarrow$

Personne1(nom, prenom1)
Personne2(nom, prenom2)

Ou

$\rightsquigarrow$

Personne(nom)
Prenom(nom, prenom)

Deuxième forme normale

Définition. Un schéma de relation est en deuxième forme normale si

- il est en première forme normale,
- tout attribut n'appartenant pas à une clé ne dépend pas d'une partie (propre) de la clé.

- Élimine les redondances.

Deuxième forme normale, contre-exemple et exemple

Contre-exemple :

Livraison(<u>ref_client, ref_article</u>, quantite, adresse_livraison)

ref_client $\rightarrow$ adresse_livraison

Pas en 2NF, car :

$$\{\text{ref\_client}\} \subsetneq \{\text{ref\_client, ref\_article}\}.$$

Exemple :

Casting(<u>ref_film, ref_acteur</u>, personnage, cachet)

Deuxième forme normale (suite)

- Transformation en DFN :
  - R(K1,K2,X,Y) avec K1 $\rightarrow$ X :
  - Décomposition en 2 relations $R1(\underline{K1,K2}, Y)$ et $R2(\underline{K1}, X)$
- Contre-exemple précédent :

> Livraison(ref_client, ref_article, quantite, adresse_livraison)
> ref_client $\rightarrow$ adresse_livraison
>
> $\rightsquigarrow$
> Livraison1(ref_client, ref_article, quantite)
> Livraison2(ref_client, adresse_livraison)

- Souvent : changer les noms.
- Rejoint les critères de normalisation des attributs d'association de Merise.

Troisième forme normale

Définition. Un schéma de relation est en troisième forme normale si :

1. il est en deuxième forme normale
2. tout attribut n'appartenant pas à une clé ne dépend pas d'un autre attribut non-clé.

- Élimine les redondances liées aux dépendances fonctionnelles transitives.
- Doit être vérifiée pour chaque clé.
- Attention : 1. est redondante, car 2. implique 1.

(Contre-)Exemple

Contre-exemple :

Voiture(<u>nv</u>, marque, type, puissance, couleur)
type $\rightarrow$ marque, puissance

est en 2FN :

- nv est la seule clé et elle est un singleton,

mais pas en 3FN car

- type $\rightarrow$ marque et type $\notin \{\text{nv}\}$.

Exemple :

$\rightsquigarrow$
Voiture(<u>nv</u>, type, couleur)
Modele(<u>type</u>, marque, puissance)

est en 3FN.

Préservation des DFs

DF(R) : DFs associées au schéma de R.

Définition. Une décomposition $(R_1, R_2 \ldots R_n)$ d'un schéma de relation R préserve les dépendances fonctionnelles si

$$\mathbf{DF}(R)^+ = (\bigcup_{i=1,\ldots,n} \mathbf{DF}(R_i))^+ .$$

C'est-à-dire : la fermeture de l'union des **DF**(R_i) est et la même que celle dee **DF**(R).

- Toute relation a au moins une décomposition en 3FN telle que :
  - préservation des dépendances fonctionnelles
  - décomposition sans perte.

Algorithme de décomposition en 3FN

Décomposition($R(a_1, \ldots, a_n), \mathbf{F}$) =

1. Rechercher une couverture minimale **C** de **F**.
2. Partitionner **C** en groupes de dépendances $\mathbf{C}_i$ ayant la même partie gauche.
   Chaque $\mathbf{C}_i$ est de la forme $X_i \rightarrow Y_i$.
3. Fusionner les groupes $\mathbf{C}_i = X_i \rightarrow Y_i$ et $\mathbf{C}_j = X_j \rightarrow Y_j$ si $Y_i = X_j$ et $Y_j = X_i$.
4. Associer à chaque groupe $\mathbf{C}_i = X_i \rightarrow Y_i$ un schéma de relation R_i dont la clé est X_i et les constituants non clés est Y_i.
5. Si aucune des relations précédentes ne contient une clé candidate K de R, créer une relation supplémentaire K qui la contient.

Forme normale de Boyce-Codd

Définition. Un schéma de relation est en BCFN si pour toute dépendance $K \rightarrow a$ avec $a \notin K$, K est une clé.

- Toute relation a une décomposition en FNBC.
- Cette décomposition ne préserve pas en général les dépendances fonctionnelles.
- Idée :
  ne pas représenter la même information plusieurs fois.

(Contre-)Exemples

- Contre-exemple :

  Movie(title, director, actor)
  title $\rightarrow$ director

  n'est pas en FNBC. Car title n'est pas une clé.

  Anomalies de mise à jour de la colonne directeur.

- La schéma Movie peut se décomposer en

  Movie_director(title,director)
  title $\rightarrow$ director

  Movie_actor(title,actor)