Bases de Données - Décomposition

Michel Mainguenaud
Département Architecture des Systèmes d'Information
Institut National des Sciences Appliquées - Rouen
michel.mainguenaud@insa-rouen.fr

Architecture des Systèmes d'Information

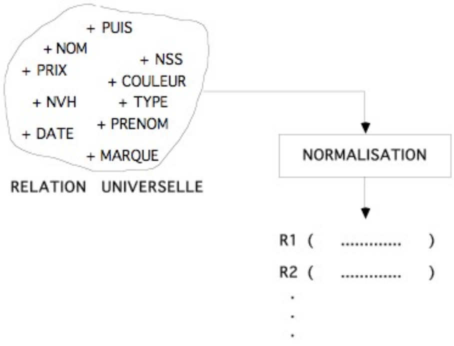
Plan. . .

Objectifs

- Conception du schéma
  - Sans redondance d'information
  - Sans anomalie en mise à jour
- Principes de base
  - Dépendances fonctionnelles/multivaluées (contraintes)
  - Formes normales
- Mise en oeuvre
  - Algorithme de décomposition

Principes

- Relation universelle
  - Sous-ensemble du produit cartésien de la totalité des attributs de la base
  - Décomposition de cette relation universelle pour obtenir des relations normalisées



Dépendances Fonctionnelles - DF

- Définition
  - Un schéma de relation R (A1, ..., An)
  - X et Y des sous-ensembles disjoints de {A1, ..., An}
- Propriété
  - X détermine Y ou Y dépend fonctionnellement de X, si quelle que soit la réalisation r de R et pour tout n-uplet t1 et t2 de r on a
  - π (X, {t1}) = π (X, {t2}) $\Rightarrow$ π (Y, {t1}) = π (Y, {t2})
- Notation
  - X $\rightarrow$ Y

Exemples

Par abus de langage, très souvent on fait "sauter" la notion d'ensemble pour ne mettre que les attributs.

- Personne
  - NSS $\rightarrow$? Nom
  - Nom $\rightarrow$? NSS
- Voiture
  - Marque, Type $\rightarrow$? Puissance
  - Marque $\rightarrow$? Puissance
  - Puissance $\rightarrow$? Type
- Possession
  - Numéro de véhicule $\rightarrow$? Numéro de propriétaire
  - Numéro de propriétaire $\rightarrow$? Numéro de véhicule
  - Numéro de véhicule, numéro de propriétaire $\rightarrow$? Date d'achat

A partir de maintenant pour le reste du cours : Type $\rightarrow$ Puissance

asi Architecture des Systèmes d'Information

Remarques

- Une DF est une assertion qui est définie sur toutes les réalisations possibles d'une relation (et non sur une réalisation particulière)
- Une DF traduit une contrainte sur les données qui doit être vérifiée en permanence
- Objectif : faire faire ce travail de contrôle par le SGBD au lieu d'un programme et si possible sans consommer de ressource (intrinsèque au modèle).

Interprétation des DF

- NVH $\rightarrow$ NPRO signifie l'interdiction de co-propriété d'une voiture
- Les DF font partie du schéma d'une base de données
  - Permettent de construire le schéma
  - En cas de perte, le DBA devra mettre en place la déclaration par contrainte d'intégrité
  - Sous-traitance du travail de contrôle au SGBD plutôt qu'à un programme d'application.

Propriétés des DF - Axiomes d'Armstrong

Objectif : se ramener à un ensemble minimal de DF afin de rendre l'ensemble manipulable. Il faut s'assurer que cet ensemble minimal corresponde bien à l'ensemble initial en terme de pouvoir de représentation (fermeture identique).

- Réflexivité
  - Tout ensemble d'attributs détermine lui-même ou une partie de lui-même
  - $X \supseteq Y \Rightarrow X \rightarrow Y$
- Augmentation
  - On peut enrichir par un même ensemble d'attributs deux ensembles qui sont en DF
  - $X \rightarrow Y \Rightarrow XZ \rightarrow YZ$
- Transitivité
  - $X \rightarrow Y \wedge Y \rightarrow Z \Rightarrow X \rightarrow Z$

Architecture des Systèmes d'Information

Exemple

A partir d'une ensemble de DF il est possible d'en déduire d'autres par la simple application des axiomes d'Armstrong.

- F : un ensemble de DF
  - F = { NVH $\rightarrow$ Type, Type $\rightarrow$ Marque, Type $\rightarrow$ Puissance, NVH $\rightarrow$ Couleur }
- Nouvelles constructions
  - {NVH, Marque} $\supseteq$ {Marque} $\Rightarrow$ {NVH, Marque} $\rightarrow$ {Marque}
  - NVH $\rightarrow$ Type $\Rightarrow$ NVH, Puissance $\rightarrow$ Type, Puissance
  - NVH $\rightarrow$ Type $\wedge$ Type $\rightarrow$ Marque $\Rightarrow$ NVH $\rightarrow$ Marque

Propriétés déduites

- Union
  - $X \rightarrow Y \wedge X \rightarrow Z \Rightarrow X \rightarrow YZ$
- Pseudo-transitivité
  - $X \rightarrow Y \wedge YW \rightarrow Z \Rightarrow XW \rightarrow Z$
- Décomposition
  - $X \rightarrow Y \wedge Y \supseteq Z \Rightarrow X \rightarrow Z$

Architecture des Systèmes d'Information

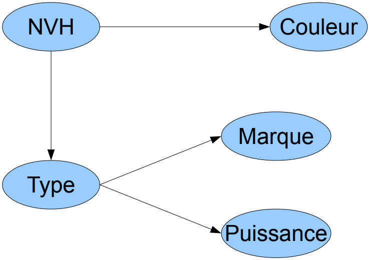
DF élémentaire

- Une DF élémentaire est une DF de la forme X $\rightarrow$ A ou
  - A $\not\subseteq$ X
  - $\nexists$ X' $\subseteq$ X / X' $\rightarrow$ A
- Exemple
  - NVH, Puissance $\rightarrow$ Type, Puissance n'est pas une DF élémentaire car NVH $\rightarrow$ TYPE
- Objectif : Ne conserver que les DF utiles et donc rendre l'ensemble des DF manipulables (utilisation de la notion de graphes)

Architecture des Systèmes d'Information

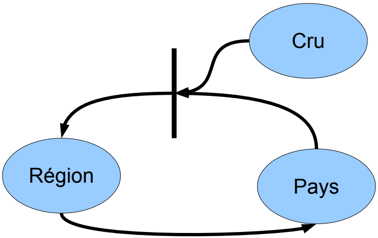
(Hyper-)Graphe des DF - 1

- Relation : VOITURE (NVH, Marque, Type, Puissance, Couleur)
- DF : F = { NVH $\rightarrow$ Type, Type $\rightarrow$ Marque, Type $\rightarrow$ Puissance, NVH $\rightarrow$ Couleur }
- Représentation
  - Graphe orienté : un noeud = un attribut, un arc = une DF



(Hyper-)Graphe des DF - 2

- Relation : VIN (CRU, PAYS, REGION)
- DF : F = { Pays, Cru $\rightarrow$ Region, Region $\rightarrow$ Pays,}
- Représentation
  - Hyper-Graphe orienté : un noeud = un attribut, une famille = un ensemble de noeud, un hyper-arc = un couple de famille



Fermeture transitive

On appelle Fermeture Transitive d'un ensemble de DF, F, l'ensemble F+ qui est l'union de F et de l'ensemble des DF déduites par transitivité.

- DF : F = { NVH $\rightarrow$ Type, Type $\rightarrow$ Marque, Type $\rightarrow$ Puissance, NVH $\rightarrow$ Couleur }
- F+ = F $\cup$ { NVH $\rightarrow$ Marque, NVH $\rightarrow$ Puissance }
- Objectif : Connaitre les attributs qui dépendent fonctionnellement d'un groupe d'attributs

Architecture des Systèmes d'Information

Algorithme (naïf) de fermeture transitive

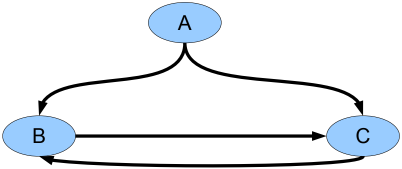
- ```
  Resultat = X
  Tant que (resultat varie)
    faire
      Pour chaque DF (Y -> Z)
         faire
            Si Y est inclus dans Resultat
            Alors
               Resultat = Resultat U Z
            Finsi
      Finpour
   Fintantque
  ```
  ```

Couverture minimale

- La couverture minimale d'un ensemble de DF, F, est un sous-ensemble minimum, FCM, de DF élémentaires permettant de générer toutes les autres (même pouvoir d'expression - même fermeture transitive)
- Tout ensemble de DF admet une couverture minimale, en général non unique
- Objectif : se restreindre à cet ensemble minimal pour concevoir le schéma (logique) de la base de données. Départ : Relation universelle et décomposition selon les DF retenues.

Exemple

- Ensemble de DF
  - F = { NVH $\rightarrow$ Type, Type $\rightarrow$ Marque, Type $\rightarrow$ Puissance, NVH $\rightarrow$ Couleur, NVH $\rightarrow$ Marque, NVH $\rightarrow$ Puissance }
  - FCM (F) = { NVH $\rightarrow$ Type, Type $\rightarrow$ Marque, Type $\rightarrow$ Puissance, NVH $\rightarrow$ Couleur }
- Non-unicité
  - FCM = { A $\rightarrow$ B, B $\rightarrow$ C, C $\rightarrow$ B }
  - FCM = { A $\rightarrow$ C, C $\rightarrow$ B, B $\rightarrow$ C }



Clé et DF

- Une clé est un (groupe minimal d') attribut d'une relation qui permet d'identifier un tuple et un seul dans une relation.
  - R (A1, ..., An)
  - $X \subseteq \{A1, ..., An\}$
  - Respect de la fermeture transitive des DF (F+)
- X clé de R $\Leftrightarrow X \rightarrow A1, \ldots An \wedge \nexists Y \subset X \,/\, Y \rightarrow A1, \ldots, An$
- Une clé est un ensemble minimum d'attributs d'une relation qui détermine tous les autres

Exemple

- F = { NVH $\rightarrow$ Type, Type $\rightarrow$ Marque, Type $\rightarrow$ Puissance, NVH $\rightarrow$ Couleur, NVH $\rightarrow$ Marque, NVH $\rightarrow$ Puissance }
- VOITURE (NVH, TYPE, MARQUE, PUISSANCE, COULEUR)
  - {NVH, TYPE} : clé ?
  - NVH : clé ?
  - TYPE : clé ?
- Si plusieurs clés sont candidates pour une même relation, l'administrateur en choisit une qui devient la clé primaire.

Décomposition

- La décomposition d'un schéma d'une relation (initialement universelle) R (A1, ..., An) est son remplacement par une collection de schéma de relations {R1, ..., RN } /
  - Schéma (R) = {A1, ..., AN}
  - Schéma (R) = Schéma (R1) $\cup$ ... $\cup$ Schéma (Rn)
- Objectif : Casser R en plus petites relations afin d'éliminer
  - Les redondances de données (place)
  - Les anomalies en mise à jour

Exemple - 1

- FCM (F) = { NVH $\rightarrow$ Type, Type $\rightarrow$ Marque, Type $\rightarrow$ Puissance, NVH $\rightarrow$ Couleur }
- VOITURE (NVH, Type, Marque, Puissance, Couleur)
- Décomposition (1) ?
  - R1 (NVH, Type)
  - R2 (Type, Marque, Puissance)
  - R3 (NVH, Couleur)
- Décomposition (2) ?
  - R'1 (NVH, Type)
  - R'2 (Type, Puissance, Couleur)
  - R'3 (Type, Marque)
- Décomposition (3) ?
  - R"1 (NVH, Type, Couleur)
  - R"2 (Type, Marque, Puissance)

Exemple

Un produit a un type et un client a une remise pour un type de produit

- R (Produit, Type, Client, Remise)
- Décomposition (1) ?
  - R1 (Produit, Type)
  - R2 (Type, Client, Remise)
- Décomposition (2) ?
  - R'1 (Type, Client, Remise)
  - R'2 (Type, Client, Produit)
- Décomposition (3) ?
  - R"1 (Produit, Type)
  - R"2 (Produit, Client, Remise)

Décomposition Sans Perte d'Information (SPI)

- Une décomposition d'une relation R en N relations R1, ... RN est sans perte ssi pour toute réalisation r de R et pour toutes les réalisation r1, ..., rN de R1, ..., RN, on a :
  - $r = r1 \bowtie r2 \bowtie \ldots \bowtie rN$
- Conséquences : Génération d'informations erronées lors de jointure ou absence d'accès à l'information
- Exemple
  - VOITURE : (R"1, R"2) sans perte
  - VOITURE : (R'1, R'2, R'3) avec perte

Architecture des Systèmes d'Information

Exemple de perte

VOITURE	NVH	MARQUE	TYPE	PUISSANCE	COULEUR
	872 RH 75	Renault	VelSatis	9	Rouge
	975 AB 80	Renault	VelSatis	9	Beige

- VOITURE $=$ R"1 $\bowtie$ R"2 (Pas de problème)

R"1	NVH	TYPE	COULEUR
	872 RH 75	VelSatis	Rouge
	975 AB 80	VelSatis	Beige

R"2	TYPE	MARQUE	PUISSANCE
	VelSatis	Renault	9

Une relation pour les instances, une pour les informations génériques

Exemple de perte - 2

VOITURE	NVH	MARQUE	TYPE	PUISSANCE	COULEUR
	872 RH 75	Renault	VelSatis	9	Rouge
	975 AB 80	Renault	VelSatis	9	Beige

- VOITURE $=$ R'1 $\bowtie$ R'2 $\bowtie$ R'3 (Problème)

R'1	NVH	TYPE
	872 RH 75	VelSatis
	975 AB 80	VelSatis

R'2	TYPE	PUISSANCE	COULEUR
	VelSatis	9	Rouge
	VelSatis	9	Beige

R'3	TYPE	MARQUE
	VelSatis	Renault

Exemple - 3

Erreur classique : se baser sur des exemples pour définir les DF

Stock	PRODUIT	TYPE	CLIENT	REMISE
	Aspirine	A	C1	3
	Compresse	A	C2	5
	Vitamine	B	C1	7
	Sirop	B	C2	6

- Décomposition ?
  - R1 (Produit, Type)
  - R2 (Type, Client, Remise)

Exemple - 4

R1	PRODUIT	TYPE
	Aspirine	A
	Compresse	A
	Vitamine	B
	Sirop	B

R2	TYPE	CLIENT	REMISE
	A	C1	3
	A	C2	5
	B	C1	7
	B	C2	6

- Problème : $<$ Vitamine, B, C2, 6 $>$ appartient à R1 $\bowtie$ R2 mais pas à Stock

Décomposition préservant les DF

- Une décomposition (R1, ..., RN) de R préserve les DF si la fermeture des DF de R est la même que celle de l'union des DFs préservées des relations R1, ..., RN
- Exemple
  - R (NVH, Type, Marque, Puissance, Couleur)
    - $F_R = \{$ NVH $\rightarrow$ Type, Type $\rightarrow$ Marque, Type $\rightarrow$ Puissance, NVH $\rightarrow$ Couleur $\}$
    - $F_R+ = F_R \cup \{$ NVH $\rightarrow$ Marque, NVH $\rightarrow$ Puissance $\}$
  - R1 (NVH, Type, Couleur)
    - $F_{R1} = \{$ NVH $\rightarrow$ Type, NVH $\rightarrow$ Couleur $\}$
  - R2 (Type, Marque, Puissance)
    - $F_{R2} = \{$ Type $\rightarrow$ Marque, Type $\rightarrow$ Puissance$\}$
    - $(F_{R1} \cup F_{R2})+ = (\{$ NVH $\rightarrow$ Type, NVH $\rightarrow$ Couleur, Type $\rightarrow$ Marque, Type $\rightarrow$ Puissance $\})+$

Exemple

- R (NVH, Type, Marque, Puissance, Couleur)
  - $F_R = \{$ NVH $\rightarrow$ Type, Type $\rightarrow$ Marque, Type $\rightarrow$ Puissance, NVH $\rightarrow$ Couleur $\}$
  - $F_R+ = F_R \cup \{$ NVH $\rightarrow$ Marque, NVH $\rightarrow$ Puissance $\}$
- R'1 (NVH, Type)
  - F'1 $= \{$ NVH $\rightarrow$ Type$\}$
- R'2 (Type, Puissance, Couleur)
  - F'2 $\{$ Type $\rightarrow$ Puissance $\}$
- R'3 (Type, Marque)
  - F'3 $\{$ Type $\rightarrow$ Marque$\}$

(F'1 $\cup$ F'2 $\cup$ F'3)$+ \neq F_R+$

On a perdu la DF : NVH $\rightarrow$ Couleur

Préservation des DF $\not\Rightarrow$ Sans perte

- R (Produit, Type, Client, Remise)
  - F = { Produit → Type, (Type, Client) → Remise}
- R1 (Produit, Type)
  - $F_{R1} = F_{R1} + = \{$Produit → Type $\}$
- R2 (Type, Client, Remise)
  - $F_{R2} = F_{R2} + = \{$(Type, Client) → Remise $\}$

$(F_{R1} \cup F_{R2})+ = F_R+$
Préserve les DF mais n'est pas sans perte.
La vraie DF est (Produit, Client) → Remise

Objectif de la décomposition

- Obtenir une décomposition
  - Préserve les DF (sinon on perd des contraintes d'intégrité)
  - Sans Perte d'Information -SPI- (sinon erreurs possibles lors de jointure)
- On peut toujours décomposer une relation suivante une DF
- On ne peut pas décomposer une relation s'il n'y a pas de DF
- La décomposition suivant une DF ne perd pas d'information
  - $R(X, Y, Z) \wedge X \rightarrow Y \Rightarrow R1(X, Y) \wedge R2(X, Z)$ correcte