# Conception d'une base de données

- Qu'est ce qu'un schéma de bases de données ?
  - une ou plusieurs relations
  - nombreux schémas possibles
- Qu'est ce qu'un bon schéma (« formes normales ») ?
  - ni perte d'information ni redondance
  - contraintes (d'intégrité) entre les valeurs des attributs
  - le but: indépendance / applications (vues particulières)
- Deux approches
  - on génère un schéma, on vérifie qu'il est bon, on le corrige s'il y a lieu
  - on met au point une méthode  pour générer uniquement de bons schémas



# Exemple  « Marché des vins »

| REGION | NOMP | RECOLTE | CRU | ACHAT | NOMB |
|---|---|---|---|---|---|
| Beaujolais | Trichard | 650 | Arbois | 12 | Gide |
| Bourgogne | de Launay | 69 | Arbois | 22 | Cocteau |
| Jura | Metral | 675 | Arbois | 12 | Gide |
| Jura | Reverchon | 850 | Arbois | 12 | Gide |
| Rhone | Berard | 100 | Beaujolais Villages | 1 | Ajar |
| Bourgogne | Dubreuil Fontaine | 350 | Bourgogne Passe Tout Grains | 80 | Fromentin |
| Bourgogne | Genelot | 500 | Brouilly | 5 | Audiberti |
| Rhone | Dumazet | 180 | Chateau Chalon | 2 | Montaigne |
| Bourgogne | Menand | 70 | Chateau Corton Grancey | 6 | Dante |
| Bourgogne | Menand | 70 | Chateau Corton Grancey | 12 | Gide |
| Alsace | Boxler | 100 | Chenas | 4 | Apollinaire |
| Alsace | Boxler | 100 | Chenas | 12 | Barthes |
| Alsace | Boxler | 90 | Chiroubles | 1 | Aristote |
| Beaujolais | Descombes | 912 | Chiroubles | 1 | Aristote |
| Jura | Chalandard | 98 | Chiroubles | 1 | Aristote |
| Jura | Michlel | 540 | Chiroubles | 1 | Aristote |
| Rhone | Dumazet | 450 | Chiroubles | 1 | Aristote |
| Jura | Maire | 25 | Clairette de Die | 10 | Woolf |
| Bourgogne | Lasnier | 986 | Clos du roi | 12 | Moliere |
| Bourgogne | Menand | 100 | Cornas | 23 | Corneille |
| Bourgogne | Menand | 100 | Cornas | 12 | Gide |

```
select region,nomp,sum(r.qte) recolte,cru,sum(a.qte) achat,nomb
from p,r,v,a,b
where v.nv=r.nv and r.np=p.np
and v.nv=a.nv and a.nb=b.nb
group by cru, region, nomp, nomb
```

## Exemple « Parc de véhicules »
### *relation unique (« universelle »)*

| NSS | NOM | PNOM | NVH | DATE | PRIX | TYPE | PUISS | MARQUE | COULEUR |
|---|---|---|---|---|---|---|---|---|---|
| 2 60… | Martin | Paule | 1RT56 | 1/2/93 | 10kF | R21 | 7 | Renault | grise |
| 1 61… | Roffé | Jean | 1RT56 | 1/2/93 | 10kF | R21 | 7 | Renault | grise |
| 1 57… | Allain | Arthur | 1RT56 | 1/2/93 | 10kF | R21 | 7 | Renault | grise |
| 1 65… | Ellul | Paul | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
| NULL | NULL | NULL | 4TR92 | 1/2/93 | 50kF | R21 | 7 | Renault | verte |
| 2 60… | Martin | Paule | 25L56 | 1/10/93 | 6kF | AX | 6 | Citroën | jaune |
| 2 70… | Moris | Marie | 1GF75 | 23/3/93 | 95kF | XM | 9 | Renault | blanche |

Mises à jour :

- Insérer une personne qui n'a pas de voiture
- Modifier le prénom du propriétaire de la voiture N° 25L56
- Supprimer la voiture N° 1GF75

2001/02/22 saglio@enst.fr 3

## Quels problèmes liés à la redondance ?

**Redondance des données**

(NSS, NOM, PNOM) d'un propriétaire de plusieurs voitures

**Incohérence en mise à jour**

Modification du prénom dans un tuple et pas dans l'autre

**Anomalies en insertion**

Personne sans voiture

Si valeurs « null » admises, comment faire une restriction sur marque ?

**Anomalies en suppression**

La suppression de la dernière voiture d'une personne doit elle entraîner la suppression de la personne ?

2001/02/22 saglio@enst.fr 4

## Exemple « parc de véhicules »
### *décomposition en 3 relations*

PERSONNE

| NSS | NOM | PNOM |
|---|---|---|
| 2 60… | Martin | Paule |
| 1 61… | Roffé | Jean |
| 1 57… | Allain | Arthur |
| 1 65… | Ellul | Paul |
| 2 70… | Moris | Marie |

VEHICULE

| NVH | TYPE | PUISS | MARQUE | COULEUR |
|---|---|---|---|---|
| 1RT56 | R21 | 7 | Renault | grise |
| 4TR92 | R21 | 7 | Renault | verte |
| 25L56 | AX | 6 | Citroen | jaune |
| 1GF75 | XM | 9 | Renault | blanche |

POSSEDE

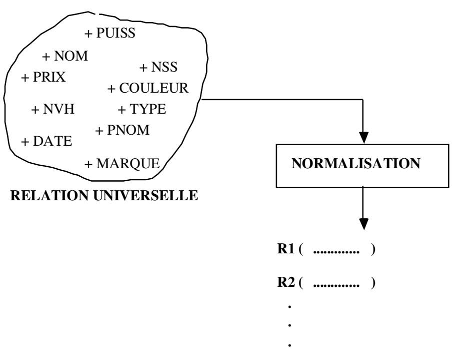
| NSS | NVH | DATE | PRIX |
|---|---|---|---|
| 2 60… | 1RT56 | 1/2/93 | 10kF |
| 1 61… | 1RT56 | 1/2/93 | 10kF |
| 1 57… | 1RT56 | 1/2/93 | 10kF |
| 2 60… | 25L56 | 1/10/93 | 6kF |
| 2 70… | 1GF75 | 23/3/93 | 95kF |

2001/02/22 saglio@enst.fr 5

## Principe de la normalisation

RELATION UNIVERSELLE

Sous-ensemble du produit cartésien de la totalité des attributs considérés dans la base




A partir de la RELATION UNIVERSELLE, le PROCESSUS DE NORMALISATION consiste à casser (DECOMPOSER) progressivement LES RELATIONS jusqu'à obtenir des RELATIONS NORMALISEES

2001/02/22 saglio@enst.fr 6

## Théorie de la normalisation

Théorie destinée à **concevoir correctement le schéma d'une base relationnelle**, c'est à dire :

- SANS REDONDANCE D'INFORMATIONS
- SANS ANOMALIES EN MISE A JOUR

CONCEPTS DE CETTE THEORIE :

Les **dépendances fonctionnelles** qui traduisent des contraintes sur les données

Les **Formes normales** qui définissent des relations "bien" conçues

mise en œuvre de la THEORIE au moyen D'ALGORITHMES (DECOMPOSITION,...)

**2001/02/22** **saglio@enst.fr** **7**

## Exemples de Dépendances Fonctionnelles (DFs)

Relation PERSONNE

NSS -> NOM ?
(NOM, PNOM) -> NSS ?

relation VOITURE

(MARQUE, TYPE) -> PUIS ?
MARQUE -> PUIS ?
PUIS -> TYPE ?

relation POSSEDE

NVH -> NSS ?
NSS -> NVH ?
(NVH, NSS) -> DATE ?

**2001/02/22** **saglio@enst.fr** **8**

## Les dépendances fonctionnelles

**… Ou comment spécifier la redondance ?**

**Définition**

Soient :

-> R(A1, A2, ..., An) un **schéma** de relation

-> X et Y des sous-ensembles d'attributs de {A1, A2, ..., An}

On dit que :

X **détermine** Y, ou Y **dépend fonctionnellement** de X, si, quelle que soit une instance I de R et pour tous tuples t1 et t2 de I, on a:

$$\Pi_X (t1) = \Pi_X (t2) \quad => \quad \Pi_Y (t1) = \Pi_Y (t2)$$

**Notation** d'une **dépendance fonctionnelle** : X -> Y

**2001/02/22** **saglio@enst.fr** **9**

## Remarques

Une **Dépendance Fonctionnelle** est une **assertion** qui est définie sur **toutes les instances possibles d'une relation**, et pas sur une réalisation particulière

Une **Dépendance Fonctionnelle** traduit une certaine **perception de la réalité**; elle correspond à une **contrainte sur les données**, qui doit être **vérifiée en permanence**

| **Exemple :** |
| --- |
| la DF NVH -> NSS signifie L'INTERDICTION de la CO-PROPRIETE d'une voiture. |

Les **Dépendances fonctionnelles** font partie **du schéma d'une BD**; en conséquence, elles doivent être :

- **déclarées par l'administrateur**
- **contrôlées par le SGBD**

**2001/02/22** **saglio@enst.fr** **10**

## Exemple : déduction de nouvelles DFs

Soit l'ensemble initial F de DF :
F = { NVH -> TYPE, TYPE -> MARQUE, TYPE -> PUISS, NVH -> COULEUR }

On déduit

1) (MARQUE) inclus dans (NVH, MARQUE)
   D'OU : (NVH, MARQUE) -> MARQUE

2) NVH -> TYPE
   D'OU : (NVH, PUISS) -> (TYPE, PUISS)

3) NVH -> TYPE et TYPE -> MARQUE
   D'OU : NVH -> MARQUE

2001/02/22 saglio@enst.fr 11

## Axiomes d'Armstrong

**Propriétés de base**

1) REFLEXIVITE
   Y inclus dans X => X -> Y

2) AUGMENTATION
   X -> Y => XZ -> YZ

3) TRANSITIVITE
   X -> Y et Y -> Z => X -> Z

**Propriétés déduites**

1) UNION
   X -> Y
   => X -> YZ
   X -> Z

2) PSEUDO-TRANSITIVITE
   X -> Y => XW -> Z
   YW -> Z

3) DECOMPOSITION
   X -> Y
   => X -> Z
   Z inclus dans Y

2001/02/22 saglio@enst.fr 12

## DF élémentaire

Une **DF élémentaire** est une DF de la forme X ->A, OU :

-> A **n'est pas inclus dans** X

-> **il n'existe pas** X' **inclus dans** X tel que X' -> A

**Exemple**

(NVH, PUISS) -> TYPE

n'est pas une DF élémentaire car on a la DF :

NVH -> TYPE

Cette notion de **DF élémentaires** sert à conserver les seules **DF utiles**

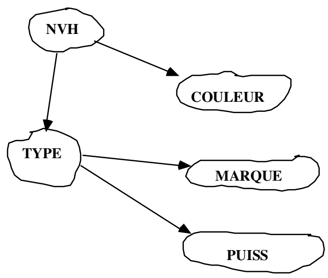
**2001/02/22** **saglio@enst.fr** **13**

## Graphe des DF

Graphe orienté dans lequel :

UN NOEUD = UN ATTRIBUT

UN ARC = UNE DF




**Exemple**

VOITURE (NVH, MARQUE, TYPE, PUISS, COULEUR)

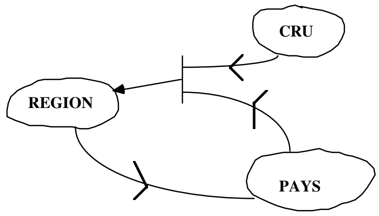
F = {NVH -> TYPE, TYPE -> MARQUE, TYPE -> PUISS, NVH -> COULEUR }

**2001/02/22** **saglio@enst.fr** **14**

## Autre exemple de graphes de DF

SOIT UNE RELATION :

VIN (CRU, PAYS, REGION)




SOIT L'ENSEMBLE DES DF ASSOCIEES :

F = { REGION -> PAYS, (PAYS,CRU) ->REGION }

2001/02/22 saglio@enst.fr 15

## Fermeture transitive

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

**Exemple**

F = {NVH -> TYPE, TYPE -> MARQUE, TYPE -> PUISS, NVH -> COULEUR }

F+ = F $\cup$ { NVH -> MARQUE, NVH -> PUISS }

2001/02/22 saglio@enst.fr 16

## Couverture minimale

**définition**

La **couverture minimale** d'un ensemble F de DF est un <u>sous-ensemble minimum</u> FCM de **DF élémentaires** permettant de **générer toutes les autres**

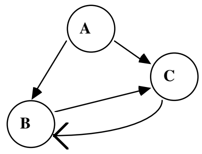
**Théorème**

Tout ensemble de DF admet une couverture minimale, en général non unique

**Exemple**

1) F = {NVH -> TYPE, TYPE -> MARQUE, TYP -> PUISS, NVH -> COULEUR }
constitue une COUVERTURE MINIMALE pour l'ensemble des DFs de VOITURE

2) SOIT LE GRAPHE DES DF :




**FCM = { A->B, B->C, C->B }**

**OU**

**FCM = { A->C, C->B, B->C }**

**2001/02/22** **saglio@enst.fr** **17**

## Couverture minimale

Un ensemble F de DF est une couverture minimale ssi :

1- X->Y élément de F => Y est un attribut simple

2- Pour aucune DF X->A, on a
F- {X->A} équivalent à F
(i.e. aucune DF ne peut être ôtée de F)

3- Si X-> A élément de F alors il n'existe pas Z inclus dans X tel que
F- {X-A} U {Z--> A} équivalent à F
(i.e.les attributs ne peuvent pas être retirés des DF de F)

**2001/02/22** **saglio@enst.fr** **18**

## Clé d'une relation

**Définition**

soient :

R (A1, A2, ..., An) un schéma de relation

F+ l'ensemble des DFs associés

X un sous-ensemble de A1, A2, ... , An

on dit que X est une **clé** de R si et ssi:

X -> A1 A2 ... An

il n'existe pas de sous-ensemble Y inclus dans X tel que Y -> A1 A2 ... An

*(sinon X n'est qu'une « superclé »)*

Une clé est un ensemble minimum d'attributs qui détermine tous les autres

2001/02/22 saglio@enst.fr 19

## Exemple

VOITURE (NVH, TYPE, MARQUE, PUISS, COULEUR)

(NVH, TYPE) = CLE DE VOITURE ?

(NVH) = CLE DE VOITURE ?

(TYPE) = CLE DE VOITURE ?

**Remarque**

Si plusieurs clés sont candidates pour une même relation, alors on en choisit une appelée clé primaire.

2001/02/22 saglio@enst.fr 20

## Fermeture d'un ensemble d'attributs X / un ensemble de DF

**Problème** : vérifier si X-> A est induit par un ensemble F de DFs

X+ = {A / X->A appartient à F}

X(0) <- X
X(i+1) <- X(i) U {Z / V->Z élément de F, V inclus dans X(i)}

X(0) inclus dans X(1) inclus dans X(2) ...X(i) inclus dans X(i+1) ... inclus dans
R est fini donc il existe k tel que X(k) = X(k+1)
=> pour tout j >= k X(j) = X(k)
=> X+ = X(k)

| Pour vérifier que X est clé de R, il suffit de vérifier que X+ = R |
|---|



## Décomposition

**Définition**

La décomposition d'un schéma de relation R(A1, …, An) consiste en son remplacement par une collection de relation (R1,…Rm) tel que

SCHEMA (R) = SCHEMA (R1) U SCHEMA (R2) U ... U SCHEMA (Rm)

**But de la décomposition**

Casser R en plus petites relations afin d'éviter :

- les redondances
- les anomalies en mises à jour

## Exemple

VOITURE (NVH, MARQUE, TYPE, PUISS, COULEUR)

R1 (NVH, TYPE, COULEUR)
R2 (TYPE, MARQUE, PUISS)

R'1 (NVH, TYPE)
R'2 (TYPE, PUISS, COULEUR)
R'3 (TYPE, MARQUE)

*CES DECOMPOSITIONS SONT-ELLES INTERESSANTES ?*

2001/02/22 saglio@enst.fr 23

## Décomposition sans perte

**Définition**

Une décomposition d'une relation R en m relation ($R_1, \ldots R_m$) est sans perte par rapport à un ensemble de DF si et seulement si, quelle que soit I une instance de R, alors

$$I = I_1 \,|X|\, I_2 \,|X|\, \ldots \,|X|\, I_m$$

$$\text{où } I_j = \Pi R_j\,(I)$$

**Exemple**

AVEC LA DECOMPOSITION DE LA RELATION VOITURE:

-> ($R_1$, $R_2$) EST SANS PERTE

-> ($R'_1$, R'2, $R'_3$) N'EST PAS SANS PERTE

**Théorème**

Une décomposition est sans perte si l'attribut de jointure de la décomposition est clé d'au moins une des deux relations

2001/02/22 saglio@enst.fr 24

| VOITURE | NVH | MARQUE | TYPE | PUISS | COULEUR |
|---|---|---|---|---|---|
| | 872RH75 | RENAULT | R21 | 7 | BLEU |
| | 975AB80 | RENAULT | R21 | 7 | BEIGE |

R1
872RH75 R21 BLEUE
975AB80 R21 BEIGE

R2
R21 RENAULT 7

VOITURE = R1 |X| R2

R'1
872RH75 R21
975AB80 R21

R'2
R21 7 BLEUE
R21 7 BEIGE

R'3
R21 RENAULT

VOITURE <> R'1 |X| R'2 |X| R'3

R'1 |X| R'2 |X| R'3

| | | | | |
|---|---|---|---|---|
| 872RH75 | R21 | 7 | BLEUE | RENAULT |
| 872RH75 | R21 | 7 | BEIGE | RENAULT |
| 975AB80 | R21 | 7 | BLEUE | RENAULT |
| 975AB80 | R21 | 7 | BEIGE | RENAULT |



## Décomposition préservant les DFs

**Définition:** Une décomposition (R1, …Rm) 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 des R1, … Rm

**Exemple**

1) R1 (NVH, TYPE, COULEUR)=>F1 = { NVH -> TYPE, NVH -> COULEUR }
   R2 (TYPE, MARQUE, PUISS)=>F2 = { TYPE -> MARQUE, TYPE -> PUISS }
   **==> OK**

2) R'1 (NVH, TYPE) => F'1 = { NVH -> TYPE }
   R'2(TYPE, PUISS, COULEUR) =>F'2 = {TYPE -> PUISS}
   R'3(TYPE, MARQUE) => F'3 = {TYPE -> MARQUE}
   **==> ON A PERDU LA DF : NVH ->COULEUR**

## Conclusion : Décomposition

**Idée:** casser une relation R en des relations plus petites R1… Rn telles que l'union des schéma des Ri soit égal au schéma de R

Pour tout r instance de R, ri = proj $_{\text{Schéma de Ri}}$ (r)

- **exigence n°1** : pouvoir reconstituer r par la jointure des ri
  -> Décomposition sans perte

- **exigence n°2** : conserver les contraintes d'intégrité
  -> Préserver les DFs



## Principe de la normalisation

Spécifier différents niveaux de redondance et le moyen de les obtenir

-> Garder le maximum de contraintes

-> Diminuer le plus possible la redondance

-> Automatiser le processus

## Première Forme Normale

**Définition**

Une relation est en 1° forme normale si tout attribut est atomique

**Conséquences**

-> Un attribut représente une donnée élémentaire du monde réel
-> Un attribut ne peut pas désigner :
1) UNE DONNEE COMPOSEE (ENTITE)
PERSON (NPRO, NOM, VOITURE )
VOITURE (NVH, TYPE, MARQUE, PUIS, COUL)
2) UNE LISTE DE DONNEES
PERSON (NSS, NOM, [PRENOM]+)
PERSON (NPRO, NOM, [VOITURE]+)

**Objectif** :

Mettre à plat les hiérarchies - cohérence avec la logique des prédicats du 1 er ordre

2001/02/22 saglio@enst.fr 29

## Deuxième Forme Normale

**Objectif**

Tous les attributs non clé doivent dépendre de la même clé

**Définition**

*Une relation est en* 2ème FORME NORMALE ssi :
1) elle est en 1FN
2) tout attribut n'appartenant pas à une clé ne dépend par que d'une partie de cette clé

**Contre-exemple**

CONTROLE (<u>ETUDIANT, EXAMEN</u>, HORAIRE, NOTE)
la clé est (ETUDIANT, EXAMEN), et on a la DF : EXAMEN -> HORAIRE

2001/02/22 saglio@enst.fr 30

## Troisième Forme Normale

**Objectif**

Elimination des redondances dues aux DF déduites par transitivité

**Définition** :

Une relation est en 3ème FORME NORMALE ssi :

1) elle est en 2FN

2) tout attribut n'appartenant pas à une clé ne dépend pas d'un attribut non clé

**Contre-exemple :**

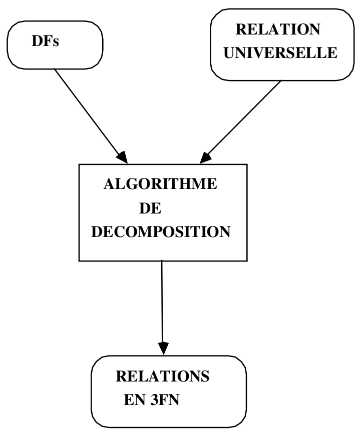
VOITURE (NVH, TYPE, MARQUE, PUISS, COULEUR)

La clé est (NVH), et on a les DF : TYPE -> MARQUE et TYPE -> PUISS

2001/02/22 saglio@enst.fr 31

## Algo 3FN

AUTOMATISATION DU PROCESSUS DE DECOMPOSITION EN 3FN, A PARTIR DE L'ENSEMBLE DES ATTRIBUTS ET DES DFs




2001/02/22 saglio@enst.fr 32

## Algorithme

1) A partir du graphe G des DFs, calculer une couverture minimale C

2) Editer l'ensemble des attributs isolés dans une même relation R0 (tous les attributs sont clés)

3) Rechercher le plus grand ensemble X d'attributs qui détermine d'autres attributs A1, … An

4) Editer la relation (X, A1, … , An) qui est en 3FN

5) Supprimer les DFs (X->A1, … X-> An) du graphe de couverture minimale C

6) Supprimer les attributs isolés de C (i.e. les attributs non-cible ni source de DF)

7) Répéter l'opération de réduction du graphe C à partir de l'étape 3, jusqu'à ce que C soit vide.

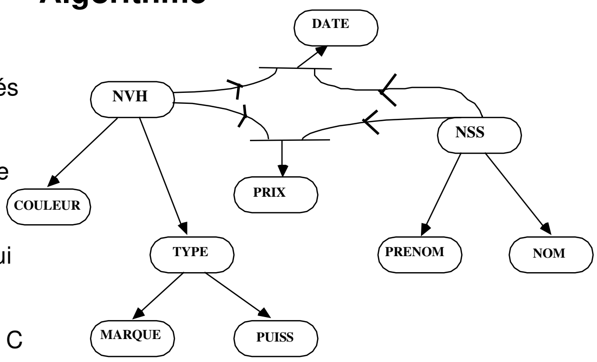



**Schéma relationnel normalisé**

POSSEDE (NVH, NSS, DATE, PRIX)
VEHICULE (NVH, TYPE, COULEUR)
MODELE (TYPE, PUISS, MARQUE)
PERSONNE (NSS, NOM, PRENOM)



## Décomposition sans perte en 3FN

L'algorithme assure une décomposition

- dans laquelle **les relations sont en 3 FN**
- la décomposition **préserve les Dépendances Fonctionnelles**

Etape 8

-> Pour **garantir** que la **décomposition** est **sans perte**, il suffit d'ajouter une relation composée des attributs formant une clé de la relation universelle décomposée (en complétant R0 si elle existe)

## 3FN : Qualités

**Théorème**

*Toute relation R admet une décomposition (R1, …, Rn) en 3FN telle que :*

1) la décomposition préserve les DF
2) la décomposition est sans perte

**Exemple**

VOITURE (NVH, TYPE, MARQUE, PUISS, COULEUR)

admet comme décomposition

VEHICULE (NVH, TYPE, COULEUR)
MODELE (TYPE, MARQUE, PUISS)

Rem: NVH, TYPE clé de la RU VOITURE est dans le schéma de la relation VEHICULE (avec NVH clé de VEHICULE et TYPE clé de MODELE)



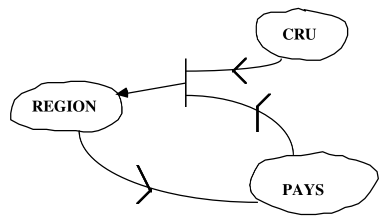
## 3FN : Défauts

**INSUFFISANCE DE LA 3FN**

.EXTENSION DE LA RELATION VIN (CRU, PAYS, REGION)

. **GRAPHE DES DF**

| VIN | CRU | PAYS | REGION |
|---|---|---|---|
| | CHENAS | FRANCE | BEAUJOLAIS |
| | JULIENAS | FRANCE | BEAUJOLAIS |
| | CHABLIS | FRANCE | BOURGOGNE |
| | CHABLIS | USA | CALIFORNIE |



## Boyce Codd Normal Form

**Définition**

Une relation est en BCNF si et seulement si les seules dépendances fonctionnelles non triviales sont celles pour lesquelles une clé détermine un ou plusieurs attributs.

**Exemple**

VIN (CRU, PAYS, REGION)

-> 2 Clés candidates : (CRU,PAYS) (CRU, REGION)

-> Il existe deux DFs Elémentaires :

1) (CRU,PAYS) -> REGION
2) REGION -> PAYS

2001/02/22 saglio@enst.fr 37

## BCNF (suite)

**Théorème**

Toute relation admet au moins une décomposition en BCNF qui est sans perte. Une telle décomposition ne préserve pas, en général, les dépendances fonctionnelles.

**Exemple**

VIN (CRU, PAYS, REGION)

-> Une décomposition possible :

1) CRUS (CRU, REGION)
2) REGIONS (REGION, PAYS)

-> ON A PERDU LA DF :

(CRU, PAYS) ->REGION

2001/02/22 saglio@enst.fr 38

## Propriétés de la BCNF

**Algorithme**

d = {R}

Pour tout S élément de d, appliquer récursivement la procédure suivante:

Si X->A élément de F+ tel que XA inclus dans S et A non élément de X et X non superclé de S

alors remplacer S par S1 et S2 tel que

S1 = XA

S2 = S-A

jusqu'à ce que toutes les relations soient en BCNF

**Problème** :

Savoir si une décomposition est en BCNF est un problème NP-Complet

Les algorithmes de décomposition connus sont exponentiels !

2001/02/22 saglio@enst.fr 39

## Récapitulatif

- **2NF**

  **R(A, B, C, D, E)**

  **R1(B, C)**

  **R2(A, B, D, E)**

- **3NF**

  **R(A, B, C, D, E)**

  **R1(C, D)**

  **R2(A, B, C, E)**

- **BCNF**

  **R(A, B, C, D, E)**

  **Rem: (A,C) est clé secondaire**

  **R1(C, B)**

  **R2(A, C, D, E)**

2001/02/22 saglio@enst.fr 40

## Autres dépendances

| OPTIONS | N°ELEVE | MODULE | SPORT |
|---|---|---|---|
| | 100 | BDA | TENNIS |
| | 100 | BDA | ESCALADE |
| | 102 | ADA | FOOTBALL |
| | 102 | BDA | FOOTBALL |

TOUT EST CLE, IL Y A DES REDONDANCES ET PAS DE DF !

LA NOTION DE DEPENDANCE FONCTIONNELLE EST INSUFFISANTE
=> dépendances de jointure et dépendances multivaluées

2001/02/22 saglio@enst.fr 41

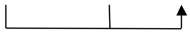
## Conclusion

Série de FN
BCNF => 3FN => 2 FN => 1 F
Mais il reste des problèmes de redondance qu'on ne sait pas spécifier avec les DF :
dépendances multivaluées (DM) et dépendances de jointure (DJ)
Les DF sont un cas particulier des dépendances multivaluées
=> 4°, 5°… formes normales
Pb : elles sont sans pertes mais ne préservent pas les contraintes monorelations
=> Alternative: normaliser vs. dénormaliser
- normaliser beaucoup mais
  - Perte de performances à cause des jointures
  - Perte de CI (CI interrelations)
- normaliser peu
  - Définir des contraintes pour vérifier la redondance

2001/02/22 saglio@enst.fr 42

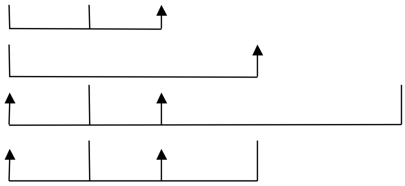
## TD énoncé

**R1(Etudiant, Cours, Note)**



**R2(Etudiant, Examen, Horaire)**

**R3(Cours, Heure, Salle, Professeur, Etudiant)**



**Ces relations et ces DF constituent une spécification / fixent la sémantique**

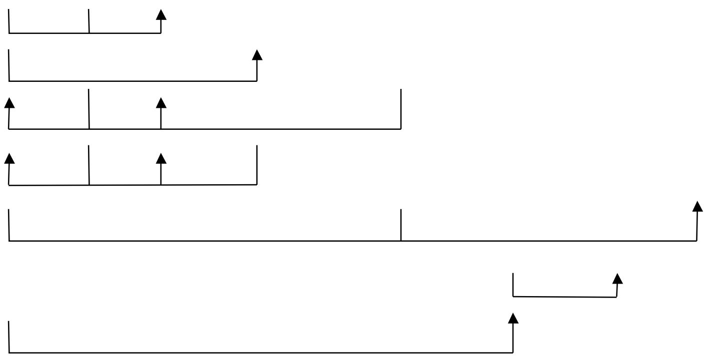
**Est-ce que ce schéma est normalisé ?**

**2001/02/22** **saglio@enst.fr** **43**

## TD graphe des DFs

**RU(<u>Cours, Heure</u>, Salle, Professeur, <u>Etudiant</u>, Examen, Horaire, Note)**

**DF1**
**DF2**
**DF3**
**DF4**
**DF5**
**DF6**
**DF7**



**Couverture minimale?** En retirant de DF3 et DF4 la -> Salle

**on peut simplifier (Cours, Examen, Horaire) en ( Cours, HeureExamen)**
**et ne conserver que DF67 déduite par transitivité de DF6 et DF7**

**2001/02/22** **saglio@enst.fr** **44**

## TD schéma en 3NF

**Application de l'algorithme de décomposition**

1) pas d'attribut isolé
**2) HORAIRES(<u>Cours, Heure</u>, Salle)**
3) Salle attribut isolé
**2') CHARGES(<u>Professeur, Heure</u>, Cours)**
3') pas de nouvel attribut isolé
**2'') PARTICIPATIONS(<u>Etudiant, Heure</u>, Cours)**
3'') pas de nouvel attribut isolé
**2''') ENSEIGNEMENTS(<u>Cours</u>, Professeur, HeureExamen)**
3''') Professeur et HeureExamen attributs isolés
**2'''') EXAMENS(<u>Etudiant, Cours</u>, Note)**
3'''') Note, Cours et Etudiant attributs isolés

N.B. la clé de la RU est entièrement dans la relation PARTICIPATIONS

**Remarque:**
**on peut vérifier sur cet exemple que l'ordre de considération des DFs ne change pas le résultat; le traitement dans l'ordre des DFs ayant le plus grand nombre d'attributs en partie gauche n 'est qu' un moyen d 'être sûrement exhaustif.**

2001/02/22 saglio@enst.fr 45