

# Chapitre 6

## Formes normales

# 1. Intérêt de la normalisation

Les relations d'une base de données relationnelle doivent obéir à certaines règles pour :

- Eviter les redondances
- Optimiser le stockage des données
- Effectuer les mises à jour (insertion, suppression, modification) sans anomalies

**Exemple :** Supposons la relation R(eleve, classe, salle) obéissant aux règles suivantes : Un élève n'appartient qu'à une seule classe ; une classe a lieu toujours dans la même salle.

R

eleve	classe	salle
DUPONT	CM1	1
DURAND	CM2	2
DUBOIS	CM1	1
DUVAL	CM2	2
DUGENOU	CM1	1
DURACUIRE	CM2	2
DUPUIS	CM1	1
DUBALAI	CM2	2

On observe un certain nombre de redondances qu'il convient de minimiser

La méthode quasi universelle pour éviter les redondances est la décomposition d'une relation en relations "plus simples". Evidemment, la décomposition doit être sans perte : une jointure des relations obtenues par décomposition doit redonner la relation originale.

R1

eleve	classe
DUPONT	CM1
DURAND	CM2
DUBOIS	CM1
DUVAL	CM2
DUGENOU	CM1
DURACUIRE	CM2
DUPUIS	CM1
DUBALAI	CM2

R2

classe	salle
CM1	1
CM2	2

Décomposition I : R est décomposée en R1 et R2

R1

eleve	classe
DUPONT	CM1
DURAND	CM2
DUBOIS	CM1
DUVAL	CM2
DUGENOU	CM1
DURACUIRE	CM2
DUPUIS	CM1
DUBALAI	CM2

R3

eleve	salle
DUPONT	1
DURAND	2
DUBOIS	1
DUVAL	2
DUGENOU	1
DURACUIRE	2
DUPUIS	1
DUBALAI	2

Décomposition II : R est décomposée en R1 et R3

R2

classe	salle
CM1	1
CM2	2

R3

eleve	salle
DUPONT	1
DURAND	2
DUBOIS	1
DUVAL	2
DUGENOU	1
DURACUIRE	2
DUPUIS	1
DUBALAI	2

Décomposition III : R est décomposée en R2 et R3

Quelle est la meilleure décomposition ? On peut déjà vérifier que les trois décompositions sont sans perte d'information :  $R1 \bowtie R2 = R$ ,  $R1 \bowtie R3 = R$ ,  $R2 \bowtie R3 = R$

Imaginons maintenant des mises à jour :

1. **Insertion** d'un nouvel élève DUCHEMIN en CM1 : Facile avec la décomposition I (seule la table R1 est concernée)  
Facile avec la décomposition II mais il faudrait ensuite lui affecter une salle.  
Impossible avec la décomposition III : pas de lien entre eleve et classe, on peut ajouter DUCHEMIN mais rien ne dit dans quelle classe il est (sauf si on lui affecte une salle)
2. **Suppression** des élèves DUPONT, DUBOIS, DUGENOU, DUPUIS : Facile avec la décomposition I (seule la table R1 est concernée)  
Moins aisée avec la décomposition II (il faut mettre à jour R1 et R3)  
Facile avec la décomposition III (seule la table R3 est concernée)
3. **Modification** : La classe CM1 est déplacée en salle 3 : Facile avec la décomposition I (seule la table R2 est concernée)  
Impossible avec la décomposition II : il faut reprendre chaque élève.  
Mise à jour incomplète avec la décomposition III : il faut mettre à jour aussi la table R3 sinon la base devient inconsistante.

On voit donc que les décompositions ne sont pas toutes équivalentes On notera que la décomposition I optimise le stockage et permet des mises à jour aisées.

Les règles de normalisation permettent d'obtenir la bonne décomposition.

## 2. Première forme normale : 1NF

Codd a défini une méthode permettant d'obtenir une décomposition d'une relation. A cet effet, il a introduit 3 degrés de normalisation : 1NF (première forme normale), 2NF (deuxième forme normale), 3NF (troisième forme normale).

**1ère forme normale :** Une relation est en 1ère forme normale si tout attribut est atomique.

**Exemple :** Soit les relations **PERSONNE (nom, prénom, adresse)** ADRESSE (numero, rue, ville, code postal, pays) La relation PERSONNE n'est pas en 1NF car adresse est une relation composée d'attributs.

Exemple :

**EMPLOYE ( matricule, nom, prenom, enfants, experience)**  
**ENFANTS ( nom\_enfant, date\_naissance,)**  
**EXPERIENCE ( date\_experience, emploi, historique-salaire)**  
**HISTORIQUE-SALAIRE ( date\_salaire, salaire)**

Les relations de cette base de données ne sont pas en 1ère forme normale. Pourquoi ? Parce que "enfants" figurant dans EMPLOYE est une relation comportant plusieurs attributs. De même, "experience" est une relation comportant plusieurs attributs. "Historique-salaire" est aussi une relation comportant plusieurs attributs.

Conversion vers la 1ère forme normale

1ère étape : normalisation de la relation EMPLOYE

**EMPLOYE ( matricule, nom, prenom)**  
**ENFANTS ( matricule, nom\_enfant, date\_naissance)**  
**EXPERIENCE ( matricule, date\_experience, emploi, historique\_salaire )**  
**HISTORIQUE-SALAIRE ( date\_salaire, salaire)**

2ème étape : normalisation de la relation EXPERIENCE

**EMPLOYE ( matricule, nom, prenom )**  
**ENFANTS ( matricule, nom\_enfant, prenom\_enfant date\_naissance)**  
**EXPERIENCE ( matricule, date\_experience, emploi )**  
**HISTORIQUE-SALAIRE ( matricule,date\_experience,date\_salaire, salaire)**

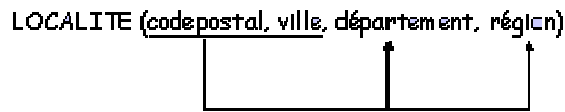
Prix à payer : la complexité des clés !

## 3. Deuxième forme normale : 2NF

**2ème forme normale :** Une relation est en deuxième forme normale si

- elle est en première forme normale
- tout attribut n'appartenant pas à une clé ne dépend pas d'une partie seulement de cette clé.

**Exemple :** Soit la relation LOCALITE (codepostal, ville, département, région). Cette relation n'est pas en 2NF car



On a codepostal -> département et codepostal -> région. Donc département et région dépendent de codepostal (partie de la clé)

Remarque : codepostal, ville n'est pas une bonne clé

Comment mettre une relation en 2NF ?

- On cherche la bonne clé.
- On décompose en deux relations.
- On attribut une clé fictive.

Quelle est la meilleure méthode ?

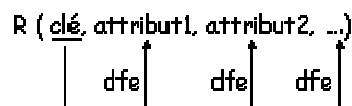
## 4. Troisième forme normale : 3NF

3ème forme normale: Une relation est en 3ème forme normale si

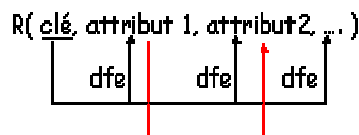
a) elle est en 2ème forme normale

b) tout attribut n'appartenant pas à une clé ne dépend pas d'un attribut qui ne serait pas dans cette clé

Une relation en 3ème forme normale est donc du type :



et non pas



**Exemple :**

PILOTE(NUM-PIL, NOM-PIL, ADR-PIL, SAL-PIL) est en 3NF.

**Exemple :**

AVION(NUM-AV, NOM-AV, LOC-AV, CAP-AV) n'est pas en 3NF car NOM-AV -> CAP-AV.

dans ce cas il faut décomposer la relation de la manière suivante :

TYPE(NOM-AV, CAP-AV)

AVION1(NUM-AV, NOM-AV, LOC-AV)

On notera que en 3NF tout attribut ne dépend que de la clé.

## 5. Forme normale de Boyce-Codd : BCNF

Cas non prévu de la 3NF : cas où un attribut constituant de la clé dépend d'un attribut ne figurant pas dans la clé

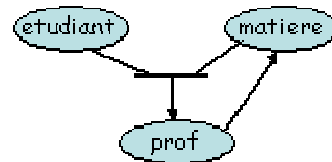
**Exemple :**

COURS (etudiant, matiere, prof) où chaque "etudiant" étudie une "matière" enseignée par un "prof" ; on suppose qu'un "prof" n'enseigne qu'une seule "matière".

Les dépendances fonctionnelles sont : (etudiant, matiere) -> prof et prof -> matiere

observation de redondances :

etudiant	matiere	prof
DUPONT	mathématiques	COSINUS
DUPONT	physique	TOURNESOL
DURAND	mathématiques	COSINUS
DURAND	physique	EINSTEIN



Que se passe-t-il si DUPONT n'étudie plus la physique ? Le prof "Tournesol" disparaît de la BD !

La forme normale de Boyce-Codd est une amélioration de la 3NF :

**forme normale de Boyce-Codd :**

Une relation est en forme normale BC si les seules dépendances fonctionnelles élémentaires sont du type: clé -> attribut. (est non pas attribut->clé)

Solution (provisoire) de l'exemple précédent

décomposition en 2 relations  
R1 (prof, matiere)  
R2 (etudiant, prof)

prof	matiere
COSINUS	mathématiques
TOURNESOL	physique
EINSTEIN	physique

etudiant	prof
DUPONT	COSINUS
DUPONT	TOURNESOL
DURAND	COSINUS
DURAND	EINSTEIN

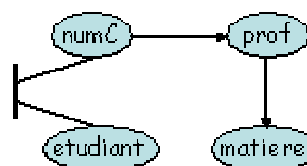
On a bien COURS = R1  $\bowtie$  R2

**Remarque 1 :** on notera que la dépendance etudiant ->> prof est multivaluée.

**Remarque 2 :** la dépendance fonctionnelle (etudiant, matiere) à prof est perdue !

Une solution adéquate serait

COURS1 (numC, prof)  
PROF (prof, matiere)  
MATIERE (matiere)  
ETUDIANT (etudiant)  
SUIVI (etudiant, numC)



## 6. Décomposition SPI SPD en 3NF

**Algorithme de Berstein (ou de synthèse)**

1. Remplacer F par une couverture minimale.
2. Regrouper toutes les dépendances selon le membre gauche (les  $X_i \rightarrow A_{ik}$  son regroupés en une seule dépendance  $X_i \rightarrow Y_i$  où  $Y_i$  est l'union sur k des  $A_{ik}$  }
3. Prendre alors l'ensemble constitué de la réunion des schémas  $R_i = (X_i, Y_i)$

4. Si aucune clé n'est dans l'un des  $R_i$  ajouter alors le schéma  $R_o = \{Z\}$  où  $Z$  est une clé.
5. L'ensemble des  $R_i$  est une décomposition de  $R$  SPI SPD et en 3NF.

### Exemple

soit **etude(Cours,Prof,Heure,Salle,Etudiant,Note)** et les DF :

$C \rightarrow P$  donc  $R_1 = \{C, P\}$   
 $HS \rightarrow C$  donc  $R_2 = \{H, S, C\}$   
 $HP \rightarrow S$  donc  $R_3 = \{H, P, S\}$   
 $HE \rightarrow S$  donc  $R_4 = \{H, E, S\}$   
 $CE \rightarrow N$  donc  $R_5 = \{C, E, N\}$

$R_4$  contient la clé (HE), il n'y a rien à ajouter.  
 Il se trouve qu'elle est également BCNF.

Attention il est impératif de partir d'une couverture minimale !

### Exemple

$R(\text{Nom}, a\text{Dr}, n\text{Um}, \text{Annee}, \text{Code}, \text{Resp}, da\text{Te}, r\text{Ecu}, \text{Jour}, \text{Heure}, \text{Salle})$

$F(\text{minimale}) = \{U \rightarrow N, U \rightarrow D, C \rightarrow R, C \rightarrow A, UT \rightarrow A, UT \rightarrow E, JHS \rightarrow C\}$

d'où l'on tire :

$U \rightarrow ND$  soit  $R_1(U, N, D)$

$C \rightarrow RA$  soit  $R_2(C, R, A)$

$UT \rightarrow AE$  soit  $R_3(U, T, A, E)$

$JHS \rightarrow C$  soit  $R_4(J, H, S, C)$

L'unique clé étant (UTJHS), il manque une clé dans la décomposition et donc on rajoute  $R_5(U, T, J, H, S)$

### Exemple

$R(A, B, C, D)$ ,  $F(\text{minimale}) = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D\}$ , possédant deux clés : AB et BC, se décompose en  $R_1(ABC)$   $R_2(A \rightarrow C)$  et  $R_3(BCD)$

mais  $R_1$  n'est pas en BCNF à cause de  $C \rightarrow A$  dans laquelle la partie gauche  $C$  n'est pas une surclé.

La présence d'une partie contenant une clef est indispensable :

### Exemple

$R(A, B, C, D)$ ,  $F(\text{minimale}) = \{A \rightarrow C, B \rightarrow D\}$ . La décomposition en  $R_1(AC)$   $R_2(BD)$  n'est pas SPI car on n'a pas  $R_1$  inter  $R_2 \rightarrow R_1 - R_2$  (ou  $R_2 - R_1$ ) puisque  $R_1$  inter  $R_2$  est vide ! Il faut absolument rajouter une clé  $R_3(AB)$  alors  $R_1, R_2, R_3$  est bien SPD SPI en 3NF.