7. Décomposition

Lors d'une conception d'une base de données relationnelle la première étape consiste à définir son schéma relationnel. Il existe plusieurs possibilités de schémas, il faut choisir les meilleurs. Encore faut-il définir les critères qui les qualifient et concevoir les algorithmes qui permettent de les obtenir.
Il s'agit de décomposer une relation R en un ensemble de relations, sans perte d'informations en évitant les **redondances** (répétition d'informations).

Exemple Soit la relation CoursesDeBateaux (<u>Nbat</u>, Nombat, <u>Ncomp</u>, Score) dont l'extension est la suivante :

CoursesDeBateaux

Nbat	Nombat	Ncomp	Score
102	TASSILI	210	2
102	TASSILI	240	1
102	TASSILI	270	4
103	EL BAHDJA	210	4
103	EL BAHDJA	215	3
104	LA COLOMBE	200	2
104	LA COLOMBE	210	1
104	LA COLOMBE	215	2
104	LA COLOMBE	220	4
104	LA COLOMBE	240	3
104	LA COLOMBE	270	3
105	HOGGAR	210	3
105	HOGGAR	220	1

Le tuple (104, LA COLOMBE, 270, 3) signifie que le bateau n° 104 nommé La colombe a participé à la course 270 avec un score = 3.

La relation CoursesDeBateaux présente plusieurs anomalies.

Les principales anomalies de mise à jour sont les suivantes :

- anomalie de modification: si on doit changer une information redondante, par exemple le nom d'un bateau, il faut modifier cette information dans plusieurs tuples.
- anomalie d'insertion : pour ajouter les informations d'un bateau, il faut l'associer à une course ou compétition.
- anomalie de suppression : si un ensemble de valeurs devient vide, on peut perdre d'autres informations. Par exemple, si on supprime le bateau 104, on perd les informations sur la compétition n°200.

Pour éviter ces anomalies, il faut supprimer la redondance en décomposant le schéma relationnel. On utilise pour cela les dépendances fonctionnelles.

Qualité d'une décomposition
La qualité d'une décomposition est liée à deux aspects :
- être sans perte d'information,
- être préservative des DFs

a)Décomposition sans perte d'information

Considérons la relation CoursesDeBateaux (Nbat, Nombat, Ncomp, Score) avec la df Nbat → Nombat

Soit les deux décompositions possibles de cette relation

a)

Bateau

Nbat	**Nombat**
102	TASSILI
103	EL BAHDJA
104	LA COLOMBE
105	HOGGAR

Compétition

Ncomp	**Score**
210	2
240	1
270	4
210	4
215	3
200	2
210	1
215	2
220	4
240	3
270	3
210	3
220	1

Dans cette décomposition, on connait les noms des bateaux, mais on ne sait plus quel bateau a participé à quelle course.
D'où perte d'information.

b)

Bateau

Nbat	**Nombat**
102	TASSILI
103	EL BAHDJA
104	LA COLOMBE
105	HOGGAR

Course

Nbat	**Ncomp**	**Score**
102	210	2
102	240	1
102	270	4
103	210	4
103	215	3
104	200	2
104	210	1
104	215	2
104	220	4
104	240	3
104	270	3
105	210	3
105	220	1

Dans cette décomposition, on connait les noms des bateaux et on sait quel bateau a participé à quelle course et a obtenu quel score.
Cette décomposition est bien sans perte d'information.

Principe de décomposition :

Théorème de Heath : Soit la relation R(A, B, C) où A, B et C sont des ensembles d'attributs de R. Si R satisfait la DF $A \rightarrow B$, alors R est égale à la jointure de sa projection sur {A, B} et {A, C}

Autrement dit, **la décomposition de R en R1(A, B) et R2(A, C) est sans perte d'information**.

Exemple:
Si on applique ce principe sur la relation CoursesDeBateaux (<u>Nbat</u>, Nombat, <u>Ncomp</u>, Score) avec la df Nbat → Nombat, on obtient la décomposition (b) de l'exemple précédent. Donc, la décomposition (b) est sans perte d'information.

b)Préservation des DFs :

Soit R une relation et F son ensemble de DFs, et soit une décomposition R1, R2, …, Rn de R, on note Fi l'ensemble de DFs de R qui portent sur les attributs de Ri ; cette décomposition préserve les DFs si et seulement si F est équivalent à (F1 U F2 U … U Fn)
C'est-à-dire si G= F1 U F2 U … U Fn alors $F^+ = G^+$

Exemple : Considérons l'exemple précédent
Considérons la relation CoursesDeBateaux (<u>Nbat</u>, Nombat, <u>Ncomp</u>, Score) avec la df Nbat → Nombat

Décomposition a) : Bateau :{ Nbat → Nombat }, Compétition : pas de Df
=> ne préserve pas les DF (on a perdu Nbat, Ncomp→ Score puisque (Nbat, Ncomp) est la clé primaire de CoursesDeBateaux) ($F^+ \neq$ de $\{ Nbat \rightarrow Nombat \}^+$)

Décomposition b) : Bateau :{ Nbat → Nombat }, Course : { Nbat, Ncomp→ Score }
=> préserve les DF $F^+ = (\{Nbat \rightarrow Nombat\} \cup \{Nbat, Ncomp \rightarrow Score\})^+$

8. Normalisation

Codd propose en 1971 les bases d'une démarche de normalisation qui conduit à décomposer une relation selon plusieurs schémas plus restreints, normalisés, sans perte d'informations. Cette démarche repose aujourd'hui sur un ensemble rigoureux de règles et la mise en œuvre d'algorithmes performants.

8.1. Les trois formes normales

a) Première forme normale (1FN)

Définition : Une relation est en 1 FN si tous les domaines sous-jacents contiennent uniquement des valeurs scalaires (valeurs atomiques).
(Scalaire : plus petite unité sémantique de données telle que la valeur du numéro d'un fournisseur, du poids d'une pièce…)

Exemple : CoursesDeBateaux (<u>Nbat</u>, Nombat, <u>Ncomp</u>, Score) est en 1FN

b) Deuxième forme normale (2FN)

Définition : Une relation est en 2FN si et seulement si elle est en 1FN et si chaque attribut non clé est en **dépendance irréductible** (élémentaire) avec la clé.

Definition Soit R une relation, X et Y des sous ensembles d'attributs de R. Y est en DF **élémentaire** avec X ssi il n'existe pas X' sous ensemble strict de X tq X' →Y.

Exemple :
CoursesDeBateaux (<u>Nbat</u>, Nombat, <u>Ncomp</u>, Score) 1FN
avec { Nbat → Nombat , Nbat, Ncomp → Score, Nbat, Ncomp → Nombat }

Nbat → Nombat => **Nombat n'est pas en DF élémentaire avec la clé**
=> CoursesDeBateaux n'est pas en 2FN.

La DF (Nbat → Nombat) peut causer des anomalies de mise à jour à cause des redondances des valeurs de Nombat. La solution à ces problèmes est de décomposer la relation CoursesDeBateaux.

Une relation qui est 1FN et pas 2FN peut être décomposée en un ensemble de relations 2FN équivalentes (sans pertes d'informations).

Ainsi CoursesDeBateaux. est décomposée en R2 et R3 tq

R1 (<u>Nbat</u>, Nombat)
R2 (<u>Nbat, Ncomp</u>, Score)

Même décomposition (b)

Les problèmes de mise à jour sont résolus

Résumé :
Une première étape de la procédure de normalisation consiste à effectuer des décompositions pour supprimer les dépendances fonctionnelles qui ne sont pas irréductibles (élémentaires).

Principe de décomposition :
Soit la relation R (<u>A, B</u>, C, D) et la DF A → D

Les principes de normalisation recommandent de remplacer R par ses deux projections R1 et R2 tq :
- R1 (<u>A</u>, D)
- R2 (<u>A, B</u>, C)

La relation R peut être retrouvée en effectuant la jointure de R1 et R2 sur la clé de R1.

c) Troisième forme normale (3FN)

Définition : Une relation est en 3FN si et seulement si elle est en 2FN et si chaque attribut non clé est en dépendance directe (non transitive) avec la clé primaire.

Définition Soit R une relation, X et Y des sous ensembles d'attributs de R. Y est en DF **directe** (non transitive) avec X ssi il n'existe pas Z sous ensemble d'attributs de R, tq X →Z et Z →Y.

Exemple :
La décomposition de CoursesDeBateaux en R1 et R2 est en 3FN.

R1 (<u>Nbat</u>, Nombat) 3FN
R2 (<u>Nbat, Ncomp</u>, Score) 3FN

Autre exemple : Soit la relation Enseignant (Nom, Bureau, Bâtiment, Discipline)
Et la Df Bâtiment → Discipline

La relation Enseignant est en 2FN.

Batiment → Discipline => Discipline n'est pas en dépendance directe avec la clé Nom
=> Enseigant n'est pas en 3FN

Enseignant

Nom	Bureau	Batiment	Discipline
Mohammedi	85	10	Chimie
Alioua	111	13	Electronique
Benzine	220	30	Informatique
Benomar	210	30	Informatique
Aliyacine	70	10	Chimie
Leila	233	30	Informatique
Semmar	93	14	Electronique
yacoubi	220	30	Informatique

La DF transitive (Nom → Discipline) peut causer des anomalies de mise à jour à cause des redondances des valeurs de Discipline

La solution à ces problèmes est de décomposer la relation Eneignant comme suit :
Enseignant2 (Nom, Bureau, Batiment)
Locaux (Batiment, Discipline) Batiment → Discipline

L'extension devient :

Enseignant 2

Nom	Bureau	Bâtiment
Mohammedi	85	10
Alioua	111	13
Benzine	220	30
Benomar	210	30
Aliyacine	70	10
Leila	233	30
Semmar	93	14
yacoubi	220	30

Locaux

Bâtiment	Discipline
10	Chimie
13	Electronique
30	Informatique
14	Electronique

Décomposition en 3FN
sans perte d'information
et qui préserve les DF

Résumé : Effectuer des décompositions pour éliminer les dépendances transitives.

Principe de décomposition :
Soit R(A, B, C) et B → C
Les principes de normalisation recommandent de remplacer R par R1 et R2 tq :
- R1 (B, C)
- R2 (A, B)

Remarque :
Toute relation a une décomposition en 3FN sans perte d'informations et qui préserve les DFs.

Plusieurs approches permettent cette décomposition, parmi elles l'algorithme de Bernstein.

ALGORITHME DE SYNTHESE (BERNSTEIN)

Cet algorithme est basé sur la décomposition de la relation universelle.

La relation universelle est une relation définie sur tous les attributs.

ENTREE : R = {A, F}, A = {ensemble de tous les attributs}, F = {ensemble des DFs}
SORTIE : D = {R1, R2,…, Rn} tel que Ri = $\{Xi, Y_i\}_{i=1,\ldots,n}$ en 3FN

ETAPE 1 : Rechercher une couverture irréductible de F notée F'
ETAPE 2 : Partitionner F' en groupes $F'_1, F'_2, \ldots, F'_k$ tels que toutes les DFs d'un même groupe aient même partie gauche Xi
ETAPE 3 : Pour chaque groupe F'i construire un schéma Ri={Xi, Yi} où Yi est l'ensemble de tous les attributs apparaissant à droite du groupe F'i
ETAPE 4 : Soient A1, A2, An les attributs de R (s'il y en a qui n'ont pas encore été pris en compte. Ajouter à D le schéma de relation constitué de la projection de R sur ces attributs.
ETAPES 5 : Si aucune relation de D n'inclut une clé candidate de R, ajouter à D le schéma de relation constitué d'une clé candidate de R.

3.3.3 Forme normale de Boyce/ Codd (BCNF) :

Dans le cas général, une relation peut avoir plusieurs clés candidates qui peuvent être composées et se chevaucher (c'est-à-dire qu'elles ont des attributs communs).

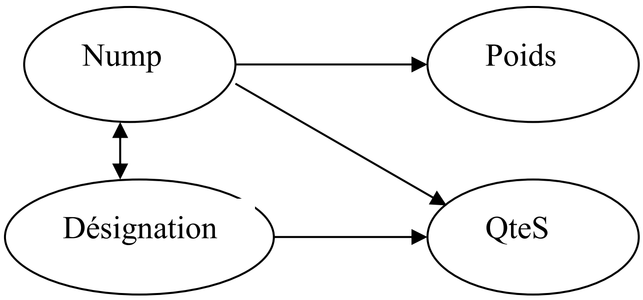
Il y a une forme normale plus forte que la troisième forme normale : la **forme normale de Boyce Codd (BCNF)**

Définition Une relation est en BCNF si et seulement si les seuls déterminants sont des clés candidates.

La BCNF impose qu'il ne peut pas exister dans une relation, d'autres dfs que celles qui partent des clés candidates.

Exemple :

Produit (NumP, Designation, Poids, QteS) et NumP et Désignation sont deux clés candidates



Cette relation est en BCNF, car les seules flèches partent des deux clés candidates.

Exemple : Relation en 3FN main non BCNF

Considérons la relation Cours (Matière, Classe, Professeur)

avec les règles de gestion suivantes:

- un professeur n'enseigne qu'une seule matière,
- une classe n'a qu'un seul enseignant par matière

desquelles on déduit les DF { Matière, Classe → Professeur, Professeur → Matière}

La relation Cours est en 3NF.

La relation Cours n'est pas en BCNF car on a (Professeur→Matière): c'est une Df dont le déterminant n'est pas une clé.

La relation Cours peut être décomposée en 2 relations BCNF :

- Spécialité (Professeur, Matière)
- Enseignant (Classe, Professeur)

Mais on perd la df Matière, Classe → Professeur

Remarques :

- Une relation en BCNF est en 3FN, l'inverse n'est pas forcément vrai.
- Une décomposition en BCNF ne préserve pas toujours les dfs.

- Fin Chapitre 2 -