Université de Sherbrooke

Théorie et modèles relationnels

Normalisation

TMR_06

Christina KHNAISSER (*christina.khnaisser@usherbrooke.ca*)
Luc LAVOIE (*luc.lavoie@usherbrooke.ca*)
(les auteurs sont cités en ordre alphabétique nominal)

—

Scriptorum/Scriptorum/TMR_06-Normalisation, version 3.1.0.a, en date du 2025-08-28

— document préliminaire —

Sommaire

Le présent module a été rédigé dans le cadre de l'exploration du thème « Modélisation, conception et exploitation de données » (MCED) par des membres du CoFELI. Le module présente l'essentiel des principes et des méthodes de normalisation d'un modèle en accord avec la théorie relationnelle.

Mise en garde

Le présent document est en cours d'élaboration ; en conséquence, il est incomplet et peut contenir des erreurs.

Historique

diffusion	resp.	description
2025-08-28	LL	Récupération de certains éléments de MLR_01-Normalisation-PRE (version 272b du 2025-04-01).
2025-03-21	LL	Revue éditoriale.
2024-08-09	CK	Récupération de notes diverses.

Table des matières

Introduction

Le présent document a pour but de présenter une synthèse de la normalisation relationnelle proposée par Edgar F. Codd et développée par la suite grâce aux contributions théoriques et fondamentales de Hugh Darwen, Christopher J. Date, Raymond F. Boyce, Nikos A. Lorentzos, Ronald Fagin et Jeffrey D. Ullman.

La présentation repose sur des bases minimales :

- la logique classique du premier ordre,
- la théorie des ensembles,
- l'arithmétique,
- les langages rationnels.

Évolution du document

Le présent document tire son origine de l'expérience d'enseignement des auteurs incluant les rétroactions des personnes étudiantes et auxiliaires d'enseignement ayant participé aux cours.

La première version du document a été rédigée sur les bases suivantes :

- le matériel pédagogique développé par les auteurs dans le cadre de formations relatives aux bases de données assurées entre 1983 et 2024 au Québec, en France, en Suisse, au Maroc, au Liban et au Cameroun ;
- différents travaux publiés par Codd, Darwen, Date, de Sainte Marie, Delobel, Elmasri, Fafin, Lorentzos, Navathe, Snodgrass et Ullman.

Contenu des sections

- La section 1 présente les objectifs et les principes de la normalisation relationnelle.
- La section 2 présente la première forme normale (1FN).
- La section 3 présente la forme normale de Boyce-Codd (FNBC).
- La section 4 présente la forme normale de projection-jointure (FNPJ), aussi appelée cinquième forme normale (5FN).
- La section 5 présente la forme normale stricte (FNS), aussi appelée sixième forme normale (6FN).

La rédaction d'une section sur la forme normale relativiste est envisagée dans un avenir prochain.

1. Objectifs et principes de normalisation

1.1. Motivation

La prémisse de la théorie relationnelle est l'équivalence entre le prédicat logique et la relation (et conséquemment entre la proposition logique et le tuplet de la relation). Afin de définir un calculateur logique (une machine relationnelle) fondé sur cette intuition, il convient de garantir l'interprétabilité des prédicats associés aux relations, donc la fermeture de ces dernières en regard des opérations à être permises (celles de l'algèbre relationnelle). Cela sera d'autant plus facile à garantir si toute variable est relationnelle.

En soi, cela n'est pas une limitation (du moins en regard des autres langages de programmation courants), puisque les opérateurs de base de l'algèbre relationnelle enchâssés dans un langage permettant la définition de fonctions récursives (comme SQL et TD) forment un noyau complet au sens de Turing (*Turing-complete*). En fait, l'état de la machine relationnelle est représenté par une variable de type tuplet dont chacun des attributs est de type relation.

Il est ainsi possible de garantir que toute évaluation d'une requête (exprimée en termes de l'algèbre relationnelle) maintient l'interprétation des relations en tant que prédicats logiques. Cette fermeture des expressions relationnelles garantit également l'exactitude de toute substitution de deux expressions équivalentes l'une par l'autre et fournit une base solide à l'optimisation des requêtes.

D'où viennent les prédicats ?

Un modèle relationnel représente un ensemble de prédicats, normalement issus de la modélisation du domaine du problème, par un ensemble de variables de relation. A priori, pour faciliter la correspondance entre les deux modèles, on fait correspondre un prédicat à une variable.

Qu'est-ce qu'un modèle logique « adéquat » ?

En fait, le modèle logique doit non seulement être équivalent à l'ensemble des prédicats retenus lors de la modélisation dite conceptuelle, mais il doit également être adéquat en tant que solution au problème ciblé à l'origine.

Pour être adéquat, le modèle logique d'une base de données doit minimalement être cohérent, valide et efficace. En pratique, il doit également être aussi évolutif, efficient et complet que possible. Scientifiquement, il faut aussi en assurer sa réfutabilité et, socialement, son adhésion aux règles éthiques.

Un modèle logique de données (MLD) est complet en regard d'un modèle conceptuel de données (MCD) si chacun des prédicats exprimables au sein du MCD y est représentable sous la forme d'une variable de relation (de base ou virtuelle) au sein du MLD.

Si la complétude est souhaitable, voire nécessaire, elle ne doit pas être obtenue au prix de la redondance, car celle-ci entraine souvent des problèmes d'efficacité, d'évolutivité et d'efficience. Elle peut même favoriser l'apparition de l'incohérence et de l'invalidité. En effet, lorsque la redondance des données entraine la redondance des propositions, elle devient un facteur important d'incohérences potentielles au moment de la mise à jour des données, de l'interrogation des données et de l'évolution du MLD. La réduction de la redondance (et, à défaut de son élimination, son contrôle) est donc un impératif majeur de qualité.

Comment construire un modèle logique adéquat ?

La conception relationnelle, et plus particulièrement la normalisation relationnelle, vise à produire un MLD adéquat tout en donnant une garantie structurelle de cohérence et de complétude en regard du MCD.

En décomposant les variables selon leurs dépendances fonctionnelles constitutives, il est possible de diagnostiquer les situations de redondance. La normalisation relationnelle a pour but de réduire, voire d'éliminer, l'occurrence de ces erreurs en redéfinissant (en décomposant) certaines variables de relations de manière appropriée tout en maintenant l'équivalence et l'expressivité avec l'ensemble des prédicats d'origine.

Par ailleurs, la redondance des contraintes en regard des prédicats est acceptable, voire souhaitable, au titre de mécanisme (automatisable) de vérification du modèle (or, il n'y a guère de vérification possible sans redondance). En effet, les contraintes n'entrainent pas de redondances de données (n'en ajoutant pas) et permettent de la contrôler (par les clés, l'inclusion, l'exclusion, etc.).

Par ailleurs, si la redondance des contraintes entre elles peut entrainer une éventuelle perte d'efficience et d'évolutivité, elle n'introduit pas d'incohérences, d'erreurs.

En résumé

La normalisation relationnelle a pour objectif de déterminer un ensemble de variables de relation :

- nécessaires et suffisantes pour couvrir un ensemble de prédicats issus du modèle du problème,
- telles que le nombre de redondances fonctionnelles soit minimal.

En outre, la normalisation relationnelle doit fournir l'information nécessaire à la détermination des contraintes garantissant la cohérence des redondances résiduelles.

Ainsi, la normalisation relationnelle contribue à rendre un modèle logique adéquat, plus particulièrement en regard de :

- la cohérence,
- l'efficacité,

- la complétude,
- l’évolutivité.

1.2. Définition

En théorie relationnelle, une forme normale désigne une représentation d’un ensemble de variables de relations permettant d’exclure toute redondance d’un certain type (chaque « forme normale » ciblant un certain type de redondance).

Par extension, la normalisation désigne une transformation applicable à un MLD permettant à celui-ci de s’approcher ou d’atteindre une forme normale sans perte d’adéquation en regard du modèle d’origine (le plus souvent un MCD).

On distingue deux catégories de normalisation :

- **Normalisation des attributs** (1FN).
  Assurer que tout attribut est associé à exactement une valeur de son type de définition.
- **Normalisation des relations** (2FN, 3FN, FNBC, 4FN, FNPJ, FNS). Éliminer les redondances fonctionnelles au sein d’un ensemble de variables de relations (ou, à défaut, identifier ces redondances afin d’en maintenir la cohérence par une contrainte).

Note

La FNPJ est souvent désignée comme étant la 5FN et la FNS comme étant la 6FN.

À strictement parler, seule la 1FN traite de la normalisation au sens mathématique usuel (toute représentation ne correspond qu’à une et une seule entité, toute entité est représentée par une et une seule représentation).

Toutes les autres formes normales relationnelles sont en fait des critères qui aident à réduire, voire à éliminer, la redondance, donc à concevoir un modèle logique cohérent, complet et évolutif. Les formes normales sont donc des moyens d’appliquer un principe de conception dans le but de minimiser la redondance. Bien sûr, appliquer un moyen (la réduction de la redondance) sur un modèle logique incorrect a peu de chance de le corriger. La normalisation offre cependant l’opportunité de corriger certaines erreurs de conception en les rendant plus manifestes.

La 2FN est insuffisante dans tous les cas et inutile en pratique.

La 3FN et la 4FN sont des formes réduites, des cas particuliers, de la FNBC et de la FNPJ, respectivement. La 3FN peut toutefois s’avérer parfois utile comme étape préliminaire à la détermination de la FNPJ. La 4FN peut toutefois s’avérer parfois utile comme soutien aux entretiens avec les parties prenantes relativement à la FNPJ, lorsque le nombre de dépendances applicables à la détermination d’une FNPJ est limité à deux dépendances complémentaires.

En résumé

Sachant que $FNS \Rightarrow FNPJ \Rightarrow 4FN \Rightarrow FNBC \Rightarrow 3FN \Rightarrow 2FN \Rightarrow 1FN$
le module se concentre sur les formes essentielles suivantes :

- La FNS n’est impérative qu’en cas de relativisation (spatiale ou temporelle) des prédicats.
- La FNPJ est la forme ultime des dépendances fonctionnelles et de la décomposition de projection-jointure. Elle doit être ciblée par tout MLD adéquat.
- La FNBC qui permet de simplifier considérablement l’évaluation de la FNPJ ; elle peut aussi se substituer à la FNPJ en la complétant par des contraintes appropriées lorsque, exceptionnellement, la FNPJ est jugée trop contraignante.
- La 1FN peut (doit) être traitée en premier lieu, car les autres formes normales sont définies en la tenant pour acquise.

Forme normale de la variable ou du type ?

Les formes normales sont généralement définies en termes des variables de relation (du moins, chez les auteurs assez rigoureux pour distinguer les types de relation, de leurs valeurs et de leurs variables).

Ceci est essentiellement dû au fait qu'une contrainte de relation (en particulier, la contrainte de clé déterminante) ne peut être définie que sur une variable au sein de leur modèle relationnel (voir Darwen, Date, de Sainte Marie et Fagin).

Dans le modèle proposé dans le module `TMR_02-Fondements`, tous les types (et pas seulement les types scalaires) permettent d'associer une contrainte à la définition du type. En conséquence, au sein de ce modèle, **toute** contrainte est définie au sein d'un type. En particulier, la contrainte de clé déterminante est associée à la définition du type relation à laquelle elle s'applique. En fait, dans la « machine relationnelle » définie par ce modèle, il n'existe qu'une variable représentant l'état de la machine. Le type de cette variable est un tuplet dont tous les attributs doivent être de type relation. Une clé référentielle (entre deux attributs de types relations) est donc exprimée comme une condition au sein de la contrainte du type de ce tuplet.

Il conviendrait donc de définir les formes normales en termes du type relation (donc applicables à toutes les valeurs légitimes du type) et non en termes de la variable de relation.

Pour éviter d'introduire de la confusion dans l'esprit de nombreux lecteurs, nous utiliserons donc simplement le terme « relation », le lecteur y substituant à son gré le groupe nominal « variable de relation » ou « type de relation », selon le modèle de référence de son choix.

2. Première forme normale

2.1. Définition

Une relation R est en première forme normale (1FN) si et seulement si tous ses attributs sont atomiques.

Un attribut est atomique si et seulement s'il ne peut être associé, en tout temps, qu'à une et une seule valeur de son type de définition.

Soit R une relation $R(a_1{:}T_1,\ldots, a_n{:}T_n)$,

- R est normalisée en regard des attributs si et seulement si
    - $\forall\ t \in R$ la valeur de l'attribut a_i de t est de type $T_i \mid 1 \leq i \leq n$.

On désigne un modèle logique relationnel normalisé en regard des attributs comme étant en « première forme normale » (1FN).

Coloraires

- Un attribut ne peut être sans valeur (pas de marqueur NULL ; l'extension de tous les types en une structure de treillis, comprenant un supremum et un infimum entrainant l'utilisation d'une logique tétra-valuée, demeure toutefois envisageable).
- La valeur d'un attribut est unique (pas d'ensemble de valeurs… à moins que l'attribut ne soit d'un type défini comme un ensemble d'« ensembles de valeurs » — auquel cas, ces derniers « ensembles de valeurs » sont des valeurs atomiques).
- Toute relation est en 1FN par définition (du moins selon le modèle présenté dans le module TRM_02-Fondements).

Observations

- La théorie relationnelle permet l'existence d'attributs non scalaires (tuplets, relations) dans la mesure où les constructeurs de types correspondants sont pris en charge par le modèle relationnel de référence.
- En pratique, les types et les constructeurs de types acceptés dépendent du SGBDR.

2.2. Transformations

Attribut non-clé

Si un attribut non-clé *x* d'une relation *R1* n'est pas en 1FN, car il représente une valeur appartenant plutôt à un *ensemble de T* :

- Retirer *x* de *R1*.
- Ajouter une nouvelle relation totale *R2* dont les attributs sont :
  - les attributs k_1, …, k_n, d'une clé stricte de R1 (préférablement la clé primaire) ;
  - l'attribut *v:T*.
- Pour chaque valeur de la clé choisie,
  - pour chacune des valeurs *v* constitutives de *x*,
    - définir un tuplet $(k_1, \ldots, k_n, v)$ dans *R2*.

Attribut clé

Si un attribut clé *a* d'une relation *R1* n'est pas en 1FN, car il représente une valeur appartenant plutôt à un *ensemble de T* :

- Remplacer *a* par un attribut *k:S* (où *S* est un type scalaire de cardinalité suffisante).
- Ajouter une nouvelle relation totale *R2* dont les attributs sont :
  - l'attribut *k:S* ;
  - l'attribut *v:T*.
- Faire correspondre une (et une seule) valeur de *k* à chacune de valeurs de *a*.
- Pour chaque valeur de *a* répertoriée,
  - pour chacune des valeurs *v* constitutives de *a*,
    - définir un couple *(k, v)* dans *R2*,
      - où *k* est la valeur atomique retenue pour correspondre à celle de *a*.

2.3. Exemple

L'Université désire constituer un répertoire des activités proposées par les enseignants et consigner les inscriptions et les résultats (notes) par étudiant, par activité et par type d'évaluation.

Tableau 1. Enseignant, clé {matricule}

matricule	nom	compétences	adresse
324567	Christina K.	IFT187, IGE487, IGE791	Whitehorse, Yukon
465768	Luc L.	IFT187, IGE487, IFT723	Hearst, Ontario
234810	Jules J.	IFT159, IFT313, MAT115	Montréal, Québec
271841	Alice N.	null	Montréal, Québec

Une erreur fréquente consiste à définir le type d'un attribut comme un texte et d'y encoder plusieurs valeurs.

Tableau 2. Enseignant, clé {matricule}

matricule	nom
324567	Christina K.
465768	Luc L.
234810	Jules J.
271841	Alice N.

Tableau 3. Enseignant_Competence, clé {matricule,activite}

matricule	**activite**
324567	IFT187
465768	IFT187
324567	IGE487
465768	IGE487
324567	IGE791
465768	IFT723
234810	IFT159
234810	IFT313
234810	MAT115

Tableau 4. Enseignant_Adresse, clé {matricule}

matricule	**ville**	**province**
324567	Whitehorse	Yukon
465768	Hearst	Ontario
234810	Montréal	Québec
271841	Montréal	Québec

En fait, ce n'est pas la seule interprétation possible de la relation Enseignant_Adresse. D'autres interprétations seront proposées lors du traitement de la FNBC.

2.4. Pourquoi la 1FN ?

- En fait, tant pour des raisons logiques que pour des raisons de commodité (expressivité) de programmation, il est important de s'assurer que l'égalité est évaluable sur la seule base des **valeurs** définies par les types et non en fonction de leurs **représentations**.
  Par exemple, si on représente un ensemble de cours par un texte, l'égalité de *{IFT187,IFT287}* et *{IFT287,IFT187}* sur la base des représentations `'IFT187,IFT287'` et `'IFT287,IFT187'` devient fausse.
  Le type `Texte` n'est pas le type `Ensemble de cours`.
- La 1FN est en fait l'opérationnalisation de cet axiome
  « équivalence logique » ⇔ « égalité algébrique ».
- Cet axiome ne fait pas partie des axiomes de la **théorie** relationnelle, telle que formulée par Codd (en 1969, comme en 2000). Par contre, dans ses écrits, Codd a rapidement (1970) défini la 1FN et postule le plus souvent que les relations et les MLD considérés sont en 1FN.
- Cet axiome est jugé tellement important qu'il a été inclus dans plusieurs **modèles** relationnels (dont ceux proposés par Date, Darwen, Lorentzos et celui présenté dans le module `TMR_02-Fondements`) où il prend la forme de l'une exigence spécifiant que tout attribut est associé à une et une seule valeur de son type de définition (voir ci-haut).
- Quant à la définition de la 1FN, elle a évolué dans le temps et il n'y a toujours pas consensus, même si Fagin a démontré qu'il n'était pas nécessaire de recourir aux logiques de second ordre pour permettre la définition d'attributs de type non scalaire (dont, en particulier, de type relation) dans la mesure où un tel type n'est pas défini récursivement (en regard de lui-même). L'atomicité (au sens de la non-considération de la représentation) suffit.

3. Forme normale de Boyce-Codd

3.1. Dépendance fonctionnelle

Un attribut A dépend fonctionnellement d'un ensemble d'attributs X si et seulement si (la valeur de) ces derniers permettent toujours de déterminer la valeur (unique) du premier.

Une dépendance fonctionnelle (DF) est notée $X \rightarrow A$ où X est le déterminant et A, le déterminé.

Une DF est **triviale** si le déterminé appartient au déterminant :

- $A \in X$

Une DF est **applicable** à une relation R si le déterminant appartient forme un sous-ensemble des attributs de R et si le déterminé est un attribut de R :

- $(X \subseteq R) \land (A \in R)$

La modélisation logique consiste notamment à déterminer (en fonction du problème) quelles sont les DF applicables devant contraindre chacune des relations.

3.2. Relation FNBC

Une relation R est en FNBC relativement à une DF applicable non triviale $X \rightarrow A$ si et seulement si X comprend une clé stricte de R.

Une relation R est dite en FNBC si et seulement si toutes les DF applicables non triviales à R sont issues d'une des clés de R.

La redondance peut survenir quand plusieurs tuplets ont les mêmes valeurs pour les attributs déterminants, l'attribut déterminé pourrait ne pas avoir la même valeur partout, introduisant ainsi une incohérence.

3.3. Transformation

En décomposant la relation en projections choisies de telle sorte à faire disparaitre les redondances et dont la jointure est égale à la relation d'origine, c'est-à-dire :

Soit $E = \{a_1, \ldots, a_n\}$ l'entête de R, choisir une décomposition $D = \{d_1, \ldots , d_k\}$ telle que

- $d_1 \subseteq E, \ldots, d_k \subseteq E$
- $R = (R \; \pi \; d_1) \bowtie \ldots \bowtie (R \; \pi \; d_k)$
- le nombre de redondances dans D soit le plus petit possible
- s'il existe encore des redondances, les contraindre de façon à ce qu'elle soit cohérente, au mieux identique.

Toute la question se résume donc à choisir la bonne décomposition, c'est-à-dire le bon D. En faisant coïncider les dépendances fonctionnelles avec les décompositions, on réduit la redondance. Malheureusement, la décomposition n'est pas toujours possible, ni toujours efficiente. Des redondances peuvent donc persister dans un modèle logique.

Si une même proposition est stockée en plusieurs endroits, il faut (pour maintenir la cohérence) s'assurer qu'elle est la même partout :

- que les données qui la représentent soient les mêmes partout ;
- que toute modification d'une donnée d'un emplacement est reflétée dans tous les autres emplacements ;
- que tout retrait de la proposition dans un emplacement entraine le retrait dans les autres emplacements.

Pour contrôler les occurrences redondantes, on peut, il faut, les lier par une contrainte.

3.4. Exemple

3.4.1. Enseignant

Tableau 5. Étudiant

matricule	nom
123456	Jeanne
124789	Jean
107834	Odette
203040	Tania
311211	Jean
234567	Mehri
312101	Tangui

DF1 : matricule → nom
Étant donné le matricule d'un étudiant, on peut déterminer son nom et ce nom est unique. L'inverse n'est pas vrai : étant donné un nom, on ne peut (en général) déterminer le matricule, car il peut y avoir plusieurs enseignants de même nom, chaque enseignant ayant un matricule distinct.

- Cela ne signifie pas que le nom associé à un matricule ne change jamais ; le nom peut changer, mais, en tout temps, on peut déterminer le nom d'un enseignant à partir de son matricule.
- Cela ne signifie pas non plus qu'à deux matricules différents, les noms associés doivent être différents (quoiqu'ils puissent l'être).
- Cela signifie qu'à un instant donné, une variable de relation (relvar) n'associe qu'un et un seul nom à un matricule.

Transformation requise

Aucune… si ce n'est de déclarer la bonne clé déterminante !

rel Étudiant {matricule, nom} ***det*** {nom}

3.4.2. Activité

Tableau 6. Activité

code	no	titre	credit	département
IFT	187	Éléments de bases de données	3	Informatique
IMN	117	Acquisition des médias numériques	3	Informatique
IGL	487	Modélisation de bases de données	3	Informatique
IFT	723	Sujets approfondis en bases de données	3	Informatique
IFT	697	Projet d'intégration et de recherche	6	Informatique
BIO	101	Biométrie	3	Biologie
PHY	131	Optique	2	Physique

DF2 : code → département
Étant donné le code d'une activité, on peut déterminer le département responsable. L'inverse n'est pas vrai : étant donné un département, on ne peut (en général) déterminer un code, car il y a plusieurs codes à un même département.

DF3 : code, no → titre, credit Ni le code seul ni le numéro (no) seul ne sont suffisants pour déterminer le titre d'un cours ou son nombre de crédits. Les deux attributs sont requis simultanément pour identifier un cours.

À l’inverse, ni le titre, ni le nombre de crédits, ni les deux ensemble ne permettent de déterminer le code du cours pas plus que son numéro.

Transformation requise

Il faut donc décomposer *Activité* en deux, les attributs de l’activité elle-même et ceux de la discipline.

> ***rel*** Discipline {code, departement} ***det*** {code}
> ***rel*** Activité {code, no, titre, credit} ***det*** {code, no} ***ref*** {code} → Discipline

Tableau 7. Discipline

code	departement
IFT	Informatique
IMN	Informatique
IGL	Informatique
BIO	Biologie
PHY	Physique

Tableau 8. Activité

code	no	titre	credit
IFT	187	Éléments de bases de données	3
IMN	117	Acquisition des médias numériques	3
IGL	487	Modélisation de bases de données	3
IFT	723	Sujets approfondis en bases de données	3
IFT	697	Projet d’intégration et de recherche	6
BIO	101	Biométrie	3
PHY	131	Optique	2

3.5. Et la 3FN ?

Une relation R est en FNBC relativement à une dépendance fonctionnelle (DF) applicable non triviale $X \rightarrow A$ si et seulement si une des conditions suivantes est vraie

- X comprend une clé stricte de R (condition FNBC) ;
- A appartient à une clé stricte de R (relaxation 3FN).

3.6. Pourquoi la FNBC et pas seulement la 3FN ?

Pour certains, le choix entre la 3FN et la FNBC repose sur l’évaluation des éléments suivants (dans l’ordre) :

- la prise en compte de contraintes supplémentaires (ressources, délais, etc.) ;
- la perte (éventuelle) d’efficience de la BD en raison de l’ajout des contraintes associées à la FNBC ;
- l’impact de l’absence de ces contraintes sur la cohérence, la validité et l’efficacité de la BD.

Ce qui pourrait se résumer ainsi : « Est-il préférable d’arriver parfois un peu plus rapidement à une réponse possiblement inexacte ou d’arriver parfois un peu plus lentement à une réponse certainement exacte ? ».

Concrètement, si votre résultat de laboratoire médical, la surcharge du transformateur électrique de votre quartier, le cout total de votre commande en ligne ou la fréquence des rappels de notification à votre poste de travail en dépend, que préférez-vous ?

Et non, contrairement à ce qu’affiche un certain assureur, ce n’est pas vrai qu’on peut tout réparer !

4. Forme normale de projection-jointure

La FNBC (comme la 2FN et la 3FN) a été formulée de façon à respecter au mieux les dépendances fonctionnelles en recourant, au besoin à des décompositions binaires par projection-jointure.

La FNPJ a été élaborée à la suite de la constatation que certaines redondances ne pouvaient pas être éliminées par décomposition binaire et que des décompositions de degré supérieur pouvaient être requises (en fait, Codd l'avait noté dès 1969, mais la communauté informatique a longtemps négligé ce problème).

La FNPJ est fondée sur la notion de dépendance de jointure et couvre tous les cas de redondance solubles par décomposition de projection-jointure (pour plus de détails, voir les théorèmes de Fagin), la formalisation antérieure de la 4FN (ou forme normale multi-valuée) s'avérant au final un cas particulier de la FNPJ.

Plusieurs autres formes normales ont été développées par ailleurs en regard d'autres formes de redondance, dont, en particulier, la 6FN, qui garantit l'indépendance des prédicats élémentaires et que nous présenterons dans la section suivante.

4.1. Dépendance de jointure

La dépendance de jointure (DJ) est aussi appelée dépendance de projection-jointure (DPJ).

Définition

Une dépendance de jointure (DJ), notée $\bowtie\{D_1, \dots, D_k\}$ où $D_1, \dots, D_k$ sont des ensembles d'identifiants d'attributs, est vérifiée par R si et seulement si :

- $D_1 \subseteq id(R), \dots, D_k \subseteq id(R)$ (applicabilité)
- $R = (R\ \pi\ D_1) \bowtie \dots \bowtie (R\ \pi\ D_k)$ (assertion)

Autrement dit, une DJ est vérifiée par une relation R si elle peut être décomposée **sans perte** selon les D_i.

Remarque 1 — corolaire

Une DJ vérifiée par R implique que l'union des D_i est égale à $id(R)$ (l'ensemble des attributs de l'entête de R).

Remarque 2 — DJ triviale

Une dépendance de jointure $\bowtie\{D_1, \dots, D_k\}$ vérifiée par R est **triviale** si et seulement si au moins un des D_i est égal à $id(R)$.

Une DF vérifiée par R est **strictement triviale** si et seulement si elle est composée du seul élément $id(R)$.

Remarque 3 — induction par les clés

Une DJ est **induite par les clés** de R si chaque D_i est une clé de R (c'est-à-dire, contient une clé stricte de R).

4.2. Relation FNPJ

Une relation R est en forme normale de projection-jointure (FNPJ ou 5FN) si et seulement si chacune des dépendances de jointure non triviales qui lui sont applicables est induite par les clés de R.

Théorème 1

Toute relation en 3FN (*a fortiori* en FNBC) dont au moins un attribut ne participe à aucune clé stricte de R est en FNPJ.

Théorème 2

Toute relation en FNBC dont tous les attributs forment ensemble une clé stricte est en FNPJ.

4.3. Transformations

Si une relation R en FNBC n'est pas en FNPJ, deux transformations peuvent être envisagées pour corriger la situation :

- Remplacer la relation R par une vue calculant la jointure sur la base des sous-relations correspondant aux clés strictes de R.
- Garder la relation R et ajouter une contrainte pour vérifier que la jointure des relations D_i est égale à la relation (cette contrainte pouvant souvent être mise en oeuvre par l'ajout des clés référentielles appropriées).

4.4. Exemple

Soit le modèle conceptuel de données suivant [1] :

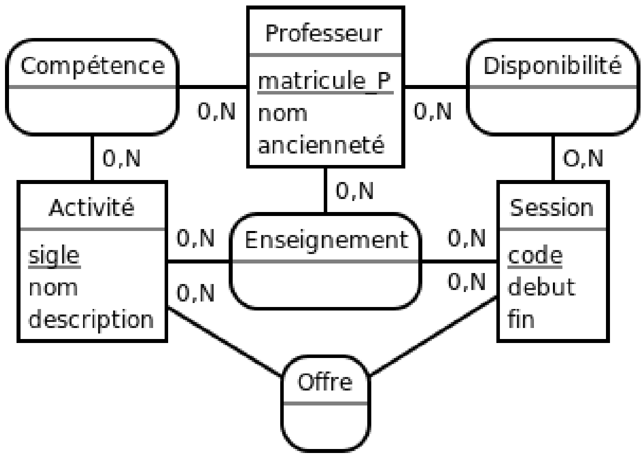

Figure 1. MCD — Université : exemple 5FN

Liste des prédicats

Professeur {matriculeP, nom, ancienneté} clé {matriculeP}

Le professeur dont le matricule est « matriculeP », le nom est « nom » enseigne à l'UdS depuis « ancienneté » an.

Activité {sigle, nom, description} clé {sigle}

L'activité « sigle », décrite par le nom « nom » et la description « description », est offerte par l'UdeS.

Session {code, debut, fin} clé {code}

La session « code » débute le « debut » et se termine le « fin ».

Compétence {matriculeP, activite} clé {matriculeP, activite}

Le professeur « matriculeP » est apte à enseigner l'activité « activite ».

Disponibilité {matriculeP, session} clé {matriculeP, session}

Le professeur « matriculeP » est disponible pour enseigner à la session « session ».

Enseignement {matriculeP, activite, session} clé {matriculeP, activite, session}

Le professeur « matriculeP » enseigne l'activité « activite » à la session « session ».

Offre {activite, session} clé {activite, session}

L'activité « sigle » est programmée à la session « session ».

Liste des contraintes

- Un professeur ne peut pas enseigner une activité qu'à condition qu'il soit apte à le faire.
- Un professeur ne peut pas enseigner une activité à une session donnée qu'à condition qu'elle soit offerte à cette session.

- Un professeur ne peut pas enseigner une activité à une session donnée qu'à condition qu'il soit disponible à cette session.

Voici une instance de base de données valide :

Tableau 9. Compétence

matriculeP	activite
324567	IFT187
324567	IGE487
465768	IFT187
465768	IGE487

Tableau 10. Disponibilité

matriculeP	session
324567	2024-3
465768	2025-1

Tableau 11. Offre

activite	session
IFT187	2024-3
IFT187	2025-1
IGE487	2024-3

Tableau 12. Enseignement

matriculeP	activite	session
324567	IGE487	2024-3
465768	IFT187	2025-1

Soit

- DJ : ⋈ ({matriculeP, session}, {matriculeP, activite}, {activite, session})

Qui correspond à la contrainte suivante :

- Si un professeur est apte à enseigner une activité offerte pour une session, alors le professeur enseigne cette activité à cette session.

Est-ce que *Enseignement* = *Disponibilité* ⋈ *Compétence* ⋈ *Offre* ?

Non, l'instance ne respecte pas la contrainte :

- L'activité "IFT187" est offerte à la session "2024-3".
- Le professeur "324567"
  - est apte à enseigner "IFT187" et
  - est disponible la session "2024-3",
  - mais **il n'enseigne pas "IFT187" la session "2024-3"**.

Alors, la relation Enseignement n'est pas en FNPJ.

5. Forme normale stricte

La forme normale stricte (FNS) est souvent désignée comme sixième forme normale (6FN).

Une relation R est en FNS si et seulement si chacune des dépendances de jointure applicables est triviale.

Ceci a pour effet de placer chaque DF non triviale dans une relation propre. Une telle contrainte est nécessaire lorsqu'une relation doit être relativisée localement (en particulier, dans le cas des chronologies).

Ceci doit être généralisé dans le cadre de la relativisation (spatiale, temporelle ou spatio-temporelle) complète d'une BD. Ceci est traité dans le module `TMR_09-Historicisation`.

6. Retour sur la modélisation

La modélisation consiste notamment à déterminer (en fonction du problème, du modèle de connaissance, du MCD) quelles sont les dépendances applicables (DF et DJ) devant être respectées.

Elles le sont très rarement et, quand elles le sont, elles le sont le plus souvent partiellement.

Quand on en arrive à la modélisation conceptuelle, on s'en tire le plus souvent en affirmant que les seules dépendances à prendre en compte sont celles (logiquement) induites par les clés. Magie ! Tout MLD est alors, par définition, automatiquement en FNPJ.

Conclusion

- Un modèle relationnel est normalisé en regard d'une forme X si et seulement si toutes ses relations sont normalisées en regard de la forme X.
- Une relation R respectant ses contraintes de type est en 1FN.
- Une relation R est en FNBC si et seulement si chacune des dépendances fonctionnelles qui lui sont applicables est induite par une de ses clés.

  Corolaire

  Une relation R en FNBC est également en 1FN.
- Une relation R est en FNPJ si et seulement si chacune des dépendances de jointure qui lui sont applicables est induite par une de ses clés.

  Corolaire

  Une relation R en FNPJ est également en FNBC.

  Corolaire

  Une relation R en FNBC qui comporte au moins un attribut non-clé est en FNPJ.

  Corolaire

  Une relation totale R en FNBC ne comportant aucune clé composite est en FNPJ.
- Une relation R est en FNS si et seulement si chacune des dépendances de jointure applicables est triviale.

  Corolaire

  Une relation R en FNS est également en FNPJ.

Références

[Elmasri2016]
Ramez ELMASRI et Shamkant B. NAVATHE;
Fundamentals of database systems;
7th Edition, Pearson, Hoboken (NJ, US), 2016;
ISBN 978-0-13-397077-7.

[Date2014a]
Chris J. DATE, Hugh DARWEN, Nikos A. LORENTZOS;
Time and Relational Theory: Temporal Databases in the Relational Model and SQL;
Morgan Kaufmann, Waltham (MA, US), 2014;
ISBN 978-0-12-800631-3.

[DeSainteMarie2012]
François de SAINTE MARIE;
Bases de données relationnelles et normalisation : de la première à la sixième forme normale;
Site Developpez.com, 2012;
http://fsmrel.developpez.com/basesrelationnelles/normalisation/

Produit le 2025-09-10 12:01:55 UTC

Université de Sherbrooke

Université de Sherbrooke

[1] (adaptation libre de DeSainteMarie2012)

