資　料

データベースの関係モデル第3正規形批判*

植　村　俊　亮**

Abstract

The disadvantages of the notion of the Third Normal Form (of relational model) are discussed. (1) TNF is not adapted to 1 to n or n to n relationship between data. (2) Duplicated keys in TNF relations may cause updating inconsistency. (3) Data independence can not be achieved by normalization alone. A well designed data manipulation language is needed. Further, keeping data base always in TNF may require extra effort, which in turn may cause data dependence.

1. はじめに

データベースシステムに関する研究分野のうちで近年もっともはなばなしい脚光をあびたのは，E.F. コッド (Codd) によるデータベースの関係モデル (relational model) 理論であろう．数学概念としての関係をデータ構造と結び付ける着想自体は目新しいものではなかったが，コッドはこれをデータベース，とくにデータ独立 (data independence) の概念とからませて説得力豊かな関係モデルを提起した．

関係モデル理論の展開にあたって，コッドはまず関係の正規化 (normalization) を論じた．関係の正規化は3段階にわたって行われ，その究極の形は第3正規形 (third normal form) と呼ばれる．第3正規形の関係は，関係モデル理論に直接たずさわる人のみならず広くデータベースシステム社会一般において，もっとも基本的で単純な関係としてもてはやされるに至っている．

本稿では，関係の正規化をコッドの観点とは視点を変えて分析し，第3正規形をデータベースのデータ構造の基礎とすることにはかなり疑問があり，場合によっては有害になりかねないことを論じる．主要な論点は以下の3点である．

(1) 現実の世界でしばしば現われるデータ相互間の1対n，n対n対応が，正規化の概念となじまないこと．

(2) 正規化は冗長な情報を追加しつつ関係を細分する形で行われるため，データ保全 (data integrity) の確保を困難にするおそれがあること．

(3) 正規化したからといって，それだけでデータ独立が保証されるわけではないこと．

2. 関係の正規化

コッドは関係モデル提案のごく初期から，関係にある種の正規形があることを指摘し，のちにこの視点をさらに進めて，正規化されていない関係から第3正規形にいたる3段階の正規化手続きを提示した．

ここではまず，以下の議論に必要な範囲でコッドの正規化理論[2),4)]を要約する．一部にコッド以外の人による改良された定義をも含んでおり[6),7)]，また数学的な厳密さはめざしていない．

n個の集合$S_1, S_2, \ldots, S_n$(同じものがあってもよい)の直積 (direct product) の部分集合を関係 (relation) Rと呼ぶ．これは数学における関係の定義そのものである．S_jを関係Rのj番目の定義域 (domain) と呼ぶ．関係を構成する各定義域を関係の属性 (attribute) とみなして，各定義域の名前を属性名 (attribute name) と呼ぶこともある．さきのRはn次 (degree n) の関係，すなわちn項関係である．n項関係の一つの要素をn個組 (n-tuples) あるいは単に組と呼ぶ．

データベースは関係の集まりである．おおまかにい

---

* Third Normal Form Considered Harmful by Syunsuke UEMURA (Computer Science Division, Electrotechnical Laboratory).

** 電子技術総合研究所ソフトウェア部言語処理研究室

えば，関係はファイル，組はレコードに相当する．データベースに新しい種類のデータが追加されても，関係の集まりはなるべくもとのまま使えるようにしたいし，組の追加，更新，削除も円滑に行えるようにしておきたい．このために関係を正規化する．

例として，部品番号，部品名，倉庫所在地，在庫量，作業の五つの定義域（属性）からなる部品という関係を考える．これを，

部品（部品番号，部品名，倉庫所在地，在庫量，
　　作業）

と書く．関係はしばしば表形式でも表現される．たとえば，部品という関係は以下のようにも表わされる．

部　品

部品番号	部品名	倉庫所在地	在庫量	作業
001	ビス	横浜	100	・・・
003	ナット	東京	500	・・・
005	ボルト	千葉	200	・・・
007	ボルト	千葉	200	・・・
⋮	⋮	⋮	⋮	⋮

一般的な関係においては，その一つの定義域自体がまたn項関係でありうる．上述の作業が実は，

作業（作業番号，作業名，使用量）

という関係であってもよいので，この場合部品関係は，

部品（部品番号，部品名，倉庫所在地，在庫量，
　　作業（作業番号，作業名，使用量））

となる．これは正規化されていない関係の例である．

ある関係でどの定義域を取り出しても，その要素が原子的な (atomic) 値であるとき，この関係は第1正規形 (first normal form, 1 NF) である，あるいは単に正規化されている (normalized) という．原子的な値とは，一つの数値，一つの文字列などを指す．部品関係を以下の二つの関係に分割すれば，それぞれ第1正規形となる．

部品1（部品番号，部品名，倉庫所在地，在庫量）
作業1（部品番号，作業番号，作業名，使用量）

第2，第3正規形の中心となる概念は関数従属 (functional dependency) である．関係Rの任意の定義域（の集まり）Xの要素の値xが定まると，対応する定義域Yの要素の値yが一意に定まるとき，YはXに関数従属であるという．YはXの関数であると考えればよい．

関係Rが第1正規形であるとする．Rの任意の定義域(の集まり)Cを考える．どのようなCをとっても，C以外の定義域はすべてCに関数従属でないか，すべてCに関数従属であるとき，Rは第3正規形であるという[7]．

関係Rのすべての組を一意に識別する定義域（の集まり)のうち，必要最小限のものをキー候補 (candidate key) という．必要最小限とは，キー候補Kのどんな真部分集合をとっても一意識別性を失ってしまうという意味である．Kは古典的なファイル編成法でいう主キー (primary key) に相当する概念で，関係モデルにおいてもキー候補のうちの任意の一つを選んだとき，それを主キーという．第3正規形の着想は，「属性Aの値が属性Pの値によってきまる（AがPに関数従属である）のであれば，AはPを主キーとする関係中に含まれるべきである．さらにこの関係には，属性Pの値によってだけ値がきまるような属性ばかりをまとめるようにするとよい．」というところにある．こうしておけば，一つの関係中に属性Aの値を冗長に繰り返して記録しなくてもすむし，関係の組の更新，追加，削除にともなう波及効果を最小限にとどめることができる．

関数従属であるかどうかの判定は属性の名前やその値をみただけでくだせるわけではない．それは属性相互間の現実世界における論理的な関係を十分に把握してはじめて判断可能になる性質のものである．いま，部品1という関係において，部品番号がきまると部品名，倉庫所在地，在庫量がそれぞれ一意にきまり，さらに部品名がきまると倉庫所在地が一意にきまるとする．また作業1という関係において，部品番号と作業番号との組み合わせがきまると使用量が一意にきまり，作業番号がきまると作業名が一意にきまるとする．そしてこれ以外には属性間に依存関係がないとすると，部品1，作業1は以下の第3正規形に展開できる．

部品3（部品番号，部品名，在庫量）
部品倉庫（部品名，倉庫所在地）
作業3（作業番号，作業名）
部品使用量（部品番号，作業番号，使用量）

下線はそれぞれの関係における主キーを示す．たとえばさきの作業1とこの作業3とにおける作業名という属性を比較すれば，第3正規形の意義は明らかであろう．作業1では，おなじ作業名がなん回も繰り返し現われるが，作業3では1回ずつ現われるだけである．また使用量が0になって，作業1の特定の組を削除すると，波及効果として作業番号と作業名との間の対応関係を示す情報が消滅するおそれがあるが，第3正規形ではその心配もない．

なお第3正規形の数学的に厳密な定義にはいくつかの変形があり[2),4),6),7)]，それらが相互に等価でないという議論もある[9)]．さきに示した定義はボイス－コッド (Boyce and Codd) の定義と呼ばれるもので，もっとも基本的な定義と考えられる．ここではこれ以上の詳細には立ち入らない．

またおおまかにいって，第3正規形の関係では主キーでない属性は互いに独立であるが，主キーでない属性の間に関数従属の関係を認めると第2正規形になる．第2正規形には本質的な重要性はないと考えられるので，これについても詳細は省略する．

3. 第3正規形批判

関係の正規化は，データベース全体をいくつかの互になるべく独立で単純な構造のファイルに分割していく作業とみなすことができる．従来の古典的なファイル編成技法によっても，直観的に第3正規形に到達することは十分にありえた．事実コッドが関係モデル理論に関する最初の論文[1)]で，access path dependent (データの呼出し経路に依存する) であるとして非難したデータ構造のうち Structure 5 (参考文献 1)，378 ページ) は第3正規形そのものである．プログラムの段階的な作成手法などについても言われるように，「経験豊かなプログラマが無意識のうちに採用していた手法に理論的な洞察を与えた」点で第3正規形は十分に評価されるべきである．しかしそれはけっしてデータベースのデータ構造の基礎として絶対視されるべき性質のものではない．

以下に第3正規形に対する筆者の批判的な分析をまとめる．

(1)　正規化の概念になじまないデータ構造が現実の世界にしばしば存在する．関係モデルはデータの集合論的な取り扱いを基礎として出発しながら，正規化の段階では組のキーによる一意識別性を中心的な概念にすえてしまって，集合論的なおもしろさをみずから放棄している．

階層構造における上位階層と下位階層の1対n対応を例として考えよう．たとえば，

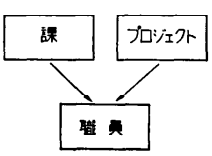
社員 (部，課，職員)

という関係がある．部と課，課と職員はそれぞれ1対nの対応をしている．部の名前を与えて，その部に属する職員を列挙する．あるいは部と課の名前を与えて，その課に属する職員のうちからある条件を満たす者を選び出すといった応用が考えられる．

部	課	職員
総務	人事	岸
		佐藤
	給与	伊藤
		大久保
		福田
営業	営業1	小沢
		岩城
		外山
	営業2	浅利
		石原

部	課	職員
総務	人事	岸
総務	人事	佐藤
総務	給与	伊藤
総務	給与	大久保
総務	給与	福田
営業	営業1	小沢
営業	営業1	岩城
営業	営業1	外山
営業	営業2	浅利
営業	営業2	石原

Fig. 1　1-to-N Correspondence and Relation



課	プロジェクト	職員
営業1	L販売	伊藤
営業1	L販売	片山
営業2	L販売	大久保
経理	L販売	福田
営業1	M特売	伊藤
⋮	⋮	

Fig. 2　Network Structure

関係モデルでこれを

社員 (部，(課，職員))

とすると，正規化されていない関係ができる．冗長な繰り返しを導入して，これを第1正規形にできる (**Fig. 1** 参照)．第2，第3正規形への正規化は関数従属すなわちデータの関数的な対応を中心にして行われる．しかしこの例では，関係の組を一意に識別する属性をしいてあげれば (部，課，職員) 全体がそれに相当し，したがってこの第1正規形は同時に第3正規形でもあることになる．データの1対n対応の部分，すなわち部がきまるとその部に属する課がいくつかきまるといったデータ構造は正規化とはうまく結びつかない．

いいかえれば，正規化とはデータの冗長性を意に介さずにデータ間の関数対応を構築することであるともいえる．**Fig. 2** の網(network)構造の例はこれを示している．Fig. 2 左のデータ構造において，ある課に属する職員を順にたどってそれぞれがどんなプロジェクトに属しているかを調べるとか，あるプロジェクトに参加している職員の属する課を調べるといった応用が考えられる．データ間のこのような関係は第3正規形ではおよそ Fig. 2 右のように表現されるのであろう．すなわち存在しうるすべての課－プロジェクト－職員という対応 (順序はどうでもよい) を列挙する．この表をいかに縮小するかとか，求める組み合わせをいかに能率よく探し出すとかといったことは，すべて議論の対象としないのである．

こうした不自然さの原因は，第2正規形，第3正規形の中心になっている関数従属の概念が基本的にデータの関数的な対応を中心とし，1対nあるいはn対m対応を無視している点にその根本原因があると考えられる．また関数従属の概念によって1対1対応を表現しようとすると，AはBに関数従属であり，かつBはAに関数従属であるとしなければならない．AとBとがともにキー候補であれば，このような1対1対応が成立し，しかもA，Bを含む関係が第3正規形でありうるが，いったんここに主キーの概念を導入すると，1対1対応の表現は困難になる．AとBとが同時に主キーであることはできないためである．

(2) 正規化は冗長な情報を導入しつつ関係を細分する形で行われるため，データ保全 (data integrity) の確保を困難にする．さきの例では，もとの関係は，

部品（部品番号，部品名，倉庫所在地，在庫量，
　　作業（作業番号，作業名，使用量））

であった．これを第3正規形に正規化すると，

部品3（部品番号，部品名，在庫量）
部品倉庫（部品名，倉庫所在地）
作業3（作業番号，作業名）
部品使用量（部品番号，作業番号，使用量）

となった．定義域の個数の増加（種類は増加していない）は一見して明らかである．冗長度がふえること自体はそれに見合うだけの利点があれば許されよう．とくに関係モデルはデータの冗長度やデータ操作の能率などに対する配慮を一切ぬきで議論を進めており，単に冗長であるというだけの批判ではかみ合わないかもしれない．

筆者はその冗長部分のデータの質を問題にしたい．すなわち関係の正規化においては，関係を分割するにあたって上位構造の主キーを原則としてそのまま下位構造に重複記録させている．部品番号，作業番号といったキーをなん重にも記録する．これらのデータはキーであるだけに無矛盾に保つことが重要である．しかしこれらのデータは応用と密着した内容なので，修正をよぎなくされるおそれがつきまとう．

ふつうのファイル編成（や第1正規形の関係）では，キーはたいてい一度だけ現われるので，かりにキー修正が必要になっても，一度の修正で作業が完了する（それでも一意識別性に十分留意しなければならないことはいうまでもない）．第3正規形の関係では，同じキーが複数個の関係に重複して現われるので，キーの修正にともなうデータ保全が容易でない．作業3の主キーである作業番号は，部品使用量という関係中ではなん回か繰り返し現われる点にも注意すべきである．一つの関係中のキーを修正してべつの関係中の同じキーを修正するまでの間の矛盾，修正を複数回行うことによって生じる誤り発生の可能性の増大などは無視できない問題である．

コッドは ALPHA 言語[3),5)] において，主キーの変更は他のデータの更新とは質の異なる重要な作業であるとして，特別の命令（の組）を想定している．しかし第3正規形がもたらす以上のような危険性については，なにもふれていない．

(3) 正規化だけによってデータ独立が保証されるわけではない．これはさきの Structure 5 の例からも明らかであろう．第3正規形の関係ばかりからなるデータベースが存在するとしても，それだけでデータベースが呼出し経路から独立になるわけではない．とくに関係の正規化の各段階で，もとの関係はいくつかの互いに独立で単純な関係に分解されてしまうので，すこし複雑なデータの呼出しは複数個の関係を操作してはじめて可能になる．たとえば，「作業Aに必要な部品の名前」を求めるためには，三つの関係にわたる演算が必要である．このような場合のデータ独立性は，まず関係相互にわたる演算が十分に可能なデータ操作言語が存在してはじめて達成される．

コッドの ALPHA はそうした言語である．しかし ALPHA でも，プログラム中で関係や定義域の名前を直接引用しているので，ALPHA によるプログラムが正しく動作するためには，これらの関係や定義域が存在しているという前提が満足されなければならない．プログラマの立場からいえば，データベース中の関係の名前や定義域の名前を知っていなければならないし，彼のプログラムのデータ独立性はけっして保証されてはいない．

関係の第3正規形は時間，環境の変化にともなって第3正規形でなくなってしまう可能性が十分にあることにも留意しなければならない．さきの例では，ある部品を格納する倉庫が在庫量の増大にともなって，二つの別の場所の倉庫に分散収容される事態が生じると，部品倉庫という関係は第3正規形でなくなる．これをしいて第3正規形を保つべく努力すれば，新しい関係に分割するなどの事態が生じるであろう．その場合には，もとの関係を参照していたプログラムは修正をよぎなくされる（ALPHA とても同様である）．すなわち第3正規形にこだわるためにかえってデータ独

立をそこなうことになる.

3. 結 言

関係モデルはきわめて単純で平坦 (flat) なデータ構造の提案であり，理論的に美しく整っている．しかしそうだからといって，現実社会に存在するデータ構造のあやを強力に把握できるとはかぎらない．本稿でのべた第3正規形の問題点はその弱点の例である．事務計算など現実の応用を意識して関係モデルによるシステム作成を行おうとすると，かならずこの種の障害に直面するであろう．関係モデルを補うために，より現実的なデータ構造を導入するか，さらに別の手段をこうじるかは議論のわかれるところであろう．データセマンティクスと呼ばれる分野は後者の方向をめざしていると考えられる．

いつも有益な討論を通して本稿のきっかけを与えていただいたデータベースモデル研究委員会の諸氏，SSL の穂鷹良介氏，日本 IBM の渋谷政昭氏，当所言語処理研究室の鳥居宏次室長，室員諸氏に感謝いたします．

参 考 文 献

1) E. F. Codd: A Relational Model of Data for Large Shared Data Banks, Comm. ACM, Vol. 13, No. 6, pp. 377-387 (1970).
2) E. F. Codd: Further Normalization of the Data Base Relational Model, Courant Computer Science Symposium 6 May 24-25, 1971 "Data Base Systems", pp. 33-64, Prentice-Hall (1972).
3) E. F. Codd: Relational Completeness of Data Base Sublanguages, ibid., pp. 65-98.
4) E. F. Codd: Normalized Data Base Structure: A Brief Tutorial, Proc. of ACM SIGFIDET Workshops on Data Description, Access, and Control, pp. 1-17 (1971).
5) E. F. Codd: A Data Sublanguage Founded on the Relatioal Calculus, ibid., pp. 35-68.
6) W. Kent: A Primer of Normal Forms (in a Relational Data Base), IBM System Development Division TR 02.600 (1973)
7) E. F. Codd: Recent Investigations in Relational Data Base Systems, Information Processing 74, pp. 1017-1021, North-Holland (1974).
8) C. J. Date: An Introduction to Database Systems, xvii+366 pp., Addison Wesley (1975).
9) 穂鷹良介：第3正規形について，情報処理投稿中

(昭和51年4月30日受付)
(昭和51年6月4日再受付)