

ソフトウェア応用演習 正規化について

堀田 敬介

2001.9.28
Revised 2002.9.26

1 正規化

正規化 (normalization) とは、データの重複 (冗長性) を廃し、データの一貫性と整合性を持った理解しやすい構造にする手法。リレーショナル・データベース (関係データベース) 設計において、源データをデータ項目間の依存関係 (従属関係) が小さい複数のテーブル (表) に再構築する。一般的なデータベース設計では第 3 正規形まで行えば充分であるといわれている (第 5 正規形までである)。

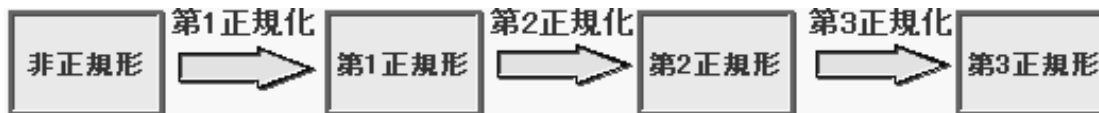


図 1.1: 正規化

それに対し、データの重複 (冗長性) や、従属関係は何も考慮されていない形でデータを保持しているもの、即ち元のままのデータの羅列で構成されている表 (テーブル) を非正規形と言う。¹

1.1 非正規形の問題点と正規化のメリット

非正規形の問題点

- 繰り返し部分が多い。冗長部分が多くデータ保持の観点から効率が悪い。
(例) 文教太郎君の学籍番号・名前・学部・学科という同じデータを重複して保持している。

¹正規化の厳密な定義は 2 節で与える。

表 1.1: 非正規形

学籍番号	名前	所属	Tel	科目コード	科目名	単位	年度	成績
A0P21000	文教太郎	情報・経情	xx-xxxx	213905	ソフトウェア演習	4	2000	A
A0P21000	文教太郎	情報・経情	xx-xxxx	213913	ソフトウェア応用演習	4	2000	B
A0P21000	文教太郎	情報・経情	xx-xxxx	443909	ネットワーク演習	2	2000	C
A0P21000	文教太郎	情報・経情	xx-xxxx	233917	意思決定論	4	2001	B
A0P21999	湘南次郎	情報・経情	xx-xxxx	213905	ソフトウェア演習	4	2000	A
A0P21999	湘南次郎	情報・経情	xx-xxxx	323926	ゼミナール	3	2000	B
A0P11999	藤沢花子	情報・広報	xx-zzzz	213906	ソフトウェア演習	4	2001	F
A0P11999	藤沢花子	情報・広報	xx-zzzz	213913	ソフトウェア応用演習	4	2001	A
A0P11999	藤沢花子	情報・広報	xx-zzzz	443909	ネットワーク演習	2	2001	B
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

- 従属関係を考慮していないので、更新が容易にできない。
 (例) 文教太郎君が学科を代わった場合、対応するデータ全てを変更しなければならない。
 (例) ソフトウェア演習(コードが213905のみ)の単位が4→2と変更になったときに全てを変更するのが大変。
- 従属関係を考慮していないので、データの整合性を保つのが大変(更新・新規レコード入力時のミスにより、同じデータのはずなのに、違う値が入ってしまう)。(例) 学科の電話番号を変更する(xx-xxxx→xx-x00x)際、あるレコードだけ(xx-xxxx→xx-xx0x)にしてしまったため、整合性がなくなる。
 (例) 新規レコードを追加入力した際に、科目コードを間違えて入れてしまい、他のレコードとの整合性がとれない。
- その他...

正規化のメリット

- データを効率的に持てるため、物理的容量が少なくて済む。
- データを共通化して持てる。業務ごとに専用のファイルを使用してデータを持つこともできるが、データの管理に無駄・矛盾がおき、また非効率である。
- 共通項目・関連項目を一つのファイルにまとめることで一元管理できる。
- データの追加・更新が容易。

1.2 正規化をしてみよう！

では実際に、非正規系から第1, 第2, 第3正規形への正規化の仕方を、先にあげた例を使って、各々の説明と共に進めてみよう。

第1正規形 非正規形に対し、第1正規化を行い第1正規形を得る。第1正規形とは、各属性に対し重複した値を持たない形のこと。非正規形のデータを見て、キーを設定し(ex. 学籍番号)、繰返し部分と固定部分に分ける(ex. 固定部分 ... 学籍番号, 名前, 学部, 学科, 繰返し部分 ... 科目コード, 科目名, 単位, 年度, 成績) 繰返し部分には、識別できるようにキー(ex. 学籍番号)を付加する(各行が識別できるように(ex. 科目コード)もキーにする。)各表では属性名に下線が引いてあるのが主キーである。

表 1.2: 第1正規形

表 1.3: 固定部分

<u>学籍番号</u>	<u>名前</u>	<u>所属</u>	<u>Tel</u>
A0P21000	文教太郎	情報・経情	xx-xxxx
A0P21999	湘南次郎	情報・経情	xx-xxxx
A0P11999	藤沢花子	情報・広報	xx-zzzz
⋮	⋮	⋮	

表 1.4: 繰返し部分

<u>学籍番号</u>	<u>科目コード</u>	<u>科目名</u>	<u>単位</u>	<u>年度</u>	<u>成績</u>
A0P21000	213905	ソフトウェア演習	4	2000	A
A0P21000	213913	ソフトウェア応用演習	4	2000	B
A0P21000	443909	ネットワーク演習	2	2000	C
A0P21000	233917	意思決定論	4	2001	B
A0P21999	213905	ソフトウェア演習	4	2000	A
A0P21999	323926	ゼミナール	3	2000	B
A0P11999	213906	ソフトウェア演習	4	2001	F
A0P11999	213913	ソフトウェア応用演習	4	2001	A
A0P11999	443909	ネットワーク演習	2	2001	B
⋮	⋮	⋮	⋮	⋮	⋮

第2正規形 第1正規形に対し、キー属性と他の属性との従属関係を調べる。キーは、ひとつの属性だけでキーになる、即ち、そのキーだけでレコード(行)を特定できるもの(ex. 固定部分の学籍番号)と、複数の属性キーでひとつのキーになる連結キー(ex. 繰返し部分の学籍番号と科目コード)がある。

第2正規化とは、連結キーを持つ表に対し、各属性がキーに完全従属（または、関数従属）している属性群と部分従属（または一部のキーに関数従属）している属性群に分けること。

連結キーに完全従属 連結キー全てを用いないと、その行を特定できないこと。

連結キーに部分従属 連結キーの一部が決まれば、その行を特定できること。

表 1.5: 第2正規形

表 1.6: 学生表

学籍番号	名前	所属	Tel
A0P21000	文教太郎	情報・経情	xx-xxxx
A0P21999	湘南次郎	情報・経情	xx-xxxx
A0P11999	藤沢花子	情報・広報	xx-zzzz
⋮	⋮	⋮	⋮

表 1.8: 学生受講科目表

学籍番号	科目コード	成績
A0P21000	213905	A
A0P21000	213913	B
A0P21000	443909	C
A0P21000	233917	B
A0P21999	213905	A
A0P21999	323926	B
A0P11999	213906	F
A0P11999	213913	A
A0P11999	443909	B
⋮	⋮	⋮

表 1.7: 科目表

科目コード	科目名	単位	年度
213905	ソフトウェア演習	4	2000
213906	ソフトウェア演習	4	2001
213913	ソフトウェア応用演習	4	2000
443909	ネットワーク演習	2	2001
233917	意思決定論	4	2001
323926	ゼミナール	3	2000
⋮	⋮	⋮	⋮

第3正規形 第2正規形に対し、非キー属性間の従属関係を調べ、あればそれを別の表に独立させる（ex. Tel は所属に従属している属性なので、所属と Tel で別の表を作る）この時、別の表に独立させる属性（ex. 所属, Tel）があった表（ex. 学生表）には、その属性への関係が判るようにキー属性（ex. 所属 or 学科コードを新たに作成）を残しておく。（ex. 学生表には、所属がわかるように所属を残すか、学科表に新たに学科コード（主キー）を設定し、それを学生表に外部キーとしてつける）

正規化により得られた第3正規形では、各表ごとに、全てのキーでない属性が主キー（or 連結キー）に完全従属し、非キー属性間に従属関係はない。

表 1.9: 第 3 正規形

表 1.10: 学生表

学籍番号	名前	所属
A0P21000	文教太郎	P21
A0P21999	湘南次郎	P21
A0P11999	藤沢花子	P11
⋮	⋮	⋮

表 1.11: 科目表

科目コード	科目名	単位	年度
213905	ソフトウェア演習	4	2000
213906	ソフトウェア演習	4	2001
213913	ソフトウェア応用演習	4	2000
443909	ネットワーク演習	2	2001
233917	意思決定論	4	2001
323926	ゼミナール	3	2000
⋮	⋮	⋮	⋮

表 1.12: 学科表

学科コード	学科名	Tel
P11	情報・広報	xx-zzzz
P21	情報・経情	xx-xxxx
⋮	⋮	⋮

表 1.13: 学生受講科目表

学籍番号	科目コード	成績
A0P21000	213905	A
A0P21000	213913	B
A0P21000	443909	C
A0P21000	233917	B
A0P21999	213905	A
A0P21999	323926	B
A0P11999	213906	F
A0P11999	213913	A
A0P11999	443909	B
⋮	⋮	⋮

2 河村 [1] による解説から正規形の定義

2.1 関数従属性を用いた Codd の正規形について

2.1.1 第 1 正規形

第 1 正規系と非正規形は以下のように定義される。

Definition 2.1. 非正規形 : non first normal form (NF)

第 1 正規形でないものを非正規形とよぶ

Definition 2.2. 第 1 正規形 : first normal form (1FN)

関係を構成するどの属性値においても, 集合や関係となっていない

ここで関係 (relation) とは, 2 つ以上の「ある対象の集まり」において発生する結びつきのことであり, 集合 (set) とは, 1 つ以上の同じ属性を持つものの集まりである。

2.1.2 第2正規形

Definition 2.3. 第2正規形：second normal form (2FN)

関係 A における第2正規形とは，以下の2つを満たすものである．

- I. 関係 A が第1正規形である．
- II. 関係 A の全ての非キーである属性が，関係 A の主キーに完全関数従属している．

ここで，関数従属 (functional dependency) とは，関係における属性間の従属性を制約として定義したもので，関係 $A(a_1, a_2, \dots, a_n)$ において任意の属性を選んだいくつかの集まりを X, Y^2 としたとき，「 Y は X に関数従属である」とは， X の任意の値に対し， Y の値が唯一つ決定されることである．ここではこれを，

$$X \rightarrow Y$$

とあらわす．

また，「 Y は X に完全関数従属 (full functional dependency) である」とは， Y が X に関数従属しており，かつ，任意の X の真部分集合 $X' (\subset X)$ に対して， Y は X' に関数従属していないことである．即ち，

$$[X \rightarrow Y] \wedge \neg[X' \rightarrow Y] \text{ for } \forall X' \subset X$$

である．

2.1.3 第3正規形

Definition 2.4. 第3正規形：third normal form (3FN)

関係 A における第3正規形とは，以下の2つを満たすものである．

- I. 関係 A が第2正規形である．
- II. 関係 A の全ての非キーである属性が，関係 A の主キーに推移的関数従属しない．

ここで，関係 $A(a_1, a_2, \dots, a_n)$ において任意の属性を選んだいくつかの集まりを X, Y, Z^3 としたとき，「 Z は X に推移的関数従属 (transitive functional dependency) である」とは，

$$\langle [X \rightarrow Y] \wedge [Y \rightarrow Z] \wedge \neg[Y \rightarrow X] \rangle \Rightarrow [X \rightarrow Z]$$

が成り立つことをいう．

² X, Y は， $a_i (i = 1, \dots, n)$ の並び．

³ X, Y, Z は， $a_i (i = 1, \dots, n)$ の並び．

2.2 Boyce / Codd 正規形

関係 A において, X, Y^4 を任意の属性を選んだ幾つかの集まりとする.

Definition 2.5. Boyce/Codd 正規形 : Boyce/Codd normal form (BCNF)

関係 A における Boyce/Codd 正規形とは, 自明な $X \rightarrow Y$ が成立するか, X が関係 A の主キーであることをいう.

2.3 第 4, 5 正規形

第 4 正規形 (fourth normal form: 4NF) は, 関数従属性を含んだ多値従属性 (multivalued dependency) により定義され, 第 5 正規形 (fifth normal form: 5NF) は, 結合従属性 (join dependency) によって定義される.

第 1, 2, 3 正規形, Boyce/Codd 正規形, 及び, 第 4, 5 正規形について以下の関係が成り立つ.

Proposition 2.6.

$$1NF \supseteq 2NF \supseteq 3NF \supseteq \text{Boyce/Codd} \supseteq 4NF \supseteq 5NF$$

参考文献

- [1] 河村一樹. 『データベース要論—関係データベースとオブジェクト指向データベース—』. 啓学出版, 1995.
- [2] 高橋麻奈. 『ここからはじめるデータベース』. 日本実業出版社, 2000.
- [3] 斉藤直樹. 『データモデル設計と RDBMS への実装』. リックテレコム, 1995.
- [4] 滝沢誠. 『リレーショナル・データベースシステム RDBMS 技術解説』. ソフト・リサーチ・センター, November 1996.

⁴ X, Y は, $a_i (i = 1, \dots, n)$ の並び.