

データベース設計と正規化

森田 互昭

平成13年6月20日

1 データベース設計

データベースを設計を行う際には、作成しようとしているデータベースの一貫性を保つことが重要になります。

データベースに一貫性があるとは、トランザクション実行の前後でデータベース内の相互に関連する部分の間で矛盾がないことを意味します。

例えばある商品に関する、「在庫数」と「出荷した数」という項目を持つテーブルがあるとします。このテーブルに対し、「出荷」という操作を行ったとき、この操作の終了時点で「出荷した数を増やす」という操作と「在庫数を減らす」という2つが両方とも行われてはじめてデータベースの一貫性が保たれます。もし、片方だけ(出荷数の増加)をおこなうと、過剰在庫のような状態に陥ってしまい、このデータベースは一貫性を失います。

2 主キー

データベースの設計を行うときには、テーブルごとの主キーを決定する必要があります。主キーとは、テーブル内の行を一意に識別するために使われる項目のことで、テーブル内のデータを効率よく検索したり並べ替えたりするために使用されます。

主キーは、そのテーブルの全てのレコードでユニークでなければなりません。つまり、同じテーブル中に同じ値が存在してはいけません。

たとえば、学生を管理するデータベースにおいて、学籍番号は主キーとなることができますが、学生の名前はかなりの低確率とはいえ同じ学部に同姓同名の人がいる可能性が考えられますので、主キーとして使うことはできません。

また、主キーには、必ず空値以外の何らかの値が入っていなければいけません。学籍番号のない学生は登録できないというわけです。

3 正規化

正規化とはデータ構造についてデータの冗長性を少なくし、かつ更新不整合が起これないようにしていくことでデータベースの一貫性を保つ作業のことです。

正規化には第1正規形、第2正規形、第3正規形、ボイスコッド正規形、第4正規形、第5正規形の六つの種類があり、それぞれの正規形は前者を包含する関係になっています。それぞれの定義に

については、以下のようになっています。

- 第一正規形… テーブル内に属性の値として繰り返しなどの集合や複合値を持たないもの
- 第二正規形… テーブルが第1正規形であり、かつすべての非キー属性が主キーに対して完全関数従属であるもの
- 第三正規形… テーブルが第2正規形であり、かつすべての非キー属性が推移的に関数従属でないもの
- ボイスコッド正規形… 他の属性がある属性(X)に完全関数従属であり、かつ属性(X)が候補キーであるもの
- 第四正規形… ボイスコッド正規形のテーブルで、すべての適切な多値従属性がキーによるものであるもの
- 第五正規形… 分解によって得られるテーブルが結合によって、もとのテーブルに戻ることが可能であることを結合従属性といい、あるテーブルがその候補キー内で結合従属性が満足されているもの

3.1 第一正規形

第一正規形とは、テーブルにおいて属性の値として繰り返しなどの集合や複合値を持たないものをいい、この条件を満たさないものは非第一正規形と呼ばれます。

図1に示すテーブルは、非第一正規形です。

このテーブルには繰り返し項目が含まれているため、例えば「stereoを買った客は誰か」という

Receipt (受注票)							
s_code (伝票番号)	date (受注日)	cu_code (顧客コード)	cu_name (顧客名)	co_code (商品コード)	co_name (商品名)	price (単価)	quantity (数量)
0001	00.01.16	a001a	morita	A005	video	18000	1
0002	00.03.26	t430c	mine	B003	PS2	39500	1
				E002	stereo	50000	1
0003	00.07.19	n503i	kubo	B003	PS2	39500	1
				C009	telephone	20000	1
0004	00.12.24	i400u	ryu	D001	printer	17500	2

図 1: 非第一正規形

ような問い合わせを処理することができません。そこで、第一正規化では繰り返し項目を元の表から分離し、独立した行として存在するような新たな表を作成します。

第一正規化されたテーブルは、図 2 のようになります

この新たなテーブルにおいて、伝票番号と商品コードの組合わせはテーブル全体で重複していな

Receipt (受注票)							
s_code (伝票 番号)	date (受注日)	cu_code (顧客 コード)	cu_name (顧客名)	co_code (商品 コード)	co_name (商品名)	price (単価)	quantity (数量)
0001	00.01.16	a001a	morita	A005	video	18000	1
0002	00.03.26	t430c	mine	B003	PS2	39500	1
0002	00.03.26	t430c	mine	E002	stereo	50000	1
0003	00.07.19	n503i	kubo	B003	PS2	39500	1
0003	00.07.19	n503i	kubo	C009	telephone	20000	1
0004	00.12.24	i400u	ryu	D001	printer	17500	2

図 2: 第一正規形

いので、主キーを構成する列は、「伝票番号」、「商品コード」の 2 つになります。これで、正規化前は不可能だった問い合わせ処理が可能になりましたが、それでもまだ幾つかの問題が残っています。たとえば、この図 2 において、伝票番号 0002 である伝票の受注日に間違いがあったとすると、その訂正をするのに、この図では二個所のデータを修正しなくては行けません。もし修正を間違えようものなら、受注日の異なる 2 つの伝票 0002 が存在してしまうことになります。また、新たに入荷した商品をこのテーブルに追加しようとしても、まだ販売していないため伝票番号、伝票番号は登録するデータが無く、それなのにこれは主キーであるため空値を指定することもできず。結局この商品は登録できません。これらの問題は、受注日が伝票番号に依存しているのに商品コードには依存していないというように、値が一部の主キーにしか存在しないような列があるために存在してしまいます。

この問題を解決するために、第二正規化を行います。

3.2 第二正規形

第二正規形とはテーブルが第一正規形であり、かつすべての非キー属性が主キーに対して完全関数従属である場合をいいます。

完全関数従属というのは、どちらかの属性値を決めてやると、もう片方の属性値が決まるようなことです。ここでは、伝票番号から受注日、顧客コード、顧客名が、商品番号から商品名と単価が決定します。また、伝票番号と商品番号の組合わせにより、数量が決定します。これらをそれぞれ独立した別の表に分離するのが第二正規化で、正規化後の表は図 3 のようになります。この正規化により、主キーの一部に依存する列もなくなり、これで完了かと思うかも知れませんが、まだ、受注表において、先ほどと同様の問題が残っています(注文をしていない顧客が追加できないなど)。というわけで、第三正規化を行います

Receipt(受注表)				Commodity (商品)			Sales (売上)		
s_code (伝票 番号)	date (受注日)	cu_code (顧客 コード)	cu_name (顧客名)	co_code (商品 コード)	co_name (商品名)	price (単価)	s_code (伝票 番号)	co_code (商品 コード)	quantity (数量)
0001	00.01.16	a001a	morita	A005	video	18000	0001	A005	1
0002	00.03.26	t430c	mine	B003	PS2	39500	0002	B003	1
0002	00.03.26	t430c	mine	C009	telephone	20000	0002	E002	1
0003	00.07.19	n503i	kubo	D001	printer	17500	0003	B003	1
0003	00.07.19	n503i	kubo	E002	stereo	50000	0003	C009	1
0004	00.12.24	i400u	ryu				0004	D001	2

図 3: 第二正規形

3.3 第三正規形

第三正規形とは第二正規形で、すべての非キー属性が推移的に関数従属でないものをいいます。ここで推移的に関数従属というのは、ある属性を通じて二つの属性が従属関係にあるような状態を言います。ここでは、非キー属性である顧客名が、顧客コードを通じて伝票番号に推移的に従属(伝票番号から顧客コードが決定し、顧客コードから顧客名が決定)しています。第三正規化では、このような推移的従属関係にある項目を分離して、非キー属性が互いに独立であるテーブルを作成します。この正規化の結果得られる、図4のような全ての属性が主キーに従属したりレレーションを第三正規形といい、これが関係データベースの基本的なテーブルになります。

Receipt(受注表)			Customer(顧客)		Commodity (商品)			Sales (売上)		
s_code (伝票 番号)	date (受注日)	cu_code (顧客 コード)	cu_code (顧客 コード)	cu_name (顧客名)	co_code (商品 コード)	co_name (商品名)	price (単価)	s_code (伝票 番号)	co_code (商品 コード)	quantity (数量)
0001	00.01.16	a001a	a001a	morita	A005	video	18000	0001	A005	1
0002	00.03.26	t430c	t430c	mine	B003	PS2	39500	0002	B003	1
0003	00.07.19	n503i	n503i	kubo	C009	telephone	20000	0002	E002	1
0004	00.12.24	i400u	i400u	ryu	D001	printer	17500	0003	B003	1
					E002	stereo	50000	0003	C009	1
								0004	D001	2

図 4: 第三正規形

テーブルが分割されたとき、リレーショナルデータベースのあるテーブルの中で、他のテーブルの主キーとして使われている項目のことを外部キーと言います。ここで、受注表と顧客の関係において、受注表の顧客コードは顧客の主キーとして使われているので、外部キーであると言えます。

リレーショナルデータベースでは第一から第三までの正規化を行うのが一般的なようで、ボイス

コード正規形以降の正規化は実際に使用する機会は少ないようです。従って、ここまで出来れば矛盾を生じないデータベース設計の出来上がりです。

4 課題

図5に、正規化されていないテーブルがあります。これを正規化してみてください。
余裕のある人は、正規化したテーブルを Jasmine に定義してみましょう。

Receipt(受注表)							
o_code (注文 番号)	date (日付)	cu_name (顧客名)	cu_adress (住所)	co_code (商品 番号)	co_name (商品名)	quantity (数量)	price (単価)
0001	01.06.19	matsuoka	tokyo	1221	Oracle	1	1500
				2311	Access	3	520
0002	01.06.20	togawa	kanagawa	0342	Jasmine	2	330
				2311	Access	1	520

図 5: 今日の課題