

データベースシステム (6)

テーブルの設計

(一貫性制約・正規形)

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

1

関係モデルを作るときの注意点

- ・表を何個作る？
- ・どのデータとどのデータを一つの表にまとめる？
- ・表のスキーマをどう設計する？
- ・現実世界に存在する制約をどのようにして表に反映する？

実世界のデータを表に変換できれば、
関係データベースに記録し管理できるが、
設計が悪いと効率が悪くなってしまいます。

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

2

・現実世界に存在する制約をどのようにして表に反映する？

一貫性制約 (Integrity Constraint, IC)

キー制約 (主キー制約)

参照制約 (外部キー制約)

チェック制約

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

3

主キー (Primary Key) 制約

相異なる2つのタプルが同じ値を持たないような
属性(の組)をキーと呼ぶ
= キーの値(のみ)でタプルを特定できる

表に複数のキーとなる属性(の組)が
存在しうるが、そのどちらかが主キーとなる
(1つの表に主キーは一つだけ)

sid	sname	faculty	address	phone
B090003	inoue mao	enginring	...	...
B090007	ueto aya	law	...	...
B090021	horikita maki	science	...	...
...	...	...	...	...

↑ 学生番号 (sid) で
特定できる

↑ 電話番号もユニークだが
学生番号を主キーにする場合は、
この属性は主キーとはしない

4

キーの例

grade	class	no	sname	address	phone
1	A	5	inoue mao	...	...
2	A	8	ueto aya	...	...
1	B	17	horikita maki	...	...
...	...	...	...	...	...

学年 (grade), 組 (class), 出席番号 (no)
の組で特定できる

この場合
学年 (grade), 組 (class), 出席番号 (no)
の3属性の組を主キーとする。

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

5

主キー (Primary Key) 制約

SQLで主キーの属性には「Primary Key制約」を
主キー以外のキーは「Unique制約」を設定する

sid	sname	faculty	address	phone
B090003	inoue mao	enginring	...	...
...	...	...	...	...

```
CREATE TABLE Student
(sid CHAR(10)
sname CHAR(20),
faculty CHAR(20),
address CHAR(50),
phone CHAR(12),
PRIMARY KEY (sid)
UNIQUE (phone) )
```

「Primary Key制約」
「Unique制約」
のある属性値の
重複あるいは空値 (未記入)
を (DBが) 許さない

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

6

grade	class	no	sname	address	phone
1	A	5	inoue mao	...	...
2	A	8	ueto aya	...	...
1	B	17	horikita maki	...	...
...	...	...	...	...	...

```
CREATE TABLE Pupil
(grade CHAR(1)
class CHAR(1),
no CHAR(2),
sname CHAR(20),
address CHAR(50),
phone CHAR(12),
PRIMARY KEY (grade, class, no)
UNIQUE (phone) )
```

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

7

主キー (Primary Key) 制約の例1

学生番号 講義コード 成績

履修 (Enrolled)	sid	cid	grade
	53831	c101	C
	53831	c203	B
	53650	c112	A
	53666	c105	B

```
(1) CREATE TABLE Enrolled
(sid CHAR(20)
cid CHAR(20),
grade CHAR(2),
PRIMARY KEY (sid,cid) )
```

(1) 一人の学生と
その人の履修する科目の成績は一つの値

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

8

主キー (Primary Key) 制約の例2

学生番号 講義コード 成績

履修 (Enrolled)	sid	cid	grade
	53831	c101	C
	53831	c203	B
	53650	c112	A
	53666	c105	B

```
(2) CREATE TABLE Enrolled
(sid CHAR(20),
cid CHAR(20),
grade CHAR(2),
PRIMARY KEY (sid),
UNIQUE (cid, grade) )
```

(2) 「学生は一つの科目のみを履修」でき、
「各科目には同じ成績の人はいない」

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

9

外部キー (Foreign Key) と参照制約 (Referential Integrity)

外部キー

あるリレーション中の属性(の組)で他のリレーション
の一つのタプルを参照するもの
(参照先のリレーションの主キーの値をもつ)

Eg. 学生をsid値で参照する場合, sidが外部キー

Enrolled(sid char(20), cid char(20), grade char(2))

外部キー制約が指定された場合,
参照制約により, 参照先のない外部キー値は
禁止される

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

10

SQL 外部キーの指定

```
CREATE TABLE Enrolled
(sid CHAR(20), cid CHAR(20), grade CHAR(2),
PRIMARY KEY (sid,cid),
FOREIGN KEY (sid) REFERENCES Students(sid) )
```

Enrolled			Students				
sid	cid	grade	sid	name	login	age	gpa
53666	c101	C	53666	Jones	jones@cs	18	3.4
53666	c203	B	53688	Smith	smith@ee	18	3.2
53650	c112	A	53650	Smith	smith@ee	19	3.8
53666	c105	B					

学生リレーションに存在する学生のみ
科目を履修できる

→ Students表にあるsid値のみEnrolled表のsid値に
することができる。

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

11

参照制約 (Referential Integrity) の指定

「履修」リレーションのsid
「学生」リレーションのsidを外部キーとして指定している
その場合

「履修」リレーションには「学生」リレーションにない
sidをもつタプルを挿入できない(エラーとなる)

「学生」リレーションからデータが削除された場合 (3種類の対処)

- そのsid値をもつ「履修」のタプルをすべて削除
 - 外部キーとして参照されているタプルは ← デフォルト
削除不可とする
 - 削除されたsid値をもつ「履修」タプルのsid値が
あるデフォルト値(あるいは空値)に書き換えられる
- 「学生」のsid値の変更の場合も同様

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

12

Enrolled			Students				
sid	cid	grade	sid	name	login	age	gpa
53666	c101	C	53666	Jones	jones@cs	18	3.4
53666	c203	B	53688	Smith	smith@ee	18	3.2
53650	c112	A	53650	Smith	smith@ee	19	3.8
53666	c105	B					

履修へのデータ挿入
学生にないsidは禁止

学生へのデータ挿入は自由

履修データの削除は自由

学生へのデータ削除
1. 履修データも削除
2. 履修データがあれば削除禁止
3. 履修データに null値などのデフォルト値

履修データのsid値変更
学生にないsid値への変更は禁止

学生のデータの (sid値の) 変更
削除, 禁止, デフォルト値

SQL 参照制約 (Referential Integrity) の指定	
- CASCADE (主キーが削除されたら削除) - NO ACTION (削除変更の禁止) - SET NULL / SET DEFAULT (空値あるいはデフォルト値)	<pre>CREATE TABLE Enrolled (sid CHAR(20), cid CHAR(20), grade CHAR(2), PRIMARY KEY (sid,cid), FOREIGN KEY (sid) REFERENCES Students(sid) ON DELETE CASCADE ON UPDATE NO ACTION)</pre> 削除時 更新時
テーブル作成時の外部キーの指定で参照しているテーブルの削除時, 更新時の動作を指定しない場合のデフォルト設定は「NO ACTION」	

主キー制約, 参照制約 (外部キー) 以外のデータ内容に関する制約	
属性値のドメインを限定したい場合など	
例:	
・学生の年齢は18歳以上にしたい ・履修学生の所属学部を「総科」と理系学部(「生」,「工」,「理」,「医」)だけに限定したい	
「チェック制約 (Check)」を指定	

SQL 検査制約 (Check) の指定	
<pre>CREATE TABLE SciStudents (sid CHAR(20), name CHAR(20), ... age INTEGER, faculty CHAR(10) ... CHECK (age >= 18), CHECK (faculty IN ('総科', '生', '工', '理', '医')))</pre>	

一貫性制約 (Integrity Constraint, IC)	
データベースのどのインスタンス (テーブル) も IC を満たさなければならない	
・IC はスキーマを定義するときに指定する ・IC はリレーションが変更されるときに検査される	
正規のインスタンス	
= 指定されたすべての IC を満足するもの	
DB が正規ではないインスタンスの存在をゆるさない	
DB 内に存在するデータは正しく信頼できるものとしてよい (データの挿入, 削除, 変更時にエラーがチェックされているため)	

関係モデルを作るときの注意点	
・表を何個作る? ・どのデータとどのデータを一つの表にまとめる? ・表のスキーマをどう設計する? ・現実世界に存在する制約をどのようにして表に反映する?	
実世界のデータを表に変換できれば, 関係データベースに記録し管理できるが, 設計が悪いと効率が悪くなってしまいます.	

関係データベースでは
リレーションの各インスタンス(テーブル)は
正規形(Normal Form)でなければならない

正規形でないテーブルの例

属性A	属性「姓, 名」	属性B
	上戸, 彩	
...	...	...

テーブルの属性内に
別のテーブル
(「姓」属性, 「名」属性を
もつテーブルのデータ)が
入っている

属性「店名」	属性「メニュー」	属性A
うさぎ亭	塩, 味噌, 醤油	
もりも軒	カレー, 豚骨	
...	...	...

ある属性値が
集合値をとっている

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

19

正規形でないテーブルの正規化(Normalization)

属性A	属性「姓, 名」	属性B
	上戸, 彩	
...	...	...

属性A	属性「姓」	属性「名」	属性B
	上戸	彩	
...	...	...	...

属性「店名」	属性「メニュー」	属性A
うさぎ亭	塩, 味噌, 醤油	
もりも軒	カレー, 豚骨	
...	...	...

「店名」	「メニュー」	属性A
うさぎ亭	塩	
うさぎ亭	味噌	
うさぎ亭	醤油	
もりも軒	カレー	
もりも軒	豚骨	
...	...	...

厳密には正規形にも, さらなる細かい制約があり
非正規形ではないというだけの正規形を
第1正規形(1NF)と呼ぶ

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

20

ケーススタディ

NO. 3492
2003年8月10日

このような
発注書に関する
テーブルを
作成する

ドラッグストア はねみつる 株式会社
調達 部 倉庫

製品の発注依頼書

拝啓

貴社, ますますご隆盛のことと, お喜び申し上げます。さて, 貴社商品を, 下記の通り注文いた
します。貴社へ発送くださいますよう, お願い申し上げます。

敬具

記

品名・数量・単価	品名	数量	単価	金額
塩	30個	98円	4,900円	
味噌	30個	218円	6,540円	
醤油	50個	92円	4,600円	
豚骨	100個	185円	18,500円	
合計				34,540円

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

以上

ケーススタディ

NO. 4918
2003年8月10日

このような
発注書に関する
テーブルを
作成する

ドラッグストア はねみつる 株式会社
調達 部 広末

製品の発注依頼書

拝啓

貴社, ますますご隆盛のことと, お喜び申し上げます。さて, 貴社商品を, 下記の通り注文いた
します。貴社へ発送くださいますよう, お願い申し上げます。

敬具

記

品名・数量・単価	品名	数量	単価	金額
塩	100個	95円	9,500円	
醤油	50個	87円	4,350円	
カレー	20個	145円	2,900円	
海鮮	100個	80円	8,000円	
チーズ	50個	77円	3,850円	
合計				28,600円

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

以上

ケーススタディ

発注書

発注番号	各注文に割り当てられる番号
会社名(注文先)	注文先の会社名
取引先ID	注文先のID番号
日付	この注文が発生した日付
扱い者	この注文の取り扱い責任者
品名	
数量	
単価	

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

23

非正規形ではない, というだけの正規形を第1正規形(1NF)

ケーススタディの発注書を(1NFで)正規化したテーブル

このテーブルのキーは
「発注番号」「品名」

発注番号	取引先ID	会社名	日付	扱い者	品名	数量	単価
3492	c61	うさぎ亭	03/8/10	倉木	塩	50	98
3492	c61	うさぎ亭	03/8/10	倉木	味噌	30	218
3492	c61	うさぎ亭	03/8/10	倉木	醤油	50	92
3492	c61	うさぎ亭	03/8/10	倉木	豚骨	100	185
4918	c45	もりも軒	03/8/19	広末	塩	100	95
4918	c45	もりも軒	03/8/19	広末	醤油	50	87
4918	c45	もりも軒	03/8/19	広末	カレー	20	145
4918	c45	もりも軒	03/8/19	広末	海鮮	100	80
4918	c45	もりも軒	03/8/19	広末	チーズ	50	77

第1正規形はキーの一部に従属する「関数従属性」をゆるす

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

24

「関数従属性」

顧客	製品名	数量	定価	合計額
A	TV	3	198000	594000
B	TV	10	198000	1980000
B	CD	5	59800	299000
C	Video	1	29800	29800

「関数従属性」

属性Aと属性Bは関数従属関係
=「Aの値(Bの値)が決まれば
Bの値(Aの値)が一意に決まる」
= BはAに従属(AはBに従属)

製品名→定価
{製品名, 数量}→合計額
{数量, 定価}→合計額
{数量, 合計額}→定価
{定価, 合計額}→数量

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

関数従属性をもつテーブルの問題点

発注番号	取引先ID	会社名	日付	扱い者	品名	数量	単価
3492	c61	うさぎ亭	03/8/10	倉木	塩	50	98
3492	c61	うさぎ亭	03/8/10	倉木	味噌	30	218
...	...	...	...	...	...	...	...

キー値の一部に従属する属性は管理上問題がある

- ・同じデータが重複して記入されるのでスペースの無駄
- ・データ変更が面倒(「うさぎ亭」が「きりん食堂」になったら)
 - タプル更新時異常
- ・注文実績のない新たな取引先が加わったら(主キー制約違反)
 - タプル挿入時異常
- ・注文実績が一つしかない会社の注文がキャンセルになったら(その会社の情報がなくなる)
 - タプル削除時異常

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

・注文実績のない新たな取引先が加わったら(主キー制約違反)
タプル挿入時異常

発注番号	取引先ID	会社名	日付	扱い者	品名	数量	単価
...	...	...	...	...	...	...	...
×	c92	くじら軒					
...	...	...	...	...	...	...	...

・注文実績が一つしかない会社の注文がキャンセルになったら
(その会社の情報がなくなる)
タプル削除時異常

発注番号	取引先ID	会社名	日付	扱い者	品名	数量	単価
...	...	...	...	...	...	...	...
0604	c75	めじろ	03/7/14	広末	醤油	40	87
...	...	...	...	...	...	...	...

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

非正規形ではないというだけの正規形を第1正規形(1NF)

第1正規形はキーの一部に従属する「関数従属性」をゆるす

(キーの一部に従属するような)
関数従属性のあるテーブルには

- ・同じデータが重複して記入されるのでスペースの無駄
- ・データ変更が面倒(「うさぎ亭」が「きりん食堂」になったら)
 - タプル更新時異常
- ・注文実績のない新たな取引先が加わったら(主キー制約違反)
 - タプル挿入時異常
- ・注文実績が一つしかない会社の注文がキャンセルになったら(その会社の情報がなくなる)
 - タプル削除時異常

などの問題があり、管理上好ましくない

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

関数従属の冗長性を解消

第1正規形(1NF)

発注番号	取引先ID	会社名	日付	扱い者	品名	数量	単価
...	...	...	...	...	...	...	...

↓

発注番号	取引先ID	会社名	日付	扱い者
3492	c61	うさぎ亭	03/8/10	倉木
4918	c45	もりも軒	03/8/19	広末

第2正規形(2NF)

発注番号	品名	数量	単価
3492	塩	50	98
3492	味噌	30	218
3492	醤油	50	92
3492	豚骨	100	185
4918	塩	100	95
4918	醤油	50	87
4918	カレー	20	145
4918	海鮮	100	80
4918	チーズ	50	77

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

第2正規形(2NF)

キーの一部からの関数従属は存在しないが
推移的関数従属が存在する

第1正規形であり、かつ、
非キー属性で関数従属する属性が
すべて主キーに従属して
いる正規テーブル

発注番号	取引先ID	会社名	日付	扱い者
3492	c61	うさぎ亭	03/8/10	倉木
4918	c45	もりも軒	03/8/19	広末

↓

発注番号	品名	数量	単価
3492	塩	50	98
3492	味噌	30	218
3492	醤油	50	92
3492	豚骨	100	185
4918	塩	100	95
4918	醤油	50	87
4918	カレー	20	145
4918	海鮮	100	80
4918	チーズ	50	77

「取引先ID」が決まれば「会社名」も決まる
「取引先ID」は主キーではないが、
「取引先ID」は主キーに従属
「会社名」は間接的に主キーに従属 推移的関数従属

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

推移的関数従属

発注番	取引先ID	会社名	日付	扱い者
3492	c61	うさぎ亭	03/8/10	倉木
4918	c45	もりも軒	03/8/19	広末
5579	c45	もりも軒	03/8/21	上戸
6048	c45	もりも軒	03/9/15	香取
7736	c61	うさぎ亭	03/9/19	広末

- ・同じデータが重複して記入されるのでスペースの無駄
- ・データ変更が面倒(「もりも軒」が「きりん食堂」になったら)
タプル更新時異常
- ・注文実績のない新たな取引先が加わったら(主キー制約違反)
タプル挿入時異常
- ・発注実績が一つしかない会社の注文がキャンセルになったら
(その会社の情報がなくなる)
タプル削除時異常

「推移的関数従属性」の解消

発注番	取引先ID	会社名	日付	扱い者
3492	c61	うさぎ亭	03/8/10	倉木
4918	c45	もりも軒	03/8/19	広末

発注番	取引先ID	日付	扱い者
3492	c61	03/8/10	倉木
4918	c45	03/8/19	広末

取引先ID	会社名
c61	うさぎ亭
c45	もりも軒

発注番	品名	数量	単価
3492	塩	50	98
3492	味噌	30	218
3492	醤油	50	92
3492	豚骨	100	185
4918	塩	100	95
4918	醤油	50	87
4918	カレー	20	145
4918	海鮮	100	80
4918	チーズ	50	77

第3正規形(3NF)

第2正規形であり、かつ、
推移的関数従属性のない正規テーブル

発注番	取引先ID	日付	扱い者
3492	c61	03/8/10	倉木
4918	c45	03/8/19	広末

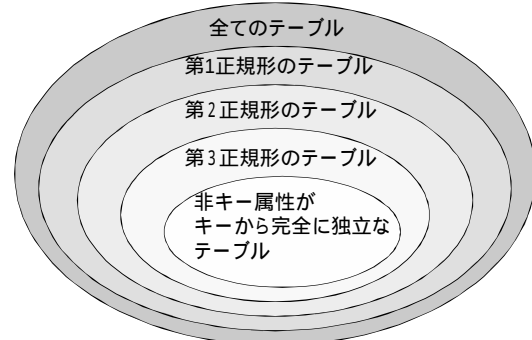
取引先ID	会社名
c61	うさぎ亭
c45	もりも軒

発注番	品名	数量	単価
3492	塩	50	98
3492	味噌	30	218
3492	醤油	50	92
3492	豚骨	100	185
4918	塩	100	95
4918	醤油	50	87
4918	カレー	20	145
4918	海鮮	100	80
4918	チーズ	50	77

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

33

正規形の種類とそれらの包含関係



データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

34

データは「第3正規形(3NF)」にしておくと管理しやすい

データの
・更新時異常
・挿入時異常
・削除時異常
がなくなる

第3正規形はデータ管理上、理想的な保存形態

ただし、問い合わせの効率を要求されるケースでは
第3正規形をくずしたデータ形態をとることもある
(複数のテーブルのジョインが必要となる問い合わせは
効率が悪い)

データベースシステム (担当: 森本康彦@広島大 2009/2/12-14)

35

発注番	取引先ID	日付	扱い者
3492	c61	03/8/10	倉木
4918	c45	03/8/19	広末

取引先ID	会社名
c61	うさぎ亭
c45	もりも軒

発注番	品名	数量	単価
3492	塩	50	98
3492	味噌	30	218
3492	醤油	50	92
3492	豚骨	100	185
4918	塩	100	95
4918	醤油	50	87
4918	カレー	20	145
4918	海鮮	100	80
4918	チーズ	50	77

「取引先」毎の
売り上げ金額(数量×単価)の合計
を頻繁に集計する必要がある場合は...

この形態のほうが
この問い合わせの計算効率は
よい(しかし更新時異常など
が起こるので管理上注意が必要)

発注番	取引先ID	品名	数量	単価
3492	c61	塩	50	98
3492	c61	味噌	30	218
...	...	...	...	...

36