

ニューロ型データベースモデリング A Model of Neuron Based Database System

小島 茂*

Shigeru KOJIMA

「脳を構成する神経細胞 (neuron) が繋がりが合って情報処理を司る」という仕組みに着目し、リレーショナルデータベース管理システム(RDBMS)のテーブル設計にこのニューロ型を採用し、汎用性と拡張性を実現したデータベースモデリングを紹介する。

In this paper, a model of a database, realizing broader utility and scalability, with neuron-like system for RDBMS tables is introduced.

1. はじめに

このニューロ型データベースモデリングは2個のテーブルだけで構築されている。

我々は、リレーショナルデータベース管理システム(RDBMS)を利用してあらゆるデータベースシステムを構築している。データベースシステムの作成には「データモデリング」が必須であり、システム開発者は実体関連図 (ER図) や統一モデリング言語 (UML) 等を駆使しこのデータモデリングに取り組み多くの開発時間を費やしている。

データモデリングの優劣はそのデータベースシステムの優劣であると言っても過言ではない。

本論文は「脳の神経細胞 (neuron) が繋がりが合って情報を司る」という仕組みをRDBMSのテーブル設計に応用し「オブジェクト指向」の処理方法を実現したものである。

2. 単語を学習し始める脳

2.1 Word_Entity_tbl の作成

ニューロ型データベースモデリング (これ以降 脳 と表す) で使用する「単語」を記憶していくテーブル「Word_Entity_tbl」を作成する (表1)。

- [ID]: 0 から始まるプライマリキー
- [単語]: 可変長文字列 (重複なし) (Not Null)

2.2 Word_Entity_tbl の入力規則

作成したWord_Entity_tblの[ID]に0を、[単語]に なし を登録し続いて「入力フォーム」で使用するボタン名, 列名, 入力項目名を「単語」として登録する。

- 登録順は自由
- 氏名は「氏」と「名」に分けて登録
- 未使用なら削除可
- フリガナは全角カタカナを使用
- 半角カタカナは使用不可
- 機種依存文字 (丸数字等) は使用可

*正会員 北浜法律事務所・外国法共同事業 情報システム課 skojima@kitahama.or.jp

表1 Word_Entity_tbl

Table 1 Word_Entity_tbl

ID	単語
0	なし
1	未来
2	現在
3	過去
4	識別名
5	入力フォーム
6	主キー
7	識別ID
8	T_ID
9	時点
10	ソート
11	F_ID
12	入力項目
13	D_ID
14	入力データ
15	識別IDリンク先
16	開始日時
17	終了日時
18	閉じる
19	登録
20	人
21	氏
22	氏ヨミ
23	名
24	名ヨミ
25	性別
26	西暦生年月日

3. 単語を繋ぎ始める脳

3.1 Identify_Entity_tbl の作成

脳はこれからあらゆるオブジェクト (物体) を識別化しユニークなID (これ以降 識別ID と表す) を付けて記憶していくテーブル「Identify_Entity_tbl」を作成する (表2)。

- [ID]: 1 から始まるプライマリキー
- ソート順: [識別ID][T_ID][ソート][ユニット]
- [識別ID][T_ID][ソート][ユニット]は (重複あり)
- 全ての列は数値 (型, Index等はRDBMS仕様に準拠)

表2 Identify_Entity_tbl

Table 2 Identify_Entity_tbl

ID	識別ID	T_ID	ソート	ユニット	F_ID	D_ID	識別IDリンク先
1	1	0	0	0	4	20	
2	1	2	1	1	21	27	
3	1	2	2	1	22	28	
4	1	2	3	2	23	29	
5	1	2	4	2	24	30	
6	1	2	5	0	25	31	
7	1	2	6	0	26	32	

3.2 識別 ID 化フォームの準備

まず「人」を識別化する為の「入力フォーム」を準備する。図1の入力フォーム（イメージ）を開くとバックグラウンドで以下の処理を実行する。

- [主キー]： 1 （自動取得）
- [識別ID]： 主キーと同じ値 1 （自動登録）
- [T_ID]： 0 （自動登録）
- [時点]： なし （自動表示）
- [ソート]： 0 （自動登録）
- [ユニット]： 0 （自動登録）

3.2.1 識別名の登録

図1は[入力項目]に 識別名 [入力データ]に 人 が入力された後の状態である。[F_ID]に 4 [D_ID]に 20 が自動表示されるのは既に Word_Entity_tbl へ単語登録が済んでいるからである。

- [F_ID]： 4 （自動表示）
- [入力項目]： 識別名
- [D_ID]： 20 （自動表示）
- [入力データ]： 人
- [識別ID]： 主キーと同じ値を登録
(今後「グループ管理用」として使用)
- 登録ボタンを押し[識別ID]を確定

入力フォーム		識別ID		1					
主キー	識別ID	T_ID	時点	ソート	ユニット	F_ID	入力項目	D_ID	入力データ
1	1	0	なし	0	0	4	識別名	20	人
*									

図1 識別名の登録

Fig.1 registry of identification

3.2.2 入力項目の登録 —縦列連結—

単語が縦列連結すると[入力項目]となる。図2は[入力項目]に単語入力を終えた状態である。赤丸は[ユニット]で、2つのグループが作成された状態である。例えば「氏」を変更する時、同じユニット番号を持つ「氏ヨミ」も変更処理対象とするのに使う。

- [主キー]： 2 ~ 7 （自動取得）
- [識別ID]： 主キーと同じ値 1 （自動登録）¹
- [T_ID]： 2 （自動登録）
- [時点]： 現在 （自動表示）
- [ソート]： 1 ~ 6 （自動連番）
- [ユニット]： 0 （自動登録）と手入力
(任意の整数で連番である必要はない)
(第4ソートキーであることに注意)
- [F_ID]： 21 22 23 24 25 26 （自動表示）
- [入力項目]： 氏 氏ヨミ 名 名ヨミ 性別 西暦生年月日

入力フォーム		識別ID		1					
主キー	識別ID	T_ID	時点	ソート	ユニット	F_ID	入力項目	D_ID	入力データ
1	1	0	なし	0	0	4	識別名	20	人
2	1	2	現在	1	1	21	氏		
3	1	2	現在	2	1	22	氏ヨミ		
4	1	2	現在	3	2	23	名		
5	1	2	現在	4	2	24	名ヨミ		
6	1	2	現在	5	0	25	性別		
7	1	2	現在	6	0	26	西暦生年月日		
*									

図2 入力項目の登録

Fig.2 registry of items

3.3 識別 ID 化フォームの入力—横列連結—

図3は[入力データ]にデータ入力完了した状態である。

- [F_ID]： 27 28 29 30 31 32 （自動表示）
- [入力データ]： 小島 コジマ 茂 シゲル 男 1964-01-08

入力フォーム		識別ID		1					
主キー	識別ID	T_ID	時点	ソート	ユニット	F_ID	入力項目	D_ID	入力データ
1	1	0	なし	0	0	4	識別名	20	人
2	1	2	現在	1	1	21	氏	27	小島
3	1	2	現在	2	1	22	氏ヨミ	28	コジマ
4	1	2	現在	3	2	23	名	29	茂
5	1	2	現在	4	2	24	名ヨミ	30	シゲル
6	1	2	現在	5	0	25	性別	31	男
7	1	2	現在	6	0	26	西暦生年月日	32	1964-01-08
*									

図3 識別 ID 化フォームの入力

Fig.3 entry of edentification forms

[入力項目]と[入力データ]が横列連結したのでこれで「情報」となり複数の情報が登録され人として識別化する最低限の情報が整ったのでここで登録ボタンを押す。

3.4 識別 ID 化の完了

これで識別 ID が 1 である「人」の登録が完了した。

注目すべき点は全レコードの[識別 ID]に先頭の主キー 1 が自動登録されたものであり手動入力出来ない、この処理は入力フォーム上で処理されたものでありこの脳の特徴を現している。

入力項目の追加は「テーブル列を追加するのではなくレコードとして縦長に追加していく」という仕様なので、縦長に追加されたレコードを1つの同じグループとして扱う為の仕組みである。テーブルに列を追加しないのでいつでも簡単に（テーブルロックを意識せず）追加出来るのは便利な機能である。

¹ グループ管理用である[識別 ID]の番号は新しく識別名を追加する度にレコード先頭行の主キー値を採用するので連番になるとは限らない

4. 拡張を始める脳

4.1 入力項目の追加

人の識別 ID は確定したが同姓同名，同一生年月日，同一性別が 2 人以上存在する場合これだけでは「個人」の特定が出来ないので識別 ID 1 に[入力項目]として E-mail を追加する (図 4)。

- [主キー]： 8 (自動取得)
- [識別ID]： 1 (自動登録)
- [T_ID]： 2 (自動登録)
- [[時点]： 現在 (自動表示)
- [ソート]： 7 (自動連番)
- [ユニット]： 0 (自動登録)
- [F_ID]： 33 (単語登録処理後自動表示)
- [入力項目]： E-mail
- [D_ID]： 34 (単語登録処理後自動表示)
- [入力データ]： skojima@kitahama.or.jp
- 登録ボタンを押してデータ追加確定
- 2つ以上登録する場合は mail2 を新規作成しても良い

入力フォーム 識別ID 1

主キー	識別ID	T_ID	時点	ソート	ユニット	F_ID	入力項目	D_ID	入力データ
1	1	0	なし	0	0	4	識別名	20	人
2	1	2	現在	1	1	21	氏	27	小島
3	1	2	現在	2	1	22	氏ヨミ	28	コジマ
4	1	2	現在	3	2	23	名	29	茂
5	1	2	現在	4	2	24	名ヨミ	30	シゲル
6	1	2	現在	5	0	25	性別	31	男
7	1	2	現在	6	0	26	西暦生年月日	32	1964-01-08
8	1	2	現在	7	0	33	E-mail	34	skojima@kitahama.or.jp
9	1	1	未来	1	1	21	氏	35	中島
10	1	1	未来	2	1	22	氏ヨミ	36	ナカジマ
*									

図5 ユニットの役割

Fig.5 function of "unit"

今すぐ変更を適用するならここで登録ボタンを押す。この脳の「ユニット」は関連[入力項目]の修正漏れを防止する便利な機能である。

4.3 切り替え処理を予定する タスクとの連携

新しい「氏」の適用開始日が 2 日後または休日である場合 [開始日時]と[終了日時]を入力し登録ボタンを押せば指定日時に切り替え処理をサーバのタスクが代わりに実行する。

- [開始日時]と[終了日時]には以下の値を設定する (図 6)。
- 2 現在 27 小島 [終了日時]： 2011-03-15 23:59
 - 2 現在 28 コジマ [終了日時]： 2011-03-15 23:59
 - 1 未来 35 中島 [開始日時]： 2011-03-16 00:00
 - 1 未来 36 ナカジマ [開始日時]： 2011-03-16 00:00
- [終了時間]は[開始時間]の 1 分前を指定しておけば，サーバ側のタスクが毎時 0 時にタスクを実行し，未来は現在に現在は過去にそれぞれ UPDATE 文で変更される (図 8)。

入力フォーム 識別ID 1

主キー	識別ID	T_ID	時点	ソート	ユニット	F_ID	入力項目	D_ID	入力データ
1	1	0	なし	0	0	4	識別名	20	人
2	1	2	現在	1	1	21	氏	27	小島
3	1	2	現在	2	1	22	氏ヨミ	28	コジマ
4	1	2	現在	3	2	23	名	29	茂
5	1	2	現在	4	2	24	名ヨミ	30	シゲル
6	1	2	現在	5	0	25	性別	31	男
7	1	2	現在	6	0	26	西暦生年月日	32	1964-01-08
8	1	2	現在	7	0	33	E-mail	34	skojima@kitahama.or.jp
*									

図4 入力項目の追加

Fig.4 addition of items

4.2 入力データの変更 - ユニットの役割

人は「氏」を変更することがある。入力データ (単語) の変更処理はレコード内の[D_ID]は書き換えず，修正対象レコードの「ユニット」番号と同じレコードを 1 未来とし [D_ID][入力データ][識別IDリンク先]¹は空欄にして複製する。複製された 1 未来レコードに新しいデータ (単語) を入力し現レコードを 3 過去として「履歴化」する。

- 27 小島 をダブルクリック
- [主キー]： 9 10 (自動取得)
- [識別ID]： 1 (自動登録)
- [T_ID]： 1 (自動登録)
- [時点]： 未来 (自動表示)
- [ソート]： 1 2 (複製元と同じ値を継承)
- [ユニット]： 1 (複製元と同じ値を継承)
- [F_ID]： 21 22 (複製元と同じ値を継承)
- [入力項目]： 氏 氏ヨミ (複製元と同じ値を継承)
- [D_ID]： 35 36 (自動表示)
- [入力データ]： 中島 ナカジマ

項目	D_ID	入力データ	開始日時	終了日時	登録者
	20	人	2011-03-12 14:23		skojima
	27	小島	2011-03-12 14:23	2011-03-15 23:59	skojima
	28	コジマ	2011-03-12 14:23	2011-03-15 23:59	skojima
	29	茂	2011-03-12 14:23		skojima
	30	シゲル	2011-03-12 14:23		skojima
	31	男	2011-03-12 14:23		skojima
年月日	32	1964-01-08	2011-03-12 14:23		skojima
	34	skojima@kitahama.or.jp	2011-03-12 15:20		skojima
	35	中島	2011-03-16 00:00		skojima
	36	ナカジマ	2011-03-16 00:00		skojima

図6 開始日時と終了日時の設定

Fig.6 setting of valid time installation

図 7 はサーバ側でタスクが実行されるまで 1 未来 が 2 現在 よりも上部に表示されている。[開始日時]を参照すれば適用日時が確認出来るし取り消しも可能だ。

¹ 図 9 を参照

入力フォーム										識別ID	1
主キー	識別ID	T_ID	時点	ソート	ユニット	F_ID	入力項目	D_ID	入力データ		
1	1	0	なし	0	0	4	識別名	20	人		
9	1	1	未来	1	1	21	氏	35	中島		
10	1	1	未来	2	1	22	氏ヨミ	36	ナカジマ		
2	1	2	現在	1	1	21	氏	27	小島		
3	1	2	現在	2	1	22	氏ヨミ	28	コジマ		
4	1	2	現在	3	2	23	名	29	茂		
5	1	2	現在	4	2	24	名ヨミ	30	シゲル		
6	1	2	現在	5	0	25	性別	31	男		
7	1	2	現在	6	0	26	西暦生年月日	32	1964-01-08		
8	1	2	現在	7	7	33	E-mail	34	sko.jima@kita hama.or.jp		
*											

図7 タスク実行前

Fig.7 before operating the task

図8はタスク処理完了後の画面である。履歴化されたデータは原則非表示にする。履歴データ表示ボタンが表示されているのでクリックすれば履歴化されたデータが表示される。

入力フォーム										識別ID	1
主キー	識別ID	T_ID	時点	ソート	ユニット	F_ID	入力項目	D_ID	入力データ		
1	1	0	なし	0	0	4	識別名	20	人		
9	1	2	現在	1	1	21	氏	35	中島		
10	1	2	現在	2	1	22	氏ヨミ	36	ナカジマ		
4	1	2	現在	3	2	23	名	29	茂		
5	1	2	現在	4	2	24	名ヨミ	30	シゲル		
6	1	2	現在	5	0	25	性別	31	男		
7	1	2	現在	6	0	26	西暦生年月日	32	1964-01-08		
8	1	2	現在	7	7	33	E-mail	34	sko.jima@kita hama.or.jp		
2	1	3	過去	1	1	21	氏	27	小島		
3	1	3	過去	2	1	22	氏ヨミ	28	コジマ		
*											

履歴データ表示 [1未来]が→[2現在]に変更 [2現在]が→[3過去]に変更

図8 タスク実行後

Fig.8 after operating the task

4.4 入力データの削除

レコードの削除は[F_ID]をダブルクリックする。現レコードを完全削除するのではなく[T_ID]を3 過去に UPDATE 文で変更し履歴化する。後日「削除したレコードを復活させたい」「削除した人物を知りたい」等を想定している。

- [入力データ]が空白のレコード
 - [入力項目]の設定ミス
 - 追加作業途中の取り消し
- については完全削除を認める。

5. 連携を始める脳

5.1 識別 ID リンク先

個人を特定する話の続きである。同姓同名，同一生年月日，同一性別であり E-mail 携帯電話がない。という場合個人を特定するにはやはり「自宅」「勤務先」が必要になる(図9)。

- [主キー]: 11 (自動取得)
- [識別ID]: 1 (自動登録)
- [T_ID]: 2 (自動登録)
- [[時点]: 現在 (自動表示)
- [ソート]: 8 (自動連番)
- [ユニット]: 0 (自動登録)
- [F_ID]: 37 (単語登録処理後自動表示)
- [入力項目]: 住所
- [D_ID]: 38 (単語登録処理後自動表示)
- [入力データ]: 自宅
- 登録ボタンを押しデータ追加確定
- [識別IDリンク先]: 空欄をダブルクリック
- [識別IDリンク先]: 識別ID番号(主キー番号) 12 (自動登録)
- 識別ID 12 の入力フォームへ移動

入力フォーム										識別ID	1	閉じる
主キー	識別ID	T_ID	時点	ソート	ユニット	F_ID	入力項目	D_ID	入力データ	識別IDリンク先		
1	1	0	なし	0	0	4	識別名	20	人			
9	1	2	現在	1	1	21	氏	35	中島			
10	1	2	現在	2	1	22	氏ヨミ	36	ナカジマ			
4	1	2	現在	3	2	23	名	29	茂			
5	1	2	現在	4	2	24	名ヨミ	30	シゲル			
6	1	2	現在	5	0	25	性別	31	男			
7	1	2	現在	6	0	26	西暦生年月日	32	1964-01-08			
8	1	2	現在	7	7	33	E-mail	34	sko.jima@kita hama.or.jp			
11	1	2	現在	8	0	37	住所	38	自宅	12		
*												

履歴データ表示 [主キー] 別から採番した 登録

図9 リンク先の作成

Fig.9 registry of rinks

自宅を2つ以上所有している場合「自宅の数だけレコードを追加」することになる。

人以外の識別IDを作成するのは初めてになるがこの脳は2つしかテーブルを持っていないので(単語登録用の Word_Entity_tbl と識別ID化用の Identify_Entity_tbl だけ)逆に考えれば、自宅を識別化し識別IDを登録する方法は人の登録とまったく同じであることに気が付かずである。これより先は出来るだけ簡潔に解説していく。

5.2 入力補助機能 リストボックス

自宅を識別化する入力フォームから「リストボックス機能」を使い[D_ID]と[入力データ]へのデータ転記を紹介する。

- 識別ID 12 として「自宅」の入力画面が完成
- 41 都道府県名 をダブルクリックしリストボックスとして使用する都道府県名識別化フォームを作成(図10)

入力フォーム		識別ID		12					
主キー	識別ID	T_ID	時点	ソート	ユニット	F_ID	入力項目	D_ID	入力データ
12	12	0	なし	0	0	4	識別名	36	自宅
13		2	現在	1	1	39	国名	48	日本
14		2	現在	2	1	40	〒		
15		2	現在	3	1	41	都道府県名		
16		2	現在	4	1	42	住所1		
17		2	現在	5	1	43	住所2		
18		2	現在	6	1	44	建物名		
19		2	現在	7	0	45	電話番号1		
20		2	現在	8	0	46	電話番号2		
21		2	現在	9	0	47	FAX番号		
*									

図10 自宅の入力

Fig.10 entry of home address

- 入力フォームは、Identify_Entity_tbl から [T_ID]= 0 AND [D_ID] = 41 という検索条件を満たすレコードの有無を確認し、存在していればその識別 ID を選択する画面を提供し、まだ存在しなければ識別 ID 22 を新規作成する。50 大阪府 52 東京都 をそれぞれ登録する (図 11)。

入力フォーム		識別ID		22					
主キー	識別ID	T_ID	時点	ソート	ユニット	F_ID	入力項目	D_ID	入力データ
22	22	0	なし	0	0	4	識別名	39	都道府県名
23	22	2	現在	27	1	49	近畿地方	50	大阪府
24	22	2	現在	13	1	51	関東地方	52	東京都
*									

図11 都道府県名の入力

Fig.11 entry of prefecture

- 図 11 で、50 大阪府 をダブルクリックし、識別 ID 12 自宅の入力画面に戻り 41 都道府県名の [入力データ]へ 50 大阪府 の転記を完了することになる。

5.3 入力補助機能 - 利用回数を学習する -

入力によく使われる順から都道府県名を表示させたい場合、事前に[ソート]へ桁数の多い値を設定しておき、リストボックスとして利用され選択される度に入力フォームが、現在の[ソート]の値から、1 を引いた数字に更新していけば利用回数が多い順に表示されるはずであるが最初から[ソート]キーを降順にしておく、フォーム上の指定によりストアドプロシージャへソートの昇降を指定するのも手であろう。

5.4 入力補助機能 - 登録データの利用 -

図12は識別IDに登録されているデータから性別を検索した結果である。この方法ならリストボックス用の識別IDをわざわざ作成しなくても良い。

検索フォーム		識別ID							
主キー	識別ID	T_ID	時点	ソート	ユニット	F_ID	入力項目	D_ID	入力データ
6	1	2	現在	5	0	25	性別	31	男
*									

図12 登録データの検索

Fig.12 searching the data

6. 管理を始める脳

6.1 使用テンプレートの管理

図 13 は「人」の識別化をする入力テンプレートとしての指定した例である。これは自分自身を指定しているが[入力データ]が「全て未入力」の識別 ID を作成しておきそれを「テンプレート」として指定する方が良い。テンプレートの指定例だけだがこれでいろいろな使い方が出来る。

入力フォーム		識別ID		1		閉じる				
主キー	識別ID	T_ID	時点	ソート	ユニット	F_ID	入力項目	D_ID	入力データ	識別IDリンク先
30	1	0	なし	0	0	0	なし	49	使用テンプレート	1
1	1	0	なし	0	0	4	識別名	20	人	
9	1	2	現在	1	1	21	氏	35	中島	
10	1	2	現在	2	1	22	氏ヨミ	36	ナカジマ	
4	1	2	現在	3	2	23	名	29	茂	
5	1	2	現在	4	2	24	名ヨミ	30	シゲル	
6	1	2	現在	5	0	25	性別	31	男	
7	1	2	現在	6	0	26	西暦生年月日	32	1964-01-08	
8	1	2	現在	7	7	33	E-mail	34	skojima@kitahama.or.jp	
11	1	2	現在	8	0	35	住所	36	自宅	12
22	1	2	現在	9	0	35	弟	20	人	22
2	1	3	過去	1	1	21	氏	27	小島	
3	1	3	過去	2	1	22	氏ヨミ	28	ゴジマ	
*										

履歴データ表示

登録

図13 使用テンプレートの登録

Fig.13 registry of template

6.2 検索処理の特徴

この脳の特徴は、識別 ID の登録が完了した時点で単語の完全一致検索が可能になっている。日付「1964-01-08」や顧客番号「117114」にも [ID] を割り当てるので、Word_Entity_tbl に単語登録が無いのは「データベース上に

