

帰納的手法による関係データベースの設計

平尾隆行 國井利泰
日本アイ・ビー・エム株式会社 東京大学

概要:

関係データベース設計の主流は、設計者によってデータ間の従属性が分析・判断され決定されていくという“設計者視点法”とも呼ぶべきものであるが、本論文では、インスタンス・データ間の従属性を自動抽出し、それに基づいた関係によって半自動的に設計を行なう“帰納法”とも呼ぶべき方法を提案する。この方法は、設計しようとするすべてのインスタンス・データを入力すると、自動的に属性間の関連を調べて、第三正規形の複数枚の表の候補の何組かを出力する。本論文で述べる方法は、従来からのデータベース設計に有効に適用できると同時に、動的に変化するデータ、たとえばピクチャー・データなどには一層有用である。

Relational Database Design
Using an Inductive Method

Takayuki Hirao
Technical Training No.1
Education Division
IBM Japan, Ltd.
1, Kanda Izumi-cho,
Chiyoda-ku, Tokyo 101
Phone: (03) 865-5758

Tosiyasu L. Kunii
Department of Information Science
Faculty of Science
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113
Phone: (03) 812-2111 Ex. 4116

Abstract:

Database design has been done mainly by a method which can be called a designer view method. In this method, data dependency is analyzed, classified and stored by designers manually. This paper proposes a new approach which can be called an inductive design method. In this method, data dependencies are automatically extracted from the given instances of data, and, based on the derived data dependencies, the database is pseudo-automatically designed. The proposed method can be effectively applied to the traditional database design and also to dynamically changing data such as pictorial data more effectively. The great deal of manpower saving is expected by applying the proposed method.

1. はじめに

多くの分野に普及している関係データベースは、データを表で表現する使いやすいデータベースである。しかし、関係データベースの設計[19]は、機械化による自動的な方法ではなく、データ間の従属性を判断し決定するという正規化 [2,3]の考えに基づいた設計者による手作業（本論文では設計者視点法と呼ぶ）で行なっており、従来からのデータベースと同様に大きな労力を要する。設計者の視点による設計では、誰でも同じ労力で同じような品質の結果を出せるかといった問題がある。複数の設計者の視点は、自ずから少しずつ異なることがあり、設計者間の力量にも差があるからである。結果として、誰でも同じ品質の結果を出せる保証はない。

また、データベースの設計では、新規のデータベース設計のみならず、既存のデータベースの仕様の変更に対しても大きな労力を必要とする。既存のデータベースの変更とは、処理効率の観点からの設計の見直しや新しい属性の追加といったデータベース自身の特性の変化などを指し、よく発生する。新規データベースの設計のみならず既存データベースの仕様変更に対しても設計者の視点によることなく、より簡単に設計できるツールがあれば非常に効果的であることは言うまでもない。

本論文では、誰が設計しても、ほぼ同じ期間で同じような品質の設計を可能にする半自動的な方式を提案する。2章では属性の変更などが多発するデータベースの特徴について考察する。3章では、インスタンス・データを用いてデータベースを半自動的(pseudo-automatic)かつ帰納的に設計する方法（帰納的設計手法: inductive design method)の入力、処理、出力について述べる。入力に関しては、どのような条件を満足すればよいかについても述べていく。4章では帰納的的手法の効用について考察する。5章は結論である。

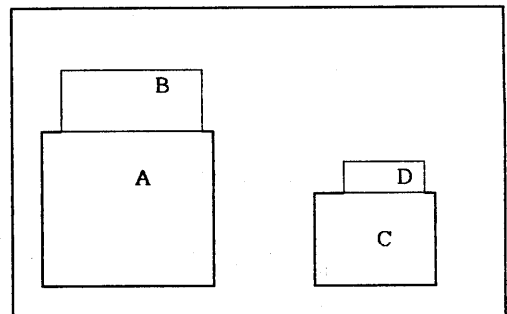
2. 多義性を持つデータベースの設計

データベースの設計は、非常に単純なデータベースの設計から、マルチメディアをサポートするような複雑なデータベースの設計まで様々である。使用する立場から見ると、単純/複雑にかかわらず、より簡単に

より早く設計し作りあげることが肝要になる。

複雑なデータベースには、CAD/CAMや画像データベースなどがあるが、これらの特徴は使用する人によって多義性を持つ。たとえば、画像データベースを例にとると[9,18]、図1に示した対象A、B、CおよびDが写っている写真のデータベースを作る場合には、次のような見方が考えられる。

- ・見方1： 写っている対象そのものに注目した見方 (図2参照)
- ・見方2： 写っている対象と、その重ね合せ位置に注目した見方(図3参照)
- ・見方3： その他



(注) BとDは影である。

図1. 写真のモデル・ケース

Figure 1. A Case of Sample Picture Model

PICTBL	SNAP#	OBJECT#	O-NAME
	1	1	A
	1	2	C
	1	3	B
	1	4	D
	1	5	BACKGROUND

図2. 図1のリレーション

Figure 2. Relation of the Figure 1.

PICTBL	SNAP#	OBJECT#	O-NAME	SUPERPOSITION
	1	1	A	1
	1	2	C	1
	1	3	B	2
	1	4	D	2
	1	5	BACKGROUND	3

図3. 図2のリレーションに列を追加したもの
Figure 3. New relation after adding new attributes.

このように、1つの写真でもいろいろな見方(意味)を持つ。ここに示したような多義性のあるデータベースの見方は適用業務の観点からの見方とみなすことができるが、このようなデータベースの利用中には新しい属性の追加を余儀無くされることがある。したがって、一度作成したデータベース・スキーマは、新しい適用業務の追加の度に変更を行なう必要が出てくる。

ここでは、複雑なデータベースの例によって、多義性のあるデータベースを見たが、単純なデータベースにおいても適用業務の追加は多かれ少なかれ発生する。すなわち、どのようなデータベースも多義性を持っているといってもよい。このような多義性を持っているデータベースの設計は、新規にデータベースを作成する時だけでなく、適用業務の追加の時にも行なう必要がある。どちらの時点においても、より簡単により早く設計できることが重要である。

3. データベースの帰納的設計

3.1 帰納的設計における入力と出力

データベースの設計には、図形的な設計手法としてのE-R図[1]や、関数従属性に基づいた正規化[2,3]などがあるが、これらはデータ間の関係を設計者の視点からまとめるものである。そもそもデータ間の関係を設計者の判断で決めているわけであるから、データがいくつか集まれば、自動的にデータ間の関係を見出すことができるはずである。本論文で提案する帰納的手法[9]は、いくつかのインスタンス・データをもとに、そのデータ間に現われる関係を見出して表のスキーマの候補を生成する。そのために必要とする入力要件は次のとおりである。

- (1) 設計しようとする対象業務の属性をまとめて、第一正規形で表す。属性名は、その第一正規形内でユニークにしておく[7]。
- (2) データは、対象業務の少なくとも1つのエンティティを表現する。エンティティを表すデータであるから、数行から数十行になる。
- (3) もしわかっていれば、キー属性が何かを示す。

重要なことは、設計時点で入手可能なある特定のエンティティに関連するデータはすべて入力することである。たとえば、図2は図1の写真に関するエンティティ情報を表すものである。

入力したデータに対して次節に述べる帰納的手法を適用することにより、いくつかの第三正規形の表候補を出力する。設計者は、表候補の中から最適と思われるものを選択すればよい。

帰納的手法への入力となるデータは、設計者あるいは分析者によって簡単に準備することができる。仮に、既存の表に対して新しい属性が必要になっても、その時点で再度帰納的手法に対してインスタンス・データを入力するだけで新しい表候補を作成しなおすことができる。

3.2 帰納的手法のアルゴリズム

いくつかのインスタンス・データを入力にして表候補を作成する帰納的手法のアルゴリズムは、次のとおりである。

- (1) 与えられた第一正規形のデータから1組の属性とデータ値を取り出す。このとき、最初の属性を「原始属性」、次の属性を「目的属性」と呼ぶことにする。
もし、入力要件としてキー属性が複数から成るときは、それらをあたかも1つの属性であるとみなすことにする。
- (2) 原始属性を大項目、目的属性を小項目としてグループ化する。そして、原始属性の各グループ値に対する目的属性の要素数をカウントする。そのカウント値のうち、最大値を求める。次に、原始属性と目的属性を交換して、同じ操作を行なう。

すなわち、2組の属性間の両方向へのマッピングを求めるわけである。マッピングの表現形式は、「属性1 (N : M) 属性2」とする[17]。ここで、Nは属性1のそれぞれの要素が属性2の0個以上の要素に対応することを表わし、Mは属性2のそれぞれの要素が属性1の0個以上の要素に対応することを表わす。

求まったマッピングを記録していくが、属性1 (1 : M) 属性2のときには、後の処理をしやすくするために、属性2 (M : 1) 属性1の形式で記録する。

この操作では、原始属性と目的属性の連関を求めているわけである。

- (3) 上記の1と2のステップを、与えられたインスタンス・データの全ての属性の組合せに対して行なう。
- (4) ステップ3までに求まったマッピングの左側の属性名を大分類とし、右側の属性名を小分類として分類した後に、(N : 1)マッピングのグループを取り出して、

属性1 (N : 1) 属性2 (M : 1) 属性3・・・の形式に順序づける。
ただし、順序づけは、同一属性名が2回以上出現しないようにする。
順序づけされた属性間の値の関係は図4のようになる。この関係は、関数従属性に基づいた第三正規形であり、表候補となる。

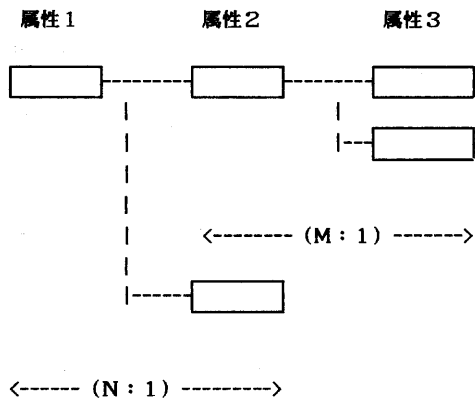


図4. 順序づけられた属性間の関係
Figure 4. The relation after sequencing.

- (5) ステップ4で作成した表候補間に推移従属を含む場合は、同じ属性名が複数箇所に出てくることがある。そのような場合には、設計者あるいは分析者に表候補を示し、推移従属によって得られる表候補を除いたり、アクセス頻度を考慮して、対話式に最適のものを選ばせるようにする。

上述したステップの1から4は自動的に処理を行なうことができるが、5のステップは対話式に処理を進めていく。この意味で、本論文に示す帰納的手法は半自動的であると言える。

3.3 帰納的手法の適用例

前節に示した帰納的手法のアルゴリズムの適用例を、図3のデータを入力に用いて示すことにしよう。入力となるものは、1枚の写真を表わすデータと属性であり、設計者あるいは分析者によって与えられる。与えられたデータをもとに次に示す処理を行なう。

- (1) 2つの属性名とデータを取り出す。最初は、SNAP#とOBJECT#と、そのデータである。
- (2) SNAP#, OBJECT#の順でグループ化して、SNAP#の各グループ毎にOBJECT#が何個対応するかを求める。
SNAP#とOBJECT#の組合せでは、図5に示すように、SNAP#の1個に対しOBJECT#の5個が対応する。
逆は、OBJECT#, SNAP#の順でグループ化しても変わらない。
結果として、SNAP#(5 : 1)OBJECT#が求まる。これは、属性1 (N : 1) 属性2の形式になっているので、そのまま記録する。

SNAP#	OBJECT#	=====>	SANP#	OBJECT#
1	1		1	1
1	2			2
1	3			3
1	4			4
1	5	5		

図5. 帰納的手法のステップ2の実行
Figure 5. Execution of the 2nd step of an inductive design method.

(3) 残りの属性間の組合せに対してステップ1と2を繰り返す。

結果として、図6に示すようなマッピングが求まる。

- (a) SNAP# (5:1) OBJECT#
- (b) SNAP# (5:1) O-NAME
- (c) SNAP# (3:1) SUPERPOSITION
- (d) OBJECT# (1:1) O-NAME
- (e) SUPERPOSITION (2:1) OBJECT#
- (f) O-NAME (2:1) SUPERPOSITION

図6. 帰納的手法のステップ3までの結果
Figure 6. Results of the execution through step 3.

(4) マッピングの結果から、左側の属性名、右側の属性名の順で分類した後に、(N:1)マッピングのグループを取り出して順序づけを行なう。分類の結果は図7のようになり、順序つけた結果が図8のようになる。その結果、3つの表候補ができたことになる。

- (d) OBJECT# (1:1) O-NAME
- (f) O-NAME (2:1) SUPERPOSITION
- (a) SNAP# (5:1) OBJECT#
- (b) SNAP# (5:1) O-NAME
- (c) SNAP# (3:1) SUPERPOSITION
- (e) SUPERPOSITION (2:1) OBJECT#

図7. 図6の結果を属性名で分類したもの
Figure 7. Results after sorting of the Figure 6.

- (g) SNAP# (5:1) OBJECT# (1:1) O-NAME
- (h) OBJECT# (1:1) O-NAME (2:1) SUPERPOSITION
- (i) O-NAME (2:1) SUPERPOSITION (2:1) OBJECT#

図8. 図7から求めた(N:1)マッピング・グループ
Figure 8. (N:1) mapping sequence from Figure 7.

(5) 図8の結果を設計者や分析者に示すことによって、最適の表スキーマを選択させる。このケースでは、(g)の順序づけと(h)の順序づけに含まれている推移従属などや表に対するアクセス頻度を考慮することによって設計者はリレーション(g)と(h)の一部を選択することになる。図9は、最終結果としてのリレーションである。

SNAP#	OBJECT#	O-NAME	O-NAME	SUPERPOSITION
1	1	A	A	1
1	2	C	C	1
1	3	B	B	2
1	4	D	D	2
1	5	BACKGROUND	BACKGROUND	3

図9. アクセス頻度を考慮した最終のリレーション
Figure 9. Final relations for Figure 3.

図9のような既成のリレーションに対して、色や境界といった新たな属性を追加しても、再度上記のステップを繰り返すだけでよい。

4. 帰納的手法の効果

帰納的手法と従来の設計者の視点による方法とを比較するために、図3の写真のデータを用いて設計者の視点による方法で設計をやってみよう。設計者は、設計に際して従属ダイアグラムを作成する。たとえば、図3に対する従属ダイアグラムは図10のようになる。

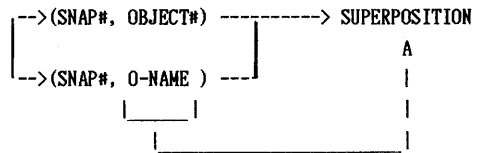


図10. 図3の従属ダイアグラム
Figure 10. A dependency diagram for Figure 3.

図10の従属ダイアグラムは、すでに述べてきた帰納的手法の結果の図9と同じである。帰納的手法を利用すれば、それだけ設計作業が減り、最適の表設計を素早く行なうことができるようになる。

次に、データベース・スキーマを半自動的に簡単に作ることができるため、既存の表に新しい列を追加してもその時点で帰納的手法を実行すればよい。たとえば、図3の写真のデータではさらに色やテキストなど多くの属性を追加した時点で帰納的手法を実行すればよい。エンティティの属性が動的に変化するようなCAD/CAMや画像データベースには帰納的手法が適しているといえる。

もちろん、エンティティの属性が動的に変化することが少ないようなデータベースであっても帰納的手法は効果的である。たとえば、設計すべきデータベースが多量のときは、設計にかける労力がかなり大きくなる。そこで、データベース設計の初期の段階で帰納的手法を適用すれば設計時間も労力も少なくなる。

5. 終りに

従来からのデータベース設計としての正規化は、設計者の視点から見た関数従属性をもとに設計を進めていく手法である。設計しようとする属性間の関係を見て関数従属性を決めている部分を機械的に行なうようにするのが、帰納的手法である。

帰納的手法は、現実には起こりうるエンティティのインスタンス・データを入力とし、データ自身がお互いに暗示している関係を一定の操作に基づいて見つけた後に表候補を出力する。帰納的手法は、データとデータのバイナリー・リレーション[16]に注目した単純だが強力な手法である。帰納的手法の結果は、設計者の視点による設計結果と同等以上のものである。さらに、既存のデータベースに対する属性の追加などの変更に対しても柔軟に対処することができる。

参考文献

1. Chen, P.P., "The Entity-Relationship Model: Toward a Unified View of Data," ACM Trans. Database Syst., Vol.1, 1(June 1976), pp.9-36.
2. Codd, E.F., "A Relational model for large shared data banks," Commun. ACM, Vol.13, 6, 1970, pp.377-387.
3. Codd, E.F., "Further normalization of data base relational model," in Data Base Systems, Courant Computer Science Symposia Series, Vol. 6, Rustin, R.,Ed., Prentice-Hall, Englewood Cliffs, N.J., 1972, pp.33-64.
4. Codd,E.F., "Extending the Database Relational Model to Capture More Meaning," ACM Trans. Database Syst., Vol.4, 4(Dec. 1979), pp.397-434.
5. Date, C.J., An Introduction to Database Systems, 4th ed., Addison-Wesley, Reading, Mass., 1986.
6. Date, C.J., Relational Database: Selected Writings, Addison-Wesley, Reading, Mass., 1986.
7. Fagin, R., Mendelzon, A.O., and Ullman, J.D., "A Simplified Universal Relation Assumption and Its Properties," ACM Trans. Database Syst., Vol.7, 3(Sep. 1982), pp.343-360.
8. Kent,W., "Consequences of Assuming a Universal Relation," ACM Trans. Database Syst., Vol.6, 4(Dec. 1981), pp.539-556.
9. Kunii, T.L., Weyl, S. and Tenenbaum, J.M., "A relational data base schema for describing complex pictures with color and texture," Proc. of 2nd Int. Joint Conf. on Pattern Recognition, 1974, pp.310-316.
10. Kunii, T.L., "DATAPLAN: An Interface Generator for Database Semantics," Information Science 10, 1976, pp.279-298.
11. Pernul, G., "An Unnormalized Data Model Based on User Views," ACM SIGMOD RECORD Vol.16, 2 (Sep. 1987), pp.51-60.
12. Pistor, P. and Andersen, F., "Designing a Generalized NF2 Model with an SQL-type Language Interface," Proc. 12th Int. Conf. on Very Large Data Bases, 1986, pp.278-285.
13. Reiter, R., "On closed world data bases," in Logic and Data Bases, Gallaire, H. and Minker, J. Ed., Plenum Press, N.Y., 1978, pp.55-76.
14. Smith, J.M. and Smith, D.C.P., "Database Abstractions:Aggregation and Generalization," ACM Trans. Database Syst., Vol.2, 2(June 1977), pp.105-133.
15. Teory, T.J. and Fry, J.P., Design of Database Structures, Prentice-Hall, Englewood Cliffs, N.J., 1982.
16. Tsichritzis, D.C. and Lochovsky, F.H., Data Models, Prentice-Hall, Englewood Cliffs, N.J., 1982.
17. Vetter, M., Strategy for Data Modeling, John Wiley and Sons, Chichester, 1987.
18. Yamaguchi, K., Ohbo, N., Kunii,T.L.,Kitagawa, H. and Harada, M., "ELF: Extended Relational Model for large flexible picture databases," Proc. of the workshop on Picture Data Description and Management, Aug. 1980, pp.95-100.

19. Zaniolo, C. and Melkanoff, M.A., "On the Design of Relational Database Schema," ACM Trans. Database Syst. Vol.6, 1 (Mar. 1981), pp.1-47.