

物理スキーマにリンク構造を用いたデータベースの設計

7J-7

木實新一* 古川哲也** 上林弥彦*
 *九州大学工学部 **九州大学大型計算機センター

1. まえがき

リンク構造では、リンクに沿ったデータ間の対応を効率よく求めることが可能である。リンク構造を用いたネットワークデータベースは検索の処理効率がよく、関係データベースでの結合処理効率化のための索引構造もリンク構造の一種であると考えられる。データベースの処理効率化の点から、リンク構造の導入は本質的であると考えられ、データモデルにかかわらず物理スキーマにリンク構造を用いることで、効率よいデータベースが実現可能となる。

データベースの設計法としては、

- (1)実体関連図を用いた直感的な方法
- (2)従属性制約に基づく方法(関数従属性[1], 多値従属性[2]等)。

- ・意味制約を反映し、冗長性を低くできる。

- (3)検索処理効率を考慮した方法[3, 4]。

- ・特定の検索の効率がよい。

- ・冗長性が付加され更新処理効率は悪くなる。

等がある。リンク構造は従属性制約を表現するだけでなく、構造が決定されるとその上で効率よく行うことができる処理とそうではない処理とが決まってくるため、設計時に従属性制約及び検索、更新の処理効率をあわせて考慮する必要がある。本稿では、まず更新処理に対して形式的な議論を行い、次に、従属性を反映し、検索処理、更新処理を効率よく行うリンク構造の設計法を示す。

2. 基本的事項

データベースにおけるリンク構造として代表的なものにネットワーク構造がある。ネットワーク構造は、バックマン線図と呼ばれる有向グラフ $B(V, E)$ で表される。 V はレコード型に対応する節点の集合、 E は親子集合型に対応する枝の集合である。図1はバックマン線図とその実現値の例である。

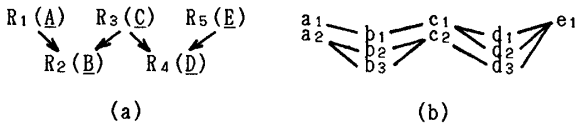


図1 ネットワークスキーマとその実現値

ここで、 $R(X)$ は属性集合 X からなるレコード型 R である。また、下線の属性集合はレコード型のキーであり、その値を定めれば各レコード型でレコードが一意に定まる。値を定めれば B の全てのレコード型でレコードが一意に定まるような属性集合を B のキーと呼び $K(B)$ で表す。一般に $K(B)$ は B 中の子レコード型を持たないレコード型のキー集合の部分集合であり、図1

(a)のキーは BD である。

検索処理効率のよいネットワーク構造を得るためには、ネットワーク構造に冗長なレコード型や親子集合型を加えたり、レコード型に冗長な属性を加えたりするとよい[3, 4]。

[例1] 図2は、属性 A と属性 E の値の対応が効率よく求められるように図1を変換したものである。冗長な構造として、図1の構造が表す属性 A と属性 E の値の対応をレコードとして保持するレコード型 $R_6(AE)$ と、それに付随する親子集合型を加えることで、検索処理を効率化している。図2では、レコード型 R_1 から R_5 までを見なくても R_6 のみで属性 A と E の対応を求めることができる。

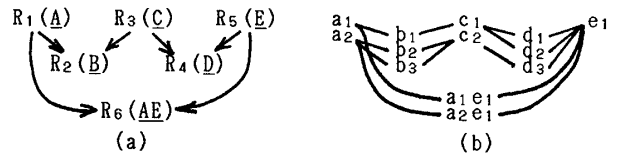


図2 検索処理を考慮したスキーマとその実現値

3. 更新処理効率のよいリンク構造

リンク構造に冗長性がない場合、更新は必要な部分を書き換える処理のみで行うことができる。しかし、リンク構造が冗長な場合は、データの書き換えのみでなく、更新のための検索処理が必要となってくる。

ネットワーク構造 B に冗長な構造 B^* が付加されたスキーマでの更新で必要な処理を次のように分類する。ただし、ここでは B に閉路が存在しない場合を対象とする。

- (1)被更新データ候補検索: B 中の更新するデータが B^* 中のどのデータと対応しているかを調べるための検索処理
- (2)更新決定検索: (1)で求められた B^* 中のデータに B 中の更新を反映させるか決定するための検索処理。
- (3)実更新: (2)での結果に応じて、(1)で求められた B^* 中のデータを実際に書き替える。

図3に冗長性のあるスキーマで更新時に必要な処理を示す。

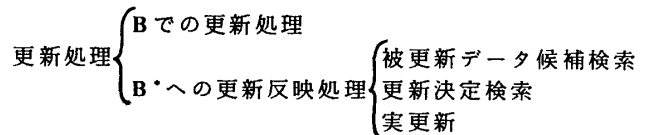


図3 冗長性のあるスキーマにおける更新時の処理

実更新は B^* 中のレコード数にほぼ比例する手間で行える。被更新データ候補検索は、 B^* に対応する B の必要な部分を検索する処理であり、ネットワーク構造のキーのデータ数を n とすると B 中のある 1 つのレコード型のすべてのレコードを更新する場合、 $O(n)$ の手間を要する。更新決定検索は、被更新データ候補それぞれについて B を検索する必要があるため、 B 中のある 1 つのレコード型のすべてのレコードの更新に $O(n^2)$ の手間を要する。しかし、スキーマが次の条件を満たすならば、更新決定検索が不要となる [5]。

[キー包含条件] 冗長な構造 B^* には、属性集合が B の対応する部分の超キー（キーの超集合）となるレコード型が存在する。

キー包含条件を満たすスキーマでは、 B^* 中のキーを包含するレコード型が B 中のレコード型をすべて祖先のレコード型にでき、 B からの更新反映を祖先から子孫の方向へ行うことができる。祖先のレコード型での更新は子孫の冗長なデータに必ず反映されるため、更新決定検索が不要となる。キー包含条件を満たすようにスキーマを設計すれば、冗長な部分に対する更新処理効率を考慮したスキーマが得られる。

4. ネットワーク構造の設計

従属性制約を反映し、検索、更新処理を効率よく行えるネットワーク構造の設計法を示す。

従属性及び検索処理、更新処理をそれぞれオブジェクト（属性集合）で表現する。従属性を表すオブジェクトは結合従属性の要素または関数従属性の閉包とし、検索処理のオブジェクトは対応を求めたい属性集合とする。これによって各従属性を反映し、検索効率もよいスキーマが得られる [4]。これらのオブジェクトから更新処理のオブジェクトを求め、更新の効率もよいスキーマを設計する。

[手続き 1] ネットワーク構造の設計

入力：従属性を表現するオブジェクトの集合

$$D = \{D_1, D_2, \dots, D_n\}$$

検索処理のためのオブジェクトの集合

$$R = \{R_1, R_2, \dots, R_m\}$$

出力：ネットワークスキーマ B

方法：

- (1) 更新のオブジェクトの集合 $U = \{U_1, U_2, \dots, U_m\}$ を求める。 $\forall D \in D$ について $R_i \not\subseteq D$ となるような R_i についての更新のオブジェクトを、 R_i を含むような最小の $U_i = U \cap D$ ($D' \subseteq D$) とする。
- (2) $D \cup R \cup U$ を含み、共通集合で閉じた最小の集合 S を求める。
- (3) S の各要素 X に対し、節点 X をつくる。
- (4) 節点 X_1, X_2 ($X_1 \subseteq X_2$) で、 $X_1 \subseteq X_3 \subseteq X_2$ となる節点 X_3 が存在しなければ有向枝 $\langle X_1, X_2 \rangle$ をつくる。
- (5) 各節点 X をレコード型 $R(X)$ 、各有向枝 $\langle X_1, X_2 \rangle$ を親子集合型 $\langle R_1(X_1), R_2(X_2) \rangle$ とする。
- (6) 可能なものについて、関数従属性からレコード型のキーを決定する。その他のレコード型 $R_i(X_i)$ のキーは X_i とする。キーが共通するレコード型は併合する。

(7) 冗長な属性を除き、結果のスキーマを B とする。

手続き 1 では、更新処理のオブジェクトが、検索処理のために作る冗長なレコード型に対応するネットワーク構造の部分の全ての属性を含んでいる。キーを包含するレコード型がこの更新処理のオブジェクトをもとにつくられ、最終的にキー包含条件を満たすスキーマが得られる。

[例 2] 手続き 1 を用いて、 $D = \{ABC, CDE\}$, $R = \{A, AE\}$, $U = \{ABCDE\}$ に対してネットワーク構造を設計した場合図 4 (a) のようになる。括弧内の属性は (7) で削除される属性を示している。(b) には実現値を示す。図 4 は、図 1 及び図 2 に対応しており、図 1 は従属性のみを考慮して設計されたもの、図 2 は従属性と検索処理を考慮して設計されたものである。図 4 ではキー包含条件が満たされ、更新も効率よく行える。

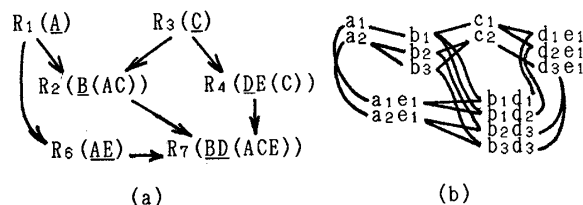


図 4 手続き 1 により設計したスキーマ

5. むすび

リンク構造を、従属性及び検索、更新の処理効率を考慮して設計する方法について述べた。検索処理効率を考慮するほど得られるスキーマは冗長なものとなって来る。キー包含条件を満たすようにして、更新の効率を余り悪くしないようにしても、冗長性の増加に伴う更新オーバーヘッドの増加を止めることはできない。このため、検索、更新について必要なもののみ効率化する方法を考える必要がある。また、シミュレーション等により、処理効率やメモリ消費の点から実用的な設計結果が得られるかどうか確かめる必要がある。

参考文献

- (1) Kuck, S.M. and Sagiv, Y., "Designing Globally Consistent Network Schemas", Proc. ACM SIGMOD Int. Conf. on Management of Data, pp. 185-195, May 1983.
- (2) Lien, Y.E., "On the Equivalence of Database Models," J. ACM, Vol. 29, No. 2, pp. 333-362, April 1982.
- (3) 古川, 上林, "質問処理効率化のためのネットワークデータベースのスキーマ変換", 電子情報通信学会論文誌, Vol. J71-D, No. 10, pp. 2111-2119, 昭和63年10月.
- (4) 古川, 上林, 木實, "ネットワークデータベース設計支援システムの開発", 情報処理学会研究報告, 88-DBS-67-5, 昭和63年9月.
- (5) 木實, 古川, 上林, "更新処理を考慮したリンク構造による物理スキーマ設計", 情報処理学会研究報告, 89-DBS-74-1, 平成元年11月.