

7L-2 システムのライフサイクルに適応した DB 設計情報管理方式

町原 宏毅 大久保 成隆 関根 純
(NTT 情報通信処理研究所)

1. はじめに

近年、情報資源辞書システム (IRDS) やリポジトリなど、各種設計情報をデータベース (DB) 化し、効率的な設計支援を行うシステムが提案されている。

このようなシステムで扱う設計情報に特徴的なことは、各設計情報間に導出関係が存在し、しかも一度に全ての情報が導出されるのではなく、徐々に情報が詳細化されていく性質を持っていることである。またさらに、基本設計において一部の概略情報から決めなければならない情報が存在し、詳細設計に於て、導出元の情報が詳細化した時点で、再度導出が必要になることである。そこで本稿では、DB 設計を例に、工程に応じて設計情報が詳細化した場合に、設計情報の変更をうまく管理する方式について述べる。

2. DB 設計の概要と設計情報

DB 設計の手順を大別すると、基本設計と詳細設計に分けることができる。基本設計では、DB の概略の設計を行い、システムとして必要なハード構成や方式等を決定する。また、詳細設計では、DB を構築するのに必要なパラメータを決定する。さらにそれぞれの設計は、図 1 に示すように、概念設計、論理設計、物理設計と 3 つの工程から成る。

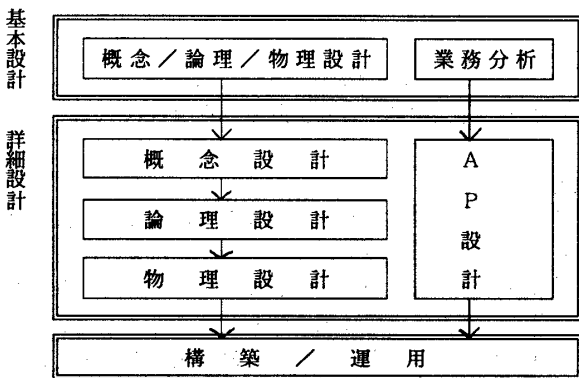


図 1 DB 設計の作業手順

概念設計では、データの従属関係を基に、テーブルを正規化し、論理設計では、検索性能を高めるために正規化されたテーブルに対して、統合や分解が行われる。また物理設計に入ってから、業務情報を基にしたインデックスの設計やファイルへの割り当てが行われる。

DB 設計で用いられる設計情報としては、列情報、テーブル情報、インデックス情報、ファイル情報、などが挙げ

られる。その設計情報の内、容量に関する情報が持つ導出関係をまとめると、図 2 のように、関数 f1 から f5 により導出される。

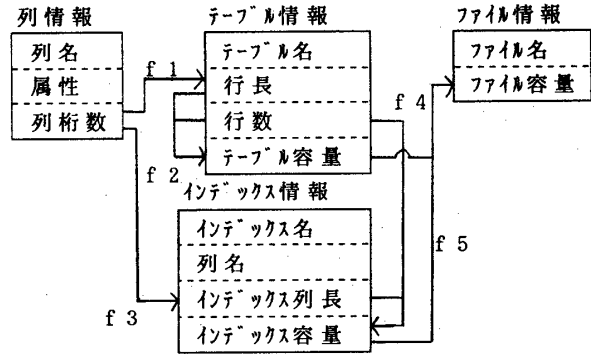


図 2 DB 設計における容量情報の導出関係

3. 設計情報管理の問題点

設計を行う場合、できるだけ早い時期にハードウェアの発注を行いたいという要求がある。そのため、基本設計では、列の詳細情報が全て導出される前に、テーブルの行長を推測したり、必要と思われる列にインデックスを付与したりして、設計を進め、導出された容量に対し安全率を掛けてファイル容量を見積るような先行設計を行う。

詳細設計に入ると、列情報やインデックス情報などが詳細に判明するため、ファイル容量も正確に計算できるようになる。しかし、既に安全率を見込んだ値に基づいてハード構成を決めているため、再構成を必要とするような大幅な変更があるような場合以外は、ファイル容量の変更を行わない。

つまり、設計情報間の導出関係の中には、必ず整合性を保たなければならないものや、ある程度保つ必要のあるもの、さらに工程によっては、保ってしまうと設計が進まなくなってしまうものが存在する。

4. 設計情報管理方式

3 節で挙げた問題を解決するために、基本設計ではある程度柔軟に設計を進めることができ、詳細設計に入っても、設計情報の修正を最小限に抑えることができるような設計情報の管理方式が必要になっている。

そこで従属関係にある、設計情報間の反映のパターンを 3 つ (波及的反映、モード別反映、条件付き反映) に分けることにより、設計情報を管理する方式を提案する。

4.1 波及的反映

「波及的反映」では、設計情報を変更した時に、その情報と導出関係のある設計情報との整合性を必ず保つように管理する。この方式は、予め設計情報間の導出関係を関数、あるいはアルゴリズムにより定義し、導出元情報の修正を「波及的に」導出先情報に反映するものである。

4.2 モード別反映

「モード別反映」は、2つのモードを用意し、初期段階で情報が曖昧な場合は、情報の変更があっても、導出関係にある情報には「反映しない」というモードに設定し、お互いの情報の変更を自由に許す。但し情報が詳細になってきたと設計者が判断した場合には、陽にそのモードを「反映する」というモードに変更する。

「反映しない」モードの場合は、情報間の相互作用は、存在しないものと同等であるが、一方の情報に変更があった場合は、他方の情報に影響を及ぼすと言う注意メッセージ(warning)を出すことにより設計者に注意を促すものとする。設計情報間の整合性は、「反映する」モードに変更した時に、保つたせるようにし、以降導出元情報を変更した場合には、必ず導出先情報に反映させる。

4.3 条件付き反映

「条件付き反映」は、導出元の情報を変更した場合に、その変更を導出先情報に反映した場合と、反映しない場合とを比較して、それらがある与えられた条件を満たしている場合は、反映しないことにし、満たしていない場合は、反映する。

5. DB設計情報への適応例

本稿の方式をDB設計情報へ適応した例を示す。まず、図2に示したDB設計情報間の導出関係を、本方式を用いて、図3のように管理することとする。ここで最初、モード別反映は、「反映しない」モードに設定し、条件付き反映の条件は、「修正後の値が、定義されている値の0.6倍から1.0倍以内であること」とする。本適応例で用いるDB設計情報の具体的な値を図4に示す。

この状態で、インデックスの付与されていない列(例えば「会社名」など)を削除したり、桁数を修正したり、また新たに列を追加しても、顧客テーブルの行長は変更されない。設計者は列情報とは独立して、行長を修正することができる。しかしインデックスが付与されている「顧客コード」の桁数を12バイトに修正した場合には、「波及的反映」となっているため、関数f3に従い、インデックス列長は12バイトに、関数f4に従い、インデックス容量も120KBにそれぞれ修正される。さらにファイル容量については「条件付き反映」となっているため、関数f5に従い、修正後の値(1,620KB)を求めるが、条件(1,440 ≤ 修正後の値 ≤ 2,400)を満たしているため、ファイル容量は変更されない。

詳細設計に入り、テーブルの正規化や、列の移動や複写

が行われ、列情報がある程度固まると判断した場合は、利用者が列桁数と行長との間のモードを「反映する」モードに変更する。この変更以後は、そのテーブルに対する列情報の変更(列削除、列追加、列長の修正)は、関数f1に従い、行長に反映されることになる。

以上のように設計情報を管理することにより、基本設計で記入した値をもとに、詳細設計を進めていくことを可能にしたのが本稿の設計情報の管理方式である。

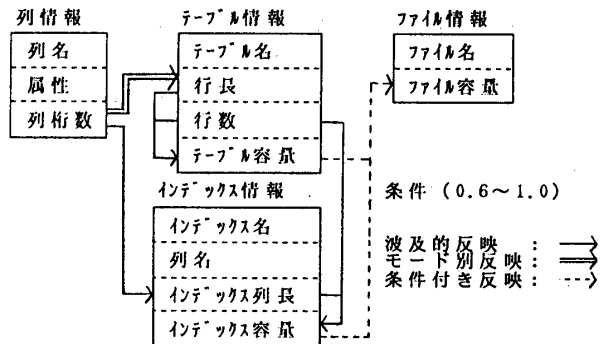


図3 設計情報変更時の反映パターン

列情報			テーブル情報			
列名	属性	桁数	テーブル名	行長	行数	テーブル容量
顧客コード*	CHAR	10	顧客TL	100	10 K	1500 K
顧客名	CHAR	20				
顧客住所	CHAR	30				
顧客電話	CHAR	8				
会社名	CHAR	12				

ファイル情報	
ファイル名	ファイル容量
顧客FL	*2400 K

インデックス情報			
インデックス名	列名	インデックス列長	インデックス容量
顧客IDX	顧客コード*	10	100 K

*)安全率1.5を掛けた値

図4 基本設計におけるDB設計情報例

6. おわりに

本稿では、DB設計を進めていく時に用いる、設計情報の管理方式を示した。本方式を用いることにより、基本設計段階で入力した設計情報を、詳細設計時に変更した場合でも、それら設計情報間の整合性を3つの方式で示した範囲内で保ちつつ、最小限の反映で、詳細設計を進めていくことができるようになる。今後、DB設計を支援するツールの中で、格納している業務情報、DB情報などの設計情報を管理するために本方式を採用していく予定である。

参考文献

[1] 関根、川下：DB設計のためのディクショナリ管理機能，情報処41全大2D-1，1990
 [2] 味村、山田、堀内：データベースシステムの設計と開発，オーム社，1983
 [3] DATE,C.J: Referential Integrity, Proc. 7th International Conference on Very Large Data Base pp.10-21(September 1981).