

第1章 データベースの基本概念と リレーショナル・データベース

見本

データベースの役割 / データモデル / データベースのアーキテクチャとデータ独立性 / データベースの利用者 / SQL / データベースの設計 / データベースの構成法 / 問い合わせ処理 / データベースの運用 / 障害復旧

1.1 データベースの役割

まずデータベースの技術(参考書[Elmasri 59][Oneil 51][Ramakrishnan 52][鶴保 50])について説明する。データの格納には従来ファイルが使われてきた。しかしながら、ファイルを利用して同じ内容のデータを重複して格納することを考えると、一般にファイルを利用するアプリケーション・プログラムごとにデータ・フォーマットが異なるし、重複データから記憶領域のむだづかいになるうえに、重複してデータを入れることにもなう労力の増大や入力されたデータ間に生じる相互矛盾という問題がある。また、ファイルでは大量のデータに対する効率的な処理や、複数のユーザからの同時アクセスやセキュリティ管理などの機能も十分ではない。

そこでデータベースは、ファイルとは異なる、以下のような役割をになっている。すなわち、

- ① データベースはデータ処理の基盤となるようなデータモデルを提供する。
- ② データベースは特定のアプリケーション・プログラムによらず、複数のプログラムや新たに作られたプログラムからも利用できる。
- ③ データベースはデータ処理を効率よく実行する。
- ④ データベースは複数のユーザから同時に利用できる。
- ⑤ データベースはそのデータが正しいことを保証する。
- ⑥ データベースは不正なアクセスからデータを守る。
- ⑦ データベースは障害が発生しても、元の状態に復旧することができる。

これらの各役割については、後に詳しく説明する。

ここでデータベースという用語をより正確に理解するための説明をしておこう。まずファイルというのは正確にはデータそのものか、もしくはその入れ物であり、ファイ

ルへの操作(作成,挿入,修正,削除など)のインターフェースは,ファイル・システムとよばれる,OS(オペレーティング・システム, Operating System)の機能の一部として提供される.それと同様にデータベースはデータそのものか,もしくはその入れ物であり,それに対する機能(上記の役割に相当する,正確にはそのインターフェース)を提供するのはデータベース管理システム(Database Management System, DBMS)とよばれる汎用ソフトウェア・システムである.

データベースは,現実世界のある側面を表現していることから,小世界(mini-world)または議論対象領域(Universe of Discourse)といわれることがある.しかしながらデータベースという言葉でデータベース管理システムをもあわせて指し示す場合があることにも注意してほしい.

1.2 データモデル

データモデルとは,データベース管理システムの処理機能の基盤を与える概念であり,以下の三つの構成要素からなる.

- ① データ構造: データ型やデータ間の関係を定義するための記述要素の集まり
- ② データ操作: データの検索や更新を行う操作の集まり
- ③ データ制約: データやデータ間に成り立つ制約条件を定義するための記述要素の集まり

通常,データ構造とデータ制約はデータ定義言語として,データ操作はデータ操作言語としてデータベース管理システムによって提供される.データモデルの種類には,階層型,ネットワーク型,関係型,オブジェクト型などがある.ネットワーク型と階層型は技術的に古いので,レガシー(遺産的)データモデルといわれる.レガシー・データモデルについては他の教科書(たとえば[Elmasriら99])を参照されたい.本書では,現在広く使われるモデルとして関係型を,比較的新しいモデルとしてオブジェクト型を取り上げて説明する.ここではまず関係型データモデルについて説明する.オブジェクト型データモデル(オブジェクト指向データベース)については節をあらためて説明する.

1.2.1 関係データモデル

関係データモデルは,数学における関係(またはリレーション, relation)をもとにデータベースを表現する.以下にくわしく関係データモデルのデータ構造を説明しよう.

関係データモデルでは,リレーションとリレーショナル・スキーマ(relational schema)を区別する.リレーションはデータベースそのものであり,個々のデータに相当する.リレーショナル・スキーマはデータベースに関する記述であり,データ型に相当する.リレーショナル・スキーマをインテンション(intention)といい,それに対してリレー

ションをエクステンション(extension)ということがある .

リレーショナル・スキーマは , リレーション名と複数の属性名からなる .

(定義)リレーショナル・スキーマ

$R(A_1, A_2, \dots, A_n)$ をリレーショナル・スキーマという . ここに R はリレーション名であり , 属性 A_i は定義域 D_i を持つ . さらに n をスキーマの次数という .

たとえばワインというリレーションは {ID, 名前, 産地, 品種, 色, タイプ, ピンテージ, 価格} という属性からなるスキーマを持つ . 各属性は対応する定義域を持ち , 属性の値はその定義域に属していなければならない .

(定義)リレーション

リレーション $r(A_1, A_2, \dots, A_n)$ は , 各属性 A_i の定義域 D_i に関する直積集合 $D_1 \times \dots \times D_n$ の部分集合である .

リレーションはしばしば図表 1.1 のように表(テーブル)として表される . 表の名前がリレーション名に相当する . 表の項目名が属性名に相当する . リレーションの要素はロウ(row) , タプル(tuple)またはレコード(record)といい , 表の横の 1 行にあたる . それに対して項目名を除いた縦の列をカラム(column)という . 本書では属性や



図表 1.1 リレーショナル・スキーマとリレーション

フィールドもカラムと同様な意味で使用する。ただしリレーションでは、個々のタプルの順番に意味はないし、項目名の順番にも意味はない。ある時点でのリレーションの状態は、スナップショット(snapshot)と呼び、個々のデータであるインスタンス(instance)あるいはオカランス(occurrence)の集合であるという。

1.2.2 関係演算

関係データモデルのデータ操作について説明する。操作は関係演算(relational operation)とよばれ、それには、選択、射影、和集合、差集合、直積というリレーションに対する基本演算がある。加えて共通集合、結合・自然結合、商集合という拡張演算があり、それらは基本演算の組み合わせで定義できる。これらを総称して関係代数(relational algebra)という。関係演算には関係論理(relational calculus)という別の体系があるが、それは表現能力の点では関係代数と等価である。関係代数がより手続き的であるのに対して、関係論理はより宣言的に問い合わせを表現する。関係論理については他の教科書([Elmasri 59])などを参照されたい。

選択操作

関係代数における選択(select)は、リレーション R について、属性に関する条件(選択条件)を満たすタプルを求める。一般形は以下のように記述する。

$$\sigma_{\text{選択条件}}(R)$$

ここに選択条件は以下の単純条件をブーリアン演算子(AND, OR, NOT)で結合したものである。

$$\begin{aligned} & \text{属性名 比較演算子 定数 または} \\ & \text{属性名 比較演算子 属性名} \end{aligned}$$

ここに比較演算子は $\{=, <, >, \neq\}$ のどれか一つである。定数は定義域の定数である。ただし選択条件の列は、どのような順で実行しても結果はかわらない。

図表 1.2(a)にワインのリレーション(図表 1.1)に対する選択操作とその結果のリレーションを示す。

射影演算

射影(project)は、リレーション R の特定の属性(カラム)を求める。一般形は以下のようになる。

$$\pi_{\text{属性リスト}}(R)$$

ここに属性リストは属性名をカンマで区切ったものとする。

図表 1.2(b)にワイン・リレーション(図表 1.1)に対する射影演算とその結果を示す。