

# 検索を科学する

塩田 紳二

## 第8回 データベース

コンピュータ関連の記事やニュースなどでよく見かけるデータベースだが、実際のところいろいろな意味で使われている。狭い意味では、データ(情報)の蓄積と管理を行うデータベースシステムを指すが、広い意味では、コンピュータがデータのまとまりを扱うことをいうことがある。

### データベースとは?

コンピュータの処理でデータを扱う場合、定型のデータを複数処理することが少なくない。このような処理を専門に行うシステムを、データベースシステムと呼ぶ。

多くのソフトウェアはなんらかの形でデータを扱うが、特にデータの蓄積や管理が処理の主体となるような場合、データベースシステムを使ってアプリケーションを構築することで、ゼロから作るよりも簡単にプログラムを作ることができる。このため、OSによっては、データベースシステムを内蔵していて、アプリケーションに対して基本機能として提供しているものもある。

また、よく名前を聞く Oracle や SQL Server、PostgreSQL などは、このデータベースシステムの名称である。こうしたデータベースシステムは、基本のデータベース機能に加え、プログラミングの機能(専用言語など)を持っており、たとえば顧客管理だとか、在庫管理といったアプリケーションを構築できるようになっている。

### データベースの基本はレコード

データベースには、いろいろなものがあり、さまざまな形のデータを扱うことができるが、基本としては情報をレコードという単位で扱う。レコードは、たとえば、名前や電話番号などからなる顧客情報などの一定の形式を持つ情報である。このレコードの集まりをテーブル(表)という。これをデータベースと呼ぶことも

あるが、多くのデータベースシステムでは、データベースは1つ以上のテーブルと、関連するさまざまなデータをひとまとめにしたものとして定義されることが多い。

レコードは、フィールドと呼ばれる単位から構成されている。フィールドは、氏名とか商品番号などの一定のサイズ、単一の形式のデータである。通常、レコードは1つ以上のフィールドの集まりである(図1)。

また、レコードを構成する個々のフィー

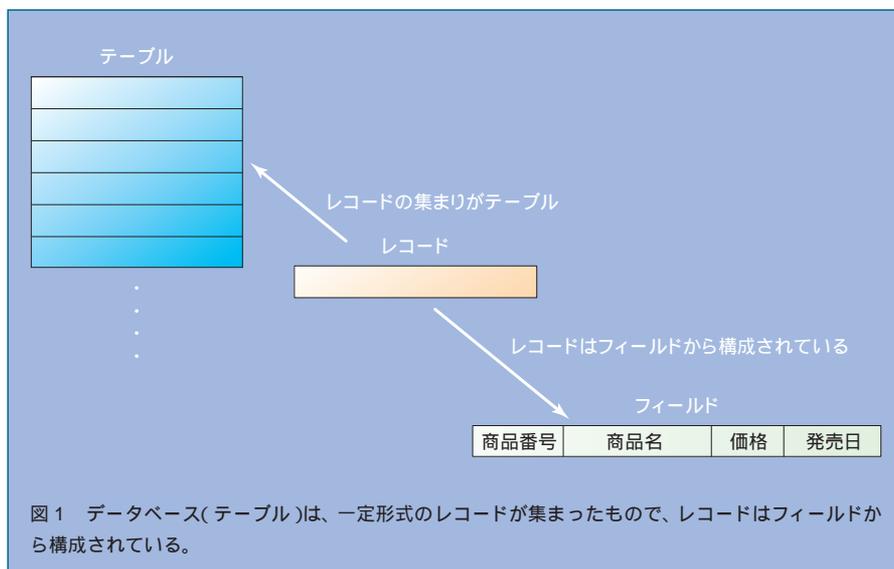


図1 データベース(テーブル)は、一定形式のレコードが集まったもので、レコードはフィールドから構成されている。

ルドを持つ、形式やサイズなどをまとめてフィールド定義(あるいはスキーマ定義)と呼ぶ。フィールド定義とは、簡単にいえばレコードをフィールドの定義の集まりとして定義することだ。

なお、スキーマ(Scheme)は日本語では一般にスキームと表記されるが、データベース関係ではスキーマと表記するのが普通だ。また、フィールド定義よりももう少し広い意味で、レコードの物理的なデータ配置などを含んで使われることもある。

フィールドは、たとえば氏名なら文字列、製品価格なら金額、購入日時なら日付といった形式を持つ。どのような形式が使えるのかはデータベースによって違うが、文字列や整数、日付、金額といった形式はほとんどのデータベースで利用が可能だ。

さらにフィールドは、形式に応じたサイズを持つ。文字列ならば最大の文字数、金額ならば最大桁数などである。

データベースシステムによっては、可変長のフィールドを扱うことができる。たとえば、文字列の長さを可変長としておき、大量のテキストを入力するなどということが可能だ。また、ユーザーが形式を定義できたり、すでに定義されている形式でも、その範囲を制限することができる場合もある。たとえば、整数であっても負の数を許さないようにすることなどができる。ただ、こうした付加機能は、フィールド自身の特徴として指定できる場合と、入力時の検査で行う場合もある。つまり、フィールド自体は負の数を入れられるようになっているが、データの入力時に負の数を入れないように検査する方法もある。

## 一般的なデータベースシステム

データベースシステムは、テーブルを動作するプラットフォームのファイルシス

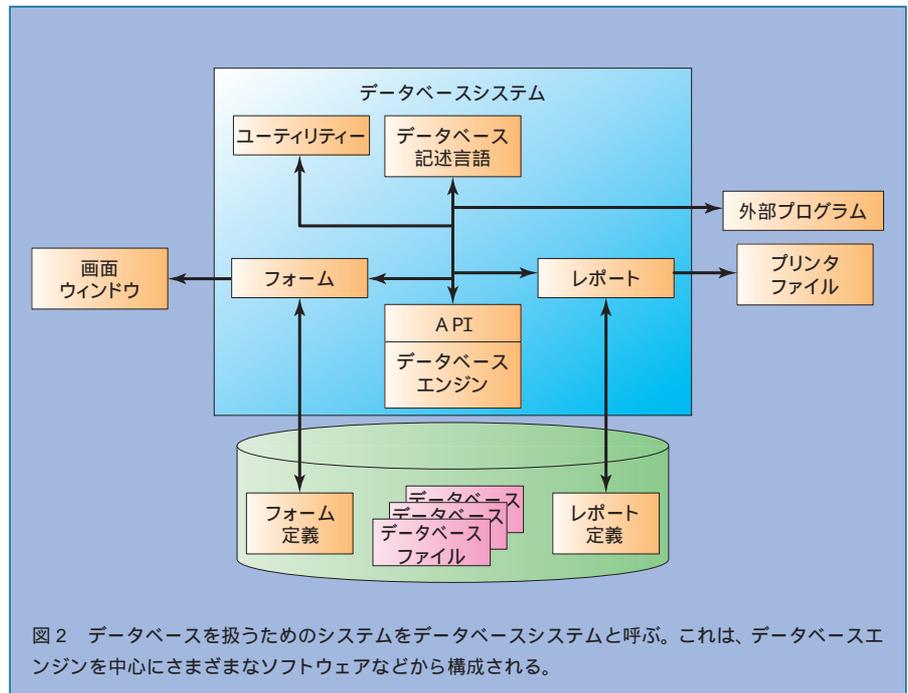


図2 データベースを扱うためのシステムをデータベースシステムと呼ぶ。これは、データベースエンジンを中心にさまざまなソフトウェアなどから構成される。

テムに格納する。このとき、特殊な構造を使って、レコードへのアクセスが高速に行えるように工夫する。また、インデックス情報を作成し、検索を高速化することがほとんどだ。

こうしたテーブル自体を扱う部分をデータベースエンジンと呼ぶ。

しかし、これだけでは実際の役に立たないので、データベースシステムには、以下のような要素が同時に提供されるのが普通だ(図2)。

- フォーム構築、実行環境
- レポート機能
- データベース記述言語
- プログラミングインターフェイス
- メンテナンスツール

のフォーム構築、実行環境とは、データを入力するためのものである。たとえば、顧客データを入力するには、氏名欄や電話番号欄など、フィールドに対応したデータを入力できるような画面が必要になる。こうした入力画面をフォームと呼ぶ。

データベースシステムでは、レコードの

フィールド定義から半自動的にフォームを作成することができる。もちろん、場合によっては、すべてのフィールドを常に入力する必要はないし、フィールドによっては、計算などで求めるものもあるだろう。こうした入力作業を行うプログラムを作成するための環境がフォーム構築機能である。また、そのフォームを使って実際の入力を行う機能がフォーム実行環境である。

テーブルに対して、検索や並べ替えなどを行わせた結果を出力させる機能をレポートと呼ぶ。レポートは、テーブルに登録されているレコードから特定の条件を満たすものを取り出し、そのレコードから必要なフィールドのみを取り出して表形式などにして出力する。また、必要に応じてテキストを付加することもできる。たとえば、レポート機能は顧客テーブルから住所ラベルを作ったり、昨日注文を行った顧客のリストを作ったりすることができる。

テーブルから特定の条件を満たすレコードを取り出すことを「問い合わせ」またはクエリー(Query)と呼ぶ。このときに指定する条件を検索条件(あるいはクラ

イテリア: Criteria)という。このときに使う条件を記述する言語を「問い合わせ言語」(Query Language)という。

問い合わせは検索と呼ばれることもあるが、一般的な検索と区別するため、データベース関連では、問い合わせやクエリーという用語を使う。問い合わせは、実際にはデータベースエンジンが行う。

レポートは、主に集計が必要な表示や印刷を前提とした計算処理などを想定している(図3)。単に、1件のレコードの内容などをコンピュータ上で表示させたい場合などには、フォーム表示を利用する場合もある。

前述したようにデータベース機能を必要とするアプリケーションは少なくない。特にビジネス系の処理のほとんどはなんらかの形でデータベースに関連するか、または、データベースの基本機能の組み合わせで実現できる。このため、データベースシステムは、テーブルを扱うための専用のプログラミング言語を持っているのが普通である。

その他、C言語などの一般的なコンピュータ言語からデータベースシステムを使うためのAPI(アプリケーションプログラミングインターフェイス)も持っていることが多い。

データベースシステムでは、本来の機能とは別に、テーブルの作成や消去、アクセス権の設定など、いろいろな設定が必要になることが多い。こうした設定などについては専用のプログラムが用意されているのが普通である。これらは多く場合、データベース記述言語や上記のAPIを使って作られている。

## テーブルとビュー

テーブルに対して問い合わせを行って得られたレコードの集まりも一種のテーブルとみなすことができる。このため、これをテーブルとしてさらにデータベース処理を行うことも可能になる。

実際にファイル上に作られるテーブルを実テーブルという。これは、実体があるという意味である。

これに対して問い合わせによって得られたテーブルは実体はなく、実テーブルと検索条件から作られるものだ。このようなテーブルをビュー(View)と呼ぶ(図4)。あるいは仮想テーブルという呼び方を使うこともある。

データベースシステムの処理は、ビューも実テーブルも区別なく行える。また、ビューを作るときに、レコードそのままの形でなく、特定のフィールドだけを取り出して仮想的なレコードを定義できるシステムもある。

ビューは、頻繁に利用する問い合わせや表示のために予め作成されることが多い。実テーブルと違うのは、ビューは、利用されるたびに問い合わせが行われるため、刻々と変化する可能性があることだ。

このため、ビューを定義する問い合わせ条件などを「外部スキーマ」定義と呼ぶこともある。これは、スキーマ定義を、実テーブルを定義するもの(概念スキーマ)、ビューを定義するもの(外部/副スキーマ)、レコードの記憶方法を定義するもの(内部/記憶スキーマ)の3段階で表現する考え方である(図5)。

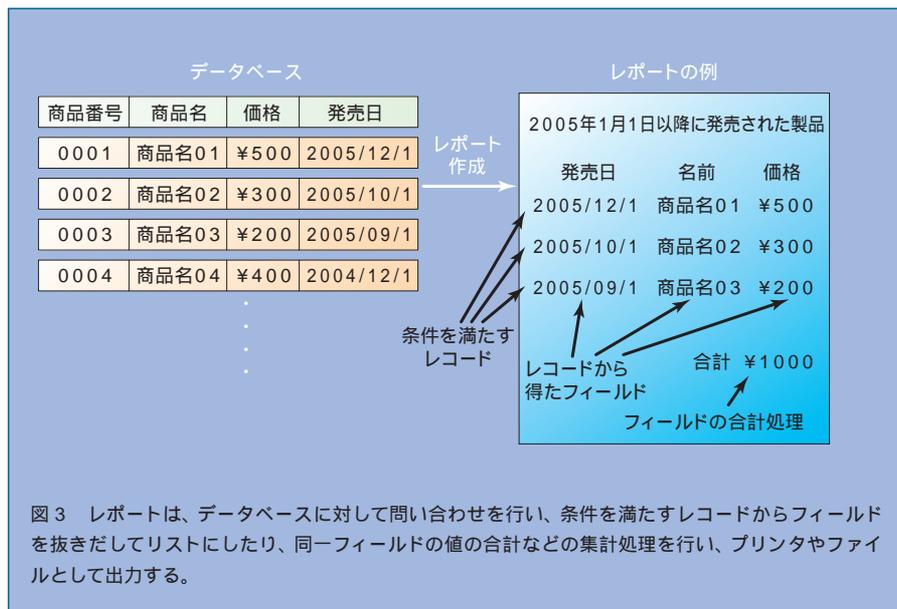


図3 レポートは、データベースに対して問い合わせを行い、条件を満たすレコードからフィールドを抜きだしてリストにしたり、同一フィールドの値の合計などの集計処理を行い、プリンタやファイルとして出力する。

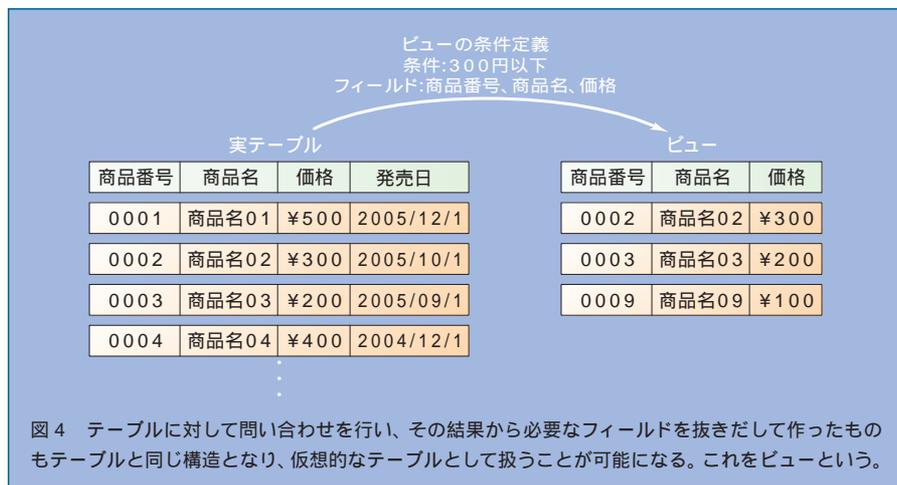


図4 テーブルに対して問い合わせを行い、その結果から必要なフィールドを抜きだして作ったものもテーブルと同じ構造となり、仮想的なテーブルとして扱うことが可能になる。これをビューという。

## テーブルを組み合わせて使う リレーション処理

企業などが商品を売る場合、在庫を調べ、商品があればそれを顧客に発送し、また請求書などを作成する。つまり、顧客からの注文が在庫を変化させ、請求処理などが発生することとなる。

小さな会社でもなければ、こうした処理にはそれぞれ担当部門があり、それぞれが管理を行っている。それぞれの処理をコンピュータで行うには、顧客、在庫、受注といったそれぞれをテーブルとして定義する必要がある。しかし、顧客の名前や住所は、在庫の発送先にも必要だし、請求書の請求先名や発送先にも必要となる。これを個々のテーブルで管理しようとする、顧客が引っ越ししたり、電話番号が変わったときに、すべてのテーブルで変更を行う必要が出てくるが、それぞれの担当が違っているなら、間違いが入る可能性がある。

このような複数のテーブルを使う場合、他のデータベースが持つ情報を取り出せるようにしておけば、1か所を訂正すれば、他のテーブルは変更を行う必要がなくなる。

上記の例でいえば、顧客のデータベースを作り、個々の顧客に番号を割り振り、発送先や請求処理などでは、この顧客番号を指定しておく。在庫処理で発送先として名前や住所が必要などときには、この顧客番号を元に顧客データベースに問い合わせを行い、名前や住所を得ればよい(図6)。

あるいは在庫管理でも、個々の商品を管理するテーブルと、商品番号と商品名、価格などを管理するテーブルを分けることで、個々の商品を管理するレコードに製品名などを重複して持つ必要がなくなる。もし、個々の製品を管理するレコードに製品名を含めると、同種の商品の各レコードに同じ情報を含むフィールドが重複して含まれることになる。

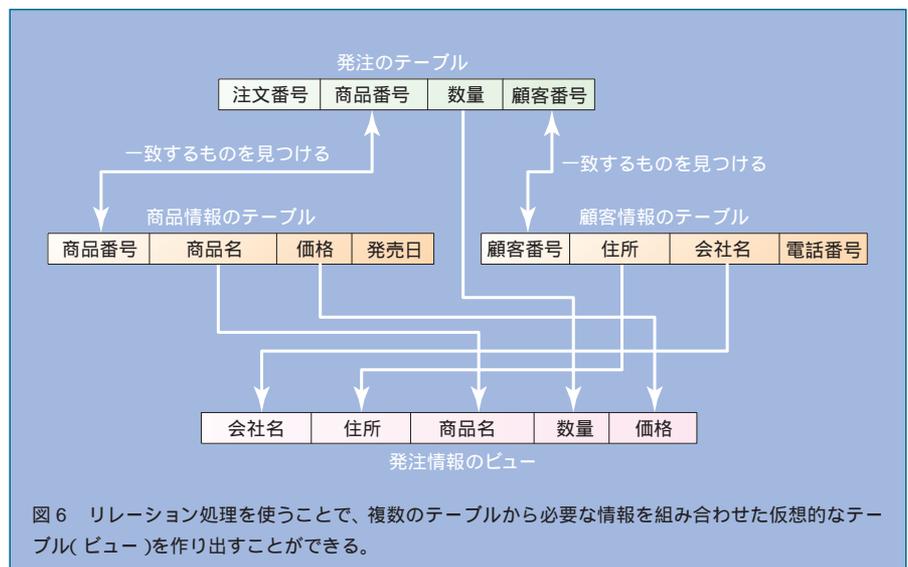
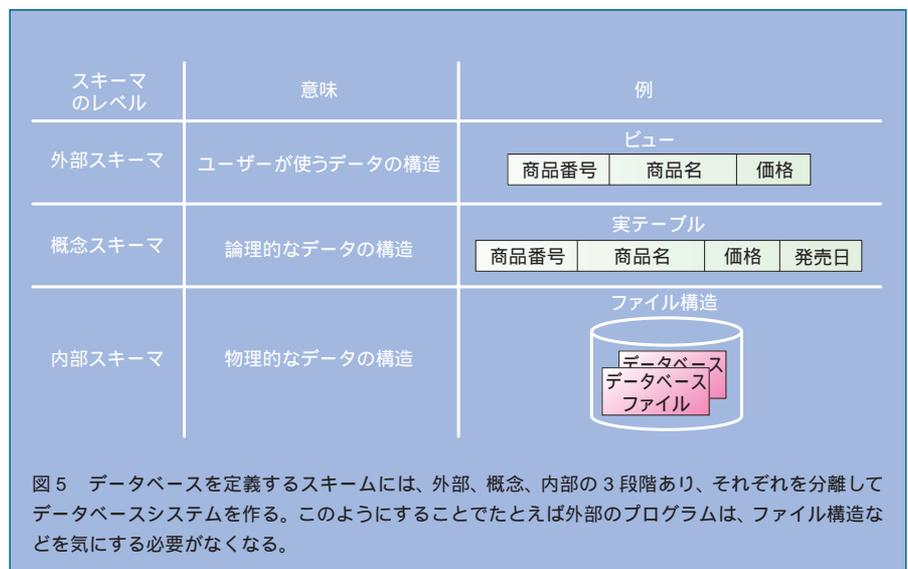
このような処理は「リレーション処理」と呼ばれ、この機能を持つデータベースシステムを「リレーショナル・データベース」と呼ぶ。広く業務用として使われているデータベースシステムの多くは、このリレーション機能を持っている。

このリレーション操作だが、基本的には2つのテーブル(あるいはビュー)に含まれるレコードから同じ意味を持つフィールド同士に着目して処理を行う。

このようなりレーション処理の結果として得られる情報は、2つのレコードを組み合わせたもので、これもビューとなる。

検索システムでは、大量のデータを扱

うこともあり、内部的にデータベースシステムを利用することもある。通常データベースシステムは、問い合わせを高速に行うためにインデックスの作成や、特殊なファイル構造を使う。また、ウェブサイトなどでも、ページをデータベースシステムが管理するレコードから生成する場合もある。たとえば、通販サイトの商品情報のページや注文履歴などはデータベースの情報を利用する。このため、データベースシステムによっては、レポートをHTML形式として出力可能なものや、HTTPサーバーと連携できるものもある。





## [インターネットマガジン バックナンバーアーカイブ] ご利用上の注意

このPDFファイルは、株式会社インプレスR&D(株式会社インプレスから分割)が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

<http://i.impressRD.jp/bn>

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容(技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど)は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者(執筆者、写真の撮影者、イラストの作成者、編集部など)が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元(株式会社インプレス R&D)、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D(当時は株式会社インプレス)と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接のおよび間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先

**株式会社インプレスR&D**

All-in-One INTERNET magazine 編集部

[im-info@impress.co.jp](mailto:im-info@impress.co.jp)