

---

# 第1章 ファイルとデータベース

---

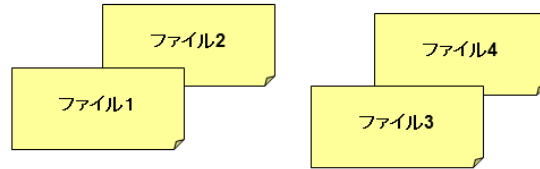
データの管理方法  
ファイルの問題点  
データベースによるデータの管理  
データベースの種類  
階層型データモデル  
ネットワーク型データモデル  
リレーショナル型データモデル

## この章のねらい

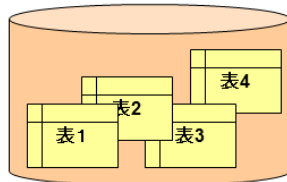
この章では、コンピュータを使った情報管理のしくみを紹介します。  
ファイルとデータベースを比べることで、データベースで情報管理する  
メリットを理解します。

## データの管理方法

- ファイル



- データベース



### ■ データの管理方法について

コンピュータが登場するまでは、データは台帳などの紙を用いて管理されていました。しかし、現代のように大量のデータを扱う時代では、コンピュータを用いたデータ管理が欠かせません。コンピュータを用いることにより、大量のデータを効率よく管理、活用できます。

コンピュータを用いたデータ管理の方法としては、おもに次の方法があります。

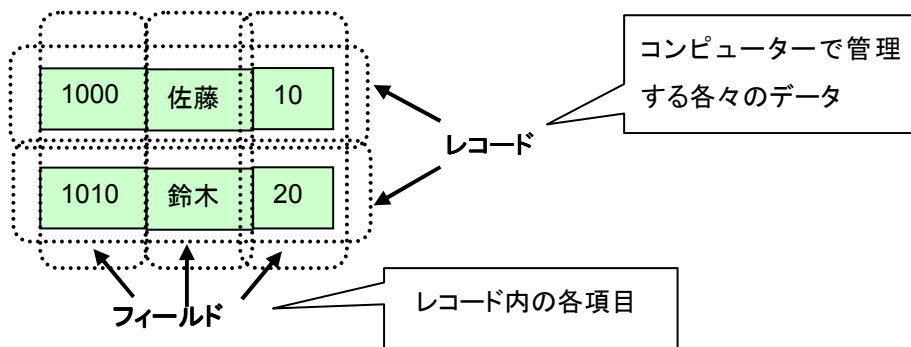
- **ファイルを用いたデータ管理**

業務アプリケーションに必要なデータを格納したファイルを OS(オペレーションシステム)のファイルシステム上に配置します。

- **データベースを用いたデータ管理**

データベース管理システムによって管理されたデータの集まりです。

### 【参考】レコードとフィールド



## ファイルの問題点

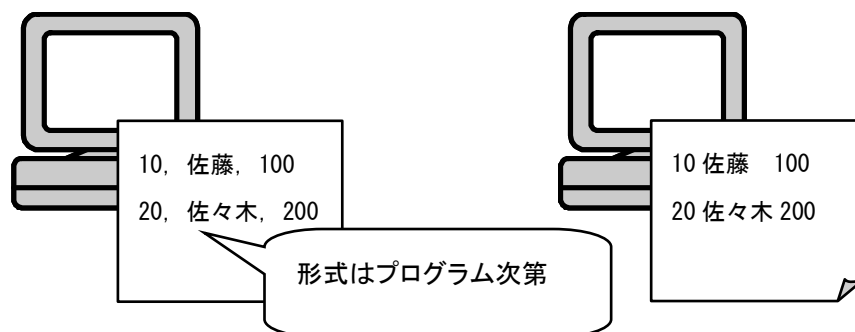
- プログラム依存
- データの重複
- データの矛盾

コンピュータが利用され始めた頃のハードウェアは高価だったこともあり、コンピュータを使用できる業務は限られていました。従って、特定業務のためだけに専用のシステムを開発することも珍しくありませんでした。この場合、データは特定のプログラムのためだけに存在し、プログラム固有の「ファイル」を用意することでデータの管理を行なうことが可能です。

しかし、コンピュータで管理する業務の数が増えると、ファイルの数も比例して増えていきます。このため、さまざまな問題が発生するようになりました。

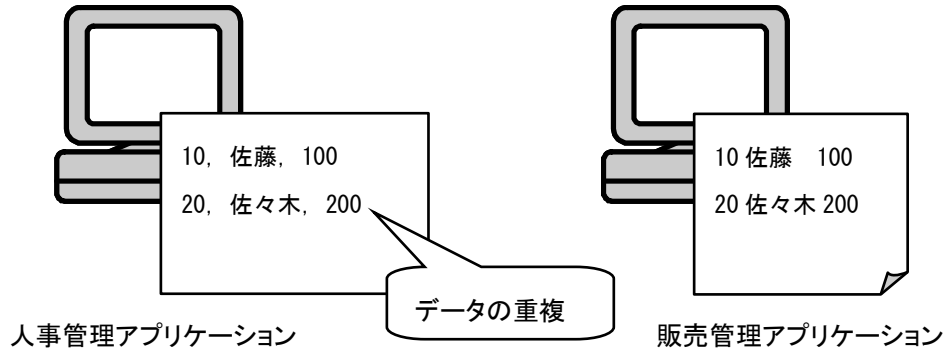
### ■ プログラム依存

プログラミングする際に、ファイルの物理構造を意識する必要があります。つまり、アプリケーションが用いるファイルの形式は、アプリケーションに依存しています。そのため、他のアプリケーションから利用できないことや、利用できてもデータの処理効率が悪くなるなどの問題が発生します。



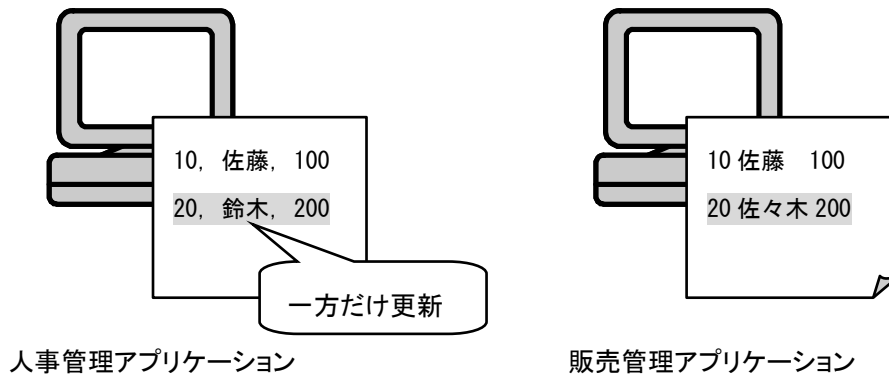
## ■ データの重複

プログラム依存の問題から、アプリケーションごとにファイルが用意されるため、類似したデータを格納したファイルが多数作成されることがあります。そのため、ファイル間にデータ重複が発生します。



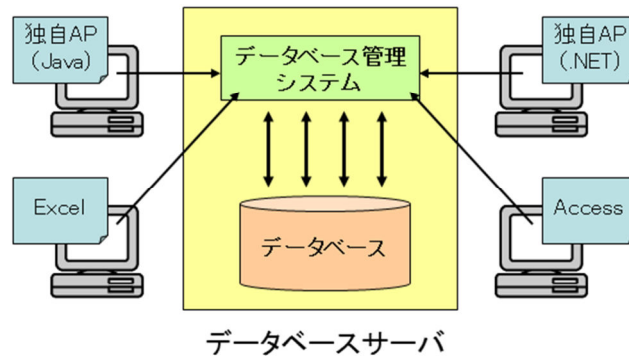
## ■ データの矛盾

データの更新をおこなう場合、データ重複の問題からデータの更新漏れが発生しやすく、全体としてのデータの一貫性を保つのが困難になります。また、重複データをすべて更新すると、処理効率の低下につながります。



## データベースによる データの管理

- データ独立
- データの一元管理



ファイルを用いた場合の問題点を解決するために、関連するデータを整理・統合して重複を省いた無駄のないデータ集合体が求められるようになりました。これを実現するのがデータベースです。

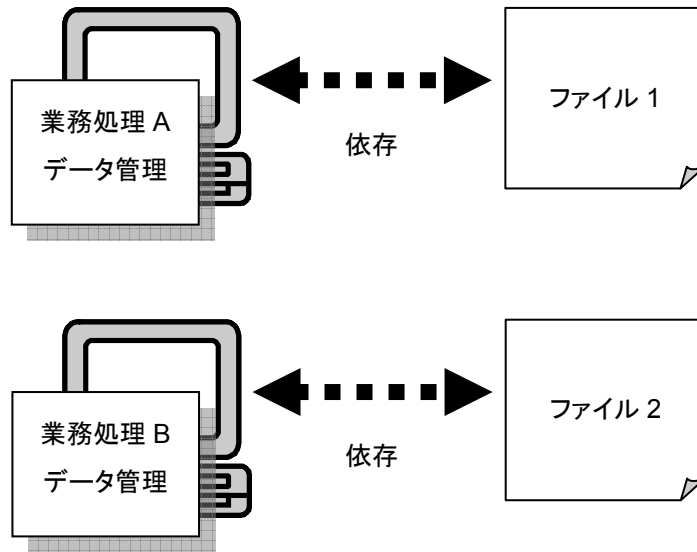
データベースとは次の特徴を持ったデータの集まりのことを意味します。

### ■ データ独立

データ独立とは、管理されるデータが特定のアプリケーションに依存しないことを意味します。つまり、プログラムの修正が生じたとしてもデータを変更する必要がありません。逆に、蓄積されたデータに変更が生じてプログラムを変更する必要はありません。データの格納場所(物理構造)をプログラムは意識する必要がないわけです。これを可能にするには、データ管理の部分を独立したソフトウェアとして扱い、アプリケーションから切り離します。

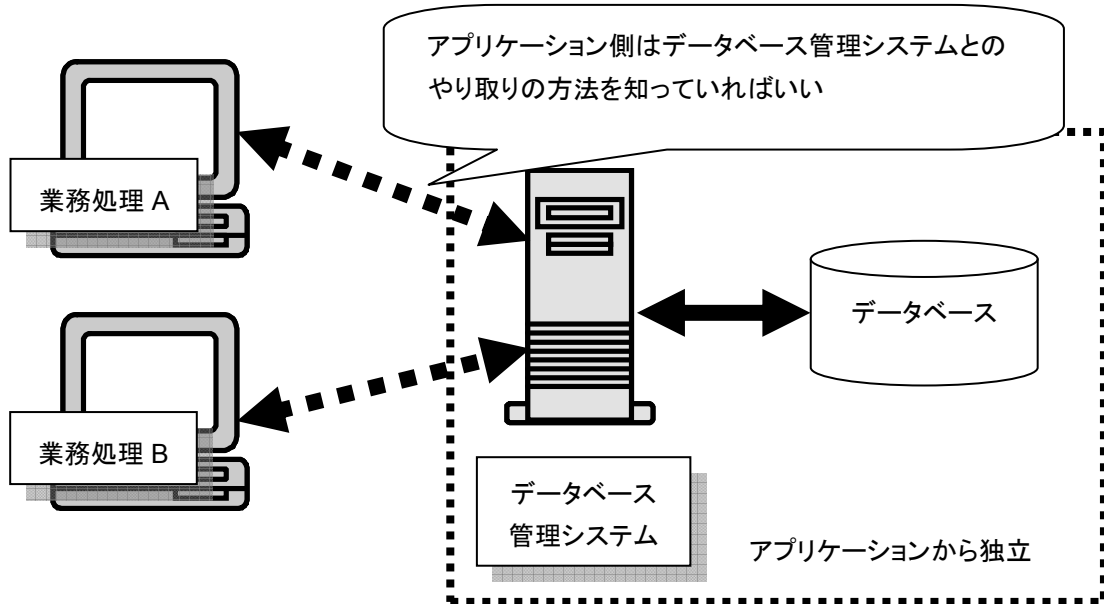
この際、データベース管理システムに管理されるデータの集まりのことをデータベースと呼びます。

**【例】ファイルを用いている場合**



プログラムは、ファイルの場所、何バイト目かなどの情報が必要となります。  
また、ファイルシステムは、複数のプログラムが 1 つのデータを同時にアクセスする  
しくみを持っていません。

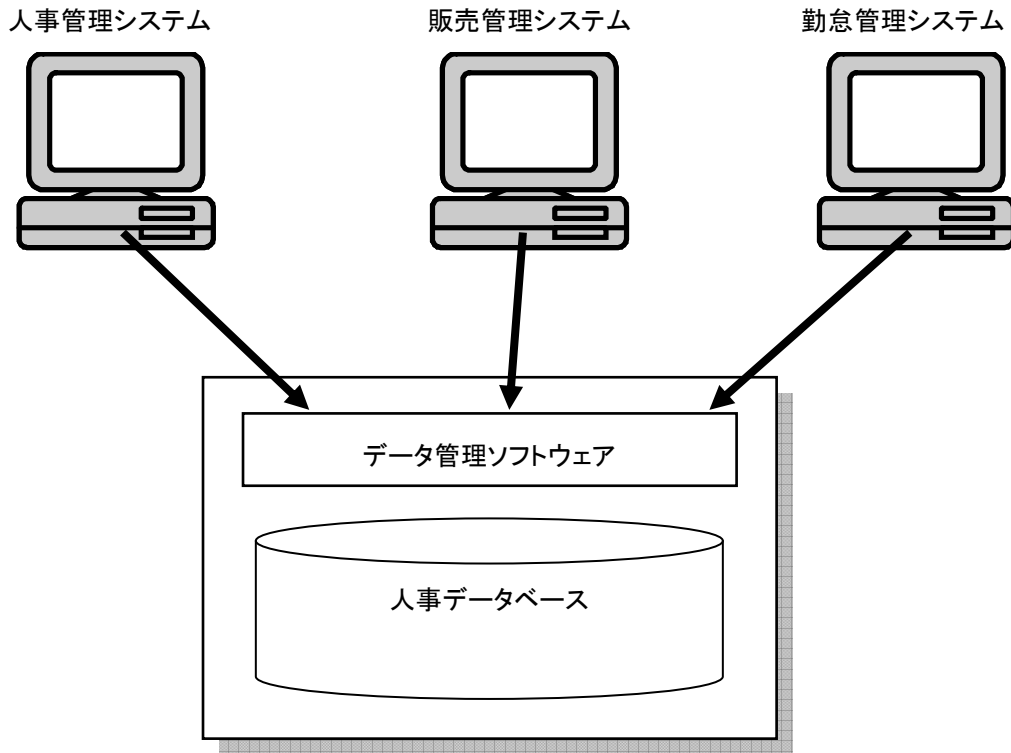
**【例】データベースを用いる場合**



データの格納場所は、データベース管理システムで管理されます。  
また、複数のプログラムが同時に 1 つのデータを利用することも可能です。

## ■ データの一元管理

データベースでは、特定のアプリケーションに依存することなく、さまざまなシステムで共有できるデータを1つの場所で管理できます。これをデータの一元管理と呼びます。



これにより、データの格納領域を効率的に使用できると同時に、データの更新漏れなどを防ぐこともできます。

## データベースの種類

- 階層型データベース
- ネットワーク型データベース
- リレーショナル型データベース

データベースを作成するには、利用したいデータをモデル化して、コンピュータ上であつかえる形にしなければいけません。

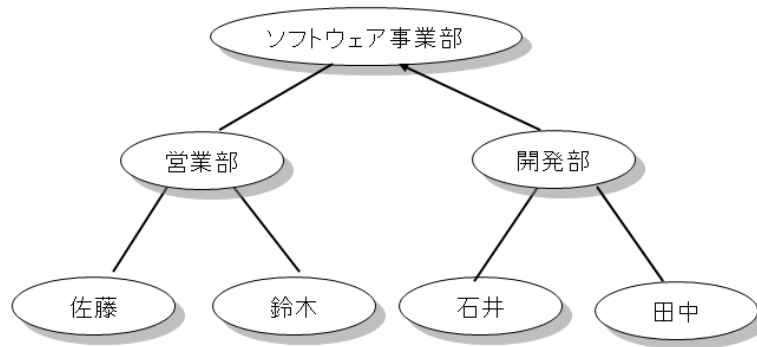
このように、**データをコンピュータ上であつかうことができるようにモデル化したものを「データモデル」と呼びます。**このデータモデルをもとにデータ構造が決定され、データベースが作成されます。データベースはもととなるデータモデルにより、以下の通りに分類されます。

- 階層型データベース
- ネットワーク型データベース
- リレーショナル型データベース



## 階層型データモデル

### 1: 多くの「親子関係」を用いてデータ構造を表現



#### ■ 階層型データモデル

「階層型データモデル」とは、データ間の関係を「親子関係」を用いて表現するものです。親子関係とは、一つの親データ(上位データ)に対して、複数の子データ(下位データ)が存在するような関係のことです。

階層型データモデルで構成されたデータベースは、検索経路(アクセスパス)が階層構造と一致するものであれば、高速に検索をおこなえます。しかし、階層構造と一致しないものであれば、検索処理が複雑になってしまいます。さらに、階層構造を満たすデータ構造しかあつかえないという問題があります。

なお、階層型データモデルを採用したデータベースを「階層型データベース」と呼びます。

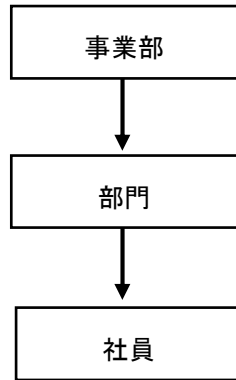
#### 階層型データモデルの特徴

- 親子関係を満たすデータをあつかうことができる
- アクセスパスが固定になる定型処理などでは処理効率がいい
- アクセスパスが固定できない非定型処理ではデータアクセスが複雑になる

### 【参考】バックマン線図

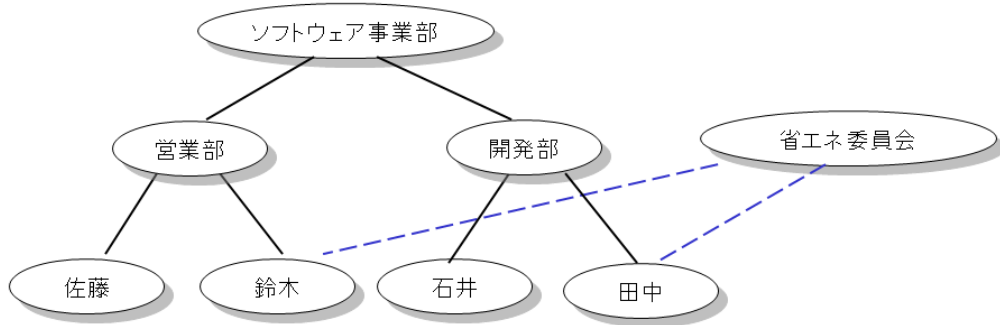
先の階層構造図は、データ構造と実際のデータを記述していました。データ構造のみを表現するには「バックマン線図」と呼ばれる図を用います。

バックマン線図では、実際のデータではなく、データの持つ属性をまとめた「レコード型」を用いてデータ構造をあらわします。



## ネットワーク型データモデル

多:多の「親子関係」を用いてデータ構造を表現



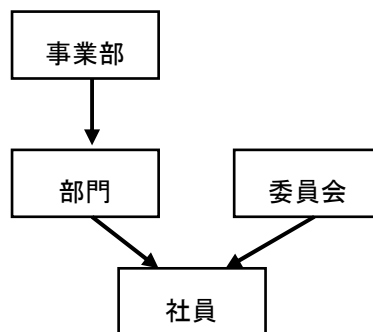
### ■ ネットワーク型データモデル

「ネットワーク型データモデル」とは、親子関係の構造に加えて一つの子データに複数の親データをもつことを許したものです。

ネットワーク型データモデルを採用したデータベースはネットワーク型データベースと呼ばれますが、ネットワーク型データモデルは CODASYL (COference on DAta SYstem Language: データシステムズ言語協会) 提案の仕様に従っていることから「CODASYL 型データベース」と呼ばれることもあります。

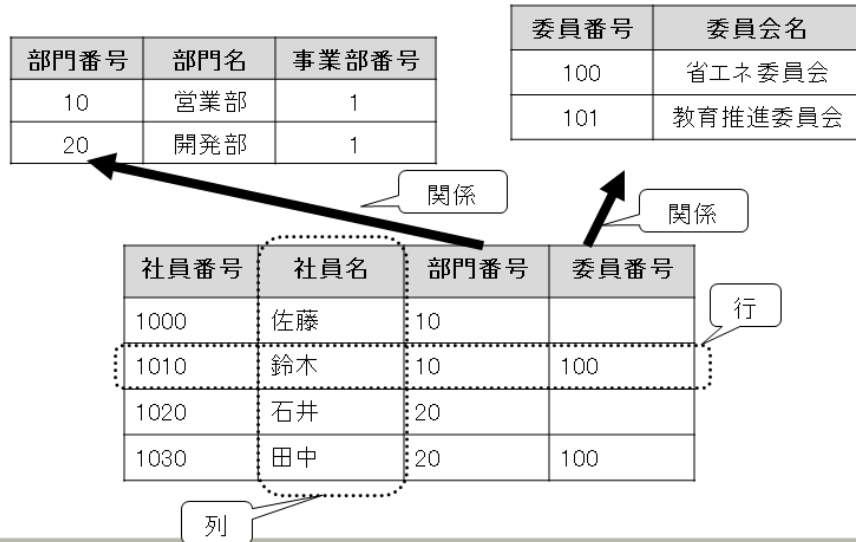
### ネットワーク型データモデルの特徴

- ネットワーク型のデータ構造を用いることにより、階層型データモデルと比べ、より多様なデータアクセスが可能である
- 構造が複雑になりやすい
- アクセスパスが固定になる定型処理などでは処理効率がいい



## リレーショナル型データモデル

2次元の表を用いてデータ構造を表現



### ■ リレーショナル型データモデル

「リレーショナル型データモデル」とは、関係代数をもとに考案されたデータモデルであり、**2次元の表を用いてデータ構造をあらわし、関連するデータを表単位にまとめます。**

他のデータモデルと比べて、構造が単純であつかいやすく、現在もっとも広範囲に用いられているデータモデルです。

また、リレーショナル型データモデルを採用したデータベースを「リレーショナルデータベース(RDB: Relational DataBase)」または「関係型データベース」と呼びます。

さらに、リレーショナル型データベースを拡張して、オブジェクト指向の概念を取り込んだ、「オブジェクトリレーショナルデータベース」なども登場しています。(巻末「付録1」参照)

列はデータの項目にあたります。列には列名やデータ型などが定義されています。また、行には1件分のデータが格納されています。表からデータを取り出すときや、データを更新するときなどは、各行に対して操作をおこないます。

### リレーショナル型データモデルの特徴

- 表を用いてデータ構造をあらわすため、単純であつかいやすい
- アクセスパスが固定されないため、非定型処理にも適している