

VI データベース

学習目標

コンピュータ利用目的の一つに膨大なデータ整理や処理加工がある。この処理ソフトとして、表計算やデータベースがある。これらのソフトの特徴を知り、それを生かし業務処理に効果的な利用ができることを目標とする。また、データベースの操作法に偏らず、データベース設計の観点も重視する。具体的には以下の項目について学ぶ。

内容のあらまし

項 目	内 容
1. データの扱い	データとは データの表現法
2. データベースとは	データベースの概念モデル データベースと表計算の違い データベースとファイルとの違い データベース構築の意義
3. データベースの概要	概要 データベースでできること データベースの表示法 データベースの分類
4. リレーショナルデータベース	特徴 リレーショナルデータベースの条件 関係代数
5. データベースの設計	データ正規化の概念 データ正規化の必要性 データ正規化の例
6. データベース管理システム	概要 データベース定義機能 データベース操作機能 データベース管理システムの位置づけ SQL 言語
7. リレーショナルデータベース操作	データベース定義 データベース操作

VI. 1 データの扱い

VI. 1. 1 データとは

(1) 身近なデータとして

個人を識別するために名前というデータを使用する。もし同姓同名者がいた場合、生年月日などのデータを加え識別している。書店や図書館で本を探す場合、書名のデータだけでは識別できない。このため著者名、出版社名、などのデータも使う。このように、一つの本を識別するために、複数データを組合せることもある。ほかにも本を識別するために、定価、本のサイズ、表紙の違いなどを識別のデータとしている。書店で本を扱う場合、同種の本を一冊ずつ区別すると膨大なデータ量になる。このため冊数という数量データを追加している。この数量データも、仕入れ数量、売上げ数量、返品数量、当日売上げ数量などがあり、本を識別する目的以外に、管理の面から必要なデータとして区分することもある。

ものを識別したり管理するため、業務上あるいはコンピュータ処理上でデータを使う場合、より少ないデータで個々の本を識別できればより良い。ISBN というコード化した 1 つのデータで本の識別ができるように、新たにコード化などによりデータを創造するケースもある。

(2) ものとの関係

ものとデータとの間には、データでものを識別したり、ものを管理したり、ものの本質を表す、などの関係がある。このために、前記したように一つ又は複数のデータを組み合わせる表現する。

ものの識別や管理用にデータを使うことから、ものに対する重要度が薄れると、識別や管理対象から外れ、データがものの本質を示す意味をなさなくなる場合もある。即ち重要でないものを示すデータは、データとしての意味がなかったり、死んだデータとなる。ものには、それぞれ異なった重要性やライフサイクルがあるように、これを反映させるデータにも、重要性やライフサイクルがある。

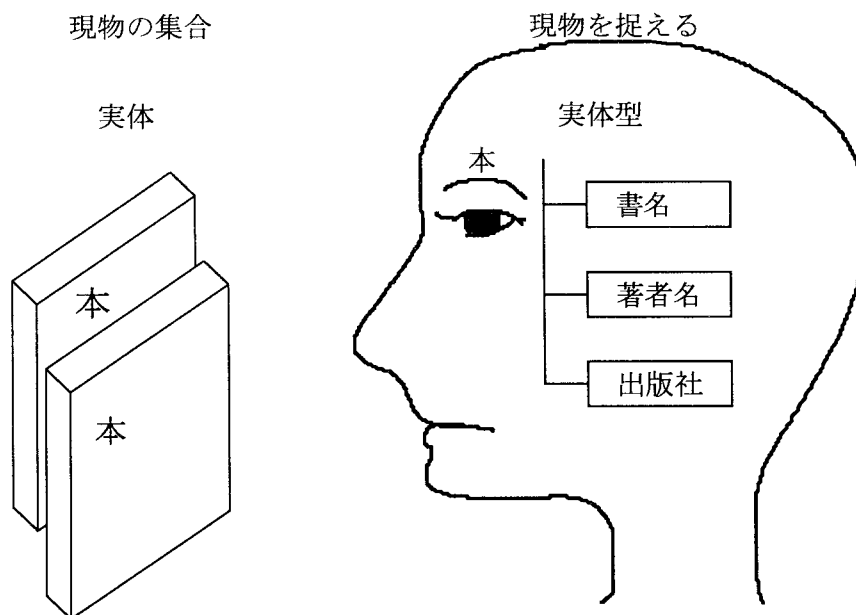
(3) データとももの分離

図書館の窓口では本の問い合わせや貸出に対し、書架にある本そのものを見て処理しているわけではない。また、工場や商社などの部品管理は、部品の大きさや管理と格納場所の違いから、部品そのものを使っての管理はできない。このように、ものを扱うために現物で処理することは難しい。そこで、ものの本質を示すデータを使いものを想定し、ものから離れてデータで扱うことになる。

ここでの留意点として、データのバックには常にものがあるとして扱う必要がある。書架に無い本を扱ったり、銀行預金の裏付けがない小切手を切ってはならない。また、工場倉庫の在庫管理において、データだけを扱う帳簿残数と現物を扱う倉庫の実残数とが不一致になると生産に支障をきたす。このようにデータがものの本質を示すことから遊離してはならない。則ちデータで離れたものを扱うが、データのみが一人歩きをしてはならない。

(4) 実体と実体型

実際に存在するものを実体と言い、実体を頭の中の想像で捉えるために、実体の中より1つの型として纏められるものを実体型とする。具体的には、個人は実体であり、個々の人物を人という実体型で捉える。この時、実体型の表現として、人の名、生年月日を使うが、これらを実体型の属性と言う。



図VI-1 実体と実体型

(5) 実体がないものと実体型

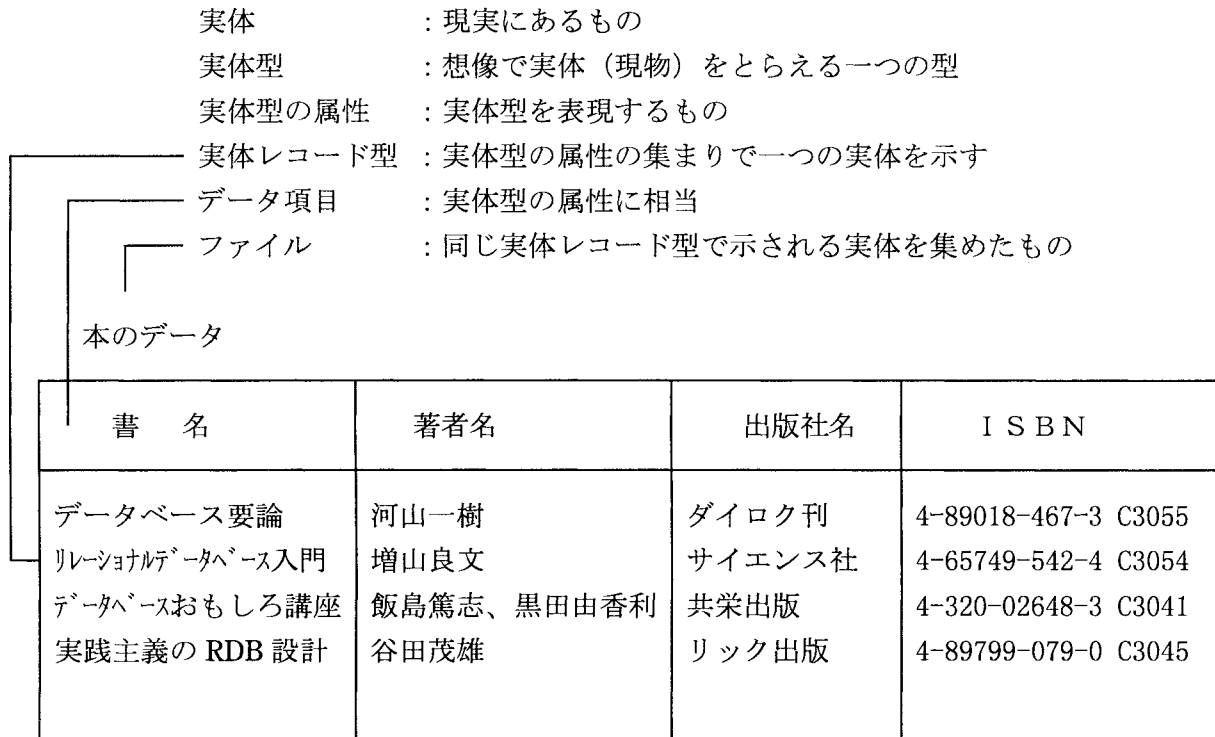
現実に実体としてのものはないが、人の意識の上で共通にイメージできるものも実体型として扱うことがある。

書店の決算日や棚卸し時期などは、ものとして存在しない。しかし、関係者間で共通にデータをイメージし、実体型として扱い業務処理上で使われる。また、学校などでなくてはならない学期や履修科目なども、学校毎に共通の認識としてこれらを決め、実体型として扱っている。このように、ものとしての実体はないが日常生活や業務処理上、必要な決めごとなども実体型として扱う。

VI. 1. 2 データの表現法

(1) レコードとファイル

実体を認識するために実体型として捉えこれらを整理格納するために、レコードとファイルの考え方がある。一冊の本を捉えるために、関連する実体型の属性を集めて実体レコード型として、レコードを定義する。また、複数ある本を整理するために、複数の実体レコード型を集め、これをファイルとする。



図VI-2 実体とファイル

(2) 実体型間の関連

実体型を表現する属性は名を変えて種々の属性で表すことがある。このことから、実体型を基として作成するファイル間に関連が生じる。データベースではこの関連を重視する。

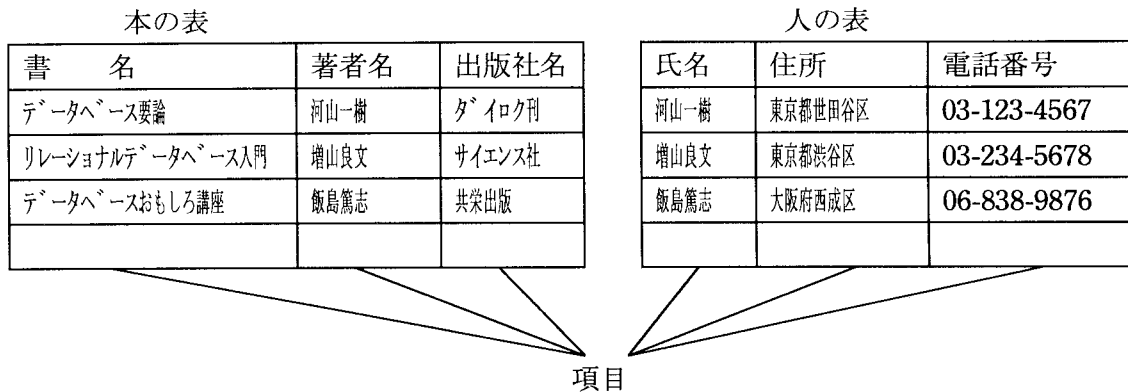
(例：住民票の氏名、本の著者名、学校の教職員名や学生名、など)

住民票ファイル：登録番号、氏名、住所
 蔵書ファイル：書名、著者名、出版社名
 在籍ファイル：学生番号、学生名、住所、電話番号
 教職員ファイル：番号、教職員名、住所、電話番号

VI. 2 データベースとは

VI. 2. 1 データベースの概念モデル

データベースとは、ある目的のために整理されたデータの集まりをいう。身近な例としての図書館のデータベースは、本を整理するための表、利用者の情報を整理する表、貸し出しを管理する表などを作成し、その表類を一纏まりにしたもので、この表内の関連するデータを更新、並べ替え、何通りかの検索法など、処理も一括してできる形式にしたものである。



図VI-3 データベースの概念モデル

- ・ファイルは、行と列からなる表の形式をとる。
- ・各行は、1件ずつのデータを格納する。
- ・行を構成する各列は、項目（データ項目）と呼ばれ、データに関連する各項目値を格納する。

VI. 2. 2 データベースと表計算の違い

- ・基本的な違い。

データを表形式で格納する点については、データベースも表計算も同じだが、両者の最大の違いは、データを操作する道具にある。机上でデータを整理する場合は、鉛筆や消しゴム、電卓などでデータを直接操作する。表計算ソフトは、この操作感覚をコンピュータの画面上で再現しており、利用者にとって直感的で理解しやすいものとなる。

しかし、膨大な量のデータを扱う場合、表データを直接操作することは難しくなる。データを項目／レコード（行）／表の構造をとり、外部記憶に格納すれば、より膨大な量のデータを扱える。同時に、簡単な命令で、この構造の単位を一括操作することができれば、より効率がよい。このデータ構造を持ち、操作命令をデータベース言語で指示し、具体的なデータ操作はコンピュータが行うのがデータベースである。

- ・具体的な構成及び処理機能のとしての違い。

表計算は表をもとに行と列の交点をセルとして、ここにデータや計算式及びマクロ命令などを入力格納し、各セルの縦列や横行の集計・編集を行う。また、セルデータの値や集計結果をグラフ化したり、印刷できる機能を持つ。表形式で表されている帳簿等は、そのものをコンピュータ画面の表形式に表示でき、手作業で行う集計を画面上で自動的に行える。表形式で表すことができるデータ類は数値等の集計、グラフ化に適しており、表計算はこれらの機能も持つ。

データベースは、データの集計やデータ単位の処理よりも、関係するデータの集まりを集合で演算する機能をもつ。項目／レコード／表とするデータ構造を確立し、これらの単位の格納／編集／整理／検索等に適している。また、必要なデータを項目／レコード単位に抽出して、表形式やカード形式でファイルとして表示したり、印刷や更新処理機能を持つ。一つのレコードをファイルに保存する場合、一つのファイルだけでなく二つ以上のファイルにも格納できる。

VI. 2. 3 データベースとファイルとの違い

ファイルは、応用プログラムでその構造を定義し、プログラムに依存した状態で使用される。データベースは、応用プログラムから独立しており、“データの独立性”を保つことができる。則ち、データ項目の追加、削除等の変更に対し、ファイルの場合、関連する応用プログラム全てを修正しなければ処理エラーとなる。データベースの場合これらの変更に対し、その項目を使用しない応用プログラムは、独立しており修正の必要はない。

VI. 2. 4 データベース構築の意義

- ・データを共同で利用できる。

データベースは、共同利用を目的に集められたデータの集合体であり、多くの応用プログラムや、ユーザからアクセスできるようになっている。

- ・データを一元管理できる。

データを一カ所だけに保存して、データの重複や冗長性を省き、データの操作、表示は種々の角度から行われる。異なったファイルを操作しているように見えるが、データそのものは一カ所にあり管理され、整合性がとれている。データベース内のアクセスは、どのデータに対しても同じ操作法でできる。

- ・データの独立性を保つことができる。

データベースは最初にデータの格納構造を決め、データを登録／更新し、はじめにデータありきとする。その後、応用プログラムへ必要なデータ項目レコードを切り出し、データ操作をさせる。このことから、応用プログラムの開発や修正が自由に行えると共に、データベース本体も独立して修正ができる。

VI. 3 データベースの概要

VI. 3. 1 概要

データベースは表／行／項目というデータ構造が確立している。各項目の関連するデータの集まりを一レコードとし、その一レコードを表の一行に相当させる。そして、そのレコードの集まりを表としている。構造を定義する場合、項目の定義で行の構成が定まり、同時に表の構造も定まる。このため、データベースを定義するには、行を構成するデータ項目をデータ構造定義法で作成し、これをデータベース構造とする。

データ操作はレコード単位に行われ、登録／更新／並べ替え／任意のレコード検索／該当レコード群を選択した一括処理などができる。これらの操作は登録／更新／検索を行うデータ操作定義法により定義する。

上記のレコード単位の操作をおこなうために、一レコードを特定できる基準が必要となる。この基準項目として、レコード内の一つのデータを使う。これがキー項目と呼ばれ、複数のデータをキー項目にする場合もある。例えば、書名と著者名と出版社名の三つのデータをキー項目として一つのレコードを特定できる。また、本毎に異なった番号というデータを付加し、それをキー項目とすると、その番号だけでレコードを特定することもある。

VI. 3. 2 データベースで出来ること

作成されたデータベースの代表的なデータ操作として、以下に示すデータの並べ替え／検索／選択がある。

・並べ替えは、指定されたデータ項目（キー項目）とある基準とで、レコード単位に並べ替える機能である。但し、並べ替えは表示上だけであり、データファイルそのものを並べ替えているわけではなく、基準をデータとして持っている。レコードを追加／削除する場合、データファイルそのものと、並べ替えの基準データも更新する。このデータの操作時に使用するキー項目の基準として以下のものがある。

文字型データ

辞書順（あいうえお順）、辞書逆順、文字符号順（J I Sコード順）

数値型データ

昇順、降順

・検索は、指定するデータ項目とデータに、完全に一致するデータを持つレコードがファイル内にある場合、そのレコードを特定する機能である。

指定するデータがレコードのキー項目であれば、検索を早めることができる。また、このキー項目データと、そのデータのレコードの格納場所を指定したアドレスデータをテーブル化した小ファイルを使うと、検索速度をレコードの格納位置によらず、均一に高速におこなえる。この小ファイルをインデックスファイルと呼ぶ。データベースはこのインデックスファイルを最大限に利用し検索処理を高速に使いやすくしたものである。

・選択は、指定したデータ項目／条件／データに、該当するデータを含むレコードを抜き出す機能である。則ち、表内のデータから選択データ項目の値が指定の条件を満たすレコードを抜き出す。また、選択の結果に対し更に別の選択条件を指定し、それを満たすレコードを選択する方法である絞り込みもできる。選択においても、表そのものを変化させるのではなく表示するデータを選択しているだけである、選択を解除すれば元の表の表示に戻る。

VI. 3. 3 データベースの表示法

データベースの表示法は一覧表形式と、カード型形式がある。

- ・カード形式は、1 件のレコードを構成する各データ項目を、一つの画面や用紙にカード形式で表示する。1 件のレコード単位の表示であることからレコード毎に処理する追加や確認しながらの削除に用いられる。また、データ項目数が多いとか、データ項目長が長い場合、一目でデータを確認できるメリットがある。しかし、表内の複数レコードを一括して処理する場合は不向きである。

- ・表形式は、一レコードを表の一行とし、データ項目を列として、一覧表の形式で一つの画面や用紙に表示する。

複数レコードを一括して表示／削除／データ更新する場合に効果的であるが、データ項目数が多いケースや、項目長が長い場合、画面のスクロールが頻繁となり操作性が悪い。

VI. 3. 4 データベースの分類

- ・データモデルの表現法による分類として、階層モデル、ネットワークモデル、リレーショナルがある。

- ・リレーショナルモデルは、表形式の行と列の関係でデータを管理していることから、多くのパソコン上のデータベースがリレーショナルデータベースと称している。中には、リレーショナルと自称していても、データの操作機能などがかなり貧弱なものとか、次節で示すリレーショナルデータベースの条件を満たさないものもある。

- ・パソコンで動作するデータベースの分類として、データベース内部の処理法の違いによるものがある。単一の表のみをカード形式や表形式で処理し表示するカード形式と、表を関連付けて処理するリレーショナル形式とに分かれる。

このカード形式は、日常の事務処理で用いる伝票の形式、名刺やカード等の一枚の形に合致した形式を、データの集まりのレコードとしてファイルに格納したものである。カード一枚が一つのレコードに相当し、カード整理などの形式に合致し、検索／並べ替え等に有用性がある。

しかし、カードの種類毎にファイルが別々に作られ、複数のファイル間の処理は考慮されていない設計のためにできない。また、複雑なデータ操作を記述するプログラミング言語もサポートされていない。

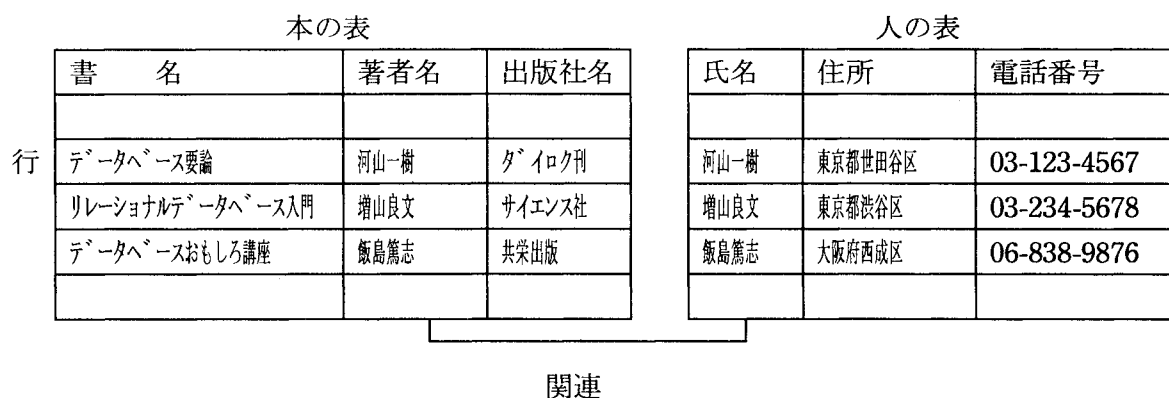
VI. 4 リレーショナルデータベース

VI. 4. 1 特徴

カード型と同じように一枚のデータの集まりをレコードとし、その集まりをファイルとする。その利用・表示において、表形式でもカード形式にもできる形式をとる。

関係づけられたデータ項目の集まりを表形式にしたファイルを、一つのテーブルとする。そのテーブルの集まりをスキーマ（データベース）として管理する。このファイル管理法によるファイルをリレーショナルデータベースと呼ぶ。本来のリレーショナルデータベースは、数学の集合概念を基礎に関係づけられており、以下に示すリレーショナルデータベースの条件が則るものとされている。

この表形式ファイルを複数個集めたデータベースの中は、表同士、それぞれ設定したキー項目で下図に示すように関連を持っている。



図VI-4 表間の関連

VI. 4. 2 リレーショナルデータベースの条件

1970年にE.F.Coddが提案した「データの関係モデル」であること。

1980年に同じくCoddが示した12項目の規則に則ること、それぞれの規則は以下のように分類される。

- ・基礎的規則
- ・構造的規則
- ・一貫性規則
- ・データ操作規則
- ・データ独立性
- ・論理的データ独立性
- ・分散独立性

VI. 4. 3 関係代数

リレーショナルデータベースは、表／行／項目を集合として操作するため、論理演算（関係演算）、集合演算が行なえる必要がある。関係演算には、選択、射影、結合の3種があり、選択と射影は、一つの表から新しい表を作り出す演算で、結合は、二つ以上の表から新しい表を作り出す。但し、これらの作り出される新しい表は、物理的に作り出されたものではなく論理的に作られたものであり、実際の表ではなくビュー表と呼ばれる。

● 選択演算例

データベース内の表（テーブル）から、指定した条件に合うレコード（行）を取り出し、新しい表を作成する。

左表より著者住所が大阪府を選択した結果、右の表となる。

著者番号	著者名	著者住所	著者電話
D0021	河島一樹	東京都世田谷区	03-123-4567
A0121	有馬泰央	大阪府北区	06-383-0946
S0111	増田良文	東京都渋谷区	03-234-5678
K2122	飯山篤志	大阪府西成区	06-838-9876

著者番号	著者名	著者住所	著者電話
A0121	有馬泰央	大阪府北区	06-383-0946
K2122	飯山篤志	大阪府西成区	06-838-9876

図VI-5 選択演算例（著者表）

● 射影演算例

表の中から、指定した複数の項目（列）を抜き出し、新しい表を作成する。

このことで、項目を選別してユーザに提供するため、データの機密や保護が可能となる。

左表より著者の番号、氏名、電話番号を射影した結果、右の表となる。

著者番号	著者名	著者住所	著者電話
D0021	河島一樹	東京都世田谷区	03-123-4567
A0121	有馬泰央	大阪府北区	06-383-0946
S0111	増田良文	東京都渋谷区	03-234-5678
K2122	飯山篤志	大阪府西成区	06-838-9876

著者番号	著者名	著者電話
D0021	河島一樹	03-123-4567
A0121	有馬泰央	06-383-0946
S0111	増田良文	03-234-5678
K2122	飯山篤志	06-838-9876

図VI-6 射影演算例（著者表）

● 結合演算例

複数の表（テーブル）を共通項目で結合して、新しい表を作成する。

この時、各表より任意の項目の抜き出し（射影演算）と、レコードの選択（選択演算）を組み合わせて結合することもできる。

書名	著者番号	出版社番号
データベース要論	D0021	S0001
データベース要論	A0121	S0003
リレーショナルデータベース入門	S0111	S0002
データベースおもしろ講座	K2122	S0003

(a) 図書登録表

出版社番号	出版社名	出版社住所	出版社電話
S0001	ダイロク刊	大阪府北区	
S0002	ソフト社	東京都渋谷区	
S0003	サイエンス社	大阪府西成区	
K0020	共栄出版	東京都小平市	

(b) 出版社表

書名	著者番号	出版社番号	出版社名	出版社住所	出版社電話
データベース要論	D0021	S0001	ダイロク刊	大阪府北区	
データベース要論	A0121	S0003	ソフト社	東京都渋谷区	
リレーショナルデータベース入門	S0111	S0002	サイエンス社	大阪府西成区	
データベースおもしろ講座	K2122	S0003	共栄出版	東京都小平市	

(c) 結合結果の表

図VI-7 結合演算例（図書登録表と出版社表を結合）

VI. 5 データベースの設計

VI. 5. 1 データ正規化の概念

リレーショナルデータベースを設計する場合、関連するデータ項目を集め括りとするレコードを決め、データ項目の中からキー項目を決定する。この時、数多いデータ項目をどのような基準で括るかの方法論が必要となる、この方法論の一つがデータ正規化である。

リレーショナルデータベースは、データ項目の関連を重視しており、設計は、正規化等の方法論でデータの括りを整理することから始まる。設計の対象となった入力及び出力データを篩にかけ、データの括りを作成すると、幾つかのレコード形式に纏められる。このレコード形式に則るレコードの集まりが表となる。データの正規化は、このデータの括りを如何にして行うかの手順と、異なるレコード形式間の関係を明確にしていく方法論である。

VI. 5. 2 データ正規化の必要性

事例

● 正規化されてないファイル

社員ファイルのデータ項目として、社員番号、氏名、部屋番号、内線電話番号がある。

このデータ項目の特徴として

- 1) 社員番号が決まると部屋番号は決まる
- 2) 部屋番号が決まると内線電話番号は決まる

がある、この条件下では以下の好ましく無い性質が発生する。

● 好ましく無い性質

1) 冗長性

同じ部屋の社員毎に、同じ内線電話番号を繰り返し格納している。

2) 更新不整合性

ある部屋の内線電話番号が変更されると、同じ部屋の全ての社員の変更が必要となり、不整合が生じる可能性がある。

3) 挿入不整合性

社員がいない部屋の内線電話番号は、社員番号がないため追加ができない。

4) 削除不整合性

ある部屋を唯一使用していた社員情報を削除すると、その部屋の内線電話番号も消去されるため、不整合が生じる。

● 解決策

社員ファイル（社員番号、氏名、部屋番号）

部屋ファイル（部屋番号、内線電話番号）

の2つのファイルを持てばよい。このことはシステム開発をした経験からくるものも多いが、データ正規化を行うことで、未経験者もベテランと同じようにファイル設計ができる。

VI. 5. 3 データ正規化の例

データベース内の表の構成は、その意味を壊さない限りできるだけ簡潔に保つのがよいとされている。正規化は、非正規形の表を第1正規化から始めて第5正規化まで行い第5正規形をつくり、簡潔に保つ表を設計できる。が、一般的に、正規化の作業は、第3正規化まで行えば十分とされている。

● 非正規形の表

書名	著者名	著者住所	著者電話	出版社名	出版社住所	出版社電話
データベース要論	河島一樹	東京都世田谷区	03-123-4567	タノイロク刊	大阪府北区	
データベース要論	有馬泰央	大阪府北区	06-383-0946	ソフト社	東京都渋谷区	
リレーショナルデータベース入門	増田良文	東京都渋谷区	03-234-5678	サイエンス社	大阪府西成区	
データベースおもしろ講座	飯山篤志	大阪府西成区	06-838-9876	共栄出版	東京都小平市	

図VI-8 非正規形の図書登録表

● 正規化された表

書名	著者番号	出版社番号
データベース要論	D0021	S0001
データベース要論	A0121	S0003
リレーショナルデータベース入門	S0111	S0002
データベースおもしろ講座	K2122	S0003

(a) 正規化後の図書登録表

著者番号	著者名	著者住所	著者電話
D0021	河島一樹	東京都世田谷区	03-123-4567
A0121	有馬泰央	大阪府北区	06-383-0946
S0111	増田良文	東京都渋谷区	03-234-5678
K2122	飯山篤志	大阪府西成区	06-838-9876

(b) 正規化後の著者表

出版社番号	出版社名	出版社住所	出版社電話
S0001	タノイロク刊	大阪府北区	
S0002	ソフト社	東京都渋谷区	
S0003	サイエンス社	大阪府西成区	
K0020	共栄出版	東京都小平市	

(c) 正規化後の出版社表

図VI-9 正規化された3つの表

VI. 6 データベース管理システム

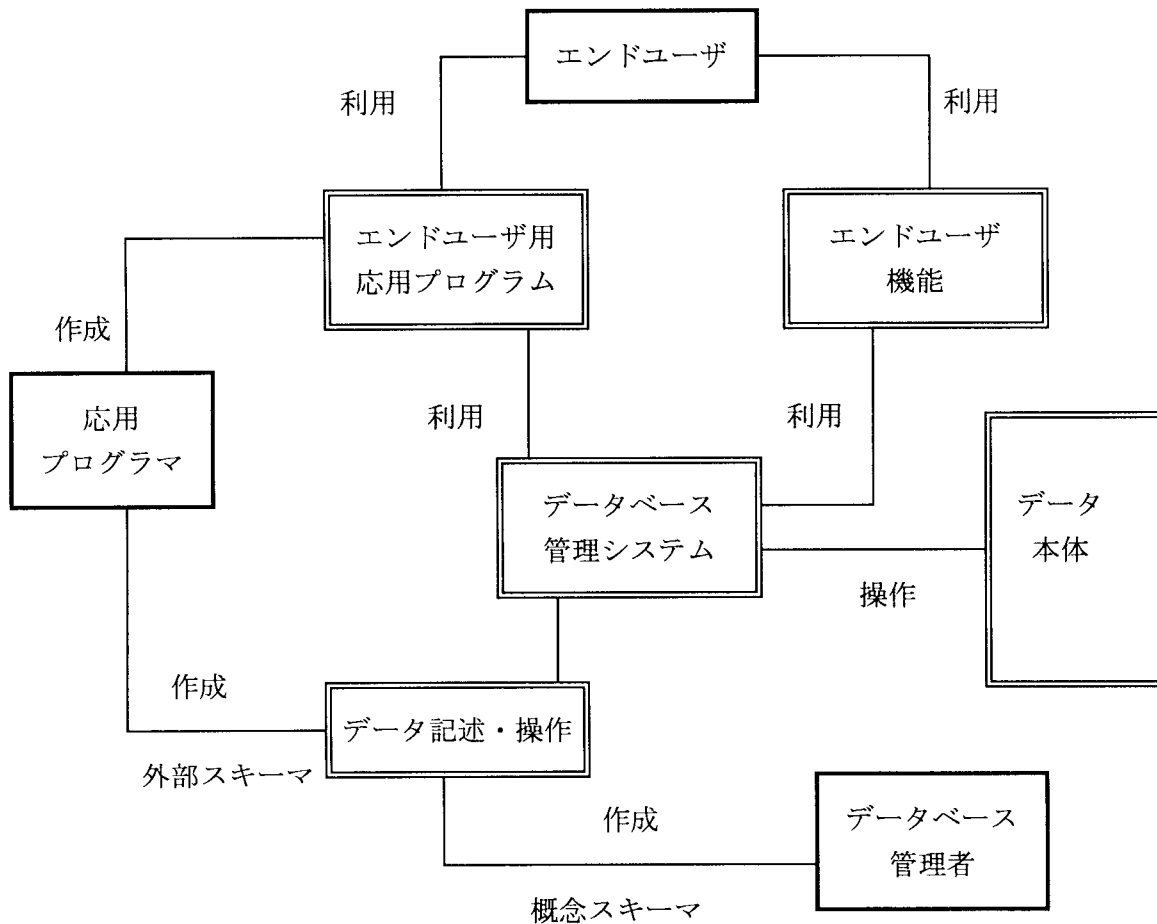
VI. 6. 1 概要

データベースには、データの独立、データの冗長性の排除、複数プログラムからの同時処理機能を要求される。これらの機能を実際に提供して、データを集中管理すると同時に、利用者へより良い操作性を提供するのがデータベース管理システムである。

エンドユーザ : 最終的にプログラムを使う人で、コンピュータメーカーから見た電算機を導入した会社をユーザ（その会社の電算機部門を指すこともある）とした考えからきている。

応用プログラマ : エンドユーザから依頼された処理を、データベース管理機能を利用して応用プログラムを作成する人。

データベース管理者 : データベース本体（概念スキーマ）を作成し、応用プログラマやエンドユーザに対し、データベースの一部（外部スキーマ）を切り出し与える管理者。



図VI-10 データベース管理システムの周辺

VI. 6. 2 データベース定義機能

スキーマを記述する機能で、データベース定義言語を使う。

- 概念スキーマの定義

データベースの基本構造を定義するもので、レコード定義／フィールド定義／キー定義を行う。

- 外部スキーマの定義

応用プログラムに対し、概念スキーマの部分を取り出し、応用プログラムが処理するレコードとフィールドのみを与える定義をする。

- 内部スキーマの定義

外部記憶装置に対し、データベースファイルを記憶する領域を定義する。

VI. 6. 3 データベース操作機能

作成されたデータベース内のテーブルに、データを格納／検索／更新する、等のデータベース利用者の操作を支援する。

- 応用プログラマに対して

応用プログラムにデータベース操作命令を記述することで、そのプログラムの実行によりデータベース内のデータが処理される機能を命令語として提供する。

- エンドユーザに対して

大量のデータを一括処理であるとか、頻繁に使うデータベース操作機能をユーティリティプログラムとして提供する。

VI. 6. 4 データベース管理システムの位置づけ

コンピュータの諸資源を有効に利用するためのシステムソフトウェア群であるオペレーティングシステム (OS) と、応用プログラムの間に位置する。

- データベース管理システムと OS

データベース管理システムは、OS が管理するハードウェアを利用して、効果的にデータベースにデータを格納するなどの処理をする。

- データベース管理システムと応用プログラム

応用プログラムは、データの書き込みをデータベース管理システムに指示し、その結果を受け取る。

VI. 6. 5 SQL

リレーショナルデータベース管理システムは、関係代数に基づく簡潔に保たれた表を組み合わせて、利用者から要求される情報を提供するため、SQLと呼ばれる利用者向けの言語を用意している。この記述文は、データベースを定義言語とデータベース操作言語に分類している。

VI. 7 リレーショナルデータベース操作

VI. 7. 1 データベース定義

データを格納するデータベース構造を以下の手順で定義する。

① スキーマ定義

データベース内に、同一環境下で一意に識別できる識別子を定義／削除する。

例：図書管理というスキーマ認可識別子データベースを、定義／削除する。

定義処理：

```
create schema authorization 図書管理
                                ↑ スキーマ認可識別子
```

削除処理：

```
drop schema 図書管理
           ↑スキーマ認可識別子
```

② テーブル定義（表定義）

データベース内の複数のテーブルを定義／削除する。

例：スキーマ内に、図書台帳と利用者台帳の2表を、定義／削除する。

定義処理：

```
create table 図書台帳
           ↑定義する表名
(
  fn__mstkey integer not null,
  ↑定義するデータ項目名 ↑データ形式 ↑登録データに空値を許さない指定

  fc__sname char(16) not null,
  ↑データ形式(文字列型)

  fc__id char(10) not null,
  fc__birth char(10),
  fn__salay smallint,
  ↑データ形式(整数型)

  fc__zip char(12),
  fc__addrel char(32),
  .....
);

create table 利用者台帳
(
  fn__mstkey smallint not null,
  fc__id char(10) not null,
  fc__birth char(10),
```



```

    fn__salay    integer,
    fc__zip      char (12),
    fc__addrel   char (32),
    .....
    .....
)

```

削除処理：

```

drop table 図書台帳 ;
           ↑削除する表名
drop table 利用者台帳 ;

```

③ インデックスファイル定義

テーブルごとの複数のインデックスファイルを定義／削除する。

例：図書台帳に2つ、利用者台帳に1つのインデックスファイルを定義／削除する。

定義処理：

```

create unique index 図書索引1
           ↑重複を許さない           ↑インデックスファイル名
           on 図書台帳 (fc__id);
           ↑表名           ↑インデックスキーとなるデータ項目
create           index 図書索引2
           on 図書台帳 (fc__sname);
create           index 利用者索引
           on 利用者台帳 (fc__name);

```

削除処理：

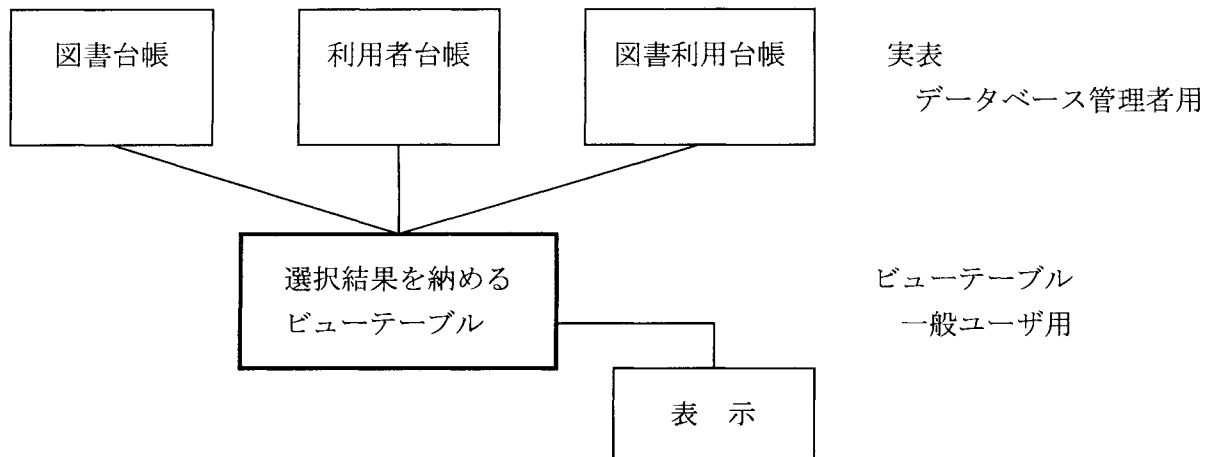
```

drop index 図書索引1 ;
           ↑削除するインデックスファイル名
drop index 図書索引2 ;

```

④ ビュー表定義

ビュー表は、実表の項目から必要なデータ項目を射影／結合／選択演算により仮想テーブルとして作成する。このビュー表を利用して実表のデータを操作する。ビュー表を利用することで、複数の表に跨る関連データ項目を同時に更新処理したり、一般ユーザのデータベース使用を制限（列単位の機密保護）する、などができる。



図VI-11 ビューテーブルと実表

例：利用台帳表を使用して利用状況ビュー表を作成する。

定義処理：

```

create view    利用状況表 (fc__id, fc__name, ...)
               ↑作成するビュー表名           ↑ビュー表のデータ項目
as
select  fc__id, fc__name, ...
        ↑基の実表のデータ項目
from    利用者台帳
        ↑基となる実表
  
```

削除処理：

```

drop viwe    利用者状況表
             ↑削除するビュー表名
  
```

VI. 7. 2 データベース操作

データベース構造が定義されたなら、実表にデータを登録格納する。その後、登録したデータに対し、検索／修正／削除などの操作がある。それぞれの操作は、以下の手順で行う。

データ登録：

```
insert into 図書台帳          → 登録する表名
      (列名1、列名2、        → 登録するデータ項目(列)名
      values (列1値、列2値、 → 登録する列値
      )
```

データ削除：

```
delete from 図書台帳        → 削除する表名を指定
      where 列名n = 列値     → 削除する行選択
      ↑ 行選択条件
```

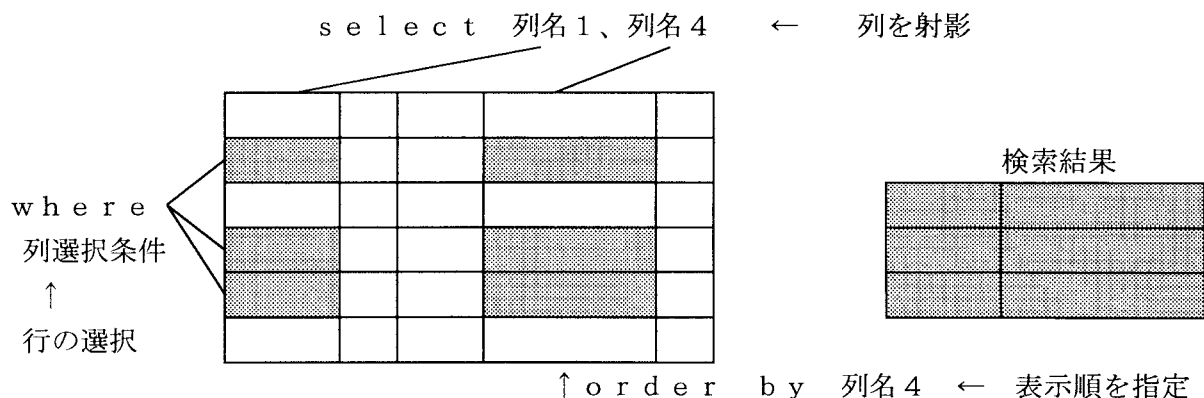
データ修正：

```
update 図書台帳            → 修正する表名
      set 列名m = 列値      → 修正するデータ項目(列)名と値
      where 行選択条件     → 修正する行選択
```

データ検索：

```
select 列名1、列名3        → データ項目1と項目3を表示
      from 図書台帳        → 検索する表名を指定
      where 列名2 >= 300   → 検索条件(データ項目2が300
      以上のレコードを選択)

      order by 列名1       → データ項目1の順番に表示
```



図VI-12 select文と実表

第VI章のまとめ

1. データは、ものを識別したり管理するために、ものそのものを使わずに想像で捉える手段として使われる。このため、データはものと分離して扱うが、バックにもものがあるとして代用しなければならない。また、実体を実体型として捉えるが、実体がないものもデータをイメージして実体型として捉えられる。ものを想像で捉える実体型を、その属性を使い整理格納するために、属性の集まりであるレコードと、レコードの集まりであるファイルを使う。
2. データベースは、ある目的のために整理されたデータの集まりである。データの集まりはデータベース以外に表計算などがあるが、目的や考え方の違いがあり、構成や処理機能が異なる。データベース構築の意義として、データの共同利用、一元管理、独立性を保つなどがある。
3. リレーショナルデータベースは、数学の集合概念を基礎にデータ項目間を関係づけた表の形式で格納され、関係代数により扱われる。このデータベースはデータ項目の関連を重視しており、設計はデータ正規化理論などでおこなう。
4. データベース管理システムは、データの独立性や冗長性の排除、同時処理を可能にする機能などをもち、データを集中管理する、また利用者に対しユーティリティプログラムを提供する。
5. リレーショナルデータベース操作は、SQL と呼ばれる利用者向けの言語を使い、データベース上の表／行／データ項目を集合として論理演算や集合演算を中心におこなう。この言語は、データベースを定義するデータベース定義言語と、データベースを操作するデータベース操作言語とに分かれる。

第VI章の主要用語

実体、実体型、実体型の属性、レコード、ファイル、実体レコード型、データ項目、実体型間の関連、概念モデル、行、列、表、表計算、セル、データの独立性、カード形式、表形式、データの一元管理、キー項目、並べ替え、検索、インデックスファイル、選択、リレーショナルモデル、リレーショナルデータベース、スキーマ、テーブル、関係代数、選択演算、射影演算、結合演算、ビュー表、データ正規化、データベース管理システム、データベース定義、データベース操作、SQL、スキーマ定義、テーブル定義、インデックスファイル定義、ビュー表定義