

ま え が き

本書は、大学、短大と高等専門学校の学生を対象として記述したものです。データベースは、今日の情報化された現代社会の膨大な情報を蓄積・管理して、利用者の要求に応じたデータを迅速に提供するシステムです。1990年代以降には、Windows という OS が日本でも一般化し、Linux という無料の OS も登場し、一般の人も個人の PC を持ち、WWW や Email が本格的に使われるようになりました。したがって、皆さん方の多くは小さいころからインターネットを通じて多くの情報を手に入れてきたと思います。

今日では、われわれは、毎日膨大な情報にさらされるようになり、“情報爆発時代”ともいわれるようになってきました。このような情報をうまく管理しようという技術がデータベースです。データベースは、データ（情報）とベース（基地）を一緒にしたものです。まさに電子化された情報を集積する基地の役割を果たすものがデータベースというわけで、情報化された現代社会においては、必要不可欠な存在になっています。

本書は、大学、短大、高等専門学校においては、週1回の半期で履修できる程度の内容になっています。また、情報処理推進機構の基本情報処理技術者試験とデータベーススペシャリスト試験の過去問題にも触れ、読者がより深く理解できるように工夫しました。また、付録にはMySQLを利用したJavaによる簡単なプログラムの構築方法について解説していますので、実習として用いるとデータベースに対する理解が深まると思います。本書がデータベースの学習への扉となれば、著者にとって望外の喜びです。

最後に、本書の出版の機会を与えていただいた株式会社コロナ社に厚くお礼申し上げます。

2011年4月

永田 武

目 次

1章 データベース

1.1 データベースの概要	1
1.2 ファイルシステムとデータベース	3
1.2.1 ファイルシステム	3
1.2.2 データベース	5
1.3 データベースの歴史	6
1.4 データベース管理システムの概要	7
演習問題	9

2章 データモデル

2.1 データモデルの概要	10
2.1.1 階層データモデル	10
2.1.2 ネットワークデータモデル	11
2.1.3 関係データモデル	11
2.1.4 オブジェクト指向データモデル	12
2.2 ANSI/SPARCの3層スキーマ構造	13
2.2.1 概念スキーマ	14
2.2.2 外部スキーマ	14
2.2.3 内部スキーマ	14
2.3 E-Rモデル	14
2.3.1 ERDの表記方法	15
演習問題	17

3章 関係代数

3.1 関係代数の概要	20
3.2 集合演算	20
3.2.1 和集合	21
3.2.2 差集合	21
3.2.3 積集合	22
3.2.4 直積集合	22
3.3 関係演算	23
3.3.1 選択演算	23
3.3.2 射影演算	24
3.3.3 結合演算	25
3.3.4 商演算	26
演習問題	27

4章 データベース設計

4.1 データベース設計の概要	30
4.2 概念データモデルのモデリング技法	32
4.2.1 E-Rモデルの表現要素	32
4.2.2 カーディナリティ	33
4.2.3 スーパータイプとサブタイプ	34
4.3 論理データモデルのモデリング技法	35
4.3.1 階層データモデル	35
4.3.2 ネットワークデータモデル	36
4.3.3 関係データモデル	36
演習問題	36

5章 リレーションの正規化

5.1 正規化の概要	39
5.2 キー	40
5.3 関数従属性	42
5.4 非正規形	45
5.5 第1正規形	46
5.6 第2正規形	48
5.7 第3正規形	50
5.8 ボイス・コッド正規形	52
演習問題	53

6章 関係データベース言語 SQL (その1)

6.1 SQLの概要	58
6.2 データ定義言語	60
6.2.1 表定義	60
6.2.2 ビュー定義	62
6.3 データ操作言語	63
6.3.1 データの参照 (SELECT文)	63
6.3.2 データの追加 (INSERT文)	68
6.3.3 データの変更 (UPDATE文)	70
6.3.4 データの削除 (DELETE文)	71
6.4 データ制御言語	72
6.4.1 権限の付与 (GRANT文)	72
6.4.2 権限の取消 (REVOKE文)	73
6.5 メタデータとリポジトリ	74
演習問題	74

7章 関係データベース言語 SQL (その2)

7.1 関係演算子	80
7.2 論理演算子	83
7.3 その他の演算子	85
7.4 集合関数	86
7.5 副問合せ	88
7.6 埋め込み型SQL	91
演習問題	92

8章 データの検索機構

8.1 磁気ディスク装置	96
8.1.1 容量計算	97
8.1.2 記憶方式	98
8.1.3 アクセス時間	99
8.2 インデックス	100
8.2.1 B+木インデックス	101
8.2.2 ハッシュインデックス	103
8.2.3 ビットマップインデックス	105
8.3 テーブルのアクセス方法	106
8.4 テーブルの結合方法	107
8.4.1 ネストループ結合	107
8.4.2 マージ結合	108
8.4.3 ハッシュ結合	109
演習問題	110

9章 トランザクション管理

9.1 トランザクション管理の概要	112
9.2 ACID 特性	114
9.3 同時実行制御	115
9.3.1 並列トランザクションに起因する問題	115
9.3.2 同時実行制御の方式	116
9.4 直列可能性	120
9.4.1 直列可能性	120
9.4.2 2相ロックングプロトコル	121
9.5 隔離性水準	122
演習問題	123

10章 障害回復

10.1 障害回復の概要	126
10.2 ロールフォワードとロールバック	127
10.3 ログファイル	128
10.4 チェックポイント	128
10.5 障害への対応	129
演習問題	130

11章 分散データベース

11.1 分散データベースの概要	133
11.2 透過性	135
11.3 テーブルの結合方法	136
11.3.1 分散入れ子ループ法	136
11.3.2 分散ソートマージ法	137

11.3.3 セミジョイン法	138
11.4 コミットメントプロトコル	139
11.4.1 2相コミットメント	139
11.4.2 3相コミットメント	140
演習問題	141

12章 応用技術と将来動向

12.1 XML データベース	145
12.2 OLAP	146
12.3 ストアドプロシージャ	147
12.4 オブジェクト指向データベース	148
12.5 レプリケーション	150
12.6 Google File System	150
12.7 BigTable	151
12.8 データベースの将来動向	152
演習問題	155
付録	157
A. MySQL のインストール	157
B. 埋め込み型 SQL のプログラミング	159
参考文献	165
演習問題解答	166
索引	178

1章 データベース

現在、私たちは数世紀前の人間が一生の間に得た何千倍、何万倍もの情報量の中で暮らしている。データベースはこのような多量な情報を効率的に管理して、利用者の要求に応じて容易に検索・抽出など再利用できるようにしたものである。

本章では、データベースの概要、ファイルシステムとデータベース、データベースの歴史、そしてデータベース管理システムの概要について述べる。

1.1 データベースの概要

現在では新幹線の切符は、インターネットを用いて簡単に購入することができるようになってきているが、それは1972年に開発されたマルス（MARS: multi access seat reservation system）までさかのぼる。マルスは、現在JR各駅の「みどりの窓口」や全国各地の旅行会社の窓口に設置された端末により、乗車券や指定券はもとよりホテルやレンタカーの予約など幅広く利用されている。そして、このマルスは、コンピュータ技術のオンラインリアルタイムシステムへの可能性を示したことが評価され、2008年に電気学会から電気技術顕彰制度の「でんきの^{いしずえ}礎」に、2009年には情報処理学会から「情報処理技術遺産」として認定されている。このような多量な情報を扱う情報処理システムの中核をなすのがデータベースである。

データベースとは、文字通りデータの基地（ベース）であるが、単なるデータの集まりではなく、さまざまな情報処理システムで利用できるように、データを保存、検索、更新、削除できるように構成されたものである。すなわち、

2 1. データベース

データベースとは、「複数の利用目的で共有できるように、相互に関連付けられた冗長のないデータの集合」である。

ここで、「データ」と「情報」の違いについて述べておく。私たちは、この両者の違いを意識しないで「データ共有」や「情報共有」などと使っているが、データと情報は区別して認識しておく必要がある。

まず、データとは「観測、測定、統計などから得られる客観的な事実を、文字、数値、図形、画像、音声など人間が知覚できるかたちで表したもの」である。例えば、学生データの学生番号や氏名は文字で表され、環境データの温度や湿度は数値で表され、パスポートの顔写真は画像で表される。これらは、客観的事実を表したデータでだれでも同じように解釈したり認識したりすることができる。

一方、情報は「ある特定の目的について、適切な判断を下したり、行動の意思決定をするために役立つデータのこと」である。

例えば、スーパーマーケットやコンビニで使用されている POS (point of sales system : 販売時点情報管理) からは、「いつ・どの商品が・どんな価格で・いくつ売れたか・購入者の年齢層・性別・当日の天気」などのデータが収集される。このデータは、売り場の責任者にとっては、売れ筋商品の補充や、在庫処分のための値引きの意思決定のための情報となる。

また、「おむつを買った人はビールを買う傾向がある」という有名な話がある。これは、「子供のいる家庭では母親はかさばる紙おむつを買うように父親に頼み、店に来た父親はついでに缶ビールを購入するので、この二つを並べて陳列したところ売り上げが上昇した」という内容で知られているものである。これは、POS から得られたデータを分析することより得られた情報である。

このように情報は、データから直接得られる場合と、データを分析することによって得られる場合がある。後者の場合は、大規模なデータの中からその中に潜んでいる有益なデータを発掘するというところで**データマイニング** (data mining) として知られている。

以上のように「データ」と「情報」には違いがあり、“利用者の状態を変化

させないデータは情報にはならない”ということに注意すべきである。

1.2 ファイルシステムとデータベース

コンピュータシステムにおいて、データベースの出現の前までは、データの集まりは、OSの提供する「ファイル」という概念で扱われていた。今日でもファイルは使用されているが、大規模なデータを扱う情報処理システムではこれから述べる理由によってデータベースが用いられている。

1.2.1 ファイルシステム

情報系の学生は、このデータベースを学ぶ以前にプログラミングの講義でファイル入出力処理を記述したことがあると思う。例えば、ファイルの入出力は、C言語ならば `fscanf` 関数や `fprint` 関数、Java言語なら `FileReader` クラスや `FileWriter` クラスなどを使用したはずである。

図 1.1 (a) は、C言語によるファイル入出力のプログラム例である。このプログラムは、図 (b) に示すように、`test.in` という入力ファイルからデータ（学生番号、氏名、数学の得点、国語の得点）を読み込み、学生数をカウントし、数学と国語の平均点を計算して `test.out` という出力ファイルにデータを書き出すものである。ここで示したように、このプログラムは入出力ファイルの構造に依存している。すなわち、データ項目などのファイルの構造に変化があった場合には、このプログラムは修正されなければならない。例えば、図 (b) の `test.in` の学生番号は 8 文字、氏名は 20 文字（半角）であるが、なんらかの理由で文字数を大きくする必要がある場合には、プログラムの複数箇所を変更する必要がある。また、英語の得点を追加したい場合にも同様にプログラムの変更が必要になる。このような修正は、この入力ファイルにアクセスするすべてのプログラムに対して実施されなければならない。

```

#include <stdio.h>
void main( int argc, char* argv[ ] ){
    FILE *fin, *fout;
    char id[8], name[20];
    int num = 0, sum_mat=0, sum_lan=0;
    int mat, lan, ave_mat, ave_lan;
    if( (fin=fopen("test.in","r") == NULL ){ printf( "Not
found (test.in)¥n" ); exit(1); }
    while( fscanf( fp,"%s %s %d %d",id,name,mat,lan ){
        num++; sum_mat+=mat; sum_lan+=lan;
        printf( "%10s%20s %3d %3d¥n", id,name,mat,lan );
    }
    close( fin );
    ave_mat = sum_mat/num; ave_lan = sum_lan/num;
    if( (fout=fopen("test.out","w") == NULL ){
        printf( "Not created (test.out)¥n" ); exit(1);
    }
    fprintf( fout,"Total number = %d", num );
    fprintf( fout,"Math. ave = %6.1f", ave_mat );
    fprintf( fout,"Lan. ave = %6.1f", ave_lan );
    close( fout );
}

```

(a) プログラム

(入力ファイル test.in)	(出力ファイル test.out)
A1000100 Aoki 85 68	Total number = 45
A1000200 Ishida 78 90	Math. ave = 72.5
A1000200 Ueki 58 84	Lan. ave = 84.8
.....	

(b) 入力ファイルと出力ファイル

図 1.1 C 言語によるファイル入出力のプログラム例

一般に、情報システムは、運用中にもデータ項目の追加は頻繁に発生するので、ファイルのみを使用する方法で大規模なシステムを開発することの限界は明らかである。また、アプリケーションごとにどのようなデータ項目にするかはアプリケーションの設計者にまかされているために、複数のファイルに同一データが散在することも予測できる。このようなことからデータベースの必要性が広く認識されるようになった。

まとめると、ファイルを用いたシステムには以下のような問題点が指摘されている。

- ① ファイルとプログラムが独立していない（ファイルの構成の変更は、利用しているすべてのアプリケーションプログラムの修正が必要である）。
- ② ファイルの内容の整合性管理が難しい（アプリケーションプログラムの誤った更新処理の検出が不可能）。
- ③ 複数のファイル間でデータ項目が重複しやすい。
- ④ 機密保護が不十分である。
- ⑤ データの障害対策が不十分である。

1.2.2 データベース

前述のファイルシステムの欠点を補うかたちで登場したデータベースには、次のような機能が要求されている。

- (1) **データの共有化** データベースの第一の目的は、企業・組織のなるべく広い範囲でデータを共有し利用することである。
- (2) **データの一元管理** データの一元管理により、データ項目などの重複が削減されデータの保守作業を容易にすることができる。
- (3) **データのプログラムからの独立** データとプログラムを独立させることにより、データ項目とプログラムの保守・変更作業がどちらも容易になる。
- (4) **データの整合性維持** 整合性制約を用いることにより、意図しない

データや間違ったデータの入力を防ぐことが可能になる。

(5) **データの障害回復** ハードウェアやソフトウェアの障害によってデータベースが障害を受けた場合に、バックアップデータやログ (log) を用いて迅速に回復させることが可能になる。

(6) **データの機密保護** アクセスするユーザを識別し、データごとにアクセスできるデータを制限することにより、機密性を保つことができる。

1.3 データベースの歴史

データベースの研究は、1950年代の米国におけるソ連との軍拡競争に端を発する。当時の米国は世界各地の軍事データを収集し、データ (data) を蓄積する基地 (base)、すなわちデータ基地 (database) と呼んでいた。

その後さまざまなデータベースの研究がなされ、1970年にはIBMのSun Jose 研究所の E. F. Codd が「大規模な共有データバンクのための関係モデル」という論文を発表した。この関係モデルという方法は、データベースの考え方に決定的な影響を与え、**リレーショナルデータベース** (RDB: relational database) として現在広く利用されることになった。

リレーショナルデータベースを管理し運用するためのソフトウェアを**リレーショナルデータベース管理システム** (RDBMS: relational database management system) と呼び、実用化に向けた製品開発が行われた。1979年には世界初の商用 RDBMS の Oracle が製品化され、1982年の SQL/DS、1984年の DB2 など RDBMS の本格的な利用が始まった。

RDBMS を利用するための言語が SQL である。初期の SQL は、IBM の Sun Jose 研究所で開発された RDBMS の System-R に実装された。当初は構造化英文問合せ言語 (SEQUEL: structured English query language) と呼ばれたが、その後構造化問合せ言語 (SQL: structured query language) と改名された。この SQL は、国際標準化機関 (ISO: International Organization for Standardization) を中心にして標準化が実施されている。1987年に SQL 規格

索引

【あ】		階層データモデル	10	グリッドコンピューティング	
アームストロングの規則	45	回転待ち時間	100		154
アクセス時間	99	概念スキーマ	14	【け】	
アトリビュート	12	概念設計	30	継承	149
		外部スキーマ	14	結合	20
【い】		格納位置	104, 105	結合演算	25
1NF	46	隔離性	114	原子性	114
位置決め時間	99	隔離性水準	122	【こ】	
1対1	33	合併律	45	高性能技術計算	154
1対多	33	カプセル化	148	後退復帰	127
一貫性	112	関係	12	コミット	113
インスタンス	148	関係演算子	80	コンピュータクラスタ	153
インターネット	1	関数従属性	42	コンフリクト	104
インデックス	31, 100	関係度数	20		
		関係データベース		【さ】	
		管理システム	12	差	20
【う】		関係データモデル	10	サーチ時間	99, 100
埋め込み型 SQL	91	完全関数従属	42	差集合演算	21
		関連	31	サブクラス	148
【え】		【き】		サブタイプ	34
エンティティ	31	キー・バリュ型データストア	152	3NF	50
エンティティタイプ	32	危険分散	134	3PC	140
		擬推移律	45	3相コミットメント	140
【お】		行	59	3層スキーマ	13
オブジェクト	148	行識別子	102		
オブジェクト識別子	148	共有ロック	116	【し】	
オブジェクト指向		【く】		シーク時間	99
データベース	148	グーグルファイルシステム	150	磁気ディスク装置	96
オブジェクト指向		クラウドコンピューティング	152, 154	時刻印	119
データモデル	10			自然結合演算	25
オブジェクトデータベース	148	クラス	148	実行計画	106
		クラス階層	148	実体	31
【か】				実体関連モデル	14
カーディナリティ	33				

実表	60	属性	12, 148	トランザクション	112
シノニム	104			ドリルダウン	147
自明な関数従属	44	【た】		【な】	
ジャーナルファイル	128	ダーティリード	115	内部スキーマ	14
射影	20	第1正規形	46	【に】	
射影演算	24	耐久性	114	2NF	48
集合関数	86	第3正規形	50	2PC	139
従サイト	139	ダイシング	147	2次キー	40
縮退フェーズ	121	代替キー	47	2相コミットメント	139
主サイト	139	第2正規形	48	2相ロックングプロトコル	121
商	20	タグ	145	二分探索	102
商演算	26	多対多	33	【ね】	
障害回復	126	ダブル	12	ネイティブXMLデータベース	
衝突	104	【ち】			145
情報	2	チェックポイント	128	ネストループ結合	107
【す】		中間ノード	101	ネットワークデータモデル	10
推移的関数従属	44	直積	20	【の】	
推移律	45	直積集合演算	22	ノンリピータブルリード	115
垂直分散	133	直列可能	120	【は】	
スーパークラス	148	直列可能性	120	ハードディスク装置	96
スーパータイプ	34	【て】		ハッシュインデックス	104
スケラブル	153	定義域	12	ハッシュ関数	104
スケールアウト	152	データ	2	ハッシュ結合	109
スケールアップ	152	データグリッド	154	ハッシング	104
ストアドプロシージャ	147	データ辞書	74	バランス木	101
スライシング	147	データディクショナリ	74	汎化	149
【せ】		データ転送時間	100	反射律	45
正規化	31, 39	データベース管理システム	7	【ひ】	
成長フェーズ	121	データベース設計	30	比較演算子	24, 25
積	20	データマイニング	2	ビットマップインデックス	105
積集合演算	22	テーブル	31, 59	否定	21
セキュア状態	139	【と】		非同期レプリケーション	150
セクタ	97	同義語	104	ビュー表	60
セミジョイン法	138	同期レプリケーション	150	表	31, 59
前進復帰	127	等結合	25	【ふ】	
選択	20	同時実行制御	115	ファイル	3
選択演算	23	導出表	60		
占有ロック	116	独立性	114		
【そ】		特化	149		
増加律	45	ドメイン	12		

ファントムリード	115	待ちグラフ	118		
フォールトトレランス	150	マルス	1	【れ】	
負荷分散	134			列	59
副問合せ	88	【め】		レプリケーション	150
物理設計	30	メソッド	148	【ろ】	
物理データ独立性	14	メタデータ	74	ロールバック	113, 127
部分関数従属	43	メッセージパッシング	148	ロールフォワード	127
ブロック	98	【ら】		ログ先書出し方式	128
ブロック化因数	98	楽観の同時実行制御	119	ログファイル	128
ブロパティ	148	【り】		ロストアップデート	115
分解律	45	リーフノード	101	ロック/アンロック	116
分散 KVS	151	リポジトリ	74	論理演算子	83
分散入れ子ループ法	136	両立性行列	116	論理積	21
分散ソートマージ法	137	リレーショナルデータベース	6	論理設計	30
分散データベース	133	リレーショナルデータベース	6	論理データ独立性	14
分散透過性	135	管理システム	6	論理和	21
【ほ】		リレーション	12	【わ】	
ボイス・コード正規形	52	リレーションシップ	15, 31, 33	和	20
【ま】		ルートノード	101	和集合演算	21
マークアップ言語	145			和両立	20
マージ結合	108				

【A】		COMMIT	113	【F】	
ACID 特性	114	COUNT 関数	88	FETCH	91
AND 演算子	83	CREATE TABLE	60	【G】	
ANSI/SPARC	13	CREATE VIEW	63	GFS	150
AVG 関数	87	CVS	74	GRANT	72
【B】		【D】		GROUP BY	66
B+木インデックス	101	DCL	7, 59, 72	【H】	
BCNF	52	DDL	7, 59, 60	HPTC	154
BETWEEN 演算子	85	DECLEAR CURSOR	91	【I】	
BigTable	151	DELETE	71	IMS	11
B 木インデックス	101	DISTINCT	64	INSERT	68
【C】		DML	7, 59, 63	INSERT ~ SELECT	69
CLOSE	92	【E】		INSERT ~ VALUES	68
CODASYL	11	E-R モデル	14, 31, 32	IN 演算子	85, 89
Codd	11	ERD	15, 32		
		EXISTS 演算子	90		

IS NULL 演算子	85				UPDATE	70
ISO	6		【P】			
					【W】	
【J】		PC クラスタ		154	WAL	128
		POS		2	WAL プロトコル	128
JIS	7		【R】		WFG	118
					WHERE	65
【M】		RDB		6		
MAX 関数	87	RDBMS		6	【X】	
MIN 関数	87	REDO		127	X lock	116
		REVOKE		73	XML	145
【N】		ROLLBACK		113	XML Enable データベース	145
NOT 演算子	84	ROWID		102, 104, 105	XML データベース	145
			【S】		XPath	146
【O】		S lock		116	XQuery	146
OLAP	146	SELECT		63	~~~~~	
OODB	148	SQL		6	θ 結合	25
OODBMS	148	SUM 関数		87		
OPEN	91		【U】			
ORDBMS	13					
ORDER BY	68	UNDO		127		
OR 演算子	83					

— 著者略歴 —

1975年 松江工業高等専門学校卒業
1978年 宮崎大学工学部電気工学科卒業
1980年 広島大学大学院博士課程前期修了（回路システム工学専攻）
1980年
～89年 株式会社 東芝勤務
1989年 松江工業高等専門学校講師
1991年 松江工業高等専門学校助教授
1995年 博士（工学）（広島大学）
1997年 広島工業大学助教授
2001年 広島工業大学教授
現在に至る

データベースの基礎

Introduction to Database

© Takeshi Nagata 2011

2011年6月17日 初版第1刷発行

★

検印省略

著者 ^{なが}永 ^た田 ^{たけし}武
発行者 株式会社 コロナ社
代表者 牛来真也
印刷所 萩原印刷株式会社

112-0011 東京都文京区千石 4-46-10

発行所 株式会社 コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替 00140-8-14844 ・ 電話 (03)3941-3131 (代)

ホームページ <http://www.coronasha.co.jp>

ISBN 978-4-339-02456-2 (吉原) (製本：愛千製本所)

Printed in Japan



本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられております。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めておりません。

落丁・乱丁本はお取替えいたします