

情報基礎シリーズ5

データベース

電子開発学園出版局

情報基礎シリーズ5

データベース

電子開発学園出版局

* 本書に掲載した会社名・製品名等は、一般にそれぞれ各社の商号・登録商標または商標です。

はじめに

コンピュータが開発されて以来、コンピュータは多くの分野で利用されてきました。現在では例えば、旅券を発行する、ご飯を炊く、ゲームをする、ニュースを知る、電話をするなど私たちの生活の中に根付いた分野で利用され、なくてはならない物として、また身近な物として存在しています。また現在のネットワーク社会はコンピュータなくして成り立ちません。

コンピュータはこれらの多くの分野で実現する様々な事柄を実現するために、“情報”を取扱い、制御し、処理しています。コンピュータで取り扱い、処理する“情報”とはいったいどのようなもので、それを取扱い、処理するためにどのような技術が使用されているのでしょうか。本書を含む「情報基礎シリーズ」では、このような知識、技術を“インフォメーションテクノロジー (IT)”と総称しております。

この“インフォメーションテクノロジー”を学ぶことは、様々な情報とそれを取り扱うコンピュータを学ぶことであり、それにより現代社会の基盤となるコンピュータについて理解を深めることとなり、そしてその社会についても知るようになります。

本書は、多くの分野から成立している“インフォメーションテクノロジー”のうち、データベース、ネットワーク・セキュリティ、システム開発など、より実際のコンピュータに密接に関連した知識を基礎から学ぶためのテキストであります。

本書によって、情報とはなにか、情報技術とはなにか、コンピュータはどのようなものを学び、理解を深め、情報処理技術者試験の取得に役立てていただければ幸いです。

編著者

目次

はじめに

第1章 データベースとDBMS	1
1.1 データベース	1
1.1.1 データベースの種類と特徴	1
1.1.2 データベースのモデル	4
1.1.3 関係モデル	8
1.2 データベース管理システム	10
1.2.1 DBMSの機能	10
1.3 データベース管理者	13
1.3.1 データベース管理者の役割	13
第2章 データベースの設計	15
2.1 データ分析とデータベース設計	15
2.1.1 データ分析	15
2.1.2 データベースの設計	17
2.1.3 正規化	26
2.1.4 データベースの物理設計	35
第3章 データ操作	37
3.1 データベースの操作	37
3.1.1 関係データベースに対する操作	37
3.2 SQLによる定義と操作	44
3.2.1 データ定義言語	44
3.2.2 データ操作言語	53
3.2.3 データ制御言語と埋め込み型SQL	67

第4章	トランザクション処理	69
4.1	データベース制御機能	69
4.1.1	データベース制御	69
4.2	排他制御	74
4.2.1	排他制御	74
4.2.2	コミットメント制御	82
4.3	障害回復	85
4.3.1	障害復旧	85
第5章	データベース応用	91
5.1	データベースの応用	91
5.1.1	データの蓄積と管理	91
5.1.2	分散データベース	94
【練習問題】	ダウンロードのご案内	97
索引		99

データベース

第1章 データベースとDBMS

現代の情報処理システムでは、さまざまなデータを集積し、検索し、出力するためにデータベースが必須となっている。データベースは効率的にデータを管理し、容易に検索でき、必要な部分を抽出できる仕組みを持っている。データベースはその仕組み、構造、動作をいくつかの理論から求めており、それがデータベースの種類となる。

また、データベースをコンピュータ上で管理するためのシステムを**データベース管理システム**（DBMS : DataBase Management System）と呼ぶ。

1. 1 データベース

1. 1. 1 データベースの種類と特徴

（1）データベースとファイル

データベースはデータの共有や共用を目的として、データを蓄積し、総合的に管理する、データ（Data）の補給基地（Base）である。

ファイルとデータベースは同じようにデータを保存するものだが、ファイルは統一的な管理やデータの分散・重複を防ぐことができず、またファイル内のデータの並び替えなどの処理も頻繁に行うことが難しいという欠点がある。さらに、アプリケーションプログラムがファイル内のデータを操作するには、ファイル内でデータがどのような形で保存されているかを理解した上で、操作する必要がある。

それに対し、データベースはデータをデータベース内で一括で処理することにより、分散や重複を防ぎ、並び替えなども容易である。さらに、データとアプリケーションプログラムを切り離して、つまりデータがどのように格納されているかを知らずとも利用できるようになっている。また、データを特定の業務とも切り離して管理できるという点もある。

（2）データベースの種類

データベースで保持するデータ1件分を**レコード**と呼ぶ。このレコードの表現方法や、レコードとレコードの間関係などの違いによって、データベースにはいくつかの種類がある。代表的なデータベースの種類としては次のものがある。

（a）関係データベース

関係データベース（RDB : Relational DataBase）は現在最も広く用いられているデータ

ベースで、後述する関係データモデルに基づいて設計されるデータベースである。

データを表に近い構造で持ち、データ間の関連を「関係」で表現する。

(b) 構造型データベース

構造型データベースは、データを親子関係で持ち、その関係で2つの構造がある。

(ア) 階層型データベース

階層型データベース (HDB : Hierarchical DataBase) はデータを親子関係で持つデータベースで、親データは複数の子データを持つことができ、子データはさらに親となり子データを持つことができる。ただし、子データは1つしか親データを持つことができない。この構造のデータベースがHDBである。

(イ) 網型データベース

網型データベース (NDB : Network DataBase) はHDB同様に、データを親子関係で持ち、親データは複数の子データを、子データは親となりさらに子データを持つことができる。ただし、HDBとは異なり、子データは複数の親データを持つことができる。

(c) オブジェクト指向データベース

オブジェクト指向データベース (OODB : Object-Oriented DataBase) はオブジェクト指向設計をデータベース設計に使用したデータベースで、データをオブジェクトという形式で格納する。オブジェクト指向プログラミングとの連携がよいという特徴を持つ。

関係データベースにオブジェクト指向の考えと機能を追加したものをオブジェクト関係データベース (ORDB : Object-Relational DataBase) と呼ぶ。

(d) その他のデータベース

(ア) ハイパーテキストデータベース

WWW (World Wide Web) で使用されているようなハイパーテキストをデータとして、これを管理して使用するデータベースがハイパーテキストデータベースである。WWWもある意味ハイパーテキストデータベースの一つであるといえる。

(イ) マルチメディアデータベース

テキスト、静止画、動画、音声などマルチメディアなデータを管理するデータベースがマルチメディアデータベースである。どのような媒体のデータかを意識させることなく、利用者に使用させることができる。

(ウ) XMLデータベース

XMLデータベース (XML DB) はマークアップ言語であるXMLの構造を利用したデータベースである。リレーショナルデータベースよりも拡張性に優れ、またXMLをそのままデータとして使用できるものもあり、今後の普及が見込まれている。

1. 1. 2 データベースのモデル

データの表現方法を形にしたものが、**データモデル**と呼ばれ、モデル化することを**データモデリング**と呼ぶ。

データモデルにより、データをどのように構成するかを決め、またそれを格納し、どのように検索、取り出し、書き込みなど操作を行うかということが決定される。

(1) データモデル

データベースを設計する際には、情報処理システムを作成したい業務、例えば給与計算処理システムを作成する場合には給与計算という業務、この業務で使用するデータなどをデータベースに反映する形になる。この設計の際にまず考える必要があるものがデータモデルである。

詳しい設計の手順や方法などは後述するが、このデータモデルには3つのデータモデルが存在する。設計に使用する順番に並べると次のようになる。

(a) 概念データモデル

概念データモデルは、データベースに反映する業務におけるデータとその流れなどをモデル化したものである。これは実際のデータベースの構造や、データの格納形式、DBMSなどからは全く関係がなく、業務だけからモデルを作成する。

現実（業務）から必要な部分を取り出して図式化したものともいえるだろう。

(b) 論理データモデル（外部モデル）

概念データモデルの作成の次に行うのが**論理データモデル**を作成することである。概念データモデルで表された実際の業務でのデータなどに対して、適切なデータモデルをあてはめ、コンピュータ上で効率よく実現できるようなデータの構造を定義する。

ここではよく使用されるデータモデルである、関係モデル、階層モデル、ネットワークモデルの3種類をあげる。

(ア) 関係モデル

関係モデルは、データの項目間の相互関連にモデル化の基本をおき、数学の集合による関係の概念を用いて表現する。データの構造を表形式（**関係表**）で表現する。

このモデルを用いて作られるデータベースが**関係データベース**である。

(イ) 階層モデル

階層モデルは、データ構造にモデル化の基本をおき、データ間の関連を木（ツリー）を用いて、階層構造（親子関係）として表現する。

このモデルを用いて作られるデータベースが**HDB**である。

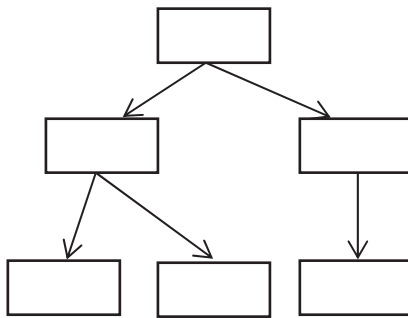
(ウ) ネットワークモデル

ネットワークモデルは、階層モデル同様、階層構造を用いて表現を行うが、子のデータは親のデータを複数持つことができるという点で階層モデルとは異なる。

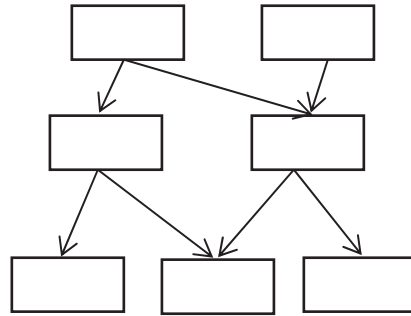
このモデルを用いて作られるデータベースがNDBである。

論理データモデル

(ア) 関係モデル



(イ) 階層モデル



(ウ) ネットワークモデル

(c) 物理データモデル (内部モデル)

論理データモデルを元に、実際のデータベースに合わせた形に設計するものが**物理データモデル**である。

アプリケーションからのアクセスに最適な格納方法、データへのアクセスの処理にかかる時間、実際の実装の準備など、実際のデータベースを作成するために必要な要件を決定する。

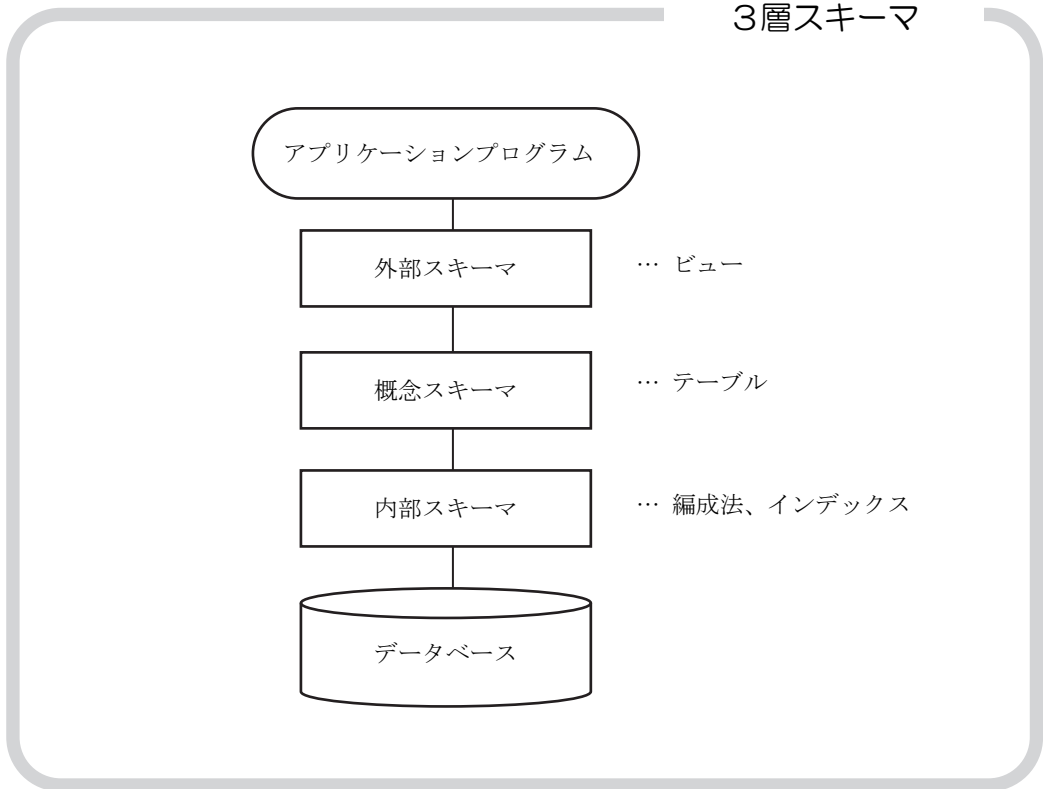
(2) スキーマ

スキーマとは、データベースを構築する実体などの関連を定義したもので、データモデルをコンピュータの世界へ写しとるために記述したものである。データベースでのデータの構造や、データベースに対しての操作に対する制約事項など、実際のデータベースの“定義”としての役割を果たす。

スキーマをどのように定義するかという点ではさまざまな手法があるが、一般的によく使用されているものが、ANSI/X3/SPARC（アメリカ規格協会／計算機情報処理部門標準化計画委員会）によって提示された、3層スキーマである。ANSI 3層スキーマとも呼ばれる。

3層スキーマでは、概念スキーマ、外部スキーマ、内部スキーマの3つのスキーマ構成を取り、これによりデータとアプリケーションプログラム間の独立を保つ。

3層スキーマ



(a) 概念スキーマ

データモデルによって、現実世界の情報を記述したものが**概念スキーマ**である。データの処理上必要な現実世界におけるデータの関連を表現したもので、データベース全体の記述であり、特定のアプリケーションプログラムやコンピュータシステムの物理的特性（補助記憶装置の特性など）から離れて定義したものである。

関係データベースでは“テーブル”に当てはまる。

(b) 外部スキーマ

外部スキーマ（副スキーマ）は、データベースを利用するアプリケーションプログラムにとってのデータベースの見え方を定義する。アプリケーションプログラムはデータベース全体を利用する必要はなく、自分の処理に必要な部分だけを認識すればよく、概念スキーマを基準に、アプリケーションプログラムの目的に応じて外部スキーマを定義する。つまり、1つの概念スキーマから、外部スキーマは利用するアプリケーションプログラムの数だけ作られることがある。

外部スキーマは概念スキーマから導き出されることで、アプリケーションプログラムの操作を概念スキーマの一部に限定するとともに、アプリケーションプログラムに都合の良いデータの構成などにもすることもできる。

外部スキーマと概念スキーマは影響を及ぼし合うことはなく、これによりデータの独立性を維持している。

関係データベースでは”ビュー”に当てはまる。

(c) 内部スキーマ

概念スキーマをコンピュータ上に実現させる技法を記述したものが**内部スキーマ**（記憶スキーマ）である。内部スキーマには、物理データベースの編成法、データ内部表現などが記述されている。内部スキーマはデータが実際に格納されている物理ファイルを管理している。

関係データベースではデータの編成法や、インデックスなどに当てはまる。

1. 1. 3 関係モデル

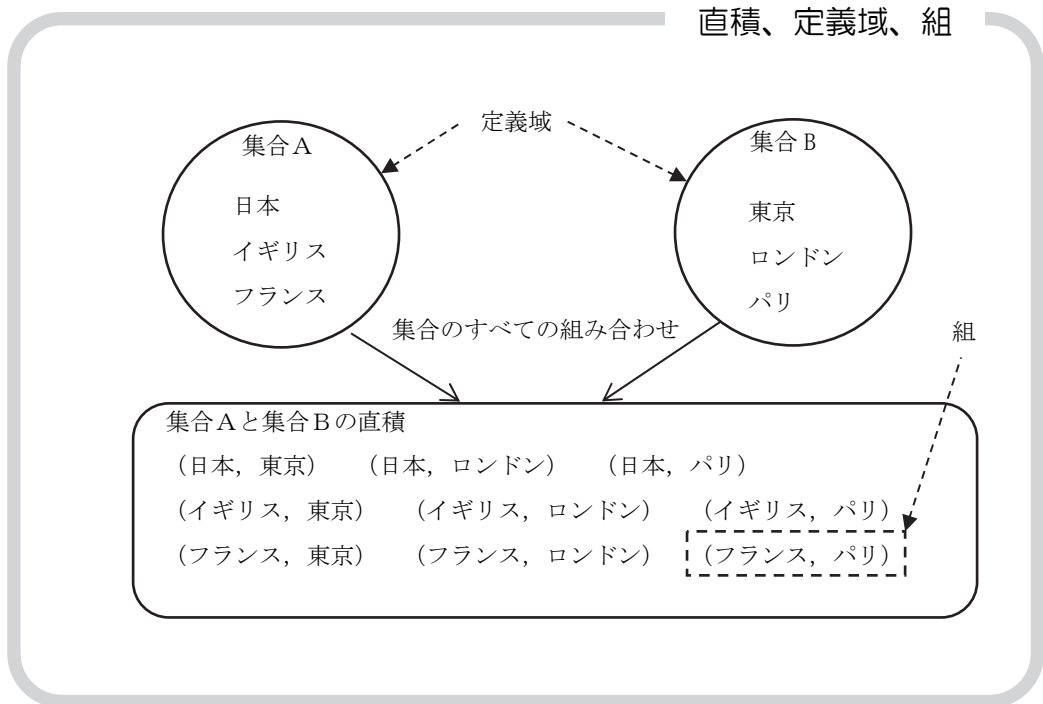
関係モデル（関係データモデル）は、現在最も使用されている関係データベースのモデルとして使われているモデルである。関係データモデルではデータの数学的な関係を用いて、データを表（関係表）として表す。

(1) 集合と関係

データベースにデータは集められて格納される。つまり、データの集合が集まっているのがデータベースである。

データベースに集まったデータの集合は、1種類の要素だけでなく、複数の要素から成り立っていることが通常である。例えば、国名データベースがあるとすれば、国名の集合A = {日本、イギリス、フランス} だけでもよいが、通常はこれに首都の集合B = {東京、ロンドン、パリ} も組み合わせて格納されている。このそれぞれの集合が持つ値を**実現値**と呼ぶ。

この2つ以上の集合のすべての組み合わせを**直積**と呼ぶ。この時、集合は直積の**定義域**（ドメイン）と呼ばれる。また、直積によって作られた要素を**組**（タプル）と呼ぶ。



しかし、直積の組の中には、ありえない組み合わせが存在する。上図でいえば、(日本, ロンドン) の国名、首都の組み合わせは存在しない。直積の中から、いずれかの対応関係を示すような組を抜き出した部分集合を**関係** (リレーション) と呼ぶ。また、定義域の名前を**属性名** または**属性** (フィールド) と呼ぶ。

この部分集合である関係を、組を行に、属性を列にして表形式にしたものが**関係表**である。

関係表

集合A (国名) = {日本、イギリス、フランス}

集合B (首都) = {東京、ロンドン、パリ} があるとき、直積、組、関係を表す。

直積

(日本, 東京)	(日本, ロンドン)	(日本, パリ)
(イギリス, 東京)	(イギリス, ロンドン)	(イギリス, パリ)
(フランス, 東京)	(フランス, ロンドン)	(フランス, パリ)

↓ 関係だけを抜き出す

関係

(日本, 東京)	(イギリス, ロンドン)	(フランス, パリ)
----------	--------------	------------

	定義域 1	定義域 2	
関係表	国名	首都	←--- 属性名
	日本	東京	←--- 行 (組)
	イギリス	ロンドン	
	フランス	パリ	
	↙-----↘ 列 (属性)		

(2) 関係表の特徴

関係表は次のような特徴を持つ。

- ・ 行 (組) は関係の集合であり、その順列 (行の順序) に意味を持たない
- ・ 同じ行 (同じ値を持つ組) が複数存在することはない
- ・ 各列に名前 (属性名) がつけられていれば、列の順序に意味はない
- ・ 属性値はこれ以上分割できない
- ・ 関係表は関係の集合であるため、集合に対する演算 (和・差・積) が可能である

1. 2 データベース管理システム

データベース管理システム (DBMS : DataBase Management System) は、データベースシステムを実現し、データベースを効率よく構築・運用・維持管理するためのソフトウェア体系である。データベースを維持管理するためには、データベースの生成・保守および利用者からのデータ参照・更新・機密保護などに正しく対応する機能が必要となる。

1. 2. 1 DBMSの機能

DBMSは大きく分けて3つの機能から成り立っている。データベース定義機能、データベース操作機能、データベース制御機能の3つである。これを操作するデータベース言語と合わせて説明する。

(1) データベース言語

データベースを扱うための言語をデータベース言語と呼ぶ。データベース言語を使うことで、アプリケーションプログラムはデータベースに対してアクセスを行うことができるようになる。また、データベース言語により、利用者、アプリケーションプログラムに対し内部構造を隠したまま、問い合わせなどに応答することができる。

データベース言語はデータベースを使用する様々なアプリケーションプログラムで利用されるため、標準化が行われている。

現在最も使用されている関係データベースでは、データベース言語としてSQL (Structured Query Language) が使用されている。SQLはANSI、ISOによって標準化が行われている。

(a) データベース言語の分類

データベース言語はDBMSの機能ごとに分類できる。

(ア) データ定義言語

データ定義言語 (DDL : Data Description Language) はデータ構造の作成、破棄などデータの定義に関する言語である。SQLではSQL-DDLと呼ばれる。

(イ) データ操作言語

データ操作言語 (DML : Data Manipulation Language) はデータの検索、更新、削除などデータの操作に関する言語である。SQLではSQL-DMLと呼ばれる。また、データ操作言語にはその方式から2種類に分けることができる。

【親言語方式】

別の高水準プログラミング言語の中に、DMLを組み込むことでプログラムからデータベースにアクセスできる方式が**親言語方式**である。この時、使用されるプログラミング言語を親言語と呼ぶ。

親言語方式には、プログラムの中に直接書き込むことができる埋め込み方式と、**モジュール言語**で記述されたデータベースアクセス用のモジュールをプログラムから呼び出す方式がある。

【独立言語方式】

コンピュータの知識を持たないユーザでも比較的簡単にデータベースを利用できるように考えられた方式が**独立言語方式**で、対話形式によって必要項目を入力して実行できる形になっている。また、通常の英語に近い構文の指令を入力することによって操作する方式を**コマンド方式**と呼ぶ。

(ウ) データ制御言語

データ制御言語（**DCL** : Data Control Language）はデータベースに対するセキュリティやトランザクションに関する言語である。SQLでは**SQL-DCL**と呼ばれる。

(2) DBMSの機能

DBMSが行う機能についての概略を説明する。詳細な内容については、後述する。

(a) データベース定義機能

データベースにおける定義はスキーマで行われる。つまり、DBMSの**データベース定義機能**はスキーマを定義する機能となる。DBMSはデータ定義言語を使用して、データの構造、関係データベースでいえば関係表の名前、属性などを定義する。

(b) データベース操作機能

データベースに対して利用者またはアプリケーションプログラムが実行するデータの検索、追加、更新、削除などを行うのが**データベース操作機能**である。DBMSはデータ操作言語を用いて操作を行う。

(c) データベース制御機能

データベース運用中に発生するデータの不正や処理の矛盾、障害などからデータベースのデータを保護する機能が**データベース制御機能**である。DBMSはデータ制御言語を用いて保護を行う。

(d) その他の機能

上記の主要な3つの機能以外にも、DBMSは次のような機能を持つ。

- ・ 保全機能 … データの不正な登録や更新を防ぐ
- ・ データ機密機能 … データのセキュリティを維持し、不正な利用を防ぐ
- ・ 排他制御 … データの二重更新などデータの矛盾を防ぐ
- ・ 障害回復 … データベースの障害から復旧する

1. 3 データベース管理者

データベースはデータの共有と共用を行うためのもので、データを特定の業務と切り離して利用ができる。そのため、個々の利用者の立場を離れて、データベースとデータ全体を管理し、運用状況を監視する役割を持つ人間が必要となる。この役割はデータベース管理者（DBA：DataBase Administrator）、またはデータ管理者と呼ばれ、個人、またはグループでその役割を果たす。

1. 3. 1 データベース管理者の役割

データベース管理者といった場合には2つの分類がある。データベース設計管理者とデータベース運用管理者である。2つをそれぞれ別の人物・グループに割り当ててもよいし、1人・1つのグループで行ってもよい。

(1) データベース設計管理者

データベース設計管理者はデータベースのデータ全体を設計・管理する管理者で、次のような役割が要求される。

- ・データベース化のためのデータの論理構造を定義する
- ・利用者や利用部門のためにデータベースの見方を定義する
- ・データベースのハードウェア装置や物理構造を定義する

(2) データベース運用管理者

データベース運用管理者はデータベースの運用状況を監視し、データベースを最善の状態に整備する管理者で、次のような役割が要求される。狭義のデータベース管理者はこちらを指すことが多い。

- ・データベースシステムの運用状況を監視する
- ・データベースシステムの性能を評価する
- ・障害に備え、更新ジャーナルなどをとり復旧処理に備える
- ・運用状況に合わせて、データベースの再編成あるいは再構成を行う
- ・データベースの障害復旧処理を行う