

THE DATABASE  
imes  
HOTCHPOTCH SOCIETY





# The Database Times

vol. 1



Hotchpotch Society



# まえがき



The Database Times vol.1 をお手にとっていただきありがとうございます。

本書は、データベースシステムを中心として、様々な情報技術に関する話題を読者のみなさんにお届けすることを目的としています。みなさんは「データベースシステム」というものにどのようなイメージをお持ちでしょうか。SQL を投げるとデータを返してくれる、なんだかよくわからないけれど枯れた時代遅れの技術だというイメージを持っている人が多いのではないかと思います。

しかしながら、実際のデータベースシステムは古の技術を土台として、その時代の最先端の技術が詰め込まれた非常に魅力的な技術の集大成なのです。世界中で生み出されるデータ量がどのくらいかご存知でしょうか。2010 年までに生み出されたデータ量はなんと 1,227 ペタバイト。しかも、2020 年には 7,910,000 ペタバイトになるであろうと予測されています。人類が産み出そうとしているデータ量は、まさに爆発的な勢いで増え続けています。このデータの膨張を支える縁の下の力持ちがデータベースシステムです。加速するデータ量の増加に伴って、データベースシステムも日々進歩を続けているのです。

そんなデータベースシステムがどのようにして生まれ、そして最近ではどんなことが話題になっているのか、本書を通してその一端に触れ、楽しんで頂ければ幸いです。

2012 年 8 月

Hotchpotch Society

## お品書き

### ■ データベースシステムの夜明け

今の世の中、データベースシステムといえバリレーショナルデータベースシステムです。そのリレーショナルデータベースシステムが、一体どのようにして生み出されたのか、そしてどのように発展を遂げていったのか、その歴史を辿ります。

### ■ PostgreSQL カンファレンス 2012 レポート

オープンソース DBMS 代表格の一つである PostgreSQL は、最近では業務でも用いられることがかなり増えてきているようです。そんな PostgreSQL の最新技術事情に触れられる PostgreSQL カンファレンスの参加レポートをお届けします。

### ■ とある世界で一番高速な Brainfuck インタプリタ

近年では SAP HANA や MemSQL などの主記憶データベースが話題となっているように、膨大するデータ量を捌ききるための「速さ」が強く求められています。データベースシステムは SQL というプログラミング言語の処理系という側面も持ちあわせており、言語処理系の高速な実装はデータベースシステムには必要不可欠です。言語処理系をいかに高速にするのか、その技術について世界最速 Brainfuck インタプリタの実装者が解説します。

### ■ クラウド時代の DNS

#### ■ IPv4 がこの先生きのこるには

最近では「クラウド」という言葉が話題となっていますが、その背景にはもはや単一のコンピュータだけでシステムを構築してすべてがうまくいく時代は終焉を迎え、ネットワークによりコンピュータを有機的に結合することが必須となりつつある現代の技術トレンドがよみとれます。ネットワーク技術では今一体何が起きているのか、その最前線に迫ります。

### ■ 電算機技能者の同人誌執筆環境構築概論

本書は L<sup>A</sup>T<sub>E</sub>X で執筆されています。プログラマが L<sup>A</sup>T<sub>E</sub>X で本を書くというのはどういうことなのか、その開発 (執筆) 環境を紹介します。

# 目次



まえがき	iii
1 データベースシステムの夜明け	1
2 PostgreSQL カンファレンス 2012 レポート	17
3 とある世界で一番高速な Brainfuck インタプリタ	19
4 クラウド時代の DNS	23
5 IPv4 がこの先生きのこるには	29
6 電算機技能者の同人誌執筆環境構築概論	35
雑談	37



## データベースシステムの夜明け

はやみず

データベースシステム無しでは、今日の社会は成り立たないと言って良いでしょう。社会が高度に情報化された現在、世界中で大量のデジタルデータが日々生み出され、飛び交い、消費され、そして蓄積されてゆきます。一方で、膨大なデジタルデータは、単に生み出され、蓄積されてゆくだけではゴミも同然です。必要なときに必要なデータが取り出せるよう、適切に管理してゆかなければなりません。そのための根幹たる存在がデータベースシステムなのです。

今日においては、「データベースシステム」と呼ばれているその殆どが「リレーショナルデータベースシステム」です。みなさんが馴染みのあるであろう MySQL や PostgreSQL,あるいは Oracle といったデータベースシステムは、すべてリレーショナルデータベースシステムですね。リレーショナルデータベースシステムの歴史を辿ると、その起源はある一篇の論文に遡ることができます。

その論文こそが、Edgar F. Codd により 1970 年に発表された “A Relational Model of Data for Large Shared Data Banks” です。この論文は、リレーショナルデータベースの最も重要な基礎となるリレーショナルモデルを提唱したもので、いわばデータベースシステム分野における金字塔です。古典力学を Newton が拓き、相対性理論を Einstein が拓いたとするならば、データベースの一大分野であるリレーショナルデータベースを拓いたのは Edgar F. Codd その人といって間違いないでしょう。

データモデルとは、どのような形式でデータを格納して、どのようにデータにアクセスするかということを取り決める枠組みです。データを蓄積・管理することを目的とするデータベースシステムにおいては、データモデルはその心臓部とも言える存在なのです。

そして、Codd によるリレーショナルモデル提唱から数年の後に、UNIX で動作する世界初のリレーショナルデータベースシステムの開発プロジェクトが立ち上がります。Michael Stonebraker 率いる **INGRES** プロジェクトです。Codd により確立されたりレーショナルデータベースシステムの基礎理論を、実際に動くソフトウェアとして実現し、そしてそれを世に広めたのが INGRES なのです。

本稿では、Codd によるリレーショナルモデルの提唱から、INGRES プロジェクトの黎明期の記録を辿り、現代の社会を支えるデータベースシステムがいかにして創り上げられ



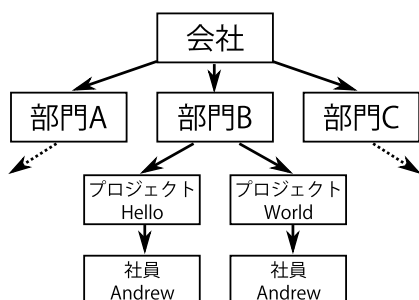


図 1.1 階層型モデル

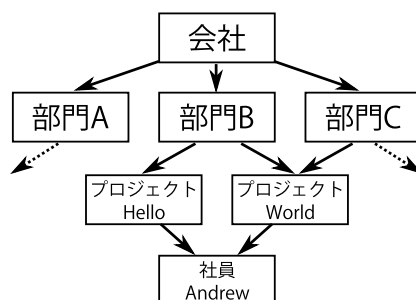


図 1.2 ネットワーク型モデル

たのか、その歴史を紐解いてみようと思います。

## §1 リレーショナルモデルの登場

リレーショナルモデルが登場する以前、つまり 1960 年代までのデータベースシステムは殆どが階層型モデルやネットワーク型モデルというデータモデルに基づいて構築されていました。

階層型モデルというのは、木構造を用いてデータを組織化したデータモデルです。たとえば会社組織を階層型モデルで表すとしましょう。この会社では、会社の中に複数の部門があります。各部門の中では複数のプロジェクトが運営されています。そして各プロジェクトに従業員が属しています。そうすると、図 1.1 のような階層構造でデータを組織化することができます。

ここで、例えば 2 つのプロジェクト Hello project と World project に属する 1 人の社員 Andrew がいたとしたらどうでしょう。階層型モデルにおいて、1 つのデータの実体 (社員やプロジェクト) が複数の親データに属することができません。つまり、1 人の社員が複数のプロジェクトに属することを直接表現することができないのです。無理やりこれを表現しようとする、プロジェクトごとに Andrew のデータを持つことになり、データの重複が生じてしまいます。また、部門 B と部門 C が共同でプロジェクト World を運営していることを表現しようと思うと、プロジェクトのデータに加えてその下にぶら下がっている社員のデータも併せてコピーしなければなりません。階層型モデルは非常にシンプルで性能が出しやすいというメリットもあるのですが、このようにデータ同士の関係を柔軟に記述することができないというデメリットもあるのです。

このような階層型モデルの欠点を克服したのがネットワーク型モデルです (図 1.2)。ネットワーク型モデルでは、木構造ではなくグラフ構造によりデータの組織化を行います。グラフ構造であれば Andrew が複数プロジェクトに属することを自然に表現できる

ために、前述のようなデータ重複の問題は発生しません。

1960年代には、IBMによってIMS(Information Management System)という階層型モデルにもとづくデータベースシステムが開発されており、企業におけるデータ管理はIMSの独壇場ともいえる状況でした。1960年代の終盤には、ネットワーク型モデルの標準化団体としてCODASYLが立ち上がり、CODASYLデータモデルが標準規格として確立されます。

そんな中、IBM社内でこの動きに疑問を持つ一人の研究者がいました。この人こそ、後にリレーショナルモデルの提唱者となるEdgar F. Coddその人でした。

階層型モデルやネットワーク型モデルに基づくデータベースは、データのアクセスに一つ大きな問題を抱えていたのです。これらのデータベースにおいてデータの問い合わせを行う際には、データの構造がどのようになっているかを把握し、そしてどのような手順でデータを取得するかをプログラマ(データベースシステムのユーザ)が知っている必要があります。例えば図1.2のデータベースにおいて社員Andrewのデータを取得するためには、「会社のデータを読みとり、部門Bのデータを読み取り、プロジェクトHelloのデータを読み取り、社員Andrewのデータを読み取る」という手順を指定してあげなければなりません。このように、欲しいデータにアクセスするために、プログラマがデータの案内(navigation)をしなければならないという意味で、階層型モデルやネットワーク型モデルは**ナビゲーションモデル**、またそれにもとづくデータベースシステムは**ナビゲーションデータベースシステム**と呼ばれます。

もしもナビゲーションデータベースシステムにおいて、図1.2の会社で組織変更があり、各部門の下には課が設置され、その下にプロジェクトが属するという構造にデータベースが修正されたとしたらどうなるでしょう。これまで社員のデータにアクセスするアプリケーションは会社→部門→プロジェクト→社員とたどっていましたが、会社→部門→課→プロジェクト→社員とたどるように修正しなければなりません。このようにナビゲーションデータベースシステムでは、その上位構造に変化があった時にはデータにアクセスする方法を再構成しなければなりませんでした。

当時のIBM社内では、データベースの仕様変更が生じるたびに、プログラマ達がデータアクセスのコード書き換えに追われていました。

この状況を打破するべく、Edgar F. Coddは1970年<sup>\*1</sup>にリレーショナルモデルを提唱します。

リレーショナルモデルでは、データの論理的な構造と物理的な構造を独立させることができるように考慮されています。そのため、データ構造を変更するたびにアプリケーショ

---

<sup>\*1</sup> 正確にはリレーショナルモデルの論文は1969年にIBMの社内技術報に掲載され、翌1970年に米国コンピュータ学会の論文誌に掲載されます。

ンを書き直す必要がありません。またユーザは「どんなデータが欲しいか」を記述するだけで、具体的なアクセス方法はデータベース側が判断してデータを取得することができます。つまり、ナビゲーションデータベースシステムでは“how”を与えなければデータのアクセスが行えなかったのですが、リレーショナルデータベースは“what”を与えるだけでデータのアクセスが可能となります。

また、リレーショナルモデルは集合論にもとづく数学的基礎の上に成り立っており、後のデータベースシステム研究のための大きな理論的基盤となりました。これまでプログラマによる職人芸の上に成り立っていたデータベースシステムは、リレーショナルモデルの登場によって科学の領域へと押し上げられたのです。

## §2 INGRES プロジェクト始動

1970 年にリレーショナルモデルが提唱されてから、しばらく大きな動きはありませんでした。Codd がいた IBM においても、発表当初はあまり注目を集めることはなかったようです。

その一方で、1973 年が終わろうとする頃、その論文が一人の若き研究者の目に留まります。その人こそ、INGRES プロジェクト、そして後のデータベース業界全体を率いる存在となる Michael Stonebraker です。Stonebraker は 1971 年にミシガン大学で博士号を取得し、UC Berkeley で研究職についたばかりでした。テニュアトラック<sup>\*2</sup>を走り始めたばかりの Stonebraker にとって、リレーショナルモデルは恰好の研究対象だったのでしょう<sup>\*3</sup>。リレーショナルモデルに出会った Stonebraker は、同僚の Eugene Wong と共にリレーショナルデータベースシステムの実装に乗り出します。こうして INGRES プロジェクトは始まりました。

INGRES という名前は、**I**nteractive **G**raphic and **R**etrieval **S**ystem の頭文字をとったもので、本来は UC Berkeley の経済学の研究チームのために、地理情報をグラフィカルに表示するためのシステム研究として予算を獲得していたプロジェクトでした。そこは本音と建前で、まずはデータを効率的に取得するためのデータベースシステムが必要だ、ということで Stonebraker 達はデータベースシステムの開発にのめりこんでゆきます。

<sup>\*2</sup> テニュアとは大学における終身雇用資格のことです。若手研究者は任期の限られた研究職につき、テニュア獲得を目指して研究業績を積み上げてゆく、というのは米国における一般的な研究者のキャリアパスです。このテニュア獲得のための若手研究者が通る道をテニュアトラックと呼びます。テニュアトラックでいかに論文を“量産”できる研究ネタを選ぶかというのは、若手研究者にとっては人生に関わる重要な決断なのです。

<sup>\*3</sup> ちなみに Stonebraker の博士論文は“The Reduction of Large Scale Markov Models for Random Chains”というタイトルで、もともとは数学寄りの研究をしていたようです。