

# データベース 第1回

---

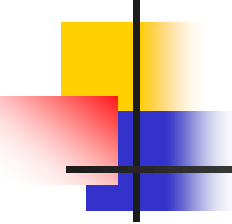
情報工学科  
木村昌臣



# データベースとは何か

---

- データベースとは
  - データベースシステムとDBMS
- データモデル
  - 概念モデルと論理モデル
  - 実体－関連モデル
- リレーショナルデータベースの意義



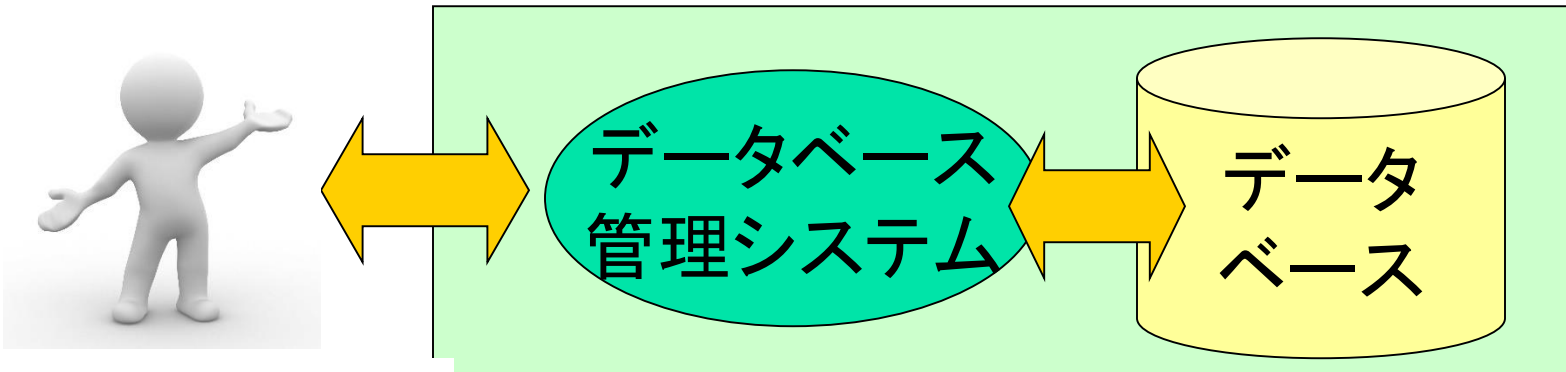
# データベースとは

---

- データベースとは？
  - 複数の応用目的での共有を意図して**組織的かつ永続的に格納されたデータ群**
  - 大量データの検索、更新処理など機能を提供
- データベースの適用例
  - 企業の顧客・在庫・売上などのビジネスデータの管理
  - 文書データ、地理データベース、DNAデータ、動画像データ などマルチメディアデータにも適用

# データベースシステムとDBMS

- データベースシステム
  - データベース
    - データを組織的・永続的に管理
  - データベース管理システム(DBMS)
    - データベースを管理するためのソフトウェア



プログラムまたは人



# DBMSの役割

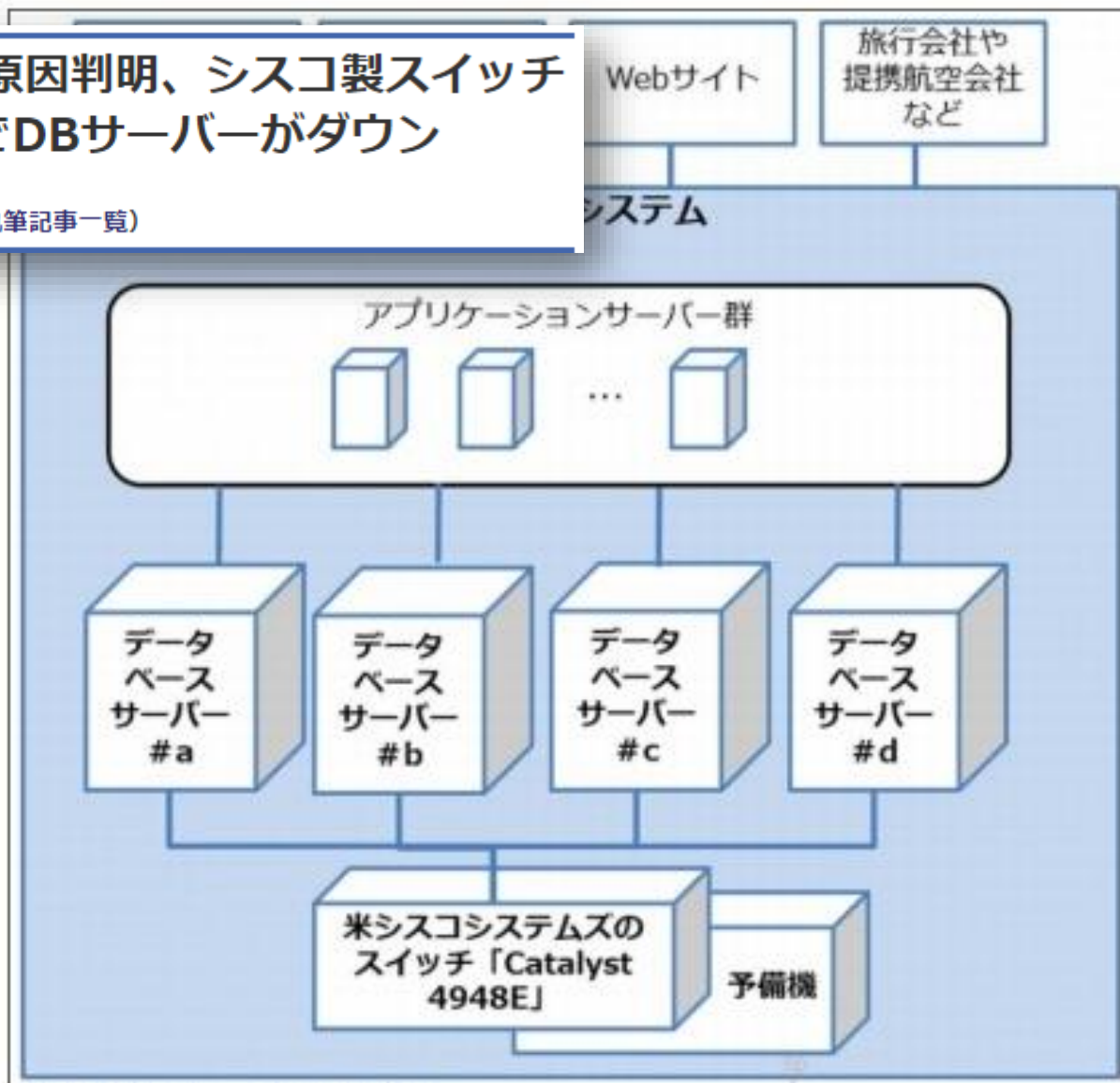
---

- ペタバイト・エクサバイト単位のデータも取り扱える
  - 格納・管理・検索
- 数百万ユーザーからの同時アクセスを（矛盾なく）処理可能
- システム障害からデータを守る
  - データは決して失われない
- 24時間365日稼動が求められる

# ANAシステム障害の原因判明、シスコ製スイッチの「世界初のバグ」でDBサーバーがダウン

2016/03/30

井上 英明=日経コンピュータ（筆者執筆記事一覧）



国内旅客システムの概要図

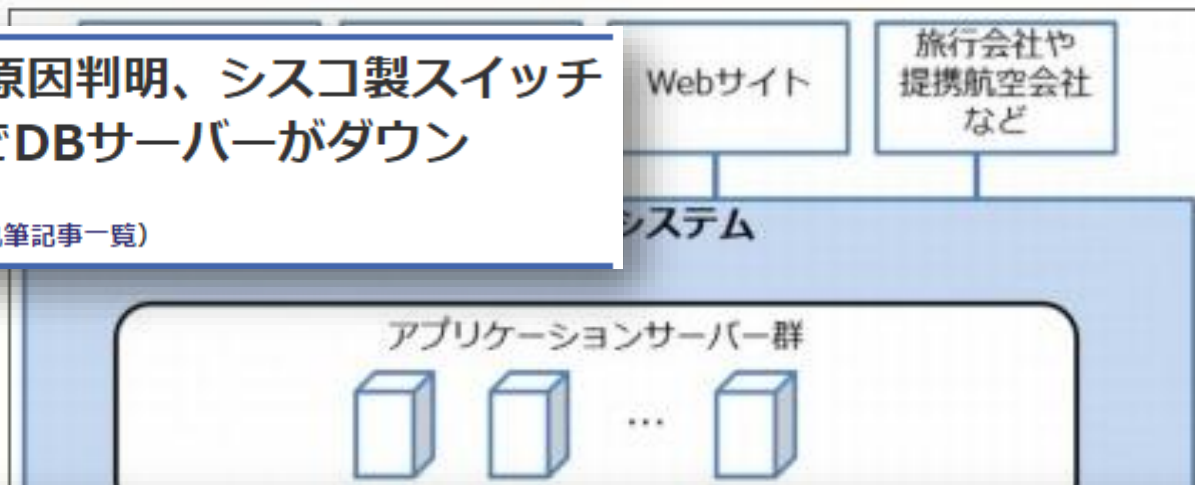
全日本空輸の公表資料を基に編集部が作成

[画像のクリックで拡大表示]

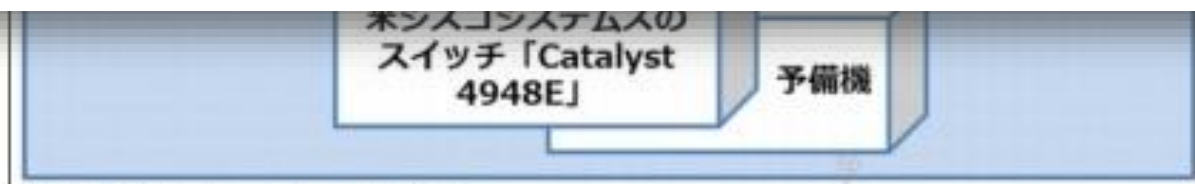
## ANAシステム障害の原因判明、シスコ製スイッチの「世界初のバグ」でDBサーバーがダウン

2016/03/30

井上 英明＝日経コンピュータ（筆者執筆記事一覧）

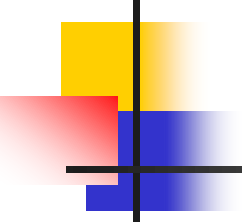


システム障害はANAだけで539便、約7万2100人に影響した。22日当日は146便が欠航し、1万8200人が足止めされた。遅延便は391便で約5万3700人の移動が遅れた。機材繰りの影響で翌23日も2便が欠航し、約200人が影響を受けた。これに加え、ANAのシステムを使う提携航空会社で少なくとも4社で欠航や遅延が生じた（関連記事：[\[続報\] ANAシステム障害、1万6100人に影響広がる](#)）。



国内旅客システムの概要図

全日本空輸の公表資料を基に編集部が作成  
[画像のクリックで拡大表示]



---

データベースの基礎

# データモデル



# データモデル

DBの基となる実世界

概念設計

現実のデータを  
整理

概念モデル：実世界を表現する抽象  
的なスキーマ

論理設計

どのようにスキーマ  
を記述するか

論理モデル：DBMSで管理可能  
なスキーマ



# 概念モデルと論理モデル

---

- 概念設計の狙い

- DBMS のスキーマとは独立に実世界のデータを整理

- 概念モデル

- 人間が見て理解できる図式
- 1976年にP. P. Chenが、実体-関連モデル (Entity-Relationship model)を提案

- 論理モデル

- DBMS で管理可能なスキーマ
- 例えば、リレーショナルデータモデル



# 概念データモデル



# 実体－関連モデル

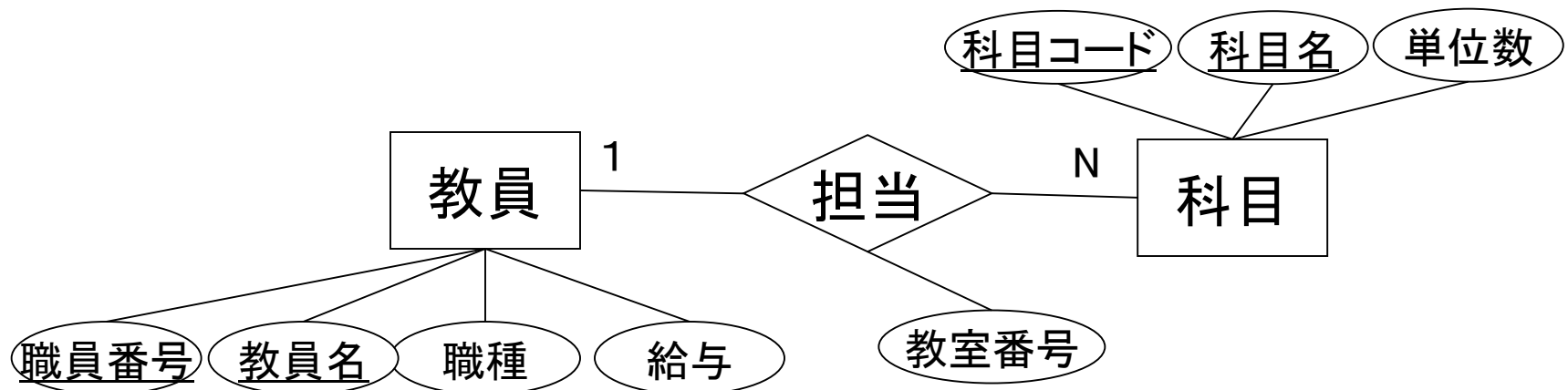
---

- 実体－関連モデルでの実世界の見え方
  - 実体 (entity) : 実世界に存在する事物
    - 名詞に相当
    - 実体型 (entity type) を構成
  - 関連 (relationship) : 実体間の相互関係を表現
    - 動詞に相当
    - 関連型 (relationship type) を構成
  - 属性 (attribute) : 実体、関連に付随する性質
    - 形容詞に相当

# 実体-関連図の構成要素

## ■ 実体-関連図 (E-R図)

- 実体と関連を図式で示したもの
- 矩形: 実体の集合
- 菱形: 関連の集合
- 楕円: 属性の集合
- 線: 実体間の対応関係



# 実体間の対応関係

## ■ 対応関係の分類

- 1対1関連型

- 1対多関連型

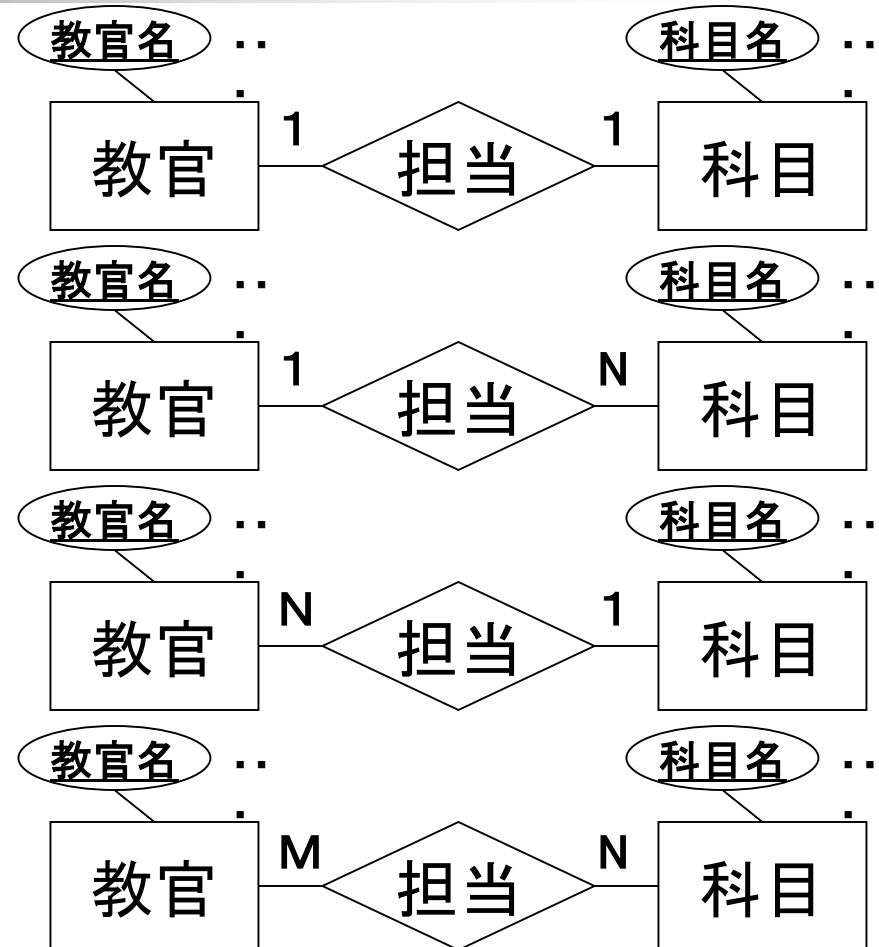
- 多対1関連型

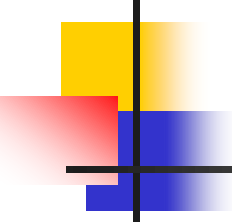
- 多対多関連型

## ■ 主キー (primary key)

- 実体を識別する属性集合

- 下線を引き、他と区別 (関連の属性の書き方に注意)





# 次の文に含まれるデータ構造を E-R図で表現してみよう

---

- 各学生は複数の科目を履修できる
- 科目の情報として科目名と単位数がある
- 学生の情報として学籍番号、氏名、住所がある
- 科目を履修したら、科目名・学籍番号の組に対して(テストの)得点が付加される

## 【参考】拡張E-R図(鳥の足法)



実は、いまはあまり使われていない。なぜなら、エンティティもリレーションも同じ「表」の形をしているのに、役割によって異なる記号をつけられてしまう。実際は、以下の鳥の足記法が使われることが多い。



1:多

1:1





---

# 論理データモデル

# 論理データモデルの種類と歴史

- 手続き型プログラミング言語(COBOLなど)を使ってファイルを直接操作
- 効率的にデータ処理を行うシステムを開発
  - ネットワークデータモデル(網構造)
  - ハイアラキカルデータモデル(階層構造)
- データモデルと物理構造の分離(データ独立)が 進展
  - リレーショナルデータモデル
  - オブジェクト指向データモデル(参照構造)
- 現在はリレーショナルデータモデルが主流



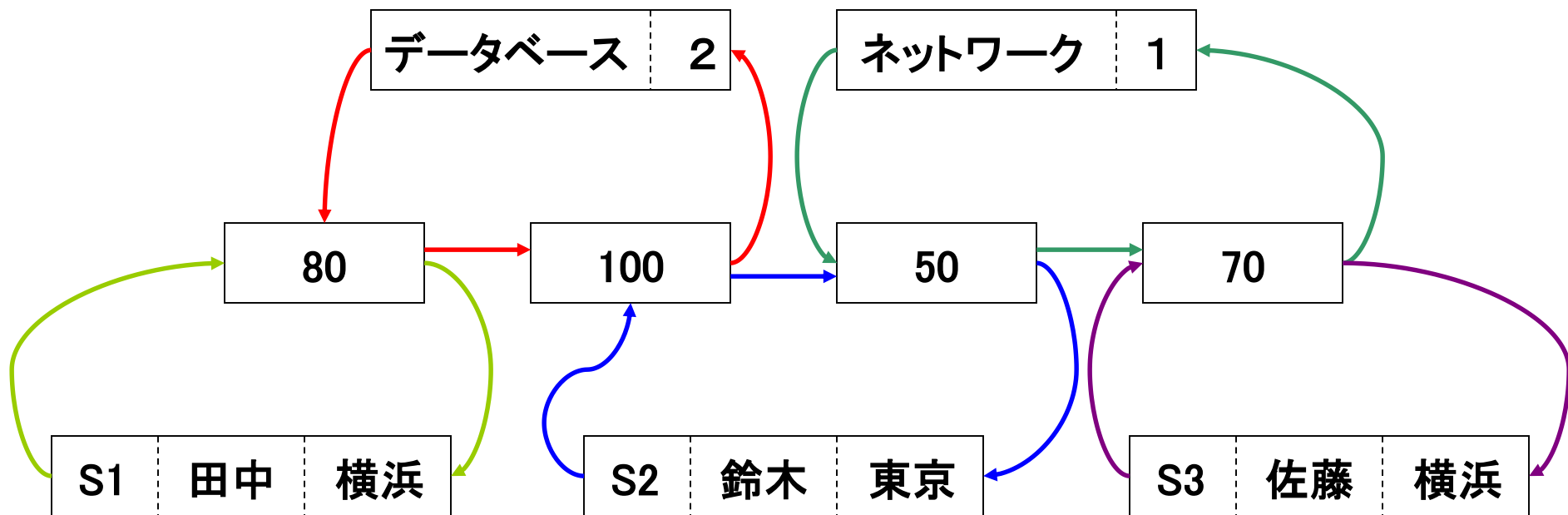
# ネットワークデータモデル

---

- ネットワークデータモデル

- 1971年にCOBOLのデータベース機能として提案される
- インスタンスはレコードオカレンスで表現
- 親からすべての子を、一連のポインタをたどることで表現
- ポインタを順にたどってデータ処理を実行

# ネットワークデータモデル例



レコードオカレンス: 

--	--	--

ポインタ:

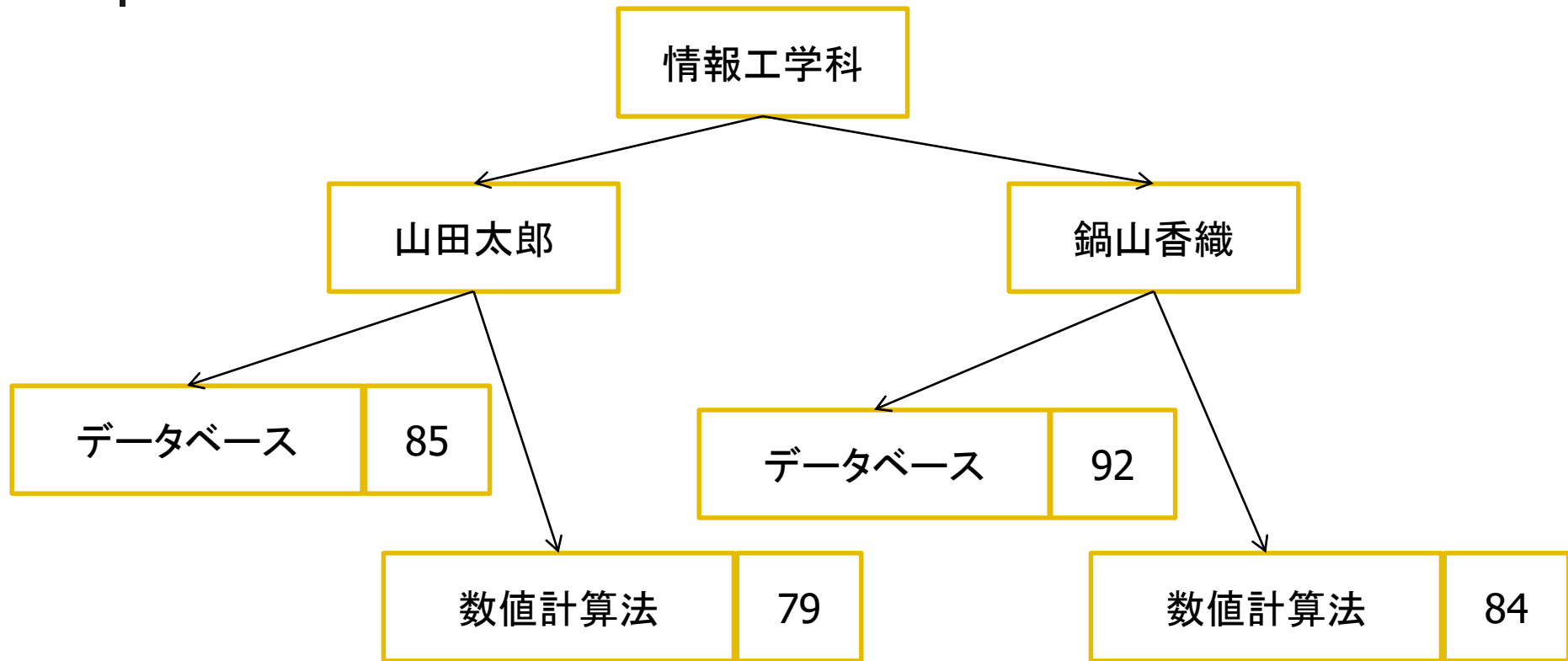


# ハイアラキカルデータモデル

---

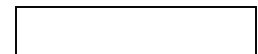
- ハイアラキカルデータモデル
  - 1968年にIMS (IBM) によって開発
  - レコードの階層構造によってデータを表現
  - 多対多関係の表現が課題
  - 階層をたどってデータ処理を実行

# ハイアラキカルデータモデル例



レコード:

ポインタ:





# リレーショナルデータモデル

---

- リレーショナルデータモデル
  - 1970年にCodd (IBM) によって関係理論をベースに提案
  - 他のデータモデルと比べて、形式的でデータ独立性が高い
    - 複数の属性の組み合わせによってリレーション（関係、あるいは2次元の表）を定義
    - リレーション同士の関係演算により関係完備（Relational Complete）な処理を実現
  - データ構造が複雑なデータ格納に課題



# リレーショナルデータモデル例

科目

科目名	単位数
データベース	2
ネットワーク	1

学生

学籍番号	学生名	住所
S1	田中	横浜
S2	鈴木	東京
S3	佐藤	横浜

履修

科目名	学籍番号	得点
データベース	S1	80
ネットワーク	S2	100
データベース	S2	50
ネットワーク	S3	70



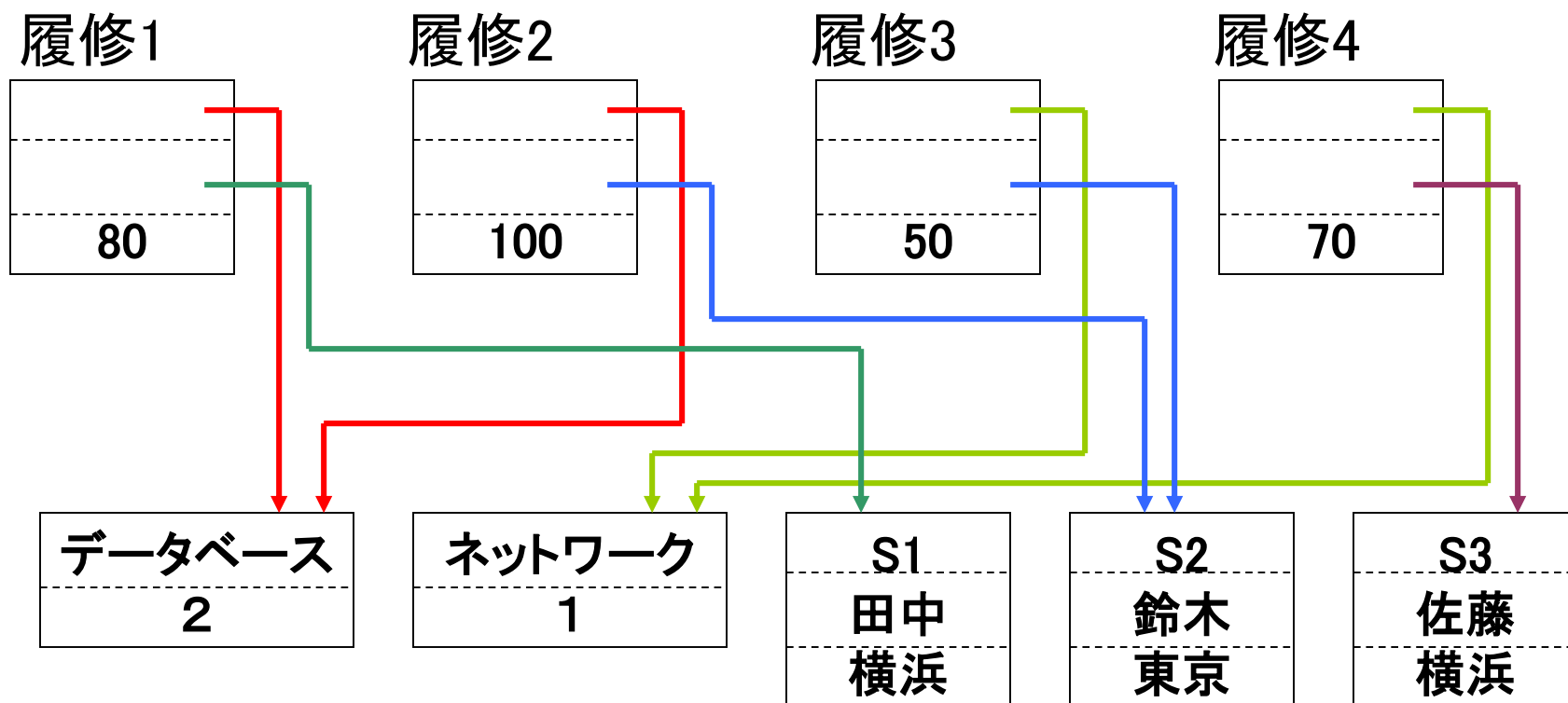


# オブジェクト指向データモデル

---

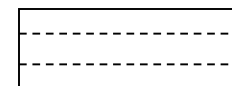
- オブジェクト指向データモデル
  - オブジェクト指向によりデータをモデリング
  - スキーマとしてクラスを定義
    - メソッドの定義
    - 継承 (inheritance) ・ポリモルフィズム (polymorphism) などのオブジェクト指向の概念
  - Java や C++ などの複合オブジェクト (composite object) を永続化
  - 手続き型プログラミング言語からシームレスに操作可能

# オブジェクト指向データモデル例



インスタンス(オブジェクト):

参照識別子:





# ファイルシステムとの比較

---

## ■ ファイルシステムの課題

### ■ データとプログラムの相互依存関係

- データを作成したプログラムだけがデータの取り出し、変更が可能で、共用できない

### ■ データの整合性を一元的に管理できない

- データの一貫性(integrity)制御に課題
- アクセス権限(privilege)の管理が不可
- 機密保護(security)、障害回復(recovery)などの支援ができない

## ■ リレーショナルデータベースは上記の課題を解消