

データベース論第5回

[データ]

実世界→情報→形式化(量としてまとめる)

(この点については前回の図を参照)

実世界の情報対象→実体(Entity)と呼ばれる

用語の使い分け
型(かた):Model
形(かた):Type

例) 学生 **レコード型(実体型)**

学籍番号	学科	氏名	出身地	現住所
7001-xxxx	情報社会	鈴木花子	豊橋	浜松

データ(インスタンス)レコード

(この場所のことを**データフィールド**と呼ぶ)

データ型：データモデル

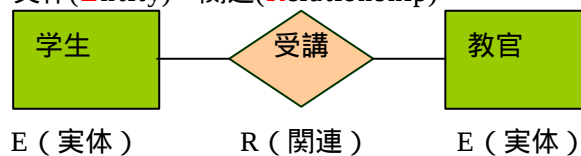
- ・階層構造(モデル)
 - ・網構造(モデル)
 - ・関係型(モデル)
- } 構造型データベース

階層モデル : 単純、1つの実体、関連(Relationship)を記述(階層化)

網モデル : 実世界の複数の実体、関連を記述(ただし、複雑で修正、更新が難しい)

データベースの分析、設計モデル

F-R モデル 実体(Entity)—関連(Relationship)



この場合、生徒が複数人いるので、n:1の関係になる

他の関係

n:n の関係 (人間関係など複雑なもの)

1:1 の関係 [学生 A]---<名前>---[静岡 太郎] [学生 B]---<学籍番号>---[7001-xxxx]など。

EDPS → データベースシステム(階層 → 網 → リレーショナルモデル) への変遷

処理方法

Batch(バッチ) : 一括まとめ処理

(給与計算、在庫管理など。時間が遅れても良い)

Real Time : 即時処理

(JR みどりの窓口の座席予約システムなど、要求に即応する必要があるものの向け)

データベースモデル出現以前のデータに対する Real Time 処理の要求

→ 逆ファイル形式(転地ファイル形式)が使われた(現在でも使われている)

Inverted File

社員ファイル

社員番号	氏名	所属	職
0201 メモリ番地 R1	上野修一	経理	保管
0202 R2	鈴木花子	庶務	事務
0203 R3	小川 広	営業	部長

名前	メモリ番地
上野修一	R1
鈴木花子	R2

職	メモリ番地
保管	R1
事務	R2

通常 番地からデータを探す、逆にデータから番地を探し、その番地から他のデータを引き出す。(二重以上にデータを作る → データ容量がやたら膨れる)

例)

「上野さんの所属は？」 → **[上野さん → R1]** → **R1 のデータを引っ張ってくる** → 「経理です」

旧型のシステムだが未だに使われているところもある。

データウェアハウス（大規模データベースを構築）

企業全体 地域情報全体

↓

データマイニング(大規模データベースから新しい知識データベースを創出する)
(知識処理、統計処理)

世界的な課題

インターネットをどうするか？データ、サーチエンジンの乱立

- ・ ダブリン・コア

（サーチエンジン等の標準化をはかるために開催されている会議、議論）

- ・ XML（データの記述の標準化）