

コンピュータ概論B

— ソフトウェアを中心に —

#09 データベース (後編)

京都産業大学
安田豊

データベースとは

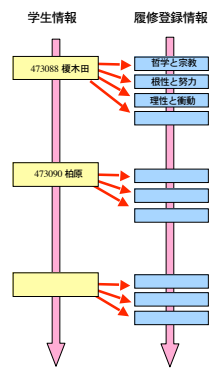
- 教科書 pp.103-
- 外見
 - データを決まった形式 (フォーマット) で整理し蓄積したもの
 - レコード (Record) の存在
 - オブジェクト指向データベースのように決まったデータ型を用意しないタイプもある (例外は常にある)
- 目的
 - 入力・更新
 - 高速な検索、再利用
- 内部構造
 - DBMS の仲介によってデータの一貫性保持と保護を実現

種類

- データモデルに適したタイプ
- カード型
 - 図書館蔵書カードのような一件一枚のもの
- ネットワーク(型)データベース
 - データの親子関係に注目
- リレーショナル(型)データベース
 - Relational Database
 - データの関係 (relation) に注目
 - 現在もっとも良く使われている
- 学生情報データベースを考える

ネットワーク型データベース

- データの親子関係に注目
- 利点
 - よく適合する用途には非常に高速 (検索処理が実質不要)
- 欠点
 - 柔軟なデータ構成がとれない (設計時に完全に決定しておかなければならない)
 - 適合しない用途が後から現れても非常に効率を下がる
 - 例: 三回生の多い履修科目はどれか?
- 銀行、業務システムなど変化の少ない用途には向いている



関係データベース

- 特徴
 - 数学的に完成したモデルがあった
 - Codd (1970, IBM) が理論的モデルを提唱
 - データを表組みで表現
 - 表と表の関係処理を集合演算モデルで定義
- RDB の完成
 - 1973 の SystemR (IBM), Ingress (UCB パークレー校)
 - 1979 Oracle
 - SQL の発明 (1986, ANSI 標準となる)
 - 現在もっとも市場で多く使われているタイプ

RDBMS

- RDB の DBMS
 - テーブル、項目の管理
 - ログ管理
 - アクセス制限
- 特徴
 - SQL 問い合わせ言語によるアクセス
 - 簡易問い合わせシステムがついている
 - プログラムからもSQLでアクセス

RDBにおける表

- データは表形式
 - 行と列による表現
 - 多様なデータを表と項目の関係で記述
- 学生情報で一人分
 - 学生レコード一行
 - 学費レコード一行
 - 履修登録レコード複数行

GNO	NAME	GAKUBU	GAKUNEN
473088	'()	E	2
859674	!*#S%	B	1

GNO	GAKUHI	SIHARAI
473088	1223000	643000
859674	1200000	1200000

GNO	KAMOKU	UNIT
473088	*+.-.+	4
473088	2345	2
473088	./01	4
859674	*+.-.+	4

RDBにおける演算

- 集合と見なして演算
- 部分集合
 - GNOが473088の行を抜く
 - GNOとGAKUBUだけを取り出す

GNO	NAME	GAKUBU	GAKUNEN
473088	'()	E	2
859674	!*#S%	B	1

GNO	NAME	GAKUBU	GAKUNEN
473088	'()	E	2

GNO	GAKUBU
473088	E
859674	B

RDBにおける演算

- 足す (集合和)
- 同じ項目名の列をそのままくわえる

GNO	NAME	GAKUBU	GAKUNEN
473088	'()	E	2
859674	!*#S%	B	1

+

GNO	NAME	GAKUBU	GAKUNEN
785412	9::)	E	1
325698	67(8	J	3

↓

GNO	NAME	GAKUBU	GAKUNEN
473088	'()	E	2
859674	!*#S%	B	1
785412	9::)	E	1
325698	67(8	J	3

RDBにおける演算

- 表どうしを結ぶ
 - 共通の項目(key)で付き合わせ
 - JOIN
 - キーによる突き合わせ

GNO	NAME	GAKUBU	GAKUNEN
473088	'()	E	2
859674	!*#S%	B	1

GNO	GAKUHI	SIHARAI
473088	1223000	643000
859674	1200000	1200000

GNO	NAME	GAKUBU	GAKUNEN	GAKUHI	SIHARAI
473088	'()	E	2	1223000	643000
859674	!*#S%	B	1	1200000	1200000

SQL

- Full Spec 無し(略語ではない)
 - 元はあったが今は SQL として仕様化
- 集合演算をプログラミング言語風に簡略化
 - SELECT 一つで殆どの処理を行う
- 選択
 - SELECT * FROM GAKUSEI WHERE GAKUBU="E"
 - SELECT * FROM GAKUHI WHERE SIHARAI > 600000

SQL

- 選択 (項目抜きだし)
 - SELECT GNO, GAKUBU FROM GAKUSEI
- 突き合わせ
 - SELECT * FROM GAKUSEI, GAKUHI WHERE GAKUSEI.GNO = GAKUHI.GNO
- カウント他
 - SELECT COUNT(*) FROM GAKUSEI WHERE GAKUBU="E"
 - SELECT GNO, GAKUHI, SIHARAI FROM GAKUSEI, GAKUHI WHERE GAKUSEI.GNO = GAKUHI.GNO

関係データベース

- 利点
 - 柔軟、プログラムとデータが独立
 - SQLという問い合わせ言語の便利さ
 - 数学的完全性
- 欠点
 - 概して低速
 - データ格納効率が高くない
- 動かしながら開発したり将来変更が多そうなシステムに向く
- 現在もっとも多く市場で使われているタイプである
 - 欠点をマシン能力でカバーするという考え方
 - 現代的なソフトとハードのバランスの典型例