



Database

第2回

三層スキーマ, データモデル,
データベース設計のための仕組み

上智大学理工学部情報理工学科
高岡詠子



No reproduction or republication without written *permission*.
許可のない転載、再発行を禁止します

Attention

**この教科は
オープンコースウェアとして
収録されます**



Attention

オープンコースウェア(OCW)とは

**大学などの授業, 関連情報を
インターネット上で無償で公開する活動
社会への還元を目的とする**

権利が判別できる場合はぼかし処理が行われる



Attention

**質問や、
名指しされて答える場合は
カットされない**

困る人は申し出てください



Schedule

	日程	内容
第1回	10月6日	ガイダンス, データベースとは?
第2回	10月13日	三層スキーマ, データモデル, データベース設計のための仕組み
第3回	10月20日	概念設計: 概念モデルとERモデル, 論理設計へ
第4回	10月27日	論理モデルと正規化
第5回	11月10日	正規化, 物理設計
第6回	11月17日	物理設計
第7回	11月24日	SQL言語(データベース定義)
第8回	12月1日	SQL言語(データベース操作: 射影制限, 結合, 合併除の各演算)
第9回	12月8日	SQL
第10回	12月15日	SQL言語(ビュー定義など)
第11回	12月22日	データベース管理システム: トランザクション処理
第12回	1月5日	データベース管理システム: 同時実行制御, 排他制御
第13回	1月12日	同時実行制御, 排他制御, デッドロック
第14回	1月19日	データベース技術動向, リレーショナル代数, まとめ



今日の授業

➡ **先週の復習**

➡ **データとは？**

➡ **データベースとは何か？**

➡ **データモデル**

➡ **データベース設計**



アプリケーションとデータベース をなぜ分ける？

▶ データの _____

- ▶ データベースの管理を簡単に
- ▶ 同じデータをアプリケーションごとに別々に作らなくて済む

▶ データの _____ を保つ

- ▶ 組織がデータベースの構成を変えても、アプリケーションソフトウェアに影響を与えない





考えてみよう

データベースって何でしょうか？

▶ Database = Data + Base

▶ **色々なアプリケーションが操作
するデータの集まり**



データベース管理システム

(DBMS:database management system)

- ▶ データベースの維持・運用を行うシステム
- ▶ データベースをユーザに提供し、データの更新、削除、追加などを効率的に行えるような仕組みをもつシステム
- ▶ 代表的なDBMS
 - ▶ オープンソースウェア: PostgreSQL, MySQL
 - ▶ 商用: Oracle, Access



今日の授業

▶先週の復習

▶データとは？

▶データベースとは？

▶データモデル

▶データベース設計

実世界のデータをデータベースに格納してコンピュータで扱えるようにするためにはどうすべきか？

実世界のデータをどう表現するのか？



データモデルとは

▶ 世の中のデータベースをイメージしよう

旅行会社サイト

- ・顧客情報
- ・顧客ごとの旅行予定
- ・日時ごとの予約状況

ネットショッピング

- ・顧客情報
- ・顧客ごとの買い物情報
- ・売りあげ情報

レストラン予約サイト
SNSサイト
etc.....

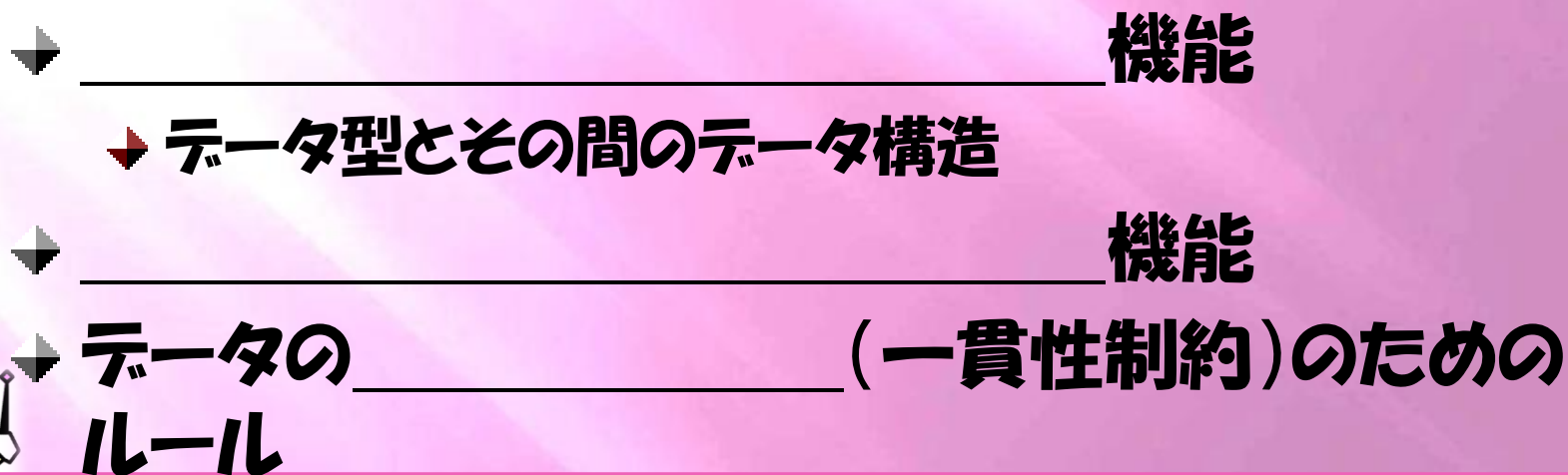
▶ これらのデータベースを構築するときの、 共通基盤となるデータ構造やルール

▶ 具体的なデータではなくて...



データモデル

データモデルの構成



データモデルの歴史

➡ _____ データモデル, ネットワークデータモデル

- ➡ 1960年代に開発
- ➡ 性能を重視しており, 少しわかりにくい

➡ _____ モデル

- ➡ 1970年, E.F.Coddにより提案
- ➡ 現在世の中でもっとも広く使われている
- ➡ わかりやすく, 数学的に扱える

➡ オブジェクト指向データモデル

- ➡ 複雑な構造を持つデータを扱うために1980年代後半から研究されている



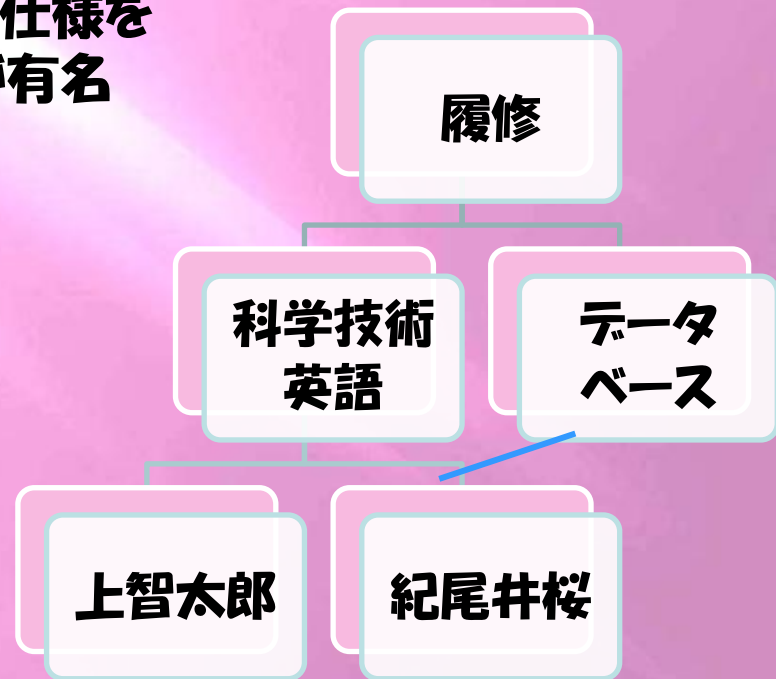
階層型データモデル

- データを木構造で表現したデータモデル
- 実体を親子関係で表現
- もっとも古くからあるデータモデル
- 1968年IBM大型計算機上の
商用データベースシステム
IMSが有名
- 現在もIBMの大型機OSなどの
上で使われている



ネットワーク型データモデル

- データを網構造で表現したデータモデル
- 階層型と同様、親子関係を表示
- 親を複数持てるようになっている
- CODASYL(Conference On Data System Language)
というIT業界団体(COBOLの言語仕様を
決めた委員会)が提案したモデルが有名



リレーショナル(関係) データベース

- ▶ 現在もっとも一般的なデータベース
- ▶ すべてのデータを2次元のテーブル(表)の形であらわす
 - ➡ この表を「_____ (関係)」
とよぶ
- ▶ 各表はDB内の別のテーブルに何らかの方法で関連付けられる
- ▶ _____は複数のタプル(tuple)から構成されている



「レージョン／表／テーブル

StudentListという

Student Number	Name
A0102345	Eiko Takaoka
S0102340	Maria Koda
B2100998	Taro Jochi

D_1

D_2

$D_1, D_2, \dots, D_n$ をドメインとするとき、
 $D_1, D_2, \dots, D_n$ の「レージョン」とは、
直積(各要素のすべての組み合わせ)

$D_1 \times D_2 \times \dots \times D_n$ の任意の
有限部分集合をいう。

$$R \subseteq D_1 \times D_2 \times \dots \times D_n$$

$D_1 = \{A0102345, S0102340, B2100998\}$

$D_2 = \{Eiko Takaoka, Maria Koda, Taro Jochi\}$

$D_1 \times D_2 = \{$
 (A0102345, Eiko Takaoka)
 (A0102345, Maria Koda)
 (A0102345, Taro Jochi)
 (S0102340, Eiko Takaoka)
 (S0102340, Maria Koda)
 (S0102340, Taro Jochi)
 (B2100998, Eiko Takaoka)
 (B2100998, Maria Koda)
 (B2100998, Taro Jochi)
 $\}$



リレーション

➡ $D_1, D_2, \dots, D_n$ を _____ とするとき、
 $D_1, D_1, \dots, D_n$ の _____ とは、
直積(各要素のすべての組み合わせ)

➡ $D_1 \times D_2 \times \dots \times D_n$ の任意の

➡ **有限部分集合**をいう。

➡ $R \subseteq D_1 \times D_2 \times \dots \times D_n$

➡ **関係 R は定義域 $D_1, D_2, \dots, D_n$ の
直積の部分集合**



直積の例

- ➡ $D_1 = \{\text{フレーン, メープルシナモン}\}$
- ➡ $D_2 = \{\text{スモークサーモン, ターキー, クリームチーズサンド, B.L.T.}\}$
- ➡ $D_1 \times D_2 =$



関係データベースモデルの理論的背景

➡ 集合での二項関係

➡ 二つの集合 A, B に対して、 A と B との間の二項関係とは、直積 $A \times B$ の部分集合

➡ 関係データベースモデルの理論的背景となっているのは数学の「集合論」



リレーション／表／テーブル

StudentListという

Student Number	Name
A0102345	Eiko Takaoka
S0102340	Maria Koda
B2100998	Jochi Taro

D_1

D_2

濃度: Rの組の総数

次数: ドメインの個数

$D_1 \times D_2 \times \dots \times D_n$ の n

次数1: 単項リレーション

次数2: 2項リレーション

...

次数 n : n 項リレーション

左の表は濃度 , 次数



リレーション／表／テーブル

StudentListという

Student Number	Name
A0102345	Eiko Takaoka
S0102340	Maria Koda
B2100998	Jochi Taro

属性名

濃度: Rの組の総数

次数: ドメインの個数

$D_1 \times D_2 \times \dots \times D_n$ の n

次数1: 単項リレーション

次数2: 2項リレーション

...

次数n: n項リレーション

左の表は濃度 , 次数



リレーショナルモデルの用語

リレーショナルモデル	DBMS
	表 (テーブル)
組 (タプル tuple)	
属性 (アトリビュート attribute)	
	行数
	列数
定義域 (ドメイン domain)	列の値の範囲



この場合のリレーションは
集合理論的な関係 (リレーション) のこと

理論的背景となっている「集合論」からきている用語

ERモデル entity-relationship modelのrelationship(関連)とは別物！！



データモデルとデータモデリング は違う！！

▶ データモデル:

▶ データモデリング

▶ 実際の業務(売上管理, 履修登録)を行うためのシステムを構築するときに_____を使って, 売上管理や履修登録をモデル化する

▶ 何もないところからDBを構築するまでの手順

▶ _____によってできたものが_____.



データモデリングを プログラミングに例える

を使って

すると

ができる



速水治夫:「リレーショナルデータベースの実践的基礎、コロナ社(2008)。

2011/10/13

©2011 Eiko Takaoka All Rights Reserved.

今日の授業

➡先週の復習

➡データとは？

➡データベースとは何か？

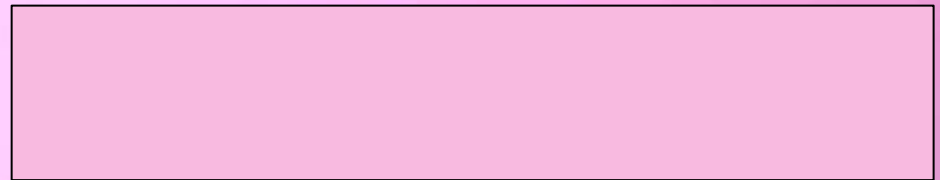
➡データモデル

➡データベース設計のための仕組み

データベース設計：



スキーマ



FacultyList

FacID	Faculty Name
100	Sci. and Tech.
200	Liberal Arts

各リレーションとそれらが
つながった体系

SubjectList

SubjectID	Subject Name	Instructor
SCT50601	科学技術英語	高岡詠子
YJO20300	イスパニア語初級	Stephen Edger
ISP10001	キリスト教の歴史	川村あさひ
SIC20200	情報科教育法	桜井有
SCT63700	情報理工学演習	渡辺さつき
AR70100	データベース	松岡修二
INTRO200	INTRODUCTION	Maria P. Harrison
AR70100	ART HISTORY	

履修登録システムの スキーマ

StudentList

Student Number	Name	FacID
A0102345	Eiko Takaoka	100
S0102340	Maria Koda	200

RegistrationList

Student Number	SubjectID	Grade
A0102345	SCT50601	A
A0102345	YJO20300	A
S0102340	ISP10001	B
S0102340	YJO20300	B



履修登録システム

学内掲示版システム

メールシステム

スキーマ(sub scheme)

- ☆アプリケーションが実際に利用するデータ項目,
データの範囲を定義したもの
- ☆ユーザ(アプリケーション)側から見たDBの構造のこと

概念スキーマが変更されても
各アプリケーションには
影響を及ぼさない

論理性データ独立

スキーマ

- ☆対象実世界全体をデータモデリングしたスキーマ

Theology

物理性データ独立

スキーマ(scheme)

- ☆概念スキーマを物理的に実装するための記述
- ☆記憶装置上におけるデータの内部表現方式
- ☆データの格納形式(フラット構造か、階層構造化など)
- ☆データ型やデータの大きさ、ファイルサイズなど

記憶方式が変更されても概念
スキーマや外部スキーマに影
響を及ぼさない

3層スキーマ

- ➡ 効率的に大量データを処理するための構造
- ➡ データベースの構造を3層にわけた考え方
- ➡ データ構造を独立性という観点でどの部分で分離するかという定義

____スキーマ

____スキーマ

____スキーマ

____性データ独立

____性データ独立



関係データベースの設計手順

業務分析

対象とする業務実態を客観的に把握する
業務の内容、処理方式等を分析結果としてまとめる

設計

実業務の中から必要なデータのあつまりとそれらの関係を洗い出し概念モデルを作成
ER図を使う。

設計

概念設計によって作成された概念モデルを、特定のデータモデルに対応した論理モデルに変換。
システムとしてコンピュータに保存しておくデータの形を決める。正規化を行ったり表に対する制約の定義を行う

設計

データベースとしての性能について考慮する。
論理設計で正規化した表の定義を崩したりインデックスを定義したりして性能を向上させる



データモデリング

- ➡ データ構造を抽象化という観点からとらえた定義
- ➡ データベースの対象実世界を抽象的に記述して操作する体系
- ➡ 何もないところからDBを構築するまでの手順

作成すべきデータの定義

____設計

概念モデル

____設計

論理モデル

____設計

物理モデル

実業務の中から必要なデータのあつまいとそれらの関係を洗い出し、まとめたもの。ERモデルを用いる。まだシステム上のデータは意識しなくてもよい。

システムとしてコンピュータに保存しておくデータの形を定めたもの。まだDB製品は意識しなくてもよい。

ハードウェアやDB製品を含む実装方法の定義



データモデリングと3層スキーマ

対象世界
企業全体

① ____ 設計

概念モデル

② ____ 設計

論理モデル

③ ____ 設計

物理モデル

____ スキーマ

____ 性データ独立

____ スキーマ

____ 性データ独立

____ スキーマ

データベース設計：

