

# データベース

## 【2: データモデリング】

石川 佳治



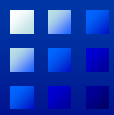
# データモデル

- 三つの構成要素
  - データ構造を記述するための規約
  - データ操作の体系
  - 整合性制約の表現の仕組み
- データモデルの役割
  - DBMSのインタフェース
    - DBMSの内部の詳細とは独立な論理的レベルでのデータの記述と操作
  - 実世界のモデル化ツール
    - 実世界の情報を取捨選択して構造化する枠組み
    - **データモデリング** (data modeling) : モデル化の過程



# モデリング

- 与えられたモデルの枠組みで対象を表現するために、対象からの情報の抽出、抽象化、簡略化、変換などを行う行為
- 例
  - コンピュータグラフィックス：対象（例：人間）をコンピュータ内でCG表現
  - 気象予測のための気象モデリング
  - 交通分析のための交通流モデリング
- モデルに求められる要件
  - 表現能力 (expressive power) : 対象を適切に表現可能
  - 実現可能性 (tractability) : 計算能力などで扱える範囲か
  - トレードオフが存在



# データモデリング

- 対象の実世界のデータを，データモデルを用いてコンピュータ上に表現
- 実世界のデータは多様で複雑
  - 情報の取捨選択
  - データ間の関連，データに対して成り立つ制約条件の抽出
- データモデルに求められる要件
  - 表現能力：対象データをうまく表せるか
  - 実現可能性：コンピュータ上に効率的に実装できるか



# 代表的データモデル(1)

- **リレーショナルデータモデル** (関係データモデル, relational data model)
  - 1970年にCoddにより提案
  - 理論的基盤があり, 明快
  - 物理レベルからの独立性
  - データベースをリレーション(関係)の集まりとしてモデル化
  - 形式的な操作体系
    - リレーショナル代数(関係代数)
    - リレーショナル論理(関係論理)



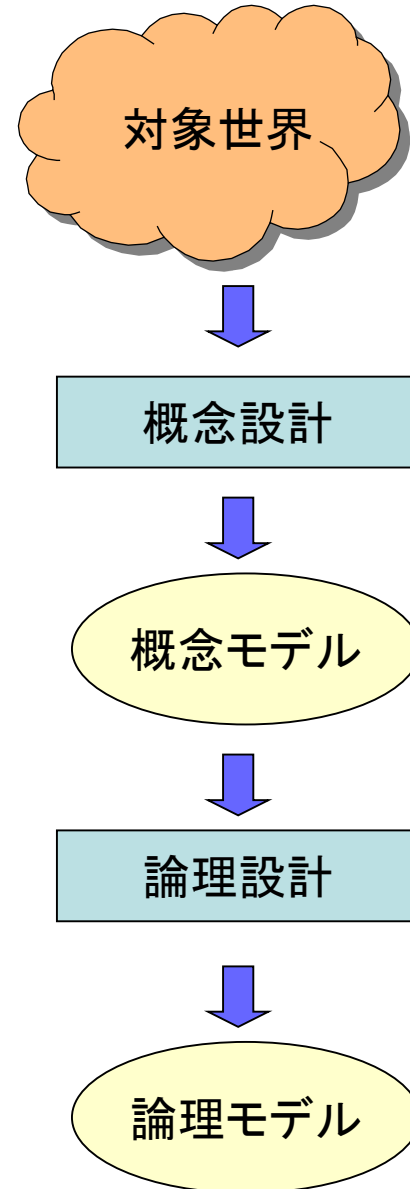
## 代表的データモデル(2)

- 古典的なデータモデル
  - ネットワークデータモデル(network data model)
  - 階層データモデル(hierarchical data model)
- **オブジェクト指向データモデル**(object-oriented data model)
  - オブジェクト指向の概念を取り入れる
  - 1980年代から90年代にかけて研究開発
  - **オブジェクトリレーショナルDBMS**(ORDBMS)
    - RDBMSにオブジェクト指向の機能(例:継承, 配列, 集合, ユーザ定義データ型)を取り込む



# 実世界のデータモデリング

- **概念設計** (conceptual design)
  - DBMSとは独立に, 対象世界を論理的に記述
  - **概念モデル**を利用
- **論理設計** (logical design)
  - DBMSが提供するデータモデルへと変換
  - **論理モデル**を利用
    - リレーショナルデータモデルなど





# 実体関連モデル(1)

- **実体関連モデル** (entity relationship model)
  - しばしば**ERモデル**と呼ばれる
  - 概念設計のための代表的データモデル
  - Chenにより提案(1976)
  - その後さまざまな拡張
- **二つの概念**
  - **実体** (entity)
    - モデリングの対象(例: 学生, 商品)を表現
  - **関連** (relationship)
    - 複数の実体の相互関係をモデル化





# 実体関連モデル(2)

- 実体

- 実体集合 (entity set) : 同一種類の実体の集まり
- 属性 (attribute) : 実体の性質を記述 (例: 氏名, 価格)

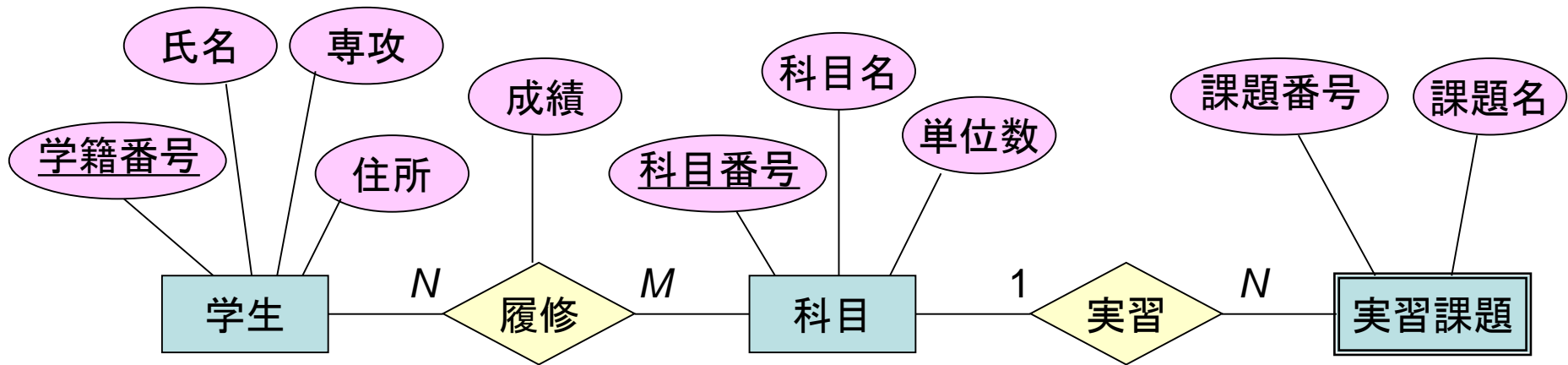
- 関連

- 関連集合 (relationship set) : 実体集合  $E_1, \dots, E_n$  に対し,  $R \subseteq E_1 \times \dots \times E_n$  である  $R$ 
  - $R$ の要素が関連に相当
  - $n$ : 関連集合の次数 (degree)
- 関連集合にも属性を付与可能



# 実体関連図(1)

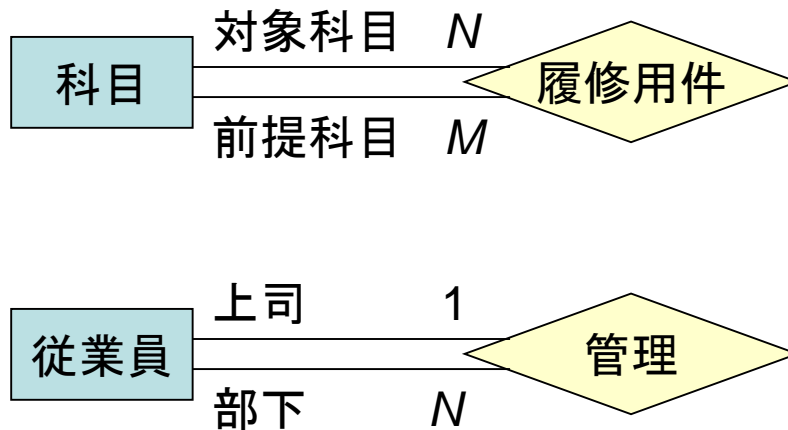
- entity relationship diagram
- 実体関連モデルでのモデル化を図式表現
  - 実体集合: 矩形
  - 関連集合: ひし形
  - 属性: 楕円





## 実体関連図(2)

- 同一の実体集合に対する複数の関連集合も記述可能
- **役割**(role)を明記





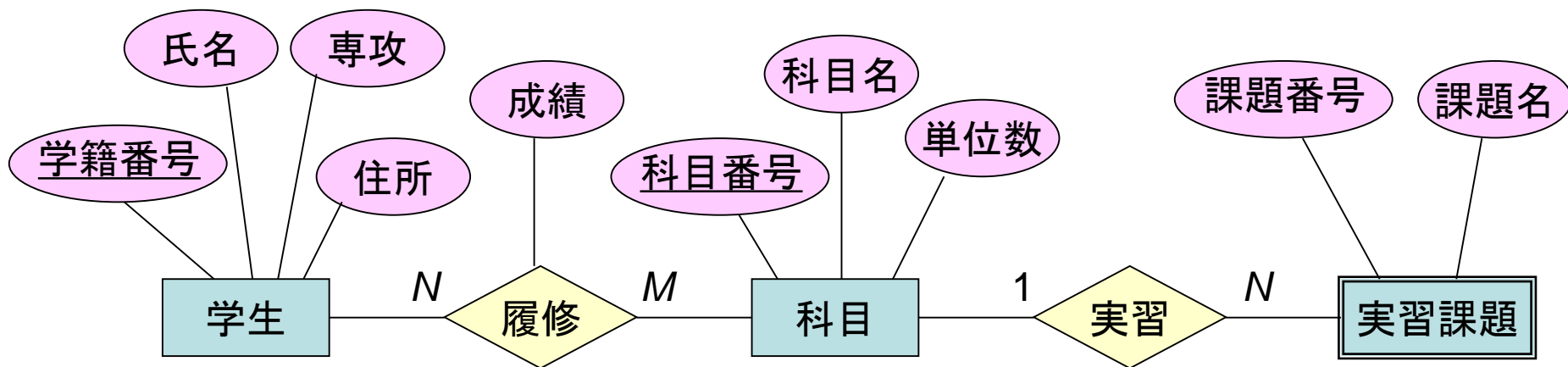
## 実体関連モデル: 整合性制約 (2)

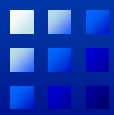
- **キー制約** (key constraint)
  - 重要な整合性制約
- **キー** (key)
  - 実体集合の個々の実体を一意 (unique) に識別できる属性もしくは属性の組合せで極小なもの
  - 例: 学生の学籍番号, 学生の氏名と住所
  - キーは複数個存在しうる
    - ⇒ 各キーを候補キー (candidate key) と呼ぶ
  - **主キー** (primary key): 候補キーのうちで主たるキーとして用いたいもの



# 実体関連モデル: 弱実体集合

- 実体関連図では, 主キーに下線を引く
- **弱実体集合** (weak entity set)
  - 自身の属性だけでは主キーを構成できない実体集合: 二重枠で表現
  - 例: 「実習課題」は「科目」との結びつきによってのみ識別可能



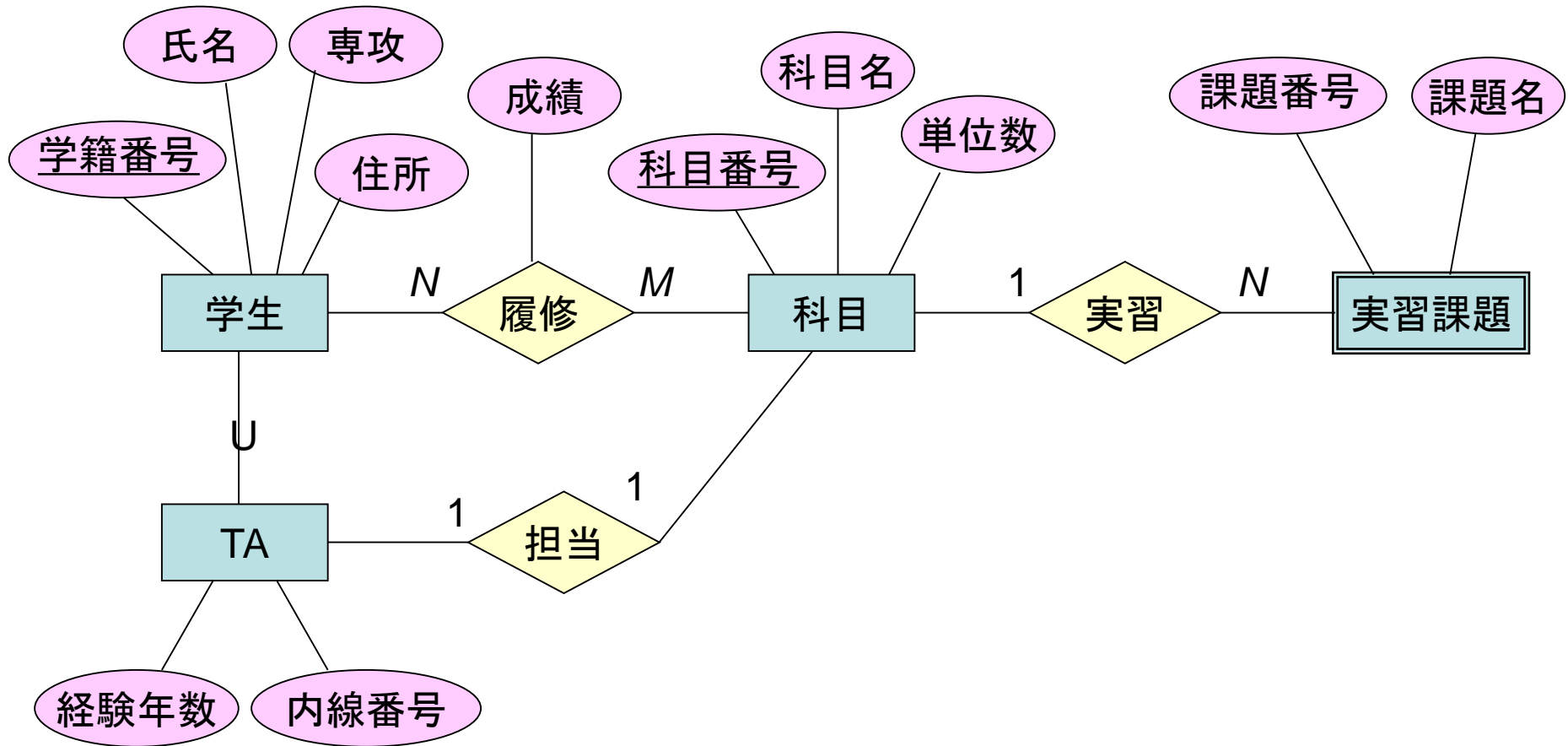


# 実体関連モデル: 汎化階層(1)

- generalization hierarchy
- 汎化 (generalization)  $\Leftrightarrow$  専化 (specialization)
- 実体集合間の**階層的関係**を表現
  - クラス/サブクラスの概念に相当
  - 例: 「学生」と「TA」
  - 「TA」は「学生」の属性を**継承** (inherit) し, 付加的な属性を持つ
- 拡張実体関連モデルで導入された機能



# 実体関連モデル: 汎化階層(2)







# 余談: UML (Unified Modeling Language)

- **オブジェクト指向設計**のための標準的言語
- オブジェクト指向言語 (Java, C++ など) によるアプリケーション設計に利用
  - クラスの構造, 関係, 振舞い
- **比較: 実体関連モデル**
  - リレーショナルデータベースとの親和性が高い
    - 適度な抽象化のレベル
  - **データ中心設計** (Data Oriented Approach, DOA) に向く
  - アプリケーションとデータベースを一括して設計する場合はUMLを用いることもあり