

# データベース設計論

---

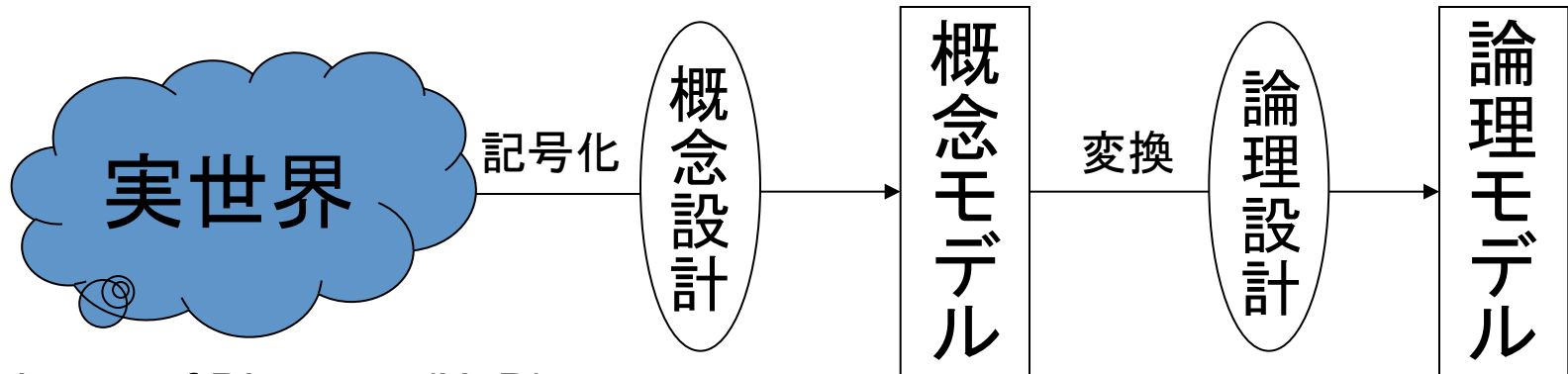
第2回

ERモデルによるデータベース設計

2015年10月13日

# データモデル

- データモデル:
  - データベース中のデータとそれに対する操作を規定する枠組み
    - 実世界の事象をデータベースに写し込む枠組み
    - データベースの操作(検索・更新)を規定する枠組み



Universe of Discourse(UoD):  
対象世界, 対象領域

# データモデルの世代

## 第一世代

ネットワークモデル(network data model)

ファイルシステムの高度化を目的に, 1971年, CODASYLが定義.

階層データモデル(hierarchical data model)

レコード型を基本にしたデータモデル.

## 第二世代

リレーショナルデータモデル (relational data model)

集合論に基づいたテーブル型のデータモデル.

1970年にDr. Coddが提案. 現在広く一般に使われている.

## 第三世代

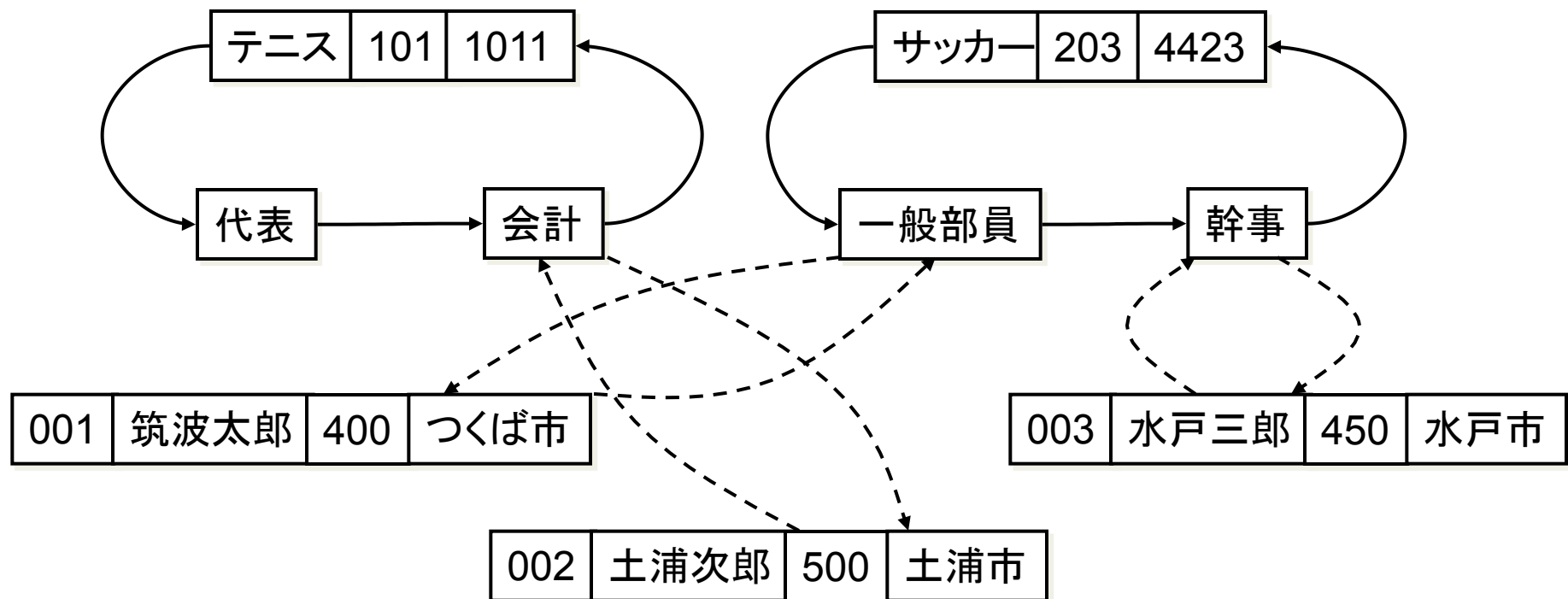
オブジェクト指向データモデル  
(object-oriented data model)

オブジェクト指向モデルに基づいたデータモデル.

多様に形を変えて浸透.

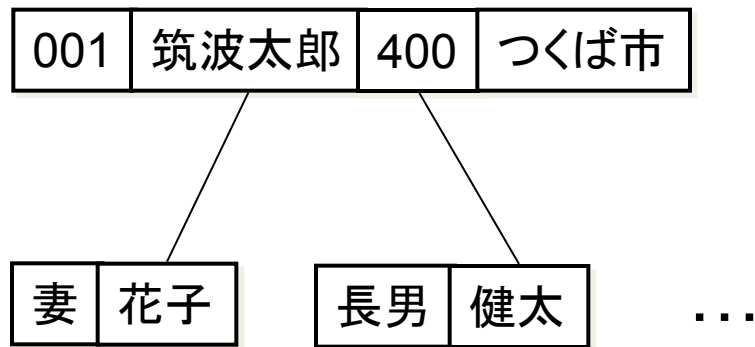
# ネットワークデータモデル

- レコード／レコード型
- レコード型間の親子関係
  - グラフ構造を記述



# 階層データモデル

- レコード型をノードとする木構造
  - 親は一つだけ(1対多関係)
    - ネットワークモデルでは, 複数の親を持てる
  - 複雑な関係の記述が困難
    - 多対多
    - ...



# リレーショナルデータモデル

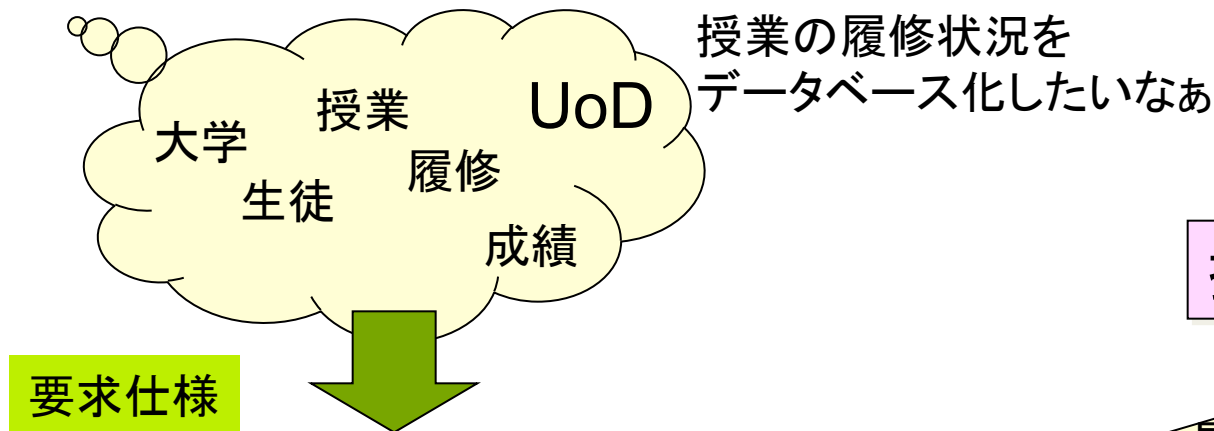
- 集合論に基づいた表形式のモデル
- 関係代数演算を用いて問合せを行う
- 論理モデルが実際のストレージの物理構造と独立  
→ データ独立性

sid	name	age
53666	Jones	18
53688	Smith	18
53650	Smith	19
53831	Madayan	11
53832	Guldu	12

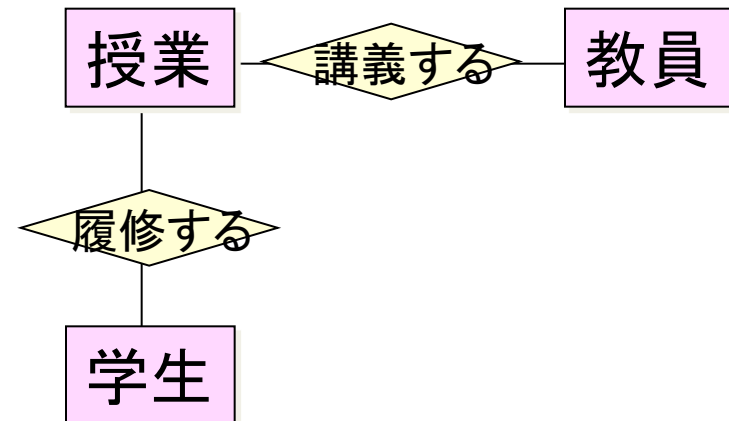
問合せ: 15歳以下の学生の名前

# 概念モデル

- UoDをデータベース設計の要求仕様として書き出し、それを記号化したもの
- ERモデルやUMLが一般的
  - 本授業ではERモデルを教えます

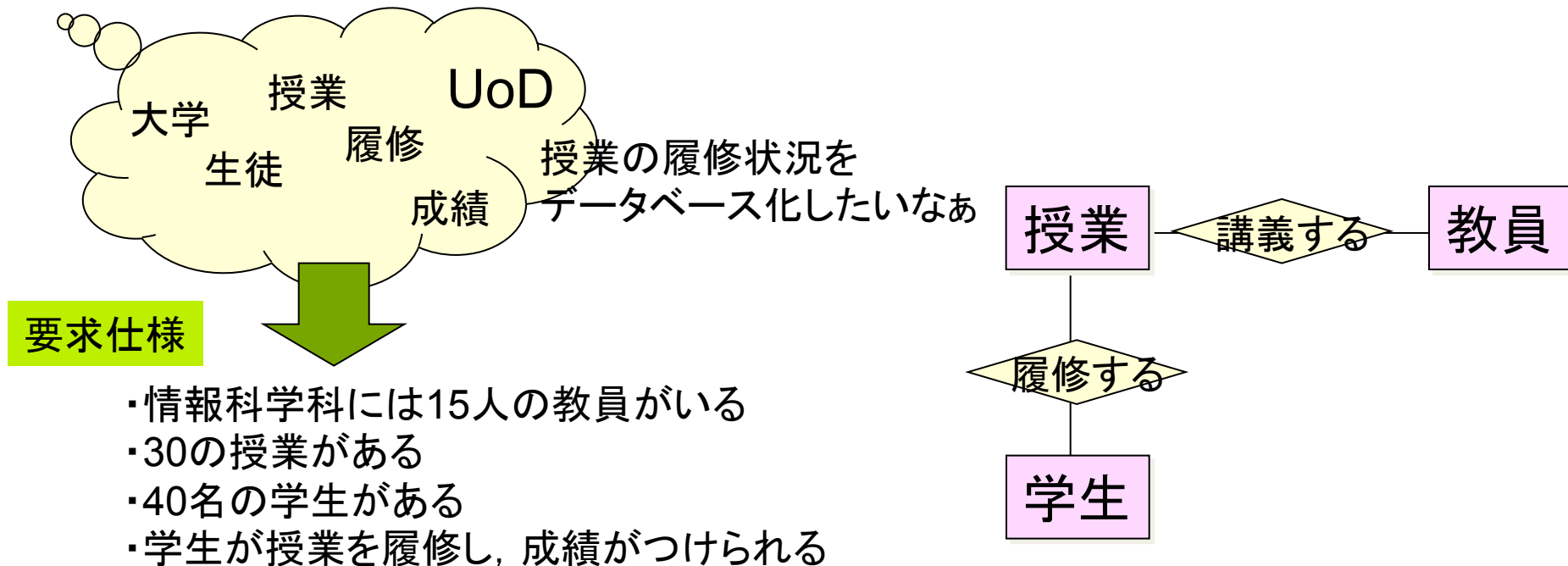


- 情報科学科には15人の教員がいる
- 30の授業がある
- 40名の学生がある
- 学生が授業を履修し、成績がつけられる



# 概念設計の手順

- UoDをデータベース設計の要求仕様として文章に書き出す
- 要求仕様をもとに概念モデルを作成する



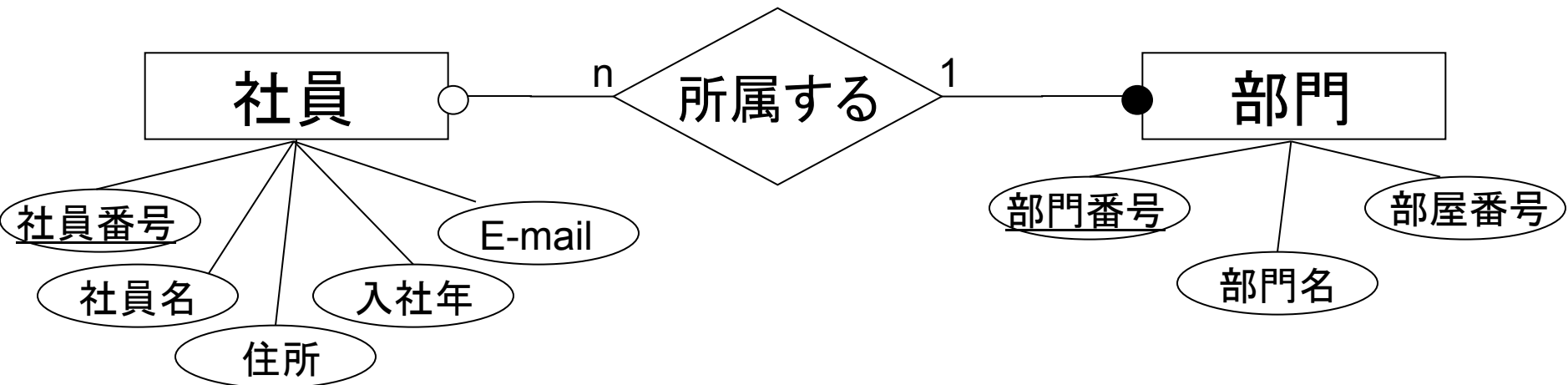


# ERダイアグラム

- Entity-Relation Diagram(実体-関連図)
- データベース化したい現実の世界(Universe of Discourse)を, EntityとRelationによって表現する

## エンティティ

- 分析の対象となるもの
- 人, もの, 場所, 事象, 情報, 概念
- リレーション
  - エンティティ間の関連



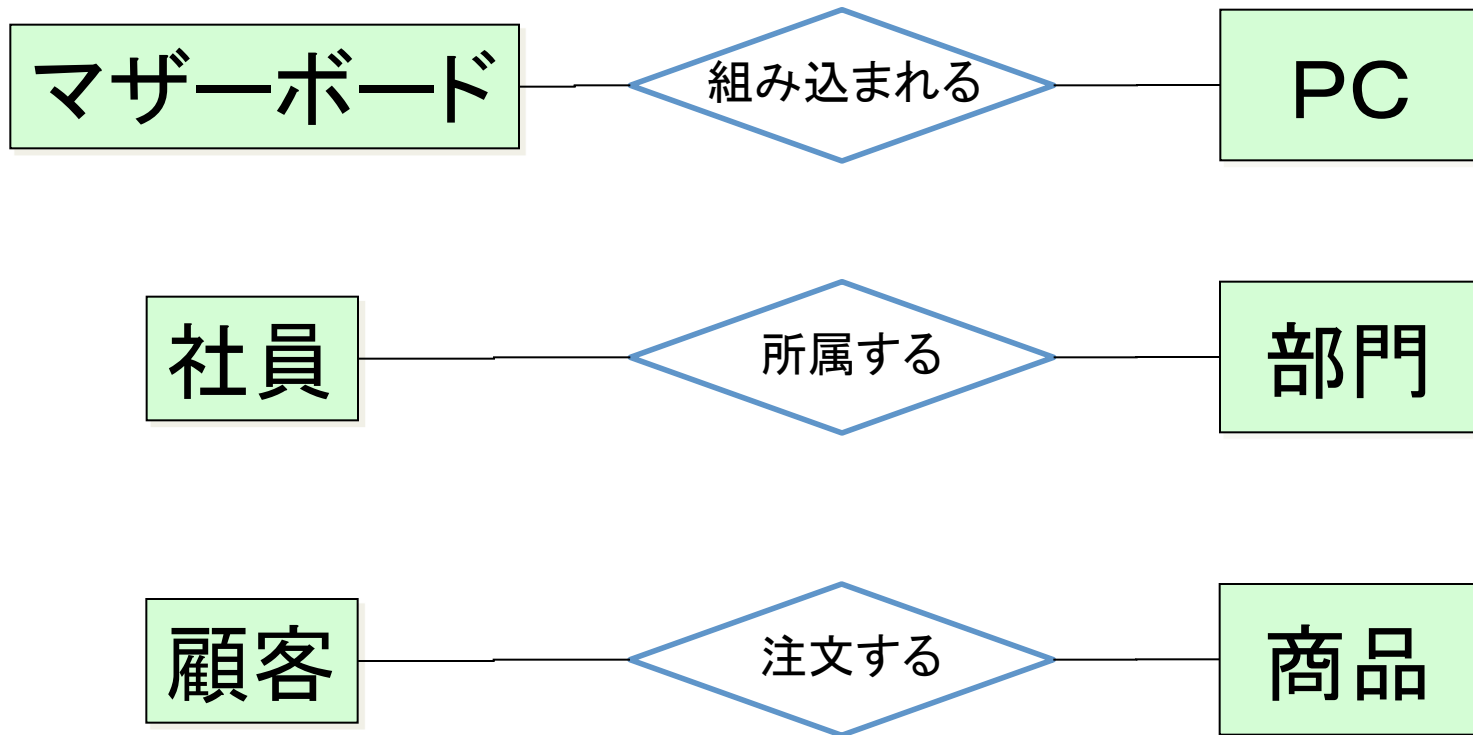
# エンティティ

- リソース系:いわゆる「もの」
  - 人, 物, 場所のような物理的に存在するもの
  - 概念的なもの
- イベント系:いわゆる「こと」
  - 出来事

リソース系	物理的に存在するもの	人	顧客, 社員, 会員, 受講生...
		物	製品, 書籍, 部品, 材料...
		場所	会場, 教室, 会議室, 倉庫...
	概念的なもの		組織, 部門, 予算, 実績...
イベント系	出来事		購入, 受注, 販売, 仕入れ...

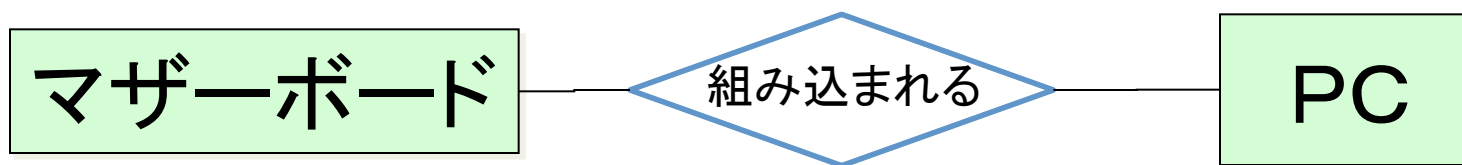
# リレーションシップ

- カーディナリティ: 何対何の関係か
  - 1対1, 1対多, 多対多



# リレーションシップ

- オプションナリティ
  - 必須: 必ず1つ以上存在しなければならない
  - 任意: 存在しない場合もある



# 概念設計の手順

- 要求仕様を洗い出す
  - データベース化したい実世界 (Universe of Discourse) をとにかく文章化する

大学の授業の履修データベースを作りたい

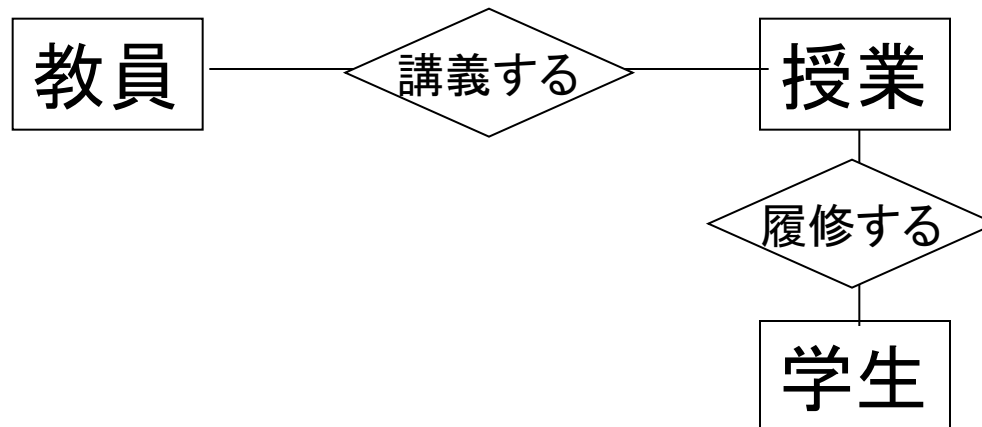
## 要求仕様

- 教員が授業の講義を行う。一人の教員が複数の授業を担当するが、ひとつの授業を複数の教員が担当することはないとする。
- 学生は授業を履修する。一人の学生が複数の授業を取り、その結果成績をもらう。ひとつの授業は複数の学生によって履修される
- 教員データは名前と役職と性別の情報が必要
- 学生データは名前と学科と血液型が必要
- 授業データは授業名と開講する部屋の名前と単位数が必要

# 要求仕様からE-Rダイアグラムを作成する

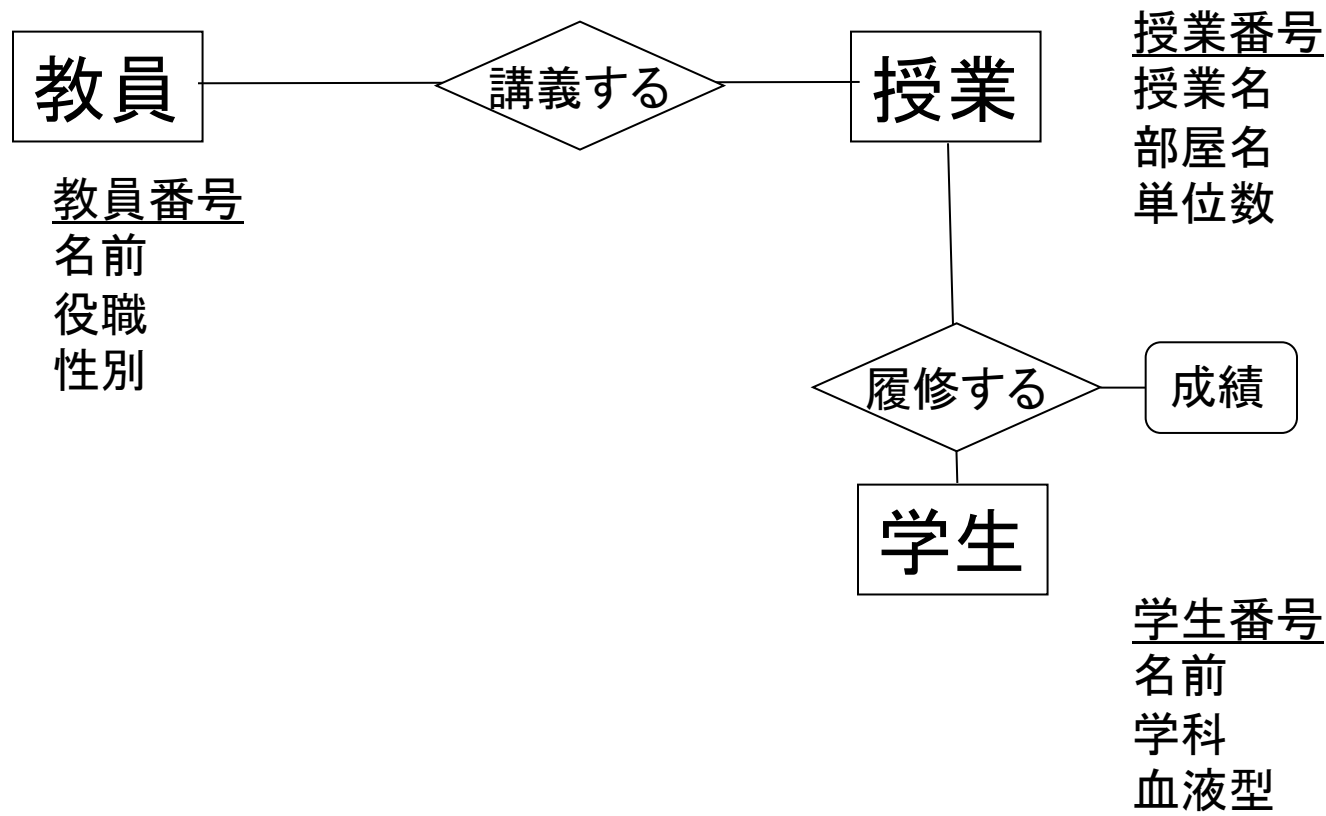
- 名詞と動詞を抜き出す

- ・**教員**が**授業**を**講義する**. 一人の教員が複数の授業を担当するが、ひとつの授業を複数の教員が担当することはないとする.
- ・**学生**は**授業**を**履修する**. 一人の学生が複数の授業を取り、その結果成績をもらう. ひとつの授業は複数の学生によって履修される
- ・教員データは**名前**と**役職**と**性別**の情報が必要
- ・学生データは**名前**と**学科**と**血液型**が必要
- ・授業データは**授業名**と開講する**部屋の名前**と**単位数**が必要



# 属性名を洗い出す

- ・教員データは名前と役職と性別の情報が必要
- ・学生データは名前と学科と血液型が必要
- ・授業データは授業名と開講する部屋の名前と単位数が必要
- ・学生は授業を履修する. 一人の学生が複数の授業を取り, その結果成績をもらう. ひとつの授業は複数の学生によって履修される



# 演習: ER図を作ってみよう

- 演習1: 回転ずしの注文履歴データベースを設計してみよう

座席番号

寿司

- 名前
- 写真
- 値段
- 「わさび入り」



カテゴリ  
(重複あり)

注文リスト

- メニュー
- 個数

画面に含まれない要素

- 注文した時刻



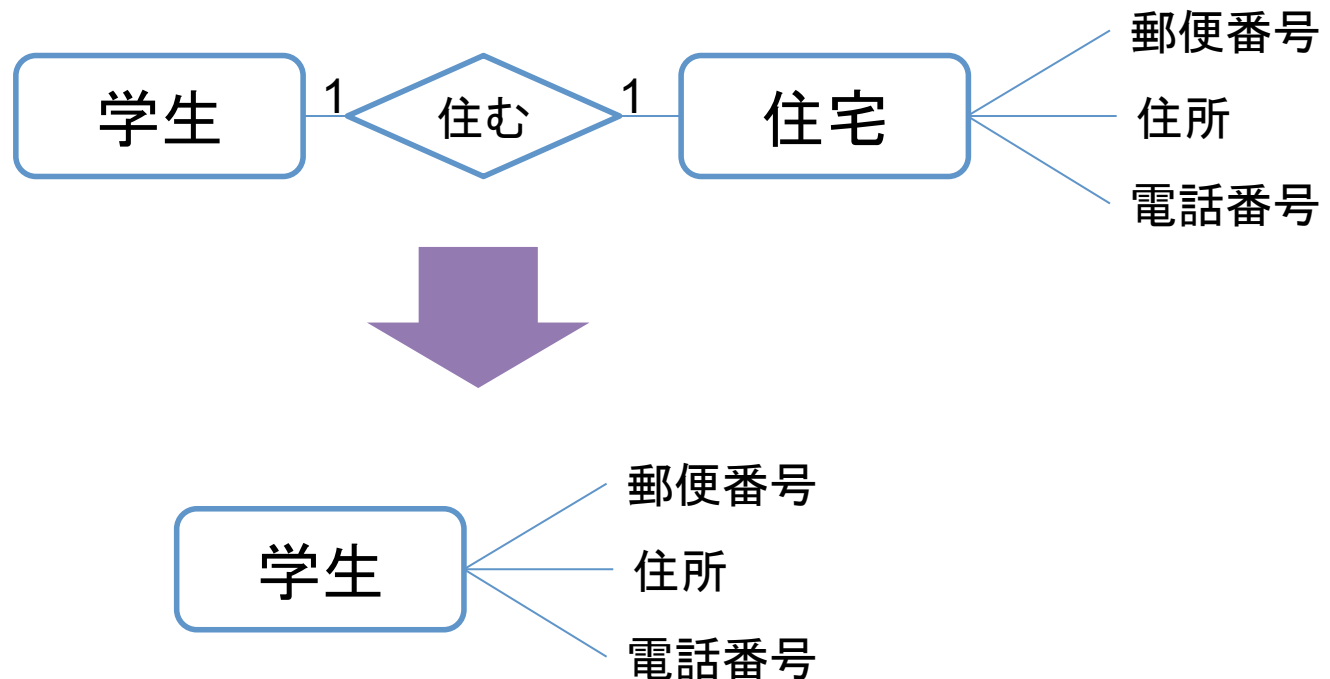
# 演習:ER図を作ってみよう

- 演習2: Twitterの基本機能を使えるようにするためのデータベースを設計してみよう



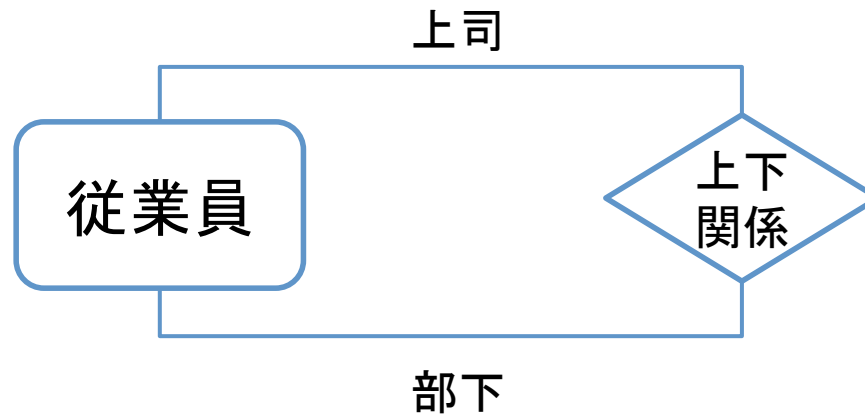
# ER図を作る時のポイント

- できるだけシンプルに
  - unnecessary データを含めない
  - 1to1の関係はエンティティと属性の関係にする



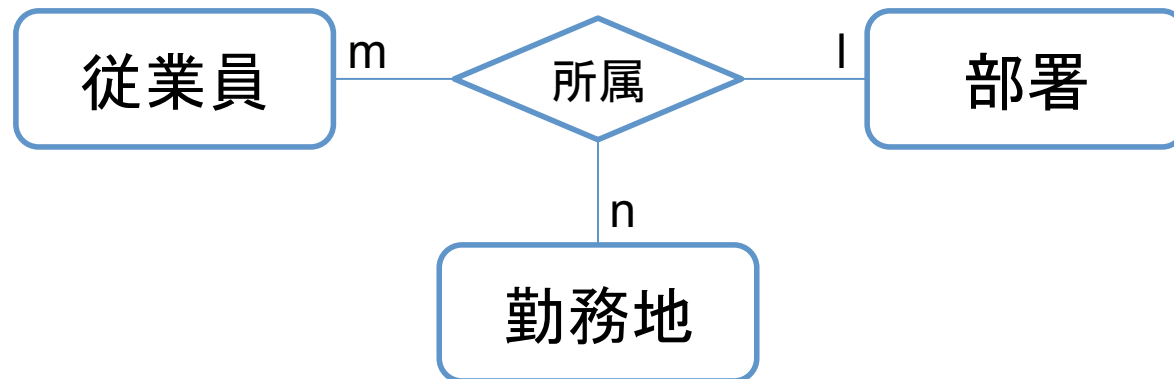
# 自己参照の関係

- 従業員で上司と部下の関係はどう表現する？
- 係長には部下もいるが上司もいる



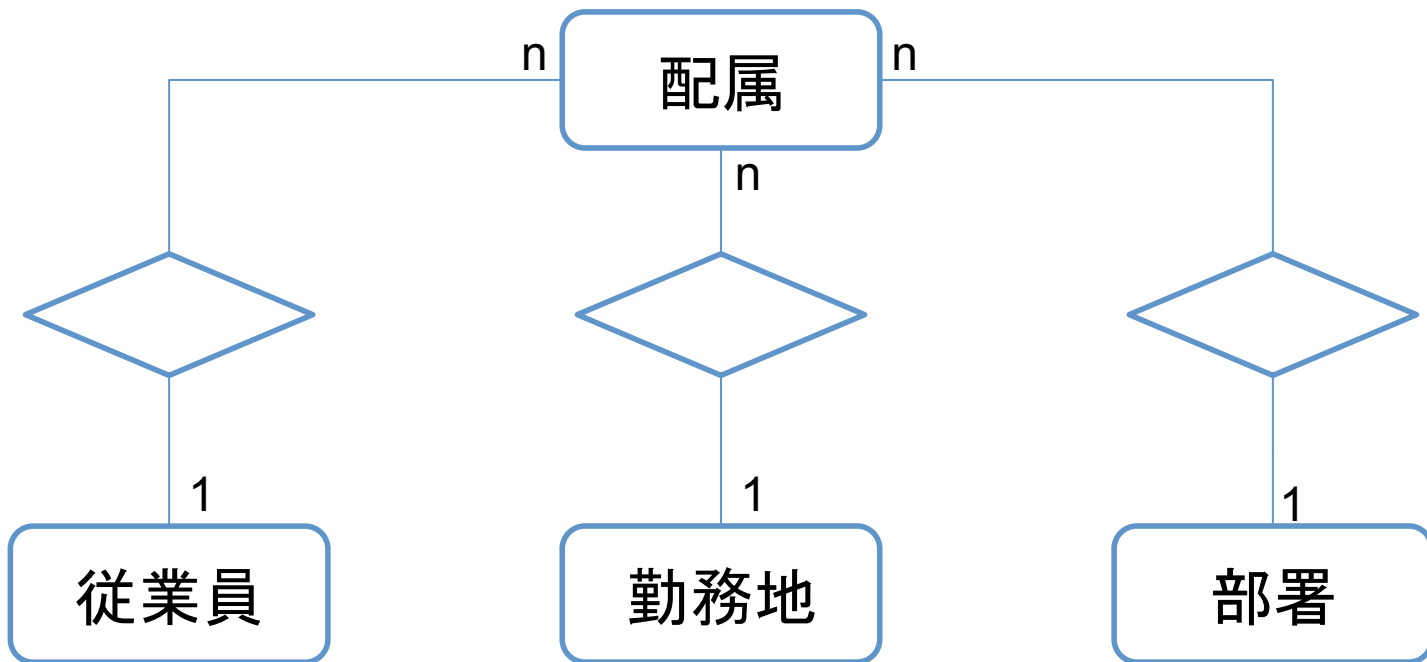
## 三つ組以上の関係

- 作ってもよい
  - ただし、できるだけ作らないように工夫しよう

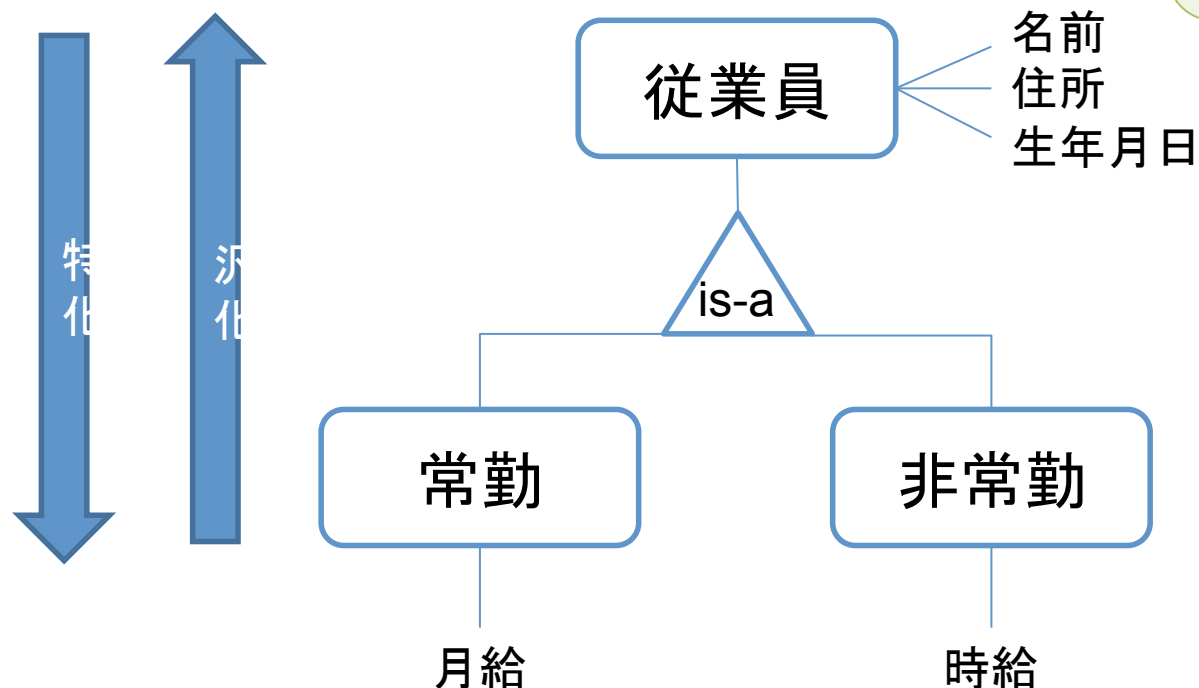


# 三つ組をばらす

- 中心のリレーションをエンティティにする



# 汎化・特化の関係



共通の属性は  
こちらに書く

それぞれのエン  
ティティに特有  
の属性はこちら