データベースシステム
第4回

理工学部情報科学科
松澤　智史

復習

- 正規化（第2回）
  - 異状を防ぐためにテーブル（リレーション）を
    分離して情報を保持する方法

- 関係演算（第3回）
  - 正規化されたテーブル（リレーション）群を任意を形に
    変形・復元する操作・テーブル（リレーション）間の演算

データベースの設計

- ER図
- トップダウンアプローチ
- ボトムアップアプローチ

ER図

- Entity-Relationship Diagram(実体関連図)
- データモデルを記述するための図法
- 構成
  - エンティティ（実体）
    - 四角で表す
  - リレーションシップ（関連）
    - 菱形で表す
  - 属性
    - 楕円で表す
  - 濃度
    - 1やM,N等で表す

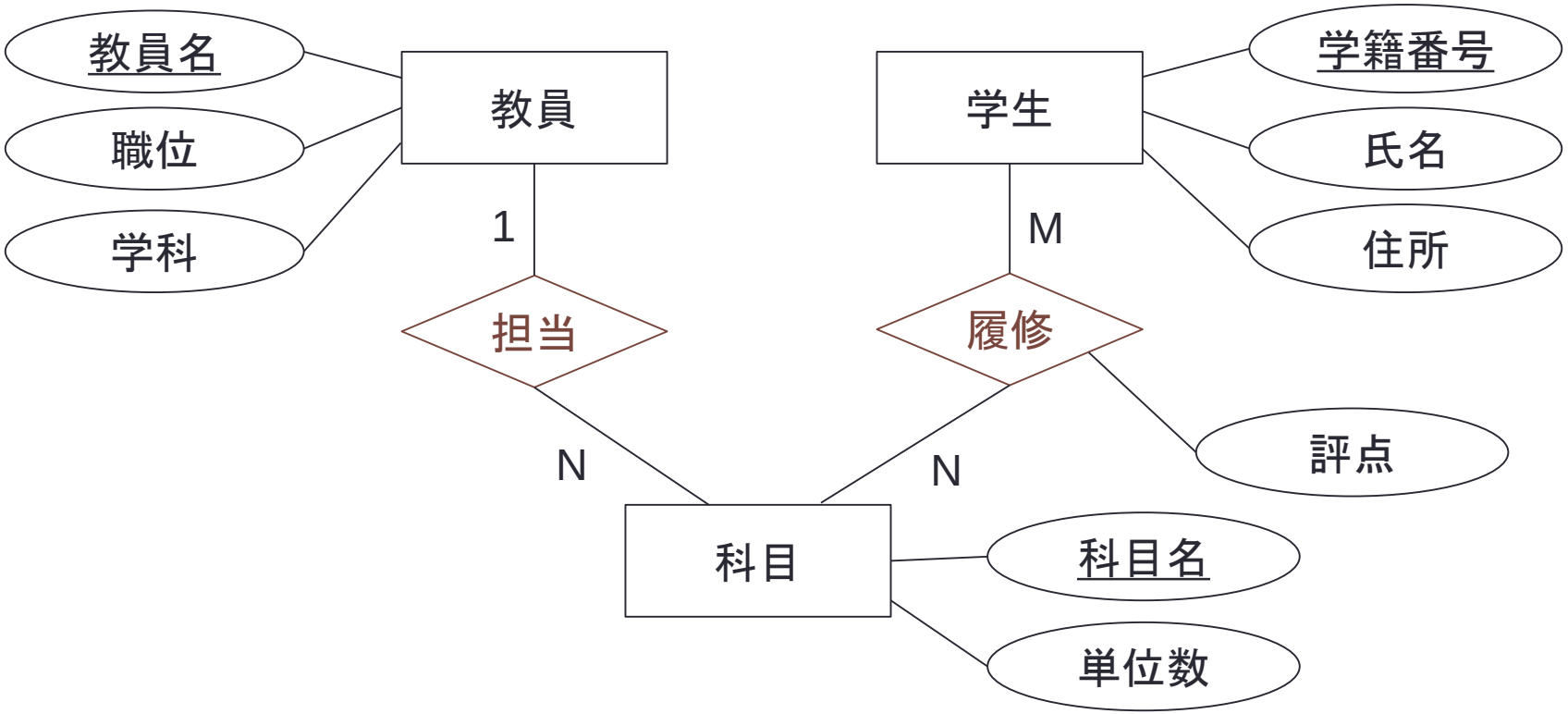
関連型の種類

実体型E1とE2があり，その間に関連型Rが成立していた場合
Rには，関連型の対応関係から以下の種類がある

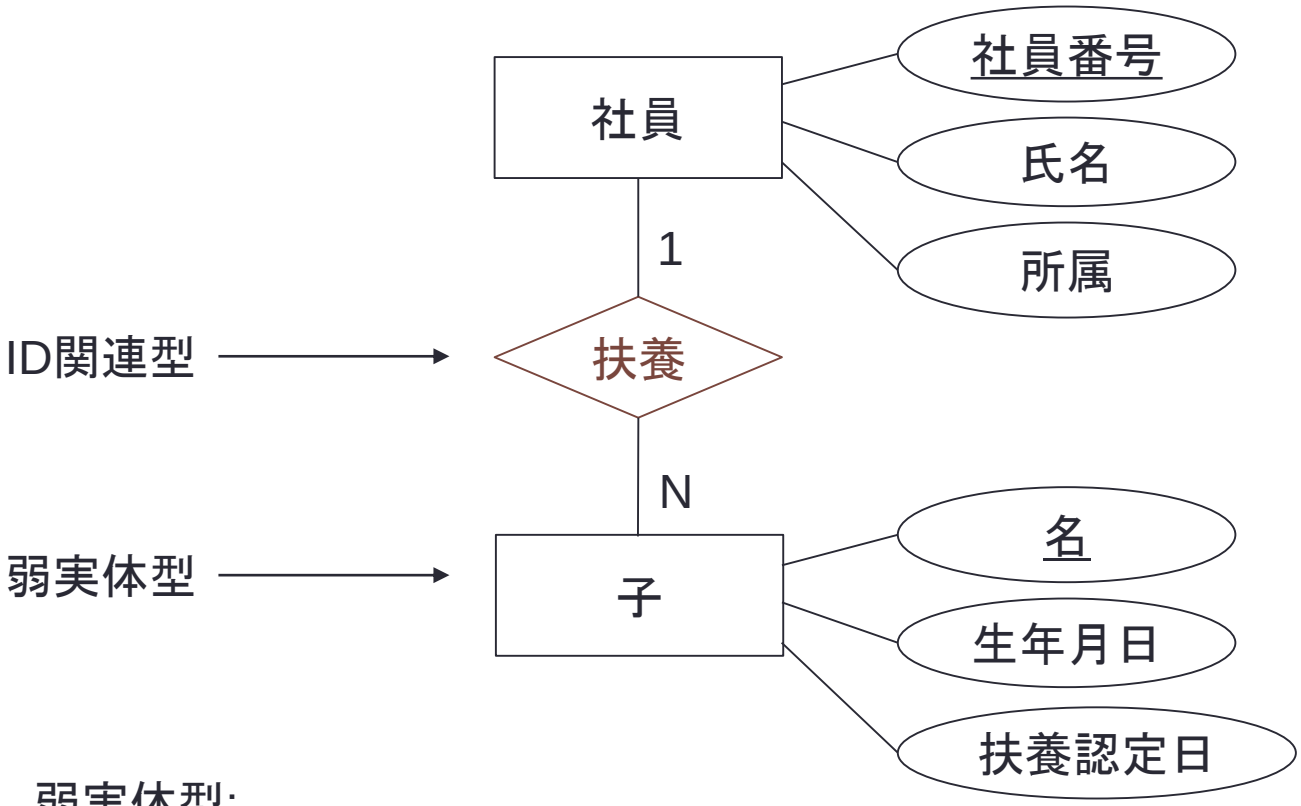
- 1対1関連型
- 1対多関連型
- 多対多関連型

1やM，Nなどと表記するが，この1やM，Nのことを濃度という
  - MはM個という意味ではなく，0以上ならどのような値でも構わない

ER図の例

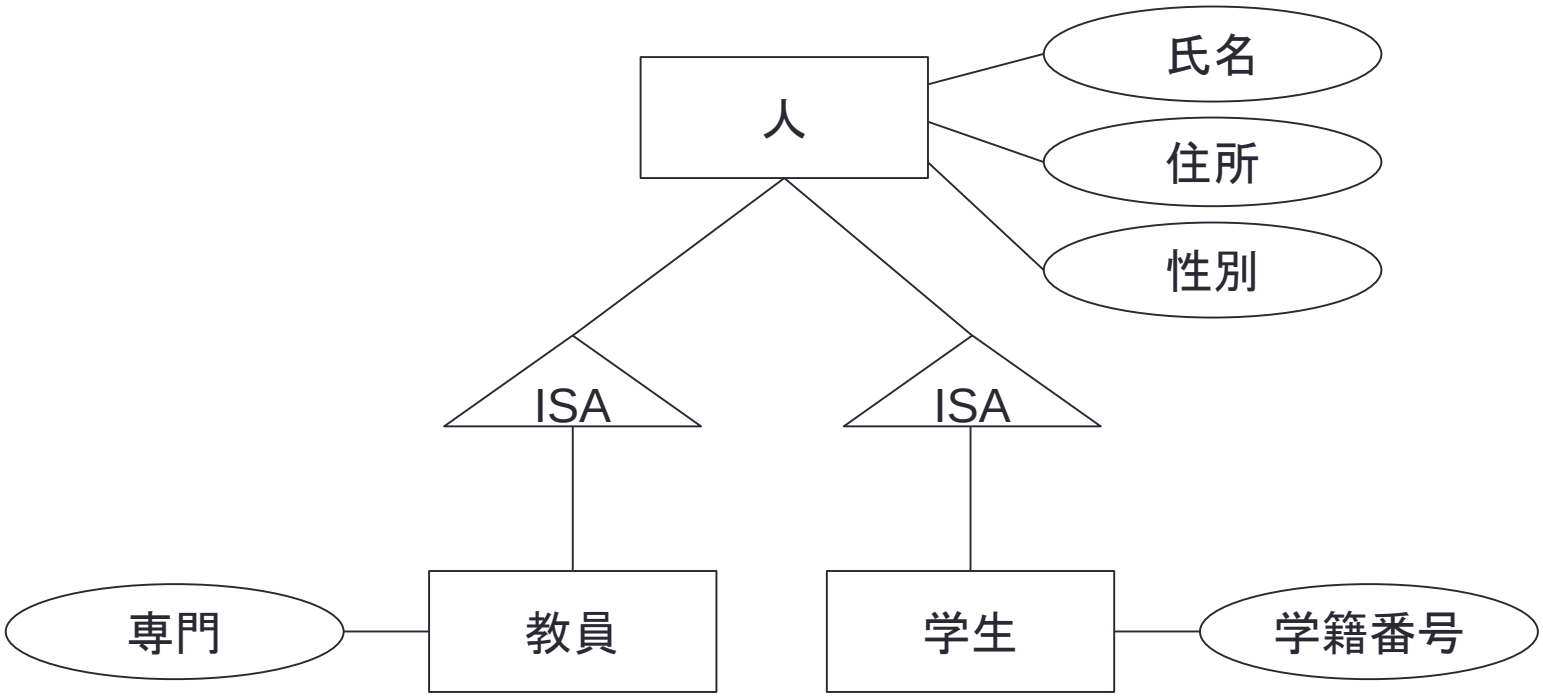


弱実体型とID関連型



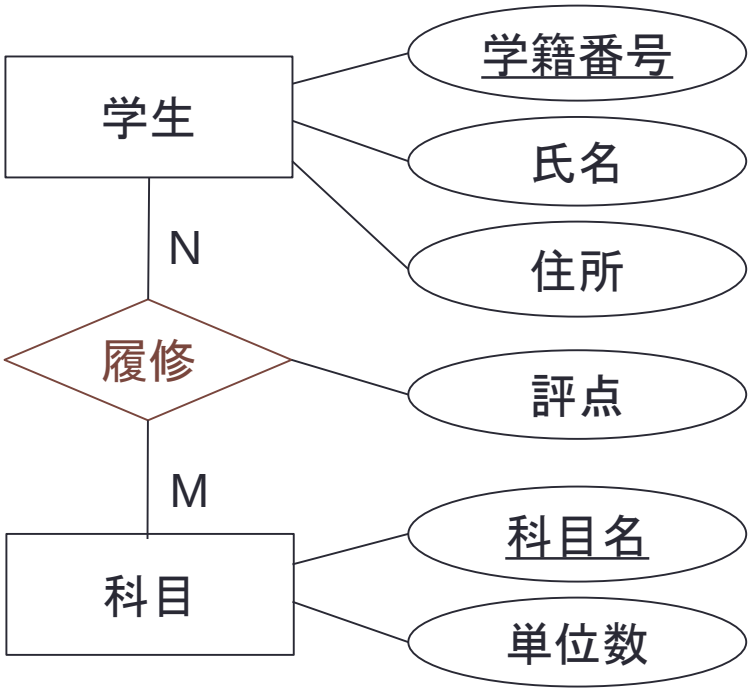
弱実体型:
他のエンティティのインスタンスが存在しなくなれば
自身のインスタンスも存在しなくなるエンティティ

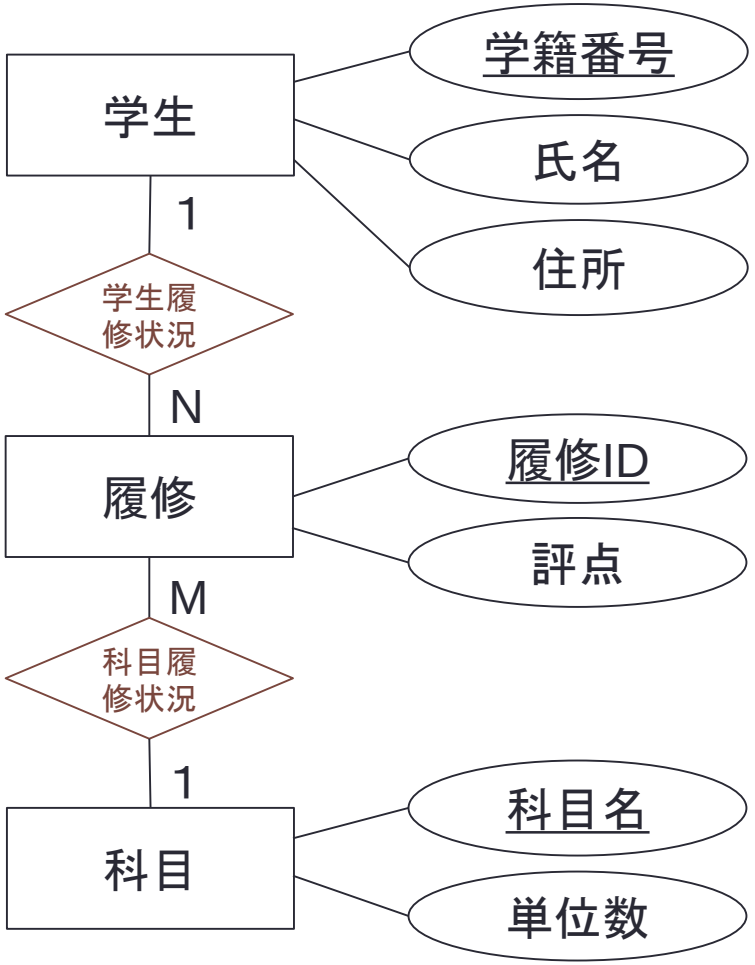
ISA(継承)関連



実体「人」をスーパタイプ，実体「教員」や「学生」をサブタイプと呼ぶこともある

表現の多様性





どちらでも表現できる

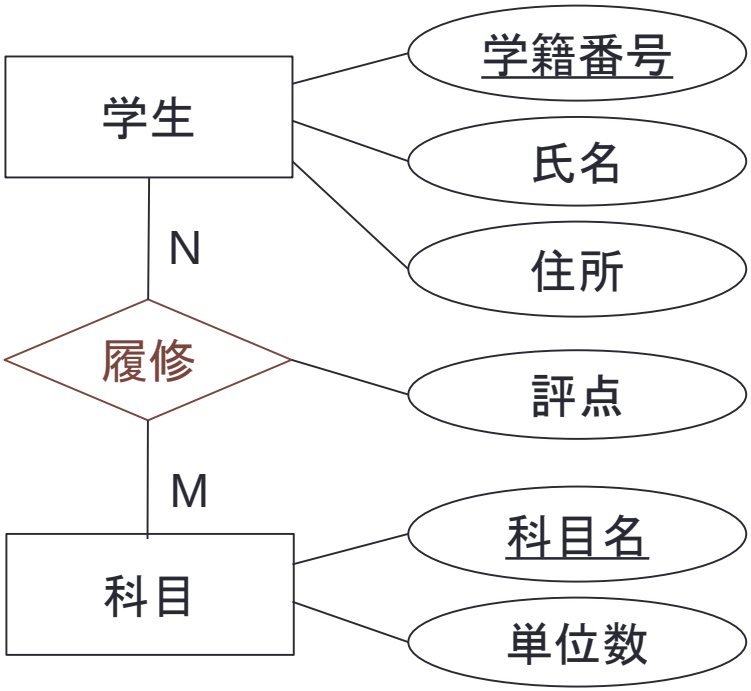
ER図→データベーススキーマ

一般的なルール

- $E(\underline{K}, A_1, A_2, A_3, .. A_n)$を実体型とする
  $\rightarrow R_E(\underline{K}, A_1, A_2, A_3, .. A_n)$

- $R(C_1, C_2, C_3, .. C_n)$ を二つの実体型$E_L(\underline{K}, A_1, A_2, A_3, .. A_n)$,
  $E_R(\underline{H}, B_1, B_2, B_3, .. B_n)$の関連型とする
  $\rightarrow R_E(\underline{K}, H, A_1, A_2, A_3, .. A_n)$ 1対1関連型(Hが主キーでも良い)
  $\rightarrow R_E(K, \underline{H}, A_1, A_2, A_3, .. A_n)$ 1対多関連型
  $\rightarrow R_E(\underline{K, H}, A_1, A_2, A_3, .. A_n)$ 多対多関連型

ER図→データベーススキーマ



学生

学籍番号	氏名	住所

履修

学籍番号	科目名	評点

科目

科目名	単位数

データベースの設計

- ER図
- トップダウンアプローチ
- ボトムアップアプローチ

トップダウンアプローチ

流れ

1. 大まかなER図の作成
   - 必要なデータの実体を決める
   - 濃度を考える
2. 属性の洗い出し
   - それぞれの実体に対して必要な属性を決める
3. 正規化
   - 洗い出した属性を正規化する

例題

現在，履修システム（以下，旧システム）が運用されており，
新規に参加部活動の情報を加える依頼が来た

旧システムの概要

- 学生は必ず一つの学科に所属する
- 学生は複数の科目を受講することもある
- 科目は必ず1人の教員が担当する
- 教員は複数科目担当することもある

科目コード:001 科目名：情報通信ネットワーク
担当教員：松澤

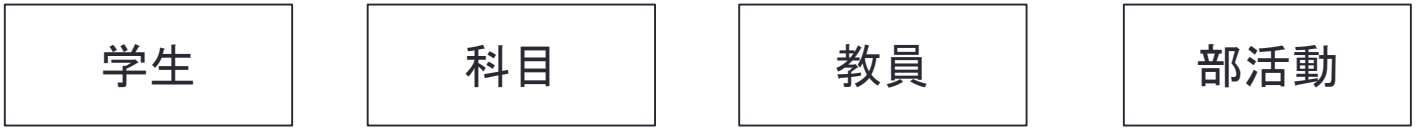
学籍番号	氏名	学科	評点
6300001	アリス	IS	80
6300002	ボブ	IS	90

参加部活動名簿の要件

- 見出し： 部活動コード，部活動名，顧問の教員名，活動場所
- 名簿：学生名，学籍番号，連絡先電話番号
- 顧問は1人の教員が担当する
- 学生は複数の部活動に参加することができる

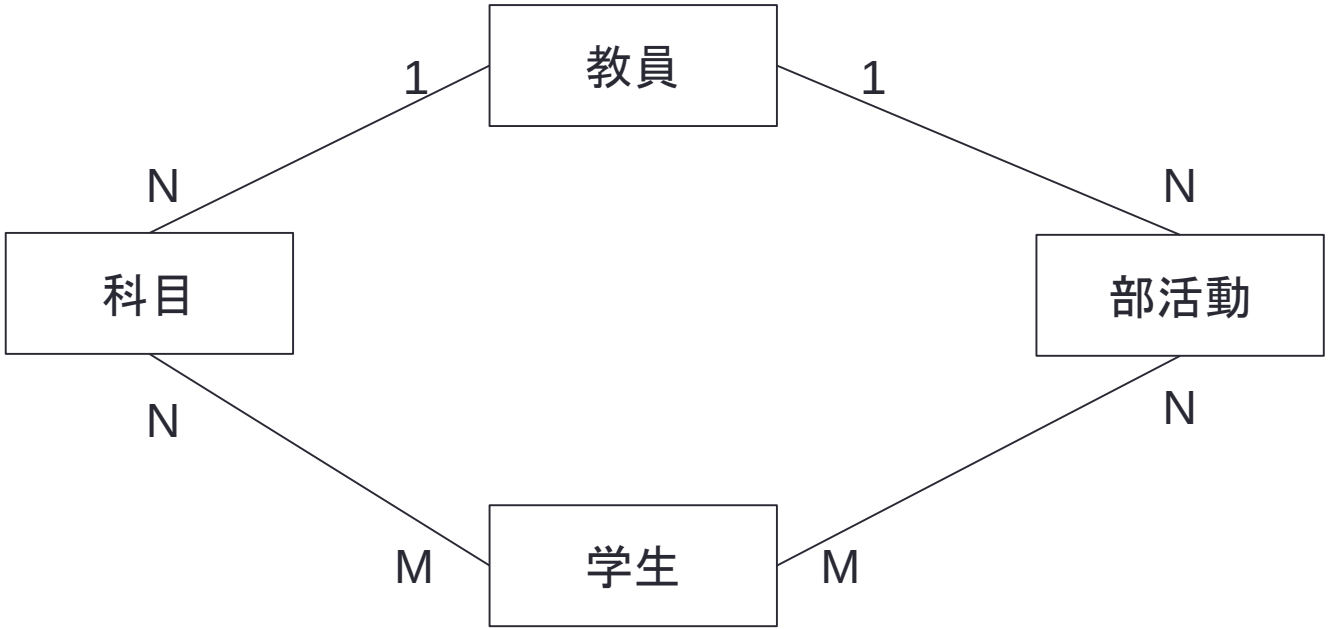
大まかなER図の作成

- データのまとまりとしての実体を洗い出す

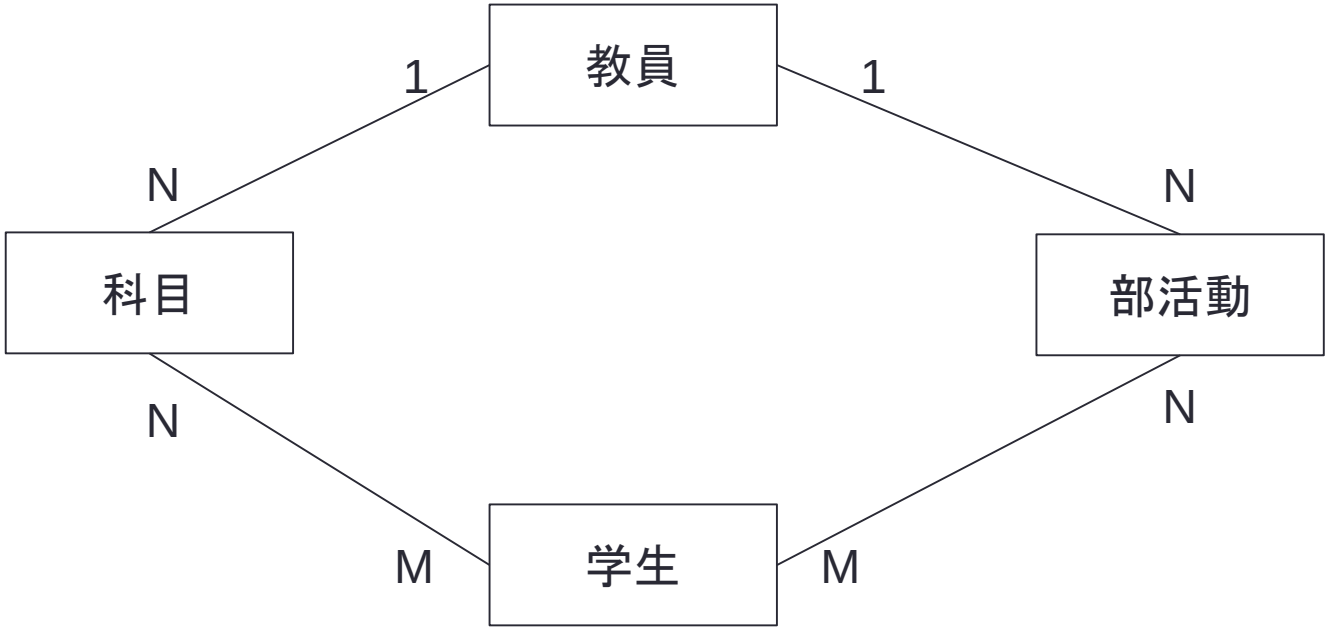


- 関連（リレーションシップ）がある実体と濃度を考える
  - 「学生は複数の科目を履修する」「一つの科目には複数の学生が履修する」
    →学生と科目は多対多の関連
  - 同様に教員と科目は1対多の関連
  - 学生と部活動は多対多の関連
  - 教員と部活動は1対多の関連

大まかなER図の作成



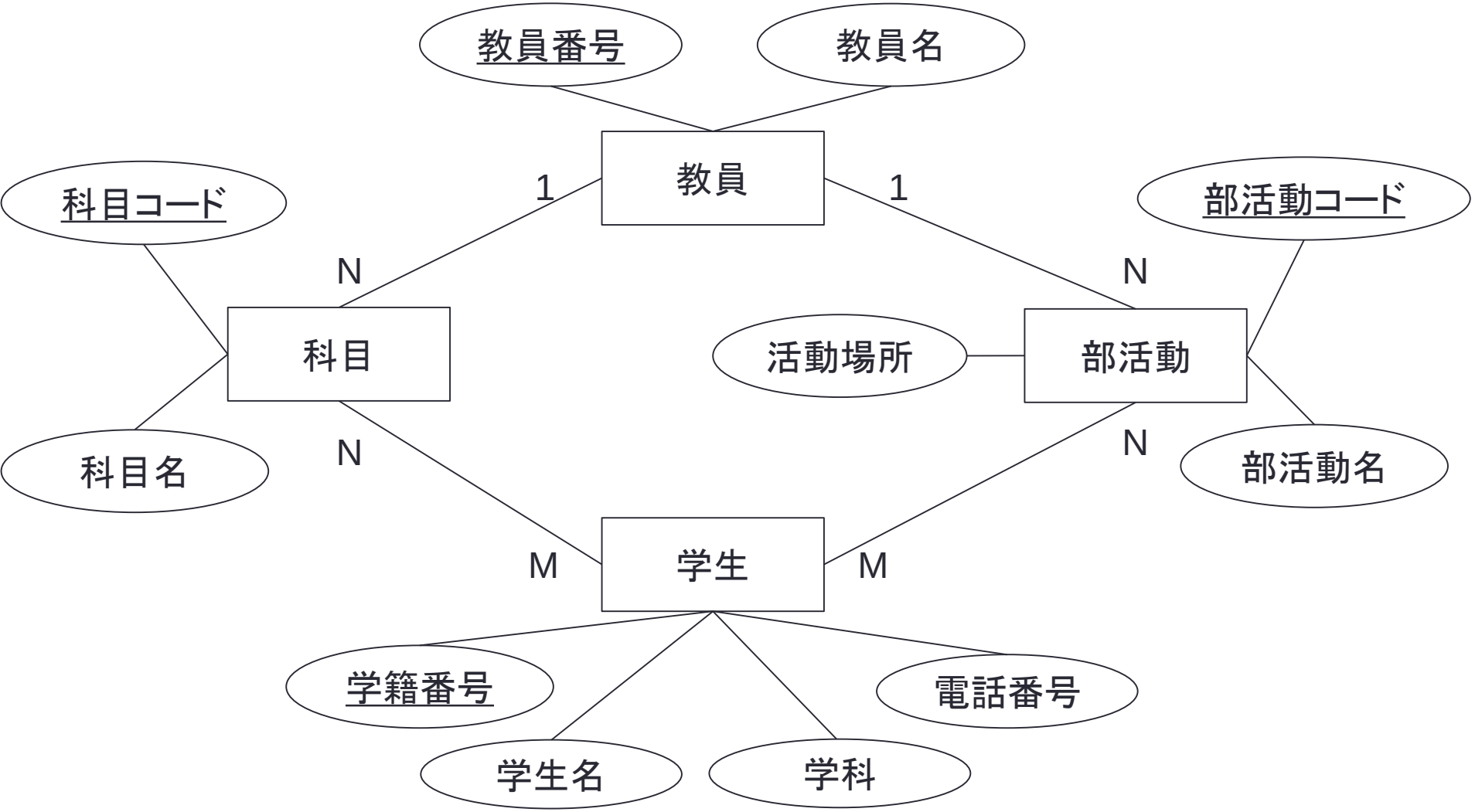
属性の洗い出し



- 学生（学籍番号，学生名，学科，電話番号）
- 教員（教員番号，教員名）
- 科目（科目コード，科目名）
- 部活動（部活動コード，部活動名，活動場所）

※赤字は適宜追加

属性の洗い出し

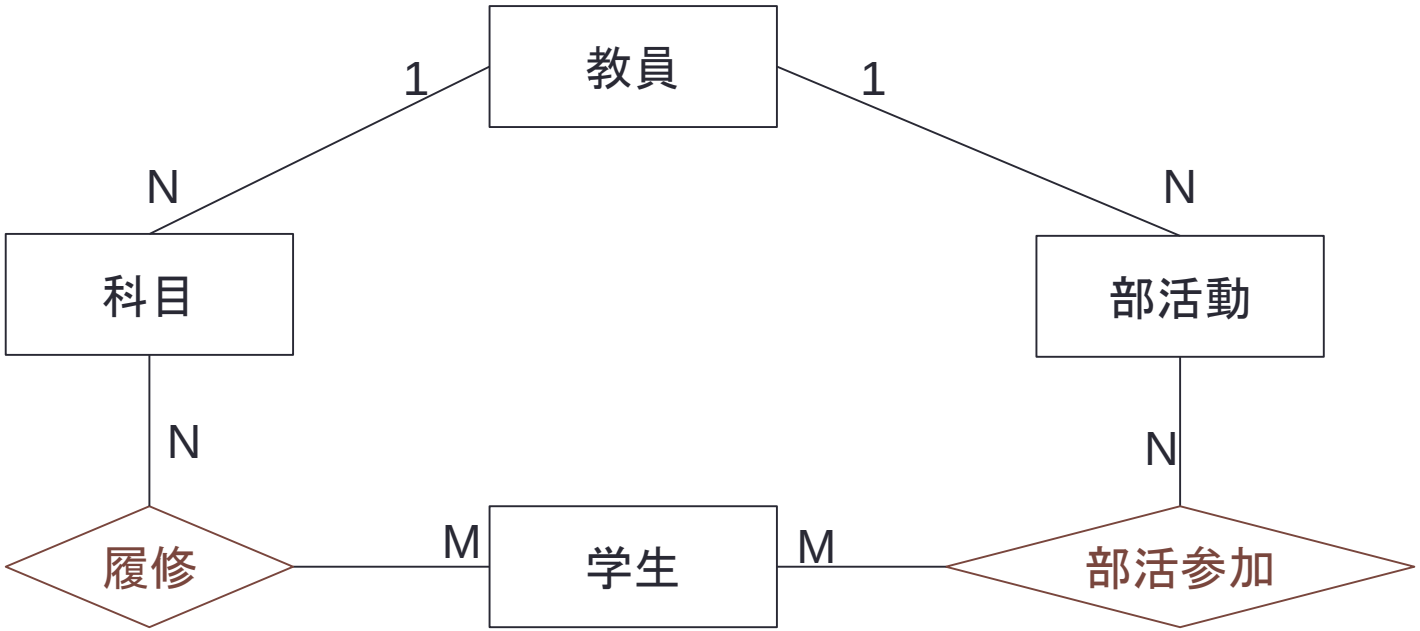

教員番号
教員名
教員
1
1
科目コード
部活動コード
N
N
科目
活動場所
部活動
科目名
N
N
部活動名
M
学生
M
学籍番号
電話番号
学生名
学科

正規化

- 先ほどの関係スキーマで第3正規形になっていないものがあれば第3正規形まで正規化する

- 濃度をもとに関連・実体を追加する
  - 1対1，1対多の関連は，一方の実体の主キーを外部キーとする
  - 多対多の関連は，必ず関連型を間に入れ，双方の主キーの組み合わせを主キーとする
    ※先ほどの組み合わせでは，科目-学生 と 学生-部活動

正規化



正規化

- 学生(学籍番号, 学生名, 学科, 電話番号)
- 教員(教員番号, 教員名)
- 科目(科目コード, 科目名, 教員番号)
- 部活動(部活動コード, 部活動名, 活動場所, 教員番号)
- 履修(科目コード, 学籍番号, 評点)
- 部活参加(部活動コード, 学籍番号)

データベースの設計

- ER図
- トップダウンアプローチ
- ボトムアップアプローチ

ボトムアップアプローチ

流れ

1. 属性の洗い出し
   - データベースに格納する必要がある属性を決める
2. 正規化
   - 洗い出した属性を正規化（第1，第2，第3正規形に順に分解）する
3. ER図の作成
   - 正規化されたテーブルを実体としてER図を作成する

例題

現在，履修システム（以下，旧システム）が運用されており，
新規に参加部活動の情報を加える依頼が来た

旧システムの概要

- 学生は必ず一つの学科に所属する
- 学生は複数の科目を受講することもある
- 科目は必ず1人の教員が担当する
- 教員は複数科目担当することもある

科目コード:001 科目名：情報通信ネットワーク
担当教員：松澤

学籍番号	氏名	学科	評点
6300001	アリス	IS	80
6300002	ボブ	IS	90

参加部活動名簿の要件

- 見出し： 部活動コード，部活動名，顧問の教員名，活動場所
- 名簿：学生名，学籍番号，連絡先電話番号
- 顧問は1人の教員が担当する
- 学生は複数の部活動に参加することができる

属性の洗い出し

科目コード:001 科目名:情報通信ネットワーク
担当教員:松澤

学籍番号	氏名	学科	評点
6300001	アリス	IS	80
6300002	ボブ	IS	90

- 履修
  （科目コード，科目名，教員番号，教員名，学籍番号，学生名，学科，評点）

※赤字は適宜追加

正規化

第1正規形にした履修

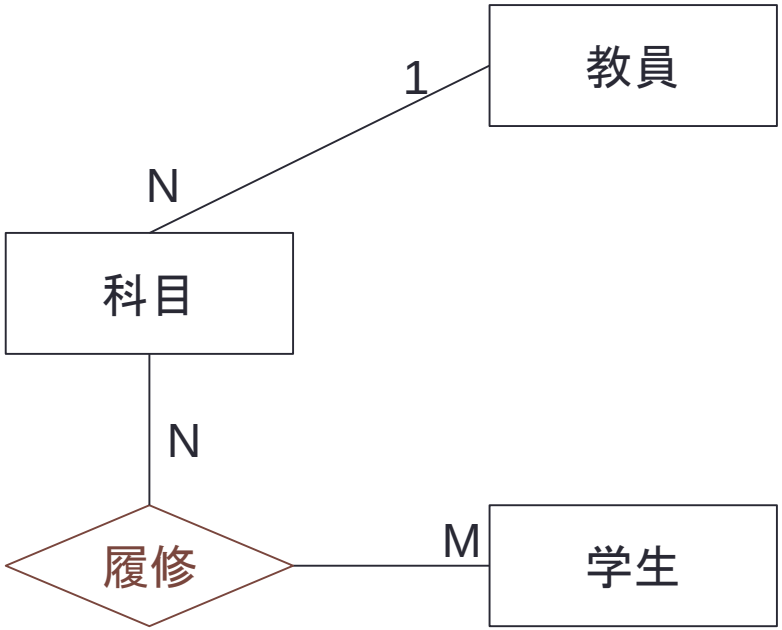
科目コード	科目名	教員番号	教員名	学籍番号	学生名	学科	評点
001	情報通信ネットワーク	200	松澤	6300001	アリス	IS	80
001	情報通信ネットワーク	200	松澤	6300002	ボブ	IS	90

正規化

- 第2正規形
  履修(科目コード, 学籍番号, 評点)
  科目(科目コード, 科目名, 教員番号, 教員名)
  学生(学籍番号, 学生名, 学科)

- 第3正規形
  履修(科目コード, 学籍番号, 評点)
  科目(科目コード, 科目名, 教員番号)
  学生(学籍番号, 学生名, 学科)
  教員(教員番号, 教員名)

ER図の作成



ボトムアップアプローチでは
帳表一つ一つに対して
- 属性洗い出し
- 正規化
- ER図作成

を行っていくため，この後
部活動に関する処理を行う必要がある

正規化の欠点

- 正規化による不都合

  正規化は「更新時異状を排除する」ために行う
  すべてのリレーションのデータが常に最新の状態を保つ
  - 履歴を残す場合
  - 売上明細など「発生した時点の情報」を保持する必要がある場合

- 処理速度の低下

  テーブルを分解すると，データ取得の度にテーブルの結合が発生する
  - 導出属性を持たせる（単価，税込み価格など）
  - テーブルを一つにまとめる（非正規化）

データベース設計の際には，正規化の欠点も頭に入れておく

データベースの制約

データベース設計の際には，格納される値に対しての制約をかけることで整合性を保つことができる

- 制約の種類
  - 検査制約（CHECK制約）
    - 特定の条件を満たす制約(>100等)
  - 非NULL制約
    - 格納するデータがNULLでないことを保証する制約
  - 一意性制約
    - 列のデータが他の行と重複しないことを保証する制約（候補キーでの使用）
  - 主キー制約（PRIMARY KEY 制約）
    - 一意性制約と非NULL制約を組み合わせたもの
  - 参照制約(外部キー制約　FOREIGN KEY制約)
    - 他のテーブルのデータを参照して一致していることを保証する制約

まとめ

- データベースの設計方法は二種類ある
  どちらもER図作成，属性洗い出し，正規化を行うが
  行う順番が異なる
  - トップダウンアプローチ
  - ボトムアップアプローチ
- ER図はデータモデルを記述するための図法

- 正規化は欠点もある

次回はSQL

- MySQLをインストールしておくこと

- https://dev.mysql.com/downloads/mysql/

- MacOSの場合は,
  Homebrew(https://brew.sh/index_ja.html)をインストールし,
  $ brew install mysql
  でも良い

質問あればどうぞ