

データベースシステム

第2回

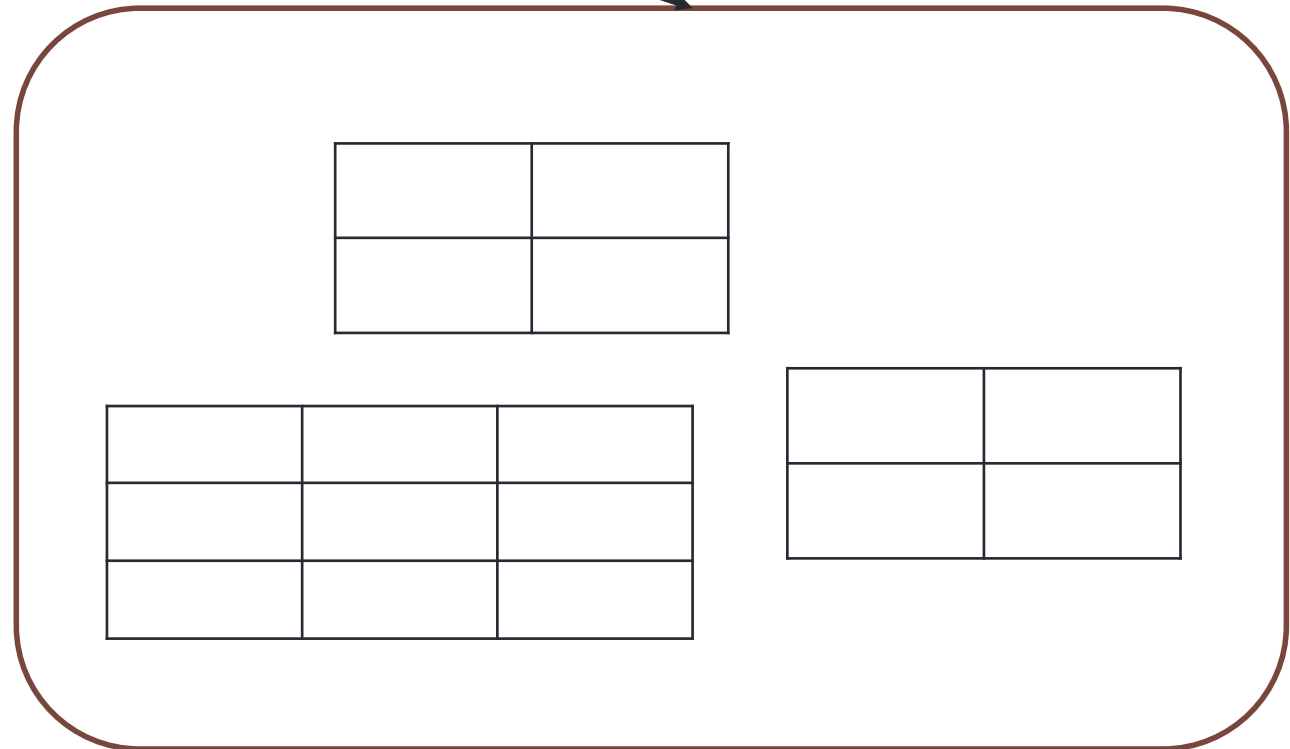
理工学部情報科学科

松澤 智史

データベースの基礎理論

リレーショナルデータベースの基本となる理論

- 正規化理論
- 関係演算



異状

- データベース内のデータがおかしくなって、直しようがないこと
- システム的なエラーでなく、登録データの追加, 更新, 削除等で起こる「データの矛盾」などのこと

このバスは無料です

このバスは運賃100円です

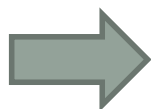


異状

会員番号	住所	注文した商品
100	千葉県流山市1-2- 2	りんご
101	千葉県柏市1-2-3-4	みかん
100	千葉県流山市1-2- 1	バナナ

異状の種類は3種類

- 挿入時異状
- 更新時異状
- 削除時異状



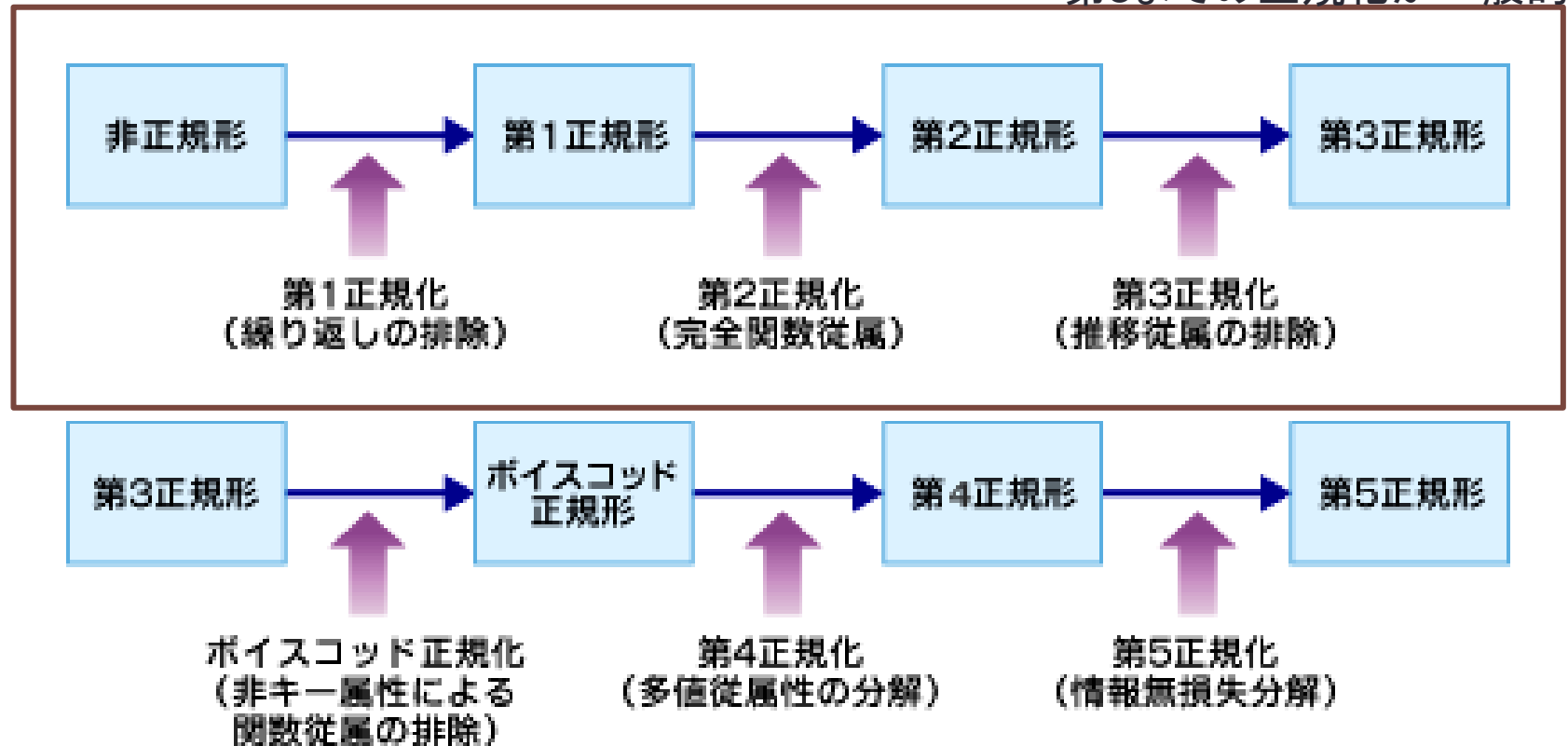
異状を排除するための設計を
正規化という

正規化 (Normalization)

- 一定のルールに従って変形を行うこと
- リレーショナルデータベースの正規化は正規化理論に従って関係を分解すること
- 異状はデータを更新する際に発生するため、更新・追加などを一切行わないデータに関しては不要
 - アクセスログ
 - 履歴

正規化の種類

第3までの正規化が一般的



第1正規化(第1正規形)

- 目的
 - 繰り返し現れるフィールドをなくすようにテーブルを分離する
 - このようなデータを「**原始的なデータ**」という
- やること
 - 直積集合の排除
 - べき集合の排除
 - 主キーの選出

教員 ID	名前	性別	職	専門
0	武田正之 (IS, 63)	男	教授	プログラミング言語論(1), ソフトウェア科学(2), 計算機ネットワーク(3)
1	松澤智史 (IS, 63)	男	講師	インターネットアーキテクチャ(4), 計算機ネットワーク(3), 複雑ネットワーク(5)

前提

名前に学科の情報も含んでいる. 職は変更される可能性がある
 専門は1つの場合もあれば複数ある場合もある

教員ID	名前	学科	学科 コード	性別	職	専門コード	専門
0	武田正之	IS	63	男	教授	1	プログラミング言語論
						2	ソフトウェア科学
						3	計算機ネットワーク
1	松澤智史	IS	63	男	講師	4	インターネットアーキテクチャ
						3	計算機ネットワーク
						5	複雑ネットワーク

直積集合の排除

各フィールドには単一の値のみ格納する形にする

教員ID	名前	学科	学科コード	性別	職	専門コード	専門
0	武田正之	IS	63	男	教授	1	プログラミング言語論
0	武田正之	IS	63	男	教授	2	ソフトウェア科学
0	武田正之	IS	63	男	教授	3	計算機ネットワーク
1	松澤智史	IS	63	男	講師	4	インターネットアーキテクチャ
1	松澤智史	IS	63	男	講師	3	計算機ネットワーク
1	松澤智史	IS	63	男	講師	5	複雑ネットワーク

べき集合の排除

べき集合→「与えられた集合からその部分集合を元として含む集合」

これで第1正規形は完成

主キー(Primary Key)の選択

- 候補キーの発見

- 行(タプル)が一意に定まる列の組み合わせを探す
- 複数ある場合は, 最小の列の組み合わせを選出

教員ID	名前	学科	学科コード	性別	職	専門コード	専門
0	武田正之	IS	63	男	教授	1	プログラミング言語論
0	武田正之	IS	63	男	教授	2	ソフトウェア科学
0	武田正之	IS	63	男	教授	3	計算機ネットワーク
1	松澤智史	IS	63	男	講師	4	インターネットアーキテクチャ
1	松澤智史	IS	63	男	講師	3	計算機ネットワーク
1	松澤智史	IS	63	男	講師	5	複雑ネットワーク

- (教員ID、専門コード),(名前, 専門),(名前, 専門コード)などが候補キー

主キー(Primary Key)の選択(2)

- 主キーを候補キーの中から1つ選ぶ
- 主キーをどれにするかはデータベースの使用目的による
- 主キーを構成する属性(列)の値はNULLを取ってはならない
※主キー制約という
- 主キーに選ばれなかった候補キーは代理キーとなる
- 表記ルールとして主キーの属性はアンダーバーで表現される
例: 顧客(顧客ID, 顧客名, メールアドレス, 住所)

第1正規形の問題

教員ID	名前	学科	学科コード	性別	職	専門コード	専門
0	武田正之	IS	63	男	教授	1	プログラミング言語論
0	武田正之	IS	63	男	教授	2	ソフトウェア科学
0	武田正之	IS	63	男	教授	3	計算機ネットワーク
1	松澤智史	IS	63	男	講師	4	インターネットアーキテクチャ
1	松澤智史	IS	63	男	講師	3	計算機ネットワーク
1	松澤智史	IS	63	男	講師	5	複雑ネットワーク
2	鈴木太郎	IS	63	男	助教	3	ネットワーク科学

新たに教員ID=2, 名前=鈴木太郎, 学科IS(63) 職=助教,
専門コード=3, 専門=ネットワーク科学のデータを追加した場合

教員ID	名前	学科	学科コード	性別	職	専門コード	専門
0	武田正之	IS	63	男	教授	1	プログラミング言語論
0	武田正之	IS	63	男	教授	2	ソフトウェア科学
0	武田正之	IS	63	男	教授	3	計算機ネットワーク
1	松澤智史	IS	63	男	講師	4	インターネットアーキテクチャ
1	松澤智史	IS	63	男	講師	3	計算機ネットワーク
1	松澤智史	IS	63	男	講師	5	複雑ネットワーク
2	鈴木太郎	IS	63	男	助教	3	ネットワーク科学

専門コード3は計算機ネットワーク？
それともネットワーク科学？



関数従属性

- ある属性Xが決まれば, 別の属性Yが一意に決まる
 - Xを決定項, Yに従属項と呼ぶ
 - $X \rightarrow Y$ と表記する
-
- 先ほどの例では教員ID(決定項)と教員名(従属項)となる

第2正規化(第2正規形)

- 目的
 - すべての非キー属性が「主キーに完全関数従属している」関係にする
- やること
 - 主キーの一部に従属する「部分関数従属性」を抜き出す

定義: 完全関数従属

- 関数従属性 $X \rightarrow Y$ において X のすべての真部分集合 X' について $X' \rightarrow Y$ が成立しない

定義: 部分関数従属

- 主キーの値が決まれば値が決定する属性

教員ID	名前	学科	学科コード	性別	職	専門コード	専門
0	武田正之	IS	63	男	教授	1	プログラミング言語論
0	武田正之	IS	63	男	教授	2	ソフトウェア科学
0	武田正之	IS	63	男	教授	3	計算機ネットワーク
1	松澤智史	IS	63	男	講師	4	インターネットアーキテクチャ
1	松澤智史	IS	63	男	講師	3	計算機ネットワーク
1	松澤智史	IS	63	男	講師	5	複雑ネットワーク

部分従属の属性(フィールド)は

- 教員ID, 名前, 学科, 学科コード
- 専門コード, 専門

教員ID	名前	学科	学科コード	性別	職
0	武田正之	IS	63	男	教授
1	松澤智史	IS	63	男	講師

専門コード	専門
1	プログラミング言語論
2	ソフトウェア科学
3	計算機ネットワーク
4	インターネットアーキテクチャ
5	複雑ネットワーク

教員ID	専門コード
0	1
0	2
0	3
1	4
1	3
1	5

これで第2正規形は完成

※第2正規形から第1正規形へは復元可能

第3正規化(第3正規形)

- 目的
 - すべての非キー属性が候補キーに推移的に関数従属しない関係にする
- やること
 - 主キー以外の属性(または属性の組)に従属する推移的関数従属性を抜き出す
- 定義: 推移的関数従属
 - 属性の集合 X , Y , Z において $X \rightarrow Y$, $Y \rightarrow Z$, $Y \not\rightarrow X$ の3つの制約が成立している関数従属

<u>教員ID</u>	名前	学科	学科コード	性別	職
0	武田正之	IS	63	男	教授
1	松澤智史	IS	63	男	講師

推移従属は, 教員ID→学科コード→学科なのでこれを分離



<u>教員ID</u>	名前	性別	職	学科コード	<u>学科コード</u>	学科名
0	武田正之	男	教授	63	63	IS
1	松澤智史	男	講師	63		

これで第3正規形が完成

最終的な第3正規形

教員ID	専門コード
0	1
0	2
0	3
1	4
1	3
1	5

専門コード	専門
1	プログラミング言語論
2	ソフトウェア科学
3	計算機ネットワーク
4	インターネットアーキテクチャ
5	複雑ネットワーク

教員ID	名前	性別	職	学科コード
0	武田正之	男	教授	63
1	松澤智史	男	講師	63

学科コード	学科名
63	IS

外部キー

- 複数の関係を結びつけるためのキー

教員ID	名前	学科	学科コード	性別	職
0	武田正之	IS	63	男	教授
1	松澤智史	IS	63	男	講師

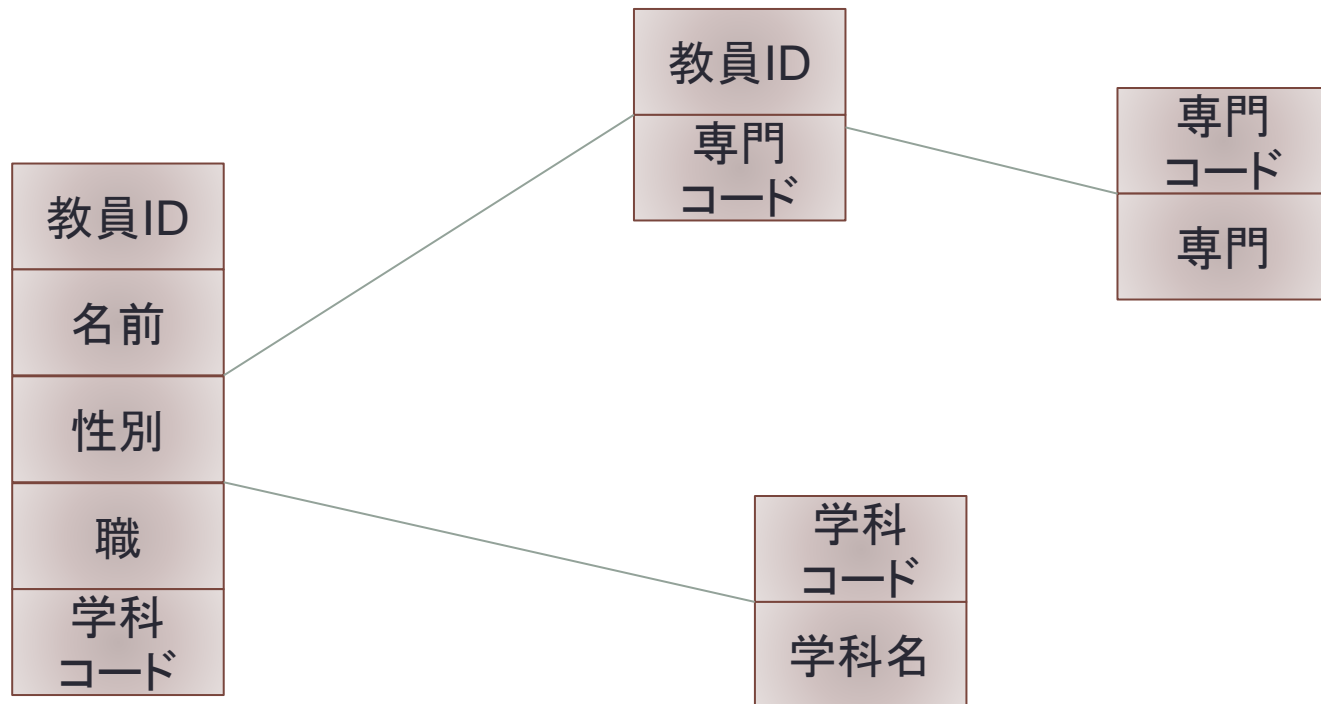
学科コード	学科名
63	IS

- この例では左のリレーション(表)の学科コードが外部キー, 右のリレーションの学科コードは候補キー(主キー)となっている
- 外部キーには以下の制約(外部キー制約または参照制約)がある
 - 左リレーションの学科コードには右リレーションに存在する学科コードしか入力できない
 - 右リレーションの学科コードは, 左リレーションに存在する学科コードを削除することができない

E-R図

- リレーショナルデータベースのテーブル構造やリレーションシップはER図(Entity-Relationship diagram)で表記することができる
- 実体(Entity)と関連(Relationship)の2つの概念がある
- 実体は人, 物, 情報などを表し, 関連はその実体間に存在する関係を定義する
- ER図の詳しい書き方はデータベース設計の回で解説する

E-R図



高次の正規形

- ボイスコード正規形
- 第4正規形
- 第5正規形

ボイスコード正規形

- 第2, 3正規形では, 切り離しの対象属性が「非キー属性」のみであったが, ボイスコード正規形は「非キー属性」の制約を外した部分関数従属性や推移的関数従属性を排除する

学生	科目	教員
A	人工知能	桂田
A	ネットワーク	松澤
B	人工知能	大村
C	プログラミング	滝本

主キーは(学生, 科目)であるが, 教員が決まると科目も一意に定まる関係
候補キーは(学生, 科目)以外にも(学生, 教員)

学生	科目	教員
A	人工知能	桂田
A	ネットワーク	松澤
B	人工知能	大村
C	プログラミング	滝本



ボイスコード正規化

学生	教員
A	桂田
A	松澤
B	大村
C	滝本

教員	科目
桂田	人工知能
松澤	ネットワーク
大村	人工知能
滝本	プログラミング

ボイスコード正規形の問題

学生	教員
A	桂田
A	松澤
B	大村
C	滝本
A	大村

教員	科目
桂田	人工知能
松澤	ネットワーク
大村	人工知能
滝本	プログラミング

主キーが(学生, 教員)であるため, (A,大村)の行(タプル)が挿入可能になる. 教員名で結合して元に戻そうとすると...

学生	科目	教員
A	人工知能	桂田
A	ネットワーク	松澤
B	人工知能	大村
C	プログラミング	滝本
A	人工知能	大村

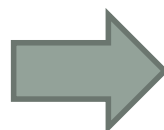
ボイスコード正規形の問題(2)

- このような例を関数従属性損失という
- 情報無損失分解でかつ関数従属性を保存したまま分解可能なのは第3正規形までである

第4正規形

- 多値従属性を分解する

旅程番号	顧客	便名
Plan1	A	JAL100
Plan1	A	ANA700
Plan1	B	JAL100
Plan1	B	ANA700
Plan2	C	JAL200
Plan2	C	ANA500
Plan2	C	ANA800



旅程番号	顧客
Plan1	A
Plan1	B
Plan2	C

旅程番号	便名
Plan1	JAL100
Plan1	ANA700
Plan2	JAL200
Plan2	ANA500
Plan2	ANA800

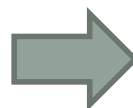
定義：多値従属性

ある属性Xが決めると別の属性Y, Z(複数可)が独立して決まる

第5正規形

- 結合従属性を分解する

仕入元	加工品	供給先
A社	Tシャツ	α販売店
A社	Tシャツ	β販売店
A社	ポロシャツ	β販売店
B社	Tシャツ	β販売店



仕入元	加工品
A社	Tシャツ
A社	ポロシャツ
B社	Tシャツ

仕入元	供給先
A社	α販売店
A社	β販売店
B社	β販売店

加工品	供給先
Tシャツ	α販売店
Tシャツ	β販売店
ポロシャツ	β販売店

定義: 結合従属性

関係が3つ以上に分解可能な従属性

n個に分解可能なことをn-分解可能という

この分解は結合によって元に戻すことが可能

まとめ

- 第1正規形→原子的データに分解
- 第2正規形→部分従属を分解
- 第3正規形→推移従属を分解
- ボイスコッド正規形→非キー属性以外の部分・推移従属を分解
- 第4正規形→多値従属を分解
- 第5正規形→結合従属を分解

- 高次の正規形(ボイスコッド, 第4, 第5)では情報損失, 関数従属性損失が起こりうる
- 実用的に行われるのは第3正規形まで
→高次の正規形はこんな正規化の仕方もあるよ・・・程度で良い

質問あればどうぞ