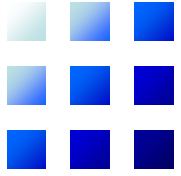


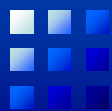
データベース

【8:リレーショナルデータベース 設計論(3)】

石川 佳治



分解



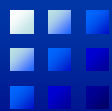
分解 (decomposition) とは

- リレーションスキーマ $RS = \{A_1, \dots, A_n\}$ に対して, $RS_i \subseteq RS$ ($1 \leq i \leq m$) なるリレーションスキーマの集合 $\rho = \{RS_1, \dots, RS_m\}$ で

$$RS_1 \cup \dots \cup RS_m = RS$$

となるもの

- 例: 4.2節 (p. 59) におけるリレーション「営業」のリレーション「販売」, 「営業担当」への分割
- 分解は**更新不整合** (4.2節) を解消したリレーションスキーマを得るための一つの手段



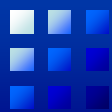
無損失結合分解(1): 定義

- 良い分解の基準の一つ
 - 分解によって得られたリレーションスキーマの集合が元のリレーションスキーマと基本的に同じ情報を表現することを保証
- **定義**: RS に関する関数従属性集合 F と RS の分解 $\rho = \{RS_1, \dots, RS_m\}$ が与えられたとき, F を満足する RS のすべてのインスタンスについて

$$R = \pi_{RS_1}(R) \bowtie \dots \bowtie \pi_{RS_m}(R)$$

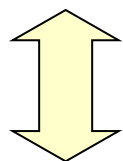
が成立するとき, 分解 ρ を RS の**無損失結合分解** (lossless join decomposition) という

- 結合処理で余分なタプルが現れない



無損失結合分解(2): 定理4.1

- $\rho = \{RS_1, RS_2\}$ がリレーション RS の分解, F が RS に関する関数従属性のとき, 分解 ρ が無損失結合分解



必要十分条件

$$RS_1 \cap RS_2 \rightarrow RS_1 - RS_2 \in F^+ \quad \text{または}$$

$$RS_1 \cap RS_2 \rightarrow RS_2 - RS_1 \in F^+$$



無損失結合分解(3):例

- リレーション「営業」(4.2節)
 - リレーションスキーマ: {商品番号, 顧客番号, 社員番号, 販売価格}
 - 関数従属性集合の極小被覆: {商品番号, 顧客番号 \rightarrow 販売価格, 顧客番号 \rightarrow 社員番号}
- 分解 $\rho_1 = \{\{\text{商品番号, 顧客番号, 販売価格}\}, \{\text{顧客番号, 社員番号}\}\}$ (図4.1)は**無損失結合分解**
 - $\{\text{商品番号, 顧客番号, 販売価格}\} \cap \{\text{顧客番号, 社員番号}\} = \{\text{顧客番号}\}$
 - $\{\text{顧客番号, 社員番号}\} - \{\text{商品番号, 顧客番号, 販売価格}\} = \{\text{社員番号}\}$
 - 顧客番号 \rightarrow 社員番号は F^+ に含まれる



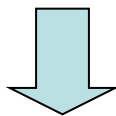
無損失結合分解(4):例(続き)

- 分解 $\rho_2 = \{\{\text{商品番号}, \text{顧客番号}\}, \{\text{商品番号}, \text{社員番号}, \text{販売価格}\}\}$ は無損失結合分解でない
 - $\{\text{商品番号}, \text{顧客番号}\} \cap \{\text{商品番号}, \text{社員番号}, \text{販売価格}\} = \{\text{商品番号}\}$
 - $\{\text{商品番号}, \text{顧客番号}\} - \{\text{商品番号}, \text{社員番号}, \text{販売価格}\} = \{\text{顧客番号}\}$
 - $\{\text{商品番号}, \text{社員番号}, \text{販売価格}\} - \{\text{商品番号}, \text{顧客番号}\} = \{\text{社員番号}, \text{販売価格}\}$
 - 商品番号 \rightarrow 顧客番号, 商品番号 \rightarrow 社員番号, 販売価格は F^+ に含まれない



従属性保存分解(1): 動機

- 分解に関する疑問
 - 元のリレーションスキーマにおける関数従属性の集合は, 分解後のリレーションスキーマにおいても保存されるか?



従属性保存分解の概念の導入



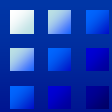
従属性保存分解(2): 関数従属性の射影

- リレーションスキーマ RS , その関数従属性集合 F , および属性集合 $Z \subseteq RS$ が与えられたとき

$$\pi_Z(F) = \{X \rightarrow Y \mid XY \subseteq Z \wedge X \rightarrow Y \in F^+\}$$

を F の X への射影という

- 例: $RS = \{A, B, C\}$, $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow B\}$ のとき
 - $\pi_{AB}(F)$ の極小被覆: $\{A \rightarrow B\}$
 - $\pi_{BC}(F)$ の極小被覆: $\{B \rightarrow C, C \rightarrow B\}$
 - $\pi_{AC}(F)$ の極小被覆: $\{A \rightarrow C\}$

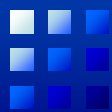


従属性保存分解(3): 定義

- リレーションスキーマ RS , RS に関する関数従属性集合 F , RS の分解 $\rho = \{RS_1, \dots, RS_m\}$ が与えられたとき, 任意の関数従属性 $X \rightarrow Y \in F$ に対して

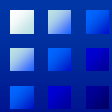
$$\pi_{RS_1}(F) \cup \dots \cup \pi_{RS_m}(F) \models X \rightarrow Y$$

が成立するならば, 分解 ρ は F に関する**従属性保存分解** (dependency preserving decomposition) である



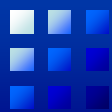
従属性保存分解(4):例(1)

- $RS = \{A, B, C\}, F = \{A \rightarrow B, B \rightarrow C, C \rightarrow B\}$
- $\rho_1 = \{\{A, B\}, \{B, C\}\}$ は従属性保存分解
 - $\pi_{AB}(F) = \{A \rightarrow B\}, \pi_{BC}(F) = \{B \rightarrow C, C \rightarrow B\}$ という射影中に元の従属性が保存される
- $\rho_2 = \{\{A, B\}, \{A, C\}\}$ は従属性保存分解でない
 - $\pi_{AB}(F) = \{A \rightarrow B\}, \pi_{AC}(F) = \{A \rightarrow C\}$ という極小被覆が得られる
 - これらを合わせても、もとの従属性集合とはならない



従属性保存分解(5):例(2)

- リレーションスキーマ {商品番号, 顧客番号, 社員番号, 販売価格}
- 分解 $\rho_1 = \{\{\text{商品番号, 顧客番号, 販売価格}\}, \{\text{顧客番号, 社員番号}\}\}$ は**従属性保存**
 - $\pi_{\text{商品番号, 顧客番号, 販売価格}}(F) = \{\text{商品番号, 顧客番号} \rightarrow \text{販売価格}\}$
 - $\pi_{\text{顧客番号, 社員番号}}(F) = \{\text{顧客番号} \rightarrow \text{社員番号}\}$
- 分解 $\rho_2 = \{\{\text{商品番号, 顧客番号}\}, \{\text{商品番号, 社員番号, 販売価格}\}\}$ は**従属性保存でない**
 - $\pi_{\text{商品番号, 顧客番号}}(F) = \{ \}$
 - $\pi_{\text{商品番号, 社員番号, 販売価格}}(F) = \{ \}$



従属性保存分解(6):注意(1)

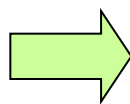
- 無損失結合分解だが従属性保存分解でない場合がある
- 例: $RS = \{\text{学生}, \text{学部}, \text{URL}\}$
 - 関数従属性の集合: $\{\text{学生} \rightarrow \text{学部}, \text{学部} \rightarrow \text{URL}, \text{URL} \rightarrow \text{学部}\}$ (先の例の具体化)
 - $\rho = \{\{\text{学生}, \text{学部}\}, \{\text{学生}, \text{URL}\}\}$ と分解
 - ρ は無損失結合分解
 - $\{\text{学生}, \text{学部}\} \cap \{\text{学生}, \text{URL}\} = \{\text{学生}\}$
 - $\{\text{学生}, \text{学部}\} - \{\text{学生}, \text{URL}\} = \{\text{学部}\}$
 - $\text{学生} \rightarrow \text{学部}$ が成立
 - 先に見たように, ρ は従属性保存分解でない



従属性保存分解(7):注意(2)

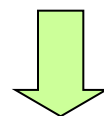
学生	学部	URL
鈴木明	工学部	URL _E
山田次郎	法学部	URL _L
田中一郎	工学部	URL _E

分解



学生	学部
鈴木明	工学部
山田次郎	文学部
田中一郎	工学部

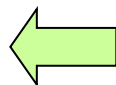
学生	URL
鈴木明	URL _E
山田次郎	URL _L
田中一郎	URL _E



田中一郎の
学部を修正

学生	学部	URL
鈴木明	工学部	URL _E
山田次郎	法学部	URL _L
田中一郎	理学部	URL _E

再結合



学生	学部
鈴木明	工学部
山田次郎	文学部
田中一郎	理学部

学部とURLの対応が失われている!



従属性保存分解(8): 判定アルゴリズム

- 従属性保存分解かを判定するアルゴリズム

```
for each  $X \rightarrow Y \in F$  do
```

```
  begin
```

```
     $Z := X$ 
```

```
    repeat
```

```
      for  $i := 1$  to  $m$  do
```

```
         $Z := Z \cup ((Z \cap RS_i)^+ \cap RS_i)^+$ 
```

```
      until  $Z$  が変化せず
```

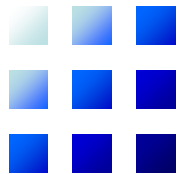
```
      if  $Y \not\subseteq Z$  then return (No)
```

```
    end
```

```
  return (Yes)
```

F に関する
閉包をとる

$X \rightarrow Y$ は
導けない



第三正規形



正規形とは

- 更新不整合などの問題点が発生しないような制約条件を与えることができる
 - 従属性の概念に基づく
- 条件の程度に応じて、いくつかの**正規形** (normal form) が定義される
- 第一正規形の制約にさらに制約を加えるため、**高次の正規化**と呼ばれる
 - **第一正規形** (first normal form; 1NF) : 各属性のとりうる値は単純値である



第三正規形(1):素属性

- **素属性** (prime attribute)
 - リレーションスキーマ RS の属性のうち, いずれかの候補キーの構成要素となる属性
 - **非素属性** (nonprime attribute): 素属性以外の属性
- 例: リレーション 学生 (学籍番号, 氏名, 専攻, 住所)
 - 学籍番号および {氏名, 住所} が候補キー
 - 素属性は学籍番号, 氏名, 住所
 - 非素属性は専攻



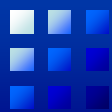
第三正規形(2): 定義

- **第三正規形** (third normal form; 3NF)
 - 最も基本的な正規形
 - 更新不整合を解消
- リレーションスキーマ RS が 3NF であるための条件: 関数従属性 $X \rightarrow A$ ($X \subseteq RS, A \in RS, A \notin X$) が成り立つ場合には以下の二つの条件の**いずれか**が常に満たされるときをいう
 - ① X が RS の超キーである
 - ② A が素属性である



第三正規形(3):例

- リレーションスキーマ {商品番号, 顧客番号, 社員番号, 販売価格}
 - 候補キーは {商品番号, 顧客番号} のみ
 - 関数従属性 顧客番号 \rightarrow 社員番号が成立
 - $X = \{\text{顧客番号}\}$, $A = \text{社員番号}$ において, 3NF の条件をチェック
 - ① $X = \{\text{顧客番号}\}$ は超キーでない \Rightarrow 条件を満たさない
 - ② $A = \text{社員番号}$ は素属性でない
 - ①, ②とも不成立のため, 3NF でない



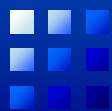
第三正規形(4):例(続き)

- 以下のように分解
 - 販売: {商品番号, 顧客番号, 販売価格}
 - 営業担当: {顧客番号, 社員番号}
- リレーション「販売」について検討
 - 関数従属性集合の極小被覆は {商品番号, 顧客番号 → 販売価格}
 - $X = \{\text{商品番号}, \text{顧客番号}\}$, $A = \text{販売価格}$ としたとき, X は「販売」の候補キー(すなわち超キー)なので条件①を満たす
- リレーション「営業担当」について検討
 - 関数従属性の極小被覆は {顧客番号 → 社員番号}
 - $X = \{\text{顧客番号}\}$, $A = \text{社員番号}$ としたとき, 同様に①を満たす
- 以上より「販売」, 「営業担当」とともに 3NF



第二正規形 (second normal form; 2NF)

- 1NF と 3NF の中間
- 定義: 関数従属性 $X \rightarrow A$ ($X \subseteq RS, A \in RS, A \notin X$) が成り立つ場合, 以下のいずれかが常に満たされる
 - ① RS のいかなる候補キーについても, X はその真部分集合ではない: 3NF よりも緩い条件
 - ② A が素属性である
- 2NF は 3NF に至る中間段階としての性質が強く, 重要度は低い
- 通常, 直接 3NF への正規化を行う



2NF, 3NFの考え方(1)

- 2NF の条件①「 RS のいかなる候補キーについても、 X はその真部分集合ではない」に反する例
 - リレーション履修: {学籍番号, 科目番号, 氏名, 得点}
 - 候補キーは {学籍番号, 科目番号} のみ
 - 学籍番号 \rightarrow 氏名 という関数従属性が存在
 - $X = \{\text{学籍番号}\}$, $A = \text{氏名}$ とすると $\{\text{学籍番号}\} \subset \{\text{学籍番号}, \text{科目番号}\}$ なので真部分集合となるので、条件①に反する

候補キー

学籍番号	科目番号	氏名	得点
...

候補キーの一部の属性集合からの関数従属性(部分従属)が存在
 \Rightarrow 2NF ではこれを禁止



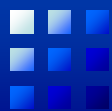
2NF, 3NFの考え方(2)

- 2NF ではあるが, 3NF の条件①「 X が RS の超キーである」に反する例
 - リレーション学生: {学籍番号, 氏名, 専攻, 専攻長}
 - 候補キーは {学籍番号} のみ
 - 関数従属性 専攻 \rightarrow 専攻長 が存在
 - $X = \{\text{専攻}\}$, $A = \text{専攻長}$ としたとき, X は候補キー {学籍番号} の真部分集合でない. よって 2NF
 - しかし, X は超キーでない(かつ A は素属性でない)ので 3NF でない

候補キー

<u>学籍番号</u>	氏名	<u>専攻</u>	専攻長
...

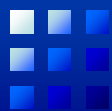
学籍番号 \rightarrow 専攻長 という関数従属性は, 二つの関数従属性の推移的関係(推移従属)で発生 \Rightarrow 3NFではこれを禁止 23



2NF, 3NFの考え方(3)

- 素属性に関する条件②: 例外的な規定
- 例: {商品番号, 販売地域, 販売担当者}
 - 関数従属性: 商品番号, 販売地域 → 販売担当者, 販売担当者 → 販売地域
 - 販売担当者 → 販売地域 は 3NF の条件①を満たさないが②を満たす
 - 確かに内部構造があるが, 分解するかは悩ましい
 - 3NFでは分解しない
 - 4.6節のボイス・コッド正規形で再度議論

<u>商品番号</u>	<u>販売地域</u>	<u>販売担当者</u>
...



2NF, 3NFの考え方(4)

- 3NFのまとめ

- リレーションスキーマ内部における, 冗長性の原因となる従属性の構造を排除する

- **部分従属**: 2NF の条件①で排除

- **推移従属**: 3NF の条件①(2NF の条件①を包含)で排除

- 素属性に関する条件(条件②)

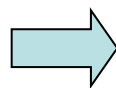
- 例外的な規定

- 3NFへの分解は良い性質を持つ

- 無損失結合を保証

- 従属性保存を保証

- 更新不整合に対処



原則として 3NF までの
正規化は必ず実施



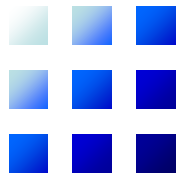
3NFへの分解アルゴリズム(1)

- 無損失結合と従属性保存を保証

```
 $\rho := \{ \}$   
 $M := F$  の極小被覆  
if  $XA = RS$  なる  $X \rightarrow A \in M$  が存在 then  
   $\rho := \{RS\}$   
else  
  begin  
    for each  $X \rightarrow A \in M$  do  
      begin  
         $\rho := \rho \cup \{XA\}$   
         $M := M - \{X \rightarrow A\}$   
      end  
    if いかなるリレーションスキーマ  $\in \rho$  も  $RS$  の候補キーを含まない then  
       $\rho := \rho \cup \{RS \text{ の任意の候補キー}\}$   
  end  
end
```

3NFへの分解アルゴリズム(2):例

- リレーションスキーマ $RS = \{I, A, S, P\}$
 - I は商品番号, A は販売地域, S は販売担当者, P は商品の製造メーカー
 - 関数従属性集合 $F = \{IA \rightarrow S, I \rightarrow P, S \rightarrow A\}$
- アルゴリズムの実行
 - for ループより $\{\{I, A, S\}, \{I, P\}, \{A, S\}\}$ を得る
 - $\{I, A, S\}$ に候補キー IA が含まれるので終了
 - 実際には $\{I, A, S\} \supseteq \{A, S\}$ なので, 分解 $\{\{I, A, S\}, \{I, P\}\}$ を用いる
 - これは 3NF



さらに高次の正規化 (簡単に)



ボイス・コッド正規形(1)

- **ボイス・コッド正規形** (Boyce-Codd normal form; BCNF)
 - 第三正規形の条件②を除外したもの
 - 第三正規形よりも厳しい条件
- リレーションスキーマ RS が BCNF であるための条件: 関数従属性 $X \rightarrow A$ ($X \subseteq RS, A \in RS, A \in X$) が成り立つ場合には以下の条件が常に満たされるときをいう
 - ① X が RS の超キーである
 - ~~② A が素属性である~~



ボイス・コード正規形(2)

- 例: {商品番号, 販売地域, 販売担当者}
 - 関数従属性: 商品番号, 販売地域 → 販売担当者
販売担当者 → 販売地域
 - 販売担当者 → 販売地域 は 3NF の条件①を満たさないが②を満たすので3NF. しかしBCNFではない
 - BCNFとするためには分解が必要

3NF

商品番号	販売地域	販売担当者
...

分解
→

BCNF

商品番号	販売地域
...	...

販売地域	販売担当者
...	...



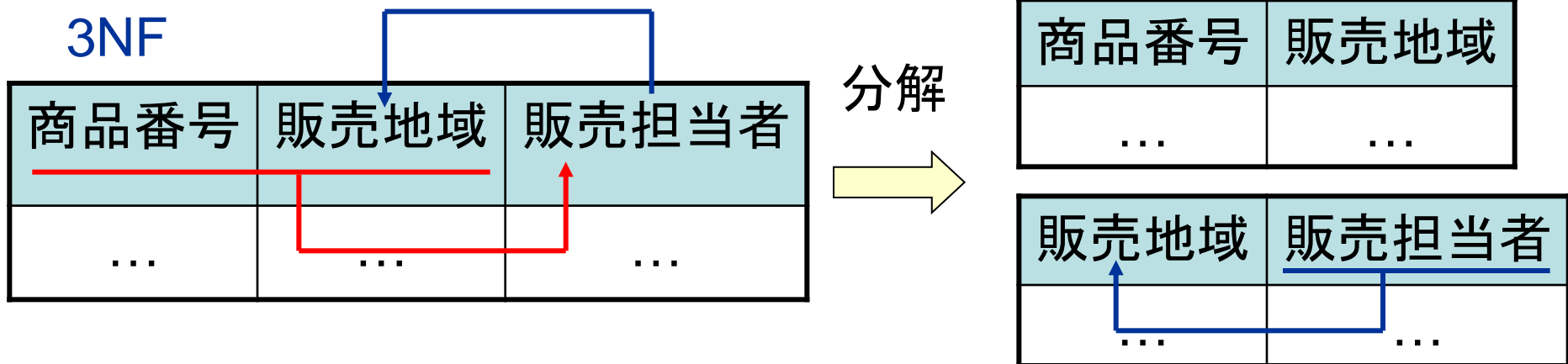
ボイス・コード正規形(3)

- BCNFへの分解の問題点

- BCNFへの分解は必ずしも従属性保存とは限らない

- 例: 商品番号, 販売地域 → 販売担当者 は保存されない

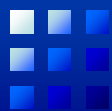
- なお, 無損失結合分解は可能





3NFとBCNF

- 3NFに正規化したリレーションがすでにBCNFでもある場合はしばしば発生
 - 問題は生じない
- 3NFではあるがBCNFでない場合
 - **本質的には対象世界の問題**
 - 対処策
 1. 3NFのままにする: 冗長性が残る
 2. BCNFに分解する: 関数従属性が失われる
 - いずれの場合も, **データベースアプリケーションなどでの対応が必要**



第四正規形と第五正規形

- さらに高次の正規形(省略)

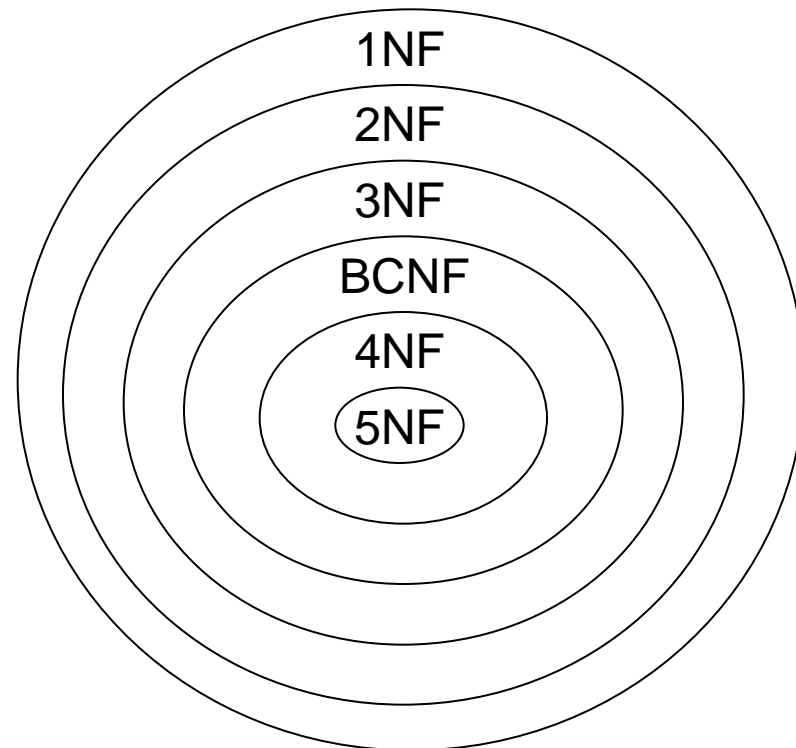
- 第四正規形(4NF)

- 多値従属性に着目

- 第五正規形(5NF)

- 結合従属性に着目

- 各正規形の関係は右図のとおり





正規化の指針

- 3NFまでの正規化は有益であり、**正規化による問題は発生しない**
 - 原則として3NFまでの正規化は実施
 - 3NFに正規化した時点で高次の正規形である場合も多い
- さらなる高次の正規化 (BCNF, 4NF, 5NF) には注意が必要
 - 一部の情報が失われる場合がある
 - アプリケーションレベルでの対応が必要なことも
 - 必ずしも高次の正規化は不要
- 更新が発生しない場合、正規化が不要なこともある