

データベースS 講義資料 第8回 リレーションスキーマの設計 (2)

九州工業大学 情報工学部 システム創成情報工学科 講義担当：尾下真樹

1. 正規形と正規化

リレーショナルモデルには、リレーションが必ず満たさなければならない第1正規形（属性値は単一の値でなければならないという制約）に加えて、第2正規形～第5正規形の**正規形**が存在する（後の正規形になるに従って、より条件が厳しくなる）。これらの正規形の条件を満たすようにリレーションスキーマを設計することで、更新不整合が生じることを防ぐことができる。以下、各正規形の定義、及び、リレーションスキーマが正規形を満たさない場合に、正規形を満たすように複数のリレーションに分解する方法（**正規化**）を学ぶ。

1.1. 第2正規形

第2正規形の定義：

第1正規形を満たし、かつ、全ての非キー属性が候補キーに関数従属する（候補キーの一部の属性のみに関数従属しない）。

例えば、以下のようなリレーションがあるとする。（下線が引かれている属性は主キーを表す。）

営業（商品番号, 顧客番号, 社員番号, 販売価格）

ただし、各顧客について、1社の顧客を担当する社員は1人だけ、というルールがあるものとする。

このリレーションには、顧客番号 → 社員番号 という関数従属性がある。そのため、社員番号は、候補キー {商品番号, 顧客番号} 全体だけでなく、候補キーの一部の属性である顧客番号に関数従属することになり、第2正規形の条件を満たさない。その結果、このリレーションは更新不整合を生じることになる（具体的にどのような不整合が生じるかは前回の資料を参照）。

このリレーションが第2正規形を満たすようにするためには、関数従属性に注目して、以下の2つのリレーションに分解する。

販売（商品番号, 顧客番号, 販売価格） 顧客担当（顧客番号, 社員番号）

顧客番号 → 社員番号 の関数従属性に注目し、{顧客番号, 社員番号} のみを別のリレーションとする。また、もとのリレーションから社員番号を取り除いたものを、もう1つのリレーションとする。分解後のリレーション名は、適当に決めて構わない。

なお、リレーションの候補キーが1つの属性のみからなる場合（上の営業担当リレーションなど）は、候補キーの属性の一部の属性は存在しないため、自動的に第2正規形を満たす。

1.2. 第3正規形

第3正規形の定義：

第2正規形を満たし、かつ、非キー属性が、候補キー以外の属性に関数従属しない。

例えば、以下のようなリレーションがあるとする。（下線が引かれている属性は主キーを表す。）

商品担当（商品番号, 社員番号, 営業所番号）

各社員は1つの営業所に所属する。

このリレーションは、第2正規形は満たす。しかし、このリレーションには、社員番号 → 営業所番号 という関数従属性があるため、非キー属性である営業所番号が、候補キー以外の属性である社員番号に関数従属することになり、第3正規形の条件を満たさない。その結果、このリレーションは更新不整合を生じることになる。

このリレーションが第3正規形を満たすようにするためには、関数従属性に注目して、以下の2つのリレーションに分解する。

商品担当（商品番号, 社員番号） 社員所属（社員番号, 営業所番号）

分解のルールは、第2正規形の場合と同様である。

1.3. ボイス・コード正規形

ボイス・コード正規形の定義：

関数従属性 $X \rightarrow A$ が存在する時、常に X は超キーである。

例えば、以下のようなリレーションがあるとする。（下線が引かれている属性は主キーを表す。）

営業（商品番号, 顧客番号, 社員番号, 販売価格）

各社員は一つの顧客のみを担当するとする。ただし、一つの顧客を複数の社員が担当しても構わない。

このリレーションは、第2正規形・第3正規形は満たす。しかし、このリレーションには、社員番号 → 顧客番号 という関数従属性があり、顧客番号が関数従属性している社員番号は、候補キー（超キー）ではないため、ボイス・コード正規形の条件を

満たさない。(逆に、顧客番号 → 社員番号 の関数従属性はないことに注意。もしそのような関数従属性があれば、第 2 正規形を満たさない。) その結果、このリレーションは更新不整合を生じることになる。

このリレーションがボイス・コード正規形を満たすようにするためには、関数従属性に注目して、以下の 2 つのリレーションに分解する。

販売 (商品番号, 社員番号, 販売価格) 社員担当 (社員番号, 顧客番号)

分解のルールは、これまでの正規形の場合と同様である。ただし、分解前と分解後で主キーが変化していることに注意する。もとのリレーションスキーマでは、{商品番号, 顧客番号} と {商品番号, 社員番号} の両方が候補キーであったが、分解により、前者の候補キーの関数従属性の情報は消失してしまう。

1.4. 第 4 正規形

第 4 正規形の定義：

多値従属性 $X \twoheadrightarrow A$ が存在する時、常に X は超キーである。

例えば、以下のようなリレーションがあるとする。(下線が引かれている属性は主キーを表す。)

プロジェクト (プロジェクト番号, 社員番号, ミーティング日)

プロジェクトごとに社員とミーティング日が決まっているものとする。

このリレーションは、ボイス・コード正規形までは満たす。しかし、このリレーションには、**プロジェクト番号** → **社員番号** | **ミーティング日** の多値従属性があり、プロジェクト番号が候補キー(超キー)ではないため第 4 正規形の条件を満たさない。このリレーションが第 4 正規形を満たすようにするためには、多値従属性に注目して、以下の 2 つのリレーションに分解する。

参加社員 (プロジェクト番号, 社員番号) ミーティング日 (プロジェクト番号, ミーティング日)

1.5. 第 5 正規形

第 5 正規形の定義：

結合従属性が存在する時、分解後の各リレーションスキーマ RS_i は、もとのリレーションの超キーである。

あるリレーションスキーマ RS が、複数のリレーション RS_i に情報無損失分解できるときに、結合従属性が存在する。

例えば、以下のようなリレーションがあるとする。(下線が引かれている属性は主キーを表す。)

部品供給 (工場番号, 部品番号, 業者番号)

工場 f は、部品 p の供給を、 f と契約している業者 s で p が供給可能な全ての業者 s から受ける。

このリレーションは、第 4 正規形までは満たす。しかし、このリレーションには、* ((工場番号, 部品番号), (部品番号, 業者番号), (工場番号, 業者番号)) の結合従属性があり、分解後の各リレーションはもとのリレーションの超キーではないため、第 5 正規形の条件を満たさない。このリレーションが第 5 正規形を満たすようにするためには、結合従属性に注目して、以下の 3 つのリレーションに分解する。

製造部品 (工場番号, 部品番号) 部品担当 (工場番号, 業者番号) 工場担当 (部品番号, 業者番号)

1.6. 正規形のまとめ

第 2 正規形、第 3 正規形、ボイス・コード正規形は、関数従属性にもとづいて判断でき、自明な関数従属性(候補キー属性から、候補キー以外の属性への関数従属性)以外の関数従属性が存在する場合は、いずれかの正規形を満たさないことになる。

このうち、第 2 正規形、ボイス・コード正規形については、候補キーが複数の属性から構成されるときにのみ、満たさない可能性がある。例えば 1.2 節の例のリレーションは、候補キーの属性が一つだけであるため、自動的に第 2 正規形を満たす。

第 4 正規形、第 5 正規形は、自明でない多値従属性・結合従属性が存在するかどうかによって、判定できる。